PKU_ICST at TRECVID 2019:
Instance Search Task

Yuxin Peng, Zhang Wen, Yunkan Zhuo, Junchao Zhang, Zhaoda Ye, and Hongbo Sun

Wangxuan Institute of Computer Technology,

Peking University, Beijing 100871, China.

pengyuxin@pku.edu.cn

Abstract

We have participated in both two types of Instance Search (INS) task in TRECVID 2019: automatic search and interactive search. For the **automatic search**, we first recognized the action and person separately. Then the two kinds of scores were merged to obtain the results. We achieved *action-specific recognition* from four aspects: frame-level action recognition, video-level action recognition, object detection and facial expression recognition. In the *person-specific recognition*, we adopted the pipeline as follows: query augmentation by super-resolution, face recognition with two deep models, and top *N* query extension. In the *instance score fusion*, we designed a score fusion strategy which aimed to mine the common information from the action-specific and person-specific recognition. For the **interactive search**, the interactive query expansion strategy was applied to expand the queries of automatic search. The official evaluations showed that our team ranked 1st in automatic and interactive search.

1 Overview

In TRECVID 2019, we have participated in both two types of Instance Search (INS)[1] task: automatic search and interactive search. We have submitted totally 7 runs: 6 automatic runs and 1 interactive run. The official evaluation results are shown in Table 1, and our team ranked 1st among all teams in both automatic search and interactive search. Table 2 gives the detailed explanation of brief descriptions in Table 1. The overall framework of our approach is shown in Figure 1.

In the 6 automatic runs, the notations “A” and “E” indicate whether the video examples were used or not. Notation “A” means no video examples were used, while “E” is the opposite. The methods of two runs are the same if there is only a difference of “A” or “E”. Run3_A/E contains all the components of our approach, including *action-specific recognition*, *person-specific* *recognition* and *instance score fusion*. The difference between Run1_A/E and Run3_A/E is that Run1_A/E does not adopt object detection and facial expression recognition for *action-specific recognition*. Compared with Run3_A/E, Run2_A/E does not adopt the top *N* query extension strategy for *person-specific recognition*. Run4 is an interactive search run with human feedback based on automatic search Run3_E.

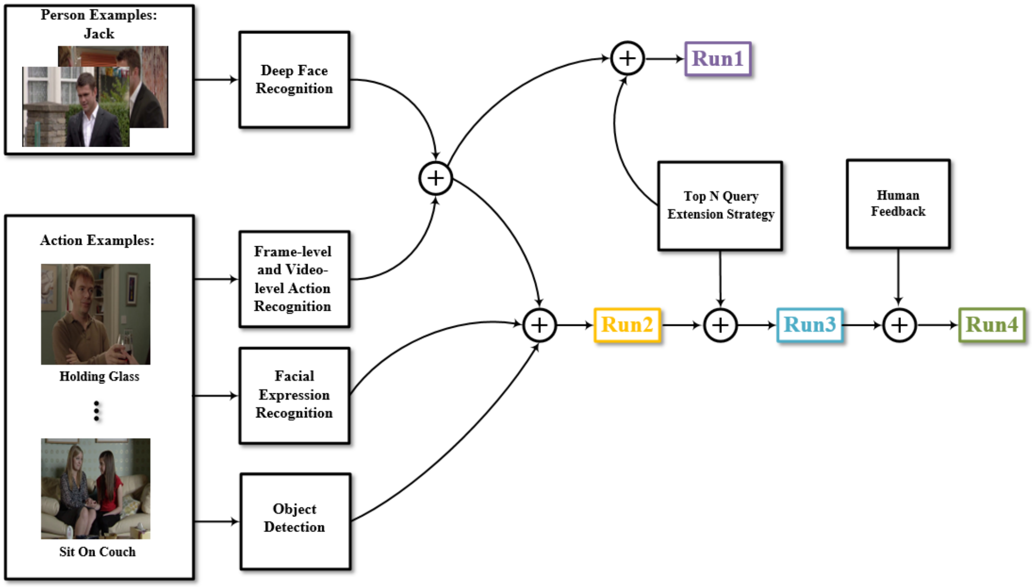

Figure 1: Framework of our approach for the 7 submitted runs.

