4D EFFECT CLASSIFICATION BY ENCODING CNN FEATURES

Thomhert S. Siadari[1], *Mikyong Han*[2], *and Hyunjin Yoon*[1,2]

[1]Korea University of Science and Technology, South Korea
[2]Electronics and Telecommunications Research Institute, South Korea

ABSTRACT

4D effects are physical effects simulated in sync with videos, movies, and games to augment the events occurring in a story or a virtual world. Types of 4D effects commonly used for the immersive media may include seat motion, vibration, flash, wind, water, scent, thunderstorm, snow, and fog. Currently, the recognition of physical effects from a video is mainly conducted by human experts. Although 4D effects are promising in giving immersive experience and entertainment, this manual production has been the main obstacle to faster and wider application of 4D effects. In this paper, we utilize pretrained models of Convolutional Neural Networks (CNNs) to extract local visual features and propose a new representation method that combines extracted features into video level features. Classification tasks are conducted by employing Support Vector Machine (SVM). Comprehensive experiments are performed to investigate different architecture of CNNs and different type of features for 4D effect classification task and compare baseline average pooling method with our proposed video level representation. Our framework outperforms the baseline up to 2-3% in terms of mean average precision (mAP).

Index Terms— Convolutional Neural Networks (CNNs), 4D Effect, Classification, Video Representation

1. INTRODUCTION

4D movies brings a new dimension to the visitor experience by providing a various physical effects synchronized along with the movie played in the theater. Effects simulated in 4D theater may include motion, vibration, flash, wind, water, scent, thunderstorm, snow, and fog, where chairs may move, vibrate, and shake to certain directions as well as water and air may be sprayed. These physical effects are presented according to the events occurring in the movie. Even though 4D movies are promising in giving such immersive experience and entertainment, the physical effects are edited mainly by human experts. This manual production has been the main obstacle to faster and wider application of 4D effects.

Deep learning and neural networks have become the state-of-the-art methods in a wide range of video analysis tasks [1]. The fact that images and videos have become pervasive on the Internet encourages the development of algorithms and techniques that can be used for various applications [2]. Recently, Convolutional Neural Networks (CNNs) have been developed as an effective method for understanding image content, bringing breakthrough on various tasks e.g. image recognition, classification, segmentation, detection, and retrieval [3, 4, 5, 6, 7]. The primary keys of these achievements were methods for scaling up the parameters of networks and tremendous annotated image datasets that support advanced learning process. It has been showed that CNNs can learn understandable and interpretable image features and have retrieved the appearance information over not only single and static images but also several video frames [8]. Since CNN architecture consists of a stacked structures, it provides us a feature representation from each layer that has different characteristic. The first layer is known to learn the features that are similar to Gabor filters and color blobs. Such features are agnostic to the task. On the other hands, the higher-level layers are usually well trained for specific tasks and used for image analysis [9, 10, 11, 12].

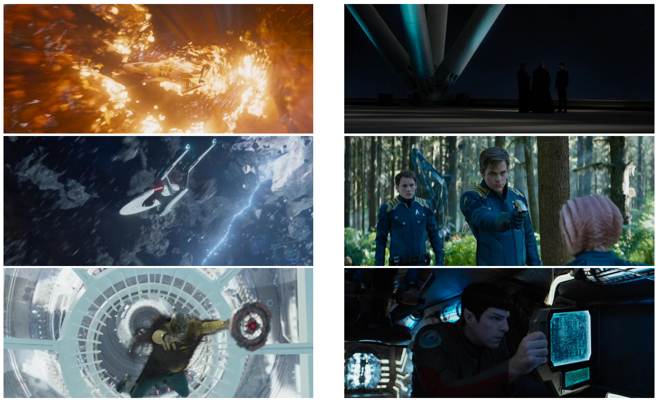

Fig. 1: Examples of video frames on each class. Left: Effect class. Right: Non-effect class.

Motivated by constructive results in the image domain and the opportunity to develop an automatic method to recognize effect types in videos, we propose a framework to classify a video whether it contains 4D effect or not. The framework basically consists of three parts: feature extraction, video representation, and classification. However, we mainly focus

