Compact video fingerprinting via an improved capsule net

Li Xinwei, Xu Lianghao and Yang Yi

School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo, People’s Republic of China

ABSTRACT
Robustness, distinctiveness and compactness are the three basic performance metrics for video fingerprinting, and the three factors affect each other. It is challenging to improve them simultaneously. For this reason, an end-to-end fingerprinting via a capsule net is proposed. In order to capture video features, a capsule net, based on a 3D/2D mixed convolution module, is designed, which maps raw data to compact real vector directly. A new designed adaptive margin triplet loss function is introduced, and it can automatically adjust the loss according to the sample distance. It is beneficial for reducing training difficulty and improving performance. Three open access video datasets FCVID, TRECVID and YouTube are composed to train and test, large experimental results have shown that the proposed fingerprinting achieves better performance than traditional and deep learning methods.

ARTICLE HISTORY
Received 23 June 2020
Accepted 5 October 2020

KEYWORDS
Mixed convolution module; end-to-end; adaptive margin triplet loss; capsule net

1. Introduction

With the rapid development of image sensor device, computer and internet, the spread of video becomes more wide and convenient. It facilitates people’s work and lives; however, the uploading of illegal video and copyright infringement occurs at the same time. Some steps, such as video fingerprinting, have been taken to prevent this. Video fingerprinting is a technology which maps video data to short vector. This process needs to meet some conditions, robustness, distinctiveness and compactness. Robustness requires that the fingerprints for original video and its modified copies should be the same or extremely similar. Distinctiveness means that the fingerprints should be different for different videos. Compactness means the fingerprints should be as short as possible for efficient matching. In general, distinctiveness is contradicted to robustness and compactness. How to balance the relationships among them to obtain a better performance is very challenging. The previous works mainly focused on robustness and distinctiveness (Coskun et al., 2006; De Roover et al., 2005; Esmaeili et al., 2011; Guzmanzavaleta et al., 2017; Hu & Lu, 2018; Jiang & Wang, 2016; Kordopatis-Zilos et al., 2017a, 2017b; Lee & Yoo, 2008). Only a few works pursued compactness while keeping the other two performances (Li & Monga, 2013). In this paper we focus on designing a compact fingerprinting while improving the robustness and distinctiveness.

According to the feature extraction process, there are two kinds of video fingerprinting: traditional methods and deep learning methods. The traditional feature is handcrafted based on experience. In the early stage various robust features have been designed, such as Radial hASHing(RASH) (De Roover et al., 2005), 3D-DCT Hashing (Coskun et al., 2006), Centroids of Gradient Orientations Hashing (CGO) (Lee & Yoo, 2008), Temporally Informative Representative Images (TIRI) (Esmaeili et al., 2011), Structure Graphic Model (SGM) (Li & Monga, 2013), Combination of Acoustic and Visual (Guzmanzavaleta et al., 2017). These features are extracted from either key frames or video clips, and the methods based on clips, such as 3D-DCT Hashing, TIRI and CGO, remedied the dropping of temporal information. In general, these methods are well designed and received satisfactory results for one aspect at that time.

Currently due to deep neural network having been applied in many visual tasks successfully, deep learning-based features are also introduced into multimedia fingerprinting (Hu & Lu, 2018; Jiang & Wang, 2016; Kordopatis-Zilos et al., 2017a, 2017b). Jiang et al. (2016) studied the fingerprinting based on standard CNN and Siamese CNN, they designed efficient network to save computation time and received better performance than traditional methods. Kordopatis-Zilos et al. (2017a) proposed layer-based CNN(CNN-L) feature aggregation scheme, which extracted intermediate CNN features to
aggregate histograms with the idea of bag-of-words model. At the same time, they also designed a fingerprinting with deep metric learning (DML) (Kordopatis-Zilos et al., 2017b). Both methods obtained state-of-the-art performance compared to competitive methods at that time. In order to extract temporal information well, Hu et al. (2018) combined the CNN features and long short-term memory unit to represent the spatial–temporal information. This method showed better performance than Siamese CNN. However, the stages of feature extraction and quantization are independent for all the existed deep features-based video fingerprinting, which leads to the mapping function is local optimized. In fact, there are several stages, such as feature extraction, feature compression, binary quantization, for all the methods based on handcraft features even most CNN features. Independent stages mean local optimization; this could be enhanced through integrated mapping function and global optimization. So we will design an end-to-end fingerprinting with an improved capsule net.

