Multimodal Egocentric Vision

Teaching Assistant: Simone Alberto Peirone (simone.peirone@polito.it)

REVISIONS (changes are highlighted in red)

- 06/04/2023: added a clarification how EK-55 is used and which checkpoints are provided.
- 11/04/2023: added more implementation details, added table 1

OVERVIEW

Figure 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 [9]. 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 steps:

1. **Reading literature**. Before beginning to implement code or run experiments, it is critical to become familiar with the literature on egocentric vision.
2. **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).

3. **Variation**. The student should start to analyze a new modality in two possible directions: by designing an ad-hoc model for action recognition or by implementing a variational auto-encoder to translate between different modalities.

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. 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.

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 N (typically 5) 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** selects a number of evenly spaced frames in the clip.

While dense sampling focuses more on the appearance of the video as the frames are close in time, 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 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.
2) Using the provided pretrained I3D checkpoints, extract the intermediate features and save them in a .pth or .pkl file for the RGB stream of the two dataset (EPIC-Kitchens [4] and ActionNet [6]). These features are the output of the last layer before the classifier of the network (*make sure the network is in evaluation mode during this step*). On EPIC-Kitchens-55 we focus our analysis on the three largest domains kitchens, as in [3], denoted as D1, D2 and D3. We provide a different I3D checkpoint for each domain here. For ActionNet, you can restrict the analysis to S04.
3) Using the extracted features
   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 and project the features to a lower dimensional space using PCA or t-SNE.*
   b) Other analyses may be carried out at the student's discretion and would be appreciated.
4) Finally, the student must use the extracted features to train a classifier to recognise the actions depicted in the video. Various strategies can be implemented at this stage. For example, each clip can make predictions on its own, which are then averaged to calculate the final predictions. Or, the

features of several clips can be further manipulated using an LSTM or another block, e.g. TRN, to calculate the output. Report the accuracy in the following table:

Table 1) Network	**Sampling**	**Accuracy RGB (%)**		
		D1	D2	D3
I3D	5 clips 16 frames per clip			

Implementation and Dataset details:

- **Models**: I3D [2] with backbone BNInception pre-trained on EPIC-Kitchens.
- **Dataset**: the split of EPIC-Kitchens for DA proposed in MM-SADA [7], which comprehends only P01_X, P08_X and P22_X files, which correspond to D1, D2 and D3 domains which represent three different kitchens. You should download only the videos used for testing, which are indicated in the D1_test.pkl, D2_test.pkl and D3_test.pkl files.
- **Action Classes**: 8
- **Metrics**: accuracy (%)

You can find at this link the data you need **here**.

The code is available at this link, where you can find a "TODO" label in correspondence to the parts you have to implement by yourself.

In `data/`, you can find the input data, and in `checkpoints/` you can find the model weights that you have to load into your model using the `args.resume_from` argument.

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

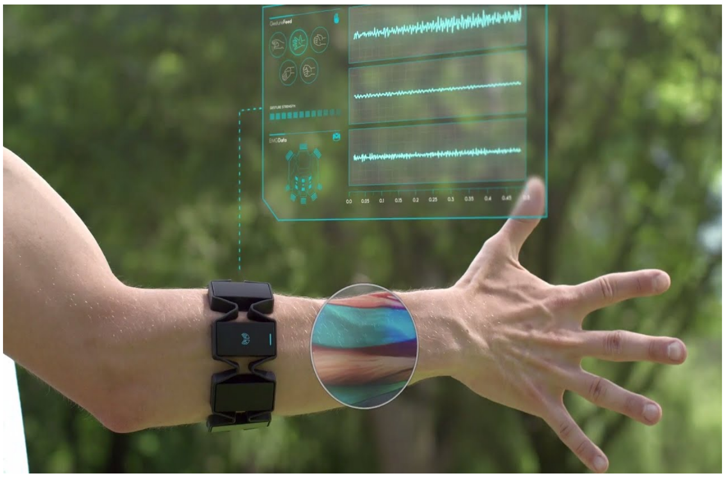

Figure 2. Myo Gesture Control Armband from Thalmic Labs

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 incorporates 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 10 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.

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 [0, 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.

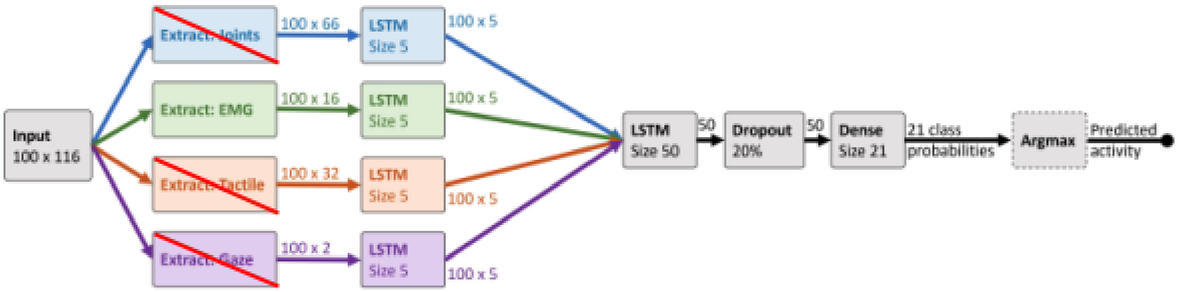

Figure 3. Overview of multi stream network used in ActionNet [3].

