

# Streamlining Forest Wildfire Surveillance: AI-Enhanced UAVs Utilizing the FLAME Aerial Video Dataset for Lightweight and Efficient Monitoring

Lemeng Zhao, Junjie Hu, Jianchao Bi, Yanbing Bai\*, *Member, IEEE*,  
Erick Mas, Shunichi Koshimura\*

**Abstract**—In recent years, unmanned aerial vehicles (UAVs) have played an increasingly crucial role in supporting disaster emergency response efforts by analyzing aerial images. While current deep-learning models focus on improving accuracy, they often overlook the limited computing resources of UAVs. This study recognizes the imperative for real-time data processing in disaster response scenarios and introduces a lightweight and efficient approach for aerial video understanding. Our methodology identifies redundant portions within the video through policy networks and eliminates this excess information using frame compression techniques. Additionally, we introduced the concept of a *station point*, which leverages future information in the sequential policy network, thereby enhancing accuracy. To validate our method, we employed the wildfire FLAME dataset. Compared to the baseline, our approach reduces computation costs by more than 10 times while improving accuracy by 3%. Moreover, our method can intelligently select salient frames from the video, refining the dataset. This feature enables sophisticated models to be effectively trained on a smaller dataset, significantly reducing the time spent during the training process.

## I. INTRODUCTION

Forest fires, posing a significant threat to human life and the ecological environment, have garnered increased attention in recent years [1]. In particular, UAV-based platforms are increasingly used for post-disaster monitoring and response due to their ability to acquire real-time and efficient ground information [2], [3]. This approach enhances data acquisition speed and reduces the risks associated with manual inspection of forest fires, ensuring the safety of lives.

This work was jointly supported by JSPS KAKENHI (Grants-in-Aid for Scientific Research, 21H05001); the Major Project of the MOE (China) National Key Research Bases for Humanities and Social Sciences (22JJD910003); Beijing Golden Bridge Project seed fund. Project No.ZZ21021; This work was jointly supported by the Co-creation Center for Disaster Resilience, Tohoku University, and the Public Computing Cloud, Renmin University of China.

Yanbing Bai is an assistant professor at the Center for Applied Statistics and School of Statistics, Renmin University of China, Beijing, 100872, China, e-mail: ybbai@ruc.edu.cn. Corresponding author

Lemeng Zhao is a student at the Center for Applied Statistics and School of Statistics, Renmin University of China, Beijing, 100872, China, e-mail: zhaolemeng@ruc.edu.cn

Junjie Hu is a researcher at The Chinese University of Hong Kong, Shenzhen, 518129, China, e-mail: hujunjie@cuhk.edu.cn

Jianchao Bi is a student at Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, 100872, China, e-mail: bijianchao@ruc.edu.cn

Erick Mas is an Associate Professor with the International Research Institute of Disaster Science (IRIDeS), Tohoku University, Sendai, 980-8572, Japan, e-mail: mas@tohoku.ac.jp

Shunichi Koshimura is a Professor with the International Research Institute of Disaster Science (IRIDeS), Tohoku University, Sendai, 980-8572, Japan, e-mail: koshimura@tohoku.ac.jp. Corresponding author

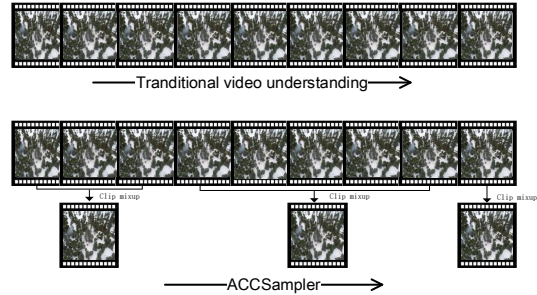


Fig. 1. Example of ACCSampler for frame consolidation.

Recently, deep learning algorithms have been at the forefront of detecting forest fires using visual cues like flame [4] or smoke [5]–[7], drawing extensive attention within the research community, in particular, models for video stream data understanding [1]. However, real-time processing of video stream data in disaster response scenarios entails significant computing costs [8], so recent advancements have focused on minimizing computational expense by designing lightweight architectures [9], [10] or selecting representative frames [11]–[13]. Building on this inspiration and considering the surplus frame information in raw videos [14], our work aims to create an elegant and efficient model for understanding aerial videos in the context of wildfires.

