# Trimps* at TRECVID 2015: Localization and Instance Search

Chuanping Hu, Jianying Zhou, Xuan Cai, Yimeng Li, Xiaoteng Zhang, Jie Shao, Lin Mei

Third Research Institute of the Ministry of Public Security, P.R.China

**Abstract.** This paper introduces our work at the automatic instance search task and localization task of TRECVID 2015 [11]. We introduce localization (LOC) task in details, especially data collection and two methods we introduced: NeC (Negative Clustering) and extra convolutional layers. Then our methods used in instance search (INS) task were summarized briefly.

## 1 Task I: Localization (LOC)

In this section, we introduce our localization task in details. We adopted fast-rcnn framework [1, 2] to do localization, and using VGG-16 model [10] as our baseline.

### Data collection

We collected and labeled data of 11 concepts, including one more extra concept 'person', which is similar with anchorperson. We believe that 'person' category can enforce the model to learn distinguishing anchorperson and normal person. In 11 concepts, each concept has about 1000 samples at least, and about 50000 samples at most, totally about 120 thousands samples. We filtered the image which has not fully annotated 11 concepts. The details of data is described in Table 1.

| **Concepts** | **aero plane** | **anchorperson** | **boat** | **bridges** | **bus** | **computers** |
|---|---|---|---|---|---|---|
| **Images** | 7551 | 3081 | 25349 | 8847 | 6274 | 6435 |
| **Objects** | 21450 | 3333 | 40638 | 10017 | 7866 | 8915 |
| **Concepts** | **flags** | **motorbike** | **person** | **quadruped** | **telephones** | - |
| **Images** | 1078 | 2997 | 47477 | 4518 | 8916 | - |
| **Objects** | 1812 | 3384 | 22569 | 22597 | 12438 | - |

**Table 1.** The images and objects numbers in training data for each concept.

### Baseline

Before training, we used EdgeBox method [3] to do region extraction. After region extraction, about 2000 regions are generated for each image. Regions which has largest IOU with one of the ground truth larger than 0.5 are used as positive samples of this class and regions as negative samples, background class, when their largest IOU less than 0.5 and larger than 0.1

* Contact information: zhoujianyingbest@gmail.com, jieshao.mail@gmail.com

during training. After about 250k iterations, we generated the localization result named as Run ID: Trimps_1.txt.

**Negative clustering (NeC)**

We noticed that, the negative samples have not just one cluster. For examples, negative samples of small IOU with the class 'airplane' objects are not similar with negative samples of small IOU with the class 'person' objects. So we clustered these samples individually. To do this, in our model, we have 23 classes, 11 classes from Table 1 and another 11 classes from the negative samples of the Table 1. We call these 11 classes as 'neg_aeroplane', 'neg_person' and so on. And samples which have IOU with any objects less than 0.2 called 'background' class. That means if the sample has largest IOU with 'person' concept, but the IOU is between 0.2 and 0.5, we labeled this sample as 'neg_person' class, and so on.

Using negative clustering, we improved our results significantly. It is submitted named with Trimps_2_NEG_04.txt. This model can improve the results about 1.9 mAP in our test environment using our training data.

**Extra convolutional layers for localization task**

We introduced an extra convolutional neural network layer before fully connected layer in the model, as deeper network is better. As we mentioned, VGG model was used as the base model to do fine-tuning. After the convolutional layers, a ROI pooling layer was used to split the last convolutional feature map to regions about 128 as the mini-batch. We added a convolutional layer followed the ROI pooling layer. The result is named as Trimps_3_NOC_015.txt. After submission, we found that this model can improve the results by about 0.4 mAP.

And we combine our negative clustering and extra convolutional layer in another model to improve our model. The result was submitted named with Trimps_3_NEG_NOC_045.txt. This got 2 point mAP improved.

**Experiments results**

We summarized our submitted result in Table 2. In Fig 1, we show some results of our localization. And we compared the results of our baseline with last year's localization results in Fig 2. The results shows that, our model improve the localization result obviously.

| | |
|---|---|
| Trimps_1.txt | Baseline model |
| Trimps_3_NOC_015.txt | With extra convolutional model |
| Trimps_2_NEG_04.txt | Negative sample model |
| Trimps_3_NEG_NOC_045.txt | Combine negative clustering and extra convolutional layers |

**Table 2.** Four models and submitted results in localization task.

Figure 3 shows the details of our results in Trimps_3_NEG_NOC_045.txt.

