4D Effect Video Classification with Shot-aware Frame Selection and Deep Neural Networks

Thomhert S. Siadari[1], Mikyong Han[2], and Hyunjin Yoon[1,2]
Korea University of Science and Technology, South Korea[1]
Electronics and Telecommunications Research Institute, South Korea[2]
{thomhert,mkhan,hjyoon73}@etri.re.kr

Abstract

A 4D effect video played at cinema or other designated places is a video annotated with physical effects such as motion, vibration, wind, flashlight, water spray, and scent. In order to automate the time-consuming and labor-intensive process of creating such videos, we propose a new method to classify videos into 4D effect types with shot-aware frame selection and deep neural networks (DNNs). Shot-aware frame selection is a process of selecting video frames across multiple shots based on the shot length ratios to subsample every video down to a fixed number of frames for classification. For empirical evaluation, we collect a new dataset of 4D effect videos where most of the videos consist of multiple shots. Our extensive experiments show that the proposed method consistently outperforms DNNs without considering multi-shot aspect by up to 8.8% in terms of mean average precision.

1. Introduction

4D technology has been widely adopted in the entertainment industry to provide more immersive experience. A 4D movie is a term when a movie synced along with physical effects during its show time at the theater. Physical effects may include motion and vibration on chairs, as well as wind, water spray, scent, and flashlight. Although the 4D movie is extensively played at the cinema, the annotating process of movies is very time-consuming and labor-dependent work. Encouraged to automate and fasten annotating process, we propose a new method to classify videos into 4D effect types. Two major steps required to classify 4D effect videos are providing the right dataset and developing proper models. In this paper we present a solution for both dimensions.

From a dataset perspective, the available datasets commonly used on video classification are KTH [17], Weizzman [2], HMDB [11], UCF Sports [20], Hollywood-2 [12], UCF-50 [14], and UCF-101 [21]. These datasets mainly focus on action recognition and human motion recognition. There are also datasets that focus on specific activities such as MPII Cooking [15], Breakfast [10], and Sports-1M [7] datasets. To the best of our knowledge, dataset that consists of 4D effect videos is still unavailable. No previous work addresses such classification task. Therefore, we collect a new dataset of 4D effect videos with at least one annotation for each video. The videos in the dataset are trimmed videos from several movies. The dataset has several effect types such motion, vibration, wind, scent, spraying, fog, smoke, and flashlight. Figure 1 depicts examples of frames consisting of motion effect in our dataset. The shortest video has only two frames and the longest one has 672 frames. The average number of video frames is 125 frames. An effect type has different types of object and places across videos. One notable aspect of our dataset is that most of the videos consist of multiple shots. Therefore, it makes our dataset even more challenging and richer than other datasets as summarized in Table 1.

Figure 1. Visual examples of frames consisting of motion effect in our dataset.

From a modeling standpoint, we are interested in building appropriate models to conduct classification task on such challenging dataset. A 4D effect classification is more complicated than image scene recognition because movie consists of many related sequences and several hundred

Table 1. Comparison of several datasets to 4D effect dataset [25].

Dataset	Detection	Untrimmed	Open world	Multi-label	Multi-shot
KTH-action	-	-	-	-	-
UCF101	-	-	yes	-	-
HMDB51	-	-	yes	-	-
Sports1M	-	yes	yes	-	-
Cooking	yes	yes	-	-	-
Breakfast	yes	yes	-	-	-
THUMOS	yes	yes	yes	-	-
MultiTHUMOS	yes	yes	yes	yes	-
Our Dataset	yes	yes	yes	yes	yes

shots. Since our dataset consists of trimmed videos from movies, it is unavoidable that most of the videos in our dataset have multiple shots, which makes every single video appear visually distinctive across frames. With this observation, we hypothetically consider that we need to carefully select frames across multiple shots in a video because using simple step-size sampling method may lose critical frames from the video for effect classification. Therefore, we tackle the problems of multi-shot 4D effect video classification by using shot-aware frame selection and different DNN models including 3D Convolutional Neural Networks (3D CNN), Convolutional Recurrent Neural Networks (CRNN), and Long Short-Term Memory (LSTM). Shot-aware frame selection is a process of selecting video frames across multiple shots based on the shot length ratios to subsample every video down to a fixed number of frames for classification. To provide fair comparison how the shot-aware frames contribute to our model, we conduct simulations using the same network parameters for shot-aware frame selection and step-size sampling method.

