SWTA: Sparse Weighted Temporal Attention for Drone-Based Activity Recognition

Santosh Kumar Yadav
College of Science and Engineering
National University of Ireland
Galway, H91TK33, Ireland
santosh.yadav@nuigalway.ie

Esha Pahwa
Department of CSIS
Birla Institute of Technology and Science
Pilani - 333031, India
f20180675@pilani.bits-pilani.ac.in

Achleshwar Luthra
Department of CSIS
Birla Institute of Technology and Science
Pilani - 333031, India
f20180401@pilani.bits-pilani.ac.in

Kamlesh Tiwari
Department of CSIS
Birla Institute of Technology and Science
Pilani - 333031, India
kamlesh.tiwari@pilani.bits-pilani.ac.in

Hari Mohan Pandey
Computing and Informatics
Bournemouth University
Fern Barrow, Poole BH12 5BB, United Kingdom
hpandey@bournemouth.ac.uk

Abstract—**Drone-camera based human activity recognition (HAR) has received significant attention from the computer vision research community in the past few years. A robust and efficient HAR system has a pivotal role in fields like video surveillance, crowd behavior analysis, sports analysis, and human-computer interaction. What makes it challenging are the complex poses, understanding different viewpoints, and the environmental scenarios where the action is taking place. To address such complexities, in this paper, we propose a novel Sparse Weighted Temporal Attention (SWTA) module to utilize sparsely sampled video frames for obtaining global weighted temporal attention. The proposed SWTA is divided into two components. First, temporal segment network that sparsely samples a given set of frames. Second, weighted temporal attention, which incorporates a fusion of attention maps derived from optical flow, with raw RGB images. This is followed by a basenet network, which comprises a convolutional neural network (CNN) module along with fully connected layers that provide us with activity recognition. The SWTA network can be used as a plug-in module to the existing deep CNN architectures, for optimizing them to learn temporal information by eliminating the need for a separate temporal stream. It has been evaluated on three publicly available benchmark datasets, namely Okutama, MOD20, and Drone-Action. The proposed model has received an accuracy of 72.76%, 92.56%, and 78.86% on the respective datasets thereby surpassing the previous state-of-the-art performances by a margin of 25.26%, 18.56%, and 2.94%, respectively.**

Index Terms—**Human Activity Recognition, Video Understanding, Drone Action Recognition**

Introduction

Human Activity Recognition (HAR) is one of the developing research areas where human actions are determined based on the surroundings and the movement of one's body parts. Its applications lie in various fields such as virtual reality, video surveillance, security, crowd behavior analysis, human-computer interaction, and many more. It comprises two main sub-tasks: classification and localization. While classification results in finding what a human is performing, localization refers to where the action is taking place in a scene of a video. Our study encompasses action classification using an efficient method to capture temporal data along with the spatial information in a single stream model while including operations to handle small objects. The model is trained and evaluated on three complex datasets, $i.e.$, Okutama-Action [1], Drone-Action [2], and MOD20 [3], and successfully achieve promising results.

Human Action Recognition poses numerous challenges that need to be taken care of. The computational cost that is invested into training the 2D models alone is huge, let alone 3D ConvNets. It takes a good amount of days to train a 3D CNN backbone and hence, delays the search of finding an optimal architecture for the task, and side by side it is bound to overfit the training information. Keeping the duration of model training aside, one should also note that HAR does not have a fixed dictionary of human activities. This can result in intraclass diversity. A person can be running or jogging but the limb movements in both activities remain similar. Some datasets even involve human-to-human and human-to-object interactions which can be a challenging task. To address this, accurate and differentiating features need to be developed.

Videos taken from long distances, such as those of video surveillance cameras, also act as a potential barrier to finding the right action performed as they can not deliver high-quality videos where the person can be seen clearly. Performance recognition also differs from the type of camera used. In events such as sports tournaments, dynamic portable recording devices with embedded cameras and smart glasses are used. On the other hand, real-time videos are full of visuals and contain variations in brightness levels, making it challenging to see actions in complex situations. Background activities performed by neighboring objects along with variation in scale, viewpoint, and partial occlusion also affect the model outcomes.

Owing to the recent surge in the literature on this topic, a large number of studies have been conducted on human

