Motion Embedded Images: An Approach to Capture Spatial and Temporal Features for Action Recognition

Tri Le[1,3], Nham Huynh-Duc[1,3], Chung Thai Nguyen[1,3] and Minh-Triet Tran[1,2,3]
[1]Faculty of Information Technology, University of Science, VNU-HCM
[2]Software Engineering Lab, University of Science, VNU-HCM
[3]Vietnam National University, Ho Chi Minh City

Keywords: action recognition, motion embedded image, sports dataset, two-stream network

Received: March 22, 2023

The demand for human activity recognition (HAR) from videos has witnessed a significant surge in various real-life applications, including video surveillance, healthcare, elderly care, among others. The explotion of short-form videos on social media platforms has further intensified the interest in this domain. This research endeavors to focus on the problem of HAR in general short videos. In contrast to still images, video clips offer both spatial and temporal information, rendering it challenging to extract complementary information on appearance from still frames and motion between frames. This research makes a two-fold contribution. Firstly, we investigate the use of motion-embedded images in a variant of two-stream Convolutional Neural Network architecture, in which one stream captures motion using combined batches of frames, while another stream employs a normal image classification ConvNet to classify static appearance. Secondly, we create a novel dataset of Southeast Asian Sports short videos that encompasses both videos with and without effects, which is a modern factor that is lacking in all currently available datasets used for benchmarking models. The proposed model is trained and evaluated on two benchmarks: UCF-101 and SEAGS-V1. The results reveal that the proposed model yields competitive performance compared to prior attempts to address the same problem.

Povzetek: Raziskava predstavi model za prepoznavanje človeških aktivnosti iz videov in testira model na novi bazi video posnetkov jugovzhodne Azije.

1 Introduction

The task of human activity recognition (HAR) pertains to the labeling of actions or activities observed within video clips. In recent years, the proliferation of online social platforms has led to an exponential increase in the volume of media data being uploaded, with short-form videos dominating the internet landscape, beginning with Tiktok and now extending to Facebook, Instagram, and Youtube. Consequently, the need for HAR has become increasingly crucial across a range of domains, including content monitoring, classification, and recommendation systems, video retrieval, human-computer interaction, and robotics.

In contrast to a still image, a video clip affords not only static spatial information confined within a single frame but also temporal information that results from integrating spatial information across frames to capture dynamic motions.

There exists a plethora of research investigating the challenging task of video classification. Currently, the majority of high-accuracy results have been obtained using 3D convolutional kernels to capture the temporal information within videos [1][7][3]. Nonetheless, this architecture may be cost-prohibitive to employ in practical scenarios due to its high computational requirements. Consequently, certain approaches prioritize computational efficiency to handle larger datasets, yet may not be suitable for real-world applications [26][15][2]. These methods often necessitate powerful processors to train successfully. Conversely, training Convolutional Neural Networks (ConvNets) to acquire temporal information in videos offers a straightforward, albeit effective alternative. Researchers following this approach vary in their methods for processing original frames, such as fusing temporal information early or late in the network [11], or combining multiple sequential frames to generate optical flow information [18]. Motivated by the positive outcomes of these studies and the effectiveness of ConvNet models in image recognition, we seek to explore the performance of ConvNet models for video classification. Notably, the extraction of temporal information in short videos remains a less explored domain, likely owing to its inherent difficulty. This paper introduces a novel approach for embedding both temporal and spatial features of consecutive video frames into images, thereby enabling effective recognition of the static features of a scene, such as objects, context, and entities, as well as the motion information. Specifically, we incorporate this method into a variant of the two-stream ConvNet model. The first stream leverages the images generated by our approach to detect motion in videos, while the second stream employs a conventional image classification network to recognize spatial information, utilizing single still video frames as inputs. This latter

stream aims to identify and preserve any spatial information that might be lacking in the former.