Our main contributions can be summarized as follows:

1. We introduce a 4D effect dataset based on real-world application.
2. We propose shot-aware frame selection and deep neural networks to classify videos into 4D effect types.
3. The proposed method outperforms the classification task using deep neural networks without considering multi-shot existence in terms of mean average precision (mAP).

2. Related Work

We review recent works on existing dataset and well-known methods on video classification task.

Dataset. There are various public datasets commonly used for video classification. KTH [17] was used to recognize human action and contains six types of human actions, walking, jogging, running, boxing, hand waving and hand clapping, performed several times in four different scenarios, outdoors, outdoors with scale variation, outdoors with different clothes, and indoors. All videos were taken over similar backgrounds with a static camera and only containend single-shot videos. The HMDB [11] was collected from various sources, mostly from movies, YouTube and Google videos. The dataset contains 6849 videos divided into 51 action categories and each class includes at least 101 clips. Hollywood-2 [12] is a dataset with 12 classes of human actions and 10 classes of scenes collected from 69 movies. The dataset is distributed over 3669 video clips and approximately 20.1 hours of video in total. The UCF-101 [21] consists of 13,320 videos with 101 classes and targets an action recognition data set of realistic action videos, collected from YouTube. The videos in 101 action categories are grouped into 25 groups, where each group can consist of 4-7 videos of an action. However, the videos on UCF 101 mostly consist of single action class. MultiTHUMOS dataset [25] contains dense, multilabel, frame-level annotations for 40 hours across 400 videos. MPII Cooking focuses on cooking activity, Breakfast dataset comprises of 10 actions related to breakfast preparation, and Sports-1M is 1 million YouTube videos consisting 487 classes of sport activities. Different from the previous dataset, our new collected dataset has untrimmed videos, trimmed videos, multi-label videos, and multi-shot videos which make it more challenging.

Deep learning models. Recently, deep learning methods show major breakthrough on video classification. It was started from AlexNet [9] on image classification task. Since then, researchers have extended the work into video classification. [19] utilized not only spatial information but also optical flow for CNNs named two-stream CNNs. [7] investigated 1M-Sports dataset to learn spatiotemporal information on video classification and tried to speed up simulations. While [24] provided thorough investigation of 3D CNNs that learn spatiotemporal information on action recognition. Long-term Recurrent Convolutional Networks (LRCN) [6] were developed to investigate a recurrent convolutional architecture which is suitable for large-scale visual learning and demonstrating good results on benchmark video recognition tasks, image description and retrieval problems, and video narration. Taking advantages of how previous research solve video classification problems, we build own model based on 3D CNNs, convolutional recurrent neural networks (CRNN) which are networks with end-to-end learning manner, and feature extraction using CNNs combined with Long Term-Short Memory (LSTM).

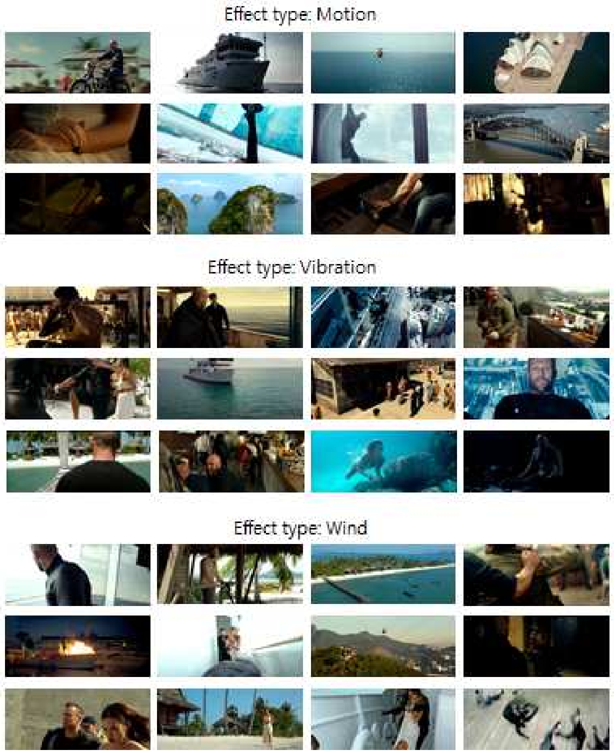

Figure 2. An overview of video clips contained 4D effects in our dataset. There are three types of effect: motion, vibration, and wind. Our dataset is very challenging due to high intra-class variation but low inter-class visual variation. Different effect type may consist of similar objects and events on the frame level. The effect may occur in any event, any place, any object, and any activity. For example, vibration effect probably occurs on fighting, cooking, walking, flying activity, indoor and outdoor with multiple concepts from objects like in beach while people are sailing.