This paper introduces AccSampler, an innovative video understanding model called Adaptive Clip-aware Compression and Frame Sampling Network. Unlike conventional methods that analyze videos frame by frame, AccSampler employs a unique approach by consolidating multiple clips into a single frame, as depicted in Fig. 1. Our model strategically decides when to merge clips, leveraging video features to make informed decisions. We efficiently reduce computational workload without compromising accuracy by dedicating computational resources to frames rich in information and disregarding less relevant clips. AccSampler not only achieves superior accuracy but also substantially minimizes computational efforts.

However, aerial videos are typically captured in challenging environments, where factors such as blurred frames and abrupt changes, like unpredictable camera vibrations due to air currents, are more prevalent. To address this, we introduce the station points to provide future features, aiding the policy network in making sensible and stable decisions when processing aerial videos. The second stage involves the Frame Selection Module. The decisions made by the policy network reflect the frames' importance. We developed

a straightforward static frame selection module based on the trained Efficient Video Understanding Module. This module scores each video frame and selects high-scoring frames to reconstruct a more concise video. This process involves no intricate pre-processing or post-processing. It is akin to editing a lengthy video to obtain a shorter, more valuable version. The resulting video preserves essential temporal and spatial information crucial for disaster identification to the greatest extent possible.

The key contributions of this study are outlined below:

- 1) We have designed a new deep-learning model that is both lightweight and efficient, specifically tailored for processing and interpreting aerial videos.
- 2) We have employed the station point approach, providing additional spatio-temporal features. This augmentation enhances the model's efficiency, robustness, and overall performance.
- 3) Through some ablation experiments, we not only confirmed the robustness of our model but demonstrated its applicability for monitoring other physical processes.

## II. RELATED WORK

An efficient video understanding model, based on representative frames, offers an effective solution to the challenge of substantial computational costs [15]. There are two prominent techniques for frame selection: serial video frame sampling and parallel video frame sampling. In sequential sampling, frames are read in temporal order, and a policy is employed to determine the subsequent frame to be read. AdaFrame [16] utilizes Memory-augmented LSTM to retain temporal and spatial information and employs hidden layer states to identify the next frame for reading. VideoIQ [17] reduces computational overhead by allocating different bits or selecting input resolution for each frame. On the other hand, parallel sampling processes frames and video clips independently and consolidates the outcomes. SCSampler [18] estimates the score of each fixed-length clip and synthesizes predictions. OCSampler [15] employs a straightforward single-step reinforcement learning optimization to directly aggregate a more comprehensive set of features for video-level modeling. Furthermore, innovative techniques have emerged in this domain. AdaFuse [19] dynamically fuses channels from current and past feature maps, enhancing temporal modeling.

The process of frame selection is rooted in the realm of efficient video understanding. For instance, AdaFrame [16], Frame Glimpse [20], and SCSampler [18] utilize frame sampling techniques coupled with specific policy networks. These approaches discern the most valuable frames and video clips from a sequence, subsequently employing these clips in cut videos for various video recognition tasks. SMART [21] employs a single-frame selector to compute the frame's spatial classification value. It then utilizes a global selector to derive temporal and spatial category values from frame pairs. Ultimately, SMART combines the evaluations of each frame's two values within the video, aiding in frame selection. On the other hand, MGSampler [22] leverages two

types of motion representation (motion-sensitive and motion-uniform) to distinguish motion-salient frames from the background efficiently. Using a motion-uniform sampling strategy based on cumulative motion distribution, the MGSampler ensures that the selected frames evenly cover all essential segments with high motion salience.

## III. PROPOSED METHOD

AccSampler aims to achieve efficient video understanding through clip compression, which allows our model to identify insignificant clips and consolidate them into a single frame. Moreover, leveraging the features obtained from clip compression, AccSampler can assess the significance of each frame and select the most informative frames to achieve dataset distillation.