In order to overcome the deficiency of CNN’s presentation capabilities, Sabour et al. (2017) proposed the capsule net which describes entities with vector neurons. Being different from traditional CNN, the capsule net intends to describe entities via the relationship between parts and whole. Finally, they obtained well and compact representations for entities with few net layers. However, because there is forward clustering iteration nested in the backward propagation, it is hard to expand to video data directly for huge computation. Since CNN features have been published, a large number of CNN networks were proposed to extract image features among which 3D CNN is a typical net for extracting video features (Tran et al., 2015). In order to reduce computation cost and compress features, a mixed 3D/2D CNN is designed (Zhou et al., 2018). Inspired by this, we will design an efficient capsule net with 3D/2D module.

During fingerprinting matching, traditional methods measure the distance or similarity between samples with handcraft functions, but it is difficult to design well for a specific problem. Metric Learning can automatically learn a metric from data (Bellet et al., 2013), which were employed in video fingerprinting usually (Hu & Lu, 2018; Kordopatis-Zilos et al., 2017b). In this paper we will use a triplet net and design an adaptive margin triplet loss.

### 2. Methodology

There are two main parts for the proposed video fingerprinting, including the weight sharing triplet net and triplet loss function. The main branch is 3D/2D mixed capsule net, which realizes the capsule computation on video data. Triplet net needs triple inputs, which are anchor sample, positive sample and negative sample. The triplet loss function will impel the net to narrow down the positive pair distance and widen the negative pair gap. The structure diagram is shown in Figure 1, and its details will be discussed in the next sections.

#### 2.1. Capsule net construction

Capsule net has shown the powerful abilities of object representation and feature compression, which is suitable for constructing compact fingerprinting. But there is a clustering iteration in the error backward propagation, and its computation is too huge to act on 3D video data. Being different with still image, there is temporal information among video frames, how to extract exactly and represent compactly it is challenging. The very natural idea is exerting 3D-DCT on video to capture spatial–temporal information, but the difficulty is huge computations as before. We pay particular attention to this problem and design a fast capsule net for video data.

The capsule net is composed of three parts: mixed convolution module, primary capsule layer and advanced capsule layer. The parameters are detailed in Table 1. The parameters after the filter represent kernel size and channel number.

For the mixed convolution module, there are three convolution layers and one average pooling layer. Tanh is employed to be the activation function for each layer. The former two layers are 3D convolution with kernel size $3 \times 5 \times 5$ and $3 \times 3 \times 3$ to extract spatial–temporal information, behind that following an average pooling along temporal dimension. The third layer is 2D convolution with kernel size $9 \times 9$. In this way, the computing cost is reduced obviously.

#### Table 1. Parameters for a Capsule net.

| Layer                  | Configuration                               | Output size |
|-----------------------|---------------------------------------------|-------------|
| Mixed convolution     | Input                                       | 64 $\times$ 56$^2$, 3                        |
| Conv1...3D            | filter $3 \times 5 \times 5.64$             | 32 $\times$ 28 $\times$ 28, 64               |
|                       | stride $2 \times 2 \times 2$               |             |
|                       | padding $1 \times 1 \times 1$              |             |
| Conv2...3D            | filter $3 \times 3 \times 3.128$           | 32 $\times$ 28 $\times$ 28, 128             |
|                       | stride $1 \times 1 \times 1$               |             |
|                       | padding $1 \times 1 \times 1$              |             |
|                       | kernel size $3 \times 3 \times 3$          | 28 $\times$ 28, 128                          |
| Average pool          |                                             |             |
| Conv3...2D            | filter $9 \times 9,256$                    | 20 $\times$ 20, 256                          |
|                       | stride $2 \times 2$                        |             |
|                       | padding 0                                 |             |
|                       | (filter $9 \times 9,64$)                   | 2304 $\times$ 8                              |
|                       | stride $2 \times 2$                        |             |
|                       | reshaping                                 |             |
| Primary capsule layer  | Weight matrix $8 \times 32$                | 2304 $\times$ 32                             |
| Advanced capsule layer|                                             |             |
| Dynamic routing       |                                             | 32         |
The function of primary capsule layer is convolution computation and dimension transformation to compose initial capsules. There are 8 groups of 2D convolution with kernel size $9 \times 9$ and 64 channels exerted on the former output. Eight groups of feature maps with $64 \times 6 \times 6$ will be derived, and then each cube is reshaped to $1 \times 2304$ linear vector. Eight corresponding elements are extracted from 8 vectors to compose of 2304 capsules. In fact, the primary capsule is composed of 8 neurons in this paper.

