I3D Light - A Simple Motion Information Stream for I3D

Ruikang Luo †*, Francois Rivest ‡, Farhana Zulkernine †
‡ Queen’s University
† Queen’s University and Royal Military College (RMC)

Abstract
Vision-based Human Activity Recognition (HAR) aims to recognize human activities based on the analysis of video data, and has extensive applications in modern industry and human life. Inflated 3D (I3D) is a deep learning architecture that is commonly used for action recognition by using two-stream video data: RGB stream and optical flow stream. I3D achieved great success on a variety of action recognition benchmarks. However, the use of optical flow incurs high computational cost, making the approach unsuitable for real-time applications. We propose an alternative simple motion information extractor to replace the optical flow branch and reduce the computational cost. It is a modified I3D that uses 128 frames of 112x112 images as input. The low spatial resolution and long temporal range of the proposed I3D RGB stream can reduce the spatial information and enhance the motion information. Experiments show that this simple motion stream can increase the accuracy of the original I3D spatial stream by 4.09% on the Kinetics 400 dataset.

Keywords: HAR, I3D, 3D-CNN, real-time, motion-information

1. Introduction
As technology continues to advance, researchers are developing deep learning models that use machine cognition to assist humans in various aspects of daily life. One area where these models can be applied is in Human Activity Recognition (HAR) systems. These systems use cameras or sensors to collect data about human activity and apply cognitive models to predict and recognize different actions [1]. Camera-based HAR systems have numerous potential applications, including detecting criminal activities and monitoring the well-being of older individuals in their homes, and monitoring patients in hospitals [2].

For the algorithms used in camera-based HAR systems, traditional methods relied on action descriptors which had to be defined manually [2][3][4]. Deep-learning methods were proposed later, which applied a layered architecture of artificial neurons to extract the data features automatically, and compose the features to predict the activity [5][6]. Convolutional Neural Networks (CNN) [5][6] and one of its variants, Graph Convolutional Networks (GCN) based deep learning models [7][8] are two of the most powerful, performant, and commonly applied deep-learning methods in this application domain. CNN methods use the RGB (red, green, blue) frames directly, while GCN methods usually use pose skeletons.

2D-CNN [5][9] and 3D-CNN based methods [6][10] have recently gained momentum in the RGB video-based HAR application domain. Some 2D-CNN methods only use still images to recognize human actions [11], while others use 2D-CNN as spatial feature extractors, followed by a time aggregation layer such as LSTM or time convolution to model the sequential data patterns [12][13]. 3D-CNN based methods can extract additional spatial features, and have better recognition accuracy than 2D-CNN based methods [6][10][14].

In order to improve the accuracy of 3D-CNN methods further, optical flow was used in a parallel data stream processing pipeline to train the model on motion features [14]. Carreira et al. proposed a new two-stream Inflated 3D ConvNet (I3D) [15], which expands the 2D ConvNet into a 3D ConvNet with an optical flow data stream processing pipeline. It achieved
a superior accuracy of 74.2% on the Kinetics dataset. However, the use of optical flow incurs considerable computational cost and makes it difficult to deploy the model in real-time HAR frameworks.

Reducing the computational cost while maintaining a high accuracy is a challenge for 3D CNN based approaches. In this paper, we propose I3D-Light, a light version of the existing I3D model. Our model replaces the optical flow with a low-resolution long-range sample of videos to obtain motion information to train the model on temporal features. The experiments show that the two RGB branches of our I3D-Light model have the same average execution time. In addition, our model achieves an accuracy of 65.09% compared to 61.00% accuracy achieved by using only the original I3D RGB branch model on the Kinetics 400 dataset [22].

For the remainder of the paper, we first review the related work in Section 2. Then we describe our proposed method in Section 3. Section 4 presents the experiments and validates the model performance. Section 5 concludes the paper with a description of future work.

2. Related Work

The 3D CNN architecture can extract spatial features from videos very well, but requires significant computational power and memory, especially for the two-stream methods [6][10][14]. It is always a challenge to achieve good accuracy and performance using fewer computing resources.

I3D made a great improvement in performance, but the requirement of computing resources is large [15]. In order to strike a good balance between speed and accuracy, Xie et al. designed a spatio-temporal Separable 3D convolutional (S3D) model [16] to reduce the computing resources required by traditional 3D convolution.

To reduce the computational cost of 3D CNN methods, Liu et al [17] proposed a new real-time convolutional architecture called the Temporal Convolutional 3D Network (T-C3D). It includes a residual 3D CNN that can capture both the appearance information of a single frame and the motion information from consecutive frames. It can learn hierarchical multi-granular spatio-temporal features quickly.

Fan et al. proposed an integrated algorithm-hardware co-design method which used an efficient 3D CNN building unit called 3D-1 bottleneck residual block (3D-1 BRB) [18] and a corresponding FPGA-based hardware architecture. Their model resulted in a nearly 37 times reduction in model size. The accuracy was improved by 5%.