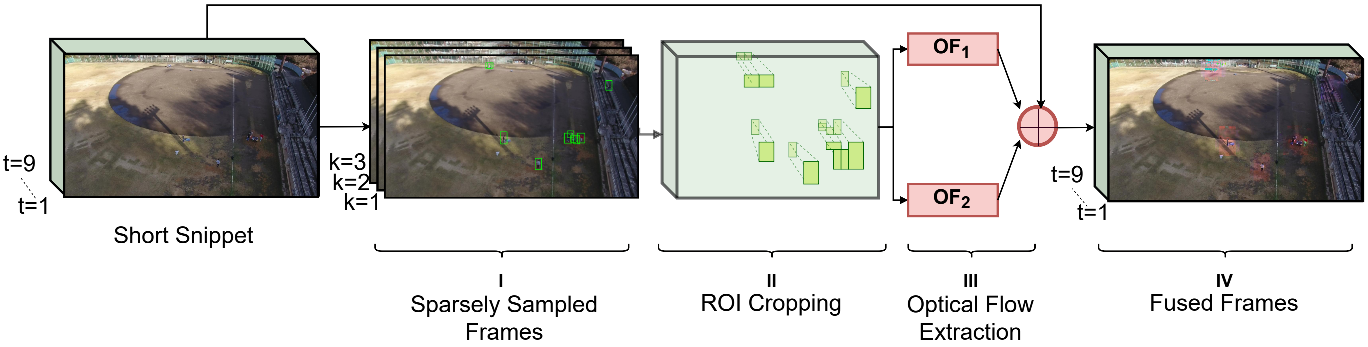

Fig. 1. Detailed description of the SWTA module. I) shows sparsely sampled frames obtained from a short snippet using a segment-based sampling technique. II) depicts how ROI(s) is(are) cropped across frames. III) and IV) illustrate optical flow extraction and their fusion with raw RGB frames respectively. Here values taken for t and k are for the demonstration purpose.

activity recognition [4] [5] [6]. Multi-modal methods for HAR have gained popularity over the last few years. That being said, for a model to be successfully deployed for use, it has to be efficient and accurate. Due to these reasons, works such as Persistent Appearance Network (PAN) [7] and Temporal Shift Module (TSM) [8], which can be utilized with both 3D and 2D CNNs, were brought about. Various types of CNN architectures $e.g.$ [9] and [10] implement two-stream networks of 2D-CNNs whereas works like [11] are excellent example of 3D-CNN networks being both efficient and precise in their prediction. To learn long-range information from videos, Wanget al. proposed Temporal Segment Networks (TSN) [12] which uses segmented samples prior to feeding them to CNN architecture. Works such as Temporal Relation Network [13], Temporal Spatial Mapping [14], VLAD3 [15] and ActionVLAD [16], utilize this approach to deploy an efficient model.

The previous works have mainly relied on a separate temporal stream to learn information available across a video. Our work tries to eliminate the need of using a temporal stream by introducing a novel approach to fusing raw RGB frames with the optical flow in an efficient way. We demonstrate in our study how our plug-in module can help reduce computation by a huge margin thus making drone-camera-based HAR more practical, fast, and coherent. Our method uses segment-based sampling [12] to include global temporal information with a minimum number of frames. We perform experiments on three diverse datasets and show that our method works better than the previous methods. Our SWTA module can also boost up the performance of existing approaches since it doesn't require knowledge about the internal details of the architecture such as activation functions, hidden layers, $etc.$ and can be easily included in any method. The approach is easy to implement thus supporting faster experimentation. Other than that, the module can also act as a teacher network (the existing network being the student network), optimize the existing network to learn temporal information [17], and then it can be removed at the time of performance evaluation.

The major contributions of this paper have been listed below:

- We introduce Sparse Temporal Sampling before the Weighted Temporal Attention (WTA) module to obtain global attention with significantly lesser computation.
- We incorporate Region Of Interest (ROI) Cropping in the WTA module to deal with the extremely small size (as shown in Fig. 1) of human subjects. This helps us to recognize human activities from the high altitude of drone camera videos.
- The proposed SWTA module can act as a plug-in module.
- We perform extensive experimental analysis on three publicly available benchmark datasets, $i.e.$, Okutama dataset [1], MOD20 dataset [3], and Drone-Action dataset [2].
- Our proposed model achieves state-of-the-art performance on these three datasets. The comparison has been shown in TABLE II, III, and IV.

Related Works

This section highlights the previous works done in the field of HAR. To discuss them, we have divided various approaches into three sub-categories below along with their shortcomings. In the end, we describe how our study deals with them.

Two-stream Networks: : Simonyan and Zisserman [18] proposed two-stream networks: spatial and temporal stream, to achieve high accuracy on action recognition. The spatial stream takes raw RGB frames as input whereas the temporal stream takes optical flow as input. The final prediction is the resultant average of both streams. Kaparthy et al. [19] proposed different fusion methods such as early fusion, late fusion, and slow fusion for video classification. Feichtenhofer et al. [10] introduce VideoLSTM which uses a spatial attention mechanism and motion-based attention mechanism. It shows promising results and demonstrates the use of learned attention in action localization. To capture long-range temporal information, Wang et al. proposed Temporal Segment Networks (TSN) [12] that deal with capturing information available across long-range videos using segment-based sampling techniques. SlowFast network [20] was proposed, which takes slow frame rate inputs and fast frame rate inputs separately into two streams. The former captures the semantic information while

