Sensors and Materials, Vol. 34, No. 5 (2022) 1747–1756
MYU Tokyo

S & M 2925

Misleading Video Detection Using Deep Image Retrieval and Dual-stage Confidence Filtering

Yonghu Yang,[1] Cheng-Fu Yang,[2,3*] and Chiang-Lung Lin[1**†]

[1]Dongguan City College, Guangdong 523419, P.R. China
[2]Department of Chemical and Materials Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan
[3]Department of Aeronautical Engineering, Chaoyang University of Technology, Taichung 413, Taiwan

(Received December 30, 2021; accepted March 31, 2022)

Keywords: misleading video detection, image retrieval, ensemble filtering, convolutional neural network

Computer vision technologies have recently been maliciously used to spread misleading information. Because of the low cost of video production, misleading videos have been used for attack ads, criminal fraud, and even political manipulation, which could undermine social progress. Hence, it is important to develop a system for detecting misleading videos that can help a fact-checking center detect misleading videos more efficiently. In this research, we propose a novel video retrieval system based on a deep convolutional neural network that extracts deep visual informatics to retrieve visually alike videos from annotated misleading videos. Moreover, we propose dual-stage confidence filtering that considers both video- and image-level retrieval. This is one of the latest studies on misleading video detection using video-level retrieval, and preliminary experiments demonstrate its superior retrieval performance, enabling it to be applied in real-world applications.

1. Introduction

In recent years, owing to the popularity of the Internet, people are spending increasing time watching videos. These videos may be a 10 min recording from a popular Internet video streaming service such as YouTube or 1 min short footage from a social networking service such as Facebook. While some media sources are reliable and trustworthy, many are randomly propagated throughout the Internet without author information or verification. Also, content on the Internet has become an important influence of the views people form on certain topics and their decisions in daily life.

However, there are many fake videos or reports on the Internet and not all Internet videos are correct. Furthermore, since most people only have limited time to digest the overwhelming amount of information received online, an ever increasing number of misleading videos are produced and propagated with evil intent on the Internet. A famous example was during the 2016 US election.(1) According to Bovet and Makse, among 30 million tweets spread by 2.2 million users, 25% of them were either fake or extremely biased news.(2) This problem has attracted

*Corresponding author: e-mail: cfyang@nuk.edu.tw
**Corresponding author: e-mail: chlulin0510@gmail.com
†Present address: College of Art, Zhongqiao Vocational and Technical University, Shanghai 201514, China
https://doi.org/10.18494/SAM3822

attention from the research community, who have started developing systems to fight against misleading information. Across the globe, organizations are setting up fact-checking centers to verify the accuracy of claims and stories. However, the massive amount of digital content due to the low cost of misleading video production overwhelms the resources of these fact-checking services. It is important to develop more advanced methods of automizing redundant video checking once newly observed videos have been tagged as legitimate or malicious.

In this study, we propose a system for detecting misleading videos, which uses deep convolutional neural networks and novel dual-stage confidence filtering. This novel and important algorithm can increase the robustness of sensing and judge the authenticity of reporting and online videos. The authenticity judgements employ a learned appearance model, meaning that our proposed system is based on machine learning techniques. The main contributions of this work are threefold: 1) This is one of the first studies specifically targeting misleading video detection using video-based retrieval techniques. 2) The proposed algorithm is evaluated on a large self-collected corpus and its robustness and generalizability are demonstrated. 3) The limitations of current studies are pointed out and future potential directions of research are indicated. The rest of the paper is organized as follows. Section 2 provides a literature review on related research, Sect. 3 details our methodology, and Sect. 4 elaborates the experimental results and analysis. Finally, Sect. 5 concludes our findings and future directions of research.

