Using Clustered Frames to Classify Videos

Itthisak Phueaksri and Sukree Sinthupinyo

Abstract—**This paper presents a semi-supervised learning technique to classify video clips. Usually, many tasks are done by categorizing video clips using deep learning techniques. However, based on the number of online videos today, it is necessary to use high computing power to accomplish this task. The authors propose methods that use Self-Organizing Map (SOM) to create a feature space representing clusters of video frames. The authors then classified them using simple voting, calculating entropy, neural networks, and Long-Short Term Memory (LSTM). The researchers also show finding frame numbers that are used to cluster video frames according to accuracy and training time. The results of this approach are presented based on testing 18 specific classes of real-world datasets from TV-programs containing 912 videos. The authors evaluated the techniques using five-fold cross-validation that our method archived 71.98% of average accuracy. Their computing time was then assessed, which achieved approximately 40 minutes of average computing time. Moreover, the researchers also compared the present proposal to other baseline models, including C3D and CNN-LSTM, and also used scene and action-recognition datasets, namely Hollywood2 to evaluate the technique. The authors archived 93.72% of average accuracy.**

Index Terms—**Computer vision, unsupervised learning, self-organizing map, LSTM.**

I. Introduction

Video classification has become a challenging task for researchers, both in terms of accuracy and computational time, even when using high-performance computing. Due to the steady increase in volume of video clips, there are also increasing numbers of video categories. This could cause issues when attempting to train a model, since computing power and time are required to complete training tasks that must support all of today's videos.

Classifying video techniques have been presented based on understanding video information, for example, frames, audio, and optical flow. Zha, S., Luisier, F., Andrews, W., Srivastava, N., & Salakhutdinov, R. [1] proposed a method based on action recognition using the Fisher Vector technique to compute feature scores from frame images. They then combined all gathering computing information using a fusion model.With the breakthrough of computing technology, adding model dimensions has become an alternative technique to classify videos. Meanwhile, Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. [2] presented a method using a three-dimensional Convolutional Neural Network, Improved Dense Trajectories (IDT), and Support Vector Machine (SVM). However, using convolutional neural network models is a technique that ignores the sequence of video information. Implementing recurrent neural network techniques such as Long-short Term Memory (LSTM) has become a popular technique to recognize videos. Wu, Z., Wang, X., Jiang, Y. G., Ye, H., & Xue, X. [3] presented a method using a Convolutional Neural Network model to extract image features and optical flow features. They then passed all extracting information to LSTM models and applied a fusion model for the final prediction. Yet classifying videos using current techniques consumes significant computing costs and time, which is likely to increase in pace with the rising number of videos.

The authors hence present a semi-supervised learning technique to reduce the feature space of videos using the Self-Organizing Map. Input features from cluster results are then used instead of video frames in the classification process. The experiments were conducted using four classification methods, namely plurality vote, summarized entropy, Neural Networks, and integrating SOM spaces with an LSTM model. The authors aimed to implement these techniques on real-world video datasets, so datasets were collected from television programs containing 912 videos. The datasets were labeled into 18 classes of video datasets.

Manuscript received November 14, 2019; revised May 11, 2020.

The authors are with Chulalongkorn University, Bangkok, 10500, Thaliand (e-mail: 6170982521@student.chula.ac.th, sukree.s@chula.ac.t).

II. Related Work

Machine learning has become one of the most popular techniques to classify videos using several features. Karpathy *et al*. [1] proposed a classification model using a two-dimensional Convolutional Neural Network to extract features from video frames in images with several resolutions. They then computed feature scores from a multi-resolution classification model using the Fisher Vector and fused their information together. Due to contemporary hardware advancements, increasing categories of Convolutional Neural Network has become a standard technique for classifying videos. Tran *et al*. presented a training approach of a three-dimensional Convolutional Neural Network model to classify videos. They also implemented Improved Dense Trajectories (IDT) and Support Vector Machine (SVM) [2]. Nonetheless, using a Convolutional neural network model is a technique that ignores consideration of motion sequence. They consequently attempted to implemented Recurrence Neural Network (RNN) to analyze the motion sequence. Long-Short Term Memory (LSTM) is one of a series of techniques that are selected for use in video classification. Wu *et al*. [3] presented integrating Long-Short Term Memory with Convolutional Neural Network models. They implemented the CNN model to extract features from images and optical flows. But some video classification works attempted to classify videos using unsupervised learning methods by

doi: 10.18178/ijmlc.2020.10.4.973

implementing clustering methods to recognize action and motions [4]. They also used clustering techniques to determine the combining of human activities and described them [5].

