Multimodal Egocentric Action Recognition

Manuel Escobar
Politecnico di Torino
s307729@studenti.polito.it

Pablo Munoz
Politecnico di Torino
s310308@studenti.polito.it

Santiago Romero
Politecnico di Torino
s314233@studenti.polito.it

Abstract

This project focuses on Multimodal Egocentric Action Recognition (MEAR) using two popular datasets: Epic-Kitchens and ActionSense. The project explores the use of existing models trained for action recognition, such as I3D, and further analyzes the implementation of a classifier for first-person POV action recognition. Different modalities, such as RGB and EMG are used, demonstrating the improvement in performance of using EMG data. This showcases the potential of using multiple modalities in the context of Egocentric Action Recognition, specially with the extensive availability of sensors we have in our everyday lives. Our codebase and models/data is available at Github and Google Drive, respectively.

1. Introduction

The action recognition task on computer vision is related to the categorization and identification of human actions in a video sequence. Focus on this area [8, 11, 14] has emerged due to its application on many areas, such as security and behavioral analysis, and due to the significant advances in computer vision of the last decade.

Based on the current efforts on action recognition and the increasing availability of sensors that can be mounted on an actor (e.g. GoPro, Google Lenses, Meta Smart Glasses), a major sub-field has emerged called Egocentric Action Recognition (EAR), focusing on first-person point of view (POV) scenarios. Recent first-person video datasets [4, 10, 13] provide huge amounts of data to be used for training EAR models. Additionally to visual data, multi-modal approaches taking advantage of non-visual sensors, such as Electromyography (EMG) [10], can also be used.

Apart from classification, EAR can be used for several other tasks, such as R3D [16], which uses a pre-trained model on diverse human video data to learn robotic manipulation tasks. This model uses the Ego4D dataset [13], containing different modalities such as audio, IMU, 3D pose, different POVs, etc. Other applications, such as IMU2CLIP [15] and AUDIO2CLIP, use IMU sensors and audio to generate video and text explanations.

Therefore, the primary goal of our study is to get familiar with the EAR concept, datasets, and the subsequent implementation of EAR on two common datasets: Epic-Kitchens [4] and ActionSense [10]. Specifically, we intend to explore two modalities: RGB streams of first-person point of view (POV) videos and EMG data, which has not been explored extensively on this context.

Using RGB streams of data, previous studies have obtained remarkable results on the video action recognition task [8, 14, 16]. Although all of them use RGB frames, which is the main modality of videos and images, some of them use different modalities, such as audio, optical flow, and/or gaze. Another major modality is optical flow, with recent promising approaches that combine RGB, Flow, and audio [12].

As many different models, datasets, training parameters, and evaluation standards are used, it is difficult to establish general baseline for action recognition tasks. Nevertheless, works such as [1], give a clear idea of the best performing models and guidelines. For example, the performance of a model can be affected by important parameters such as the number of frames to be analysed per clip or the backbone used by the feature extraction model. Therefore, this study provides some insights related to the settings that will be used for the feature extraction and training of the classifier described for this project.

Based on this, we will use a pre-trained I3D model on the kinetics dataset [19] for action recognition, which uses an Inception backbone as in [8]. While recent advancements in action recognition have led to models achieving impressive accuracy [1, 3, 5, 7], I3D was chosen due to its popularity and extensive use. This model will be used as an RGB feature extractor for the Epic-Kitchens [4] and ActionSense [10] datasets. As of the classification task, different models will be implemented, analyzing their performances on the Epic-Kitchens [4] dataset. Then, we extended this classification to the ActionSense dataset [10], combining the previously explored model with a newly trained model for the EMG modality, demonstrating EMG data's usefulness on the EAR context.

Our main results can be summarized on the following:

- For the EPIC Kitchen dataset, we showed the benefit of using a pre-trained feature extractor and how a simple classifier is enough to applying it to another task.

- For the EPIC Kitchen dataset, we had an equal result with a Multi layer Preceptron and a transformer, as our best models. The MLP also had a better performance across different configurations, making it the overall best classifier.

- For the ActionSense dataset, the EMG modality demonstrated to be more important than the RGB stream by a significant margin. Nevertheless, the best results were obtained by using a multi-modal approach, combining the EMG and RGB streams of data using weighted late-fusion.