Being similar with primary capsule, the advanced capsule is composed of 32 neurons, which is a highly compact representation for a 3D video data at last. There are two crucial issues for primary capsule with 8-dimension transforming to advanced capsule with 32-dimension. First is a $8 \times 32$ transformation matrix, which increases the dimension. Second is dynamic routing, which is a clustering process in essence (Sabour et al., 2017). After the matrix transformation and dynamic routing, a capsule fingerprint with 32-dimension is received. The details of dynamic routing process can be found in the work (Sabour et al., 2017).

### 2.2. Triplet loss with adaptive margin

In the metric learning methods, Siamese network and triplet network are employed usually. In the Siamese network, the positive pairs and negative pairs are used to be inputs. Because one pair is fed into the net each time during training, only the absolute similarity is exerted on the net. In fact, the sample similarity is continuous rather than yes or no. In order to solve this problem, triplet samples are fed into the triplet network each time, thus the relative similarity is introduced. Generally, the formula for triplet loss $T$ is

$$T = \frac{1}{n} \sum_{t=1}^{n} \max\{0, ||v_t, v_t^+||_2 - ||v_t, v_t^-||_2 + \beta\}$$

(1)

where $v_t, v_t^+, v_t^- \in \mathbb{R}^l$ represent the fingerprint vector corresponding to the sample of anchor, positive and negative, $l$ is the fingerprint length, $n$ is the triple number, $\beta$ is a constant. Formula (1) will explain the positive similarity is higher $\beta$ than negative. However, the relative similarities are different for all kinds of samples. Hard $\beta$ will lead to inefficient or non-convergence network training. In addition, the positive similarity may be lower than the negative pair when neglecting absolute similarity. Thus it will cause precision reduced. Figure 2 shows the distances among triple samples.
Considering these issues, we design a new adaptive triplet loss $L$. The absolute similarity, dynamic similarity and hard threshold are integrated. Its formula is

$$L = \frac{1}{n} \sum_{t=1}^{n} \max \{0, S_t^+ - S_t^- + \beta\}$$

$$= \frac{1}{n} \sum_{t=1}^{n} \max \{0, S_t^+ - S_t^- + \lambda \cdot \sigma(S_t^+ - S_t^-) + ||1 - S_t^+||_2\}$$

(2)

where $\sigma(\cdot)$ is the sigmoid function, $\lambda$ is a scale factor, which is set to 0.8 in this paper. $S_t^+$, $S_t^-$ are Cosine similarities corresponding to positive and negative pair, respectively. Their formulas are denoted as

$$S_t^+ = \frac{1}{2} + \frac{v^+_t \cdot v^+_l}{2 \cdot ||v^+_l||_2 \cdot ||v^+_t||_2}$$

(3)

$$S_t^- = \frac{1}{2} + \frac{v^-_t \cdot v^-_l}{2 \cdot ||v^-_l||_2 \cdot ||v^-_t||_2}$$

(4)

For formula (2), the loss function will push the gap between $S_t^+$ and $S_t^-$ to vary forward to $\lambda \cdot \sigma(S_t^+ - S_t^-)$ and $||1 - S_t^+||_2$. At the same time, $S_t^-$ will be close to 1 with the hard constraint $||1 - S_t^+||_2$. Thus the factors, including absolute, relative and dynamic, are all involved. Its effectiveness will be shown in the experiment section.

### 2.3. The process for training and testing

There are two stages for the whole fingerprinting, training and testing. Figure 3 shows the flow chart. The details of training and testing process are listed below.

**ALGORITHM 1** Training process.

Input: Training data set.
Output: Network parameters.

Step 1. Video pre-treatment. Rescale the video with size $64 \times 56 \times 56$ and transform to YCrCb format.

Step 2. Parameters initialization. Initialize the kernel filter, transform matrix with random data.

Step 3. Select $n$ triples randomly from the training set to fed to the net, compute the advanced capsule vector through forward propagation.

Step 4. Compute the loss according to formula (2), and update the network parameters with SGD optimizer.

Step 5. Repeat Step 3, Step 4 800 iterations in each epoch, until the loss keeps unvarying.