3. Dataset

In this section, we describe our collected dataset including the collecting process and detail information regarding the dataset. Until now, no particular dataset can be used for research on 4D effect video classification based on real-world application. Our dataset comprises of 500 video clips labeled at least with an effect type. We hired human experts to annotate movies according to real effects occurred when watching movies in the cinema. Workers were provided with the possible type of effects and asked to annotate start and end time of effects happened in the movies. A physical effect may be visible in anytime during the show-time. After having the recorded time, we annotated start and end frames of these effects. Once we have annotation list, we trimmed the movie into video clips according to their annotation. Then, we divide the group of videos into training, validation, and testing dataset by 70%, 10%, and 20%. Our dataset has high inter-class visual variation but low intra-class visual variation. Also, most of the videos have multiple shots. Different effect type may consist of similar objects and events on the frame level. The effect may occur in any event, place, object, and activity. For example, vibration effect probably occurs on fighting, cooking, walking, flying activity, indoor and outdoor with multiple concepts form object such as taken at the beach while people are sailing. Figure 2 depicts visual appearance of each class in our dataset and Figure 3 provides an example of all physical effects occurred in a movie consisting of 180,000 frames.

4. Proposed Method

Unlike a video with a single shot, a video with multiple shots contains richer information and visually distinct across frames making it harder to classify. In this section, we elaborate how we develop our proposed method to classify videos into 4D effect types. First, we explain the shot-aware frame selection. Second, we give details of 4 different models that we build: 2D CNN as a baseline, 3D CNN, convolutional recurrent neural networks (CRNN), and CNN+LSTM. Detail architecture of these models can be found in Figure 5.

4.1. Shot-aware frame selection

A single shot is a sequence of frames running for an uninterrupted period and recorded from a single camera. Also, a single shot in a movie is the continuous footage or sequence between two edits or cuts. Usually, a two-hour movie has several hundred thousand shots in it because shots are used to demonstrate different aspects of a film's setting, scenes, characters, stories, ideas, and themes. There are several types of shots such as extreme long shot, long shot, full shot, mid shot, close up, and extreme close up. Since our dataset consists of trimmed videos from movies, it is unavoidable that most videos in our dataset have multiple shots, which makes every single video appear visually distinctive across frames. With this observation, we hypothetically consider that we need to carefully select frames across multiple shots in a video because using a simple step-size sampling method may lose critical frames from the video for effect classification.

Our shot-aware frame selection method consists of two steps: detecting shots and selecting frames from these shots. We name the selected frames as shot-aware frames. We de-