To evaluate the performance of action recognition models, various publicly available datasets such as UCF-101 [19] and UCF Sport [17] have been introduced, containing 101 action and 10 sport classes, respectively. Some datasets attempt to cover a broader range of activities by including more classes[11][12], while others incorporate user-uploaded data from multiple media sources such as Youtube and Vimeo to simulate daily human activities [8][5]. Despite these efforts, most video datasets lack the complexity of videos edited using text, filters, and effects that are prevalent in short-form videos on social networks like Tiktok, Facebook, and Youtube. These limitations can lead to inaccurate benchmarking of models when applied to this new form of video content. In this research, we also aim to collect a novel dataset that includes both non-effected and effected clips. Inspired by previous datasets [17][11], we gathered data within the same Sport category and focused on South-East Asian Game sports. Our dataset, SEAGS_V1, consists of **8** sports classes and **1,168** videos sourced from Youtube and Tiktok. The availability[1] of this dataset will enable researchers to evaluate the performance of their models on a more diverse range of video content.

In this study, we evaluate the performance of our proposed MEI Two-stream network on two widely-used action recognition datasets, UCF-101 and SEAGS_V1. To investigate the potential of our approach further, we also experiment with different backbone architectures and integrate them into an EnsembleNet. Our empirical results demonstrate that our proposed method holds considerable promise in enhancing the accuracy of Activity Recognition on short-form videos.

The content of this paper is organized as follows. In Section 2, we briefly review existing work related to action recognition. Then we present our proposed method in Section 3. We discuss our experiments in Section 4. Finally, the conclusion and future work are discussed in Section 5.

2 Related Work

The early-stage methodologies employed for video classification tasks typically involve a three-stage process. Firstly, visual features of a video segment are extracted densely [20] or at a sparse set of interest points[14]. Secondly, these extracted features are combined into a fixed-sized video-level description. Lastly, a classifier, such as a SVM, is trained on the resulting ”bag of words” representation to discriminate between the pertinent visual classes. Subsequently, ConvNets have replaced all three stages with a single neural network that is end-to-end trainable. However, there are several approaches to augment the connectivity of a ConvNet in the time domain, exploiting local spatio-temporal information[9] [11]. However, these approaches are challenged by the limitations of ConvNets in capturing motion information among frames, leading to the loss of temporal features.

[1]SEAGS_V1 is currently available online here.

2.1 Two-stream architecture

To mitigate the aforementioned challenge, researchers investigated a novel two-stream ConvNet architecture [18] [21] [25]. This architecture involves feeding the input videos into two distinct streams: the spatial and temporal streams. Each stream employs a deep ConvNet, with softmax scores combined by late fusion. Notably, the inputs for each stream differ slightly. The spatial stream processes individual video frames to recognize actions from still images. In contrast, the temporal stream works on pre-computed optical flow features using optical flow estimation techniques, such as [23].

2.2 Spatial-temporal feature fusion method

The two-stream architecture has inspired numerous studies, with many seeking to improve its performance by focusing on two key areas: the fusion stage and the temporal stream. In an effort to optimize the fusion stage, Feichtenhofer *et. al.* conducted a comprehensive investigation of various approaches to fusing the two networks over space and time [4]. They ultimately discover that fusing a spatial and temporal network at the convolution layer instead of the softmax layer results in comparable performance, while also significantly reducing a substantial number of parameters. Another approach involves using a separate architecture to combine image information. Yue *et. al.* explored two video-classification methods [22] which are both capable of aggregating frame-level ConvNet outputs into video-level predictions: Feature Pooling methods max-pool local information through time, while LSTM’s hidden state evolves with each subsequent frame.

2.3 Variations of temporal stream

Various approaches have been explored in an effort to improve the performance of the temporal stream in the two-stream architecture. Zhang *et. al.* investigates the replacement of optical flow with motion vector, which can be obtained directly from compressed videos without additional calculation [24], resulting in a more than 20x speedup compared to traditional two-stream approaches. However, motion vectors tend to lack fine structures and contain noisy and inaccurate motion patterns, leading to a decline in recognition performance. An alternative approach involves learning to predict optical flow using a supervised ConvNet. Ng. *et. al.* proposes a multitask learning model, ActionFlowNet, that trains a single stream network directly from raw pixels to jointly estimate optical flow while recognizing actions with ConvNet, capturing both appearance and motion in a single model [16].