### A. Network Architecture

1) *Overview*: Fig. 2 illustrates the structure of AccSampler. Considering videos  $v$  comprising a sequence of frames  $X$ , denoted as  $X = \{x_t\}_{t=1}^N$ , where  $N$  represents the total number of frames in the video, and  $t$  denotes the current time step. At  $t = 1$ , AccSampler uniformly selects multiple station points from the original input and feeds them into a 2D-CNN to extract station point features, denoted as  $\{s_m\}$ . These station points provide future information for the policy network  $\pi$ ; at  $t = i$ ,  $\pi$  receives the concatenation of clip-level features from the feature extraction network  $f_s$  and the corresponding station point features. Subsequently,  $\pi$  evaluates the significance of the following frames and determines how many frames will be compressed at  $t = i + 1$ . After processing the entire video, the classifier  $f_c$  classifies the video. Frame selection is static and relies on the well-trained  $\pi$ . This policy network effectively assesses the importance of subsequent frames. Consequently, based on the output of  $\pi$ , AccSampler calculates a score for each frame.

2) *Clip Mixup*: Mixup [23] is a data augmentation technique that involves scaling two samples and their corresponding labels from the same batch to create a new sample. The mixup formula is as follows:

$$x' = \lambda x_i + (1 - \lambda)x_j \quad (1)$$

$$y' = \lambda y_i + (1 - \lambda)y_j \quad (2)$$

Here,  $x_i$  and  $x_j$  are raw data inputs, and  $y_i$  and  $y_j$  represent their labels.  $\lambda$  is the scale parameter, and  $\lambda$  follows a  $Beta(\alpha, \alpha)$  distribution.

We employ clip mixup to consolidate several frames into one to optimize computational efficiency while preserving crucial video information. Since video classification does not demand frame-level labels, clip mixup employs only Eq. (1). Specifically, clip mixup merges the initial and final frames of input clips longer than one frame into a single frame  $x_i^{mixup}$ , representing original clips since frames within a short clip tend to be quite similar.

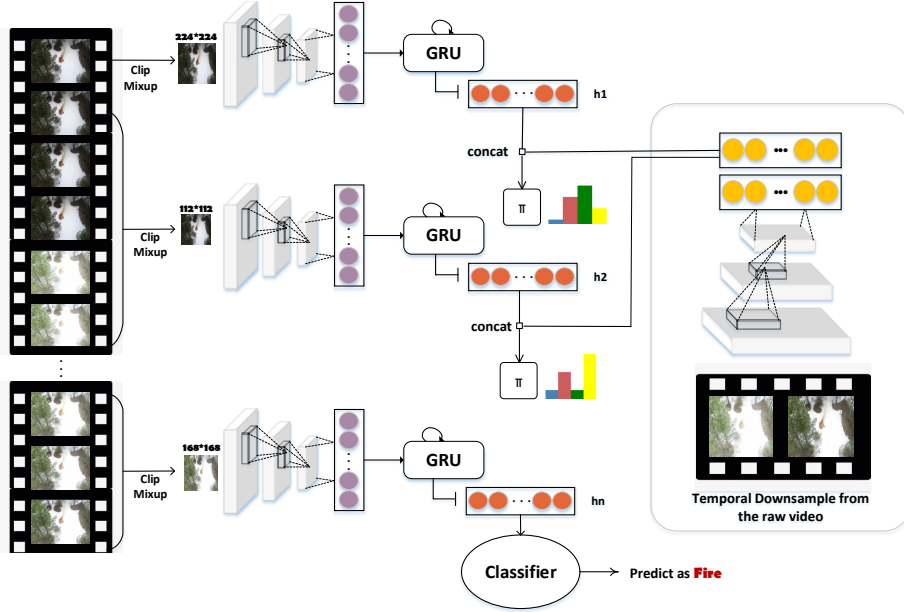


Fig. 2. The architecture of AccSampler. Frame selection is static and based on the distribution produced by  $\pi$

3) *Feature Extraction Network*: The feature extraction network  $f_s$  combines a lightweight CNN and an RNN. The output of  $f_s$  is the hidden state of the RNN. Therefore, the video features up to time  $t = i$  can be expressed as:

$$h_i = f_s(x_i^{mixup}) \quad (3)$$

4) *Policy Network*: The policy network  $\pi$  determines the number of frames to be fused at the next time step by selecting action  $k_i$  from the discrete action space  $A$ . Upon receiving the hidden state from  $f_s$  at  $t = i$ ,  $\pi$  chooses the closest future station point feature from  $\{s_m\}$ . Subsequently, the hidden state and the station point feature are concatenated and input into  $\pi$ , producing the predicted distribution  $a_i^p$  over the action space  $A$ . This process can be represented as:

$$a_i^p = \text{softmax}(\pi([h_i : s_m^{\text{nearest}}])) \quad (4)$$

Here, we define  $A$  as  $\{1, 3, 5, 7\}$ . If  $\pi$  makes the decision  $k_i$  at  $t = i$ , where  $k_i \in A$  means that at  $t = i+1$ , the  $k_i$  frames will be combined into one frame using clip mixup. The smaller the  $k_i$ , the more crucial the clip is. Additionally, frames of varying importance will be resized to different resolutions. Consequently, we define  $R = \{224, 168, 112, 84\}$ , corresponding to the resolution for each action in the action space  $A$ , which means  $\pi$  will determine whether detailed features are needed. If  $\pi$  assesses that the current features are sufficient, it may fuse five or seven frames with a lower resolution, while  $\pi$  would compress three or one with a higher resolution when detailed features are needed. We designed the structure of  $\pi$  as a GroupNorm followed by a single fully connected layer, where GroupNorm is conducive to expediting convergence.

5) *Frame Selection*:  $\pi$  is qualified to assess the importance of frames, as reflected in the decisions it makes. The more frames the policy network decides to fuse, the less

crucial they are. Consequently, we introduce the preference score  $S$  to assess the significance of each frame. According to Eq. (4), when  $t = i$ , the probability distribution of the policy network's prediction for the input in action space  $A$  is  $p_i = \{a_{ij}^p\}$ ,  $1 \leq j \leq \|A\|$ . Given the clip to be fused at  $t = i + 1$  is  $clip_{i+1}$ , we define the fraction  $S_{i+1}$  of  $clip_{i+1}$  as:

$$S_{i+1} = \sum_{j=1}^{\|A\|} a_{ij}^c / A_j \quad (5)$$

where  $c \in \{p, gm, gs\}$ . To refine the preference score  $S$  down to the frame level, the score of the middle frame is  $S_i$ , with a gradual 10% decay of the score towards the two sides.

6) *Gumbel Softmax Trick*: Noticeably, the action space  $A$  is discrete, making optimizing it with gradient backpropagation challenging. In such cases, reinforcement learning is typically utilized to train the policy network. However, reinforcement learning often suffers from slow convergence in various applications [24]. To address this issue, we employ the Gumbel Softmax Trick [25], which optimizes the policy network through gradient backpropagation. Gumbel Softmax is an effective re-parameterization technique that replaces previously non-differentiable probability distributions with differentiable ones, introducing randomness during training and encouraging the model to explore more. To illustrate, the policy network will determine the action through Gumbel Max (6) instead of an argmax [25]:

$$a_i^{gm} = \text{argmax}(\log(a_i^p) + G_i) \quad (6)$$

Here,  $G_i = -\log(-\log(U_i))$ , where  $U_i \sim \text{Uniform}(0, 1)$ . About backpropagation, a differentiable Gumbel-Softmax will be used to approximate argmax to calculate gradients:

$$a_i^{gs} = \text{softmax}((\log(a_i^p) + G_i)/\tau) \quad (7)$$

Here,  $\tau$  is a hyperparameter representing the softmax temperature. When  $\tau > 0$ , the labels become relatively smooth, while they degrade to one-hot labels when  $\tau = 0$ . As per prior research [12], [26], we initialize  $\tau$  to 5 and anneal it to 0 as the number of epochs increases.

### B. Loss Function

We designed three loss functions to train AccSampler. The first one is a cross-entropy loss, which calculates the classification loss  $L_c$  between the prediction and the ground truth:

$$L_c = \mathbb{E}[-Y \log(f_c(h_{final}))] \quad (8)$$

where  $h_{final}$  represents the final hidden states. Additionally, to prevent the policy network from converging to trivial solutions where certain actions are ignored, we introduce the balance loss  $L_b$ :

$$L_b = \sum_{k \in A} \mathbb{E} \left[ \frac{1}{T} \sum_{t=1}^T \mathbb{I}(k_t = k) - \frac{1}{\|A\|} \right] \quad (9)$$

where  $k_t \in \{1, 3, 5, 7\}$  and  $T$  represents the total time steps. Lastly, to minimize computation and reach a balance between accuracy and computation, we employ the GFLOPs loss  $L_g$ , defined as follows:

$$L_g = \frac{1}{T} \sum_{t=1}^T GFLOPs(f_s(x_i^{mixup})) \quad (10)$$

Here, we calculate the GFLOPs from the inference of  $f_s$ , so the  $L_g$  of the input  $V$  represents the average GFLOPs. In summary, the total loss  $L$  can be defined as:

$$L = L_c + \beta L_b + \gamma L_g \quad (11)$$

where  $\gamma$  and  $\beta$  are hyperparameters that balance the losses. In the experiment,  $\beta$  is set to 0.3 and  $\gamma$  is set to 0.1 referenced from VideoIQ [17].

