# Towards Real-Time Online Egocentric Action Recognition on Smart Eyewear

Riccardo Santambrogio[1](✉), Federico Caspani[1], Greta Corti[1], Francesca Palermo[2], Simone Mentasti[1], Diana Trojaniello[2], and Matteo Matteucci[1]

[1] Department of Electronics, Information and Bioengineering (DEIB), Politecnico di Milano, Via Ponzio 34/5, 20133 Milan, Italy
{riccardo.santambrogio,federico.caspani,greta.corti,simone.mentasti, matteo.matteucci}@polimi.it

[2] EssilorLuxottica Italia S.p.A., Piazzale Cadorna 3, 20123 Milan, Italy
{francesca.palermo,diana.trojaniello}@luxottica.com

**Abstract.** Recently, augmented reality and wearable devices, such as smart eyewear systems, have gained significant attention due to advancements in computer vision technology and the proliferation of compact wearable cameras. This has led to an increased interest in egocentric vision, which offers a unique perspective for recognizing human actions and understanding behavior from a first-person view. However, existing approaches for egocentric action recognition often rely on complex architectures with high computational demands, such as large transformers, which are unsuitable for real-time applications on wearable devices with limited processing power. This work aims to develop a lightweight, real-time egocentric action recognition system tailored for resource-constrained environments. We evaluate the recent LaViLa model for online adaptation and explore the use of the lightweight MiniROAD model, initially designed for exocentric Online Action Detection, on egocentric data. By creating a focused dataset, EgoClip Office, we can optimize the model for our specific application. Our approach is validated on an Nvidia Jetson platform, demonstrating the feasibility of achieving real-time performance on low-power embedded devices.

**Keywords:** Smart Eyewear · Egocentric Vision · Human Action Recognition · Edge Device

## 1 Introduction

Recent years have seen a growing interest in the topics of augmented reality and wearable devices, such as smart eyewear systems. This trend is driven by the growing availability of compact wearable cameras and significant advancements in computer vision technology. With the recent explosion of egocentric video data, the task of recognizing human actions from a first-person perspective becomes a very interesting prospect, representing a unique and innovative

A. Del Bue et al. (Eds.): ECCV 2024 Workshops, LNCS 15645, pp. 281–294, 2025.
https://doi.org/10.1007/978-3-031-91989-3_18

way to understand and interpret human behavior. The nature of egocentric vision brings it close to the perception of the camera wearer, highlighting important details such as interactions with objects. At the same time, it poses peculiar challenges compared to exocentric vision, like partial body visibility and the unpredictable motion of a head-mounted camera. Overall, this emerging field holds significant implications for a wide range of applications, from sports and health monitoring to augmented reality and human-computer interaction [21].

Various approaches have been proposed for egocentric action recognition and related tasks of egocentric video understanding. Some try to exploit different modalities [31,35], combine data from the exocentric and egocentric domains [17,27,34], or make use of large language models (LLM) to build video-language representations [3,19,38]. While these methods can provide very accurate results, most of them adopt complex architectures, like large transformers, with a high parameter count and long inference times. These computational requirements become a critical limitation in the context of real-time applications when we are required to recognize user actions as they are being performed, especially when limited computational power is available.

In this work, we seek to find an approach for egocentric action recognition that can be performed in real time close to the sensor, in resource-constrained environments, or in embedded systems such as smart eyewear. This task proves challenging since wearable devices usually lack the computational power required by complex state-of-the-art egocentric models, and requires to accurately balance a trade-off between accuracy and computational requirements. Moreover, our algorithm must be designed to be applicable online, operating on the incoming stream of data with minimal use of explicit memory or look-ahead information, in order to reduce the recognition latency.

We start our analysis from the recent LaViLa model [38], which builds a powerful vision-language representation that achieves state-of-the-art results on many benchmarks for video-level egocentric action recognition, and we investigate the feasibility of adapting it to work in an online fashion. We then turn to the lighter MiniROAD [1], which has been proposed specifically for the task of Online Action Detection (OAD) [14], showing that a small Recurrent Neural Network (RNN), if trained properly, can compete and even outperform larger transformers in this specific task. As MiniROAD has been evaluated only on third-person recognition tasks, we are, to the best of our knowledge, the first to test it on egocentric video data. To do this, we make use of popular egocentric datasets including EgoClip [19], from which we extract a restricted set of actions focused on the office setting, obtaining a custom, more manageable dataset which we will refer to as EgoClip Office.