The clustering image is a technique to group images using unsupervised techniques. The most popular clustering technique is K-Mean to cluster grayscale biomedical images [6], [7]. Yet clustering images using the K-means algorithm is not able to achieve good results since image features are non-linear. Researchers considered implementing hybrid techniques, such as combining Particle Swarm Optimization (PSO) and the K-Means algorithm to cluster images based on content [8], while others used non-linear techniques by implementing feed-forward Neural Network called Non-linear Subspace Clustering (NSC) [9]. Moreover, the Self-Organizing Map (SOM) is also a technique for image clustering. Mo *et al.* presented a self-organizing map for image clustering, cluster refinement, and cluster merging [10].

The self-organizing map is a form of unsupervised learning that uses a Neural Network to create neuron space. Training of the self-organizing model is done through neural nodes adjusting from learning through training sets that are in a vector form. Kohonen [11] used a learning algorithm to adjust neuron nodes that were proposed by Teuvo Kohonen. It adjusts the weights of neurons by calculating the Euclidean distance between input vectors and the best matching node which is called the Best Matching Unit (BMU) [12]. Finding a distance between vectors in the Self-Organizing map is called finding a neighborhood, and a function for calculating it is called neighborhood function. The Gaussian function is one of the popular techniques that are implemented to calculate distance.

Furthermore, there was also an application of a self-organizing map to classify videos. Farooq, Jalal, and Kamal [13] presented the use of SOM to find clusters of human action based on action recognition. Furthermore, Wickramasinghe, Amarasinghe, and Manic. [14] proposed a technique to improve a SOM model by adding CNN as the top layers of SOM to extract image features before finding clusters of images to maintain SOM size.

III. Dataset

The growth of videos nowadays has increased both videos and categories, for example, vlogs, news, games casting, and TV programs. While these categories have become conventional labeling for most video datasets, classifying specific categories of videos remains a challenging task.

As this work focused on presenting video classification methods to classify specific video categories, the authors built the datasets based on eighteen different classes of videos from several channels which consisted of 912 videos. Each of the video has a different playback length with various production methods, including studio-based, outdoor production, mixed (studio-based and outdoor production), vlog, and gaming scenes. Fig. 1 presents examples of the video dataset.

The authors therefore assembled video datasets from multiple data sources that represent the different classes, consisting of fourteen TV programs, two news TV channels, a vlog, and a game casting channel. The video datasets were collected in an mp4 encoded format to reproduce the datasets and evaluate against other methods. Due to the intention to present frame-based video classification in our work, the authors further prepared all videos by extracting all the first frame images of each video. The image size was reduced to obtain an original image size of 224 pixels wide and 224 pixels high. Next, the authors labeled all videos into 18 categories to represent all the frames of each video within the same class. The datasets demonstrate the evaluation with five-folds cross-validation. They were hence split into five portions.

