Object Detection Oriented Feature Pooling for Video Semantic Indexing

Kazuya Ueki and Tetsunori Kobayashi
Faculty of Science and Engineering, Waseda University, Tokyo, Japan

Keywords: Video Semantic Indexing, Video Retrieval, Object Detection, Convolutional Neural Network.

Abstract: We propose a new feature extraction method for video semantic indexing. Conventional methods extract features densely and uniformly across an entire image, whereas the proposed method exploits the object detector to extract features from image windows with high objectness. This feature extraction method focuses on "objects." Therefore, we can eliminate the unnecessary background information, and keep the useful information such as the position, the size, and the aspect ratio of a object. Since these object detection oriented features are complementary to features from entire images, the performance of video semantic indexing can be further improved. Experimental comparisons using large-scale video dataset of the TRECVID benchmark demonstrated that the proposed method substantially improved the performance of video semantic indexing.

1 INTRODUCTION

Today, many digital videos are uploaded to social networks such as YouTube and Facebook. According to 2015's statistics, 400 hours of video are being uploaded to YouTube every minute. For this reason, video semantic indexing are becoming vastly more important.

Video semantic indexing has been studied for many years in the semantic indexing (SIN) task of TRECVID (Smeaton et al., 2006) (Over et al., 2015): TRECVID is an annual benchmarking conference organized by the National Institute of Standards and Technology (NIST). Before 2012, most research groups extracted local features such as SIFT, HOG, and LBP, densely and evenly from a entire image. On the other hand, after 2013, for the high performance reason, the deep learning, especially convolutional neural network (CNN), have been mainly used as the feature extraction, and now CNNs account for most of SIN systems. However, there was no substantial breakthrough over the past two or three years, other than a certain amount of improvement by making CNN's structure deeper (Simonyan and Zisserman, 2014) (Szegedy et al., 2014).

Conventional methods, e.g. local descriptor based feature extraction and CNN based feature extraction, have a common major disadvantage: features are evenly extracted from a entire image. For example, in dense SIFT sampling, features are extracted from both relevant and irrelevant image patches in a uniform manner using a fixed pixel interval between regions. As a result, extracted features are forced to contain redundant information for video retrieval. Similar problem occurs in CNN based feature extraction, because an entire image is directly inputted to CNN and convolution is performed by sliding the filter over the image.

In this paper, by focusing on "objects" using a object detector as a feature extractor, we attempted to remove the useless noise (e.g. background) and add more information (e.g. the position and the size of objects) that were deleted by conventional methods. As for the object detector, we used recently proposed Faster R-CNN (Ren et al., 2015) that is known for the high detection rates and the high computational speed. In our experiments, we confirm that features extracted with the proposed method are complementary to the conventional features and they contribute much to the performance of video semantic indexing by combining with the state-of-the-art feature extractor.

This paper is organized as follows: In Section 2, we describe system perspective of video semantic indexing. In Section 3, we present the proposed method. In Section 4, we discuss experiments to validate the effectiveness of our proposed method. In Section 5, we give our conclusions and suggestions for future research.

Ueki K. and Kobayashi T.
Object Detection Oriented Feature Pooling for Video Semantic Indexing.
DOI: 10.5220/0006099600440051
In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, pages 44-51
ISBN: 978-989-758-226-4