2. Related Work

In the domain of Action Recognition, recent advancements have been made through the introduction of innovative models and techniques that try to improve the understanding and categorization of human actions in video data, with the addition of other sensors for better results.

Action Recognition: One notable contribution is the Two-Stream Inflated 3D ConvNet (I3D), as discussed in ”Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset.” [8]. It serves both a baseline and classification algorithm, providing a solid foundation for further research in this field.

Complementing the pursuit of efficient video understanding is the Temporal Shift Module (TSM), proposed in ”Temporal Shift Module for Efficient Video Understanding.” [7] TSM offers a lightweight but powerful solution for video recognition, creating a combination of 3D and 2D CNNs to optimize performance without compromising computational efficiency.

Moreover, the field has been more extensively analyzed in ”Deep Analysis of CNN-based Spatio-temporal Representations for Action Recognition” [1]. By systematically analyzing various architectural components and training strategies, the paper presents feature extraction and representation learning guidelines, contributing significantly to the theoretical understanding and practical implementation of CNN-based approaches in action recognition tasks.

Multi-Modal Action Recognition: In the realm of Egocentric Action Recognition, several approaches have emerged to tackle the unique challenges posed by first-person perspective data.

EPIC-Fusion: Audio-Visual Temporal Binding for Egocentric Action Recognition [12] presents a novel architecture for multi-modal fusion specially cretated for EAR. By integrating RGB, Flow, and Audio modalities within a temporal binding window, this work addresses the temporal asynchrony between action appearance and audio cues, resulting on a increase on recognition accuracy. Moreover, the incorporation of extra data alongside traditional RGB inputs showcases the potential for further improvement in action recognition capabilities.

Multi-Modal Domain Adaptation: A general problem of action recognition tasks is the domain shift between seen and unseen environments. Previous studies have shown a strong dependence on the environment, causing significant performance decrements in unseen environments. In order to solve this issue, notable works such as Multi-Modal Domain Adaptation for Fine-Grained Action Recognition [9] tackle the environmental bias in fine-grained action recognition datasets and proposes a multi-modal domain adaptation approach utilizing multi-modal self-supervision and adversarial training per modality.

Egocentric Action Recognition: Different studies have been done on the EAR task, mainly using Epic-Kitchens. Methods like E2(G0)MOTION: Motion Augmented Event Stream for Egocentric Action Recognition [2], analyze the use of other modalities such as event-based cameras. This method argues that event data is a valuable modality for egocentric action recognition, as it has advantages over traditional RGB and optical flow modalities, like low latency, high temporal resolution, low power consumption, and robustness to environmental bias.

In summary, recent research efforts in Action Recognition have led to the development of sophisticated models and techniques created to address the diverse challenges present in this field, ranging from efficient video understanding to domain adaptation and egocentric action recognition. These advancements pave the way for further exploration and innovation in understanding human actions in video data.

3. EPIC-Kitchens Experiments and Results

3.1. Datasets

Epic-Kitchens: EPIC-Kitchens [4] is a large-scale egocentric video dataset that was recorded by 32 participants in different kitchen environments. The videos consist of nonscripted daily activities recorded every time they entered their kitchen. The dataset features 55 hours of video consisting of 11.5 million frames, which were densely labeled for a total of 39.6 thousand action segments and 454.3 thousand object bounding boxes. The dataset is annotated with 97 verb classes (‘put’, ‘wash’, ‘open’, ‘cut’, etc.) and 300 noun classes (‘plate’, ‘knife’, ‘water’, ‘onion’, etc.). Due to limited computational capabilities, for our experiment we took RGB frames for a subset of EPIC-Kitchens-55, concentrating only on subject P08. This subject only performed

Sampling	Average ARI	Average NMI
Dense	0.252	0.331
Uniform	0.233	0.318

Table 1. Average Adjusted Rand Index and Normalized Mutual Information for different sampling methods

activities in one kitchen (D1), and from the whole set of possible verbs (97), he only recorded 8 of them, which are the ones that we will aim to classify.