### 3. Experiments

#### 3.1. Experiment setting

All the experiments are completed with the computer by the following settings: CPU is Intel Core i7-8700 K @ 3.70 GHz 6-Core, GPU is NVidia GeForce RTX 2080 8GB and memory is 32GB.

The dataset is composed of training set and testing set. There are 4000 videos in the training set, which are randomly captured from FCVID (Jiang et al., 2018). There are 801 videos in the testing set, in which 1800 videos come from YouTube (Test set and Matlab) and 201 videos come from TRECVID (TRECVID).

Each video of the dataset is transformed with 8 kinds of distortion. The specific parameters are listed in Table 2. These distortions are usually displayed in the real scene. These videos and their copies are used to train and test the proposed network.

For the network parameters setting, the hyper parameters are set as follows. The iteration is set to 800 for each epoch, and batch-size, momentum factor and initial learning rate are 10, 0.9, and 0.01, respectively. The advanced capsule dimension is 32, and the dynamic routing iteration is 3.

| Distortions | Parameters |
|-------------|------------|
| Loss frame  | Dropping randomly 30% |
| Median filtering | Filter size $9 \times 9, 15 \times 15, 19 \times 19$ |
| Insert logo  | Ratio 15%, 18%, 20% |
| Insert subtitles | Typeface size 1.5 |
| Gaussian noise | Mean 0, variance (0.01, 0.05, 0.1) |
| Gamma adjust | Gamma (0.6, 0.8, 1.2, 1.4) |
| Rotation and Cropping | Rotation degree (5, 8, 10) |
| Insert logo and Loss frame | ratio 20% - dropping 25% |

### 3.2. Performance evaluation metrics

The Receiver Operating Characteristic (ROC) is used to evaluating the fingerprinting performance usually. In this paper we will plot ROC curves to display the results. The miss probability $P_M$ means the ratio of missed true copies to all true copies which reflects the robustness. The false alarm probability $P_F$ means the ratio of mistaking copies to all negative copies which reflects the distinctiveness.
Their formulas are

\[ P_M(\alpha) = \Pr(\text{HD}(b, b^{\top}) > \alpha) \]  

\[ P_{FA}(\alpha) = \Pr(\text{HD}(b, b^{\top}) < \alpha) \]  

where \( \alpha \in [0, \text{HD}_{\text{max}}] \), \( \text{HD}_{\text{max}} \) is the maximum of hamming distance. \( F_1 \) score can display a comprehensive evaluation, and its formula is

\[ F_1 = \frac{2 \times (1 - P_M) \times (1 - P_{FA})}{(1 - P_M) + (1 - P_{FA})} \]  

3.3. Experimental results and analysis

In order to prove the effectiveness of proposed fingerprinting, the traditional and deep learning methods are compared. The traditional methods include RASH (De Roover et al., 2005), TIRI (Coskun et al., 2006), CGO (Lee & Yoo, 2008) and SGM (Li & Monga, 2013). The deep learning methods include CNN-L (Kordopatis-Zilos et al., 2017a), DML (Kordopatis-Zilos et al., 2017b) and CNN + LSTM (Hu & Lu, 2018). All the parameters for these methods are set as their literatures to get best performance. Figure 4 shows the ROC curves of these methods for kinds of distortion. The horizontal axis indicates false alarm probability \( P_{FA} \), and the vertical axis indicates miss probability \( P_M \). The point \((0, 0)\) is the best ideal case, so which is close to this point means well performance. It can be seen from Figure 4(a–h), the red curves show the best, which is corresponding to our methods. There is an obvious gap between traditional methods and deep learning methods. It demonstrates deep learning features are better than handcraft features. From Figure 4(b,e,h), our method has the obvious superiority compared with the CNN-L, DML and CNN + LSTM. It is own to the mixed 3D/2D convolution capsule net and end-to-end pattern.