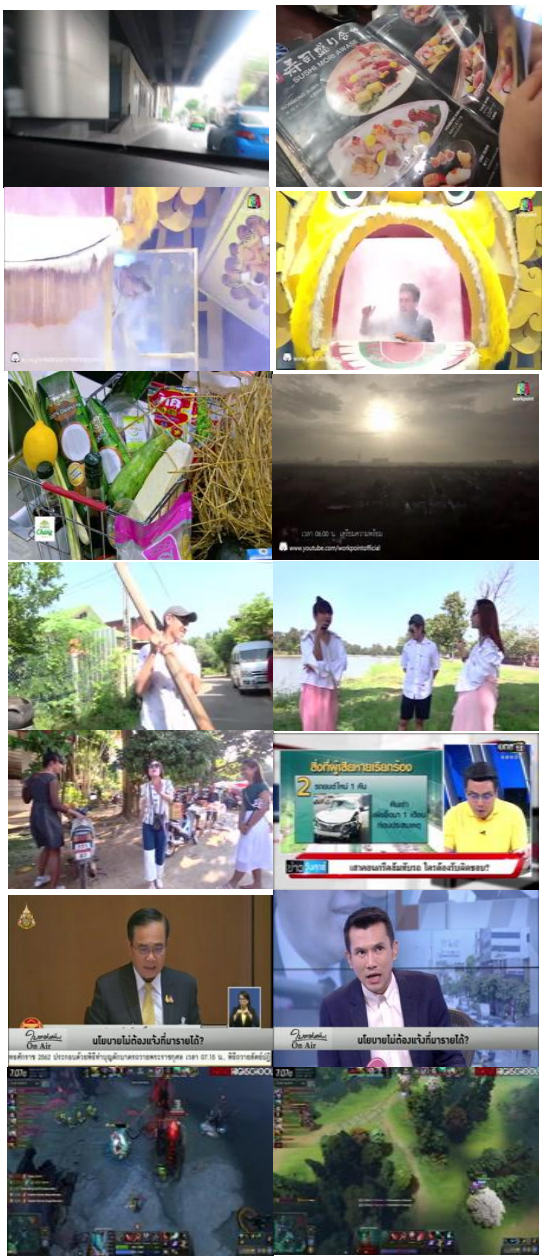

Fig. 1. Examples of the 18 classes of TV programs, including samples of vlog videos, studio production videos, outdoor videos, news videos, and game videos.

Nevertheless, this paper also presents an evaluation of all presented techniques with other benchmark datasets that are based on frame-based video classification, for example, UCF 101 [15] and Sprot1m [16]. Some datasets that are related to the study were also evaluated. YouTube8m [17] is the most popular dataset to evaluate video classification model which

is proposed by YouTube. It contains over 6.8m datasets for the purpose of assessing large-scale video classification. TRECVID [18] is recommended for high-resolution video for scene-based classification and aims to identify the categories of difference key-frames of each video. Moreover, Hollywood2 [19] is another dataset which consists of action recognition dataset and scene recognition with the aim of classifying differences of action and scene. The researchers hence applied Hollywood2 as an evaluation benchmark for this study.

IV. Experiments

The experiment presented the use of semi-supervised learning by combining unsupervised learning with simple methods and supervised learning. The researchers implemented the self-organizing map which is unsupervised learning to cluster the images. Then, the clustered results were integrated with diverse approaches, including simple voting, entropy, a neural network, and an LSTM model. Additionally, the researchers introduced obtaining a decent number of input frames from uncertainty video playback times in the pre-processing.

A. Self-organizing Map

The self-organizing map is a technique to build feature space and was proposed by Kohonen [11]. This work consequently presented building the Self-Organizing map (SOM) to create space for video features. The authors built SOM models indifference map dimensions among 20, 40, and 80 nodes. The experiment demonstrates 40 nodes for width and 40 nodes for height which was achieved using the best results. The self-organizing map training algorithm is demonstrated in Algorithm I. The Adam algorithm was further implemented to optimize the training method.

ALGORITHM I: Self-Organizing Map Training Algorithm

Start	Initialize weight vectors for nodes and iterator is 0
Step I	Build input vectors from video frames
Step II	Find Euclidian distance between inputs and weights
Step III	Update weight with the best matching unit (BMU)
Step IV	Set iterator + 1 to iterator Do *step II* if iterator less than 20

The Gaussian function was implemented to be a neighborhood function to find the best matching unit (BMU). The equation to find the best matching unit is shown in Equation 1.

$$W_v(s+1) = W_v(s) + \theta(u,v,s) \cdot \alpha(s) \cdot (D(t) - W_v(s)) \quad (1)$$

The Gaussian function is shown to be a neighborhood function to find BMU in Equation 2.

$$\theta(u,v,s) = \exp\left(-\frac{[d(u,v)]^2}{2\sigma^2(t)}\right) \quad (2)$$

B. Pre-processing

Due to the uncertain lengths of the datasets, the total and size of frames became the highest requirement for the computing cost. The researchers hence analyzed frame numbers between 100 to 900 frames. First, input sets were built from datasets by reducing the frame size of the datasets from 224 pixels wide to 28 pixels, and 224 pixels high to 28 pixels. We next selected frames among 100, 300, 500, 700, and 900 frames from each video. We recognized selected frames would be representative of each video. Therefore, we implemented calculating several skipping frames to select frames in each video. A skipping number is calculated by dividing a total frame number by several selected frames, for example, selecting 100 images from 5,000 video frames. The skipping number is 500 to select frames in this video. We tested each selected frame number by splitting 80 percent of the datasets to be train sets, and 20 percent to be test sets. Table I shows the number of training and testing sets. The researchers trained each dataset by implementing a self-organizing map with 40 × 40 nodes.