3.2. Feature Extraction

We first perform a feature extraction on the RGB stream of data, which consisted of extracting a compressed representation of the frames. We exploited the pre-trained I3D network model [8] with an Inception-V1 [17] backbone. This model was used as a feature extractor, loaded with the weight values of an I3D model trained over the Kinetics Action Recognition Dataset [19], and then fine-tuned to work on the EPIC-Kitchens dataset.

The feature extractor takes as input a sample of each activity. This sample consists of taking the frames corresponding to that activity, dividing them in a number of clips, and extracting K frames from each clip. Therefore the input representation has size (number of clips, number of frames per clip, 3, 224, 224). The clips were sampled using either dense or uniform techniques. Dense sampling takes a center frame, and a number K of adjacent frames with a stride S between them. The uniform sampling takes K evenly spaced frames throughout the clip.

We then extract the features by passing the input clips through the I3D architecture, and extracting the values just before the layers corresponding to the classifier. This extracts a smaller representation of the clips, having output of N x O values, where N represent the number of clips per record (5), and O is the output layer size (1024).

We analysed this features using traditional clustering algorithms to see if we could find a correlation of the data extracted and the defined clusters. For this, we averaged the features over the temporal axis of the clips, and then we applied the k-means algorithm. The algorithm searched for 8 clusters (each one corresponding to a verb), and then computed the results. We measured the effectiveness of the model based on two accuracy metrics which measure the similarity between two clusters, the Adjusted Rand Index (ARI) ranging from -1 to 1, and the Normalized Mutual Information (NMI) that ranges from 0 to 1.

The extracted feature clusters had an average ARI of 0.233 when sampled uniformly, and 0.252 when sampled densely. It also had an average NMI of 0.318 and 0.331 for uniform and dense sampling respectively. The ARI values indicates that the clusters, compared to the ground truth, are almost as if selected at random. For the NMI, the resulting values indicate that there exists some association within the generated clusters and the ground truth. By this analysis we observed that the dense sampling method tends to give a better clustering result than the uniform one. Additionally, the higher the number of frames, the higher the ARI and NMI value for the generated clusters.

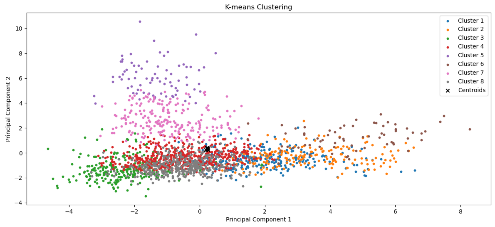


Figure 1. Clusters obtained from the extracted features on the Epic-Kitchens dataset. This clustering was obtained with 10 frames per clip and dense sampling, as it is the configuration whose ARI and NMI are closer to the average of the metrics. A PCA dimensional reduction was performed for the visualization of the clusters.

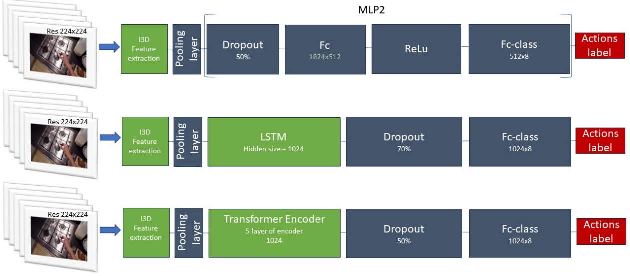


Figure 2. Architectures implemented for the Epic-Kitchens dataset, including MLP, LSTM and a Transformer. This models are used individually and the change on dimensions on each path is presented on the blocks.

3.3. Classifier

We tested four different classifiers for the EAR task: MLP, LSTM, I3D, and a Transformer. The I3D classifier consists of the classifier architecture used on the original paper on action recognition. A grid search approach was used for each model, testing 6 different scenarios, consisting of a combination of the frames per clip (5, 10 and 25) and the type of sampling (dense or uniform). Therefore, a total of 24 classifiers were trained. The architectures of this models are presented on Fig. 2.

Multi layer Perceptron (MLP): This classifier receives as input the saved features of shape (batch size, 1024), passes this data through a dropout layer with a drop probability of 0.7. This data is then processed by a fully connected layer of input 1024 and output 512 with a ReLU activation layer. Finally it goes through another fully connected layer with an output of the size of the number of classes.