It was notable that we do NOT use multi-model combination is our submissions. We believed that combining several models can further improve accuracy.


**Fig. 1.** Localization show case

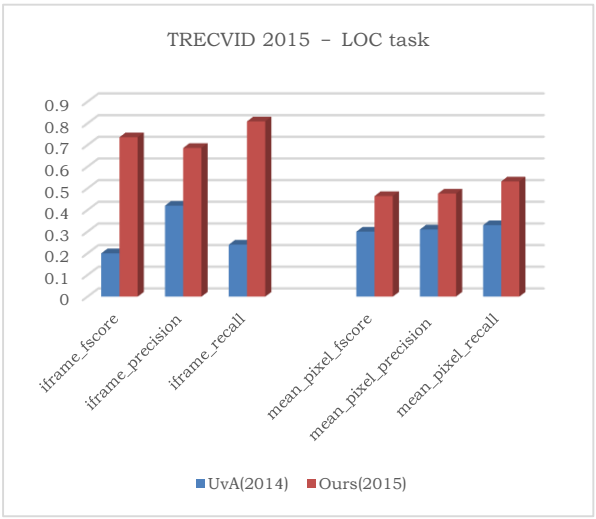


**Fig. 2.** Results from Trimps_1.txt compared with last year' LOC task. **Should note that two concepts were not same between two year's tasks.