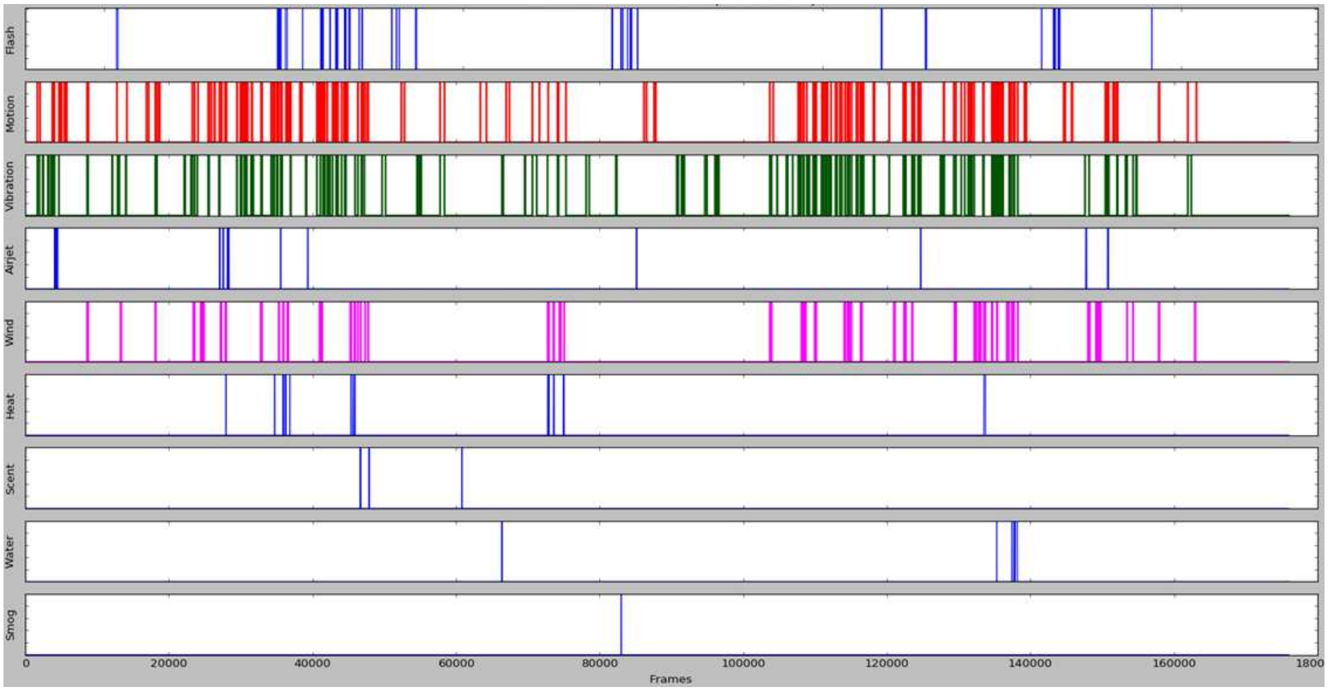

Figure 3. An example of effects occurred in a movie.

tect shots between scenes in a movie by finding the boundaries where difference between two consecutive frames exceeds certain threshold. Given a single video with N number of frames and M number of shots, s_m for $1 \leq m \leq M$, let the length of shot s_m be denoted as $|s_m|$. That is, a shot s_m consists of $|s_m|$ number of frames. Assume that we require K number of frames to represent the given video, we define that a single shot s_m contributes $K \times \, |s_m| \, /n$ frames out of K frames. Then, we select $K \times \, |s_m| \, /n$ number of frames from the shot s_m by using step-size sampling starting from the first frame within the shot. Therefore, the number of contributed frames from a shot depends on its length. Figure 4 shows the comparison between evenly subsampled frames and shot-aware frames. The odd number rows are the shot-aware frames selected from a video and the even number rows are the frames selected using simple step-size sampling. Even though some frames are visually similar, in this example there are consecutive frames that are not selected by step-size sampling method which may essential to classification task.

4.2. 2D CNN

2D CNN is a single-frame baseline architecture to understand the contribution of a static image to classification task. In this paper, we use transfer learning to retrain certain layers of InceptionV3 model on our collected dataset. InceptionV3 is 2015 of Google's Inception architecture for image recognition pre-trained on ImageNet dataset [16]. The model uses a network-in-network approach, stacking Inception modules to build a network layer. InceptionV3 takes a single image to be passed through multiple Inception modules, each of which applies in parallels. Details of InceptionV3 can be found in [23].

4.3. 3D CNN

3D CNN is one of the networks that learns spatiotemporal features by modeling temporal information using 3D convolutional and 3D pooling operations [24]. The main difference from 2D CNN is that convolution and pooling operations are performed spatiotemporally while in 2D CNN both are done only spatially. Hence, temporal information of the input is ignored right after the convolutional operation. In this paper, we design a 3D CNN that has three convolutional layers and three pooling layers (a pooling layer immediately follows a convolutional layer), two fully connected layers and a softmax loss layer [1] as the last layer. The input of the network is $80 \times 80 \times 3$ video frame. Each convolutional layer yields 32, 64, and 128 feature maps, respectively. The number of neurons in each fully connected layer is set to 256. Additionally, the output of the last layer is set according to the number of effect types in the classification task. Every convolution layer has different filter size followed by a max pooling layer. Filter size of $7 \times 7 \times 7$, $3 \times 3 \times 3$, and $2 \times 2 \times 2$ with stride 1 are used to result in conv1, conv2, and conv3, respectively. All pooling layers are max pooling with kernel size $1 \times 2 \times 2$ and

