A Comparative Study: Error Analysis and Model Efficiency in Event-Based Action Recognition

Mira Adra
Eurecom
Biot, France
mira.adra@eurecom.fr

Jean-Luc Dugelay
Eurecom
Biot, France
dugelay@eurecom.fr

Cécile Ichard
GTD International
Toulouse, France
cecile.ichard@gtd.eu

***Abstract*—In the domain of event-based vision, action recognition stands as a significant challenge that pushes the boundaries of advanced computational models. This paper compares three cutting-edge architectures - Spiking Neural Networks, Graph Convolutional Neural Networks, and Video Transformer-based Networks - to determine their effectiveness in this domain. Our study extends beyond accuracy, focusing on the error nature of each model and its corresponding complexity. We observed that these models while achieving comparable accuracies, tend to make different types of mistakes. Capitalizing on this complementary error phenomenon, we aim to leverage their strengths by proposing an ensemble learning strategy, which improves overall performance. Moreover, we further investigated the Video Transformer model, retraining it on subsets of data that the Spiking Neural Network misclassified. This resulted in higher accuracy than when trained on subsets correctly classified, highlighting the Transformer's ability to learn differently and complement the Spiking Network's weaknesses. Our findings challenge the sole focus on accuracy for model efficacy emphasizing the significance of error analysis. This research provides a road map for model selection in event-based vision tasks and introduces innovative ways to integrate these models.**

***Index Terms*—Event Camera, Action recognition, Ensemble learning**

I. Introduction

Action recognition has always been a prominent field in computer vision due to its significant applications in areas like security and human-computer interaction. It has become more capable by leveraging the dynamic capabilities of the new-emerging bio-inspired event sensors. Unlike conventional cameras that capture static frames at fixed intervals, event cameras detect changes in light intensity at each pixel in an asynchronous manner and in order of microseconds. Their unique ability to record changes in a scene with precise temporal accuracy and low latency and their high dynamic range allows us to capture fast and subtle motions without any blur or being affected by lighting conditions. Moreover, event cameras offer an enhanced level of security for identity protection as they only record pixel changes caused by motion, without capturing static background scenes. This inherent characteristic is crucial in sensitive environments, ensuring that individuals' identities remain obscured, while still providing detailed movement analysis, which is crucial in high-security zones.

Despite the advancements in computational models for event-based vision, accurately recognizing actions remains a complex task, often hindered by the limitations of current model architectures. Recent research has mainly focused on three neural network architectures: Spiking Neural Networks (SNN), Graph Convolutional Neural Networks (GCNN), and most recently, Video Transformer-based Networks (VTN). Each of these models contributes its unique strengths to the table; however, there is a critical gap in understanding how these models perform specifically and whether each of these models builds upon the others' limitations or just proposes an independent approach. Our study aims to fill this gap by presenting a comparative analysis of these three architectures. Our motivation stemmed from the paradox that, despite the significant potential of this new emerging technology, its application is restricted by the limited availability of state-of-the-art models and public datasets tailored for this type of data. Besides, we aim to focus our evaluation beyond the conventional metric of accuracy, and discover the nature of the errors made by each model, and study the trade-off between model performance and complexity in action recognition tasks. Through this analysis, we observed that the three networks, while achieving comparable accuracies, tend to make different and often contrasting types of errors. Accordingly, we attempted to complement the strength of our models through an enhanced ensemble learning approach. We further investigated the Video Transformer-based model by retraining it on subsets of data misclassified by the other models, demonstrating its unique learning pattern. This approach revealed that the VTN's accuracy, although lower than that of the original VTN on these subsets, was still significantly better compared to training the same model on correctly classified subsets. To our knowledge, this is the first paper to delve into the specifics of the errors made by event-based models and aim to compare and combine their strengths for optimal implementation of event-based action recognition.