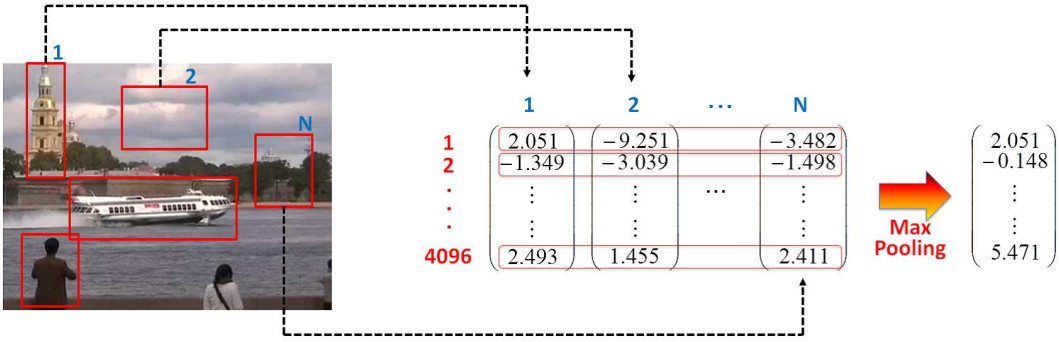

Figure 1: Features are encoded into a single vector by max-pooling.

2 SYSTEM PERSPECTIVE

Generally, the video semantic indexing pipeline of representative methods consists of three steps (Snoek et al., 2015) (Ueki and Kobayashi, 2015):

1. Extracting features from image frame (local feature extraction or CNN-based feature extraction),
2. Classifying the presence or absence of a detection target (with SVM),
3. Fusing results of multiple classifiers (mainly with late fusion).

In this paper, we also utilize the same pipeline.

For feature extraction, the previous mainstream methods were the combination of local feature extractor (e.g. SIFT (Lowe, 1999) (Mikolajczyk and Schmid, 2004) (Lowe, 2004), HOG (Dalal and Triggs, 2005), or LBP (Ojala et al., 1994)) and the encoding (e.g. Bag-of-Features (Csurka et al., 2004) or Fisher Vector (Sánchez et al., 2013)) to obtain fixed length vectors. Recent years, however, because of the very high performance of CNNs trained on the large-scale dataset (e.g. ImageNet (Russakovsky et al., 2015)), CNN's hidden layer outputs have been commonly used also as the feature extraction for video semantic indexing (Snoek et al., 2015) (Ueki and Kobayashi, 2015). In this paper, we exploit CNN based feature extraction as the baseline, and attempt to compensate for the disadvantages of CNN based method with our proposed object detection oriented feature extraction.

Next, to judge whether a detection target exists in an image, a SVM is trained using positive and negative training samples for each target, respectively. There is only a limited number of positive samples, and the number of positive samples in each detection target is generally unbalanced. Therefore, the CNN/SVM tandem architecture is more appropriate than CNN alone directly trained with target data.

Table 1: Twenty object categories used the Pascal VOC.

aeroplane	bus	dining table	potted plant
bicycle	car	dog	sheep
bird	cat	horse	sofa
boat	chair	motorbike	train
bottle	cow	persont	tv monitor

As for the fusion of results acquired from multiple classifiers, the late fusion, namely the score-level fusion, is carried out. In our setting, object detection based (specifically, Faster R-CNN based) feature extraction method is integrated with the CNN based feature extraction method to verify whether these features are complementary or not. Previously, multiple kernel learning (MKL) (Varma and Ray, 2007) was used to combine different types of features. However, MKL has not been used recently, because the score-level fusion is simple and fast, and its perfromacne is comparable to MKL. For this reason, we integrate multiple results by simply summing SVMs' scores.

3 PROPOSED METHOD

3.1 Object Detection Oriented Feature Extraction

The proposed method extracts features using a object detector to obtain complementary features to the conventional CNN based method. The CNN based method extracts features from an entire image, and so it is affected by the useless background information, and other useful information such as the position and the size of objects is excluded. Thus, our object detection oriented feature extraction would extract features from high objectness image region to compensate for these shortcomings.