Throughout our investigation, with each modification to the model's architecture, we have to closely monitor the algorithm's size and speed. Given our ultimate goal of bringing egocentric action recognition to wearable devices with restricted computational capabilities, we validate our approach by deploying it on the Nvidia Jetson Orin Nano platform, which represents a popular commercial solution for low-power embedded devices. On this platform we can achieve real-time or quasi-real-time performance using appropriately adapted proven models.

## 2 Related Works

In the last decade, as affordable and lightweight wearable devices became increasingly available, the number and size of egocentric video datasets has also grown [25,26,31,36]. Epic-Kitchens-100 [6,7] released a dataset of unprecedented size in egocentric vision, with 100 h of video containing over 90K action clips, although it is focused on the single domain of cooking and kitchen activities. EGTEA Gaze+ [18] is a smaller dataset of cooking activities, containing about 10K samples divided into 106 classes, which additionally provides annotations of human gaze tracking and hand masks. It is not until the release of Ego4D [12], however, that egocentric vision sees its first massive-scale dataset, consisting of $3,670$ hours of video recorded in 74 worldwide locations, annotated using natural language narrations. In addition to the data, Ego4D presents a variety of benchmarks and challenges that have garnered attention to egocentric video understanding. Although Ego4D is unmatched in terms of size and diversity, the nature of its annotations makes it challenging to adopt for the task of action recognition. EgoClip [19] aims to bridge this gap by filtering and processing Ego4D to obtain 3.8M precise clip-text pairs, covering $2,927$ hours of video.

While the availability of egocentric data improved, different techniques were proposed for the automatic analysis and understanding of egocentric video. Powered by the large scale of language-annotated datasets, and inspired by the success of analogous approaches in third-person vision [22,24], some works adopt contrastive learning to build vision-language representations.

In [19], egocentric video-language pretraining (EgoVLP) was proposed to leverage the EgoClip dataset. This work has now progressed to its second iteration [23]. Similarly, HierVL [3] builds on the same principle but introduces a hierarchical structure to the embedding space during training, linking fine-grained clip-level embeddings with video-level ones. LaViLa [34] also trains a dual-encoder using contrastive learning, but it uses two LLMs, repurposed to be visually conditioned, to greatly expand the number and variety of video narrations: a *narrator* model is used to label new clips, and a *rephraser* model is used to paraphrase the narrations. Using this approach, LaViLa obtains a powerful pre-trained representation that achieves state-of-the-art results in many downstream tasks of egocentric vision, including action recognition on EGTEA, Epic-Kitchens-100, and other benchmarks.

Much effort is dedicated, in egocentric vision, to identifying actions through the analysis of whole clips. The field of Online Action Detection (OAD) deals instead with the general task of recognizing human action in a streaming video as soon as it happens, without access to future information [8,14]. Earlier methods relied on extracting frame-level features, typically using a two-stream backbone [28], which are then fed to an RNN-based model to extract temporal patterns and recognize the beginning of an action [9–11,16,32]. With the emergence of the transformer [8], the focus later shifted towards this architecture which obtains higher accuracy [30,33,37]. However, MiniROAD [1] shows that minimal RNN models can still compete and even outperform transformers on the task of OAD, where their intrinsic bias towards most recent information is beneficial, when they are trained using a nonuniform loss that only considers the latest time step. The authors validate MiniROAD only on exocentric datasets, while we apply it for the first time to egocentric video. Only recently, egocentric OAD is starting to receive greater attention [2].

## 3 LaViLa for Online Recognition

LaViLa has demonstrated exceptional performance in egocentric action recognition, achieving state-of-the-art accuracy on a variety of datasets. However, it is designed for video-level recognition, and it classifies the action occurring in a video having access, in principle, to all frames. In practice, in its evaluation procedure, it takes multiple clips from the full video as input, with lengths of 16 or 32 frames each. This approach faces some issues when applying the model to a video stream that is recorded in real time. First of all, as we are required to collect a lot of frames to feed to the model, it introduces a considerable latency before an action can be recognized, thus limiting also the granularity of actions that can be identified. Additionally, storing long durations of video data may be too onerous on a low-power embedded device.