## IV. EXPERIMENTS

### A. Experiment Setup

1) *FLAME Dataset*: The FLAME dataset [27] is a publicly available collection of fire images and videos captured by drones. We utilized the seventh and eighth repositories of this dataset for image classification. These repositories contain 39,375 images for the training/validation set and 8,617 images for the test set, all with a resolution of  $254 \times 254$ . Notably, we utilized this dataset uniquely for the video classification task. In this context, we defined video samples as sequences of 64 consecutive images extracted from the same video. If any 64 frames were labeled as containing fire, the entire video sample was marked as having fire. Consequently, we curated a dataset comprising 615 clips for training and 134 clips for testing. Among these, 483 samples were fire-related, while 266 samples had no fire, maintaining a ratio of approximately 2:1.

2) *HMDB51 Dataset*: The HMDB51 dataset is an open-domain video recognition dataset, encompassing many genuine videos derived from diverse platforms [28]. Comprising 6,766 video segments, it covers 51 action categories, ensuring each category has at least 101 clips. This paper uses HMDB51 (split 1) as a supplementary dataset to verify the effectiveness of AccSampler’s inefficient video recognition.

3) *Implementation Details*: AccSampler sets the number of station points to 2, and the hyperparameter  $\alpha$  for clip mixup is 0.3. To optimize computational efficiency, we utilize MobileNet-v2 to construct  $f_s$  (the CNN for feature extraction of station points aligns with  $f_s$ ). The GRU’s hidden layer dimension is 512, and there is a single layer. Our training approach consists of multiple stages. In the first stage, we employ frame-level labels and fine-tune MobileNet-v2, pre-trained on ImageNet, over 100 epochs. The initial learning rate is set to 0.01 and is reduced by a factor of 10 at epochs 50, 70, and 90. In the second stage, we freeze MobileNet-v2 and train GRU using clip-level labels for 20 epochs, starting with an initial learning rate of 1.45e-5. In the third stage, MobileNet-v2 and GRU remain frozen, and we train the policy network with an initial learning rate of 0.01. We employ  $a_i^{gm}$  to calculate the preference score for frame selection. We then train ResNet18-TSM on the selected frames to evaluate the frame selection’s performance. We train ResNet18-TSM for 10 epochs, initiating with a learning rate of 0.001 and a weight decay of 3e-6.

4) *Evaluation Metrics*: We consistently employ accuracy as the metric for the video classification task. Additionally, computational cost is measured in giga floating-point operations per second (GFLOPs), which can be used to calculate the complexity of the model.

### B. Main Result

1) *Video Classification*: We use  $f_s$  and  $\pi$  to perform video classification, comparing them with GRU (i.e.,  $f_s$  only). The results are outlined in Table I.

TABLE I  
VIDEO CLASSIFICATION RESULTS ON THE FLAME DATASET.

Model	Test acc	Training acc	GFLOPs/v <sup>1</sup>	GFLOPs/f <sup>2</sup>
GRU	79.10	<b>98.61</b>	26.28	0.411
<b>GRU+policy</b>	<b>82.84</b>	97.40	<b>2.79</b>	<b>0.044</b>

<sup>1</sup> GFLOPs/v refers to the average GFLOPs on the video level

<sup>2</sup> GFLOPs/f refers to the average GFLOPs on the frame level

Notably, our model substantially enhances test accuracy. At the same time, the computational load (measured in GFLOPs) is approximately one-tenth that of GRU, meaning the policy network learns to pay closer attention to informative frames.

2) *Frame Selection*: The distilled dataset is used to train a sophisticated classification model TSM [29] to evaluate the performance of frame selection. The results are given in Table II.

