TRACKED INSTANCE SEARCH

Andreu Girbau[†] *Ryota Hinami*[⋆] *Shin'ichi Satoh*[⋆]

[†] Universitat Politècnica de Catalunya, Barcelona
[⋆] National Institute of Informatics, Tokyo

ABSTRACT

In this work we propose tracking as a generic addition to the instance search task. From video data perspective, much information that can be used is not taken into account in the traditional instance search approach. This work aims to provide insights on exploiting such existing information by means of tracking and the proper combination of the results, independently of the instance search system. We also present a study on the improvement of the system when using multiple independent instances (up to 4) of the same person. Experimental results show that our system improves substantially its performance when using tracking. Best configuration improves from $mAP = \mathbf{0.447}$ *to* $mAP = \mathbf{0.511}$ *for a single example, and from* $mAP = \mathbf{0.647}$ *to* $mAP = \mathbf{0.704}$ *for multiple (4) given examples.*

1. INTRODUCTION

An important need in many situations involving video collections (archive video search/reuse, surveillance, law enforcement, protection of brand/logo use...) is to find images or video segments of a certain specific person, object, or place, given a visual example. To this purpose, instance search is defined as the problem of finding instances of the specific query in a set of images or videos given a visual example.

Typically, an instance search system takes a query given as an image (or images) optionally region specified (rectangle or segment), and returns a ranked list of possible instances of that query from a dataset. A main problem is that, usually, only a single image is provided, and the results become solely dependent to the specific characteristics of that image (pose, illumination, viewpoint...). In other words, the system becomes very dependent towards the matching between the given visual example and the dataset.

For queries coming from videos, we propose to exploit the already existing video data. The objective is to provide more variance to the system, so that it is not so biased towards a single visual example. By applying tracking, we can collect many sample images of the target instance with sufficient variation which may result in better instance search performance. In figure 1 we show the expected mapping of the query expansion by using tracking to the feature space.

In order to merge multiple ranked lists obtained by multiple sample images we propose to use a voting scheme. In this paper we study two possible voting schemes, one assuming dependence between the examples provided by tracking and the other assuming independence between them. We found that, by considering dependence between samples from tracking, our system achieves better performance than considering these samples as independent.

We developed a baseline to test the hypothesis on whether tracking helps or not in the instance search task. We used the TRECVID [1] instance search task for this purpose. From 2016, TRECVID INS task is based on retrieving a specific person in a specific location. This means that the correct person (i.e. the query) will be only tagged as good if he/she is in the specified location. In order to correctly evaluate our method we generated a person ground truth for this purpose based on the TRECVID INS dataset. Such ground truth will be made publicly available to help further research.

Our contribution: we provide insights on the influence of tracking as a generic and automatic query expansion for video instance search, and propose a way to combine the results. Also we show the differences on performance between using queries provided by tracking the original query, independent queries provided in a supervised manner, and the combination of both.

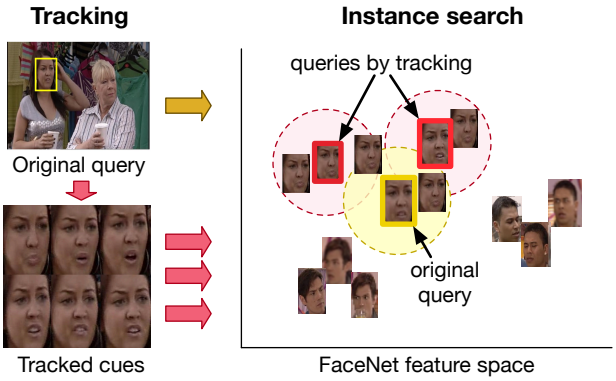

Fig. 1: Single query being expanded by means of tracking. The new query examples coming from the tracked cues introduce new information of the queried person.

2. RELATED WORK

Person search:

Most of the literature on person instance search relies on CNN models to extract face features in order to look for instances corresponding to the original query. [2, 3, 4, 5, 6, 7] use VGG-faces [8] for face feature extraction, [9] maps the face features on a FaceNet [10] embedding, and [11] uses a Faster-RCNN [12] approach.