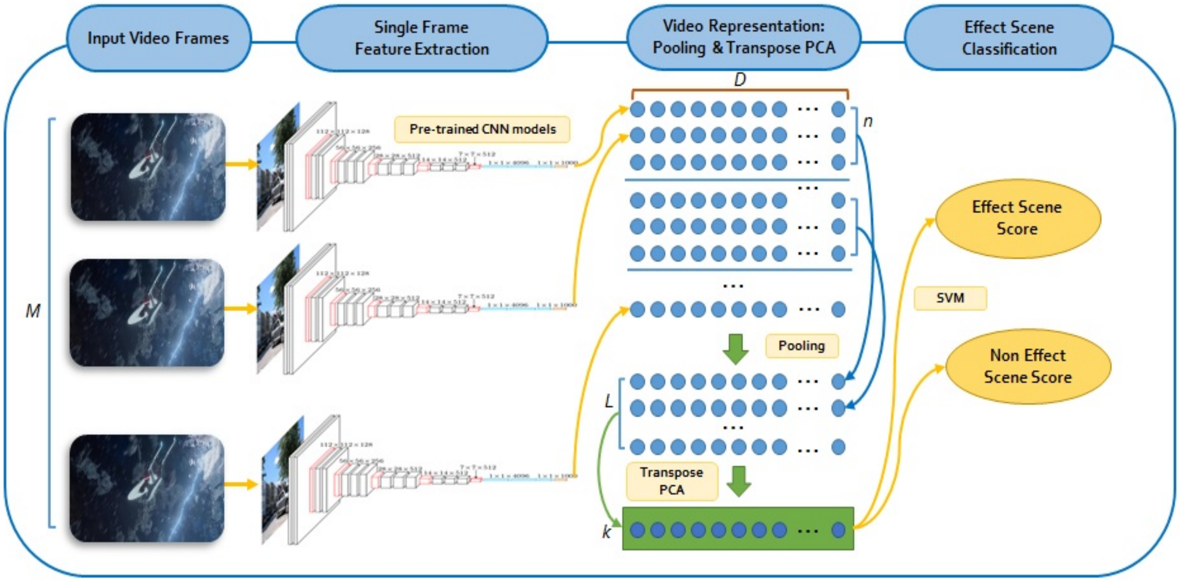

Fig. 2: An overview of our framework. Video frames feed into CNN feature extraction. Then, extracted features are represented by video representation. Finally, SVM is used to classify video based on its class.

on how well CNN architectures extract the features, which features are more favorable to be used in video level representation, and how to represent frame-based features into video features before conducting classification using SVMs. In summary, our contributions are three-folds: (1) a framework to classify 4D effect videos (2) comprehensive experiments on different CNN architectures and layer-wise features (3) a new video representation pooling to increase classification performance. To our best knowledge, this work is the first attempt to apply deep learning method for 4D effect classification on videos.

2. PROPOSED METHOD

This section describes the details of our approach for 4D effect classification. Two important processes of our system are image-based feature extraction and video feature representation. First, we extract deep features from each frame of video using CNN, and then represent those features into video-level features to be used as an input to SVMs as illustrated in Fig. 2.

2.1. Feature Extraction

Instead of designing new CNN with random initialization, we adopt the pre-trained CNN models and their parameters towards our classification target. Particularly, we employ the CNN configuration of CaffeNet [13] and VGG-16-layer [14] (for the rest we use the term ”VGGNet” to describe VGG-16-layer). CaffeNet is a replication model of AlexNet [15] with some modifications with the same performance results. The differences are that CaffeNet was not trained with relighting data-augmentation and the order of pooling and normalization layers is switched. Like AlexNet, CaffeNet has 5 convolutional layer and 3 fully-connected layer (fc6, fc7, fc8) and trained over the ImageNet dataset for the image classification problem in the ILSVRC-2012 competition [16]. While VGGNet extends the AlexNet by expanding convolutional layers and has 16 layers. VGGNet was proposed by Simoyan et al. [14], which is the improvement of their winning model on ILSVRC 2014 competition. Nonetheless, these networks are still designed for image classification but not necessarily good for video classification tasks. In the performance evaluation section, we study the effect of using different deep convolutional architectures for 4D effect classification. The output of each layer is considered as visual representation of each frame.

2.2. Video Representation

The purpose of video representation is to combine the whole CNN extracted frames by pooling all the features in a video. Based on [17], average pooling (AvgP) method is relatively the common standard to be used in video classification. AvgP is defined as $x_{video} = \frac{1}{S}\sum_{i=1}^{S} x_i$, where S is the total number of frames from a video and x_i represents the features extracted from the CNN models for the $i-$th frame. Therefore, after doing average pooling representation, we obtain $D-$dimensional vector, where D is size of output dimension from corresponding layer. In the case of fc6 and fc7 of CaffeNet and VGGNet, one video is represented as 1×4096 vec-

Table 1: Real-world 4D effect video dataset

	Effect				Non-effect
	Motion	Vibration	Wind	Flash	
Number of video clips	137	75	93	42	238
Avg. frame number	611				329
Avg. length (s)	21				11

tor.