We chose Faster R-CNN (Ren et al., 2015) from some of object detectors, specifically the pre-trained network on the Pascal VOC detection data (20 object classes as shown in Table 1) (Everingham et al., 2010). Faster R-CNN, as its name suggests, is a fast object detector based on CNNs, and achieves the state-of-the-art performance on the Pascal VOC detection dataset . This includes a region proposal network (RPN) and an object detection network, and so the network is effectively trained end-to-end. When we input an image to the Faster R-CNN, approximately 200 bounding boxes and their probability scores for individual object categories can be obtained. In this paper, a 4,096 dimensional feature vector corresponding to each bounding box is extracted from the first fully-connected layer. That is, we can obtain the set of bounding boxes and their corresponding feature vectors: $\{(b_i, v_i)\}_{i=1}^{N}$, where $b_i = (x_i, y_i, w_i, h_i)$ is the i-th bounding box, that specifies its top-left corner (x_i, y_i) and its width and height (w_i, h_i), and N is the number of bounding boxes in an image.

3.2 Feature Pooling

Here, we explain how to extract a fixed-length feature vector from multiple feature vectors with the Faster R-CNN. The basic method is that feature vectors over all the bounding boxes are bound to one fixed-length feature vector by element-wise max-pooling. That is, the value of the elements in the same dimension are compared across all the bounding boxes, and the maximum value is selected as shown in Fig. 1. This method, however, eliminate the position and the size of objects, and so we attempt to pool feature vectors in the following three ways.

First, we leverage the idea of spatial pyramid matching (SPM) (Schmid, 2006): an image is divided into sub-regions and features are pooled over each image sub-region. We divide an image into three sub-regions; on the upper, in the middle, and at the bottom of the image, and assign bounding boxes to one of three sub-regions based on the center pixel of bounding boxes as shown in Fig. 2. Then a feature vector is created by the max-pooling for each sub-region. Hereinafter, this method is referred to as *spatial pooling*. The SPM generally has to handle very high-dimensional features, because a vector extracted from an entire image and multiple vectors obtained from sub-regions are concatenated. To reduce the computational cost, we separately treat a feature from each sub-region, so that features are fed into the individual SVM training. Using this approach, statistical spatial information can be saved into feature vectors: e.g. "bicycle" and "person" tend to appear in the middle of the image, the background image region is mainly at the top or bottom of the image, or there are only few objects on the upper part of the image, and so on.

Secondly, features are pooled depending on the size of bounding boxes as shown in Fig. 3. Hereinafter this method is called *size pooling*. This method can help distinguish detailed differences for detecting similar targets by treating small, medium, and large sized objects separately. In our experiments, we divide all the bounding boxes equally into three parts by their sizes; small, medium, and large.

Thirdly, features are pooled depending on the aspect ratio of bounding boxes as shown in Fig. 4. Hereinafter this method is called *aspect ratio pooling*. In our experiments, we divide bounding boxes into three; objects that are vertically long, nearly square, or horizontally long.

Object detection oriented feature extraction and these three pooling methods is expected to compensate for information lost by the simple CNN-based feature extractor and improve the performance of video semantic indexing.

3.3 Classification

After extracting pooled features with the object detector, SVMs are trained on task-specific limited training data. This is because the number of positive training samples for each target category is very limited in most categories: for example, there are only several hundred or approximately one thousand samples for each category in TRECVID benchmark data. Therefore, the CNN/SVM tandem architecture is consider to be a better choice than a single CNN trained from scratch on very limited training samples.

The object detector based feature extraction has similar properties. Thus, we also train SVMs with task-specific data after extracting and pooling features.

3.4 System Integration

We explain how to combine the results of both the conventional CNN based and the proposed Faster R-CNN based feature extraction methods. There are mainly two types of fusion methods; (1) concatenating multiple feature vectors to create one feature vector, and (2) computing final scores by simply summing multiple scores from individual SVMs. Because the former method has a problem of computational cost caused by the very high dimensional features, we