II. Related Work

Initially, few studies focused on action recognition with event cameras, but recent research has increased due to their dynamic nature and unique capabilities. The earliest and most

979-8-3315-4184-2/24/$31.00 ©2024 European Union

common approach is to transform the events into image frames such as time surface, histogram image, and two channel image and use them with traditional convolutional neural networks (CNNs). In another approach, Plizzari et al. [2] combined optical flow and event stream data using 3D CNNs to better capture micro-movements in the event stream. They trained their model to learn from both data types by freezing the flow stream and encouraging the event stream features to match those of the optical flow making the model simpler and faster.

Nevertheless, all these approaches use an image-like representation of event data. Another innovative direction in research is the development of networks specifically designed to handle event data directly. One straightforward approach, proposed by Tavanaei et al. [8], involves slicing event data into time windows and feeding them directly into Spiking Neural Networks as spike tensors. These networks, based on Integrate-and-Fire (IF) neurons, process information in the form of temporal spikes instead of numerical values, and thus, can handle event-based data without any pre-processing. GCNNs have been notably studied for action recognition due to their ability to capture complex spatiotemporal relationships using graph theory. These networks, such as those proposed by Schaefer et al. [9], adapt convolutional operations from traditional CNNs to handle graph-structured data. Key techniques include Gaussian Mixture Model-based graph convolutions for irregular data formats and Graph Residual Network layers to prevent vanishing gradients. These methods enhance spatial and temporal feature extraction, making GCNNs effective for action recognition and human tracking [11]. Most recently, De Blegiers et al. [4] introduced the EventTransAct video Transformer-based Network (VTN), which excels in handling the temporal and spatial dynamics of event data using the latest advancements in vision transformer technology. This model utilizes a spatial encoder and a LongFormer module for extracting spatial features from each event frame and learning global temporal dependencies across these frames, respectively. The main contribution of this model is the proposition of the Event Contrastive Loss (ECL) which enhances the model's ability to distinguish temporal details in event data by increasing the agreement between differently augmented versions of the same frame, while simultaneously ensuring that frames with different timings do not align. Although promising, the high complexity of this approach requires thorough comparative studies to determine the most effective model across various applications. Moreover, Rebecq et al. [10] set a new benchmark in performance by reconstructing high-resolution gray-scale frames from event data. While achieving high accuracy with traditional CNNs, their approach still faces certain limitations [12].

III. Methodology

A. Event Data Format

Event data is formatted as a sequence of individual events, each represented by a tuple e = (t, x, y, p) where t is the timestamp, (x,y) are the spatial coordinates, and p is the polarity which is assigned a value of 1 for positive changes and -1 for negative changes in light intensity. Given the asynchronous and sparse nature of event data, we will be using unique representations designed for event-specific neural networks.

For SNN, we directly slice the event data into discrete time windows, forming multi-dimensional spike tensors where each non-zero value element represents a spike at a specific time and location. For Graph CNN, we construct graph G= (V,E) from event data where V is the set of nodes and E is the set of edges or connections between them. For that, we consider each event to be an independent node with (x,y) as its spatial coordinates and polarity p as its initial node feature. Neighboring nodes are then connected with an edge based on their Euclidean distance in the spatio-temporal space. In the VTN model, we sample event data to obtain voxel grids that are later transformed into video clip frames by following the method proposed in [4]. Accordingly, we create a video V(i) from each sequence of events having T total frames, and we randomly pick n of these frames to make a shorter video clip.

B. Model Architectures

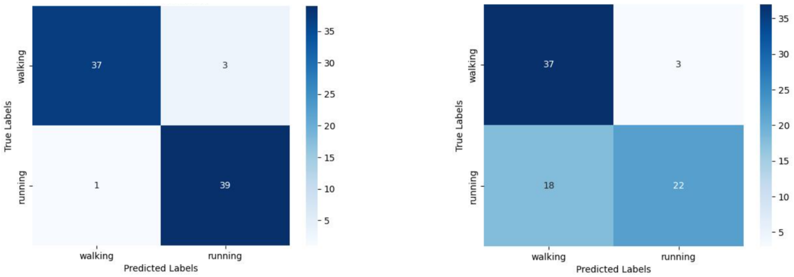
SNN. For our Spiking Neural Network (SNN), we utilize the Spike-Element-Wise-ResNet (SEW ResNet) architecture proposed in [1]. It is a novel adaptation of the traditional ResNet framework by substituting ReLU activation layers with spiking neurons. Moreover, we also use surrogate gradient learning for the back-propagation problem in SNNs allowing the network to overcome its limitations and handle more complex tasks, such as action recognition, and to create deeper SNNs with more than 100 layers.