Multiple queries - Tracking:

Multiple queries are shown to be useful in [13] and [14]. Many query expansion techniques have been tested for the instance search task. [2] uses the first 20 ranks outputted from a first run for query expansion, [5] makes use of a first run too (top 50 ranks) to fine tune a VGG-Face CNN model and use these feature vectors to do the final search. On a more supervised way, [11] annotated every instance of the main characters in an episode of a TV show to train a Faster-

The first author performed the work while at the NII, Tokyo

RCNN, while [7] looked for faces of the actors on the Internet.

In what tracking concerns, [9] and [6] make use of person tracking for query expansion, and [5] uses person tracking for person re-identification. [9] makes use of backward and forward face tracking to provide an average over a FaceNet embedding, and [6] tracks both the original query (for query expansion) and the dataset (for correlating object apparitions).

This work studies the behavior of an instance search system when applying tracking to the given query/queries as a generic way to improve the performance of the system for free, i.e., without requiring any extra data besides the video and the query. Our proposal is to treat every tracking proposal as an independent query, and vote among the results having into account the tracking on the initial query. We study the implications of applying tracking to the instance search problem, and propose several approaches.

3. INSTANCE SEARCH

We developed a person retrieval instance search system following an approach based on [15] and [16]. It extracts region-based CNN features [17] from object candidates to generate the database. They are indexed using product quantization (PQ) and an inverted index [18], which enables to search relevant objects efficiently.

Here, we will differentiate between face feature space generation and querying.

On database generation: first, we sample the dataset videos at 1fps. Then, for each frame, we extract the faces of the people in it by means of a multi-task cascaded CNN [19]. For every face, its features are extracted using FaceNet [10]. These features are clustered to build an inverted index, and the PQ-compressed codes are stored in them.

On querying: given an initial mask or bounding box of the person of interest, his/her face bounding box is detected and the face features are extracted. The distances between these features from the query and the cluster centroids are calculated (in this work we used the Euclidean distance). Then, the top-k closest centroids to the query are chosen, and every face contained in each cluster is compared against the original face query to provide a similarity score to, finally, propose a ranked list of frames.

The above methodology is specified for a single, independent query. To combine multiple queries we refer to our voting scheme in Section 4.3, on a set of independent queries. In short, we combine the resulting ranked list from every query and do a re-ranking based on the final score of each shot. As can be seen in table 1 (No tracking column), using multiple queries of the same person instead of a single one improves, by a huge margin, the results. On average, the difference between using a single query vs multiple queries per person is, for 2 provided queries: $\Delta mAP = +0.098$, and for 4 provided queries: $\Delta mAP = +0.200$.

4. TRACKED INSTANCE SEARCH

In the previous section we stated that multiple queries of the same instance provide, if the multiple queries do not contain a bad query that maximizes a bad retrieval score, a boost in performance.

We want to extend this idea to unsupervised queries, this is, to generate query examples derived from an initial query example from a video. We achieve this by tracking, backward and forward, the original query example. Tracking will provide a new set of occurrences that will correspond to the original query, and thus, diversity among the results from the instance search system. Figure 2 shows our pipeline.

4.1. Tracking

The aim of tracking is to provide diversity to the given query making use of the video information. This is, given an initial query example from a video, track it forward and backward in order to provide more examples of it. For a general case, tracking would come in handy for many obvious reasons, which can be summarized in *automated query expansion*. Our tracker uses [19] to detect faces in a frame and provide their alignment points (eyes, nose and mouth), and [20] to extract the face features.

We first position at the frame of the video corresponding to the given example $q_{n=0}$ ($n \in \mathbb{N}$), and define a temporal window $w \geq 0$ centered over it ($w = 0$ means no tracking). The backward cue $b_{n<0} \geq 0 \in \mathbb{N}$ and the forward cue $f_{n>0} \geq 0 \in \mathbb{N}$ contain the neighboring frames of the given example, where $n \in [-w, \dots, w]$.

We compare the distance between the face feature vector from $q_{n=0}$, $v_{n=0}$, against the feature vector of every face in the frames included in w, $v_n^{(k)}$, where $k \in \mathbb{N}$ corresponds to each face in the frame (as there can be $0 \dots K$ faces in a frame). To do so we use the Euclidean distance. Then, the most similar face per frame is chosen and thresholded, so we get 1 or 0 examples for that frame. The resulting examples for a given query will result in $q \leq 2\dot{w} + 1$ ($2\dot{w}$ for examples provided by tracking + 1 given example).