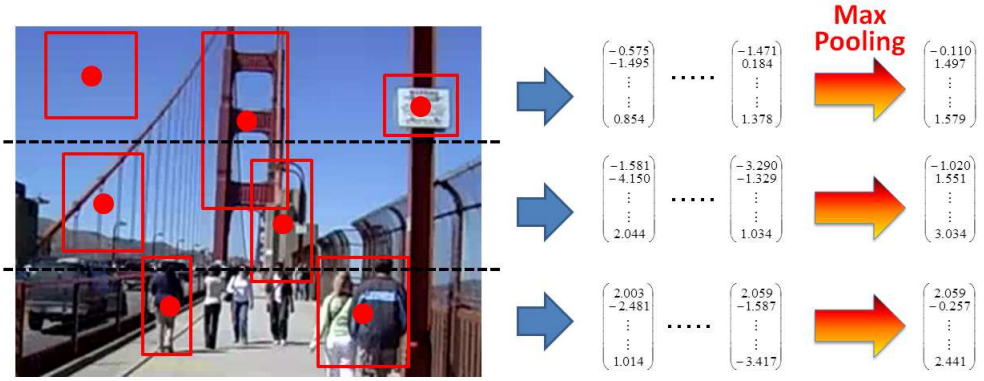

Figure 2: Example of creating feature vectors by spatial pooling.

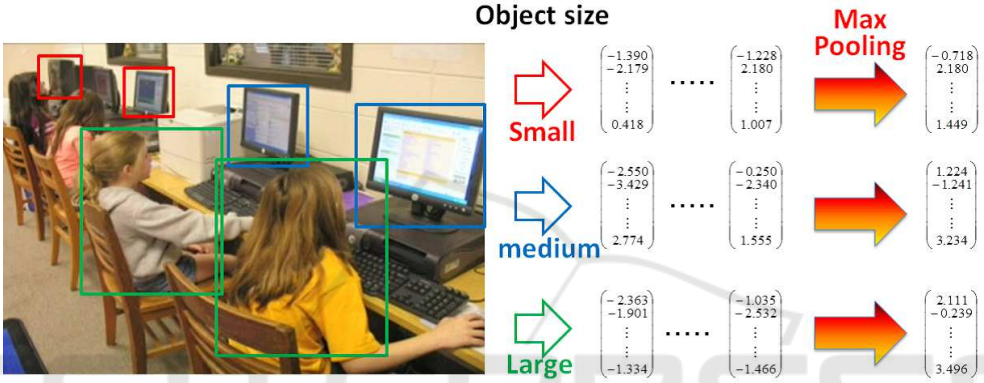

Figure 3: Example of creating feature vectors by size pooling.

chose the latter method[1]. The fusion score is calculated by:

$$s_{\text{total}} = \alpha s_{\text{CNN}} + (1-\alpha) s_{\text{R-CNN}}, \quad (1)$$

where s_{CNN} and $s_{\text{R-CNN}}$ are scores by the CNN based and the Faster R-CNN based methods, respectively. α is the fusion weight having a value from zero to one. The score of Faster R-CNN method $s_{\text{R-CNN}}$ is calculated by summing the score with entire image s_{entire} and scores with divided parts $s_{\text{part}}(i)$:

$$s_{\text{R-CNN}} = \beta s_{\text{entire}} + (1-\beta)\frac{1}{d}\sum_{i=1}^{d} s_{\text{part}}(i), \quad (2)$$

where β is the fusion weight having a value from zero to one, and d is the number of partitions (three in our experiments).

[1]In a preliminary experiment, we found that the latter method showed superior performance.

4 EXPERIMENTS

4.1 Database

We evaluated the proposed method on TRECVID's 2014 SIN task dataset. This video material used in TRECVID SIN task consists of consumer videos from the Internet Archive. Therefore, these videos include not only the detection target, such as objects (e.g., Airplane, Computers, and etc.), actions (e.g., Running, Singing, and etc.), scene (e.g., Classroom, Nighttime, and etc.), but also the huge number of irrelevant data. In TRECVID, participants have to judge whether the target is visible or not at any time within a *shot*. Here, a *shot* is an uninterrupted video clip recorded by a single camera. The average length of each video shot is approximately 5.4 seconds. The TRECVID 2014 dataset includes 549,434 training shots (approximately 800 hours of videos) and 106,913 testing shots (approximately 200 hours of videos).