Graph CNN. As for the Graph CNN, we rely on the EV-Gait-3DGraph model architecture originally proposed in [3] for gait recognition, and we adapt it to the task of action recognition from event data. This model starts by downsampling event streams and transforming them into 3D graphs for feature extraction. The network further applies graph clustering and MaxPooling to refine these features, leading up to a fully connected layer and a softmax classifier for final recognition.

Video Transformer Network. In our research, we adapted the EventTransAct video Transformer-based Network (VTN) framework as outlined in [4], drawing on the latest advancements in vision transformer technology. For our VTN model, we implement SlowFast's Vision Transformer code base inspired by [7]. For the Temporal Encoder, we used a 3-layer LongFormer with 8 attention heads. Within the framework of the Event Contrastive Learning (ECL), we utilized a multilayer perceptron (MLP) featuring a single hidden layer as the non-linear projection head, and configured it to produce an output with a feature dimensionality of 128. Moreover, we use stochastic data augmentation strategies, specifically implementing random cropping and random event dropping for a more robust learning process.

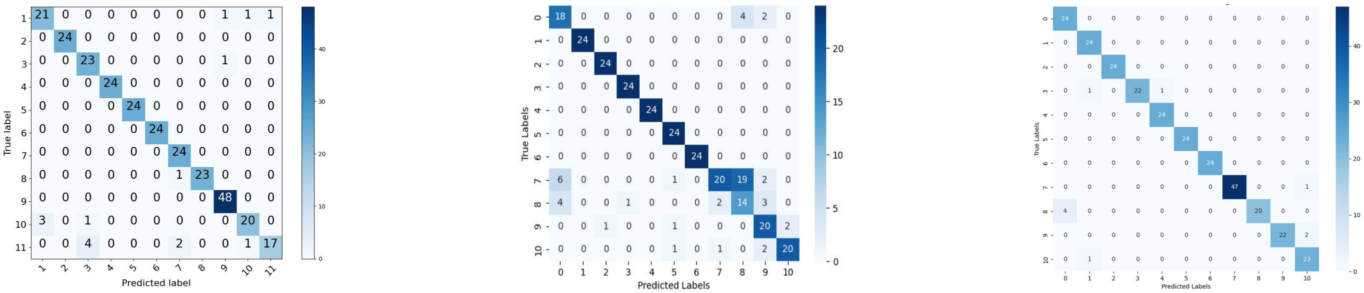
C. Criteria for Comparative Analysis

Our main aim is to go beyond the conventional comparative criteria for model performance and try to understand the types



(a) (b)

Fig. 1. Confusion matrices comparing the performance of Spiking Neural Network (SNN) and Graph Convolutional Neural Network (Graph CNN) on Gait3 dataset. Each image shows the classification results for one model: (a) SNN and (b) Graph CNN.



(a) (b) (c)

Fig. 2. Additional confusion matrices comparing the performance of Spiking Neural Network (SNN), Graph Convolutional Neural Network (Graph CNN), and Video-based Transformer Network (VTN) on DVSGesture dataset. Each image shows the classification results for one model: (a) SNN, (b) Graph CNN, and (c) VTN.

of mistakes these models are making. This way, we can analyze whether the better-performing models fix previous errors or make different mistakes. In our research, we use accuracy and confusion matrices as our primary comparison criteria. Confusion matrices are ideal for multi-class applications like action recognition because they provide straightforward visual representations of how the model is classifying each class, making it simpler to identify which actions are misclassified as others. Furthermore, we incorporate precision and recall as additional metrics to gain deeper insights into each model's ability to correctly predict and identify instances of each class. However, to comprehensively evaluate model complexity and performance, we also consider additional metrics such as the number of training parameters and the training time. The number of training parameters provides insight into the model's capacity and potential over-fitting risks while the training time is crucial for understanding the computational resources and efficiency required to train the model. These additional metrics offer a more comprehensive view of each model's complexity and performance, enabling a thorough comparative analysis.