Long Short Term Memory (LSTM) [6]: A LSTM model was used, taking as input (batch size, sequence length, 1024), where sequence length corresponds to the number of clips per record. The temporal progression corresponds to each one of the clips and the output corresponds to the last temporal output after passing all the clips through the network. The LSTM network has a hidden size of 1024 and is followed by a dropout of 0.7 and a fully connected layer used to get the final prediction.

Transformer Network [18]: We applied just the encoder portion of the Transformer Network, with an input of (batch size, N, 1024), and with a feed-forward layer of size 2048. We trained an encoder transformer with 5 layers of encoders. This output is then passed through a dropout layer of 0.5 and finally a fully connected layer with an output size of the number of classes, used to output the prediction.

3.4. Training Parameters

The training of each classifier was done with the same training parameters. It was trained with a learning rate of 0.01 which decayed every 3000 steps by a factor of 10, a Stochastic Gradient Descent optimizer with a momentum of 0.9, and a weight decay of 10^{-7}. A cross entropy loss is used, computing the loss per clip (N = 1) for training and the loss of the average of the predictions of 5 clips (N = 5) for the validation. The input data was fed to the model in batches of 32 samples.

3.5. Results

Using this four classifiers, we obtained the accuracy seen on Fig. 3. We see that the best classifier is MLP with 10 and 25 frames per clip and a uniform sampling. The Transformer encoder also obtained the same result when trained with 25 frames. When analysing the results, it was observed that the accuracy difference related to the sampling method and number of frames was at most 0.23% and 2.07% respectively. Based on this, we see that a change on the number of frames can have up to ten times the impact on the accuracy than the one obtained by changing the sampling method. We can also see that each model tends to have a slightly better performance using a specific sampling method, as seen on Tab. 2.

4. ActionSense Experiments and Results

4.1. Dataset

ActionSense: The ActionSense dataset [10] is a comprehensive multimodal framework designed for wearable sensing in a kitchen setting. It captures a wide array of information, including motion, force, and attention data, using an eye tracker, muscle activity sensors, a body-tracking system, finger-tracking gloves, and tactile sensors on the hands.

Model	Num. Frames	Accuracy
MLP (U)	5	57.47%
MLP (U)	10	**59.54%**
MLP (U)	25	**59.54%**
I3D (U)	5	55.86%
I3D (U)	10	58.62%
I3D (D)	25	59.31%
LSTM (U)	5	57.24%
LSTM (D)	10	59.08%
LSTM (D)	25	58.16%
Transformer (D)	5	52.87%
Transformer (D)	10	57.70%
Transformer (D)	25	**59.54%**

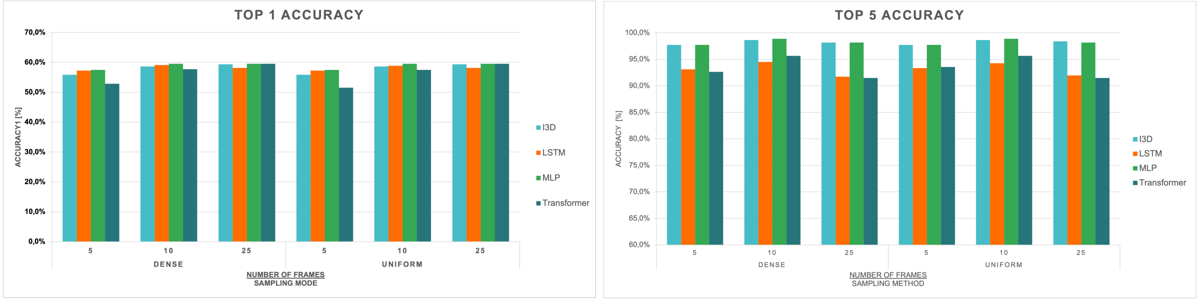
Table 2. Results of the RGB classifiers. Only the best top 1 accuracy per model is reported. We also specify the sampling methods as (D)ense or (U)niform.