Table 3 shows the \( F_1 \) score for all the methods. It should be noted that the fingerprint code length for our method and SGM is 32 bit. The length for all the other methods is much higher than 32 bit. It is 256 bit for DML method. Whether average or total, our method has obtained the highest \( F_1 \) score among all methods. Compared with DML, which is the best among compared methods, there is still 0.21 and 0.25 higher than that. However, our code length is very shorter than that of DML. This shows our method reduces the code length greatly while enhancing robustness and distinctiveness.

| Distortions        | Proposed | Loss frame | Median filtering | Insert logo | Insert subtitles | Gaussian noise | Gamma adjust | Rotation and Cropping | Average \( F_1 \) | Total \( F_1 \) |
|--------------------|----------|------------|-----------------|-------------|-----------------|---------------|--------------|-----------------------|-----------------|---------------|
| DML (Kordopatis-Zilos et al., 2017b) | 0.998    | 0.998      | 0.992           | 0.997       | 0.997           | 0.997         | 0.997        | 0.999                 | 0.997           | 0.996         |
| CGO (Lee & Yoo, 2008) | 0.974    | 0.986      | 0.981           | 0.976       | 0.976           | 0.972         | 0.972        | 0.995                 | 0.971           | 0.971         |
| SGM (Li & Monga, 2013) | 0.923    | 0.937      | 0.933           | 0.935       | 0.935           | 0.935         | 0.935        | 0.927                 | 0.927           | 0.927         |
| TIRI (Coskun et al., 2006) | 0.784    | 0.767      | 0.767           | 0.772       | 0.772           | 0.772         | 0.772        | 0.772                 | 0.772           | 0.772         |
| RASH (De Roover et al., 2005) | 0.841    | 0.681      | 0.714           | 0.952       | 0.952           | 0.952         | 0.952        | 0.952                 | 0.952           | 0.952         |
Distortion intensity also affects the performance; in order to test the stability we have performed large experiments for different distortion intensity. The results are shown in Figure 5, where there is a slight variation when varying the distortion intensity. This proves that the proposed method is stable for different distortion intensity.

Loss function is crucial to the fingerprinting performance. In order to prove the designed adaptive is effective; experiments on loss function have been test. Figure 6 shows the curves of margin affected. The margin $\beta$ in formula (2) is 0.2, 0.3, 0.4 and 0.5, respectively for comparing. When the margin is equal to 0.4 the curve displays best. The adaptive case shows the obvious advantages.
The computational cost of video fingerprints is also one of the important indicators to measure the performance of the algorithm. Based on the extraction of fingerprint features of a single video, this paper analyses the computational cost from three aspects: time complexity, space complexity and running time. Based on the same computer hardware configuration, it is compared with the three deep learning algorithms of CNN-L, DML and CNN + LSTM. It can be seen from Table 4 that the space complexity and time complexity of the 3D/2D mixed convolutional capsule network designed in this paper are 13.8M and 7.36G, respectively. Compared with the CNN-L algorithm using the VGG16 network and CNN + LSTM algorithm combining resnet50 and LSTM as backbone.

Figure 5. ROC curves for different distortion intensities.

Figure 6. The performance effects of margin.
network has obvious advantages in the amount of network parameters and the number of floating point operations, which saves a lot of network computing time. Compared with the DML algorithm with the lowest calculation cost, the running time gap is small, which shows that the algorithm has a certain high efficiency in extracting video fingerprints.

4. Conclusions
We presented a novel end-to-end fingerprinting, which utilized a novel mixed 3D/2D convolution capsule net and adaptive triplet loss function. The designed capsule net was applied to video data efficiently and obtained a compact representation. The adaptive loss was better than hard threshold. Experiments on the public dataset showed that our proposed method achieved the state-of-the-art performance. However, how to design a fast capsule net for 3D video is challenging. Moreover, how to output binary code directly is a valuable research issue.

Acknowledgements
The authors would like to thank the editor and the anonymous reviewers for their valuable comments and constructive suggestions.

Disclosure statement
No potential conflict of interest was reported by the author(s).

Funding
This work was supported by the National Natural Science Foundation of China [grant number 61402152], Programs for Science and Technology of Henan Province [grant number 192102210099, 192102210100], National Key Research and Development Project [grant number 2018YFC0604502]; Key scientific and technological projects in Henan Province.

Notes on contributors
Lianghao Xu received the B.Sc. degree from Luoyang Institute of Science and Technology, China, in 2018. He is currently pursuing the M.Sc. degree in the School of Henan Polytechnic University, China. His research interests include video fingerprinting and deep learning.

Xinwei Li received the B.S. degree from Henan University, China, in 2006. He received the Ph.D. degree from Xidian University, China, in 2011. He is currently a lecturer in the School of Electrical Engineering and Automation, Henan Polytechnic University, China. His research interests include video fingerprinting, computer vision and machine learning.

