

Multimodal Egocentric Action Recognition

TAs: Gabriele Goletto (gabriele.goletto@polito.it)
Simone Alberto Peirone (simone.peirone@polito.it)
Link to this file: <https://shorturl.at/cxJU2>

CODE

(Please read carefully the README file present in the code and the comments that you will find in the code)

Slides: <https://shorturl.at/LRS27>

OVERVIEW



Figura 1: Some frames of egocentric videos taken from Ego4D [4], a massive-scale egocentric dataset of daily life activity

The primary goal of this project is to become familiar with the topic of egocentric vision and the application of the multimodal learning paradigm in this context. In the first part of the project, the student will specifically work with standard visual modalities, such as RGB streams, that have been recorded in first-person perspective and analyzed using a cutting-edge action recognition algorithm (Nair et al., 2022). Then, the student will be asked to investigate a new modality that has received little attention in the computer vision community but could play an important role in this context, i.e., ElectroMyoGraphy (EMG).

The overall project will be split in three main points: I) reading literature; II) coding; and III) variation.

I) **Reading literature.** Before beginning to implement code or run experiments, it is critical to become familiar with the literature on egocentric vision.

II) **Coding.** The student should be able to reuse and adapt existing pre-trained architectures for action recognition, as well as to design new signal reconstruction models (e.g., auto-encoder).

III) **Variation.** The student should start to analyze a new modality in three different directions: by designing an ad-hoc model for the action recognition task, implementing an auto-encoder to learn how to pass from one modality to another, and finally, by recording a new multi-modal dataset for a new task.

GOALS

1. Read all the materials provided in order to get familiar with the egocentric vision, multimodal learning and the common techniques to perform an action recognition;
2. Extract features for a specific dataset starting from a pretrained model. Then, the pre-extracted features are used to train a classifier and learn a specific task for FPAR (other secondary analysis could be appreciated in this section).
3. **[Only for AML students]** Implement *one* of the following variations:
 - a. The Myo Armband is a wearable device provided with eight electro-myographic electrodes, a 9-axis Inertial Measurement Unit, and a transmission module. Start to investigate this new sensor and the data that it records. Train a model to solve a gesture recognition task.
 - b. Design a model that can learn how to reconstruct EMG data from an RGB stream using an existing RGB representation, and then use it to augment an existing egocentric vision dataset where this modality is missing.

STEP 1 - Related works

Getting familiar with egocentric vision

Before starting it is mandatory to take time to familiarize yourself with FPAR, multimodal learning and other modalities. More in detail, read:

[in-depth reading]

- [EPIC-Fusion: Audio-Visual Temporal Binding for Egocentric Action Recognition - Kazakos, Evangelos and Nagrani, Arsha and Zisserman, Andrew and Damen, Dima, ICCV 2019. \[code\]\[project page\]](#)
- [Multi-Modal Domain Adaptation for Fine-Grained Action Recognition - Jonathan Munro, Dima Damen, CVPR 2020](#)

- [E\(GO\)^2MOTION: Motion Augmented Event Stream for Egocentric Action Recognition](#) - Chiara Plizzari, Mirco Planamente, Gabriele Goletto, Marco Cannici, Emanuele Gusso, Matteo Matteucci, Barbara Caputo, 2021
- [R3M: A Universal Visual Representation for Robot Manipulation](#) - Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, Abhinav Gupta 2022 [\[project page\]](#)
- [Deep Analysis of CNN-based Spatio-temporal Representations for Action Recognition](#) - Chun-Fu Chen, Rameswar Panda, Kandan Ramakrishnan, Rogerio Feris, John Cohn, Aude Oliva, Quanfu Fan CVPR2021

[quick reading¹]

- [Ego4d: Around the world in 3,000 hours of egocentric video.](#) - K. Grauman, et al. CVPR 2022.
- [Scaling Egocentric Vision: The EPIC-KITCHENS Dataset](#) - Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, Michael Wray ECCV 2018
- [Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset](#) - Joao Carreira, Andrew Zisserman, CVPR 2017
- [Temporal Shift Module for Efficient Video Understanding](#) - Ji Lin, Chuang Gan, Song Han, ICCV 2019
- [Learning Transferable Visual Models From Natural Language Supervision](#) Alec Radford et al., ICML, 2021
- [AudioCLIP: Extending CLIP to Image, Text and Audio](#)
- [IMU2CLIP: MULTIMODAL CONTRASTIVE LEARNING FOR IMU MOTION SENSORS FROM EGOCENTRIC VIDEOS AND TEXT NARRATIONS](#)

STEP 2 - How to exploit existing models? Working with features.

Videos are an ordered collection of frames. Unlike images, where a single sample may be sufficient to predict its content, videos incorporate a temporal dynamic that can not be ignored. Spatial and temporal information are complementary and crucial to understand the content of the video.