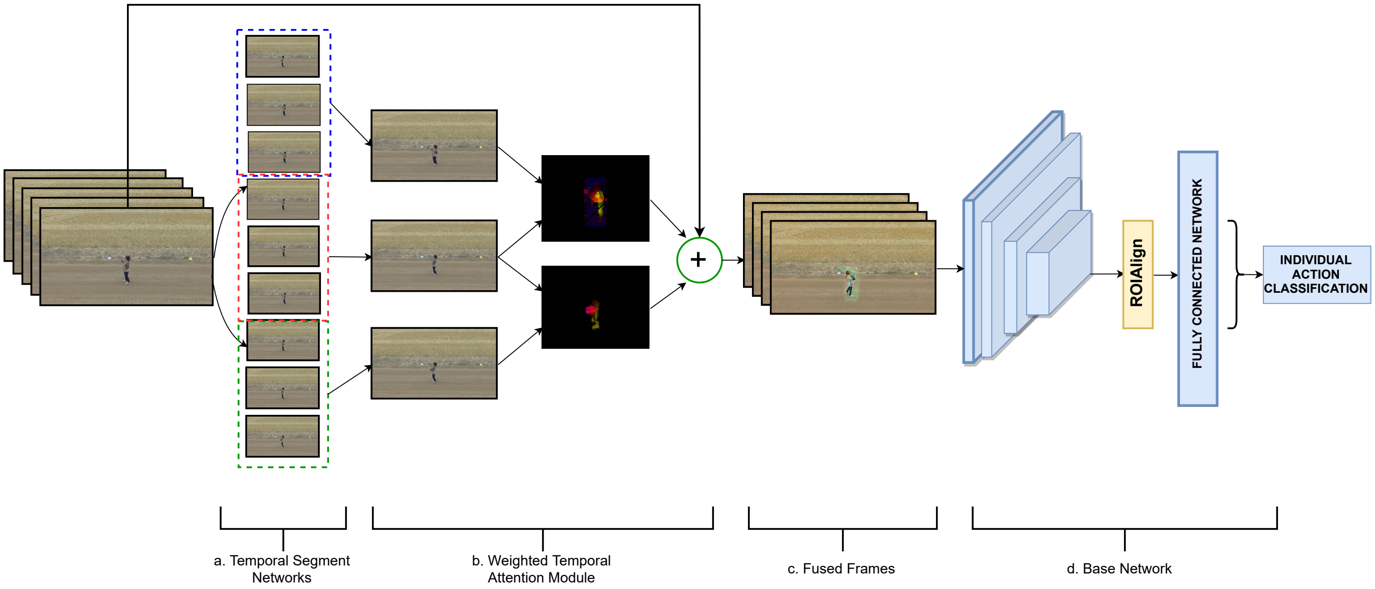
ROIAlign
FULLY CONNECTED NETWORK
INDIVIDUAL ACTION CLASSIFICATION
a. Temporal Segment Networks
b. Weighted Temporal Attention Module
c. Fused Frames
d. Base Network

Fig. 2. Block diagram of the SWTA Network. (a). A clip from the extracted frames is sparsely sampled and segmented into equal halves. (b). Random frames are chosen out of those segments and using the Weighted Temporal Attention module, we derive optical flow feature maps. (c). These maps are fused with the original frames. (d). The fused frames are fed into the base network which provides us with the activity classification results.

the latter operates at a high temporal resolution to learn the swiftly changing movements. However, a standard SlowFast network trained on benchmark datasets takes 10 days to get completed on average. Approaches such as MARS [21] and D3D [22] use a knowledge distillation mechanism for combining a two-stream network into a single-stream network. This reduces the computational cost to a good extent but some amount of information is lost.

Multi-stream Networks: : In the RGB-D domain, Depth2Action [23] uses off-the-shelf depth estimators to extract depth information from videos and use it for action recognition. In [24], the authors collect a multi-modal dataset combining data from 360° camera stream, LiDAR stream, and RGB-D stream captured by Depth cameras and achieve high accuracy but their dataset is limited to indoor activities. In [25], the authors argue the importance of a representation derived from the human pose. They crop RGB image patches and flow patches for the right hand, left hand, upper body, full-body, and full image, based upon the joint estimations. Using these patches they use separate CNN architectures to extract their appearance and motion features which are further aggregated to provide video-level descriptors.

3-Dimensional CNNs*:* In an attempt to improve shallow neural network's [26] performance, C3D [27] with a deeper 3D network that used a simple temporal pooling technique for action recognition was introduced. However, it was unable to perform well on benchmark datasets as deep 3D CNNs are hard to optimize. I3D introduced in [28] puts to use inflated ImageNet weights of 2D network to their respective counterparts in the 3D network as proposed in [29].To employ the benefits of pre-trained models on large datasets as already done for 2D-CNNs, Chen *et al.* [30] created ResNet3D by altering 2D into 3D filters. P3D network [31] factorizes 3D kernels into 2D and 1D kernels to better cope with the running complexity. Methods such as S3D [32] utilize the approach in [31] after replacing the bottom half of the 3D kernels with 2D kernels to generate a ”top-heavy” network. The remaining 3D convolutions are factorized by P3D to further minimize the size of the model and reduce the time complexity. Temporal Shift Module (TSM) [8] is another efficient approach wherein part of the channels are shifted along the temporal dimension which promptly helps in information exchange among neighboring frames.

Limitations: : The existing state-of-the-art on Drone-Action dataset [25] separately uses a pose-stream that heavily relies on the correct joint estimations. The state-of-the-art on MOD20 dataset [3] uses a two-stream approach and depends on motion-CNN for their accuracy. The state-of-the-art on Okutama-Action dataset [33] uses features computed by 3D convolution neural networks plus a new set of features computed by Binary Volume Comparison (BVC) layer, which comprises three parts: a 3D-Conv layer with 12 non-trainable ($i.e.$, fixed) filters, a non-linear function and a set of learnable weights. Features from both the streams: 3D CNNs and BVC layer are concatenated and passed to Capsule Network for final activity prediction. Our approach yields competitive results on the Drone-Action dataset even without using pose-stream separately. On the MOD20 dataset, we surpass the previous state-of-the-art without including a separate temporal stream as the SWTA module efficiently learns global temporal information using weighted temporal attention. Similarly, on the Okutama-Action dataset, we surpass the previous state-of-the-art and our model is comparatively computationally cheap as we do not need a different stream of 3D CNNs to deal with temporal information.