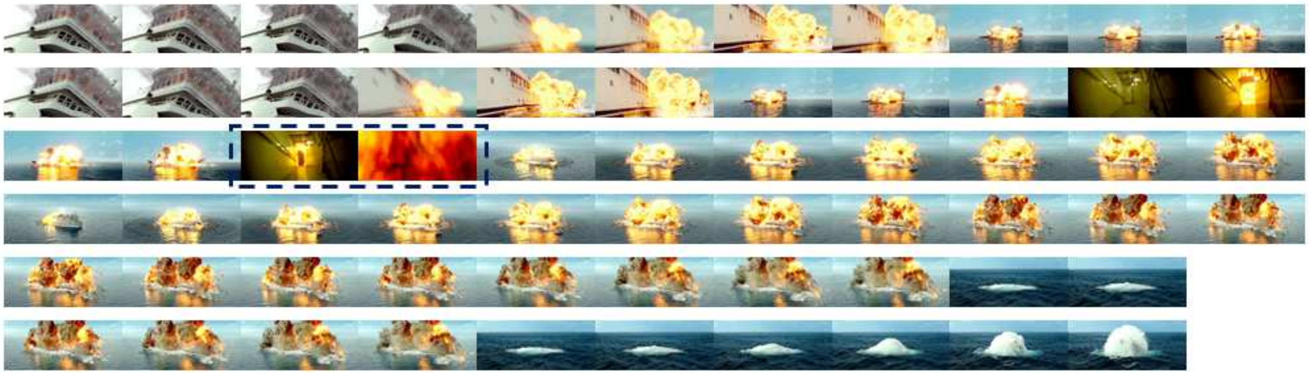

Figure 4. The odd number rows are selected frames from shot-aware frame selection while the even number rows are results from step-size sampling method. There are consecutive frames that are not sampled by step-size sampling method which may essential to classification task.

stride 1 × 2 × 2. Dropout [22] is implemented after each fully connected layer for regularization. Rectified Linear Unit (ReLU) [13] is used as the activation function for all convolutional layers. However, the last layer uses softmax regression and acts as multi-classifier to predict 4D effect classification.

4.4. CRNN

The idea of CRNN is proposed by [18] for scene text recognition which integrates feature extraction, sequence modeling, and transcription into a unified framework in end-to-end learning. One of the advantages of CRNN is that it naturally handles sequences in arbitrary lengths. Therefore, CRNN is suitable for a video classification task. In this paper, we re-design the CRNN to satisfy our purpose by keeping the end-to-end learning fashion. And instead of using a general RNN, we implement LSTM to be integrated with CNNs. The overall structure of CRNN has eight convolutional layers, four max pooling layers (one pooling layer after two convolution layers), an LSTM layer, a fully connected layer, and a softmax output layer. The frame size of input is 80 × 80. The number of filters for eight convolutional layers from 1 to 8 are 32, 32, 48, 48, 64, 64, 128, and 128, respectively. Rectified Linear Unit (ReLU) is used as the activation function for all convolutional layers. Furthermore, all pooling layers are max pooling. The LSTM layer has 1024 hidden units that return a sequence for each 32-frames clip. We add a fully connected layer that yields 2048 outputs followed by a dropout as regularization.

4.5. CNN+LSTM

Different from CRNN, CNN+LSTM is integrating two networks together by separately training each network. The CNNs are employed to extract feature maps to be used as input of LSTM. In this paper, we extract feature maps from final pooling layer of CNN using InceptionV3 pre-trained on ImageNet without fine-tuning process. The LSTM has two layers with 1024 hidden units in the first layer and 512 hidden units in the second layer followed by dropout in each layer. After two LSTM layers, we add a fully connected layer and a softmax loss layer to predict the 4D effect classification.

5. Performance Evaluation

We explain our experimental setting and evaluate our model using own dataset. The comparison between step-size sampling method and shot-aware frames is measured using mean average precision (mAP).

5.1. Dataset

We evaluate our proposed method, shot-aware frame selection and deep neural networks, using the new collected dataset as discussed in Section 3. We divide the dataset into 3 groups: 70% training, 10% validation, and 20% testing data. Furthermore, we only consider effect types that have more than 100 videos and exclude videos that consists of less than 32 frames. By doing so, we yield 3 effect types: motion, vibration, and wind.

5.2. Experimental Setting

For shot-aware frame selection, we use PySceneDetect [4] to detect shots in videos from our dataset. During training, 2D CNN is trained in the similar method to image recognition task. We do transfer learning of the pre-trained InceptionV3 model on ImageNet without its top layers with our dataset and add a new fully connected layer with neurons of 2048. The model is trained using the Adam optimizer [8] with a learning rate of 0.001 for ten epochs. Then, we retrain the network by fine-tuning the top 2 inception blocks using stochastic gradient descent (SGD) [3] with a batch size of 128, a learning rate of 0.0001, and momentum