TABLE I: Number of Frames to Evaluate an Input Number

Number of frames	All datasets	Train sets	Test sets
100	91,200	71,700	19,500
300	273,600	215,100	58,500
500	456,000	358,500	97,500
800	729,600	573,600	156,000
900	820,400	645,300	175,500

Simple voting was implemented to evaluate each frame number by mapping all test sets to the self-organizing map to find the cluster number index of the frame. We subsequently defined a class of each node from the most frequent class that appears in their nodes. We hence defined the most nodes in the self-organizing map as classes of videos. However, some nodes were analyzed that were not able to represent the class and declared them as undefined nodes to ignore them in the final prediction. Algorithm II presents the mapping techniques used in this research.

ALGORITHM II: Mapping Input Features to Locations

Start	Collect all weights of the SOM map and all map locations in the same index.
Step I	Select input vector
Step II	Find an index of minimum value from calculating Euclidian norms by subtraction of an input vector and all weights
Step III	Get a map location from an index in step II

In step II of the algorithm, we calculated the norm of input vectors and weights with the Euclidian norm for calculating a distance of vectors is n-dimensional, with the formula shown in Equation 3.

$$norm = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2}$$

TABLE II: Prediction Accuracy with a Voting Method

Number of frames	Accuracy (%)
100	59.47
300	60.00
500	65.79
700	58.33
900	57.29

We evaluated the number of frames by training the self-organizing map models with different numbers three times. We then selected the best accuracy of all the training. Table II shows the final prediction accuracy of each frame

number and also shows the relationship between accuracy and training in Fig. 2.

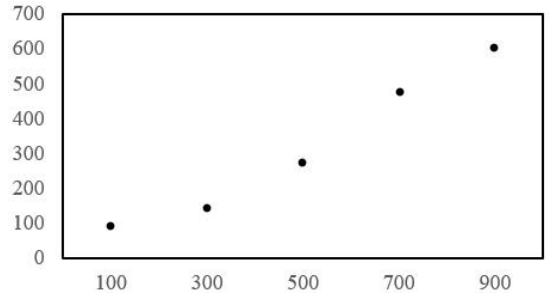

Fig. 2. Training time (mins) of each frame number.

To find the prediction accuracies and training times of each frame number, we found that increasing the number of frames was not a factor that of achieving high accuracy. Meanwhile, rising numbers of frames consumes greater computation-time. The results from the table and graph represent the accuracy reduction after increasing frame numbers from 500 frames to 700 and 900 frames. We then considered training times, even if using 100 frames consumed less computation-time than 500 frame training, but the accuracy of 100 frames was significantly different from 500 frames. We hence selected the 500 frames number to be our input feature number.

C. Entropy

Because of analyzing mapping results from the self-organizing map model, we found disorders and randomness of classes in the clustered results. We hence implemented calculating entropy as a feature selection method to find a level of disorder and randomness of the clusters. We present Equation 3 as a formula to calculate the entropy of each class of every cluster.

$$\text{Entropy} = \sum_{i=1}^{C} - p_i \log_2 p_i \quad (3)$$

We calculated the entropy of the classes in each cluster by mapping all training sets to the self-organizing map model. The mapping results represent several classes of videos. We next calculated the entropy of all categories that occur in the clusters and defined classes that did not occur in groups to be zero. We also normalized entropy values with the total number of frames. The algorithm to calculate entropy for each cluster is shown in Algorithm III. The final prediction of calculating entropy summarizes the entropy of each class. We consequently selected categories of each video from the maximum.

ALGORITHM III: CALCULATING ENTROPY FOR EACH CLUSTER

Start	Collect all clusters
Step I	Select a cluster
Step II	Count all appeared classes in the cluster
Step III	Calculate entropy of each class and define not appear class to zero.
Step IV	Multiple each entropy values with the ratio of counting class in this cluster to a total of train sets.

However, the maximum entropy values represent the randomness of the classes. We also analyzed the inverse of the entropy value of the classifying videos. The results of the inverted entropy in our experiment did not present an improvement of the prediction accuracy.