These concerns raise the question of whether we can take advantage of LaViLa when operating on streaming video data. To answer this and gain insights into the model's behavior, we conducted a series of experiments focused on assessing the impact of reducing the number of sampled clips and adjusting the number of frames for each clip. Our goal is to evaluate the trade-off between inference time, data efficiency, and model performance, thereby informing the development of an optimized deployment strategy within the constraints of the online setting.

For these experiments, we use the publicly available implementation of LaViLa, based on the TimesSformer (TSF-B) [4] backbone, and the weights for action recognition trained on the EGTEA dataset. We refer to the original paper [38] for details on how this dual-encoder was trained. Results of this first test are shown in Table 1, where we compare the usage of fewer clips and frames with the official model evaluation that takes ten evenly spaced clips of 32 frames each.

From this test, we find that we can drastically reduce the total number of frames considered by the model, using only three clips of 16 frames, while losing less than 1% in Top-1 Accuracy. However, sampling multiple clips from different parts of the video, albeit only a few, is still unrealistic in the context of online recognition. Using a single clip of 16 consecutive frames achieves remarkable accuracy, losing only about 10% from the evaluation reported by the original paper. On the other hand, further reducing the amount of data by decreasing the clip length can quickly degrade the performance, and using only 4 frames drops the accuracy by an additional 20%. Overall, we argue that the 16-frame single-clip configuration strikes a reasonable trade-off between accuracy and latency,

**Table 1. LaViLa classification on EGTEA** with varying number of clips per video and clip lengths, all numbers are percentages. *Results reported in the original paper.

| | 1 clip | | 2 clips | | 3 clips | | 10 clips | |
|---|---|---|---|---|---|---|---|---|
| Frames per clip | Mean Acc. | Top-1 Acc. | Mean Acc. | Top-1 Acc. | Mean Acc. | Top-1 Acc. | Mean Acc. | Top-1 Acc. |
| 4 | 41.15 | 48.42 | 52.37 | 59.60 | 58.22 | 64.49 | – | |
| 8 | 49.27 | 57.91 | 60.54 | 67.85 | 64.47 | 71.74 | | |
| 16 | 59.22 | 68.45 | 64.15 | 72.80 | 69.00 | 76.61 | | |
| 32 | – | | – | | – | | 70.12* | 77.45* |

considering that, at a standard camera framerate of 30 fps, the model can predict an action after a little more than 0.5$s$ from its beginning.

## 4 MiniROAD for Real-Time Egocentric Recognition

In the previous section, we showed that a powerful egocentric model like LaViLa can be applied to an incoming video stream with a minimal delay, sacrificing a modest amount of accuracy. However, other issues arise when using such large models, like memory size and inference time. On the other hand, MiniROAD has shown great promise in the context of exocentric OAD using a lightweight RNN, but its effectiveness for first-person videos remains uncertain.

MiniROAD is structured into two components: a frame-level feature extractor, which converts frames into a sequence of feature vectors, and a recurrent classification head. The feature extractor can be, in principle, any model, but a common choice in the literature for OAD is the Temporal Segment Network (TSN) [28]. TSN is based on a two-stream architecture: a *spatial ConvNet* extracts spatial features from standard RGB frames, while a *temporal ConvNet* extracts temporal and motion information from optical flow fields. For the convolutional backbones of TSN, the ResNet-50 [13] is usually adopted.

This feature extraction scheme requires that optical flow be computed beforehand, to then feed into the temporal branch of TSN. In [1], this is achieved using Denseflow [29] to estimate optical flow from RGB frames. Denseflow can generate accurate optical flow results, but its accuracy comes at a significant computational cost. Indeed, the authors of MiniROAD observe that optical flow computation represents the latency bottleneck of the entire pipeline, hindering its real-time application.

To overcome this challenge, we restructure and simplify the temporal stream of the feature extraction segment. Instead of computing optical flow and then processing it with a convolutional model, we merge the extraction of motion features into a single step, by applying a deep optical flow encoder directly to the RGB frames. Specifically, we take the pre-trained encoder of the Light Flow [39] model, which is characterized by an efficient network design, tailored for real-time processing on mobile devices with limited computing power. Since

**Table 2. Architecture of the flow feature extractor**, ‘dw’ denotes a depthwise separable convolution. Highlighted rows represent the additional convolutional layers that we introduce, while the initial part uses the pre-trained layers from the encoder of Light Flow [39].