Even though average pooling is commonly to be used, we find that it could not represent a video properly because averaging all frames could lose temporal information especially in the case of long video. Therefore, we propose a new video level representation for our classification task that takes an advantage of average pooling and Principal Component Analysis (PCA). In Fig 2, it takes two steps to create a video level representation from image-based features. The first step is taking an average of several frames. Assume that a video has M number of frames. After extracting all frames using CNNs, we have a $M \times D$ feature matrix. Then, we take an average of every n-row of $M \times D$ to get a smaller matrix $L \times D$, where L is $\lceil M/n \rceil$. The second step is applying transpose PCA. We apply PCA to the transpose matrix of $L \times D$ and maintain k number of principal components in order to have a $D \times k$ matrix. We again transpose $D \times k$ matrix into $k \times D$ matrix and consider this matrix to represent a single video.

3. PERFORMANCE EVALUATION

We first introduce our collected dataset and explain experimental settings for 4D effect classification task. Then, we show the performance evaluation results of our simulation. Furthermore, we discuss the results in terms of different CNN architectures and features, performance of our proposed video pooling representation, and our effort on fine-tuning. Generally, the purpose of our framework is for binary classification. However, we also provide a table of performance result on multi-class classification. Mean average precision (mAP) and F1-score are adopted as performance evaluation metric. Simulations were conducted on computer with NVIDIA GPU GeForce TITAN X 12GB memory.

3.1. Dataset

The collected dataset is based on manual annotation of real physical effects simulated in 4D theaters. We trimmed a movie into several clips according to its annotation so that each clip was labelled with a single 4D effect type. All clips labelled with any of 4D effect are grouped into a single 4D effect class. Subsequently, we trimmed the part of movie that has no effect annotation, then we put these all clips into non-effect class. The effect class can be further classified into 4 different effects: motion, vibration, wind, and flash. Finally, we collected 570 video clips in our dataset. The trait of our data is that even when a video is properly annotated, it may consist various different content on the frame level. Different effect type may consist of similar objects and events on the frame level. The effect may be occurred in any event, any place, any object, and any activity. For example in effect class, it probably occurs on fighting, cooking, walking, flying activity, indoor and outdoor with multiple concpets form object like in beach while people are sailing. Using this dataset we can conduct two tasks: binary classification using effect and non-effect classes and multi-class classification using 4 different effect types. We provide an information of our collected dataset in Table 1.

3.2. Experimental Setting

Feature extraction is conducted using pre-trained model of CaffeNet and VGGNet on Caffe deep learning framework [13]. During feature extraction on frame level, we resized the data into 320×240 and oversampled the data. After obtaining video representations, we employed SVM with a linear kernel to do classification using scikit library[18]. We trained one-vs-the-rest classifiers for all the input features. Training data is 80% of dataset while testing data is 20% of dataset. At first we used all clips from dataset. However, we decided to exclude too short and too long video clips for better performance.

3.3. Results and Discussion

Architectures and features. We utilized two popular state-of-the-art deep networks to extract image features for 4D effect classification: CaffeNet and VGGNet. As for layer-wise features, we focus on fc6 and fc7 feature maps since these two features give better result on classification task [19, 20]. Table 2 shows experimental result on different CNN architectures using different features. For this experiment, we use average pooling method in video level representation. As shown in Table 2, features from fc6 are more favorable than features from fc7. Surprisingly using CaffeNet for feature extraction outperforms feature extraction using VGGNet because CaffeNet has shallower layer than VGGNet indicating that too deep networks do not guarantee to work better on all tasks.

Table 2: Performance result of framework using different architectures and features. AvgP is used for video pooling representation

Model-Feature	mAP (%)	F1-score
VGGNet-fc6	63.87	0.64
VGGNet-fc7	61.53	0.63
CaffeNet-fc6	66.67	0.68
CaffeNet-fc7	63.63	0.69

Video pooling representation. We further report the results of using our proposed video pooling on fc6 features. As shown in Table 3 our proposed method outperforms the baseline AvgP method. The improvement from proposed method shows that by averaging several numbers of frames from a video and selecting principal components of group of average frames can achieve better video-level representations. Having better video representation can boost the the classification process. From our simulations, our approach increased the performance up to 2-3% in terms of mAP regardless CNN architecture. This result validates the effectiveness of our proposed idea on video representation pooling.

Table 3: Comparison between average pooling method and proposed pooling method

Model-Feature	mAP (%)		F1-score	
	AvgP	Proposed	AvgP	Proposed
VGGNet-fc6	63.87	66.76	0.64	0.67
CaffeNet-fc6	66.67	68.42	0.68	0.69

Binary vs multi-class classification. Even though our main purpose is to classify video clips that contain effect and non-effect, we also tried multi-class classification tasks to better understand our framework. Since we have 4 different effect types on our effect class, this purpose can be easily accomplished. While the best result is produced when CaffeNet is used for binary classification, it did not work well for multi-class classification. The highest mAP is obtained by using VGGNet as shown in Table 4.