IV. Experimental Results

In this section, we evaluate the performance of our three proposed architectures on three event-based gesture and action recognition datasets.

A. Datasets

Our study evaluates the three models across four distinct event-based datasets: our own Gait3 Dataset, the widely recognized DvsGesture dataset, the TUM Action Recognition dataset, and most recently DailyAction-DVS dataset.

Gait3 Action Dataset: This dataset is a modification of our Gait3 Database [5] collected for gait recognition. The original Gait3 was collected from 56 subjects with 414 recordings. Each subject walks in three ways: normally, quickly, and while carrying a backpack. Each recording has two variations: walking from left to right and from right to left. For action recognition, we use 276 recordings corresponding to the 2 classes: Normal walking and Quick walking. This way, we end up with around 9.5 minutes of data per class which allow us to obtain reliable results and leverage the high quality of the original Gait3 Dataset. The videos are split into 196 for training and 80 for testing.

DailyAction-DVS Dataset: This dataset consists of 1447 video clips featuring 15 individuals performing 12 everyday actions such as bending, climbing, falling, getting up, jumping, lying down, carrying a box, running, sitting down, standing up, walking, and picking up objects. The variety of recording conditions enhances its real-world applicability. Notably, this dataset has around 12 minutes of data per class which allows us to obtain reliable results. The videos are split into 1032 for training and 415 for testing.

DvsGesture Dataset: Comprising 1464 clips, this dataset captures 29 individuals performing 10 distinct gestures plus one category for random gestures (11 classes in total) under three different lighting scenarios. The participants were stationary while performing these gestures in front of a fixed DVS camera. Moreover, the average duration per class for this dataset is 12.2 minutes which allows us to obtain reliable results. The videos are split into 1176 for training and 288 for testing.

Action Recognition TUM Dataset: Featuring 291 recordings, this dataset showcases 15 individuals executing 10 different actions. The recordings were made using a DAVIS camera from 3 different points of view. However, this dataset has a limited duration of around 2.5 minutes of training data per class which might not be sufficient for having comparative results. The videos are split into 201 for training and 90 for testing.

B. Training Details

For the Spiking Neural Network, we use SEW ResNet architecture based on SpikingJelly. We train the model for 90 epochs at a learning rate of 0.001 for the first 60 epochs then it decreases until 0.0001 to prevent overfitting. For the Graph CNN we use MATLAB for data preprocessing then we train the model for 90 epochs at a learning rate of 0.001. For the Video Transformer, we use a VIT-based architecture and we train the model for 100 epochs at a learning rate of 4e-5. We also use cosine learning schedule for the last 90 epochs. All the models were trained using PyTorch and Adam optimizer on a 24 GB NVIDIA GeForce RTX 3090 GPU.

C. Results

We assess the accuracy of SNN and Graph CNN across the four datasets. The Video Transformer, however, is specifically trained on the DvsGesture dataset with 11 classes, as this dataset is well-suited for demonstrating the model's capabilities and aligns with the implementation presented in the original paper. As shown in Table 1, SNN achieves higher accuracy than Graph CNN on all datasets, and both particularly perform better on DailyAction-DVS and DvsGesture. This is due to the fact that there is sufficient data per class unlike ActionTUM, with roughly one-fifth that of the other datasets, which presents relatively low accuracy. Evidently, the Video Transformer attains the highest accuracy of 97.91% and it remains the better-performing model in research until now.

The models display intriguing complementary error patterns in the confusion matrices presented in Fig.1 and Fig.2 for both datasets. For the Gait3 Dataset, both the SNN and Graph CNN models misclassify walking as running at the same rate. However, the Graph CNN model more frequently misclassifies running as walking, with 18 false negatives, compared to only 1 false negative for the SNN model. A similar trend of contrasting errors is evident in the DvsGesture Dataset; the SNN model (a) shows a more uniform distribution of misclassifications, predominantly to the left of the diagonal, while the Graph CNN specifically struggles with gestures 1 and 7 to the right of the diagonal, showing a higher rate of false negatives. In comparison, the VTN model (c) exhibits an uneven distribution of errors, with classes such as 8 and 9 being more prone to misclassification. The variation in error patterns across these models highlights the strengths and weaknesses of each model in recognizing complex actions. This suggests an opportunity for ensemble methods that could leverage the diverse error tendencies of each model to complement each model's errors, thereby improving overall classification accuracy.