| Name | Filter Shape | Stride | Output size | Input |
|---|---|---|---|---|
| Images | | | $256 \times 256 \times 6$ | |
| Conv1.dw | $3 \times 3 \times 6$ dw | 2 | $128 \times 128 \times 6$ | Images |
| Conv1 | $1 \times 1 \times 6 \times 32$ | 1 | $128 \times 128 \times 32$ | Conv1.dw |
| Conv2.dw | $3 \times 3 \times 32$ dw | 2 | $64 \times 64 \times 32$ | Conv1 |
| Conv2 | $1 \times 1 \times 32 \times 64$ | 1 | $64 \times 64 \times 64$ | Conv2.dw |
| Conv3.dw | $3 \times 3 \times 64$ dw | 2 | $32 \times 32 \times 64$ | Conv2 |
| Conv3 | $1 \times 1 \times 64 \times 128$ | 1 | $32 \times 32 \times 128$ | Conv3.dw |
| Conv4a.dw | $3 \times 3 \times 128$ dw | 1 | $16 \times 16 \times 128$ | Conv3 |
| Conv4a | $1 \times 1 \times 128 \times 256$ | 1 | $16 \times 16 \times 256$ | Conv4a.dw |
| Conv4b.dw | $3 \times 3 \times 256$ dw | 1 | $16 \times 16 \times 256$ | Conv4a |
| Conv4b | $1 \times 1 \times 256 \times 256$ | 1 | $16 \times 16 \times 256$ | Conv4b.dw |
| Conv5a.dw | $3 \times 3 \times 256$ dw | 1 | $8 \times 8 \times 256$ | Conv4b |
| Conv5a | $1 \times 1 \times 256 \times 512$ | 1 | $8 \times 8 \times 512$ | Conv5a.dw |
| Conv5b.dw | $3 \times 3 \times 512$ dw | 1 | $8 \times 8 \times 512$ | Conv5a |
| Conv5b | $1 \times 1 \times 512 \times 512$ | 1 | $8 \times 8 \times 512$ | Conv5b.dw |
| Conv6a.dw | $3 \times 3 \times 512$ dw | 1 | $4 \times 4 \times 512$ | Conv5b |
| Conv6a | $1 \times 1 \times 512 \times 1024$ | 1 | $4 \times 4 \times 1024$ | Conv6a.dw |
| Conv6b.dw | $3 \times 3 \times 1024$ dw | 1 | $4 \times 4 \times 1024$ | Conv6a |
| Conv6b | $1 \times 1 \times 1024 \times 1024$ | 1 | $4 \times 4 \times 1024$ | Conv6b.dw |
| Conv7.dw | $3 \times 3 \times 1024$ dw | 2 | $2 \times 2 \times 1024$ | Conv6b |
| Conv7 | $1 \times 1 \times 1024 \times 1024$ | 1 | $2 \times 2 \times 1024$ | Conv7.dw |
| Conv8.dw | $3 \times 3 \times 1024$ dw | 2 | $1 \times 1 \times 1024$ | Conv7 |
| Conv8 | $1 \times 1 \times 1024 \times 1024$ | 1 | $1 \times 1 \times 1024$ | Conv8.dw |

this encoder outputs spatially structured feature maps, we extend it with two additional convolutional layers to obtain feature vectors of the same size as the spatial stream features. These layers are trained together with the recurrent classification head. The full network architecture of our flow feature extractor is shown in Table 2.

**Table 3. Taxonomy considered by actions in EgoClip Office**. Verb and noun groups contain synonyms for each term, as defined in [19].

| Action | Verb Groups | Noun Groups |
|---|---|---|
| background | | |
| use computer | operate, inspect, touch, hold, scroll, read | computer |
| use phone | operate, inspect, touch, hold, scroll, read | phone |
| talk with people | talk | person |
| walk | walk | |
| watch tv | watch | television |
| use calculator | operate, press | calculator |
| move cup | hold, move | cup |
| use keyboard | press | keyboard |
| use mouse | hold, move, operate, press, put | mouse |
| move book | move | book |
| read book | read | book |
| move bottle | hold, open, close, put, touch, take, move, consume | lid |
| move pen | hold, open, close, put, touch, take, move | pen, pencil |
| write with pen | write | pen, pencil |

## 5 EgoClip Office

EgoClip [19] was obtained from a process of data curation of the Ego4D dataset. This involves various steps, including a pairing strategy to associate narrations with appropriately sized clips centered around their timestamps, and the identification of a taxonomy of verbs and nouns where synonyms are merged. The result is a dataset that is more suitable for the action recognition task since it provides short clips representing individual actions. Nevertheless, it is intended as a pre-training dataset, and its scale and annotation variety make it hardly manageable for traditional action recognition. Indeed, it contains 116 verbs and 555 actions, which are combined in various ways.