In addition, a *keyframe*, which is the single video frame image, is assigned in a shot. In our experi-

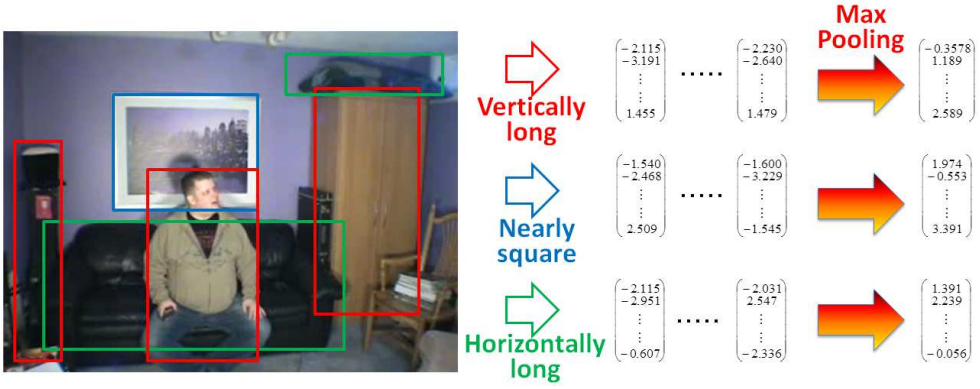

Figure 4: Example of creating feature vectors by aspect ratio pooling.

ments, we used these keyframes to judge whether or not targets exists in a video.

4.2 Evaluation Criteria

We used the same evaluation criterion as one used in TRECVID's SIN task benchmark, namely, the average precision (AP). The AP of each category is defined as

$$AP = \frac{1}{N_{\text{pos}}^{(\text{te})}} \sum_{r=1}^{N^{(\text{te})}} P_r \cdot Rel_r, \tag{3}$$

where $N^{(\text{te})}$ denotes the number of test shots, $N_{\text{pos}}^{(\text{te})}$, the number of positive test shots, and r, the rank in the ordered list of results retrieved from $N^{(\text{te})}$ shots. P_r is defined as the precision computed at the r-th rank and Rel_r takes the value 1 or 0, representing relevant or irrelevant, respectively. Finally, the developed system is evaluated using the mean AP (mAP): the AP scores averaged across all categories.

At TRECVID 2014, the participants evaluated the entire testing set (106,913 shots), outputted their scores for each category, and submitted ranked lists of the top 2,000 shots for each of 60 categories. Finally, 30 of 60 categories were evaluated using the mAP. In our experiments, this truth judgement of 30 categories was used to compare the performance.

4.3 Experimental Conditions

For object detector, we utilized the python implementation of Faster R-CNN, and the pre-trained Zeiler and Fergus (ZF) model (Zeiler and Fergus, 2013).

To train SVMs, provided collaborative annotations (Ayache and Quénot, 2008) (Blanc-Talon et al., 2012) are used. The number of positive shots for each category was approximately 1,750 on average, whereas the number of negative shots was much larger. Therefore, negative shots were randomly subsampled such that the number of positive and negative shots would be 30,000 in total.

After obtaining SVM scores, a re-scoring method, which is called video-clip score (N. Inoue and Shinoda, 2015), was carried out both for the CNN based and the Faster R-CNN based methods to improve the video indexing performance. Specifically, shot scores $s_j (j = 1, 2, \cdots, n)$ of a video that consists of n shots is re-computed as

$$\hat{s}_j = (1-p)s_j + p s_{\max}, \tag{4}$$

where

$$s_{\max} = \max_j s_j, \tag{5}$$

p is the probability of the occurrence of a target category in a video:

$$p = \gamma \left\langle \frac{\#(\text{positive shots in a video})}{\#(\text{shots in a video})} \right\rangle, \tag{6}$$

and γ is a parameter to balance the original score and the maximum score in a video. The final score $\hat{s}_j$ would be close to $s_{\max}$ when the target appears frequently in a video.