Table 1: Results of our submitted 7 runs on Instance Search task of TRECVID 2019.

Type	ID	MAP	Brief description
Automatic	PKU_ICST_RUN1_A	0.198	A+F+T
	PKU_ICST_RUN1_E	0.201	A+F+T
	PKU_ICST_RUN2_A	**0.242**	A+O+E+F
	PKU_ICST_RUN2_E	0.230	A+O+E+F
	PKU_ICST_RUN3_A	0.235	A+O+E+F+T
	PKU_ICST_RUN3_E	0.239	A+O+E+F+T
Interactive	PKU_ICST_RUN4	**0.360**	A+O+E+F+T+H

Table 2: Description of our methods.

Abbreviation	Description
A	Frame-level and Video-level **A**ction Recognition
O	**O**bject detection
E	Facial **E**xpression Recognition
F	Deep **F**ace recognition
T	**T**op *N* Query Extension Strategy
H	**H**uman feedback

2 Our Approach

2.1 Action-specific Recognition

There were 12 action categories involved in INS task this year, covering various human activities in daily life, such as “holding glass”, “sit on couch”, etc. The challenges were as follows: (1) Some categories were highly similar and hard to distinguish. For example, the categories “open door enter” and “open door leave” both contained the action of opening the door, but we had to distinguish whether the person entered or left the room. (2) Some actions were very complicated. For example, the action “eating” might happen in various scenes and involve various foods. (3) The training data was insufficient. There were only 4~6 video clips provided for each action category, which were not enough to train the action recognition models.

We considered multiple strategies for action recognition. First, we recognized the action from frame-level and video-level. Second, we exploited object detection techniques to help specific action recognition, such as “holding glass” and “carrying bag”. Third, we also applied facial expression recognition methods to boost the recognition accuracy of some actions, such as “crying” and “laughing”. For addressing the problem of insufficient training data, we resorted to web data and existing external datasets to construct training data. Finally, we computed the average value of the prediction scores of a shot as the final prediction score $ActScore$.

2.1.1 Frame-level Action Recognition

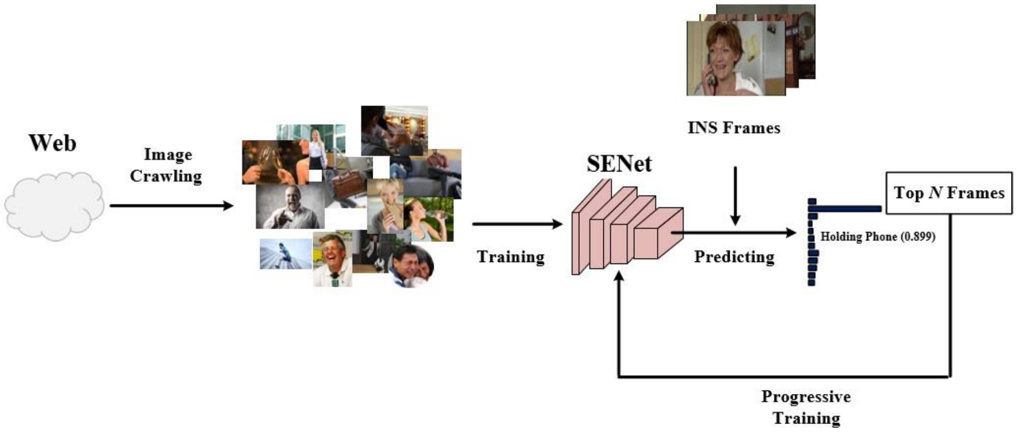

Figure 2: The pipeline of frame-level action recognition.

Frame-level action recognition method aimed to recognize the actions that can be inferred directly from a single frame, such as “crying” and “holding phone”. Figure 2 shows the pipeline of frame-level action recognition method. We first crawled web images to form the training data. Then we trained an image classification model SENet with the progressive training strategy. Finally, the recognition result of a video was derived according to frame-wise prediction scores.