In streaming video data, many video frames contain redundant information and cause wastage of computing resources while containing very little useful long-term spatio-temporal features for action recognition. Li et al. proposed a Deep Key Clips-Video Feature Fusion Framework, which can select important informative clips for dynamic model training [19].

Jiang et al. proposed a new two-branch 3D CNN structure to capture multi-resolution spatio-temporal information [20]. One branch extracted large-scale temporal features from low resolution video frames (coarse branch) with fast temporal downsampling. The other branch (fine branch) processed high resolution video with slower progressive downsampling and reduced channel capacities to extract small-scale temporal features.

Instead of selecting the key clips dynamically or applying fast temporal downsampling from the same 32 frames, we used a lower-resolution in space (spatial resolution = 112x112 compared to 224x224) and longer temporal range (temporal range = 128 frames compared to 32 frames) RGB branch to capture the motion features. The new RGB branch has a similar computing time as the original I3D RGB branch (temporal range = 32 frames, spatial resolution = 224x224).

3. Method

Carreira et al. [15] use optical flow to capture motion information and increase the accuracy in their two-stream I3D model. However, optical flow requires huge computing resources and time [16]. To address this problem, we propose a light version of the I3D model that replaces the optical flow stream, which obtains the information about spatial
displacement or motion, with a second RGB stream using lower resolution in space (112x112) and longer temporal range (128 frames). This second RGB stream reduces spatial information and enhances the motion information compared to the first RGB stream.

Inflated 3D ConvNet (I3D) is based on 2D Inception-v1 [15]. For I3D, the convolution kernels and pooling kernels of Inception-v1 are expanded into 3D. The original I3D architecture and its detailed inception submodule are shown in Figure 1.

The inflated 3D Inception-v1 (I3D) is based on 2D Inception-v1 [15]. For I3D, the convolution kernels and pooling kernels of Inception-v1 are expanded into 3D. The original I3D architecture and its detailed inception submodule are shown in Figure 1.

Inflated 3D ConvNet (I3D) is based on 2D Inception-v1 [15]. For I3D, the convolution kernels and pooling kernels of Inception-v1 are expanded into 3D. The original I3D architecture and its detailed inception submodule are shown in Figure 1.

Inflated 3D ConvNet (I3D) is based on 2D Inception-v1 [15]. For I3D, the convolution kernels and pooling kernels of Inception-v1 are expanded into 3D. The original I3D architecture and its detailed inception submodule are shown in Figure 1.

The standard two-stream I3D model [15] includes an RGB branch and an optical flow branch. Both branches use the same I3D architecture. The RGB branch uses 64 RGB frames as input. The optical flow branch uses 64 optical flow (OF) frames as input and needs a time-consuming optical flow extractor. The architecture of standard two-stream I3D is shown in Figure 2.

The standard two-stream I3D model [15] includes an RGB branch and an optical flow branch. Both branches use the same I3D architecture. The RGB branch uses 64 RGB frames as input. The optical flow branch uses 64 optical flow (OF) frames as input and needs a time-consuming optical flow extractor. The architecture of standard two-stream I3D is shown in Figure 2.

For studying comparative performance, we implemented and trained the RGB branch of the standard I3D model and our I3D-Light model using the Kinetics 400 dataset. In order to reduce training time, we use 32 RGB frames in our first I3D RGB branch. We denote I3D224 as the standard I3D RGB branch that uses 32 frames at resolution 224x224 as input. We denote I3D112 as our second RGB branch that uses 128 frames at resolution 112x112 as input for temporal information.

The published I3D pretrained model was trained on Kinetics 400 with resolution 224x224. In order to use the pretrained I3D224 weights [20] to reduce the training time, we change the pooling size of the last average pooling layer from 7x7 to 4x4.

In implementing the two-stream I3D-Light model, we use the I3D112 branch to replace the optical flow branch in the original two-stream I3D model [15]. The optical flow branch can provide highly accurate motion information, but the computing cost is prohibitively high [16]. Compared to the optical flow based model, the accuracy of our I3D112 branch is not as good, but the computing cost is much lower. We will show that our modified two-stream I3D-Light model, which uses the I3D112 branch to replace optical flow, is a good solution for real-time applications. The architecture of our modified two-stream I3D is shown in Figure 3 and the results are discussed in Section 4.
4. Implementation and Results

We selected PyTorch as our framework, and our PyTorch implementation of I3D is based on Miracleyoо's implementation [21]. Our experiment system includes 16 CPUs (Intel Xeon Gold 6130 CPU @ 2.10GHz), 64G memory, and one GPU (NVIDIA Tesla V100 PCIe 32 GB). The OS is Ubuntu 18.04 LTS. The Kinetics 400 dataset is from academic-torrents [22] which contains 400 human action classes, each containing at least 400 video clips. Each clip lasts about 10 seconds and is taken from a different YouTube video. The dataset from academic-torrents has been split into 240,618 training and 19,404 validation video clips.

Since the temporal-range of I3D112 is 128 frames and the average number of frames in a 10 second video clip is about 250-300, we removed the videos which have fewer than 100 frames. During training and validation, if the length of any video is less than 128 frames, it is looped as many times as necessary to fit the model’s input length. Our final dataset had 234,506 video clips for training and 19,044 video clips for validation.