For system fusion, we set $\alpha = 0.5$ in (1) such that both the CNN and the Faster R-CNN based methods have the same weight. Because there are four models (features are pooled in a entire image and three divisions), we set $\beta = 0.25$ such that all the models have the same weight. As for the parameter of video-clip scores, we set $\gamma = 0.8$, which was selected by our preliminary experiment.

4.4 Experimental Results

Table 2 shows the APs and the mAPs for both the CNN based method and the fusion of two feature extraction methods. These results show that the CNN based and object detector based feature extraction

Table 2: Average precision for the CNN feature extraction and the combination of CNN and Faster R-CNN.

Detection target	CNN (baseline)	Fusion of CNN and Faster R-CNN		
		Spatial pooling	Size pooling	Aspect ratio pooling
Airplane	23.75	24.40	**24.42**	24.24
Basketball	4.50	**6.02**	5.94	5.84
Beach	52.58	54.93	54.95	**55.06**
Bicycling	14.03	18.76	**19.19**	18.86
Boat_Ship	21.45	22.57	**23.06**	22.72
Bridges	5.30	8.75	8.77	**8.87**
Bus	2.63	**4.50**	4.39	4.42
Chair	20.32	**28.64**	28.39	28.25
Cheering	12.58	**12.98**	12.91	12.80
Classroom	10.50	16.49	16.72	**16.81**
Computers	25.90	33.71	**33.80**	33.65
Demonstration_Or_Protest	30.55	**33.42**	32.75	33.21
Hand	2.25	2.85	**3.04**	2.88
Highway	37.67	38.09	**38.63**	38.31
Instrumental_Musician	**43.32**	41.40	42.03	41.68
Motorcycle	28.66	35.63	**36.03**	35.71
News_Studio	72.72	73.06	72.97	**73.09**
Nighttime	22.72	26.23	**26.50**	26.32
Running	8.75	10.43	**10.47**	10.42
Singing	**14.41**	13.98	14.05	14.09
Stadium	25.71	27.28	27.30	**27.67**
Telephones	3.60	6.01	5.92	**6.14**
Baby	7.01	7.72	**8.20**	7.80
Flags	21.84	**22.34**	21.95	22.29
Forest	28.16	28.50	29.06	**29.10**
George_Bush	55.60	58.47	**59.04**	58.77
Lakes	**9.15**	8.75	8.64	8.51
Oceans	48.66	**49.08**	48.92	48.69
Quadruped	16.98	**24.48**	24.32	24.34
Skier	18.95	26.88	26.71	**27.21**
mAP	23.00	25.55	**25.64**	25.59

methods are complementary and the fusion of these helps improve the performance of video semantic indexing. Especially, categories that achieve high improvement rate by proposed method are closely related to categories of object detectors, namely 20 object classes of Pascal VOC as shown in Table 1. The followings are the examples:

- Chair (TRECVID) $\leftrightarrow$ chair (Pascal VOC),
- Motorcycle (TRECVID) $\leftrightarrow$ motorbike (Pascal VOC),
- Quadruped (TRECVID) $\leftrightarrow$ cat / cow / dog / horse / sheep (Pascal VOC).

We carried out three types of pooling methods in our experiments. However, there is no significant difference in these three methods. This result shows that the Faster R-CNN based method could eliminate the redundant background information and effectively extract features even from the small object region, while the information about the position, the size and the aspect ratio of objects did not contribute a lot to find the target categories.