In addition to the confusion matrices, we have also analyzed the precision and recall values for each class across the three models as shown in Fig.3. The GCNN model shows lower recall for Class 7 and Class 8, indicating it misses many instances of these classes, which is complemented by the SNN and Transformer models that maintain higher recall for these classes. The SNN model, while generally strong, exhibits lower recall for Class 10, which is better handled by the Transformer model. When examined alongside the confusion matrices, these variations in precision and recall highlight the different error patterns of each model, underscoring the complementary strengths that can be leveraged through an ensemble approach.

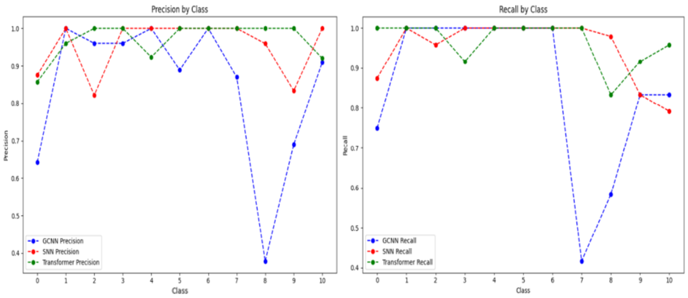


Fig. 3. Precision (a) and Recall (b) curves per class for comparing the performance of SNN, Graph CNN, and VTN on DVS Gesture dataset

V. Ensemble Learning

A. Ensemble Model Description

In order to complement the mistakes made by the three models, we implement ensemble learning using a stacking module. We train stacking using LogisticRegression, GradientBoosting, RandomforestClassifier, and a custom Stacking neural network to decide which has the best results. We also analyze whether the ensemble model favors one of the models when making a decision. The main issue with implementing such an approach is that the data representations for Graph CNN, SNN, and VTN are very different so it is not possible to just feed the testing data into the models.

That's why, we create a pre-processing pipeline to align data instances from the three models and then shuffle them with the same Random seed. After data processing, we train the best-performing stacking model using the predictions of the three architectures on the training dataset and their corresponding confidence levels. Then, we compare the performance of the ensemble model on the test set.

One challenge with this approach is that the models have already been trained on the training dataset. So, for ideal results, we retrain the models using 5-fold cross-validation. In this method, the dataset is divided into 5 folds, and for each fold, the model is trained on 4 parts of the dataset and then validated on the remaining part. We combine the 5 parts of the validated predictions to get a whole run of predictions on the training dataset as if they were never seen before by the model and that is then fed to the stacking model.

TABLE I
COMPARISON OF ACTION RECOGNITION ACCURACY.

Method	DailyAction-DVS	DvsGesture	Gait3	ActionTUM
Spiking Neural Network	94.69%	93.40%	95.00%	89.69%
Graph CNN	90.60%	81.94%	73.75%	64.62%
Video Transformer	-	97.91%	-	-

TABLE II
ENSEMBLE LEARNING RESULTS ON DVS GESTURE DATASET.

Model	Accuracy	Training Time	Number of Parameters
Spiking Neural Network	94.31%	24.1 mins	130426 params
Graph CNN	84.46%	36.8 mins	7658159 params
Video Transformer	96.21%	785 mins	113400715 params
Ensemble1 (SNN & GCNN)	96.18%	<1 min	5195 params
Ensemble2 (SNN & VTN)	97.34%	<1 min	5195 params
Ensemble3 (GCNN & VTN)	97.34%	<1 min	5195 params
Ensemble4 (all 3 models)	97.34%	<1 min	5323*params

* The ensemble has 5323 parameters, but it requires training the SNN, GCNN, and VTN models a priori.