2. Related Research

In this section, we review existing works that are closely related to our study of video retrieval. Video retrieval is usually split into three main blocks: video preprocessing (segmentation), feature extraction and enhancement, and indexing. Video preprocessing is a traditional prerequisite step typically implemented to compress the streaming format of videos into compact and visually representative frames. A method based on principal component analysis (PCA) that takes advantage of the characteristics of the data in video shots has been proposed.[3] Also, a general framework for shot boundary detection was defined by Yuan *et al.* for better optimization.[4] Feature extraction can be further categorized into two aspects: traditional visual descriptors and deep semantic features. Homogeneous texture descriptors, which were investigated by Manjunath *et al.*,[5] and Bag of Visual Words, which was investigated by Shen *et al.*,[6] are commonly applied visual descriptors for retrieval tasks. Recent advances in deep neural networks have provided another viewpoint for describing images. A hybrid of Gaussian mixture model (GMM) supervectors and deep convolutional neural networks (CNNs) was introduced by Inoue and Shinoda.[7] Other studies showed that pure deep features can even be superior to the traditional visual descriptors based on signal processing for retrieval tasks.[8,9] Finally, the Euclidean metric is the most common metric used to measure the distance between images. Other studies on, for example, deep hashing[10] and quantization,[11] have focused on utilizing data structure characteristics to encode descriptors and speed up retrieval. However, none of these studies specifically targeted misleading video retrieval, and we believe that the results of our study could be important in the fight against misleading videos.

3. Methodology

In this section, we describe our methodology, which consists of two stages: system setup (*StageS*) and misleading video querying (*StageQ*). In *StageS*, we preprocess and aggregate all the already annotated misleading videos (*Vm*) through various blocks of data manipulation to build a large data bank (*D*). Then in *StageQ*, a newly incoming video query (*Vq*) is transformed and indexed through the established data bank (*D*) and finally filtered by our dual-stage confidence thresholding to return the final prediction on whether the video is misleading. Each functional block is described in detail in the remainder of this section.

3.1 *StageS*

Figure 1 shows an overview of the system setup (*StageS*). The system is divided into two main blocks, video preprocessing (*P*) followed by feature extraction and enhancement (*F*), which process the video and the image-level granularity, respectively.

3.1.1 Video preprocessing

a) Frame detection

A video is composed of sequential images. During the first phase of *StageS*, we want to keep as much visual information of each annotated misleading video (*Vm*) as possible, because the more visual cues we preserve, the more authentic and reliable the data bank (*D*) is for later verification in the querying phase. However, it is impractical to keep every single frame of every video in a data bank, which would dramatically increase the consumption of memory for video storage and exponentially increase the query time. To deal with this problem, we apply a frame detection technique in the first block of our preprocessing.

The main idea of frame detection is to detect the "key frames", which are only detected when there are obvious frame differences between two consecutive frames. This process can be further separated into two steps. First, we calculate the similarity scores for all consecutive frames. Second, we manually set a threshold and only keep frames with similarity lower than this threshold as key frames that are visually representative for the video. We implement the pixel-based algorithm proposed by Yuan *et al.* for frame detection.[4] An example of four key frames detected from a series of frames is demonstrated in Fig. 2.

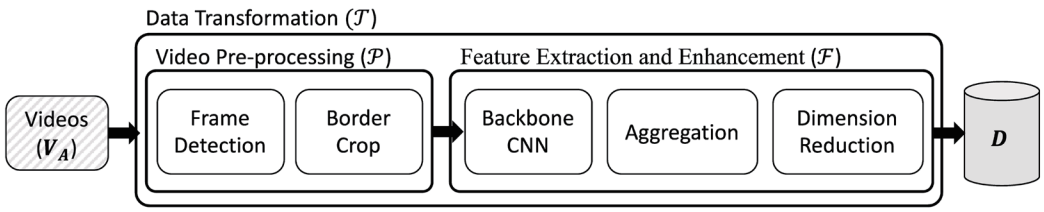

Fig. 1. Flowchart of the system setup phase.

Fig. 2. (Color online) Frame detection results. After the frame detection algorithm, the original (top) frames were reduced to four key frames (bottom) for later applications.

b) Border crop

As another video preprocessing step, we apply a border-crop algorithm, which removes the borders of each video. This step is performed because many videos spread on social networking services (SNS) have been postprocessed, which often includes adding a border. To ensure that our video retrieval system focuses on a video's visual content and to prevent it from being confused by irrelevant artificial borders, we designed a pixelwise border detection algorithm as follows. First, we examine all frames of a video and create a binary mask indicating pixels that do not change throughout 80% of the video. Then we apply a shape detection algorithm to this mask to extract the area of actual video content.[12] Figure 3 demonstrates the result of applying the border-crop algorithm.

3.1.2 Feature extraction and enhancement

After our video preprocessing (*P*) block, all the annotated misleading videos (*Vm*) have been transformed into key frames (images) providing important visual cues. Then two steps are followed for image-level processing. We first transform each image into vectors using CNNs. More specifically, we adopt CNNs that have been pretrained on various image classification tasks and extract the latent feature vectors in the hidden layers of convolutional neural blocks as deep embedding to represent the visual information of each frame.