In this study, we build upon the ideas of the two-stream architecture [18] and modify the temporal stream. Rather than relying on optical flow, we introduce a novel approach that embeds motion into the original frames, generating motion-embedded images that retain spatial features in the temporal stream. This is based on the belief that motion and appearance should not be separated. However, the spatial stream is still considered, as our current method for gener-

ating motion-embedded images may contain noisy and inaccurate motion patterns caused by background movement.

3 Proposed Method

In this section, we introduce our novel approach called motion embedded image (MEI) and two-stream network. The input video is fed into two distinct streams, the normal and motion streams, as illustrated in Figure 1. The processes in these streams are implemented separately. Prior to being input into the streams, the input can be pre-processed. These inputs are then fed into a ConvNet to perform image classification, and the prediction scores of both streams are fused to produce the final prediction. In the following subsections, we provide comprehensive details of the motion embedding technique, motion stream, normal stream, and fusion stage.

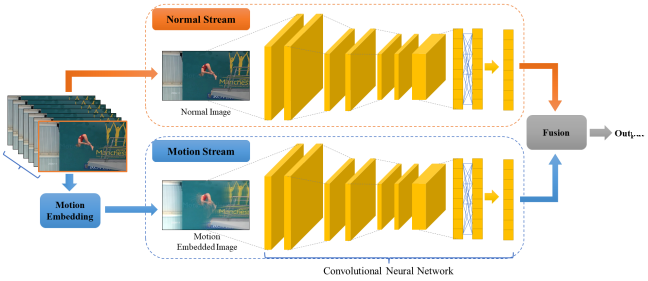

Figure 1: Illustration of our proposed two-stream architecture. Normal stream (top) takes individual frames as inputs, while Motion stream (bottom) requires motion embedded images which are a combination of consecutive video frames. Then, the convolutional neural networks in both streams learn to classify them. Finally, a fusion algorithm is performed to combine normal-motion information. Both streams are end-to-end trainable.

3.1 Motion Embedding

As per the requirements of the Motion stream, the input video frames must undergo a motion embedding stage. Our proposed motion embedding techniques are illustrated in Figure 2, which depict the workflow involved in this stage. The resulting output of this stage is motion-embedded images that convey the direction and order of motion of a single image. Furthermore, we believe that the spatial and temporal information stored simultaneously gives more features for Convolutional Neural Network to learn, which is described in detail in a later sub-section.

All frames extracted from the input video are orderly numbered as T and segmented into batches consisting of N consecutive frames. Each batch is fed into the motion embedding stage, which comprises two components: image processing and combinator. The image processing component is responsible for generating new images from origins, while the combinator aggregates the processed images to create motion-embedded images. It is noteworthy that the aggregation of consecutive frames in a video emphasizes the parts containing static objects and contexts, highlighting the contours of the different stages in the motion that can be easily distinguished from the static parts. The combinator is often dependent on the method used in the image processing component. In the following sub-section, we present our studies about two methods for processing images and their corresponding combinators.

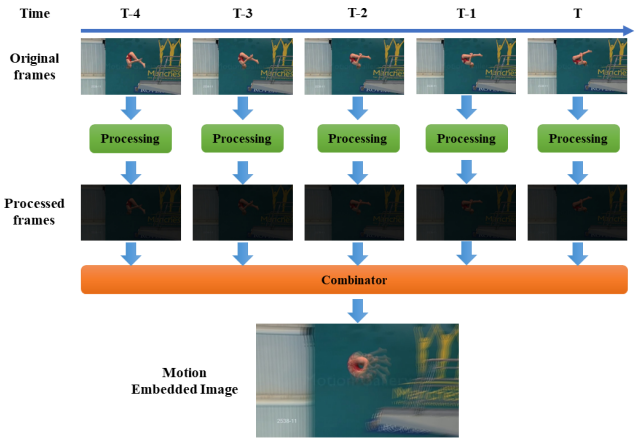

Figure 2: Workflow of our motion embedding technique. The figure illustrates a batch of N=5 consecutive frames from an input video before and after processing which uses the Equal Division method. A combinator, then, merges all processed frames to generate relevant motion embedded images.

3.1.1 Equal division

To ensure that all frames contribute equally to MEI, we divide the values of all pixels in each frame by N. This technique also enables the combinator to keep the pixel values between 0 and 255. The formula for this technique is presented below:

$$processed\_img = original\_img * \frac{1}{N}$$

In the formula, $processed\_img$ and $original\_img$ are 2D arrays representing the pixel values of the processed and original frames, respectively. The operation is performed element-wise.