(1) Web Image Crawling

There are large quantities of images on the Internet. According to the official definitions of actions, we utilized several keywords for each action category to collect web images by Baidu[1], one of the widely-used search engines. We collected 5000 images in total with about 400 images for each action category.

(2) Model Training

Considering that some action categories were hard to distinguish due to small inter-class difference, such as "holding glass" and "drinking", we adopted fine-grained image classification techniques to construct the frame-level action recognition model. Fine-grained image classification methods[2][3][4][5] aim to recognize the subcategories that belong to the same coarse-grained category, which usually focus on the vital parts of subcategories to achieve the fine-grained classification. Here, we took a state-of-the-art fine-grained classification model, namely SENet[5], to address the frame-level action recognition.

We adopted the progressive training strategy to train the SENet model. As shown in Figure 2, we first collected web images to train the SENet model, which was then used to perform the frame-level predicting. According to the prediction scores, for each action category, the frames from INS database with top *N* scores were selected to augment the training data. Then we continued to train the SENet model with the augmented training data. The above data augmentation enlarged the scale of training data, as well as forced the distribution of training data to be close to the INS database, thus could improve the recognition accuracy. With the trained SENet model above, we obtained the prediction scores of video frames. Then we took the maximal score of the frames in each shot to predict the shot category.

2.1.2 Video-level Action Recognition

Video-level action recognition aimed to recognize the actions that should be inferred from multiple frames, such as "go up/down stairs". Here, we acquired the training data from Kinetics-400[6] dataset, and applied them to train StNet model[7]. The details are introduced in the following sections.

(1) Training Data Collection

There were only 4~6 video clips provided for each category, which were insufficient to train the deep models. Thus we resorted to the external dataset Kinetics-400 to acquire enough training data. Kinetics-400 is a large-scale dataset for action recognition, which covers 400 human action categories and collects more than 400 video clips for each category. However, not all the action categories in Kinetics-400 are relevant to INS task. To address this issue, we used the data that matched 5 relevant categories in INS task to construct the training data. As shown in Table 3, data with the categories in the right column was regarded as the training data with corresponding

[1] https://image.baidu.com

categories in the left column, along with the video clips provided in INS task.

Table 3: The corresponding relationships of categories between INS task and Kinetics-400.

Categories in INS task	Categories in Kinetics-400
drinking	drinking, drinking beer, drinking shots, tasting beer
eating	eating burger, eating cake, eating carrots, eating chips, eating hotdog, eating doughnuts, eating ice cream, eating spaghetti, eating watermelon
crying	crying
laughing	laughing
go up/down stairs	climbing ladder

(2) Model Training and Testing

We adopted StNet model[7] to conduct video-level action recognition. StNet was constructed based on ResNet[8] backbone, which applied 2D and 3D convolutions to capture the local and global spatio-temporal information. Specifically, we took ResNet-50 network as the backbone. As Kinetics-400 contained too many irrelevant categories of INS task, we fine-tuned StNet model to recognize the 5 categories in INS task to further boost the accuracy. Then we took the model to classify all the video shots in INS database.

2.1.3 Object Detection

We observed that some action categories described the interactions between people and objects, where the objects could be clues to remove a large number of negative shots. Inspired by this, we exploited object detection technology to help action recognition. We adopted Mask R-CNN[9] as the object detection model, pre-trained on MS-COCO dataset[10] with 80 object categories, including “bottle”, “couch”, “handbag”, etc. There were 5 actions in INS task involving the objects included in MS-COCO, and Table 4 details the corresponding relationships between actions and objects.

The Mask R-CNN model was directly exploited to detect objects on all frames. We took the maximal score of objects in each frame as the frame score, and the maximal frame score as the detection score of this shot.

2.1.4 Facial Expression Recognition

Facial expression recognition aimed to recognize the emotion in the human’s facial expression, such as “happy”, “sad”, etc. There were 3 actions in INS task relevant to human’s facial expressions, namely “shouting”, “crying” and “laughing”. Thus, we exploited facial expression recognition techniques to help action recognition.