After the extraction of the deep features, we apply a feature enhancement technique to all of the extracted features from all video frames. There are two advantages of this feature enhancement step. First, it greatly reduces the feature dimension of the extracted embeddings, which can dramatically improve the retrieval speed in *StageQ*. Second, we can map the

Fig. 3. (Color online) Example of postprocessed videos with same visual content and a distinct border. The yellow bounding box is the final cropped video content used for the later setup.

embeddings into more condensed feature embeddings that can improve the visual retrieval accuracy. The final feature enhancement method used is PCA and the final feature dimension after enhancement is 512.[13] At this point, we have finished the setting of *StageS* and collected a large feature data bank *D*. We design our video query algorithm based on this large feature data bank in the following *StageQ*.

3.2 *StageQ*

3.2.1 Video transformation and indexing

In *StageQ*, the main objective is, given an arbitrary query video *Vq*, to utilize this video's visual cues to establish whether *Vq* is in our previously built feature bank *D*. Hence, first we apply the video and image preprocesses *P* and *F* to *Vq*. Then, the extracted image-level embeddings *Iq* are retrieved in our dual-stage confidence filtering system.

3.2.2 Dual-stage confidence filtering

To accurately retrieve potential misleading videos, we propose a novel dual-stage confidence filtering algorithm to improve the accuracy and reliability of image-based video retrieval. First, all query embeddings *Iq* (visual embeddings of key frames) retrieve their l-2-distance-closest images *Ir* from the original data bank *D* as candidate videos. Then two indexes are explicitly defined to filter the candidates:

i) Dominance video index (D_v): For all retrieved images *Ir*, we first map back to their original misleading video (*Vm*) IDs and calculate the dominant video ID among all *Vm*. In the example shown in Fig. 4, we can see that Video(A) occupies the largest proportion among all retrieved

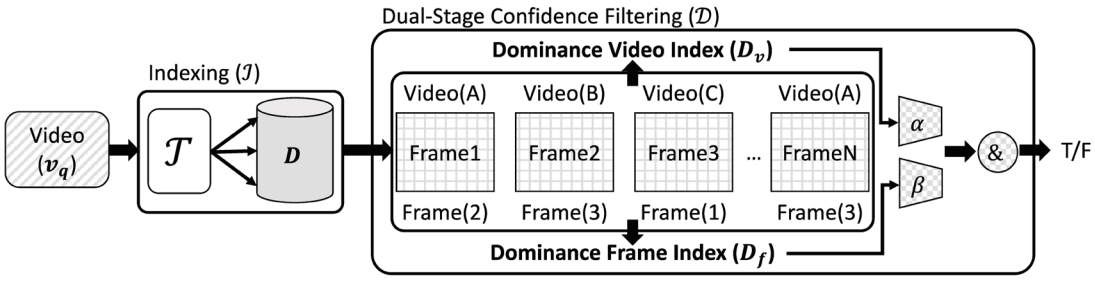

Fig. 4. Flowchart of query phase.

videos of 0.5 (2/4); thus, D_v in this example is 0.5. Note that this index refers to the similarity of the query video *Vq* to the specific video in our previously built data bank *D*, and the larger this index, the more visually alike two videos are. Hence, we set up a system parameter α such that only a query video with $D_v > \alpha$ is regarded as passing the similarity test.

ii) Dominance frame index (D_f): Whereas D_v indicates the similarity between two videos, D_f is an image-level index measuring the similarity between images from query videos *Iq* and those from data bank *D*. Again from the example of Fig. 4, we see that Frame3 has the largest weight among all retrieved images; thus, D_f in this example equals 0.5 (2/4 frames for Frame3). Note that in contrast to D_v, D_f should not be larger than a certain system threshold β. This is because since we have performed frame detection in the video preprocessing, the retrieved frames should be equally distributed among the detected key frames, and a larger value of D_f would indicate unusual results that usually result in incorrect retrieval of a single image. In conclusion, our dual-stage confidence filtering (*D*) can be summarized as follows:

$$\begin{array}{ll} \text{Successfully retrieved,} & \text{if } \left(D_v > \alpha\right) \text{and} \left(D_f < \beta\right), \\ \text{Not retrieved,} & \text{else.} \end{array} \tag{1}$$

A video that passes both stages of confidence filtering is regarded as successful retrieval and returns the retrieved misleading video *ID*; otherwise, the system does not return a video. In the following section, we provide some experimental results of misleading video detection to verify the validity of the system.