It is important to state that the sample rate r over n is not irrelevant. Let us define a sample rate $r > 0 \in \mathbb{N}$. n will be sampled as $n \in r[-w, \dots, w]$, e.g. if n is sampled on a rate $r = 1$ over a window $w = 2$ then $n = [-2, 1, 0, 1, 2]$. This will have impact on the variance of the proposed samples by the tracked examples $q_{n\neq 0}$ with respect to the original query $q_{n=0}$. If $r = 1$ the neighboring frames will be the immediate anterior and posterior frames, but if $r = 20$ the neighboring frames will be further away from the given example. This means that the variance of the samples provided by tracking, in a general case, will depend on the sampling rate r, e.g. $r = 1$ might produce little variance over the original given example while $r = 20$ might produce almost independent queries.

Caveat: the tracking is performed on the shot where the given example is provided. This means that, working in a high rate / big window configuration (e.g. $(r = 20, w = 5)$), some frames may fall outside the shot and, therefore, not taken in account. Further research could track over a temporal window without shot time constraint, taking in account the whole video.

4.2. Instance search

The instance search system is described in Section 3. Examples provided by the tracking step are processed independently to produce a set of ranked results (one ranked list per example).

4.3. Voting

Finally, a voting step is proposed in order to combine the results provided by the instance search system. To do so, we have worked on two possible voting schemes, $VoSc\_1$ and $VoSc\_2$. The first one assumes that the visual examples coming from the tracked cues of a provided example are dependent between them, and the second that they are independent. The original given examples are considered always as independent between them.

The goal of video instance search is to find videos from a dataset where a certain object instance appears. In the TRECVID challenge the dataset is generated from a TV show, and these videos are shots from different chapters. Our instance search system returns a scored set of frames where the queried person is likely to appear, while each

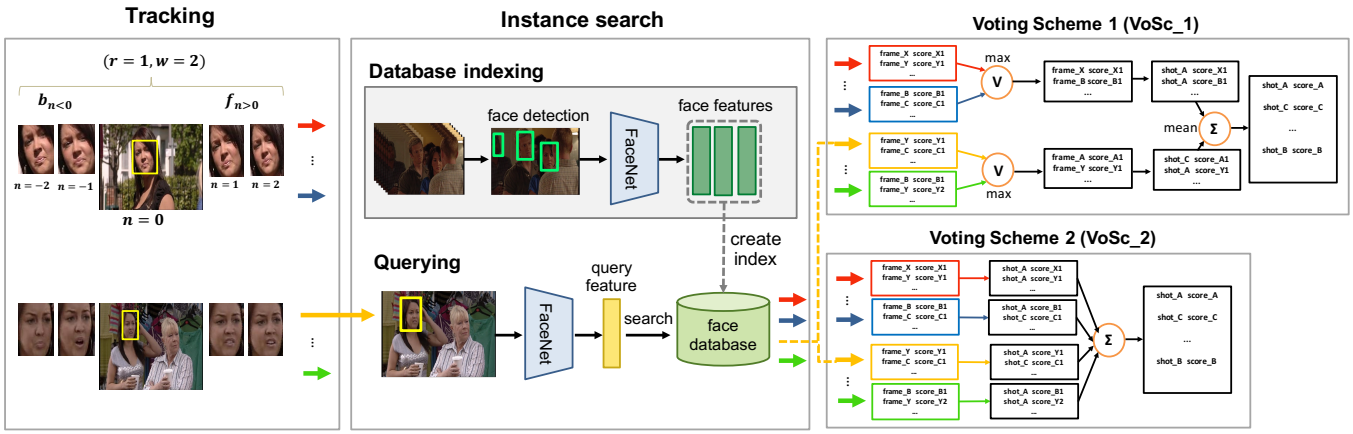

Fig. 2: Our pipeline. First we track the original query example along a defined window, which will propose a set of new query examples. Then the instance search is performed individually for all the proposed query examples (original query example and query examples provided by tracking). Finally, we combine the results of every query example in order to have a ranked list of shots that contain the queried person.

one of these frames correspond to a certain shot in the dataset. The final result is an ordered list of shots depending on their score.