The combinator we suggest for this method is simply a summation of all processed images. Therefore, the final MEI for a batch concluding at frame T is formulated by the following equation:

$$\text{MEI}_T = \sum_{i=T-N+1}^{T} processed\_img\ i \qquad (1)$$

In Figure 2, a batch of five consecutive frames from an input video is depicted, which is processed through the motion embedding stage using the Equal Division method. As evident from the figure and equations, it is obvious that the final MEI likely presents a stack of images. Due to the identical contributions of all frames to the final image, the motion transitions are presented in a uniform manner throughout the sequence.

3.1.2 Gradient division

The Equal Division method is limited in that it fails to capture the directionality of the motion, as it presents all action steps in an identical manner. To overcome this limitation, we propose the Gradient Division method. This method prioritizes the most recent frame in a batch to serve as the base frame for activity recognition and appropriately weights the contribution of each frame in the batch, with later frames carrying higher weights than earlier ones.

The following describes our proposed formulas for image processing component:

$$sum\_N = \sum_{i=1}^{N} i, \quad contrib = \frac{T \ mod \ N + 1}{sum\_N}$$

$$processed\_img = original\_img * contrib$$

In the above formula, $processed\_img$, $original\_img$ are 2D arrays of the processed and original frames' pixel values, respectively. The equation is performed element-wisely. The two scalars sum_N, $contrib$ are aimed to calculate the contribution of frame T in a batch of N frames.

The combinator we suggest for this method is similar to the formula 1 for the Equal Division combinator.

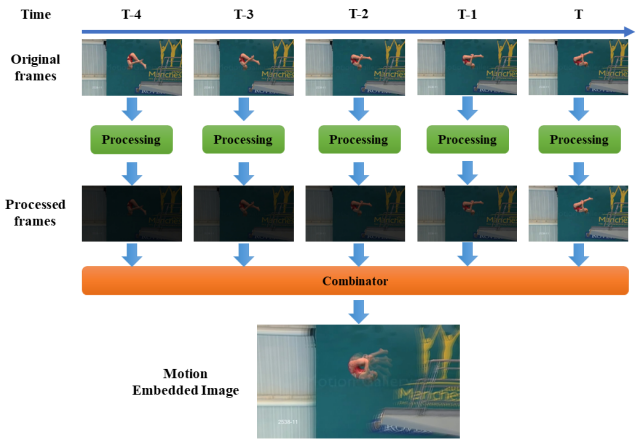

Figure 3: Workflow of our motion embedding technique. The figure illustrates a batch of N=5 consecutive frames from an input video before and after processing which uses the Gradient Division method. A combinator, then, merges all processed frames to generate relevant motion embedded images.

Figure 3 shows a batch of 5 consecutive frames from an input video. It is fed into the motion embedding stage using the Gradient Division method. As shown in the figures and formulas above, the later frames in the batch contribute more to the final output image. This leads to a much better presentation of the direction of action in final motion embedded images. We believe that based on this motion trail, Convolutional Neural Network can learn temporal and spatial information simultaneously.

3.2 Motion stream

The motion stream proceeds in a sequential manner, where batches of N consecutive frames are sequentially fed into the stream. The motion stream operation involves two primary stages. Firstly, the input batch is transformed into an MEI through the motion embedding stage. Subsequently, the generated images are processed by a ConvNet to predict the spatial-temporal features from MEI.

3.3 Normal stream

Initially, we endeavored to investigate the feasibility of employing MEI exclusively for action recognition. However, our experiments revealed that contemporary motion embedding techniques tend to retain motion trails from extraneous objects and backgrounds, resulting in suboptimal outcomes. Consequently, we discerned that static appearance remains a valuable source of information, given its capacity to capture immobile objects without motion trails. Accordingly, we resolved to supplement our approach by adding a normal stream to perform classifications grounded in still images. This stream comprises an image classification ConvNet architecture and can be enhanced by leveraging recent breakthroughs in large-scale image recognition methods [13]. By pre-training this network on a comprehensive image classification dataset, such as the ImageNet challenge dataset, we can further enhance its predictive capabilities.