Yi Yang received the B.S. and M.S. degrees from Henan Polytechnic University, China, in 2004 and 2007, respectively. He received the Ph.D. degree from Beihang University, China, in 2017. He is currently a lecturer at the School of Electrical Engineering and Automation, Henan Polytechnic University, China. His research interests include Artificial intelligence, Reinforcement learning and Adaptive Dynamic Programming and their application in coal mining.

References
Bellet, A., Habrard, A., & Sebban, M. (2014). A survey on metric learning for feature vectors and structured data. arXiv:1306.6709v4 Technical report.

Coskun, B., Sankur, B., Memon, N., & Macq, B. (2006). Spatio-temporal transform based video hashing. IEEE Transactions on Multimedia, 8(6), 1190–1208. http://doi.org/10.1109/TMM.2006.884614

De Roover, C., De Vleeschouwer, C., & Lefebvre, F. (2005). Robust video hashing based on radial projections of key frames. IEEE Transactions on Signal Processing, 53(10), 4020–4037. http://doi.org/10.1109/TSP.2005.855414

Esmaeili, M. M., Fatourechi, M., & Ward, R. K. (2011). A robust and fast video copy detection system using content-based fingerprinting. IEEE Transactions on Information Forensics and Security, 6(1), 213–226. http://doi.org/10.1109/TIFS.2010.2097593

Guzmanzavaleta, Z. J., Feregrinouribe, C., Moralessandoval, M., & Menendez-Ortiz, A. (2017). A robust and low-cost video fingerprint extraction method for copy detection. Multimedia Tools and Applications, 76(22), 24143–24163. http://doi.org/10.1007/s11042-016-4168-6

Hu, Y., & Lu, X. (2018). Learning spatial-temporal features for video copy detection by the combination of CNN and RNN. Journal of Visual Communication and Image Representation, 55, 21–29. http://doi.org/10.1016/j.jvcir.2018.05.013

Jiang, Y. G., & Wang, J. (2016). Partial copy detection in videos: A Benchmark and an evaluation of popular methods. IEEE Transactions on Big Data, 2(1), 32–42. http://doi.org/10.1109/TBDA.2016.2530714

Jiang, Y. G., Wu, Z., Wang, J., Xue, X.-y., & Chang, S.-F. (2018). Exploiting feature and class relationships in video categorization with regularized deep neural networks. IEEE Transactions on Pattern Analysis & Machine Intelligence, 40(2), 352–364. http://doi.org/10.1109/TPAMI.2017.2670560

Kordopatis-Zilos, G., Papadopoulos, S., Patras, I., & Yiannis, K. (2017a). Near-duplicate video retrieval by aggregating intermediate CNN layers. Proceedings of International Conference on multimedia Modeling, Heidelberg: Springer, Jan.4. 251–263.

Kordopatis-Zilos, G., Papadopoulos, S., Patras, I., & Yiannis, K. (2017b). Near-duplicate video retrieval with deep metric learning. Web-Scale Vision and Social Media (VSM), ICCV 2017. IEEE computer Society.Oct.23.
Lee, S., & Yoo, C. D. (2008). Robust video fingerprinting for content-based video identification. *IEEE Transactions on Circuits and Systems for Video Technology, 18*(7), 983–988. [http://doi.org/10.1109/TCSVT.2008.920739](http://doi.org/10.1109/TCSVT.2008.920739)

Li, M., & Monga, V. (2013). Compact video fingerprinting via structural graphical models. *IEEE Transactions on Information Forensics and Security, 8*(11), 1709–1721. [http://doi.org/10.1109/TIFS.2013.2278100](http://doi.org/10.1109/TIFS.2013.2278100)

Sabour, S., Frosst, N., & Hinton, G. E. (2017). *Dynamic routing between capsules.* Advances in neural information Processing Systems 30, Long Beach: NIPS, Dec.4-9. 3856.

Tran, D., Bourdev, L., Fergus, R., Lorenzo, T., & Manohar, P. (2015). *Learning spatiotemporal features with 3D convolutional networks.* Proceedings of IEEE International Conference on Computer Vision, Los Alamitos: IEEE Computer Society Press, Dec.7-13. 4489–4497.

Zhou, Y., Sun, X., Zha, Z. J., & Zeng, W. (2018). *MiCT: Mixed 3D/2D convolutional tube for human action recognition.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Los Alamitos: IEEE Computer Society Press, 449–458.