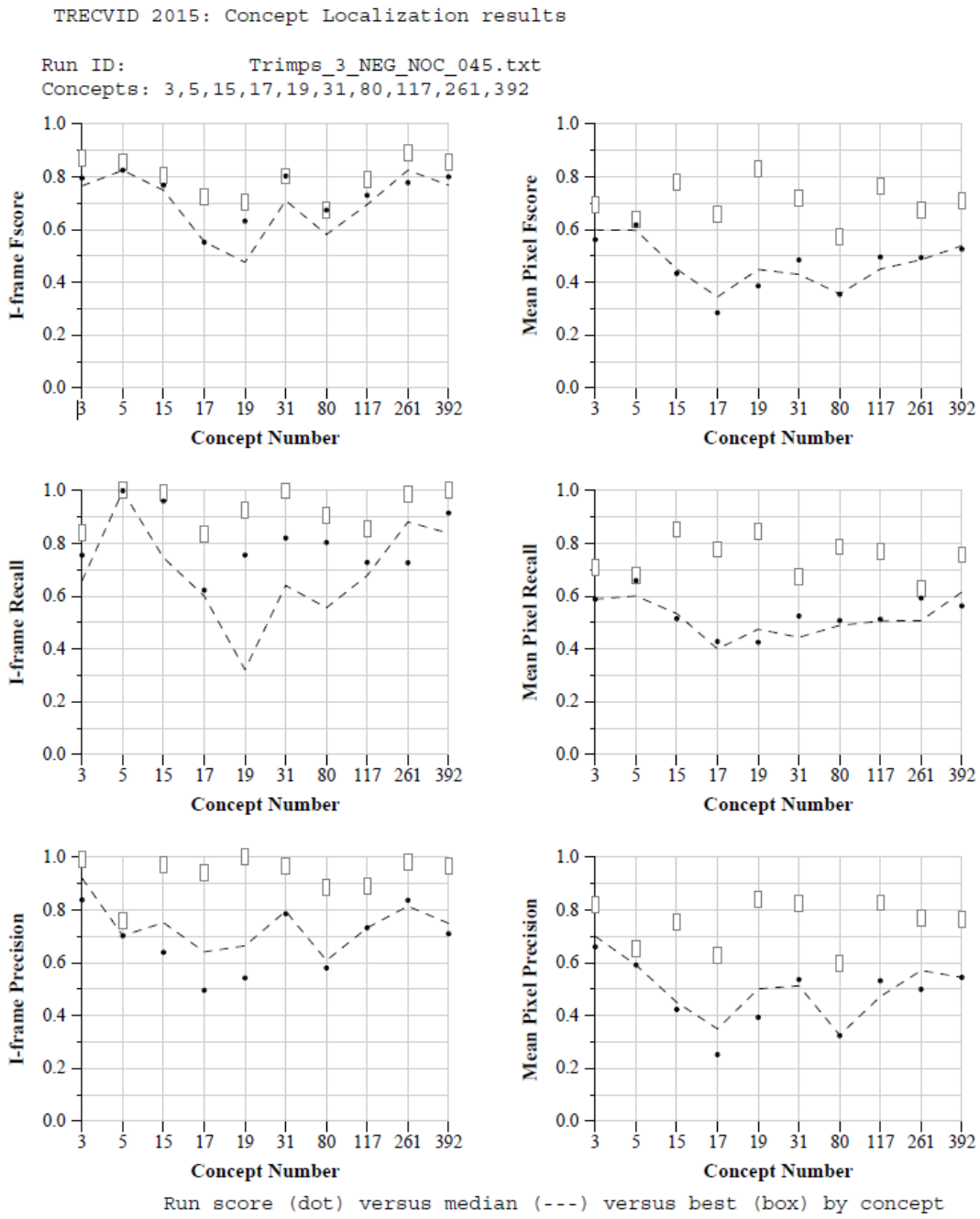


**Fig. 3.** Run ID: Trimps_3_NEG_NOC_045.txt.

## 2 Task II: Instance search (INS)

In instance search task we submitted four results. The results we submitted as shown in Table 3.

To deal with instance search problem, we extract key frames from the 244 given videos and store them in JPEG format. And we get a key frame storage with more than 1M images. Four frameworks are proposed to solve this instance search problem and they are introduced in the submitting order:

| | |
|---|---|
| F_A_Trimps_1 | oppo-SIFT + Streamed-KMeans + FastANN |
| F_A_Trimps_2 | RCNN global features |
| F_A_Trimps_3 | Selective Search + CNN + LSH |
| F_A_Trimps_4 | HOGgles + local features |

**Table 3.** The four models and submitted results in instance search task.

- Using the bag of words (BOW) framework [4]. Firstly, extract oppo-SIFT [5] features from the key images. Then use K-Means algorithm to get the clusters. Since the features' size is too big to put into memory directly, we have to get over obstacles to implement the algorithm. Then use FastANN[6] to generate the codebook based on the features and the clusters. Then build an inverted index based on the codebook. Finally, get all the frames with the similar features as the query image.
- Using Deep Learning global features. Firstly, extract Fast-RCNN features as the reference features from the key frames. Then, compute the global features of the target images, which are provided by the TRECVID community. We utilize the Euclidean metric as the major similarity measurement between a target feature and a reference feature.
- Combining Deep Learning and BOW algorithms. Firstly, using selective search [7] algorithm to find the suspected bounding boxes from reference frames. Then, extract DL features from these boxes. Then, use LSH [8]to get the hash value of each boxes. Then build an inverted index based on these hash values. Then put in the query image and its boxes and get all the frames with similar boxes.
- Using HOGgles[9] localization and SURF features. Firstly, manually choose an interest area from target frames. Then compute the HOGgles image of the interested area and use it as the convolution kernel to localize the area from a key image. Then extract the SURF feature from the target image’s interest area and the key frame’s interest area. Compare the features and compute the similarity between them. We get the 1000 frames with the most similarity of the interest area.

## References


1. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. R. Girshick, J. Donahue, T. Darrell, J. Malik. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
2. Fast R-CNN, Ross Girshick. IEEE International Conference on Computer Vision (ICCV) 2015.
3. Edge Boxes: Locating Object Proposals from Edges, C. Lawrence Zitnick and Piotr Doll. European Conference on Computer Vision (ECCV),2014.
4. Csurka G, Dance C, Fan L, et al. Visual categorization with bags of keypoints[C]//Workshop on statistical learning in computer vision, ECCV. 2004, 1(1-22): 1-2.
5. Van De Sande K E A, Gevers T, Snoek C G M. Evaluating color descriptors for object and scene recognition[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2010, 32(9): 1582-1596.
6. Muja M, Lowe D G. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration[J]. VISAPP (1), 2009, 2.

7. Van de Sande K E A, Uijlings J R R, Gevers T, et al. Segmentation as selective search for object recognition[C]//Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011: 1879-1886.
8. Datar M, Immorlica N, Indyk P, et al. Locality-sensitive hashing scheme based on p-sTable distributions[C]//Proceedings of the twentieth annual symposium on Computational geometry. ACM, 2004: 253-262.
9. Vondrick C, Khosla A, Malisiewicz T, et al. Hoggles: Visualizing object detection features[C]//Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE, 2013: 1-8.
10. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [J]. arXiv preprint arXiv:1409.1556, 2014.
11. Paul Over and George Awad and Martial Michel and Jonathan Fiscus and Wessel Kraaij and Alan F. Smeaton and Georges Quénot and Roeland Ordelman, (2015) TRECVID 2015 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In: TRECVID 2015, NIST, USA.