The data is complemented by activity labels and externally-captured data from multiple cameras and microphones. The tasks recorded a range from simple object manipulations to complex action sequences, designed to underscore both physical skills and scene reasoning. The dataset offers synchronized labels for 20 distinct activities, with 64.9% of the data having verified labels inputted in real time. The specific tasks recorded in ActionNet are designed to highlight lower-level physical skills and higher-level scene reasoning or action planning. They include simple object manipulations (e.g., stacking plates), dexterous actions (e.g., peeling or cutting vegetables), and complex action sequences (e.g., setting a table or loading a dishwasher).

In our experiments, data from nine subjects is used for the Electromyography (EMG) modality, that give us information on 8 channel measurements for each hand (16 total channels). However, due to our computation capabilities and extensive amount of data contained in the RGB modality, only RGB data from subject S04 is used. The video downloaded from ActionSense was used for extracting 108,771 RGB frames with a frequency of 30 frames per second and a resolution of 456x256 pixels.

Using just S04 data is problematic, as in the training split we have only frames related to 19 of the 20 possible actions. Additionally, the test split used only contains 7 classes. Therefore, it is highly recommended to use more data for training the RGB model.

Preprocessing: The EMG data from ActionSense was



(a) Top 1 Accuracy

(b) Top 5 Accuracy

Figure 3. Accuracy top 1 **(left)** and Accuracy top 5 **(right)**. On the x-axis we compare the sampling types **(bottom)**, and the number of frames **(top)**, for each of the models trained.

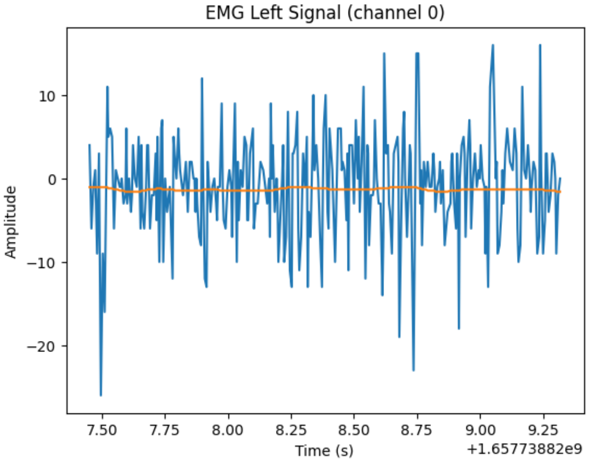


Figure 4. Comparison between pre-processed data and original data for myo left readings of channel 0, taking only the first 100 timestamps of the first activity.

recorded by 8 electrodes, which each register the muscle activation of certain parts of the forearm. With that, we finish with a total of two sets of vectors of 8 values, one per arm (16 values in total). An activity consists of a list of this measurements at different timesteps during the action.

For this project, the same pre-processing as in the ActionSense [10] documentation was used, which consists of three pre-processing steps: rectification, a low-pass filter with a cutting frequency of 5 Hz, and then normalizing the data to [-1, 1]. This pre-processing is of great importance in order to smooth the data, getting rid of noise or any phenomena that can be caused by the sensors, as well as to preserve relative magnitude across locations of the forearm. The result of this transformation can be seen on Fig. 4

Data Augmentation: As there are a small number of annotations on the dataset and some activities take much more time than others, we needed to augment the data so that the model has more annotations to train/test. For this, we implemented two strategies that cut the actions on smaller clips. The first strategy divides the total time of an action into clips of a given duration D. No fixed number of clips is given, so ultimately only large actions are going to be divided. The second strategy splits the entire activity into C clips of duration D. Therefore, we obtain C new samples for each previous record. We expect the later method to be better, as it generates the same number of samples for each activity, which does not modify the previous class distributions. In the first case, only long actions are augmented.

RGB Feature Extraction As done on Sec. 3.2 with the Epic-Kitchens dataset, we performed a feature extraction on the RGB stream of data of S04 first-person video without gaze. We utilized a pre-trained I3D network model with an Inception-V1 backbone [8], with the same parameters as before.

We then analyzed these features using traditional clustering algorithms. For this we fistly joined the testing and training features for its analysis. Specifically, we applied the k-means algorithm and hierarchical clustering to the mean of the extracted features over the temporal axis for each clip.