It can be seen that AccSampler demonstrates superior effectiveness, maintaining classification accuracy even when using just 8 frames, surpassing the results obtained from

TABLE II

TEST ACCURACY(%) OF TSM TRAINED ON THE DIFFERENT NUMBER OF FRAMES SELECTED BY FOUR METHODS ON FLAME DATASET

	Uniform [15]	Random [15]	SMART <sup>1</sup> [21]	AccSampler
8	87.3	85.1	84.3	<b>87.3</b>
20	88.1	88.1	88.1	<b>91.0</b>
28	84.3	86.6	88.8	<b>91.0</b>
All(64)		88.1		

<sup>1</sup> Our implementation because of no official code

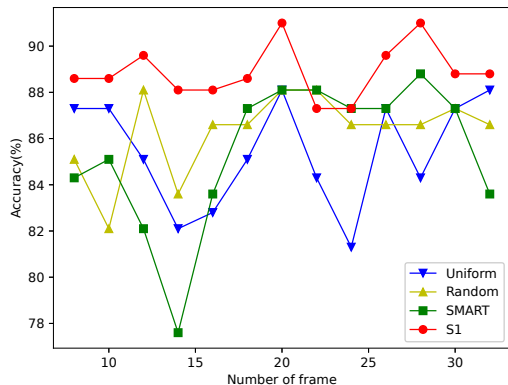


Fig. 3. Comparison of frame selection methods at various frame rates.

analyzing the full video. AccSampler significantly outperforms the baseline (Uniform and Random) and SMART results when considering an equivalent number of frames. This outcome implies that AccSampler adeptly selects more representative frames, facilitating a more efficient and accurate classification of fire videos.

Further tests were conducted over a broader range of frames to investigate whether the observed results were influenced by data randomness due to the dataset’s limited size, as shown in Fig. 3. The findings reveal that AccSampler consistently outperforms the other three frame selection methods.

However, it is noteworthy that SMART did not exhibit a significant advantage over other models. This lack of distinction might be attributed to FLAME’s small size and homogeneity, causing complex models like SMART to suffer severe overfitting during training.

3) *Classification Results on the HMDB51 Dataset:* To verify the generalization of AccSampler, we further extend the experiment in Table. I on the HMDB51 dataset. The results are shown in the Table. III.

TABLE III

VIDEO CLASSIFICATION RESULTS ON THE HMDB51 DATASET

Model	Test acc	Training acc	GFLOPs/v	GFLOPs/f
GRU	<b>40.72</b>	<b>95.78</b>	30.19	0.32
<b>GRU+policy</b>	38.50	92.34	<b>1.29</b>	<b>0.013</b>

Although accuracy declines modestly (about 2%), the AccSampler’s GFLOPs decrease disproportionately by a factor of 23, further demonstrating the effectiveness of our method.

## V. DISCUSSION

1) *Preference Score S:* Apart from the calculation utilizing  $a_i^{gm}$  in Eq. (5), we also explored the computation of S using  $a_i^p$  in Eq. (4) and  $a_i^{gs}$  in Eq. (7). The scores derived from these three methods are denoted as S1, S2, and S3, respectively. We conducted experiments to evaluate the performance of these three approaches, and the results are depicted in Fig. 4. The results indicate that the S1 score exhibits significantly superior performance.

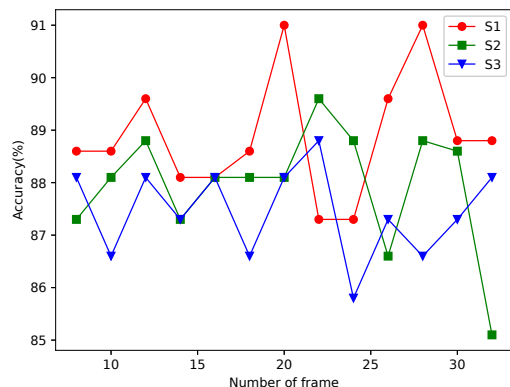


Fig. 4. Comparison of different calculation methods for S.

2) *Effect of the Action Space:* We investigated the impact of the action space  $A$  on the classification performance, and the results are shown in Table IV. Comparative analysis reveals that the model achieves the highest test set accuracy and the lowest computational load when the action space is defined as  $\{1, 3, 5, 7\}$ . Notably, when extending the action space to  $\{1, 3, 5, 7, 9\}$ , AccSampler sees an increase in GFLOPs and a drop in the test accuracy, which means broader action space makes the policy network more conservative instead.