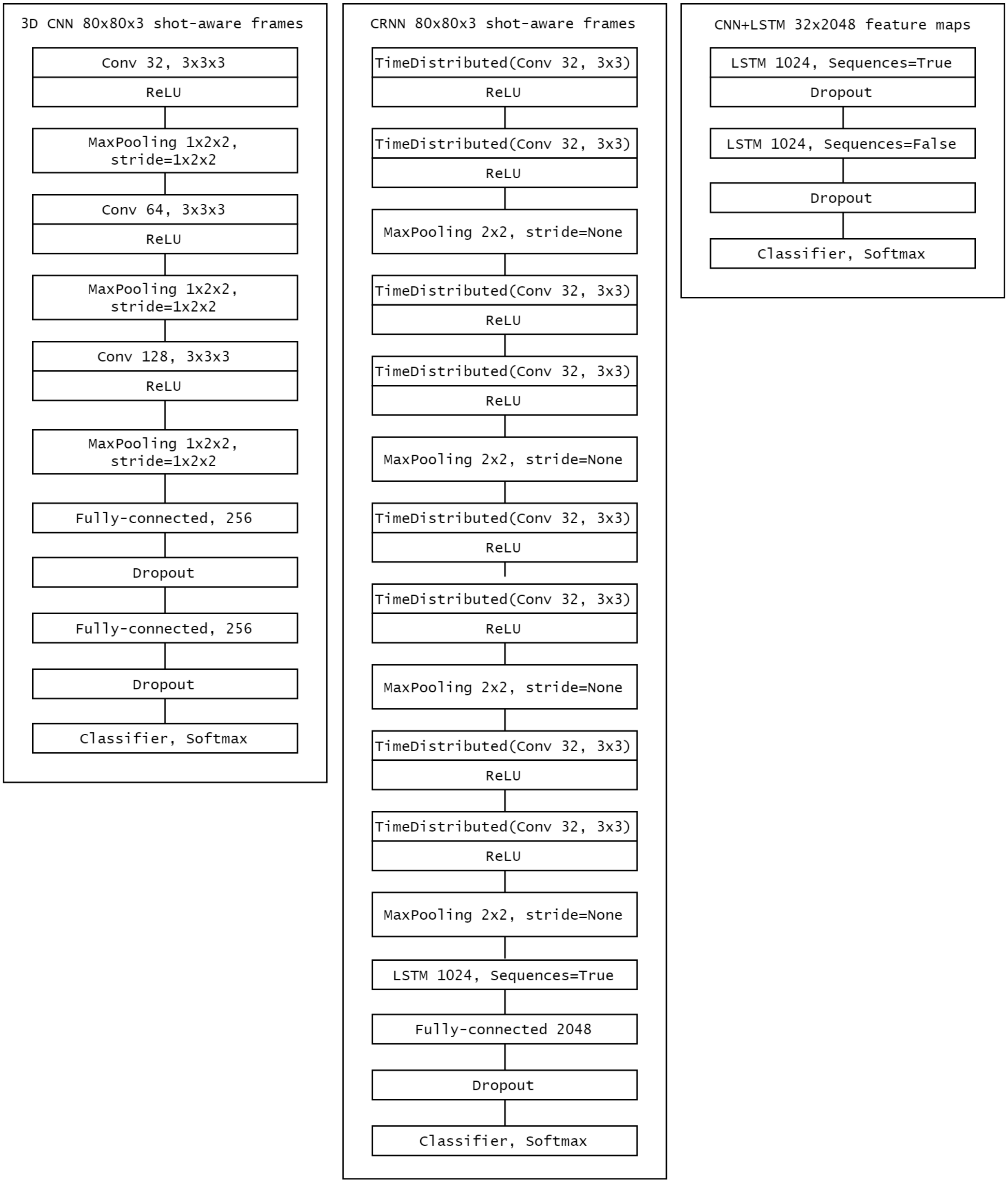
3D CNN 80x80x3 shot-aware frames
Conv 32, 3x3x3
ReLU
MaxPooling 1x2x2, stride=1x2x2
Conv 64, 3x3x3
ReLU
MaxPooling 1x2x2, stride=1x2x2
Conv 128, 3x3x3
ReLU
MaxPooling 1x2x2, stride=1x2x2
Fully-connected, 256
Dropout
Fully-connected, 256
Dropout
Classifier, Softmax
CRNN 80x80x3 shot-aware frames
TimeDistributed(Conv 32, 3x3)
ReLU
TimeDistributed(Conv 32, 3x3)
ReLU
MaxPooling 2x2, stride=None
TimeDistributed(Conv 32, 3x3)
ReLU
TimeDistributed(Conv 32, 3x3)
ReLU
MaxPooling 2x2, stride=None
TimeDistributed(Conv 32, 3x3)
ReLU
TimeDistributed(Conv 32, 3x3)
ReLU
MaxPooling 2x2, stride=None
TimeDistributed(Conv 32, 3x3)
ReLU
TimeDistributed(Conv 32, 3x3)
ReLU
MaxPooling 2x2, stride=None
LSTM 1024, Sequences=True
Fully-connected 2048
Dropout
Classifier, Softmax
CNN+LSTM 32x2048 feature maps
LSTM 1024, Sequences=True
Dropout
LSTM 1024, Sequences=False
Dropout
Classifier, Softmax

Figure 5. Detail layers from different DNN models.