Table 4: Performance comparison between binary classification and multi-class classification

Model-Class	mAP (%)		F1-score	
	AvgP	Proposed	AvgP	Proposed
VGGNet-binary	63.87	66.76	0.64	0.67
VGGNet-multi	37.83	41.1	0.35	0.41
CaffeNet-binary	66.67	68.42	0.68	0.69
CaffeNet-multi	31.29	33.72	0.29	0.34

Fine-tuning. Fine-tuning is a common method to further increase the performance of pre-trained network using provided dataset. However, our experimental results demonstrated that fine-tuning did not work well for 4D effect classification tasks. We fine-tuned the pre-trained network using our training dataset collected for binary classification. The loss function still did not significantly decrease. Using fine-tuned model barely demonstrates better performance, even using fine-tuned CaffeNet model decreases the performance results as shown in Table 5. We assumed that fine-tuning does not turn out well due to two reasons; First, the images of the same category may be in different effect class which will confuse deep network which is trained from ImageNet classification data. Second, the noisy images further confuse the deep network.

Table 5: Performance comparison between framework using pre-trained model and fine-tuned (FT) model

Model	mAP (%)		F1-score	
	AvgP	Proposed	AvgP	Proposed
VGGNet	63.87	66.76	0.64	0.67
VGGNet-FT	68.44	69.46	0.66	0.67
CaffeNet	66.67	68.42	0.68	0.69
CaffeNet-FT	55.55	58.33	0.61	0.65

4. CONCLUSIONS

In this paper we propose a new framework for 4D effect classification on videos. First, we collected a new video dataset which consists of two classes: effect and non-effect classes. In effect class dataset, there are 4 type of effects: motion, vibration, wind, and flash. We then applied CNNs to extract visual features from the collected video frames and combined the extracted visual features using a new video representation method. Finally, an SVM classifier was trained to classify whether each clip contains a 4D effect or not. Our experimental results showed that feature extraction using CaffeNet achieved better results that VGGNet. Fc6 feature is also more favorable than fc7 features. Our proposed video representation constantly outperforms the baseline average pooling method up to 2-3% in terms of mean average precision (mAP). However, our attempts to fine-tune pre-trained model brings no impact to classification performance. For future direction, we plan to collect more dataset and improve the 4D effect recognition performance by exploiting temporal features in the underlying videos.

5. ACKNOWLEDGEMENT

This work was supported by the ICT R&D program of MSIP/IITP, Rep. of Korea (17ZF1200, Development of XD-Media Solution for Invigoration of Realistic Media Industry).

6. REFERENCES

[1] Yichuan Tang, "Deep learning using linear support vector machines," *arXiv preprint arXiv:1306.0239*, 2013.

[2] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[4] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.

[5] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[6] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun, "Learning hierarchical features for scene labeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[7] Dan Ciresan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in neural information processing systems*, 2012, pp. 2843–2851.

[8] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*. Springer, 2014, pp. 818–833.

[9] Tao Chen, Damian Borth, Trevor Darrell, and Shih-Fu Chang, "Deepsentibank: Visual sentiment concept classification with deep convolutional neural networks," *arXiv preprint arXiv:1410.8586*, 2014.

[10] Can Xu, Suleyman Cetintas, Kuang-Chih Lee, and Li-Jia Li, "Visual sentiment prediction with deep convolutional neural networks," *arXiv preprint arXiv:1411.5731*, 2014.

[11] Quanzeng You, Jiebo Luo, Hailin Jin, and Jianchao Yang, "Robust image sentiment analysis using progressively trained and domain transferred deep networks," 2015.

[12] Victor Campos, Amaia Salvador, Xavier Giro-i Nieto, and Brendan Jou, "Diving deep into sentiment: Understanding fine-tuned cnns for visual sentiment prediction," in *Proceedings of the 1st International Workshop on Affect & Sentiment in Multimedia*. ACM, 2015, pp. 57–62.

[13] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[14] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[17] Robin Aly, Relja Arandjelovic, Ken Chatfield, Matthijs Douze, Basura Fernando, Zaid Harchaoui, Kevin Mcguiness, Noël O'Connor, Dan Oneata, Omkar Parkhi, et al., "The axes submissions at trecvid 2013," 2013.

[18] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al., "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[19] Zhaofan Qiu, Qing Li, Ting Yao, Tao Mei, and Yong Rui, "Msr asia msm at thumos challenge 2015," .

[20] Zhongwen Xu, Yi Yang, and Alex G Hauptmann, "A discriminative cnn video representation for event detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1798–1807.