To provide both spatial and temporal information to the network, a subset of frames is selected from each sample. First, videos are divided in segments of fixed length, called *clips*, by randomly selecting the central point of each segment. Then, the frames of each clip are sampled using one of two different strategies. Dense sampling takes a number of adjacent frames, possibly spaced by a small stride, e.g. 1 or 2. Uniform sampling, on the other hand, selects a number of evenly spaced frames in the clip. The former focuses more on the appearance of the video as the

¹ (abstract, captions, conclusions, plus some other parts of the paper) just to understand the main aspects and ideas that have been introduced.

frames are close in time, while the latter captures frames that may be further apart, better highlighting the temporal dynamics.

However, processing long sequences of frames incurs high costs in terms of time and resources required. To address this issue, the student is asked to adopt a two phase strategy. First, the student should extract a compressed representation, the so-called *features*, of the samples in the dataset using the pretrained model provided and save them. These features are the output of the deep convolutional part of the network, also called *backbone*, and lie in a space which is smaller than the original input but encodes all the necessary information for the network to compute its predictions. Once the extraction is completed, the pretrained model can be discarded. Then, use the features to train a classifier for action recognition.

The sub-step to follow in this section are the following ones:

- 1) The student should start from the pretrained model provided and extract the intermediate features (saved in a .pth or .pkl file) for the rgb stream of two dataset (EPIC-Kitchens [Damen et al., 2018] and ActionNet [DelPreto, 2022]), just before the final fully connected layer (*make sure the network is in evaluation mode during this step*).
- 2) The student should implement a dataset class to work with the datasets provided, using both dense and uniform sampling to select a K-number of frames [5-10-25]. The labels are provided in a .pkl file together with the start and end timestamps of each action.
- 3) For each sample in the dataset, once the features produced by the network have been extracted (as per 2.1) the student must:
 - a) Try to cluster the extracted features using a standard algorithm, i.e. K-Means, and analyze the output. Do some interesting patterns emerge? *Suggestion: use the central frame of each sample to represent the points.*
 - b) Other analyses may be carried out at the student's discretion and would be appreciated.
- 4) Finally, the student should use the extracted features to train a classifier. Before feeding them into the classifier, they must be aggregated along the temporal axis using convolution or a pooling layer.

STEP 3.a - Action recognition through a new modality.



Figure 2. Myo Gesture Control Armband from Thalmic Labs

A Myo Gesture Control Armband from Thalmic Labs is worn on each forearm. It contains 8 differential pairs of dry EMG electrodes to detect muscle activity, an accelerometer, a gyroscope, and a magnetometer. It also fuses the IMU data to estimate forearm orientation, and classifies a set of five built-in gestures.

[Dataset] : ActionNet, a multimodal dataset and recording framework with an emphasis on wearable sensing in a kitchen environment. It is composed of rich, synchronized data streams along with ground truth data for FPAR tasks, and it offers the opportunity to learn how humans interact with the physical world during activities of daily living. The wearable sensing suite captures motion, force, and attention information; it includes eye tracking with a first-person camera, forearm muscle activity sensors, a body-tracking system using 17 inertial sensors, finger-tracking gloves, and custom tactile sensors on the hands that use a matrix of conductive threads. This is coupled with activity labels and with externally captured data from multiple RGB cameras, depth camera, and microphones.

The current dataset contains 10² subjects, and is actively growing with a target of containing approximately 25 subjects by the Fall of 2022. It currently spans approximately 778.0 minutes of recorded data, averaging 77.8 ± 16.4 minutes per subject. Approximately 543.5 minutes of that time is occupied by performing kitchen activities (55.6 ± 13.7 minutes per subject), while the remainder is occupied by calibration routines. The dataset provides synchronized labels as ground truth data, spanning 20 unique activities. Of the time spent performing activities, 64.9% of the data has ground-truth labels entered in real time during the experiment. This leaves approximately 19.5 minutes of unlabeled data per subject, or 0.98 minutes per activity per subject; this generally represents the time spent providing instructions between activities.

[Preprocessing]: The 8 channels of muscle activity recorded from each forearm are processed to highlight general muscle activation levels. Each channel is rectified by taking the absolute value, and then a low-pass filter with cutoff frequency 5 Hz is applied. All 8 channels from an armband are then jointly normalized and shifted to the range $[-1, 1]$ using the minimum and maximum values across all channels. This preserves relative magnitude comparisons across locations on the forearm. This process results in 8 channels of normalized data from each of the 2 arms.