(1) Training Data Collecting

On the one hand, we exploited the data from two widely-used datasets for facial expression recognition, including CK+[11] and FER2013[12]. On the other hand, we used the web data to augment the training data. We crawled images from Internet by Baidu search engine with text keywords, and then cropped the human faces from them by using face detection model MTCNN[13] to construct the training dataset.

(2) Model Training and Testing

We adopted 19-layer VGGNet[14] model as the facial expression classifier. We fine-tuned the VGGNet model pre-train on ImageNet dataset with the above training data. For testing stage, we detected human faces from video frames and fed the cropped face images into the trained VGGNet model. Similarly, the maximal score of the frames in a shot was adopted as the prediction score of this shot.

Table 4: The corresponding relationships between actions and objects.

Actions in INS task	Objects in MS-COCO
holding glass	bottle, wine glass, cup
drinking	
sit on couch	couch
holding phone	cell phone
carrying bag	backpack, suitcase, handbag

2.2 Person-specific Recognition

In the person-specific recognition stage, we first performed face image super-resolution reconstruction to enhance the quality of person query examples. Second, we combined the scores calculated by two deep convolutional neural networks for face recognition. Finally, we adopted top *N* query extension strategy and text-based search to refine the search results.

2.2.1 Face Super-Resolution of Query Person Examples

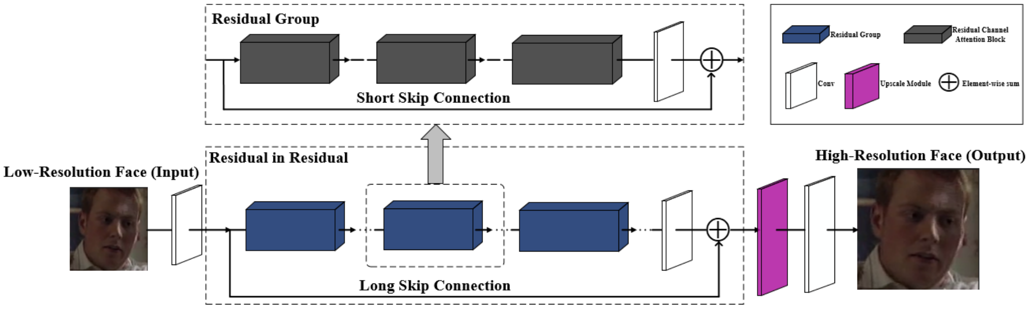

Figure 3: The overview of Residual Channel Attention Networks (RCAN).

We detected faces by MTCNN[15] from the video key frames and query person examples. We observed that some of the detected faces of query person examples were rather blurry due to low resolution, which were difficult to distinguish and recognize. In order to address this issue, we utilize image super-resolution (SR) to recover a high-resolution image from low-resolution image. Specifically, we adopted RCAN[16] model to transform low-resolution images into high-resolution images, pre-trained on DIV2K dataset[17]. The overview of RCAN network structure is shown in Figure 3.

The pre-trained RCAN model accepted a low-resolution image as input, and output the high-resolution one with a constant upscaling factor. We performed face image super-resolution on the given query person examples, which facilitated the search performance.

2.2.2 Face Recognition Based on Deep Models

As mentioned above, we adopted MTCNN[15] to detect the faces in video key frames and query person examples. For face recognition, we adopted Face-Net model[18] and VGG-Face model[19], which were both pre-trained on VGGFace2 dataset[20]. We extracted the feature vectors of all faces by these two models, and calculated the scores via cosine distance between query face and key frame face respectively.

However, not all query faces were actually helpful for person recognition. We considered the outlier of the given query faces as "bad" face, which was quite different from the others. Specifically, we calculated the cosine distance scores s_{ij} between i-th and j-th query faces of specific person, and defined $c_i = \sum_{j \neq i} s_{ij}$ as the confidence score of the i-th query face. The i-th query face of specific person would be detected as "bad" face if $c_i + \theta < \sum_{j \neq i} c_j / 3$, where θ was set to be 0.05 here. After removing the "bad" faces, we calculated the cosine distance scores between query face and shot faces with feature vectors extracted by two deep models respectively. Concretely, we calculated the cosine distance score between the shot i and person j, denoted as $FaceNet\_cos_{ij}$ and $VGGFace\_cos_{ij}$, and then integrated two scores to get the final ranking score as $PerScore_{ij} = FaceNet\_cos_{ij} + VGGFace\_cos_{ij}$.