The normal stream is designed to process individual video frames. In each batch, the most recent frame, referred to as the base frame when using Gradient Division for the motion stream, is extracted and fed into the Convolutional Neural Network (CNN) of this stream.

3.4 Fusion stage

The predictions generated by the two streams of image classification are integrated through a fusion process to produce the ultimate prediction output. At present, our approach to this fusion stage is to compute the arithmetic mean of the predictions, as explicated by formula 2.

$$pred(x) = \frac{normal\_pred(x) + motion\_pred(x)}{2} \quad (2)$$

where x indicates the input image and $normal\_pred$, $motion\_pred$ and $pred$ present the prediction of normal, motion stream, and the final prediction result, respectively.

4 Experiments and Results

4.1 Dataset

4.1.1 UCF-101

The UCF-101 dataset [19] is a prominent benchmark for evaluating the performance of human action recognition models. The dataset comprises a diverse collection of 101 action classes, spanning over 13,000 clips and 27 hours of video data. Notably, the dataset features realistic user-uploaded videos that capture camera motion and cluttered backgrounds. To evaluate the performance of our approach, we adopt the split-test 01 provided by the authors of this dataset.

4.1.2 SEAGS_V1

We present a novel dataset, SEAGS_V1, that features a diverse mix of effect and non-effect videos.

Our dataset is obtained from a variety of video platforms, including Youtube, TikTok, and Facebook reels. We leverage normal videos as the base data for actions, while short videos with added image effects, text, and stickers serve to enrich the dataset for improved recognition of short effect videos. Figure 6 showcases some examples from our dataset that include text and stickers. Short videos of less than 20 seconds are included in their entirety, except for the intro and outro, while longer videos are manually split into 2-4 segments that are 5-20 seconds in duration.

To facilitate our experiments, SEAGS_V1 is structured in the same manner as UCF-101, with videos organized into folders corresponding to their respective class labels.The name of the video is formatted as

`v_<class label>_<index>.mp4`

We also provide the following files:

`classInd.txt` file contains index of each class label.

`testlist.txt` file contains the path to testing videos accounting for 30% of dataset.

`trainlist.txt` file contains the path to training videos accounting for 70% of dataset.

After data collection, SEAGS_V1 is completed with 8 classes. Each class consists of 100 - 160 videos, each video is between 1 and 20 seconds long. Figures 4, 5 and Table 1 show the statistics of SEAGS_V1 dataset.

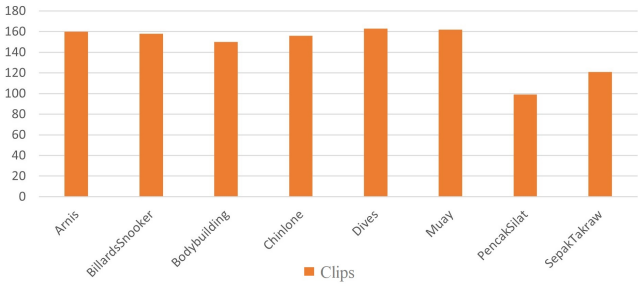

Figure 4: Statistical chart of the clip amount of classes

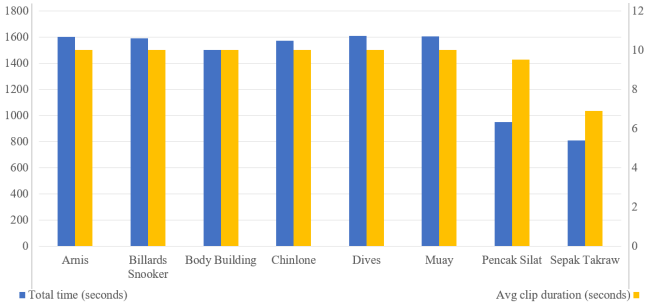

Figure 5: Statistical chart of the total time and average video duration of classes

4.2 Data Augmentation

Upon close examination of our dataset, SEAGS_V1, we figure out that many behaviors are labeled with the same action class, yet differed only in their direction. To further augment the dataset and facilitate learning in these cases, we implemented a data augmentation technique that involves flipping the original images. Figure 6 shows some examples of flipped and original video frames from our dataset.