D. Neural Network

Neural Network has become a traditional model to predict the output from a learning sample dataset. As a result of the self-organizing map that represents 40 × 40 nodes of video space, we hence built the nodes to be input features of a neural network model. Our building in Algorithm IV shows the transforming 40 × 40 nodes of each video space to input feature as a vector in 1600 nodes. We then encoded 18 classes of videos into five nodes as output vectors of the final prediction.

ALGORITHM IV: GENERATING INPUT FEATURE

Start	Define empty 2d array and set m, n to zero
Step I	Set 2d array
Step II	Count all appeared classes in the cluster
Step III	Calculate entropy of each class and define not appear class to zero.
Step IV	Multiple each entropy values with the ratio of counting class in this cluster to a total of train sets.
Step V	Do *Step I* for all input features

Due to the vector of inputs and outputs, we hence built a neural network model for the final prediction. The model architecture includes 1,600 nodes of the input layer, 800 nodes of the first hidden layer, 128 nodes of the second layer, and five nodes of the output layers. We also implemented the ReLu function as the activation function of the model. Fig. 3 represents the model architecture.

Input layer	1600 nodes
Activation	ReLu
Hidden layer 1	1600 nodes
Activation	ReLu
Hidden layer 2	800 nodes
Activation	ReLu
Hidden layer 3	128 nodes
Activation	ReLu
Output layer	Five nodes
Activation	Softmax

Fig. 3. A neural network architecture with 1600 nodes of an input layer with ReLu activation, 1600 nodes of a hidden layer one with ReLu activation, 800 nodes of a hidden layer two with ReLu activation, 128 nodes of a hidden layer three with ReLu activation, and five nodes of an output layer with a Softmax activation.

We trained a model with all training sets that were split into 400 batch sizes within 40 epochs of training. We further monitored the validation value in training to pick the model that achieved the best lowest loss.

E. LSTM

Long-Short Term Memory (LSTM) is a popular model to recognize the information in sequence. Due to investigating the information on video content, they represent a sequence of images of content information. The researchers assumed that a sequence of clusters should have an essential improvement for accuracy. Under this assumption, we implemented the Long-Short Term Memory that would analyze the occurred sequence of clustered in the training process.

We presented integrating the self-organizing map to the Long-Short Term Memory. We then encoded 1,600 neuron

nodes of the self-organizing map into 16 bits for being input features for Long-Short Term Memory. We then found sequences of clusters from sorted frames in each video. Algorithm V shows the steps for building the video input.

ALGORITHM V: GENERATING INPUT FEATURES FOR LSTM

Start	Encode 1,600 neuron nodes to 16 bits
Step I	Select video from all input features
Step II	Collect all video clusters with frames sorted.
Step III	Map all sorted clusters to encoded nodes
Step IV	Repeat *Step I* for all videos

Our Long-Short Term Memory consisted of 500 channels of 16 input nodes at the input layer, two layers of 500 channels of LSTM layer, and five nodes of the softmax output layer. All input and hidden layers of this model were implemented ReLu as the activation function. Fig. 4 shows the model architecture.

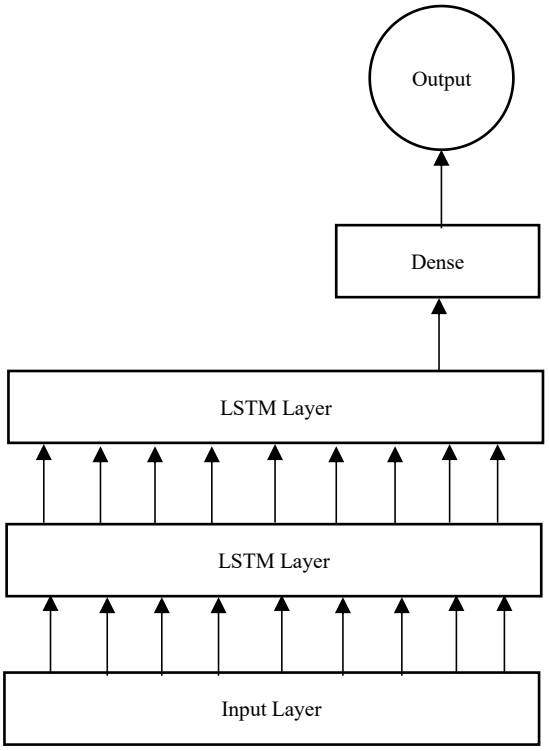

Fig. 4. LSTM model to classify video with SOM results, including 500 nodes of input layers, 1,000 nodes of the first LSTM layer, 500 node of the second LSTM, and five nodes of dense (output layer).

