Using External Knowledge to Improve Zero-shot Action Recognition in Egocentric Videos

Adrián Núñez-Marcos[1[0000−0002−5324−4514]], Gorka Azkune[2[0000−0002−2506−7426]], Eneko Agirre[2[0000−0002−0195−4899]], Diego López-de-Ipiña[1[0000−0001−8055−6823]], and Ignacio Arganda-Carreras[3,4,5[0000−0003−0229−5722]]

[1] Deustotech Institute, University of Deusto, Avenida de las Universidades, No. 24, 48007 Bilbao, Spain
{adrian.nunez,dipina}@deusto.es

[2] IXA NLP Group, Faculty of Computer Science, University of the Basque Country, P. Manuel Lardizabal 1, 20018 San Sebastian, Spain
{gorka.azcune,e.agirre}@ehu.es

[3] Department of Computer Science and Artificial Intelligence, University of the Basque Country, P. Manuel Lardizabal 1, 20018 San Sebastian, Spain
ignacio.arganda@ehu.eus

[4] Ikerbasque, Basque Foundation for Science, Maria Diaz de Haro 3, 48013 Bilbao, Spain

[5] Donostia International Physics Center (DIPC), Manuel Lardizabal 4, 20018 San Sebastian, Spain

Abstract. Zero-shot learning is a very promising research topic. For a vision-based action recognition system, for instance, zero-shot learning allows to recognise actions never seen during the training phase. Previous works in zero-shot action recognition have exploited in several ways the visual appearance of input videos to infer actions. Here, we propose to add external knowledge to improve the performance of purely vision-based systems. Specifically, we have explored three different sources of knowledge in the form of text corpora. Our resulting system follows the literature and disentangles actions into verbs and objects. In particular, we independently train two vision-based detectors: (i) a verb detector and (ii) an active object detector. During inference, we combine the probability distributions generated from those detectors to obtain a probability distribution of actions. Finally, the vision-based estimation is further combined with an action prior extracted from text corpora (external knowledge). We evaluate our approach on the EGTEA Gaze+ dataset, an Egocentric Action Recognition dataset, demonstrating that the use of external knowledge improves the recognition of actions never seen by the detectors.

Keywords: Egocentric Action Recognition · Zero-shot Learning · External Knowledge

1 Introduction

Vision-based action recognition is a major emerging field, mainly due to the broad range of applications in domains such as health [9] or surveillance [11, 3, 4]. The majority of the research has focused on exocentric videos, where the action is being observed from a third-person's perspective. Nonetheless, in the last decade, thanks to the growth in the amount of wearable camera devices, the Egocentric Action Recognition (EAR) field has attracted the interest of the computer vision community. EAR is specially well suited to recognise actions performed by a person, since the visual information of the working space is usually perfectly visible. From the application point of view, such potential makes EAR interesting for Ambient Assisted Living, where the visual information captured by egocentric devices can be used to assist users.

Egocentric action videos are usually labelled with a verb and a set of objects, creating an action when combined, e.g., "open fridge" or "cut cucumber". However, datasets are quite limited in the number of combinations of verbs and objects, thus constraining the scalability of the developed systems. In fact, for an action recognition system to be useful in real world settings, being able to generalise to any action is crucial. This problem is known as Zero-Shot Learning (ZSL). In the zero-shot action recognition literature it is common to find solutions that disentangle the action classification into the verb (the movement) and the active object (the visually manipulated object) classification [18, 13]. Following such approach, both the verb and the active object would be separately inferred. Therefore, should the system receive a never seen action, it would be able to make a prediction by combining the knowledge acquired from those two separated branches, as long as the verb and the object have been previously seen. More formally, assuming $|V|$ and $|O|$ are the number of verbs and objects in the training set respectively, the system would be able to recognise $|V| \times |O|$ actions only requiring $|V| + |O|$ labels.

Nonetheless, naively combining verbs and objects may wind up with action predictions that do not exist, such as "cut fridge", following the previous examples, or action predictions that are rare, instead of those that are performed more frequently. Thus, we propose to add external knowledge to the system to address those problems. As the action prediction can be represented as a string of text, external text corpora containing pairs of verbs and objects (actions) can be efficiently used to create an action prior. The latter provides a probability distribution over the set of possible actions created from the Cartesian product between a set of verbs and objects (those learnt from the two separate verb and object detectors).