Table 1: An overview of the SEAGS_V1 dataset

Actions	8
Clips	1169
Total Duration	188 m
Mean Clip Length	9.64 s
Min Clip Length	1.0 s
Max Clip Length	20.0 s
Audio	No

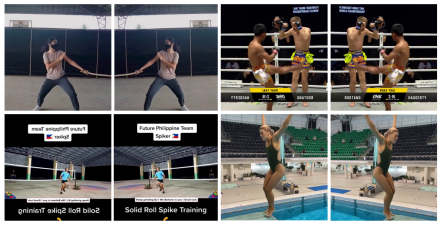

Figure 6: Some flipped and original video frames from dataset SEAGS_V1

4.3 Image classification backbones

For UCF-101, we consider to use EfficientNetB0 as the backbone. For SEAGS_V1, we conduct experiments using a range of backbones, including EfficientNetB0, DenseNet201, InceptionNetV3, ResNet50, and MobileNetV2. Moreover, we explore the potential benefits of ensembling multiple base ConvNet models into a stronger classifier, which we refer to as EnsembleNet, by summing the probability prediction of each model.

$$ensemble\_net(x) = \frac{1}{K}\sum_{k}^{K} base\_net_k(x)$$

where x indicates the input image and K represents the number of base models.

4.4 Motion embedding implementation

We use some specific parameters to create embedded motion images, namely $N = 10$ and interval_frames = 5 for SEAGS_V1 and $N = 10$ and interval_frames = 10 for UCF-101.

Here, interval_frames refers to the distance, in terms of frame count, between two consecutive batches or the distance from the first frame of batch k to the first frame of batch $k + 1$. Each embedded motion image is generated from a batch of N frames. As depicted in Figure 7, a comparison of three types of images - normal image, MEI with

Gradient Division, and Equal Division - highlights the effectiveness of Gradient Division in preserving the direction of motion in activities, whereas Equal Division does not. Accordingly, we employ Gradient Division as the method for the motion embedding process in our experiments. Figure 8 shows some motion-embedded images from both the SEAGS_V1 and UCF-101 datasets.

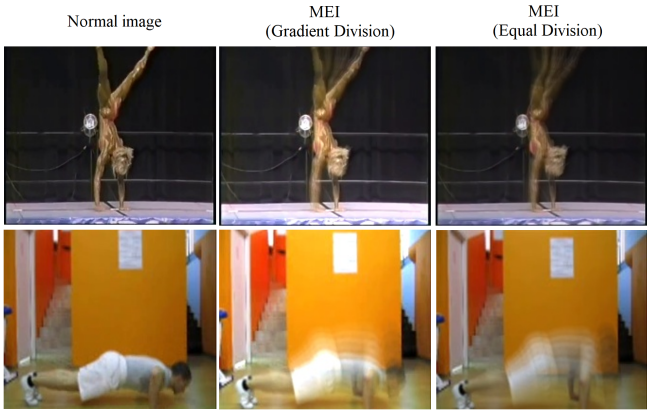

Figure 7: Some examples of normal image (left), MEI with Gradient Division (middle) and with Equal Division (right) from two datasets