For these reasons, to allow training and testing action recognition models, we build a restricted version of EgoClip. From the vast array of actions, we handpick 15 action labels focusing on those that are likely to occur in an office setting. These include typing on a keyboard, using a phone, engaging in conversation, and navigating spaces. First, we filter out clips that are shorter than 10 frames. Then, a clip is assigned to a class if its narration contains a verb and a noun belonging to a predefined set of words. We limit the number of clips in each class to $2,000$, and define an additional *background* class which is populated with random unassigned clips. Table 3 reports the taxonomy that was used to build our EgoClip Office, while the full distribution of frames per class is depicted in Fig. 1. This refined selection allows us to consider more targeted applications and work on a contained scale.

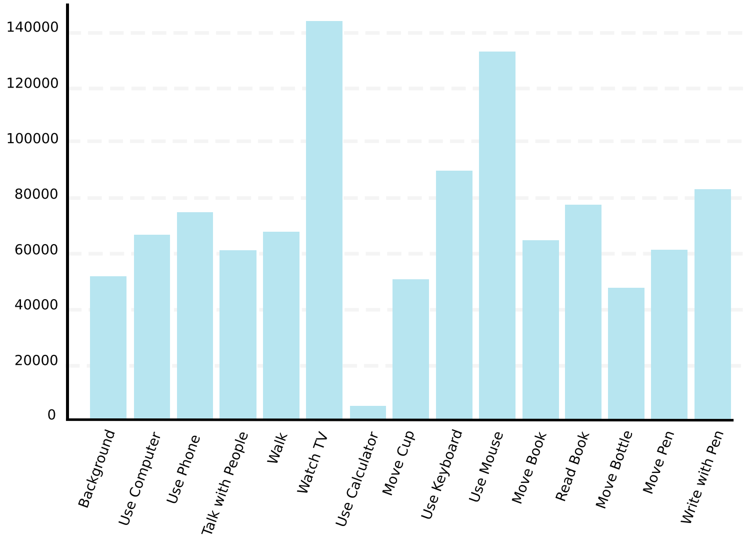

**Fig. 1. Class distribution for EgoClip Office.**

# 6 Experiments

## 6.1 MiniROAD with Lightweight Feature Extraction

To validate the modified feature extraction scheme described in Sect. 4, we first train and test MiniROAD on the THUMOS'14 dataset [15]. THUMOS'14 is a third-person dataset containing 20 h of sport scenes. We follow the same training and testing procedure as in [1] to allow a direct comparison with their results: we use windows of 128 frames with a stride of 4, and train the model for 10 epochs using the AdamW optimizer with a learning rate of $1e-4$. The evaluation metric is the commonly used per-frame mean Average Precision (mAP). In addition to this metric, we report model size and the inference speed in terms of frames per second (FPS) to show the speedup of our approach compared to the slower optical flow computation, measured on an NVIDIA RTX A6000 GPU. Table 4 presents our results.

Although our lightweight feature extractor leads to a drop in mAP of about 9%, this is counterbalanced by a massive speed improvement of more than 5 times. The number of parameters is also drastically lower, by almost 6 times, which is crucial since our final goal is to operate on an embedded system with limited resources. Finally, we test the model when the temporal branch is completely removed and only RGB features are used. The mAP, in this case, drops at 59.7%, proving that our lightweight approach to extracting motion features is beneficial and improves model performance.

**Table 4. Comparison of flow feature methods** on THUMOS'14, using MiniROAD classifier. Light Flow uses our modified encoder architecture.

| OF Feature Method | Parameters | OF Feature FPS | Overall FPS | mAP (%) |
|---|---|---|---|---|
| TSN | 24.3M | 27.3 | 23.8 | 71.8 |
| Light Flow (Ours) | 4.2M | 312.2 | 126.4 | 62.7 |

### 6.2 MiniROAD on Egocentric Video

Once verified that our lightweight features extraction approach can achieve good results on a standard exocentric benchmark, we now aim to understand the accuracy of the model in the context of egocentric vision. To this end, we consider two egocentric datasets: the EGTEA dataset and our EgoClip Office presented in Sect. 5.