The absolute value of EMG data across all 8 forearm channels are summed together in each timestep to indicate overall forearm activation; this provides an estimate of wrist stiffness, which is induced by activating the antagonistic muscle pairs, and grasp strength. The streams are then smoothed to focus on low-frequency signals on time scales comparable to slicing motions.

NOTE: the ActionNet dataset has a limited number of action annotations which span long temporal intervals. In order to have enough samples to train your models, you should augment the dataset by dividing each action in shorter segments (subactions) lasting only a few seconds.

² In the current dataset, data for one of the subjects only contains camera data due to an anomalous technical issue with the wearable streaming; this will be remedied shortly and updated on the repository, although it can still be useful for computer vision in the meantime. The remainder of the dataset statistics focus on the 9 subjects with complete data. Due to an issue with the Manus gloves, finger-tracking data is not reliable for the initial subjects. This will be remedied shortly.

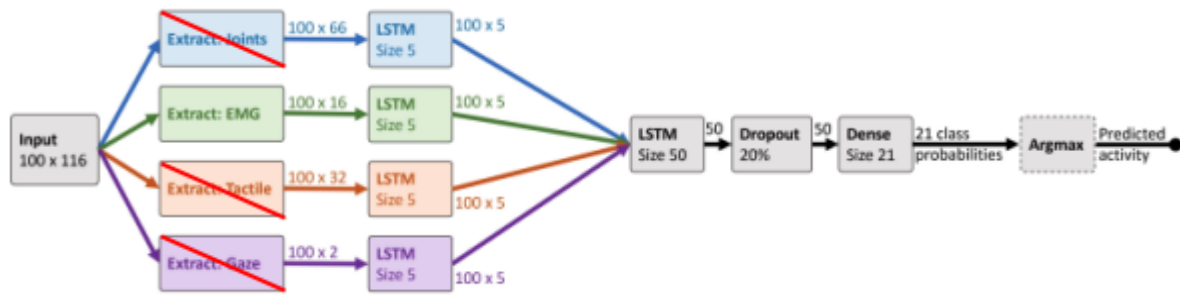


Figure 3. Overview of multi stream network used in ActionNet [3].

[Network]: The model should be able to take in input the EMGs (and/or IMU) data and learn a representation useful to classify the action.

The model is based on long short-term memory (LSTM) recurrent neural networks. Since LSTMs have feedback connections to process sequences of data, they are well-suited to the task of classifying the segments of wearable data sequences. Future pipelines could explore alternative structures, such as convolutional approaches. The network is summarized in Figure X. The first portion consists of parallel pathways that each process a single sensor modality (you should focus only on a single branch). Each one consists of a single LSTM layer that outputs a sequence matrix of size 100×5 . These outputs are concatenated and passed to an LSTM layer that outputs a vector of size 50. This is followed by a 20% dropout layer, and a dense output layer with softmax activations. The output has 21 classes: the 20 activities and a baseline class representing that no activity is being performed. The dropout layer aims to reduce overfitting during training. Alternative structures can be explored in the future, but the current pipeline is sufficient to demonstrate applicability of the ActionNet data to activity classification and to explore the impact of using multiple modalities.

The sub-steps to follow in this section are the following ones:

- For the EMGs' data (optionally gaze), in order to compute the accuracy, implement:
 - dataloader
 - preprocessing
 - network
- Modify or change the architecture to take advantage of a 2D CNN, as in [Kazakos et al., 2019] does with audio modalities, obtaining 8 channels 2D representation forearm (Discuss with the project's TAs for more details about this step).

STEP 3.b - Visual2Signal.