We trained a Long-Short Term Memory model with 400 batch size of train sets within 40 epochs. We consequently selected the best model by monitoring validation loss value.

V. EVALUATION

The evaluation addressed finding the input frame numbers of video datasets by examining adjusting the input features from 100 to 900 frames. The researchers then presented a comparison between prediction accuracy from our techniques with other models. Moreover, we also discussed the computation cost of each model.

Due to addressing the number of frames, we showed a self-organizing map model with a different number of input features from starting at 100 input features. We then added 200 input features in each experiment until it reached 900. We then used a simple vote to evaluate accuracy. By comparing the accuracy, we decided to use 500 frames as a decent frame number.

We analyzed the accuracy from the final prediction, which consisted of calculating entropy, training a neural network model, and training an LSTM model. We hence evaluated all final prediction methods with five-fold cross-validation, with the considered accuracies shown in Table III.

TABLE III: RESULTS OF EACH PREDICTION MODEL

K	Entropy	Neural Network	LSTM
1	60.08	73.85	73.17
2	54.48	71.80	71.46
3	55.37	71.64	68.14
4	66.48	72.39	71.08
5	58.79	71.54	66.58
	59.18	71.98	70.9

However, we chose our best accuracy using a neural network to evaluate our techniques by comparing it with other state-of-the-art techniques. We selected baseline models based on the difference in model architectures. We selected a large-scale model that implemented a three-dimensional Convolutional Neural Network called C3D and a model that integrated Convolutional Neural Network to Long Short Term Memory. Table IV details the training results of all the models.

TABLE IV: RESULTS OF EACH MODEL

K	Our model	C3D	LSTM
1	73.85	60.00	61.71
2	71.80	60.62	62.24
3	71.64	59.96	60.13
4	72.39	60.11	59.07
5	71.54	59.67	60.07
	71.98	60.07	60.64

We further analyzed the computation time of classifying video without transfer learning from the other models. We hence collected them from processing all models in the same environment, which included 16 vCPU, 104 GB memory, and 2 GPU of Tesla T4. Table V shows the approximation of averaged computing-time.

TABLE V: BENCHMARK DATASET ACCURACY

Models	Accuracy (%)
Our model	**93.72**
SalCLSTM	93.33
ACLNet	91.13

Moreover, we also evaluated our best technique that implements a neural network model to Hollywood2 datasets. We consequently compared our method with results from the other models [20], including SalCLSTM+AUC-J and

ACLNet+AUC-J, with Table VI comparing the average accuracy results.

VI. Conclusion

This paper introduced the use of semi-supervised learning to classify videos and also applied a classification model to closely detail eighteen dataset classes. We additionally represented obtaining a sampling frame number of unreliable video playback time. We hence integrated the self-organizing map with sample voting which demonstrated five-hundred sampling frames that achieved the best results of the final prediction. We examined the application of entropy to attain the final prediction. That does not represent enhancing prediction accuracy when compared with simple voting. We then implemented a simple neural network model to calculate an LSTM model to derive the final prediction. The experimental results of both show nearby results, while an LSTM model consumes higher resources than the other methods. The experiments result from five-fold cross-validation achieved 72.8% of average accuracy, which is the highest accuracy when comparing them with other states-of-the-art techniques. We also analyzed the computation times of our model with other baseline models. Our model achieved approximately 320 minutes of training time, while other methods consumed over 420 minutes of computing time.

Furthermore, we validated our techniques with the benchmark dataset named Hollywood2, with both scene recognition and action recognition datasets. Our model achieved 93.7% average accuracy, more than the other methods.

Notwithstanding, large-scale datasets remain a challenging task in video classification work. We further look to improve our techniques to handle them. We mostly realize the strength of using convolutional layers to extract features that reasonably increase clustering performance and reduce computing time.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Phueaksri I., Sinthupinyo S. conducted the research; All authors had approved the final version.

References

[1] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov, "Exploiting image-trained CNN architectures for unconstrained video classification," *arXiv preprint arXiv:1503.04144*, 2015.