2.2.3 Top *N* Query Extension Strategy

To make the person ranking more consistent with related topic, we adopted the top *N* query extension strategy. After the first instance score fusion (see Section 2.3 for details), we obtained the top *N* returned shots of each topic. We selected all the faces in the top *N* returned shots, and calculated the mean feature vector as new query feature to perform the iterative query process for further improving the person recognition results to specific topic.

2.2.4 Text-based Search

The text-based search strategy was similar to our approach last year. The video transcripts provided by NIST contained clear clues, which was complemental to visual information. We used the transcripts to perform text-based search on each topic, where the people's information was extended by structured data from Wikipedia web pages (such as nick names, role names, family members of specific people, and the names of his/her closest friends, etc.). For each topic, we generated a list of shots whose transcripts contain the topic keywords. The search results based on text were used to adjust the score in fusion stage with a reward mechanism (See section 2.3 for details).

2.3 Instance Score Fusion

So far, we have obtained the action prediction scores from the action-specific recognition, as well as the person prediction scores from the person-specific recognition. As the instance search task of this year required to retrieve specific persons doing specific actions, we designed a score fusion strategy to fuse the action and person prediction scores in different aspects as follows:

(1) We searched the specified person from candidate action shots. We selected candidate action shots by top *N* (*N* > 1000) ranked shots according to action prediction score $ActScore$ as described in Section 2.1, which had a considerable probability of containing the given action. Then, we proposed a text-based reward mechanism to adjust person prediction score as follows:

$$s_1 = \mu \cdot PerScore \tag{1}$$

where μ is the reward parameter. We set $\mu > 1$ if the shot existed in text-based person search results, and $\mu = 1$ otherwise. In this way, the shots whose transcripts contained the keywords of the query topic would gain higher scores. For those shots not included in top *N* action-specific results, we set $s_1 = 0$. Finally, we re-ranked the candidate action shots according to the score s_1.

(2) We searched the specific action from candidate person shots. We selected candidate person shots by top *M* (*M* > 1000) ranked shots according to person prediction score $PerScore$ as described in Section 2.2, which had a considerable probability of containing the given person. Similarly, we employed text-based reward mechanism to adjust action prediction score and got the score s_2 as follows:

$$s_2 = \mu \cdot ActScore \tag{2}$$

We set $s_2 = 0$ for the shots not in the top M person-specific results and re-ranked the candidate person shots according to the score s_2.

(3) Moreover, in order to integrate action-based ranking and person-based ranking to further improve the search performance, we proposed a fusion strategy based on s_1 and s_2. The fusion score of a shot would be calculated as:

$$s_f = \omega(\alpha s_1 + \beta s_2) \tag{3}$$

where α and β are weight parameters to balance s_1 and s_2, and ω is a reward parameter. We set $\omega > 1$ if the shot simultaneously existed in the top *N* action-specific results and top *M* person-specific results, otherwise $\omega = 1$. The reward parameter ω could help to highlight the common shots of both action-specific and person-specific results, which were more likely to be the right instances. Finally, the obtained fusion score preserved information of both action and person aspects, which further improved the instance search accuracy.

(4) We proposed a time sequence based re-ranking algorithm to refine the fused scores, as we

observed that some long-term actions usually appeared continuously in the adjacent shots. Concretely, the fusion scores of some long term actions shots like “sit on couch” and “carrying a bag” were recalculated by its neighbor shots’ score as follows:

$$s_f^{(i)} = \sum\nolimits_{-T<k<T} s_f^{(i+k)} + \theta_k \tag{4}$$

where $s_f^{(i)}$ denotes the score of i-th shot and $s_f^{(i+k)}$ denotes the score of $(i+k)$-th shot, namely i-th shot’s neighbor shot in time sequence. k is the index gap between these two shots $(-T < k < T)$ and θ is a parameter to adjust the score. We used the adjusted scores to re-rank shots and got the final shot ranking list.