TABLE IV

THE EFFECT OF DIFFERENT ACTION SPACES ON VIDEO CLASSIFICATION

$A$	Test acc	Training acc	GFLOPs/v	GFLOPs/f
$\{1, 3, 5\}$ <sup>1</sup>	<b>82.84</b>	97.24	3.41	0.053
$\{1, 3, 5, 7, 9\}$ <sup>2</sup>	77.60	<b>97.56</b>	3.26	0.051
<b><math>\{1, 3, 5, 7\}</math></b>	<b>82.84</b>	97.40	<b>2.79</b>	<b>0.044</b>

<sup>1</sup> Here  $R=\{224, 168, 112\}$

<sup>2</sup> Here  $R=\{224, 168, 140, 112, 84\}$

3) *Effect of Hyperparameter in Clip Mixup:* Variations in the parameter  $\alpha$  in clip mixup impact the scale parameter  $\lambda$ , influencing the model’s performance. The findings in Table V indicate that the model exhibits relatively low sensitivity to changes in  $\alpha$ . Optimal performance across all metrics is observed when  $\alpha$  is set to 0.3.

4) *Effect of the Station Points:* We experimented with varying the number of station points to assess their impact on classification, and the results are detailed in Table VI. The outcomes of the policy network with and without station points exhibit notable differences. Introducing just one station point leads to a 5% increase in test accuracy

TABLE V  
EFFECT OF DIFFERENT  $\alpha$  ON VIDEO CLASSIFICATION

$\alpha$	Test acc	Training acc	GFLOPs/v	GFLOPs/f
0.2	80.60	97.07	2.827	0.0442
0.4	<b>82.84</b>	97.07	2.831	0.040
0.5	81.36	97.24	2.811	0.0439
<b>0.3</b>	<b>82.84</b>	<b>97.40</b>	<b>2.792</b>	<b>0.044</b>

and a significant reduction in GFLOPs. In contrast, the policy network without station points tends to adopt a more conservative approach. As the number of station points increases, the policy network is inclined to fuse more frames. Ultimately, the policy network strikes a balanced trade-off between computation and accuracy with two station points.

TABLE VI  
EFFECT OF DIFFERENT NUMBER OF STATION POINTS ON VIDEO CLASSIFICATION

	Test acc	Training acc	GFLOPs/v	GFLOPs/f
0	75.37	<b>98.2</b>	6.61	0.103
1	80.60	98.1	<b>2.54</b>	<b>0.040</b>
<b>2</b>	<b>82.84</b>	97.40	2.79	0.044
3	78.36	97.7	3.06	0.048

## VI. CONCLUSION

In this study, we introduced AccSampler, a novel approach to enhancing the efficiency of aerial video understanding in wildfire monitoring tasks. Extensive experiments have demonstrated that AccSampler outperforms several baseline methods in fire detection, offering an excellent distilled dataset for TSM to achieve high precision with low computational costs. More importantly, AccSampler's compression mechanism is model-agnostic and compatible with various backbones, meeting the standards of diverse video-related disaster response tasks.

## REFERENCES

- [1] A. Bouguettaya, H. Zarzour, A. M. Taberkit, and A. Kechida, "A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms," *Signal Processing*, vol. 190, p. 108309, 2022.
- [2] O. Ozkan, "Optimization of the distance-constrained multi-based multi-uav routing problem with simulated annealing and local search-based matheuristic to detect forest fires: The case of turkey," *Applied Soft Computing*, vol. 113, p. 108015, 2021.
- [3] R. Nagasawa, E. Mas, L. Moya, and S. Koshimura, "Model-based analysis of multi-uav path planning for surveying postdisaster building damage," *Scientific reports*, vol. 11, no. 1, pp. 1–14, 2021.
- [4] S. Goyal, M. Shagill, A. Kaur, H. Vohra, and A. Singh, "A yolo based technique for early forest fire detection," *Int. J. Innov. Technol. Explor. Eng.(IJITEE) Vol.*, vol. 9, pp. 1357–1362, 2020.
- [5] D. Alexandrov, E. Pertseva, I. Berman, I. Pantiukhin, and A. Kapitonov, "Analysis of machine learning methods for wildfire security monitoring with an unmanned aerial vehicles," in *2019 24th conference of open innovations association (FRUCT)*. IEEE, 2019, pp. 3–9.
- [6] F. A. Hossain, Y. M. Zhang, and M. A. Tonima, "Forest fire flame and smoke detection from uav-captured images using fire-specific color features and multi-color space local binary pattern," *Journal of Unmanned Vehicle Systems*, vol. 8, no. 4, pp. 285–309, 2020.
- [7] J. Zhan, Y. Hu, W. Cai, G. Zhou, and L. Li, "Pd-am-stpnnet: a small target detection approach for wildland fire smoke through remote sensing images," *Symmetry*, vol. 13, no. 12, p. 2260, 2021.