**EGTEA.** EGTEA [18] is a large collection of everyday actions in a kitchen environment, captured through a head-mounted camera that well reflects the perspective of the camera wearer. Although it provides annotations of gaze tracking, we only use the RGB frames for our experiments. This dataset is divided into standard splits for training, validation and testing. We train our modified MiniROAD architecture according to the guidelines described by the original authors: the model undergoes training using sequences of 128 frames, optimizing a loss function that focuses exclusively on the final time step. To deal with the strong class imbalance of this dataset, we use the focal loss [20], which dynamically scales the contribution of hard misclassified samples. We find experimentally that this loss improves the performance compared to the regular cross-entropy loss.

We also implement a larger version of MiniROAD, with increased complexity in the recurrent classification head, which in the original model is simply composed by an embedding layer, a single GRU [5] and a final classification layer. Instead, we increase the number of GRU layers to 3, we add a dense layer as a parallel branch to the GRU, and combine the features from the two branches with an additional dense layer before feeding into the classification layer. These changes increase the model's parameter count from 15.8M to 31.5M, excluding the feature extraction backbones. Timing measurements of this larger architecture on an Nvidia Jetson platform are shown in Sect. 7.

In Table 5 we report the results of our modified MiniROAD model, both using the original classification head and the larger one. In addition to the mAP, we report the clip mean accuracy computed by majority voting over the predictions for all frames. Having a clip-level metric allows us to compare with LaViLa, considering the results observed in Sect. 3 when this model is applied using a single clip to simulate the online setting. MiniROAD is outperformed by LaViLa, which uses a larger backbone and can rely on a strong pre-trained representation. Nevertheless, MiniROAD proves to be competitive also on egocentric video.

**Table 5. Results on EGTEA.**

| | mAP (%) | mean Acc. (%) |
|---|---|---|
| MiniROAD | 32.1 | 43.0 |
| MiniROAD-L | 36.3 | 52.3 |
| LaViLa | – | 59.2 |

**Table 6. Results on ClipOffice.**

| | mAP (%) | mean Acc. (%) |
|---|---|---|
| MiniROAD-L | 76.2 | 77.3 |
| LaViLa | – | 79.2 |

**EgoClip Office.** We randomly generate a testing split from EgoClip Office using 12.5% of the data and leave the rest of the data for training. We maintain the larger classification network that proved more effective on EGTEA and train the model on sequences of 128 frames for 50 epochs, optimizing a weighted cross-entropy loss with weights equal to the inverse class frequencies. In order to compare the modified MiniROAD with LaViLa, this time we train a 15-dimensional classifier from scratch on top of LaViLa using the same TSF-B backbone, and we follow the implementation of [34] for fine-tuning on downstream datasets.

Results are shown in Table 6, using the same metrics as above. Similarly to the previous scenario LaViLa outperforms the modified MiniROAD, but this time by a much smaller margin. Moreover, both models achieve higher accuracy on this dataset compared to EGTEA. This suggests that our EgoClip office is less challenging than EGTEA, and in this scenario a lighter model like MiniROAD can close the gap from stronger and larger models, and perform well on first-person video.

## 7 Deployment on Nvidia Jetson

Given our efforts to identify an algorithm that can be applied in online fashion while adapting to the constraints of a low-power device, we finally deploy the modified models on an Nvidia Jetson Orin Nano, representing a resource-constrained embedded platform. We implement a pipeline that receives frames in real time and gathers them into a buffer, allowing the models to consume them according to their execution speed. We adopt a sliding window mechanism such that when the model is ready to process new data, it retrieves the most recent frames from the buffer to form an input clip. Processing only the most recent frames and discarding the older ones is a practical solution to avoid accumulating delay when using computationally demanding models such as LaViLa. This approach is reasonable in the context of human action recognition when the granularity of actions is not too fine and they extend over several seconds, so that skipping few frames does not pose significant performance issues. Timing measurements of the adapted algorithms are reported in Table 7. On the selected hardware, we run LaViLa at 10 Hz, where each execution loads a clip of 16 frames from the buffer. On the other hand, the modified MiniROAD runs above the real-time frequency 30 Hz, confirming the expected trade-off between this smaller model and LaViLa. It should also be remembered that the proposed architecture for MiniROAD, including feature extractors and classification head,

**Table 7. Inference frequency on the NVIDIA Jetson Orin Nano.**

| | Inference Frequency (Hz) |
|---|---|
| MiniROAD-L | 34.42 |
| LaViLa | 9.74 |

comprises approximately 60M parameters, while the TSF-B backbone adopted by LaViLa uses twice as many, 121M parameters. This becomes a relevant matter when moving towards smaller embedded devices, where memory size may be strongly limited.