To investigate the difference between the CNN and the Faster R-CNN based methods, we looked at images in the higher ranks for each method. CNN used in the experiments was trained with ImageNet dataset, and mostly single object is located in the center of images in the ImageNet dataset. For this reason, in the higher ranks using the CNN based method, target objects tended to be large and located in the center of images. On the other hand, in the higher ranks using the Faster R-CNN based method, there were not

Feature extraction with the CNN based method

Feature extraction with the Faster R-CNN based method (Size pooling)

Figure 5: Typical example images of “Airplane” in the higher ranks.

Feature extraction with the CNN based method

Feature extraction with the Faster R-CNN based method (Size pooling)

Figure 6: Typical example images of “Bicycling” in the higher ranks.

only one object but also multiple object in an image, and those objects were relatively small. Typical example images are shown in Fig. 5 and 6.

5 SUMMARY AND FUTURE WORKS

We attempted to exploit the object detector as feature extraction to reduce the useless information derived from redundant background and complement with conventional CNN based feature extraction. Our experiments showed that object detection oriented feature extractor successfully compensates for the information loss by the CNN based method and contributed to the improvement for the video semantic indexing. The improvement rate was high for categories that were related to object detector, and so we plan to create the extensive object detector that can detect various kinds of objects, i.e. increase the number of object categories. Another future work is to effectively use multiple frames in a video instead of using a single keyframe. These information enhancement methods are expected to lead to enhanced visual representation power by being able to treat the combination of multiple objects in a video.

ACKNOWLEDGEMENTS

This work was partially supported by JSPS KAKENHI Grant Number 15K00249 and Waseda University Grant for Special Research Projects 2016A-026.

REFERENCES

Ayache, S. and Quénot, G. (2008). Video corpus annotation using active learning. In *30h European Conference on Information Retrieval (ECIRf08)*, pages 187–198.

Blanc-Talon, J., Philips, W., Popescu, D. C., Scheunders, P., and Zemcík, P. (2012). Advanced concepts for intelli-

gent vision systems. In *Proceedings of 14th International Conference, ACIVS 2012*.

Csurka, G., Bray, C., Dance, C., and Fan, L. (2004). Visual categorization with bags of keypoints. In *Proceedings of ECCV Workshop on Statistical Learning in Computer Vision*, pages 1–22.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338.

Lowe, D. G. (1999). Object recognition from local scale invariant features. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1150–1157.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86.

N. Inoue, T. H. Dang, R. Y. and Shinoda, K. (2015). TokyoTech at TRECVID 2015. In *TRECVID 2015*.

Ojala, T., Pietikäinen, M., and Harwood, D. (1994). Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Proceedings of the IAPR International Conference*, volume 1, pages 582–585.

Over, P., Awad, G., Michel, M., Fiscus, J., Kraaij, W., Smeaton, A. F., Quénot, G., and Ordelman, R. (2015). TRECVID 2015 – An overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2015*. NIST, USA.

Ren, S., He, K., Girshick, R. B., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497.

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

Sánchez, J., Perronnin, F., Mensink, T., and Verbeek, J. (2013). Image Classification with the Fisher Vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245.

Schmid, C. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of CVPR 2006*, pages 2169–2178.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Smeaton, A. F., Over, P., and Kraaij, W. (2006). 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. ACM Press.

Snoek, C. G. M., Cappallo, S., van Gemert, J., Habibian, A., Mensink, T., Mettes, P., Tao, R., Koelma, D. C., and Smeulders, A. W. M. (2015). Qualcomm Research and University of Amsterdam at TRECVID 2015: Recognizing Concepts, Objects, and Events in Video. In *TRECVID 2015*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions. *CoRR*, abs/1409.4842.

Ueki, K. and Kobayashi, T. (2015). Waseda at TRECVID 2015: Semantic Indexing. In *TRECVID 2015*.

Varma, M. and Ray, D. (2007). Learning the discriminative power-invariance trade-off. In *Proceedings of the IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil*.

Zeiler, M. D. and Fergus, R. (2013). Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901.