4. Experiments

4.1 Experimental setup

4.1.1 Corpus

To verify our designed algorithm, we collect two video corpuses to verify our method:

i) Misleading Video Corpus (*Cm*): This corpus consists of 668 annotated misleading videos with an average length of 112 s. We use this corpus to set up data bank *D* as described in Sect. 3.1.

ii) Internet-Crawled Corpus (*Cc*): The second corpus consists of randomly crawled videos from the YouTube-8M corpus,[14] which we randomly select from top 10 categories representing videos most likely to spread on the Internet that average users may encounter in their daily lives. Note that we use both *Cm* and *Cc* for query videos *Vq* to evaluate the system retrieval results.

Two large image datasets, ImageNet[9] and Place365,[15] are also used for CNN pretraining.

4.1.2 Comparison models

To fully evaluate our designed misleading video retrieval system, three state-of-the-art CNNs are compared for different feature extractions, described in Sect. 3.1.2:

i) VGG1: Proposed in 2014, VGG16 was one of the earliest CNNs to achieve success in an image classification task.[16] We extract the visual feature from the last Global Average Pooling (*GAP*) layer. This model serves as the naïve baseline in our experiments.
ii) ResNet-50: Proposed in 2016, the ResNet series model is one of the most commonly used CNNs in various computer vision tasks.[17] The specially designed residual blocks enable very deep neural network stacking without gradient vanishing during model training. We also extract the visual feature as the last *GAP* layer.
iii) FixEfficientNet-L2: Proposed by Touvron *et al.*, this newer version of EfficientNet targets the potential resolution mismatch between training and testing data.[18] The visual feature is also extracted from the last pooling of the network.

4.1.3 Hyperparameters and evaluation metrics

Several hyperparameters are set as the default: frame detection threshold cutoff = 0.98, feature dimension after PCA = 256, α and β in the dual-stage confidence filtering = 0.95 and 0.6, respectively. Two metrics are designed for final evaluation:

$$\begin{aligned} &\text{Hit}-\text{Rate Accuracy}\left(hAcc\right) = \text{Correctly Predicted Videos/Total Queried Videos},\\ &\text{Retrieval}-\text{Accuracy}\left(rAcc\right) = \text{Correct VideoID/Correctly Predicted Videos}, \end{aligned} \tag{2}$$

where *hACC* indicates the accuracy of predicting whether the query video *Vq* is in *D* or not (a binary problem), while *rAcc* indicates the accuracy of video retrieval (only correctly retrieved video IDs are counted).

4.2 Experimental results

4.2.1 Comparison among different deep features

Table 1 summarizes the results of our system's retrieval of misleading videos under different settings. Note that to fit the real-world usage scenario, in addition to the original query videos (*Vq*) described in Sect. 4.1.1, we augment another Random Resize query video set. In this video

Table 1
Results of misleading video retrieval. Two dataset settings are compared: original and Random Resize, in which the resolution of the query samples is randomly resized by a random ratio (0.5–2). The reported metric is *hAcc*/*rAcc*.

Original	VGG16			ResNet-50			FixEfficientNet-L2		
PCA Dimension	256	512	1024	256	512	1024	256	512	1024
ImageNet	0.721/0.733	0.726/0.742	0.785/0.841	0.772/0.819	0.776/0.821	0.781/0.828	0.808/0.920	0.836/0.925	0.822/0.906
Place365	0.689/0.731	0.708/0.723	0.712/0.723	0.717/0.798	0.744/0.809	0.767/0.817	0.808/0.904	0.813/0.905	0.804/0.903
ImageNet + Place365	0.692/0.731	0.689/0.739	0.741/0.832	0.735/0.809	0.774/0.818	0.777/0.825	0.808/0.914	0.831/0.907	0.804/0.904
Random Resize (×0.5–2)	VGG16			ResNet-50			FixEfficientNet-L2		
PCA Dimension	256	512	1024	256	512	1024	256	512	1024
ImageNet	0.702/0.738	0.715/0.745	0.715/0.745	0.728/0.765	0.735/0.778	0.728/0.770	0.742/0.837	0.775/0.832	0.755/0.819
Place365	0.656/0.719	0.689/0.738	0.695/0.733	0.722/0.779	0.742/0.788	0.735/0.786	0.735/0.806	0.762/0.821	0.748/0.817
ImageNet + Place365	0.677/0.726	0.699/0.742	0.713/0.744	0.724/0.766	0.741/0.779	0.733/0.784	0.741/0.835	0.769/0.825	0.754/0.818