Proposed Methodology

In this section, we give a detailed description of the individual components used in our model architecture. Then we explain how we have compiled those components to perform effective drone-camera-based human action recognition. Our model takes a short snippet of video frames that undergoes necessary preprocessing. Then we select K frames using sparse temporal sampling. We use OpenCV to obtain optical flow for K frames and then fuse optical flow with the RGB frames using the Weighted Temporal Attention (WTA) module. After that, we extract features from fused frames (RGB and Optical Flow) using our backbone network, *i.e.*, Inception-v3 with Batch-Normalization. This is followed by the ROIAligning module which is used to concatenate features corresponding to our subjects. These features are further flattened and passed to fully connected layers which are followed by max-pooling resulting in individual-level action classification.

We provide complete details regarding data preprocessing techniques that we experimented with on all three datasets: MOD20, Okutama, and Drone-Action dataset. Each step in the proposed methodology has been described below. Section III.A talks about Data Preprocessing techniques. Section III.B describes Temporal Segment Network [34]. Details about our novel Sparse Temporal Sampling-based Weighted Temporal Attention module are given in Section III.C and Section III.D discusses the Backbone Network, ROIAlign module, and certain modifications that we made on top of it.

Data Preprocessing

We have used three different datasets to verify the performance of our approach. Each dataset differs from the others in terms of actions, number of frames, frame rate, resolution of cameras used to record the videos, environment, camera motion, and even annotations. While the MOD20 dataset only comes with ground truth action labels, the Okutama dataset provides ground truth bounding boxes as well. Drone-Action dataset goes one step ahead and provides ground truth pose annotations along with bounding boxes and frames. We have predicted bounding boxes separately for MOD20 as the intermediate layers of our novel WTA module rely on bounding box coordinates as discussed in Section III.C. We utilize the joint annotations provided with the Drone-Action dataset in a separate pose stream for a fair comparison with previous state-of-the-art methods. Nonetheless, our model achieves competitive results even without a pose stream.

We use data augmentation techniques, such as random cropping and horizontal flip to prevent our model from adversarial examples. We resize our images while maintaining the aspect ratio, for which the details are discussed in Section IV.B. All the images are rescaled before being fed to the model, and bounding box coordinates are normalized as well.

$$Final\,Image = \left(\frac{Original\,Image}{255.0} - 0.5\right) \times 2.0 \tag{1}$$

Temporal Segment Network

As discussed in [this paper], dense sampling causes 2D ConvNets to overfit the training dataset as the frames are densely recorded in a video, and the content changes relatively slowly resulting in limited temporal information. Instead of using all the frames, we adopt a computationally efficient method [34] which helps to speed up the training process. We use sparse and global sampling techniques constructed using segment-based sampling to extract information across the entire snippet with a very less number of frames. The segment count is fixed thus guaranteeing that the computational cost will be constant throughout all the snippets.

Given a short snippet S whose shape is (T, C, H, W) where T is no. of frames in a snippet, C is a channel (eg: 3 for RGB), H and W are height and width respectively.

$$S = \{f_1, \ldots, f_T\}; \quad where \;\; f_i \in \mathbb{R}^{(CxHxW)} \;\; \forall i \in [1, T] \tag{2}$$

where f$_i$ denotes i_{th} frame in the snippet.

We divide it into K segments $\{SGM_1, \ldots SGM_K\}$ of equal durations and select one frame from each segment based on random sampling, as demonstrated in Fig. 3.

$$SGM_i = \{f_{((i-1)\cdot k)+1}, ..., f_{(i\cdot k)}\}; \qquad \forall (i \cdot k) \leq T, \, i \leq K \tag{3}$$

We randomly select one frame F from each segment which implies:

$$F_i \in SGM_i \quad \forall i \in [1, K] \tag{4}$$

The use of a sparse sampling strategy reduces computational complexity dramatically and prevents overfitting which would have otherwise occurred due to a limited number of frames. Thus, it provides us with an efficient video-level framework that is capable of capturing long-range temporal structures.

Weighted Temporal Attention

We have developed our novel Weighted Temporal Attention (WTA) module inspired by extensive research on applications of optical flow in the past years. This module takes sparsely sampled frames from Temporal Segment Network as input whose shape = (K, C, H, W). It captures the motion of specific parts (Fig. 1) of input relevant to the task in hand and the resultant feature maps automatically lead to a sizable improvement in accuracy over baseline architectures.