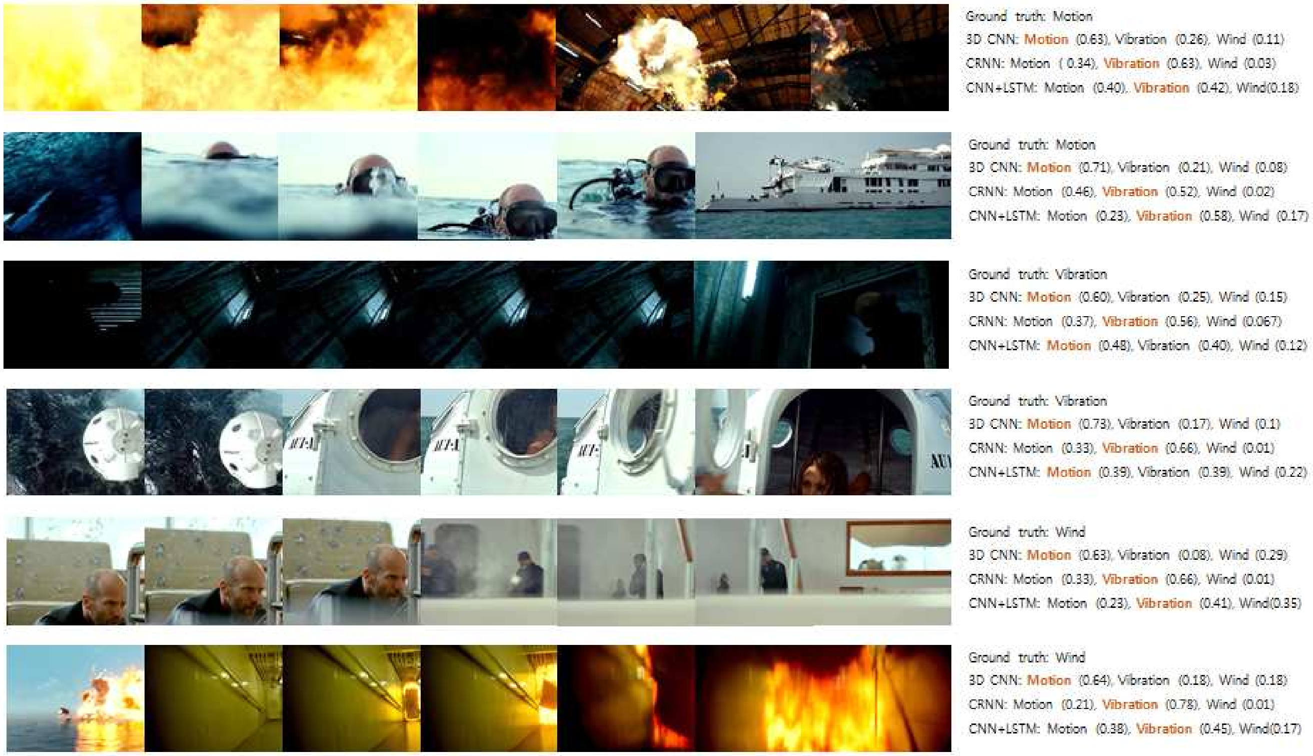
Ground truth: Motion
3D CNN: Motion (0.63), Vibration (0.26), Wind (0.11)
CRNN: Motion (0.34), Vibration (0.63), Wind (0.03)
CNN+LSTM: Motion (0.40), Vibration (0.42), Wind(0.18)
Ground truth: Motion
3D CNN: Motion (0.71), Vibration (0.21), Wind (0.08)
CRNN: Motion (0.46), Vibration (0.52), Wind (0.02)
CNN+LSTM: Motion (0.23), Vibration (0.58), Wind (0.17)
Ground truth: Vibration
3D CNN: Motion (0.60), Vibration (0.25), Wind (0.15)
CRNN: Motion (0.37), Vibration (0.56), Wind (0.067)
CNN+LSTM: Motion (0.48), Vibration (0.40), Wind (0.12)
Ground truth: Vibration
3D CNN: Motion (0.73), Vibration (0.17), Wind (0.1)
CRNN: Motion (0.33), Vibration (0.66), Wind (0.01)
CNN+LSTM: Motion (0.39), Vibration (0.39), Wind (0.22)
Ground truth: Wind
3D CNN: Motion (0.63), Vibration (0.08), Wind (0.29)
CRNN: Motion (0.33), Vibration (0.66), Wind (0.01)
CNN+LSTM: Motion (0.23), Vibration (0.41), Wind(0.35)
Ground truth: Wind
3D CNN: Motion (0.64), Vibration (0.18), Wind (0.18)
CRNN: Motion (0.21), Vibration (0.78), Wind (0.01)
CNN+LSTM: Motion (0.38), Vibration (0.45), Wind(0.17)