[2] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proc. the IEEE International Conference on Computer Vision*, 2015, pp. 4489-4497.

[3] Z. Wu, X. Wang, Y. G. Jiang, H. Ye, and X. Xue, "Modeling spatial-temporal clues in a hybrid deep learning framework for video classification," in *Proc. the 23rd ACM international conference on Multimedia*, 2015, pp. 461-470.

[4] B. Peng, J. Lei, H. Fu, C. Zhang, T. S. Chua, and X. Li, "Unsupervised video action clustering via motion-scene interaction constraint," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[5] O. Sener, A. R. Zamir, S. Savarese, and A. Saxena, "Unsupervised semantic parsing of video collections," in *Proc. the IEEE International Conference on Computer Vision*, 2015, pp. 4480-4488.

[6] N. Dhanachandra, K. Manglem, and Y. J. Chanu, "Image segmentation using K-means clustering algorithm and subtractive clustering algorithm," *Procedia Computer Science*, vol. 54, pp. 764-771, 2015.

[7] R. Supriyanti, A. Rifai, Y. Ramadhani, and W. Siswandari, "Characteristics identification of myeloblast cell using K-means clustering for uncontrolled images," *International Journal of Machine Learning and Computing*, vol. 9, no. 3, pp. 351-356, 2019.

[8] Z. S. Younus, D. Mohamad, T. Saba, M. H. Alkawaz, A. Rehman, M. Al-Rodhaan, and A. Al-Dhelaan, "Content-based image retrieval using PSO and k-means clustering algorithm," *Arabian Journal of Geosciences*, vol. 8, no. 8, pp. 6211-6224, 2015.

[9] W. Zhu, J. Lu, and J. Zhou, "Nonlinear subspace clustering for image clustering," *Pattern Recognition Letters*, vol. 107, pp. 131-136, 2018.

[10] H. Mo, B. Xu, W. Ouyang, and J. Wang, "Color segmentation of multi-colored fabrics using self-organizing-map based clustering algorithm," *Textile Research Journal*, vol. 87, no. 3, pp. 369-380, 2017.

[11] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464-1480, 1990.

[12] S. Ng and M. Chan, "Effect of neighbourhood size selection in SOM-based image feature extraction," *International Journal of Machine Learning and Computing*, vol. 9, no. 2, pp. 195-200, 2019.

[13] A. Farooq, A. Jalal, and S. Kamal, "Dense RGB-D map-based human tracking and activity recognition using skin joints features and self-organizing map," *KSII Transactions on Internet & Information Systems*, vol. 9, no. 5, 2015.

[14] C. S. Wickramasinghe, K. Amarasinghe, and M. Manic, "Parallalizable deep self-organizing maps for image classification," in *Proc. 2017 IEEE Symposium Series on Computational Intelligence*, 2017.

[15] 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.

[16] A. Karpathy, G. Toderici, S. Shetty *et al*., "Large-scale video classification with convolutional neural networks," in *Proc. the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725-1732.

[17] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[18] Z. Wu, Y. G. Jiang, J. Wang, J. Pu, and X. Xue, "Exploring inter-feature and inter-class relationships with deep neural networks for video classification," in *Proc. the 22nd ACM International Conference on Multimedia*, pp. 167-176, 2014, ACM.

[19] M. Marszałek, I. Laptev, and C. Schmid, "Actions in context," in *Proc. CVPR 2009-IEEE Conference on Computer Vision & Pattern Recognition. IEEE Computer Society*, 2009.

[20] P. Linardos, E. Mohedano, J. J. Nieto, N. E. O'Connor, X. Giro-i-Nieto, and K. McGuinness, "Simple vs complex temporal recurrences for video saliency prediction," *arXiv preprint arXiv:1907.01869*, 2019.

Itthisak Phueaksri was born in Nonthaburi, Thailand in 1989. He has a bachelor's degree in computer engineering from Rajamagala University of Technology Phra Nakhon. He is currently studying computer science at the Department of Computer Engineering, Chulalongkorn University. His research interests include machine learning, computer vision, unsupervised learning, and semi-supervise learning.

Sukree Sinthupinyo received his bachelor's degree, master's degree, and Ph.D. in computer engineering from Chulalongkorn University, Thailand. His main areas of research include artificial intelligence, machine learning & pattern recognition, and engineering.