Let $O(x, y)$ denote optical flow between x and y, x and y being two frames:

$$OF_i = O(F_i, F_{i+1}) \tag{5}$$

Let x_F denote the weighted temporal attention of snippet S:

$$x_F = \sum_{i=1}^{K-1} W_i \cdot OF_i \tag{6}$$

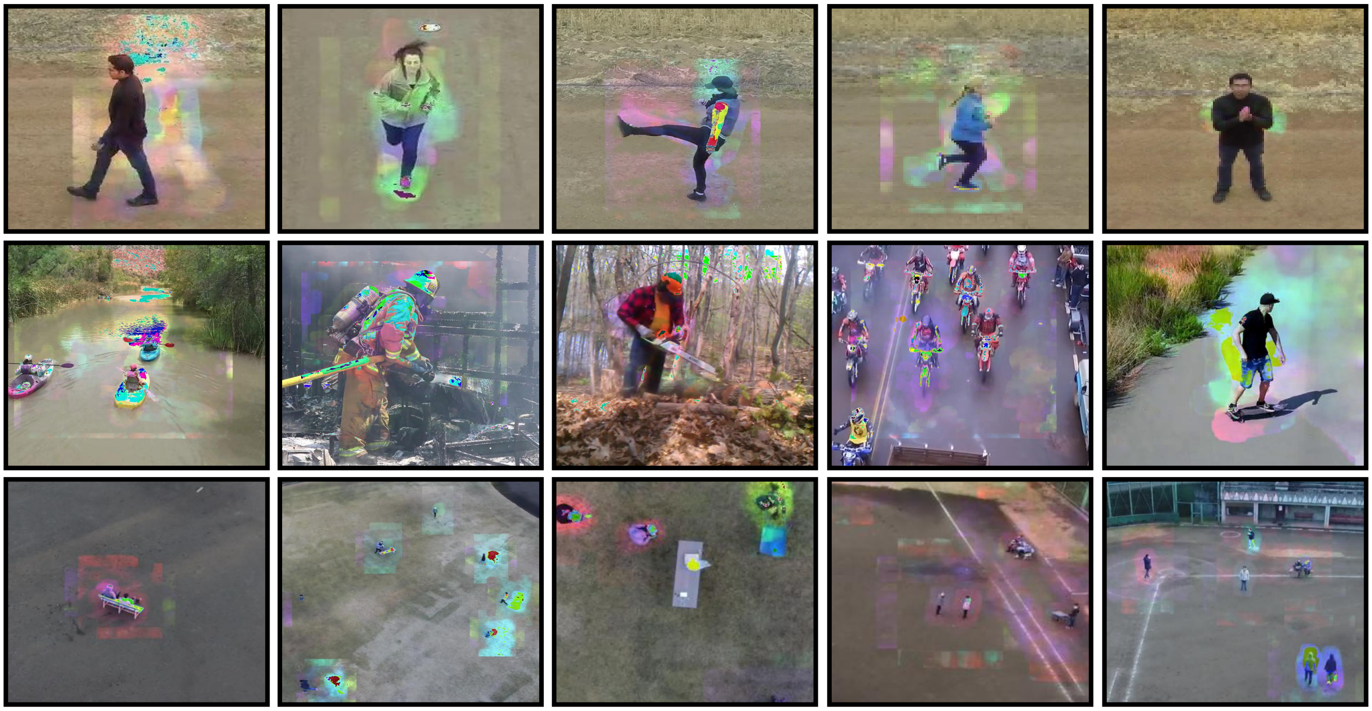

Fig. 3. The given set of images depicts the action of weighted temporal attention of some selected frames. The first row contains examples from the Drone-Action dataset, of classes namely 'walking-sideways', 'running', 'kicking', 'running-sideways', and 'clapping' respectively. The second row shows examples from the MOD20 dataset of classes 'kayaking', 'fire-fighting', 'chainsawing-trees', 'motorbiking', and 'skateboarding' respectively. The third row shows examples from the Okutama dataset of various scenes where people can be seen sitting on a bench and interacting (first column); walking, standing (second column); carrying objects and interacting (third column); and standing (fourth and fifth column).

We perform element-wise multiplication of x$_F$ with all the T frames in our snippet S such that:

$$x_t := x_F \odot x_t; \quad \forall t \in [1, T] \tag{7}$$

In our work, we take a clip of T=15 frames. This snippet is divided into 3 segments of 5 frames each, and a random frame is sampled from each of these segments. Thus, we now get 3 frames - (F_1, F_2, F_3) from which optical flow is calculated: $OF_1 = O(F_1, F_2), OF_2 = O(F_2, F_3)$. Each of these optical flow values is multiplied with W_i as 0.033 and summation takes place (x_F). This value is then multiplied by each of the 15 frames.

WTA module is a simple yet powerful module to incorporate optical flow in action recognition. Our novel formulation is simple to implement and can be seen as an extension of "weighted average pooling". We add bounding box coordinates in the intermediate layers which encourage our novel module to look for relevant actions. We finally get an output of the shape - $(1, C, H, W)$, which are further fused with RGB frames as shown in Fig. 2. This approach helps us in reasoning for the long-term temporal relations even by looking at a single frame. We finally combine appearances from static RGB images and motion inferred by the module to perform action recognition [35]. Examples of fused frames are given in Fig. 3.

Backbone Network

There has been a significant amount of research lately in the field of video analysis using deep learning and interestingly, the advancements made in image classification methods have played a huge role in its progress. Most of the previous work including state-of-the-art algorithms incorporate two-stream ConvNets in their architecture to deal with appearance and motion separately. But the question is can we classify activities using a single stream of CNNs? In our approach, we have tried to deal with this issue of extra computation by merging optical flow features with the static images and used a single-stream of Inception-v3 [36] to predict individual-level actions.

Pre-training the backbone on large-scale image recognition datasets, such as ImageNet [37], has turned out to be an effective solution when the target dataset does not have enough training samples [18]. As spatial networks take RGB images as inputs, it is natural to exploit models trained on the ImageNet as initialization. We use Inception-v3 [36] with Batch Normalization pre-trained on ImageNet, as a backbone network, due to its balance between accuracy and efficiency.