The algorithm was designed to search for 19 clusters, each corresponding to a specific action, and subsequently computed the results. Upon testing, the calculated clusters had an average ARI of 0.187 and an average NMI of 0.415.

The ARI values suggest that the clusters, when compared to the ground truth, means that there is a slight similarity between the data clusters, but in the end values are almost random. The resulting NMI means that there is a moderate correlation between the two variables. The closer the NMI is to 1, the stronger the correlation between the variables.

4.2. Architecture

EMG: For the model, we based ourselves again on the Action Sense documentation, which describes a neural network architecture for multi-modal EAR using LSTMs. The network used for this study consists of only the EMG

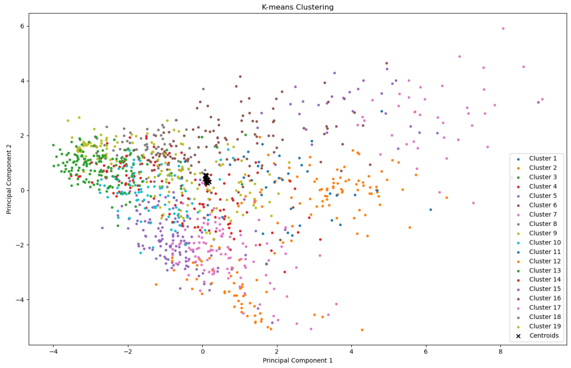


Figure 5. Clusters obtained from the extracted features on the ActionSense dataset on subject S04. This clustering was obtained with 10 frames per clip and dense sampling, as it is the configuration whose ARI and NMI are closer to the average of the metrics. A PCA dimensional reduction was performed for the visualization of the clusters.

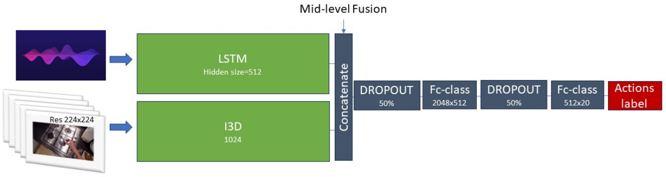


Figure 6. Architecture implemented on the ActionSense dataset, representing mid-fusion between LSTM and I3D classifiers. The change of dimensions of data in each step is represented inside each block.

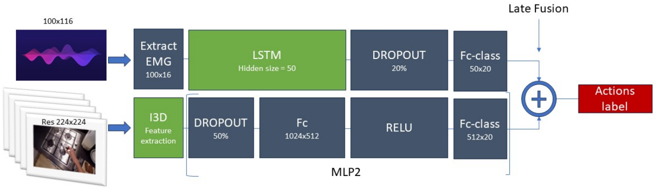


Figure 7. Architecture implemented on the ActionSense dataset, representing late-fusion between LSTM and I3D classifiers. The change of dimensions of data in each step is represented inside each block.

branch of the architecture proposed. The LSTM, with hidden size 50, receives an input of size (batch size, 100, 16), where 100 corresponds to the number of readings taken from the sample and 16 corresponds to the number of measurement channels. The selection of the readings from each sample (100) is done using the previously described uniform and dense sampling techniques. This values then go through a dropout layer with a drop probability of 0.2, and a fully connected layer with output size 20.

RGB: For the RGB classification, we use the exact same model configuration as the MLP classifier described on Sec. 3.3. We selected this over the transformer as both the models had the same top 1 accuracy, but the MLP showed a better result across all the trained variations, and it carries a lower computational cost (over the Transformer).

Mid-level Fusion: Another model was created for performing mid-level fusion of EMG and RGB streams. For this model, the EMG architecture is changed. The EMG model consists of a LSTM with hidden size 1024, so that the number of features corresponds to the RGB features size. Once both features are extracted, a dropout layer of 0.5 is used, followed by a fully connected layer of size (1024 + 1024, 512). Finally, it passes through another 0.5 dropout and a final fully connected layer of size (512, 20).

Late-fusion: Another approach was late fusion, which joins the RGB and EMG streams at the end of the classifciation task. In this type of fusion, both streams of data are processed independently from one another and their logits are added, resulting in a combined prediction. Nevertheless, it was found that a weighted late-fusion obtained significant performance improvements, assigning a greater weight to the EMG stream. Therefore, the final weighted late-fusion assings a weight of 0.8 to the EMG logits and a weight of 0.2 to the RGB logits.