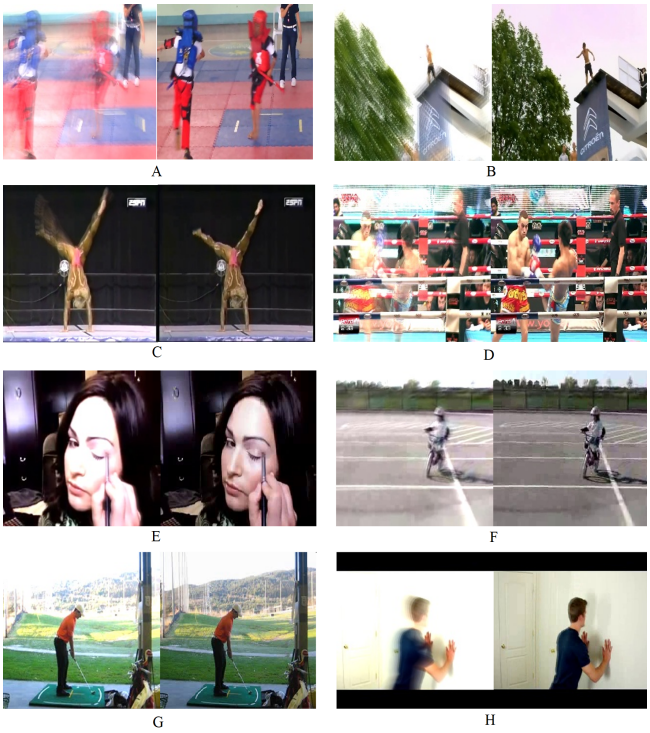

Figure 8: Some motion embedded (left) and its original images (right) from SEAGS_V1 (A, B, C, D) and UCF-101 (E, F, G, H) datasets

4.5 Training

We partition the dataset into training and validation sets at a ratio of 7:3. We conclude the training process once the validation accuracy exceeded 0.9. Notably, training with normal images requires only 10 epochs to achieve the desired validation accuracy, whereas training with MEI takes 50 epochs. Each stream is trained independently, and the probabilities are subsequently fused for prediction purposes.

4.6 Two-stream implementation

We train both the spatial and temporal streams using the same model architecture, albeit independently. The motion stream is fed with the MEIs generated using the parameters specified in the previous section. During testing, the normal stream processes all the last frames of the batches to make predictions.

4.7 Results

Our experimental results on the UCF-101 dataset demonstrate that our proposed method achieved significantly higher accuracy than the initial models developed by Soomro *et. al.* [19], Karpathy *et. al.* [11], and a two-stream model [6]. However, when compared to the original two-stream model [18] and the state-of-the-art approach developed by Wang *et. al.* [10], our method exhibits a noticeable performance gap, as shown in Table 2.

Table 2: Experiment result on UCF-101 dataset (split test 01) (ours with backbone EfficientNetB0)

Model	Accuracy (%)
Soomro et al [19]	43.9
Karpathy et al [11]	65.4
Han et al [6]	68.0
Simonyan et al [18]	88.0
Kalfaoglu et al [10]	**98.69**
Ours (with normal image)	68.54
Ours (with MEI)	67.04
Ours (Two-stream)	70.08

Table 3: Experiment result on SEAGS_V1 dataset with normal image

Backbone	Accuracy (%)
EfficientNetB0	84.9
DenseNet201	89.2
MobileNetV2	87.2
ResNet50	64.1
InceptionV3	86.9
Ensemble (5 base models)	**92.9**

(Done on 1/10 of the total frames of each video)

Overall, the experimental results presented in Tables 2, 3, and 4 suggest that the accuracy of models trained with MEIs is marginally lower than that of models trained with normal images. In particular, the incorrect predictions of MEI-based models are primarily observed in videos with

Table 4: Experiment result on SEAGS_V1 dataset with motion embedded image

Backbone	Accuracy (%)
EfficientNetB0	88.3
DenseNet201	87.5
MobileNetV2	81.5
ResNet50	52.7
InceptionV3	85.8
Ensemble (5 base models)	**92.9**

Table 5: Experiment result on SEAGS_V1 dataset with proposed two-stream model

Backbone	Accuracy (%)
EfficientNetB0	**90.02**
DenseNet201	89.46
MobileNetV2	88.89
ResNet50	60.11
InceptionV3	88.32

moving contexts, where the MEIs generated from these videos make it difficult for the models to distinguish between actions and context, resulting in suboptimal performance. Figure 8 (B, F) provides examples of poorly generated MEIs from such videos. In contrast, normal images are found to preserve clear visual information among objects, even in the presence of moving contexts.

Conversely, MEIs exhibit a distinct advantage in videos with static or minimally moving contexts, where they can effectively highlight the motion of activities that may not be apparent in normal images. Figure 8 provides examples of such scenarios (A, C, D, H). Hence, the fusion of these two types of images in a two-stream architecture significantly improves the accuracy of the final result on both datasets, as evidenced by the results presented in Tables 2 and 5. Notably, in cases where the motion of activities is relatively consistent, MEIs and normal images exhibit similar characteristics, and the models can effectively learn spatial information. Figure 8 (E, G) provides examples of such cases.