Our model falls under the risk of overfitting due to a limited number of training samples. To prevent this, we have relied on various regularization techniques. Batch Normalization is able to deal with the problem of covariate shift by estimating the activation mean and variance within each batch to normalize these activation values. This also helps in faster convergence. Further, we add dropout layers between our last fully connected layers having a dropout ratio of 0.3 before the global pooling layer. We use Adam optimizer with weight decay parameter set to 1e^{-4} which adds L2 norm regularization. These techniques prevent the high risk of overfitting and help in the generalization of our network.

Experimental Results

The complete architecture of the CNN network used is given in Fig. 2. The same architecture is used respectively for all three datasets. The number of features, input shapes, and output feature map shapes however differ in all three cases owing to the difference in the dimension of the input images. The summary is provided in TABLE I.

TABLE I

Summary of Datasets used

Dataset Name	Classes	#Clips	Duration	FPS
Okutama (2017)	12	43	60.00 s	30.00
DroneAction (2019)	13	240	11.15 s	25.00
MOD20 (2020)	20	2324	7.40 s	29.97

Experimental Settings

Training of the model was carried out for 80 epochs for each dataset, on a system with an Intel Xeon processor, 12GB VRAM, and Nvidia Titan XP GPU. The model was compiled using the Pytorch backend. All the frames collected from the video datasets were first resized into a shape of 420×720 and normalized. Along with this, they were grouped into a batch size B=2 while taking frames T=15 at a time. The resulting data had a shape of (B, T, H, W, C) where H, W, and C denote height, width, and the number of channels of the frame respectively.

Using the Inceptionv3 backbone, the feature maps obtained were processed in the ROIAlign function, having a crop size of 5, to get our desired region of interest. Therefore, the result was flattened and fed into a fully connected (FC) layer having $M = 512$ units, followed by a dropout layer, with the dropout ratio being 0.3, and a batch normalization layer. The output of the FC block was passed to the classifier which gave us the resulting probability. The train-to-test split ratio was held constant at 80:20 for all datasets. Adam optimizer with an initial learning rate of 10^{-5}, β_1 and β_2 with a value of 0.9 and 0.999, and a weight decay of 10^{-4} was found to be the most suitable optimizer for the given task of action recognition as well as to prevent overfitting. The learning rate scheduler was utilized to decrease the learning rate by one-tenth of its value after every 30 epochs. The one-hot encoded targets and predictions were fed into Binary Cross Entropy Logits Loss owing to its satisfactory usability to process softmax outputs of the last layer of the model.

In order to understand the significance of having a separate pose stream for the Drone-Action dataset, while also comparing our model's performance with the previously obtained results, another stream of the network was added to the above model which predicted the activity label using the normalized OpenPose [38] joint coordinates as features. Two 1D CNN blocks, each followed by batch normalization and dropout layer, with a dropout ratio of 0.5 were used. The resultant features were fed into an LSTM model having 14 units, followed by a classifying layer. Average pooling was enabled to get the results from the two models. The rest of the hyperparameters remained unchanged.

Evaluation Metric

Overall, the top-1 accuracy of the various model outputs for different datasets was chosen as an evaluation metric. Officially defined as the number of correct predictions over the total number of samples, for the respective categories, we use it as almost all classes in each dataset contain equal amounts of data. Hence, this metric is suitable for the given task of action recognition and classification.

$$Accuracy = \sum_{b=1}^{B} \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{Correct\ Predictions}{Total\ Samples} \tag{8}$$

For each dataset, we get the shape of activity labels and bounding boxes as (B, T, N) where B denotes the batch size, T, is the number of frames per batch, and N is the number of objects in each frame. We calculate the accuracy by computing the number of correct outputs for each frame for all N objects. The resulting value is then averaged over all batches for each dataset to get our result.

Performance Evaluation

In this subsection, we discuss the evaluation results obtained on the ablation studies performed on Okutama, MOD20, and Drone Action datasets. TABLE II, III and IV summarize the results of the respective datasets with the mentioned backbone that is utilized.

Initially, we start our experiments using an Inception-v3 module [36] to capture the spatial features of the original images without the weighted temporal attention module. This helps us understand the critical and influential effect of the Weighted Temporal Attention module. Training the model using a basic backbone with RoiAlign, fully connected, batch-normalization, and dropout layers resulted in 61.34% accuracy for the Okutama dataset. This alone can be seen as outperforming the previous state-of-the-art values achieved using different backbones. To further improve the outcome, the proposed Weighted Temporal Attention module is added, and the backbone is fed with ”fused” frames instead of original ones, resulting in an overall accuracy of 72.76%, marking an increase of 11.42% from the basenet architecture and 25.26% from the previous state of the art. Using the optical flow backdrop, the network can specifically focus on the region where the action is taking place, and disregard the background which may contain noise.

Similarly for MOD20 dataset, consists of a diverse range of action classes, each significantly different from the other.

TABLE II
Comparison with the state-of-the-art results on the Okutama dataset. **Blue** represents the previous state-of-the-art. **Red** denotes the best results.