4.3. Training Parameters

We trained all the models first up to 3000 epochs, with a learning rate decay step of 1750 epochs. Then, we picked the best model at the end of the 3000 epochs, and took it further to a training of 5000 epochs. The initial learning rate of the EMG was of 0.1. All the other training parameters where the same as in Sec. 3.4.

4.4. Results

Initially, we trained a total of 12 classifiers for the EMG modality using a grid search approach. For each model, we changed the sampling method (uniform or dense with stride 1 or 2), the data augmentation method (Fixed and not fixed number of clips), and the duration of the new clips (5

Data Augmentation	Approach	Sampling	Best Top-1
10 sec x 20 clips	LSTM	Dense	47.54%
	LSTM	**Uniform**	**51.44%**
	LSTM	Dense (Str 2)	43.98%
5 sec x 20 clips	LSTM	Dense	44.92%
	LSTM	Uniform	47.46%
	LSTM	Dense (Str 2)	49.32%
Fixed 5 sec	LSTM	Dense	32.60%
	LSTM	Uniform	
	LSTM	Dense (Str 2)	35.36%
Fixed 10 sec	LSTM	Dense	36.57%
	LSTM	Uniform	33.14%
	LSTM	Dense (Str 2)	36.00%

Table 3. Results of the EMG classifier. We reported the Augmentation method, the approach of the classifier, the sampling method and the best Top-1 accuracy.

Domain	Accuracy
S00-S09 → S00-S09	53.56%
S00-S09 → S04	61.88%
S04 → S00-S09	25.68%
S04 → S04	61.25%

Table 4. Accuracy of the classifiers for EMG trained on different data domains and their accuracy on a target domain. The first domain reports the domain on which the classifier was trained (source domain) and the domain represents the target domain.

Modality	Top-1	Top-5
RGB (S04)	42.50%	74.38%
EMG (ALL)	53.56%	**90.17%**
RGB (S04) + EMG (ALL) †	**66.25%**	87.50%
RGB (S04) + EMG (S04) ‡	57.50%	86.25%

Table 5. Final classifier accuracy on the ActionSense dataset, validated on S04. †: late fusion and weights [0.2, 0.8]. ‡: mid-level fusion

and 10 seconds). We can see the resulting accuracy of all the classifiers trained in Tab. 3. The best combination was obtained with 20 fixed clips, a duration of 10 seconds, and uniform sampling.

As observed in Tab. 4, the domain on which the classifier is trained is of great importance. In this case the domain shift is not represented by the kitchen. Instead, the domain shift occurs because of the different subject's EMG measurements. We can see that models trained on a single subject do not generalize well for unseen subjects. Another important thing to notice is the advantage of training over a variety of subjects, where the model being trained on all subjects obtained better results on S04 than the one trained only on S04. This can also be caused due to the quantity of data on which it is trained but it is generally advisable to use different subjects.

With the EMG classifier trained over all subjects, the RGB classifier was then trained over S04 samples. When comparing the individual performance of the EMG and RGB classifiers, we see that they obtained an accuracy of 61.88% and 42.50%, respectively. It can be observed that the accuracy of the EMG modality surpasses RGB by a big margin. This demonstrates than EMG can be a stronger modality than RGB for EAR, although further analysis must be made to train RGB on all subjects.

Finally, two different multi-modal approaches were used. The first strategy, which uses mid-level fusion, was found to be less accurate than late-fusion, obtaining accuracies of 57.5% and 66.25%, respectively. For late-fusion, a weighted approach was used, as the approach with equal weights between RGB and EMG logits obtained poor results (worse than just using EMG). Nevertheless, when a weight of 0.8 is assigned to EMG and 0.2 to RGB, the model obtains the best results of all of them, benefiting from a 4.63% improvement. This weights were assigned by trail and testing. The comparison of all the accuracy of the best models per method can be seen on Tab. 5.