Figure 6. We only show some frames from total shot-aware frames. We compare the ground truth with prediction results from each model. Highlighted colors are prediction labels with highest scores.

Table 2. Mean average precision (mAP) for 4D effect classification

Model	Step-size sampling	Shot-aware frames	# of Parameters
2D CNN	0.301		23.7M
3D CNN	0.467	0.493	14.9M
CRNN	0.454	0.494	72.2M
CNN+LSTM	0.464	0.505	15.7M

0.9. The other networks are trained in video-level classification basis. We re-scale all video clips using shot-aware frame selection to yield 32 input frames. Adam optimizer is used with learning rate set to 10^{-5}. All experiments were carried out on NVIDIA GTX TITAN X 12GB GPU using Keras [5].

5.3. Result

The results of 4D effect classification on testing data are summarized in Table 2. As can be seen from the table, the baseline network, 2D CNN, achieves 0.301 mAP, lower compared to other networks. Even though deep neural networks without considering multi-shot existence in videos result in better performance than the baseline, implementing shot-aware frame selection and deep neural networks consistently outperforms these network. The shot-aware frame selection leads to substantially different results on classification despite our networks only see 32 sampled frames. The 3D CNN, CRNN, and CNN+LSTM using step-size sampling method bring 0.467, 0.454, and 0.464 mAP. The shot-aware frame selection and DNNs achieve 0.493, 0.494, and 0.505 for 3D CNN, CRNN, and CNN+LSTM, respectively. Figure 6 visualizes the performance of 3 different models using shot-aware frame selection. From the figure, we can see that models mistakenly predicts motion as vibration and vibration as motion. From these results, we realize that motion and vibration effect are very similar and difficult to distinguish. Besides, the wind effect is more difficult to predict than other effect types. This problem is arguably caused by number of videos that have wind effect is smaller that motion and vibration.

6. Conclusions

This paper proposed a new 4D effect video classification method using shot-aware frame selection and deep neural networks. We emphasize that our dataset is challenging since it consists of multi-shot videos and has high interclass yet low intra-class visual variation. After collecting a new 4D effect video dataset, we built different models of DNNs and fed these models with shot-aware frames. We showed that our shot-aware frame selection and deep neural networks consistently achieved better results than only using deep neural networks with a step-size sampling method by up to 8.8% in terms of mean average precision. For future work, we hope to incorporate different types of input such as shot metadata extracted from videos to further improve classification performance.

Acknowledgement

This work was supported by the ICT R&D program of MSIP/IITP, Republic of Korea (17ZF1200, Development of XDMedia Solution for Invigoration of Realistic Media Industry). We also would like to thank Hoon-Ki Lee, Noh-Sam Park, and Jae-Kwan Yun for their work on data collection.

References

[1] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1395–1402. IEEE, 2005.

[3] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[4] B. Castellano. Pyscenedetect. `https://github.com/Breakthrough/PySceneDetect`, 2017.

[5] F. Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[6] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[8] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] H. Kuehne, A. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014.

[11] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563. IEEE, 2011.

[12] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2929–2936. IEEE, 2009.

[13] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[14] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013.

[15] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1194–1201. IEEE, 2012.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[17] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[18] B. Shi, X. Bai, and C. Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2016.

[19] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[20] K. Soomro and A. R. Zamir. Action recognition in realistic sports videos. In *Computer Vision in Sports*, pages 181–208. Springer, 2014.

[21] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[22] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[24] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[25] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *arXiv preprint arXiv:1507.05738*, 2015.