- [8] S. Koshimura and E. Mas, "Digital twin computing for enhancing resilience of disaster response system," in *EGU General Assembly Conference Abstracts*, 2023, pp. EGU–11756.
- [9] A. Piergiovanni, A. Angelova, and M. S. Ryoou, "Tiny video networks," *Applied AI Letters*, vol. 3, no. 1, p. e38, 2022.
- [10] S. Liu, J. Cheng, L. Liang, H. Bai, and W. Dang, "Light-weight semantic segmentation network for uav remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 8287–8296, 2021.
- [11] C. Guo, B. Zhao, and Y. Bai, "Deepcore: A comprehensive library for coreset selection in deep learning," *DEXA*, 2022.
- [12] Y. Meng, C.-C. Lin, R. Panda, P. Sattigeri, L. Karlinsky, A. Oliva, K. Saenko, and R. Feris, "Ar-net: Adaptive frame resolution for efficient action recognition," in *European Conference on Computer Vision*. Springer, 2020, pp. 86–104.
- [13] Y. Bai, X. Wu, L. Xu, J. Pei, E. Mas, and S. Koshimura, "Towards efficient disaster response via cost-effective unbiased class rate estimation through neyman allocation stratified sampling active learning," *arXiv preprint arXiv:2405.17734*, 2024.
- [14] R. Gao, T.-H. Oh, K. Grauman, and L. Torresani, "Listen to look: Action recognition by previewing audio," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10457–10467.
- [15] J. Lin, H. Duan, K. Chen, D. Lin, and L. Wang, "Ocsampler: Compressing videos to one clip with single-step sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13894–13903.
- [16] Z. Wu, C. Xiong, C.-Y. Ma, R. Socher, and L. S. Davis, "Adaframe: Adaptive frame selection for fast video recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1278–1287.
- [17] X. Sun, R. Panda, C.-F. R. Chen, A. Oliva, R. Feris, and K. Saenko, "Dynamic network quantization for efficient video inference," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 7355–7365.
- [18] B. Korbar, D. Tran, and L. Torresani, "Scsampler: Sampling salient clips from video for efficient action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6232–6242.
- [19] Y. Meng, R. Panda, C.-C. Lin, P. Sattigeri, L. Karlinsky, K. Saenko, A. Oliva, and R. Feris, "Adafuse: Adaptive temporal fusion network for efficient action recognition," *arXiv preprint arXiv:2102.05775*, 2021.
- [20] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2678–2687.
- [21] S. N. Gowda, M. Rohrbach, and L. Sevilla-Lara, "Smart frame selection for action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1451–1459.
- [22] Y. Zhi, Z. Tong, L. Wang, and G. Wu, "Mgsampler: An explainable sampling strategy for video action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1513–1522.
- [23] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.
- [24] Z. Wu, C. Xiong, Y.-G. Jiang, and L. S. Davis, "Liteeval: A coarse-to-fine framework for resource efficient video recognition," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [25] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.
- [26] X. Sun, R. Panda, C.-F. R. Chen, A. Oliva, R. Feris, and K. Saenko, "Dynamic network quantization for efficient video inference," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7375–7385.
- [27] A. Shamsoshoara, F. Afghah, A. Razi, L. Zheng, P. Z. Fulé, and E. Blasch, "Aerial imagery pile burn detection using deep learning: The flame dataset," *Computer Networks*, vol. 193, p. 108001, 2021.
- [28] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: A large video database for human motion recognition," in *2011 International Conference on Computer Vision*, 2011, pp. 2556–2563.
- [29] 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*, 2019, pp. 7083–7093.