set, we randomly resize the width and length of videos by a factor of 0.5–2 to mimic the potential size variation resulting from video quality issues or manual manipulation. Several observations are made. First, in the original set, we observe that the backbone of the CNN model generally has a dominant impact on the final retrieval results. More precisely, the more advanced CNN model leads to higher retrieval accuracy: the best retrieval pretrained with ImageNet using VGG16 is 0.785/0.841, compared with 0.781/0.828 for ResNet-50 and 0.836/0.925 for FixEfficientNet-L2. On the other hand, the pretrained data also affect the retrieval. Generally speaking, pretraining using the ImageNet dataset outperforms the pretraining using the Place365 dataset, and there is no notable improvement when we jointly pretrain the CNN model with these two datasets together. We hypothesize that since most of our misleading videos are human-related content (i.e., people, daily life scenes), the additional information drawn from different scenes (Place365) will not boost the retrieval rate. Finally, the performance saturates with increasing number of dimensions used in PCA. We conclude that a dimension of 512 gives the optimal trade-off between retrieval accuracy and computational efficiency.

We then observe that, in comparison with the original set, there is an average decrease in *hAcc*/*rAcc* of 5% for the Random Resize set, which suggests that changing the video quality/resolution could deteriorate the retrieval in a real-world scenario. However, again, FixEfficientNet-L2 outperforms the other methods of feature extraction in most of the settings, which indicates that the resolution-aware retraining process introduced in this model leads to a more size-invariant representation, maintaining the retrieval accuracy as high as 0.775/0.832.

4.2.2 Results on different hyperparameters (leading label rate/leading frame rate)

In this section, we explore how our designed dual-stage confidence filtering affects misleading video retrieval under different parameter settings. Figure 5 shows *hAcc* under different parameter settings. First, we can see that increasing the dominance video index threshold α is equivalent to stricter filtering, resulting in our system predicting videos existing in the data bank D with higher confidence, which leads to a higher *hAcc*, which reaches a plateau around $\alpha = 0.7$. On the other hand, the dominance frame index threshold β has a different effect on the system. From the right plot, there is no clear correlation between *hAcc* and β. *hAcc* reaches

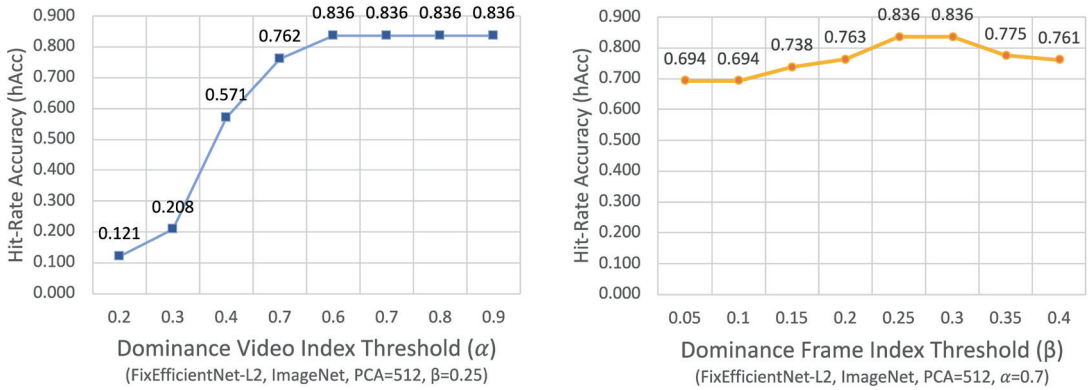

Fig. 5. (Color online) Hit-rate accuracy (*hAcc*) results under different threshold settings in dual-stage confidence filtering. (a) $\beta = 0.2$. (b) $\alpha = 0.7$.