## 8 Conclusions

In this work, we addressed the issue of bringing egocentric action recognition to smart eyewear system, operating online on embedded, low-power hardware. We began by assessing the feasibility of applying the state-of-the-art LaViLa model in an online fashion, using less data at inference time to reduce recognition latency and remove the need to store long clips. Then, we focused on optimizing MiniROAD to work in computationally constrained settings. In particular, we adopted a lightweight extractor of optical flow features by leveraging the encoder of Light Flow, balancing accuracy and inference speed. Next, we validated for the first time the MiniROAD model on two egocentric datasets, including EgoClip Office, a custom dataset that we constructed as a restriction of the larger EgoClip, focused on the office scenario. The procedure that we described can be used to generate datasets suitable for traditional action recognition from the large-scale EgoClip dataset. Finally, we deployed our modified models on an Nvidia Jetson Orin Nano, showing that MiniROAD and LaViLa can be brought to work in real-time or quasi-real-time on embedded hardware.

**Acknowledgments.** This paper was carried out in the EssilorLuxottica Smart Eyewear Lab, a Joint Research Center between EssilorLuxottica and Politecnico di Milano.

## References

1. An, J., Kang, H., Han, S.H., Yang, M.H., Kim, S.J.: Miniroad: minimal rnn framework for online action detection. In: International Conference on Computer Vision (ICCV) (2023)
2. An, J., Park, Y., Kang, H., Kim, S.J.: Object aware egocentric online action detection (2024). https://arxiv.org/abs/2406.01079
3. Ashutosh, K., Girdhar, R., Torresani, L., Grauman, K.: Hiervl: learning hierarchical video-language embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 23066–23078 (June 2023)

4. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, 18–24 Jul, vol. 139, pp. 813–824. PMLR (2021), https://proceedings.mlr.press/v139/bertasius21a.html
5. Cho, K., et al.: Learning phrase representations using rnn encoder-decoder for statistical machine translation (2014). https://arxiv.org/abs/1406.1078
6. Damen, D., et al.: Scaling egocentric vision: the dataset. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11208, pp. 753–771. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01225-0_44
7. Damen, D., et al.: Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. Inter. J. Compute. Vis. (IJCV) **130**, 33-55 (2022). https://doi.org/10.1007/s11263-021-01531-2
8. De Geest, R., Gavves, E., Ghodrati, A., Li, Z., Snoek, C., Tuytelaars, T.: Online action detection. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 269–284. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_17
9. Eun, H., Moon, J., Park, J., Jung, C., Kim, C.: Learning to discriminate information for online action detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 806–815 (2020). https://doi.org/10.1109/CVPR42600.2020.00089
10. Gao, J., Yang, Z., Nevatia, R.: Red: reinforced encoder-decoder networks for action anticipation (2017). https://arxiv.org/abs/1707.04818
11. Geest, R.D., Tuytelaars, T.: Modeling temporal structure with lstm for online action detection. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1549–1557 (2018), https://api.semanticscholar.org/CorpusID:21788692
12. Grauman, K., et al.: Ego4d: around the world in 3,000 hours of egocentric video. In: IEEE/CVF Computer Vision and Pattern Recognition (CVPR) (2022)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
14. Hu, X., Dai, J., Li, M., Peng, C., Li, Y., Du, S.: Online human action detection and anticipation in videos: a survey. Neurocomputing **491**, 395–413 (2022). https://doi.org/10.1016/j.neucom.2022.03.069, https://www.sciencedirect.com/science/article/pii/S0925231222003617
15. Idrees, H., et al.: The thumos challenge on action recognition for videos "in the wild". Comput. Vis. Image Understanding **155**, 1–23 (2017). https://doi.org/10.1016/j.cviu.2016.10.018, https://www.sciencedirect.com/science/article/pii/S1077314216301710
16. Kim, Y.H., Nam, S., Kim, S.J.: Temporally smooth online action detection using cycle-consistent future anticipation. Pattern Recogn. **116**, 107954 (2021). https://doi.org/10.1016/j.patcog.2021.107954, https://www.sciencedirect.com/science/article/pii/S0031320321001412
17. Li, Y., Nagarajan, T., Xiong, B., Grauman, K.: Ego-exo: transferring visual representations from third-person to first-person videos. In: CVPR (2021)
18. Li, Y., Liu, M., Rehg, J.M.: In the eye of beholder: joint learning of gaze and actions in first person video. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11209, pp. 639–655. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01228-1_38