Let us define $s_{n,k}^{(i)}$ as the score of containing the person to be searched for a frame i in the resulting ranked list n corresponding to the tracked cue of a provided example k, where $n \in [-w, \dots, w]$, being $b_q \in [-w, \dots, 0)$ and $f_q \in (0, \dots, w]$ the tracked cues defined in Section 4.1, and 0 the original query. Each of these scores are associated to a single frame in the database without repetition.

Every visual example (original and coming from tracking) generates its own ranked list of frames likely to contain an instance of the query. $VoSc\_1$ assumes dependency between results coming from the same tracked cues, so they are merged to a single ranked list of frames by doing a max pooling (taking the maximum score for every frame). Let $s_k^{(i)}$ be the resulting score list of the combination between $s_{n,k}^{(i)}$ for $n \in [-w, \dots, w]$. Then:

$$s_k^{(i)} = max(s_{-w,k}^{(i)}, \cdots, s_{w,k}^{(i)}) \quad (1)$$

After doing this for every given example k, we then proceed to the combination among them. First, we map every resulting frame i into its corresponding shot u, where $u \leq i$. Then, we combine the shots depending on their score and number of instances retrieved. This combination will result in $s^{(u)}$, which is the resulting score of the shot u. This combination between shots is performed as the sample mean.

$$s^{(u)} = \frac{1}{K} \sum_{1}^{K} (s_1^{(u)}, \cdots, s_k^{(u)}) \quad (2)$$

$VoSc\_1$ follows this two-step voting scheme (merge the results from tracked cues and combine them with other given examples if any), and $VoSc\_2$ only follows the second step (combine all the proposals as if they were all given examples). Here, a discussion on how we consider a query to be independent or not arises. Generally speaking, a query coming from the tracking phase could be considered as independent when it differs largely (in terms of pose, illumination...) within its neighbors. We expect an independent query to produce different results than the queries that are only a little variation of the original query, which should have similar results. In this case, the results in Figure 4 conclude that the tracked cues should not be considered as independent.

5. EXPERIMENTS

Dataset

We used the dataset provided by TRECVID for instance search task. It is composed by 244 chapters of the BBC show *Eastenders*, resulting in 464 hours of video data separated in 471527 shots.

5.1. Ground truth generation

In order to evaluate this work, we generated the ground truth for person retrieval (independent of place) on the TRECVID [1] 2016 instance search task. To do so, we proceeded in the following manner: for every resulting ranked list of shots in every experiment (e.g. $(r = 2, w = 3)$), we evaluated the first 300 results. If the queried person appeared in the resulting shot, we added that shot to the ground truth for the corresponding query. We evaluated the first 300 results for all the combinations between window size (0 to 5) and rate (1,2,5,20). We ended up with a ground truth of 1143 samples per person on average (8006 samples in total), being *Stacey* (query 9174) the query with maximum number of annotated shots (1583), and *Jim* (query 9162) the query with minimum number of annotated shots (409) (see TRECVID INS 2016 task for details). This ground truth will be made publicly available to help further research.

5.2. Tracked instance search

The combination of tracking and voting make the system performance improve. As seen in table 1, by applying our method with an $(r = 2, w = 5)$ configuration, the mAP has a percentage increase of 14.3% for single query tracking. For multiple queries, the system improves by 18% when having 2 provided queries, and by 8.2% for 4 provided queries. In figure 3 different configurations of our method are studied.

Best configuration for each case scenario makes mAP increase by: $\Delta mAP = +0.064$ for a single query $(r = 2, w = 5)$, $\Delta mAP = +0.108$ for 2 queries $(r = 5, w = 5)$, and $\Delta mAP = +0.057$ for 4 queries $(r = 5, w = 2)$.

A discussion on different frame rates arises. We stated along the work that a higher rate should lead to better results due to a larger variation of the produced examples by tracking. In figure 3, rate $r = 5$ outperforms $r = 1$ and $r = 2$ but, clearly, a sampling rate of $r =$