5. Conclusion

We have shown the difference between using different training configurations for the EAR task, specifically the use of different classifiers, sampling techniques, and frame numbers for the Epic-Kitchens dataset. The use of a pre-trained feature extractor was also analyzed, demonstrating it's usefulness for other tasks, effectively adapting it from

Action Recognition to Egocentric Action Recognition.

Subsequently, the EAR task for Epic-Kitchens was extended to the ActionSense dataset, where we used a multimodal approach. It was shown that the EMG data provided significant performance improvements compared to RGB. Additionally, the late-fusion approach was proven better than the mid-level approach possibly due to the fact that the mid-level is only trained over S04. Nevertheless, using the two modalities was shown to be beneficial and increased the performance of the task.

Further works can use the entire ActionSense dataset to train the EAR classifiers, as well as using more modalities available on both datasets, such as audio, gaze, and/or tactile.

References

[1] Panda R. Ramakrishnan K. Feris R. Cohn J. Oliva A. Fan Q Chen, C.-F. Deep analysis of cnn-based spatio-temporal representations for action recognition. *CVPR*, 2021. 1, 2

[2] Gabriele Goletto Marco Cannici Emanuele Gusso Matteo Matteucci Barbara Caputo Chiara Plizzari, Mirco Planamente. Motion augmented event stream for egocentric action recognition. *CVPR*, 2021. 2

[3] Jitendra Malik Christoph Feichtenhofer, Haoqi Fan and Kaiming He. Slowfast networks for video recognition. *arXiv*, 2018. 1

[4] Sanja Fidler Antonino Furnari Evangelos Kazakos Davide Moltisanti Jonathan Munro Toby Perrett Will Price Michael Wra Doughty, Giovanni Maria Farinella. Scaling egocentric vision: The epic-kitchens dataset - dima damen, hazel. *ECCV*, 2018. 1, 2

[5] Lorenzo Torresani Heng Wang, Du Tran and Matt Feiszli. Video Modeling With Correlation Networks. Conference on Computer Vision and Pattern Recognition. *CVPR*, 2020. 1

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997. 4

[7] Song Han Ji Lin, Chuang Gan. Temporal shift module for efficient video understanding. *ICCV*, 2019. 1, 2

[8] Andrew Zisserman Joao Carreira. Quo vadis, action recognition? a new model and the kinetics dataset. *CVPR*, 2017. 1, 2, 3, 5

[9] Dima Damen Jonathan Munro. Multi-modal domain adaptation for fine-grained action recognition. *CVPR*, 2020. 2

[10] Yiyue Luo Michael Foshey Yunzhu Li Antonio Torralba Wojciech Matusik Joseph DelPreto, Chao Liu and Daniela Rus. Actionsense: A multimodal dataset and recording framework for human activities using wearable sensors in a kitchen environment. 2022. 1, 4, 5

[11] Andrew Zisserman Karen Simonyan. Two-stream convolutional networks for action recognition in videos. page 568–576, 2014. 1

[12] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. *ICCV*, 2019. 1, 2

[13] Eugene Byrne Zachary Chavis Antonino Furnari Rohit Girdhar Jackson Hamburger Hao Jiang Miao Liu Xingyu Liu Kristen Grauman, Andrew Westbury. Ego4d: Around the world in 3,000 hours of egocentric video. *CVPR*, 2022. 1

[14] Yuanjun Xiong Limin Wang, Yu Qiao Zhe Wang, Xiaoou Tang Dahua Lin, and Luc Van Gool. Temporal segment networks for action recognition in videos. *ECCV*, 2016. 1

[15] Zhaojiang Lin Alireza Dirafzoon Aparajita Saraf Amy Bearman Babak Damavandi Seungwhan Moon, Andrea Madotto. Imu2clip: Multimodal contrastive learning for imu motion sensors from egocentric videos and text narrations. *cs.CV*, 2022. 1

[16] Vikash Kumar Chelsea Finn Abhinav Gupta Suraj Nair, Aravind Rajeswaran. A universal visual representation for robot manipulation. 2022. 1

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. 3

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 4

[19] K. Simonyan B. Zhang C. Hillier S. Vijayanarasimhan F. Viola T. Green T. Back P. Natsev M. Suleyman W. Kay, J. Carreira and A. Zisserman. . The kinetics human action video dataset. 2017. 1, 3