B. Evaluation and Analysis

1) Performance

Our research focuses on comparing three cutting-edge models and evaluating the effectiveness of ensemble learning in enhancing action recognition accuracy. To this end, we systematically explored all possible pairings between the three models, creating a series of dual-model ensembles to fully assess their combined potential on the DVSGesture dataset. The results of this table were obtained with different data shuffles than the ones in Table 1, making them specific to this particular run.

As shown in Table 2, Ensemble1 (SNN & GCNN) achieved a 1.87% marginal increase over SNN and a notable 11.72% increase over the GCNN model. Ensemble2 (SNN & VTN) shows significant improvement, matching the top accuracy of 97.34%, demonstrating that SNN and VTN work well together. Ensemble3 (GCNN & VTN) continues to perform well and match the highest accuracy, outperforming the standalone VTN by 1.13%. It is noteworthy that the combined application of all three models in Ensemble4, combining all three models, did not surpass the accuracy of Ensembles 2 and 3, suggesting that while VTN is crucial for optimal performance, the combined benefits with SNN and GCNN may not be additive. The halt in performance could mean that the ensemble has achieved the highest level of efficacy given the available data and technique, or it could mean that the models are extracting features that overlap.

Although the ensemble models generally outperform their individual counterparts, the improvements are modest. In high-performance models with accuracies around 90%, even minor increases are significant. These results highlight the importance of carefully choosing models for ensemble integration to achieve optimal performance. To validate this hypothesis, we calculated p-values between the ensemble and the original models at a 95% confidence level. For Ensemble1, the p-value is 0.033 compared to SNN and 7.2e-22 compared to GCNN, indicating that the differences in accuracy are statistically significant. For Ensemble2, the p-value is 0.0002 compared to SNN, but 0.12 compared to VTN, suggesting that while SNN and GCNN complement each other effectively, the improvement when adding VTN to SNN is not statistically significant. This could be because VTN's high performance leaves less room for noticeable improvements, despite any complementary error patterns.

2) Complexity

Now we extend our comparative analysis beyond performance and also compare the complexity of the models, measured in terms of training time and the number of parameters. As shown in Table 2, the SNN model required a training time of 24.1 minutes and had 130,426 parameters. The Graph CNN had a slightly higher training time of 36.8 minutes and 7,658,159 parameters. In contrast, the VTN model, while achieving the highest accuracy, demanded a significantly longer training time of 785 minutes and had the highest number of parameters at 113,400,715. When analyzing the ensembles, they all showed a reduced complexity with a training time of less than 1 minute and only 5,195 parameters for the dual ensembles and 5323 for the ensemble combining the three models, while achieving improved accuracies of 96.18% for Ensemble1 and 97.34%, for the other combinations. It is important to note that the ensemble models require prior training of the SNN, GCNN, and VTN models, which adds to the overall complexity and computational cost. This additional step is necessary to generate the predictions used for training the ensemble.

These results underscore the trade-off between accuracy and complexity. While the VTN model provides the highest individual accuracy, it requires significantly more computational resources. On the other hand, the ensemble models that include SNN and GCNN maintain lower combined complexity compared to VTN alone while still achieving high accuracy. However, when all three models are combined in Ensemble4, the complexity increases significantly, suggesting that the marginal gains in accuracy may not justify the added complexity. Overall, combining SNN and GCNN offers a good balance of performance and manageable complexity, whereas incorporating VTN, while boosting accuracy, significantly increases complexity. Therefore, careful consideration is needed when including VTN to ensure the trade-off between accuracy and complexity aligns with the application's requirements.

VI. Ablation Study on Video Transformer Model

A. Description

The Video Transformer model, while demonstrating the highest accuracy in our analysis, presents certain challenges in terms of training speed and complexity. To further investigate its performance and complementary nature with Spiking Neural Networks (SNNs), we conducted an ablation study using a targeted training approach. This study aimed to analyze the impact of selectively training the VTN model on subsets of data identified by the SNN model as challenging and to prove the fact that both models' errors are complementary. Accordingly, We first train the SNN on the entire DVSGesture dataset to pinpoint a subset of data points that are particularly challenging, indicated by the model's low prediction confidence. Then, the VTN model is trained on this challenging data and tested on the test set.