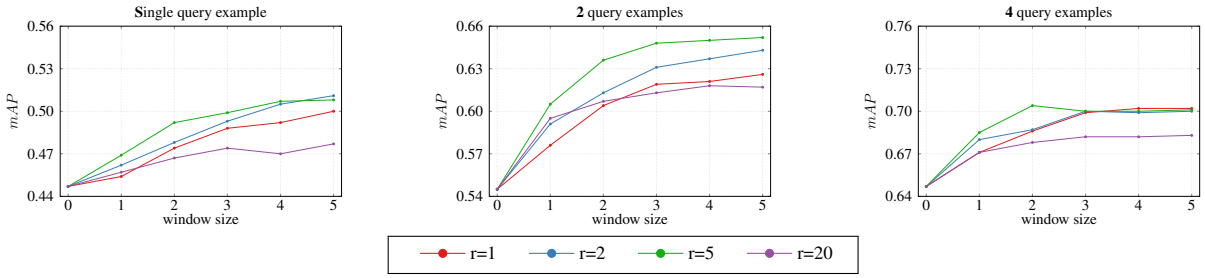

Fig. 3: mAP for different window sizes ($w = 0, 1, 2, 3, 4, 5$), where $w = 0$ corresponds to no tracking, and rates ($r = 1, 2, 5, 20$) for a single given query example (left), 2 provided query examples (middle), and 4 provided query examples (right), using voting scheme $VoSc\_1$.

20 performs way worse than the others. This has two big caveats: first, it is more likely to find the same person's face in closer frames. As we work on a TV show dataset, there are many scene changes where the original query might not be present. If a sampled frame does not contain an instance of the original query (the person to track is not present), we are not able to use that as extra information for the instance search.

Second caveat: as stated in section 4.1, we work on a shot-defined length per query. This means that, exclusively, we track inside the specific shot containing the original query. The reasoning is that we can only be sure that instances of the original query will be present inside that single shot. Further research could explore the possibility on tracking along the video (intra search) until finding a specified number of instances of the original query for the further instance search (inter search).

Examples given	No tracking	Tracking
1	0.447	**0.511**
2	0.545	**0.643**
4	0.647	**0.700**

Table 1: Average mAP with and without tracking for the 4 examples per query provided by TRECVID (2016) INS task. The configuration used is ($r = 2, w = 5$) for tracking and $VoSc\_1$ for voting.

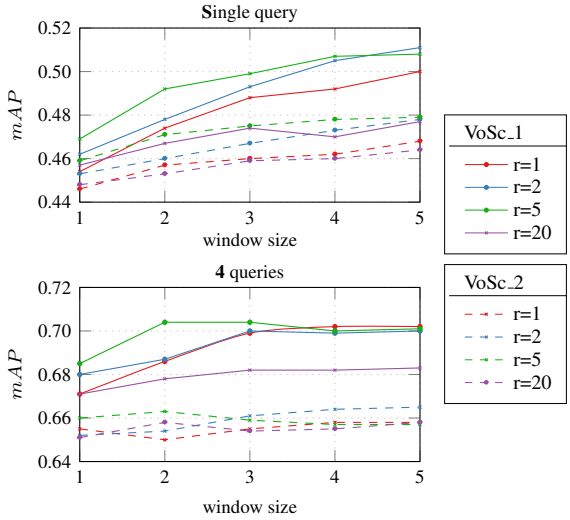

Fig. 4: mAP for a single query example and multiple query examples (4) for two different voting schemes (VoSc_1 and VoSc_2). VoSc_1 considers tracked cues as dependent and provided queries as independent, VoSc_2 considers both tracked and given query examples as independent.

5.3. Voting

In Figure 4 we compare the resulting mAP considering the cues coming from tracking dependent or independent, this is, merging them following the defined two-step voting scheme (*VoSc_1*) or directly mapping each frame into its corresponding shot and combining them by applying directly the second step (*VoSc_2*). Results show that by merging first the results coming from the tracked cues better results are achieved, rather than directly combining the outputs from the instance search for all visual examples (tracked and provided) as they were independent among them.

5.4. On the amount of extra data

The extra data feeded into the system by means of tracking helps to have a relative gain of almost a 15% (1 provided query example) for free (using only the given query and the associated video). It is interesting to note in figure 3 that the system seems to get to a saturation point (for 2 and 4 provided queries), where increasing the amount of data provided by the tracker does not improve the performance at any sampling rate. Further research could explore the performance of the system when using larger windows on a single query case, as it seems that the saturation in that case has not yet been reached, to test if the system could perform as well as having multiple independent queries.

6. CONCLUSIONS AND FURTHER RESEARCH

We have investigated the performance of incorporating tracking to the instance search task, and a way to combine the resulting scored lists. Having a query and its associated video, we can state that using tracking as a generic query expansion the performance of the system can be improved for free (without requiring any more data). We have also observed that multiple independent queries provide a considerable increase of performance. Further research could aim at providing more variable query examples of the same person, so they could be considered as independent queries.