5 Conclusion

In this paper, we propose an approach of applying motion embedded Image (MEI) in a human activity recognition two-stream ConvNet model for short-form videos. We also propose an unprecedented dataset called SEAGS_V1, which consists of both non-effected and effected short videos of 8 local Southeast Asian Sports.

Currently, our experiments on UCF-101 and SEAGS_V1 datasets show that combining the motion stream with the normal spatial stream gives significantly better results than using each stream as an independent model. Moreover, ConvNet models using the ensembled backbone have notably higher accuracy than those using only one backbone. The derived results show a promising potential of the model to advance prediction efficiency in the human activity recognition problem.

Extra training data is beneficial for our model to learn spatial and temporal information, so we are planning to train it on large video datasets such as Sports-1M. Our next direction is to modify the architecture so it can focus more on the activity instead of the whole image and the extracted information will not be diluted. The most important improvement plan is to make the motion stream retain more spatial information so the model only consists of one motion stream and becomes more lightweight.

Acknowledgement

This research is supported by research funding from Honors Program, University of Science, Vietnam National University - Ho Chi Minh City.

References

[1] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. https://doi.org/10.48550/arXiv.1705.07750.

[2] C. Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 203–213, 2020. https://doi.org/10.1109/cvpr42600.2020.00028.

[3] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. https://doi.org/10.48550/arXiv.1812.03982.

[4] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016. https://doi.org/10.1109/cvpr.2016.213.

[5] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. https://doi.org/10.1109/iccv.2017.622.

[6] C. Han, C. Wang, E. Mei, J. Redmon, S. K. Divvala, Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Yolo-based adaptive window two-stream convolutional neural network for video classification. 2017.

[7] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d

cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. https://doi.org/10.1109/cvpr.2018.00685.

[8] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015. https://doi.org/10.1109/cvpr.2015.7298698.

[9] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013.

[10] M. E. Kalfaoglu, S. Kalkan, and A. A. Alatan. Late temporal modeling in 3d cnn architectures with bert for action recognition. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 731–747. Springer, 2020. https://doi.org/10.1007/978-3-030-68238-5_48.

[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. https://doi.org/10.1109/cvpr.2014.223.

[12] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017. https://doi.org/10.48550/arXiv.1705.06950.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. https://doi.org/10.1145/3065386.

[14] Laptev and Lindeberg. Space-time interest points. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 432–439 vol.1, 2003. https://doi.org/10.1109/iccv.2003.1238378.

[15] J. Lin, C. Gan, and S. Han. Temporal shift module for efficient video understanding. *CoRR*, abs/1811.08383, 2018. https://doi.org/10.48550/arXiv.1811.08383.

[16] J. Y.-H. Ng, J. Choi, J. Neumann, and L. S. Davis. Actionflownet: Learning motion representation for action recognition. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1616–1624. IEEE, 2018. https://doi.org/10.1109/wacv.2018.00179.

[17] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. https://doi.org/10.1109/cvpr.2008.4587727.

[18] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014. https://doi.org/10.48550/arXiv.1406.2199.

[19] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012. https://doi.org/10.48550/arXiv.1212.0402.

[20] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR 2011*, pages 3169–3176, 2011. https://ieeexplore.ieee.org/document/5995407.

[21] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream convnets. *CoRR*, abs/1507.02159, 2015. https://doi.org/10.48550/arXiv.1507.02159.

[22] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015. https://doi.org/10.1109/cvpr.2015.7299101.

[23] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. volume 4713, pages 214–223, 09 2007. https://doi.org/10.1007/978-3-540-74936-3_22.

[24] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang. Real-time action recognition with enhanced motion vector cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2718–2726, 2016. https://doi.org/10.1109/cvpr.2016.297.

[25] Y. Zhao, K. Man, J. Smith, K. Siddique, and S.-U. Guan. Improved two-stream model for human action recognition. *EURASIP Journal on Image and Video Processing*, 2020, 06 2020.

[26] Y. Zhu, Z. Lan, S. Newsam, and A. Hauptmann. Hidden two-stream convolutional networks for action recognition. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 363–378. Springer, 2019. https://doi.org/10.48550/arXiv.1704.00389.