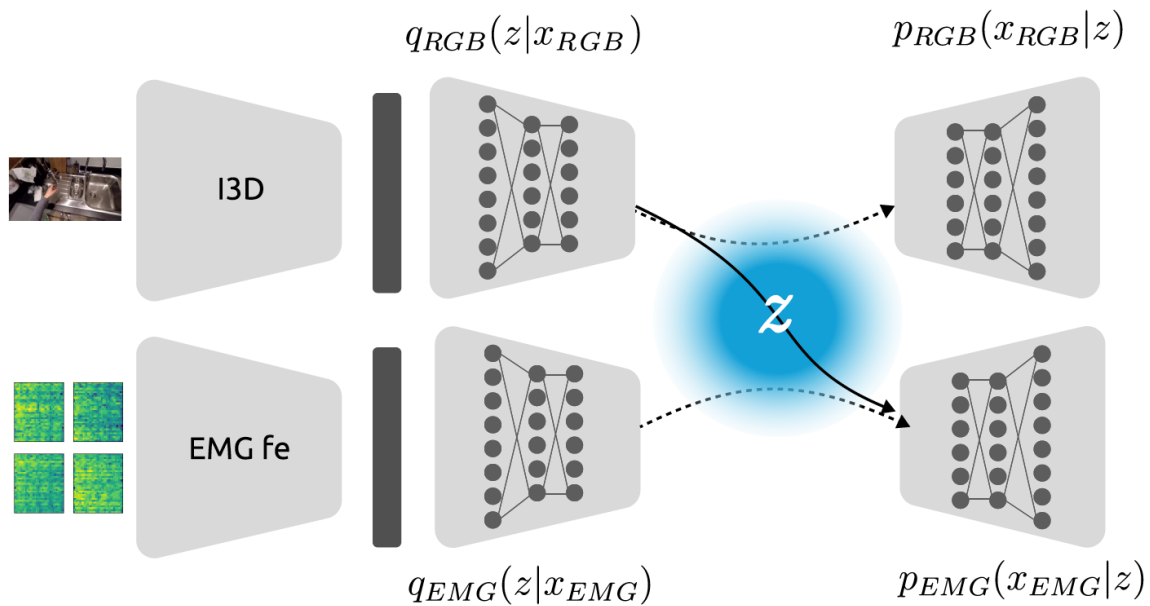


Figure 4. Overview of the multimodal autoencoder architecture.

In this step the student should investigate a solution to transfer the visual input (RGB) to another modality. Primarily, the model should be able to map the visual input in a latent space, using the previous features extractor. Then, passing from this representation reconstruct the signal of other modalities (EMG, Gaze, ...).

This mapping is accomplished through the use of autoencoders, which are neural networks composed of an encoder (mapping to the latent space) and a decoder (mapping back to the original space of other modalities).

The sub-steps to follow in this section are the following ones:

- 1) The students should focus mainly on the two modalities: RGB frames and sEMG signals³ ;
- 2) Assuming that the feature extractor of RGB data is frozen for computational reasons, the students should implement an autoencoder with proper latent properties that is able to extract one modality starting from the other one (i.e., $RGB \rightarrow sEMG$);

³ Read the part of Step 3.a, "Dataset and Preprocessing," to understand the EGMs data and how to deal with them.

- 3) Finally, in Epic Kitchens, where this modality is missing, extract the simulated sEMG signal and use it to compute classification accuracy in single and multimodal (RGB-sEMG) fashion.


The outcome of this step can be expressed in two terms: first of all there will be qualitative results derived from the autoencoder, the students will be able to show how an RGB frame is translated into its sEMG counterpart; secondly they will show the accuracy obtained by adopting this modality on an egocentric dataset (EK).

Examples from the literature, not related to egocentric vision data:

Spurr, Adrian, et al. (2018). [Cross-modal deep variational hand pose estimation](#). *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Yu, H., & Oh, J. (2022). [Anytime 3D Object Reconstruction Using Multi-Modal Variational Autoencoder](#). *IEEE Robotics and Automation Letters*, 7(2), 2162-2169.

(Discuss with the project's TAs for more details about this step)

 [CODE](#) (Please read carefully the readme file present in the code and the comments that you will find in the code.)

Questions you should be able to answer at the end of the project.

- Describe the egocentric vision topic, popular dataset, task, and main challenges.
- Describe the main modalities involved in this context.
- Differences from uniform and dense sampling.
- Describe differences between 2D and 3D CNN and RNN.
- ...

References

1. Chen, C.-F., Panda, R., Ramakrishnan, K., Feris, R., Cohn, J., Oliva, A., & Fan,

Q. (2020, October 22). *Deep analysis of CNN-based spatio-temporal representations for action recognition*. ArXiv.Org.

<https://arxiv.org/abs/2010.11757>

2. Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., & Price, W. (2018). Scaling egocentric vision: The epic-kitchens dataset. *Proceedings of the European Conference on Computer Vision (ECCV)*, 720–736.
3. DelPreto, J. (n.d.). ActionNet: A Multimodal Dataset for Human Activities Using Wearable Sensors in a Kitchen Environment. *Thirty-Sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
4. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., & Liu, X. (2022). Ego4d: Around the world in 3,000 hours of egocentric video. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18995–19012.
5. Guzhov, A., Raue, F., Hees, J., & Dengel, A. (2022, May 23). Audioclip: Extending clip to image, text and audio. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. <http://dx.doi.org/10.1109/icassp43922.2022.9747631>
6. Kazakos, E., Nagrani, A., Zisserman, A., & Damen, D. (2019). Epic-fusion: Audio-visual temporal binding for egocentric action recognition. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5492–5501.
7. Nair, S., Rajeswaran, A., Kumar, V., Finn, C., & Gupta, A. (2022). R3m: A universal visual representation for robot manipulation. *ArXiv Preprint ArXiv:2203.12601*.

8. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Aspell, A., Mishkin, P., & Clark, J. (2021). Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*, 8748–8763.