19. Lin, K.Q., et al.: Egocentric video-language pretraining. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022). https://openreview.net/forum?id=nE8_DvxAqAB
20. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell. **42**(2), 318–327 (2020). https://doi.org/10.1109/TPAMI.2018.2858826
21. Núñez-Marcos, A., Azkune, G., Arganda-Carreras, I.: Egocentric vision-based action recognition: A survey. Neurocomputing **472**, 175-197 (2022). https://doi.org/10.1016/j.neucom.2021.11.081
22. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding (2019). https://arxiv.org/abs/1807.03748
23. Pramanick, S., et al.: Egovlpv2: egocentric video-language pre-training with fusion in the backbone. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 5285–5297 (October 2023)
24. Radford, A., et al.: Learning transferable visual models from natural language supervision (2021). https://arxiv.org/abs/2103.00020
25. Ragusa, F., Furnari, A., Livatino, S., Farinella, G.M.: The meccano dataset: understanding human-object interactions from egocentric videos in an industrial-like domain. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 1569–1578 (January 2021)
26. Sigurdsson, G.A., Gupta, A., Schmid, C., Farhadi, A., Alahari, K.: Charades-ego: a large-scale dataset of paired third and first person videos (2018). https://arxiv.org/abs/1804.09626
27. Truong, T.D., Luu, K.: Cross-view action recognition understanding from exocentric to egocentric perspective (arXiv:2305.15699) (May 2023). https://doi.org/10.48550/arXiv.2305.15699
28. Wang, L., et al.: Temporal segment networks: towards good practices for deep action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_2
29. Wang, S., Li, Z., Zhao, Y., Xiong, Y., Wang, L., Lin, D.: denseflow (2020). https://github.com/open-mmlab/denseflow
30. Wang, X., et al.: Oadtr: online action detection with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 7565–7575 (October 2021)
31. Xu, L., et al.: Towards continual egocentric activity recognition: A multi-modal egocentric activity dataset for continual learning. IEEE Trans. Multimedia **26**, 2430–2443 (2024). https://doi.org/10.1109/TMM.2023.3295899
32. Xu, M., Gao, M., Chen, Y.T., Davis, L.S., Crandall, D.J.: Temporal recurrent networks for online action detection. In: IEEE International Conference on Computer Vision (ICCV) (2019)
33. Xu, M., et al.: Long short-term transformer for online action detection. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems, vol. 34, pp. 1086–1099. Curran Associates, Inc. (2021), https://proceedings.neurips.cc/paper_files/paper/2021/file/08b255a5d42b89b0585260b6f2360bdd-Paper.pdf
34. Xue, Z., Grauman, K.: Learning fine-grained view-invariant representations from unpaired ego-exo videos via temporal alignment. In: NeurIPS (2023)
35. Yang, L., Huang, Y., Sugano, Y., Sato, Y.: Interact before align: leveraging cross-modal knowledge for domain adaptive action recognition. In: 2022 IEEE/CVF

Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14702–14712 (2022). https://doi.org/10.1109/CVPR52688.2022.01431

36. Yonetani, R., Kitani, K.M., Sato, Y.: Ego-surfing first person videos. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), pp. 5445–5454 (2015). https://doi.org/10.1109/CVPR.2015.7299183
37. Zhao, Y., Krähenbühl, P.: Real-time online video detection with temporal smoothing transformers. In: Computer Vision - ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXIV. p. 485-502. Springer-Verlag, Berlin, Heidelberg (2022). https://doi.org/10.1007/978-3-031-19830-4_28
38. Zhao, Y., Misra, I., Krähenbühl, P., Girdhar, R.: Learning video representations from large language models. In: CVPR (2023)
39. Zhu, X., Dai, J., Yuan, L., Wei, Y.: Towards high performance video object detection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7210–7218 (2018). https://doi.org/10.1109/CVPR.2018.00753