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 3. 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 x 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:

1) Implement a preprocessing pipeline for EMG data following [6]. Then, train a simple LSTM for action recognition on the EMG samples.
2) Compute the spectrogram of the EMG data. Then, train a classification model using a 2D CNN, like [3] does with audio. ***(Discuss with the project's TAs for more details about this step)*.**
3) Implement a multi-modal classifier using both RGB and EMG samples. Make sure the sampling process is synchronized across the two modalities, i.e. the boundaries of the selected RGB frames coincide with the start and endpoint of the EMG sequence. The simplest approach consists in having two separate feature extractors for each modality. Features from RGB and EMG are then summed and fed to the final classifier.

STEP 3.b - Visual2Signal.

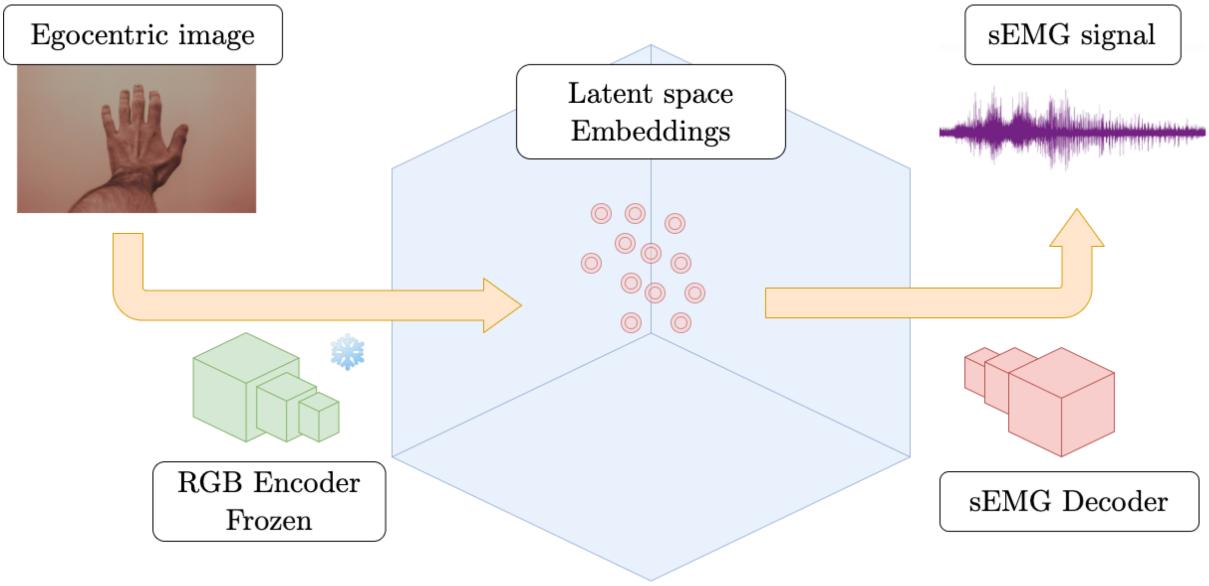

Figure 4. Overview of the multimodal autoencoder architecture.

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 signals[1] ;
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 → sEMG);
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 possibly 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).

An example from the literature, not related to egocentric vision data:

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)

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

EXAMPLE OF QUESTIONS YOU SHOULD BE ABLE TO ANSWER AT THE END OF THE PROJECT

- Describe the egocentric vision topic, popular dataset, task, and main challenges.
- What are the main strengths and weaknesses of the modalities commonly used for action recognition?
- What are the differences from uniform and dense sampling?
- Describe differences between 2D and 3D CNN and RNN.
- Which preprocessing techniques are used for RGB and EMG data?
- How does a variational autoencoder work?

REFERENCES

1. Carreira, Joao, and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6299-6308). 2017. PDF
2. Lin, Ji, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 7083-7093). 2019. PDF
3. Kazakos, E., Nagrani, A., Zisserman, A., & Damen, D. (2019). Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 5492-5501). PDF
4. Kazakos, E., Huh, J., Nagrani, A., Zisserman, A., & Damen, D. (2021). With a Little Help from my Temporal Context: Multimodal Egocentric Action Recognition. *arXiv preprint arXiv:2111.01024*. PDF
5. Chen, Chun-Fu Richard, Rameswar Panda, Kandan Ramakrishnan, Rogerio Feris, John Cohn, Aude Oliva, and Quanfu Fan. "Deep analysis of cnn-based spatio-temporal representations for action recognition." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* (pp. 6165-6175). 2021. PDF
6. Joseph DelPreto, Chao Liu, Yiyue Luo, Michael Foshey, Yunzhu Li, Antonio Torralba, Wojciech Matusik, and Daniela Rus. ActionSense: A Multimodal Dataset and Recording Framework for Human Activities Using Wearable Sensors in a Kitchen Environment. In *Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks* 2022. PDF
7. Tan, Shuhan, Tushar Nagarajan, and Kristen Grauman (2023). "EgoDistill: Egocentric Head Motion Distillation for Efficient Video Understanding." arXiv preprint arXiv:2301.02217. PDF
8. Moon, Seungwhan, et al. (2022). "IMU2CLIP: Multimodal Contrastive Learning for IMU Motion Sensors from Egocentric Videos and Text." *arXiv preprint arXiv:2210.14395*. PDF

9. Nair, Suraj, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. (2022). "R3m: A universal visual representation for robot manipulation." arXiv preprint arXiv:2203.12601. PDF
10. Gao, Ruohan, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani (2020). "Listen to look: Action recognition by previewing audio." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10457-10467.
11. Ehsani, Kiana, Daniel Gordon, Thomas Hai Dang Nguyen, Roozbeh Mottaghi, and Ali Farhadi (2021). "What Can You Learn From Your Muscles? Learning Visual Representation from Human Interactions." In *International Conference on Learning Representations*.