S.no.	Method	Backbone	Accuracy
Past Work	AARN [39] [33]	C-RPN + YOLOv3-tiny	33.75%
	Lite ECO [40] [33]	BN-Inception + 3D-Resnet-18	36.25%
	I3D(RGB) [28] [33]	3D CNN backbone	38.12%
	3DCapsNet-DR [41] [33]	3D CNN + Capsule	39.37%
	3DCapsNet-EM [41] [33]	3D CNN + Capsule	41.87%
	DroneCaps [33]	3D CNN + BVC + Capsule	**47.50%**
Ours	BaseNet	Inception-v3	61.34%
	SWTA	**Weighted Temporal Attention + Inception-v3**	**72.76%**

TABLE III
State-of-the-art results on the MOD20 dataset. **Blue** represents the previous state-of-the-art. **Red** denotes the best results.

S.no.	Method	Backbone	Accuracy
Past	KRP-FS [42] [3]	VGG-f + motion-CNN	**74.00%**
Ours	BaseNet	Inception-v3	90.03%
	SWTA	**Weighted Temporal Attention + Inception-v3**	**92.56%**

It contains complex outdoor scenarios. That being said, our basenet model was able to achieve a higher accuracy of 90.03% as compared to the previous state-of-the-art value which was 74% [3]. After integrating the Weighted Temporal Attention module, a slight increase of 2.56% was obtained.

The Drone Action dataset contained various action classes which were similar to one another. For example, jogging from the front, back, and sideways was similar to running front, back, and sideways. It was critical to exactly locate the joint positions in order to determine which action was being performed. Hence, without the pose annotations, results were obtained from our simple basenet: 62.79% and integrated Weighted Temporal Attention module with basenet: 71.79%. Training the model along with pose joints in a separate stream led us to achieve greater results than the previous state-of-the-art, marking the increase by 2.84%.

Discussion and Comparison

Our approach outperforms the previously existing methods. It successfully achieves state-of-the-art results in all three datasets, namely 72.76% on the Okutama dataset, 92.56% on the MOD20 dataset, and 71.79% on the Drone Action dataset without pose-stream whereas 78.86% with pose-stream. For the Okutama dataset specifically, our Weighted Temporal Attention module with RoiAlign leads the network to focus on the keypoints where the action is currently taking place, and ignores the background noise, as opposed to the previously used 3D CNNs in [33]. It is also computationally less expensive, being a single stream network as compared to the approaches used in the previous works for MOD20 and Okutama dataset evaluation.

TABLE IV
Comparison with the state-of-the-art results on the DroneAction dataset. **Blue** represents the previous state-of-the-art. **Red** denotes the best results.

S.no.	Method	Backbone	Accuracy
Past	HLPF [43] [2]	NTraj+ descriptors	64.36%
	PCNN [44] [2]	'VGG-f' + Action Tubes	**75.92%**
Ours	BaseNet	Inception-v3	62.79%
	SWTA	**Weighted Temporal Attention + Inception-v3**	**71.79%**
	SWTA+Pose-Stream	**Weighted Temporal Attention + Inception-v3+Pose-Stream**	**78.86%**

Jhuang et al. [43] uses the HLPF approach which focuses on temporal and spatial information but ignores the additional data of the objects or props used in performing the action. Consequently, Cheron et al. [44] P-CNN which uses the two-stream network to process RGB patches and flow patches is able to surpass HLPF results. With our simple CNN-LSTM model that is decently able to distinguish between similar classes, to get results using pose data and computationally cheap temporal segment network to process ROI cropped regions, our model is able to produce better results in a shorter amount of time.

Conclusion

In this study, we propose an SWTA network consisting of the Sparse Weighted Temporal Attention module which helps to improve the performance of our basenet by a significant margin without adding much to the computational cost. We believe that our module can optimize spatial streams in learning temporal features with low complexity by eliminating the need for a temporal stream which is common in activity recognition tasks involving deep learning. We have presented a novel approach to fusing the concept of temporal segment networks, weighted temporal attention, and convolutional neural networks to determine the activity being performed in videos collected by drone cameras. A significant increase is observed in the case of the Okutama-Action dataset which can be highly useful for drone-based activity recognition tasks at very high altitudes such as crowd analysis or video surveillance. While being less complex as compared to other approaches, our model also generalizes well on the challenging outdoor scenes depicted in MOD20, and Drone-Action datasets, achieving state-of-the-art results in the same. Deployment of such a model on an appropriate device could be increasingly beneficial. This study has the potential to solve the computational barriers that prevent the deployment of deep learning-based HAR systems on drones.

References

[1] M. Barekatain, M. Martí, H. Shih, S. Murray, K. Nakayama, Y. Matsuo, and H. Prendinger, "Okutama-action: An aerial view video dataset for concurrent human action detection," *CoRR*, vol. abs/1706.03038, 2017. [Online]. Available: http://arxiv.org/abs/1706.03038

[2] A. G. Perera, Y. W. Law, and J. Chahl, "Drone-action: An outdoor recorded drone video dataset for action recognition," *Drones*, vol. 3, no. 4, 2019. [Online]. Available: https://www.mdpi.com/2504-446X/3/4/82

[3] A. Perera, Y. Law, T. Ogunwa, and J. Chahl, "A multiviewpoint outdoor dataset for human action recognition," *IEEE Transactions on Human-Machine Systems*, vol. PP, pp. 1–9, 02 2020.