3 Interactive Search

This year, we adopted a similar strategy as what we used in the interactive search task of INS 2018. The interactive search was based on RUN3_E. First, the user labeled positive or negative samples for each topic’s top-ranked results in the automatic search ranking list. Then the query expansion strategy was applied with the labeled positive samples. Note that we only use the samples to conduct the person recognition, because it is observed that the action can be very ambiguous and may bring negative effect. For efficiency, only 10 positive samples were selected in each topic for interactive search. Finally, we merged the scores of expanded and original queries, to obtain the merged score list and discarded the negative samples.

4 Conclusion

By participating in the INS task in TRECVID 2019, we have the following conclusions: (1) Action recognition is a challenging sub-task and plays a significant role for INS task. (2) Objects and facial expressions are important clues to help the action recognition. (3) Human feedback is very useful to boost the accuracy of INS task.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 61771025. For the using of BBC EastEnders video or images snapshots, we thank for the programme material copyrighted by BBC.

References

[1] George Awad, Asad Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Alan F. Smeaton, Yvette Graham, Wessel Kraaij, and Georges Quénot. TRECVID 2019: An Evaluation Campaign to Benchmark Video Activity Detection, Video Captioning and Matching, and Video Search & Retrieval. Proceedings of TRECVID 2019, 2019.

[2] Yuxin Peng, Xiangteng He, and Junjie Zhao. Object-Part Attention Model for Fine-grained Image Classification. IEEE Transactions on Image Processing, 27(3): 1487-1500, 2018.

[3] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Zheng Zhang. The Application of Two-level Attention Models in Deep Convolutional Neural Network for Fine-grained Image Classification. IEEE Conference on Computer Vision and Pattern Recognition, pp. 842-850, 2015.

[4] Xiangteng He, Yuxin Peng, and Junjie Zhao. Which and How Many Regions to Gaze: Focus Discriminative Regions for Fine-grained Visual Categorization. International Journal of Computer Vision, 127(9): 1235-1255, 2019.

[5] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation Networks. IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132-7141, 2018.

[6] João Carreira, and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299-6308, 2017.

[7] Dongliang He, Zhichao Zhou, Chuang Gan, Fu Li, Xiao Liu, Yandong Li, Limin Wang, and Shilei Wen. StNet: Local and Global Spatial-temporal Modeling for Action Recognition. AAAI Conference on Artificial Intelligence, pp. 8401-8408, 2019.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. IEEE International Conference on Computer Vision, pp. 2961-2969, 2017.

[10] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. European Conference on Computer Vision, pp. 740-755, 2014.

[11] Patrick Lucey, Jeffrey F. Cohn, Takeo Kanade, Jason M. Saragih, Zara Ambadar, and Iain A. Matthews. The Extended Cohn-kanade Dataset (ck+): A Complete Dataset for Action Unit and Emotion-specified Expression. IEEE Conference on Computer Vision and Pattern Recognition-Workshops, pp. 94-101, 2010.

[12] Challenges in representation learning: Facial expression recognition challenge: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data.

[13] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters, 23(10): 1499-1503, 2016.

[14] Karen Simonyan, and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image

Recognition. International Conference on Learning Representations, pp. 1-14, 2015.

[15] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499-1503, 2016.

[16] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image Super-Resolution Using Very Deep Residual Channel Attention Networks. Proceedings of the European Conference on Computer Vision, pp. 286-301, 2018

[17] Radu Timofte, Eirikur Agustsson, Luc Van Gool, et al. Ntire 2017 Challenge on Single Image Super-resolution: Methods and Results. IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 114-125, 2017.

[18] Florian Schroff, Dmitry Kalenichenko, and James Philbin. A Unified Embedding for Face Recognition and Clustering. IEEE Conference on Computer Vision and Pattern Recognition, pp. 815-823, 2015.

[19] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep Face Recognition. British Machine Vision Conference, pp. 41.1-41.12, 2015.

[20] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. Vggface2: A Dataset for Recognising Faces across Pose and Age. IEEE International Conference on Automatic Face & Gesture Recognition, pp. 67-74, 2018.