7. REFERENCES

[1] George Awad, Jonathan Fiscus, Martial Michel, David Joy, Wessel Kraaij, Alan F. Smeaton, Georges Qunot, Maria Eskevich, Robin Aly, Gareth J. F. Jones, Roeland Ordelman, Benoit Huet, and Martha Larson, "Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking," in *Proceedings of TRECVID 2016*, 2016. 1, 3

[2] Noel E O'Connor, Jiang Zhou, Eva Mohedano, Alan F Smeaton, Jinhua Du, Haithem Afli, Manuela Hürlimann, Debasis Ganguly, Xavier Giro-i Nieto, Wei Li, et al., "Dublin city university and partners' participation in the ins and vtt tracks at trecvid 2016," in *TRECVID 2016 Workshop*, 2016. 1

[3] Foteini Markatopoulou, Anastasia Moumtzidou, Damianos Galanopoulos, Theodoros Mironidis, Vagia Kaltsa, Anastasia Ioannidou, Spyridon Symeonidis, Konstantinos Avgerinakis, Stelios Andreadis, Ilias Gialampoukidis, Stefanos Vrochidis, Alexia Briassouli, Vasileios Mezaris, Ioannis Kompatsiaris, and Ioannis Patras, "Iti-certh participation in trecvid 2016," in *TRECVID 2016 Workshop*, 2016. 1

[4] Duy-Dinh Le, Sang Phan, Vinh-Tiep Nguyen, Benjamin Renoust, Tuan A Nguyen, Van-Nam Hoang, Thanh Duc Ngo, Minh-Triet Tran, Yuki Watanabe, Martin Klinkigt, et al., "Nii-hitachi-uit at trecvid 2016," in *TRECVID 2016 Workshop*, 2016. 1

[5] Yuxin Peng, Xin Huang, Jinwei Qi, Junjie Zhao, Junchao Zhang, Yunzhen Zhao, Yuxin Yuan, Xiangteng He, and Jian Zhang, "Pku-icst at trecvid 2016: Instance search task," in *TRECVID 2016 Workshop*, 2016. 1, 2

[6] Jin Ye, Linjie Xing, Xiaolong Fan, Changzhi Song, Diping Song, Cai-Zhi Zhu, and Yu Qiao, "Cas, china at trecvid ins 2016," . 1, 2

[7] Zhongling Wang, Jinghao Lu, Yisheng He, Siyuan Li, Bin Xu, and Zhenzhong Chen, "Iipwhu@ trecvid 2016," . 1, 2

[8] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al., "Deep face recognition.," in *BMVC*, 2015. 1

[9] Boris Mansencal, Jenny Benois-Pineau, Hervé Bredin, Alexandre Benoit, Nicolas Voiron, Patrick Lambert, and Georges Quénot, "Irim at trecvid 2016: Instance search," in *TRECVID 2016 Workshop*, 2016. 1, 2

[10] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015. 1, 2

[11] Stefan Kahl, Christian Roschke, Markus Rickert, Daniel Richter, Anna Zywietz, Hussein Hussein, Robert Manthey, Manuel Heinzig, Danny Kowerko, Maximilian Eibl, et al., "Technische universitat chemnitz at trecvid instance search 2016," . 1

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015. 1

[13] Relja Arandjelovic and Andrew Zisserman, "Multiple queries for large scale specific object retrieval.," in *BMVC*, 2012. 1

[14] Cai-Zhi Zhu, Yu-Hui Huang, and Shin'ichi Satoh, "Multi-image aggregation for better visual object retrieval," in *ICASSP*, 2014. 1

[15] Ryota Hinami and Shinichi Satoh, "Large-scale r-cnn with classifier adaptive quantization," in *ECCV*, 2016. 2

[16] Ryota Hinami, Yusuke Matsui, and Shinichi Satoh, "Region-based image retrieval revisited," in *ACMMM*, 2017. 2

[17] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014. 2

[18] Herve Jegou, Matthijs Douze, and Cordelia Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2011. 2

[19] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016. 2

[20] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao, "A discriminative feature learning approach for deep face recognition," in *ECCV*, 2016. 2