B. Results

In our approach, we use the DVSGesture dataset comprising 1176 training data points. After training the SNN model, we obtained 385 data points that we identified as challenging data based on a confidence threshold of 90%.

TABLE III
Comparison of three Transformer models.

Method	**Accuracy**	**Time (hours)**
Original VTN	97.91%	13.083
VTN trained on Difficult Data	92.01%	3.667
VTN trained on Easy Data	85.41%	3.667

As shown in Table 3, the VTN model trained on difficult data achieved an accuracy of 92.01% with a training duration of 3 hours and 40 minutes. Although this accuracy is lower than the original VTN model's 97.91%, the training time was reduced by a factor of approximately 3.57, and the data size was reduced by about 3.05 times, utilizing only around 32% of the entire dataset. This indicates a favorable balance between maintaining high accuracy and addressing the training speed of the original model.

To further validate our findings, we conducted another experiment where the SNN model selected a subset of data points with the highest prediction confidence, maintaining the same size as the previous subset. Under identical training conditions, the transformer model's accuracy decreased to 85.41%. This result confirms that the relatively high performance of the VTN model when trained on difficult data points is due to the strategic selection of challenging data. This highlights the complementary nature of the VTN and SNN models, as the VTN model can effectively learn from the errors identified by the SNN model. This also emphasizes that different methodologies (ensemble learning vs. targeted training on misclassified data) reveal different aspects of model performance and complementarity.

VII. Conclusion

In this study, we conducted an extensive comparison of three model architectures for event-based action recognition. We also leveraged the strengths of these models by creating an ensemble learning approach, which demonstrated promising results. However, despite its superior performance, the Transformer model proved to be relatively complex. For further comparison and error analysis, we performed targeted training of the Transformer on challenging data identified by the SNN network, demonstrating the complementary nature of their errors. Our findings emphasize the significance of model selection based on both performance and complexity, paving the way for more efficient solutions in event-based action recognition.

Acknowledgment

This research is a part of the HEIMDALL project, funded by the BPI as part of the AAP I-Démo.

References

[1] W. Fang, Z. Yu, Y. Chen, T. Huang, T. Masquelier, and Y. Tian, "Deep Residual Learning in Spiking Neural Networks," *arXiv*, 2022.

[2] C. Plizzari et al., "E2(GO)MOTION: Motion Augmented Event Stream for Egocentric Action Recognition," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 2022, pp. 19903–19915.

[3] Y. Wang et al., "Event-Stream Representation for Human Gaits Identification Using Deep Neural Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021, pp. 1–1.

[4] T. de Blegiers, I. R. Dave, A. Yousaf, and M. Shah, "EventTransAct: A video transformer-based framework for Event-camera based action recognition," *arXiv*, 2023.

[5] M. J. Eddine and J. Dugelay, "GAIT3: An Event-based, Visible and Thermal Database for Gait Recognition," in *2022 International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2022.

[6] Y. Xing, G. Di Caterina, and J. J. Soraghan, "A new spiking Convolutional Recurrent Neural Network (SCRNN) with applications to Event-Based Hand Gesture recognition," *Frontiers in Neuroscience*, vol. 14, 2020.

[7] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video Transformer Network," in *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021.

[8] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. S. Maida, "Deep Learning in Spiking Neural Networks," *Neural Networks*, vol. 111, pp. 47–63, 2019.

[9] S. Schaefer, D. Gehrig, and D. Scaramuzza, "AEGNN: Asynchronous Event-based Graph Neural Networks," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[10] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "Events-to-Video: Bringing Modern Computer Vision to Event Cameras," *arXiv*, 2019.

[11] D. Eisl, F. Herzog, J.-L. Dugelay, L. Apvrille, and G. Rigoll, "Introducing a framework for Single-Human tracking using Event-Based cameras," 2023.

[12] M. Adra and J.-L. Dugelay,"TIME-E2V: Overcoming limitations of E2VID," in *IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2024.