[4] N. Ikizler and D. Forsyth, "Searching video for complex activities with finite state models," 06 2007.

[5] L. Lo Presti and M. La Cascia, "3d skeleton-based human action classification," *Pattern Recogn.*, vol. 53, no. C, p. 130–147, May 2016. [Online]. Available: https://doi.org/10.1016/j.patcog.2015.11.019

[6] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," 06 2007.

[7] C. Zhang, Y. Zou, G. Chen, and L. Gan, "PAN: towards fast action recognition via learning persistence of appearance," *CoRR*, vol. abs/2008.03462, 2020. [Online]. Available: https://arxiv.org/abs/2008.03462

[8] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[9] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, "Towards good practices for very deep two-stream convnets," 2015.

[10] Z. Li, E. Gavves, M. Jain, and C. G. M. Snoek, "Videolstm convolves, attends and flows for action recognition," 2016.

[11] D. Tran, H. Wang, L. Torresani, and M. Feiszli, "Video classification with channel-separated convolutional networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[12] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks: Towards good practices for deep action recognition," 2016.

[13] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal relational reasoning in videos," 2018.

[14] X. Song, C. Lan, W. Zeng, J. Xing, X. Sun, and J. Yang, "Temporal–spatial mapping for action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, pp. 748–759, 2020.

[15] Y. Li, W. Li, V. Mahadevan, and N. Vasconcelos, "Vlad3: Encoding dynamics of deep features for action recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1951–1960.

[16] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell, "Actionvlad: Learning spatio-temporal aggregation for action classification," 2017.

[17] J. C. Stroud, D. A. Ross, C. Sun, J. Deng, and R. Sukthankar, "D3d: Distilled 3d networks for video action recognition," 2019.

[18] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," 2014.

[19] 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*, 2014, pp. 1725–1732.

[20] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6201–6210, 2019.

[21] N. Crasto, P. Weinzaepfel, K. Alahari, and C. Schmid, "Mars: Motion-augmented rgb stream for action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[22] J. C. Stroud, D. A. Ross, C. Sun, J. Deng, and R. Sukthankar, "D3D: distilled 3d networks for video action recognition," *CoRR*, vol. abs/1812.08249, 2018. [Online]. Available: http://arxiv.org/abs/1812.08249

[23] Y. Zhu and S. Newsam, "Depth2action: Exploring embedded depth for large-scale action recognition," 2016.

[24] M. Moencks, V. D. Silva, J. Roche, and A. Kondoz, "Adaptive feature processing for robust human activity recognition on a novel multi-modal dataset," 2019.

[25] G. Chéron, I. Laptev, and C. Schmid, "P-cnn: Pose-based cnn features for action recognition," 2015.

[26] 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*, vol. 35, no. 1, pp. 221–231, 2013.

[27] 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 (ICCV)*, December 2015.

[28] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4724–4733.

[29] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, "Towards good practices for very deep two-stream convnets," *CoRR*, vol. abs/1507.02159, 2015. [Online]. Available: http://arxiv.org/abs/1507.02159

[30] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "Multi-fiber networks for video recognition," *CoRR*, vol. abs/1807.11195, 2018. [Online]. Available: http://arxiv.org/abs/1807.11195

[31] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," *CoRR*, vol. abs/1711.10305, 2017. [Online]. Available: http://arxiv.org/abs/1711.10305

[32] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[33] A. M. Algamdi, V. S. Silva, and C.-T. Li, "Dronecaps : recognition of human actions in drone videos using capsule networks with binary volume comparisons," in *27th IEEE International Conference on Image Processing*. IEEE, 2020. [Online]. Available: http://wrap.warwick.ac.uk/141611/

[34] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks: Towards good practices for deep action recognition," *CoRR*, vol. abs/1608.00859, 2016. [Online]. Available: http://arxiv.org/abs/1608.00859

[35] R. Gao, B. Xiong, and K. Grauman, "Im2flow: Motion hallucination from static images for action recognition," *CoRR*, vol. abs/1712.04109, 2017. [Online]. Available: http://arxiv.org/abs/1712.04109

[36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[38] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *CoRR*, vol. abs/1812.08008, 2018. [Online]. Available: http://arxiv.org/abs/1812.08008

[39] F. Yang, S. Sakti, Y. Wu, and S. Nakamura, "A framework for knowing who is doing what in aerial surveillance videos," *IEEE Access*, vol. PP, pp. 1–1, 07 2019.

[40] M. Zolfaghari, K. Singh, and T. Brox, "Eco: Efficient convolutional network for online video understanding," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[41] P. ZHang, P. Wei, and S. Han, "CapsNets algorithm," *Journal of Physics: Conference Series*, vol. 1544, p. 012030, may 2020. [Online]. Available: https://doi.org/10.1088/1742-6596/1544/1/012030

[42] A. Cherian, S. Sra, S. Gould, and R. Hartley, "Non-linear temporal subspace representations for activity recognition," *CoRR*, vol. abs/1803.11064, 2018. [Online]. Available: http://arxiv.org/abs/1803.11064

[43] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. Black, "Towards understanding action recognition," 12 2013, pp. 3192–3199.

[44] G. Chéron, I. Laptev, and C. Schmid, "P-cnn: Pose-based cnn features for action recognition," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3218–3226.