The videos from the Kinetics 400 were converted into jpg images using the OpenCV library without adjusting the sampling rate. To preprocess images for spatial feature extraction, images were first rescaled to 256x256, and then randomly cropped to 224x224 for training and center-cropped to 224x224 for validation. To preprocess images for temporal feature extraction, image sequences were randomly cropped to the specific length (32 or 128) for training and center-cropped to the specific length (32 or 128) for validation.

During training, an SGD optimizer was used with a list of gradually decreasing learning rates. Momentum was set to 0.9 and other hyperparameters used default values. Validation was performed after training with every 1/10 of the training dataset. If the validation accuracy did not improve after looping through the whole dataset, the learning rate was decremented to the next one from our predefined list of decreasing learning rates. At the same time, the model weights were set to the weights corresponding to the best accuracy achieved so far.

In training the I3D224 model with 32 frames, we used [0.008, 0.004, 0.002, 0.0008, 0.0004, 0.0002] as the list of learning rates. We used the pre-trained weights of standard I3D (using 64 frames) [21] as the initial weight values of I3D224 (using 32 frames). The final results are shown in Table 1.

To train I3D112, we used [0.01, 0.001, 0.0008, 0.0004, 0.0002] as the list of learning rates. The pre-trained weights of standard I3D (using 64 frames) [21] were used as the initial weight values of I3D112 (using 128 frames). The final results are shown in Table 1.

The proposed two-stream I3D uses the simple fusion method as in the original two-stream I3D. The outputs from the I3D224 branch and the I3D112 branch are averaged before the softmax function is applied. Finally, we trained I3D224 and I3D112 together to improve the accuracy further. The learning rate schedule used was [0.001, 0.0001, 0.00008]. The final results of the proposed two-stream I3D (I3D224 + I3D112) are shown in Table 1.

<table>
<thead>
<tr>
<th>Model</th>
<th>Input</th>
<th>Validation Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>I3D224</td>
<td>32 frames</td>
<td>61.09%</td>
</tr>
<tr>
<td>I3D112</td>
<td>128 frames</td>
<td>60.02%</td>
</tr>
<tr>
<td>I3D224+I3D112</td>
<td>32 frames + 128 frames</td>
<td>65.09%</td>
</tr>
</tbody>
</table>
In order to compare the computing resources required by I3D224 and I3D112, we used a python package, torchinfo [23], to estimate the total memory size and the total number of addition and multiplication operations required by the PyTorch model. The estimated values as shown in Table 2 are very close for I3D224 and I3D112. We also computed the average prediction or scoring time for I3D224 and I3D112 on our system using 100 predictions on one video that shows very similar performance of 0.028 when rounded to the third digit. As a comparison, the average execution time of the optical flow branch in the original I3D was computed using Gunnar Farneback’s algorithm in OpenCV [24] and took 0.638s. The results are shown in Table 2.

<table>
<thead>
<tr>
<th>Model</th>
<th>Estimated Total Memory Size (MB)</th>
<th>Total mult-adds (G)</th>
<th>Average Execution Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I3D224</td>
<td>12490.13</td>
<td>944.80</td>
<td>0.028</td>
</tr>
<tr>
<td>I3D112</td>
<td>12528.13</td>
<td>949.73</td>
<td>0.028</td>
</tr>
<tr>
<td>I3D224+I3D112</td>
<td>25018.26</td>
<td>1894.53</td>
<td>0.056</td>
</tr>
<tr>
<td>Optical flow branch in I3D</td>
<td>N/A</td>
<td>N/A</td>
<td>0.638</td>
</tr>
<tr>
<td>I3D224+optical flow I3D</td>
<td>N/A</td>
<td>N/A</td>
<td>0.666</td>
</tr>
</tbody>
</table>

Table 2. Comparison of computing resources needed by I3D224 and I3D112

4.1. Discussion

From Table 1, it can be seen that I3D224+I3D112 can increase the accuracy of I3D224 by 4.09%. From Table 2, it can be seen that computing resources required by I3D224 and I3D112 are very close. The average execution times of I3D224 + I3D112 is 0.056(s). It can handle 17.9 frames per second. If we downsample the videos by half, our computing system can handle 35.7 frames per second as needed by regular real-time applications (The common frame rate of video is 30fps [25]).

5. Conclusion

In this paper, we proposed a simple alternative solution, I3D112, to the time-consuming optical flow branch in I3D. Since I3D112 reduces the spatial size from 224x224 to 112x112 and expands the temporal range from 32 to 128, the computing resources required by I3D112 are the same as I3D224. I3D224+I3D112 only used approximately about 1/12 the computing time of I3D224+optical flow I3D, and still improved the accuracy of I3D224 by 4.09% on Kinetics 400. I3D224+I3D112 can provide a good solution for real-time applications. However, the improvement of accuracy is not ideal compared to the original two-stream I3D, and future work should seek an improved network architecture to replace I3D112.

Acknowledgements

This project was supported by Defence Research and Development Canada (DRDC).

References