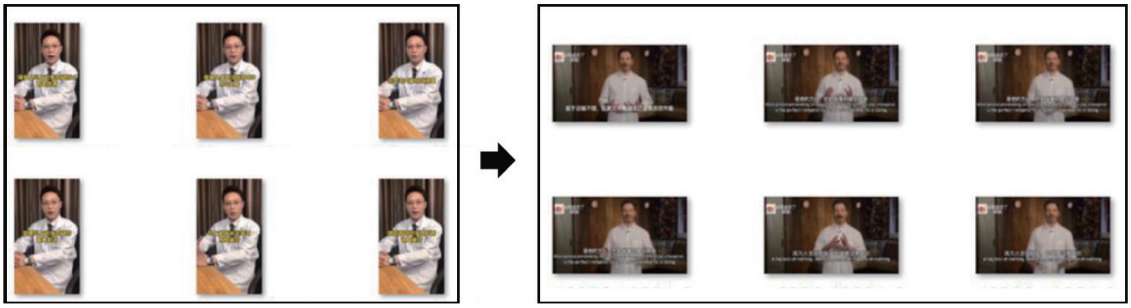

Fig. 6. (Color online) Example of incorrect retrieval when β is set too large.

a maximum value at $\beta = 0.2$–0.3 with no improvement at larger β. Figure 6 shows an incorrectly retrieved example when $\beta = 0.4$. Since the larger the value of β in our filtering system, the looser the constraint of the repeat image retrieval, a larger β leads to multiple visually alike query images being incorrectly retrieved from D and mapped into the same image, resulting in incorrect video retrieval. In conclusion, the experiments have demonstrated that our designed dual-stage confidence filtering with α and β respectively filtering the video-level and image-level retrieval can improve the performance of misleading video detection.

5. Conclusion

In this research, we propose a system for misleading video retrieval to help fact-checking centers efficiently detect maliciously manipulated videos. Experimental results on our collected misleading video dataset demonstrate the robustness and generalizability of our system. Further parameter testing suggests that our proposed novel dual-stage confidence filtering can effectively prevent incorrect detection, particularly in misleading video detection usage. This work has several future directions. For example, our dataset can be enlarged to include more examples of human manipulation (such as reshaping, color filters, etc.) to mimic real-world

video manipulation. Integration with other visual features and indexing methods also has the potential to further improve the detection. We believe that the thorough analysis of malicious videos will improve a variety of fact-checking applications and improve the quality of information in society.

Acknowledgments

This work was supported by project numbers MOST 109-2622-E-390-001-CC3 and MOST 109-2221-E-390-023.

References

1 H. Allcott and M. Gentzkow: J. Econ. Perspect. **31** (2017) 211.
2 A. Bovet and H. A. Makse: Nat. Commun. **10** (2019) 7.
3 D. Lelescu1 and D. Schonfeld: IEEE Int. Conf. Management of Multimedia Networks and Services (2001) 128–141.
4 J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang: IEEE Trans. Circuits Syst. Video Technol. **17** (2007) 168.
5 B. S. Manjunath, P. Wu, S. Newsam, and H. D. Shin: Signal Process. Image Commun. **16** (2000) 33.
6 X. Shen, L. Zhang, Z. Wang, and D. Feng: IEEE 17th Int. Workshop on Multimedia Signal Processing (MMSP, 2015) 1–6.
7 N. Inoue and S. K. Shinoda: ITE Trans. Media Technol. Appl. **4** (2016) 209.
8 G. Kordopatis-Zilos, S. Papadopoulos, L. Patras, and Y. Kompatsiaris: Int. Conf. Multimedia Modeling (2017) 251–263.
9 J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei: 2009 IEEE Conf. Computer Vision and Pattern Recognition (2009) 248–255.
10 Z. Zhang, Q. Zou, Y. Lin, L. Chen, and S. Wang: IEEE Trans. Multimedia **22** (2020) 540.
11 L. Yuan, T. Wang, X. Zhang, F. EH Tay, Z. Jie, W. Liu, and J. Feng: Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR, 2020) 3083–3092.
12 H. Moon, R. Chellappa, and A. Rosenfeld: IEEE Trans. Image Process. **11** (2002) 1209.
13 S. Wold, K. Esbensen, and P. Geladi: Chemom. Intell. Lab. Syst. **2** (1987) 37.
14 S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan: arXiv preprint arXiv:1609.08675 (2016).
15 B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva: arXiv preprint arXiv:1610.02055 (2016).
16 K. Simonyan and A. Zisserman: 2th Int. Conf. Learning Representations (ICLR 2014).
17 K. He, X. Zhang, S. Ren, and J. Sun: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2016) 770–778.
18 H. Touvron, A. Vedaldi, M. Douze, and H. Jégou: ArXiv abs/2003.08237 (2020).