In that sense, it is important to find a suitable source of external knowledge, since different action priors from different knowledge domains may have different results and effects. For example, using a cooking book corpus will benefit actions often appearing in recipes, whilst a corpus created from several books may provide more general knowledge. We raised this question and proposed several experiments to test a number of corpora in order to provide insights on the matter.

Therefore, this paper presents the following two main contributions:

1. A novel method which uses external knowledge in form of text corpora to improve the performance of vision-based action recognition systems for ZSL in egocentric videos.
2. A thorough analysis to measure the effects of applying action priors extracted from different sources.

2 Related work

Even though the EAR field has gained popularity in the last decade [2, 10, 16], the Zero-Shot EAR subfield is still developing, and, to the best of our knowledge, the number of works is quite limited.

The idea of fusing verbs and objects to infer new combinations is already introduced by Zhang et al. [18]. They used the Fisher Vector encoding of features such as Improved Dense Trajectories or Histogram of oriented Gradients and visual CNN features, respectively. In fact, the idea of dividing the verb and the object influenced other researchers, as well as this work. In addition, they analysed various fusion methods among *early*, *late*, and *early+late* stage fusion. In a similar fashion, Al-Naser et al. [1] used a Myo armband sensor and a Multi-layer Perceptron to classify verbs and a gaze-point-based cropping of video frames as input to a GoogleNet [15] to predict objects. Any new action composed from the combination of the learnt verbs and objects can be inferred from their system.

Guadarrama et al. [5] aimed at inferring descriptions of the actions in videos. When unseen actions appeared, they used a semantic hierarchy built from free annotations to provide a less specific and more general answer. In their case, they used triplets of subject, verb and object in their hierarchy. In our case, we only include verb and object, as the subject in first-person videos is always the one recording the video, and we use text corpora to build a probability distribution to help us decide which prediction of the system is more suitable, rather than to be able to provide a more generic answer.

Although there are other approaches in the exocentric vision domain using zero-shot learning [7], we would like to highlight the work performed by Shen et al. [13]. In particular, they aim to recognise Human-Object Interactions in images using a system based on a Faster Region-Based CNN [12] that branches, on top of it, into two streams: the verb and the object detection networks. The output of the system provides two probability distributions: one for verbs and the other one for objects. By multiplying these, a matrix is obtained where the probability at a position (i,j) refers to the probability of the action created by fusing the i^{th} verb with the j^{th} object.

In general, our work follows those where the verb and the active object are separately modelled. However, instead of focusing on improving those detectors by means of new computer vision and/or machine learning techniques, we provide a novel way to leverage external knowledge on top of those vision-based detectors and improve the overall action recognition performance. Hence, in principle, our

proposal could be used to improve any approach which relies on separated verb and object detectors.

3 Methodology

In the context of EAR, a ZSL approach aims to create a model which is capable of recognising actions that have never been seen during the training phase. Inspired by the literature, we separate an action into a verb and an active object. In consequence, recognising the verb and the active object in a given video, we can infer the performed action. Following that idea, we built two identical neural networks to detect the verb and the active object. We pose the problem as a classification problem, where given a video, i.e., a sequence of frames, the detectors have to estimate the probability distribution over the set of verbs and active objects. Then, we combine both probability distributions to infer the action such that $a = max_i\{p^v(a_i)\}$, where a is the action label and $p^v(a_i)$ denotes the probability for the i^{th} action estimated by the vision-based system. This probability is calculated as $p^v(a_i) = p(v_{a_i}) \times p(o_{a_i})$, where $p(v_{a_i})$ and $p(o_{a_i})$ denote the probability of the verb and object disentangled from a_i. Those probabilities are estimated by the neural networks D_V and D_O.

Moreover, we use external knowledge in form of text corpora to compute a probability distribution of all the combinations of verbs and objects. Specifically, we look for co-occurrences of those verbs and objects within N-grams extracted from text corpora to create the probability distribution which we call the action prior. This prior is combined with the probabilities of the actions obtained from the combination of the verb and object detectors. The final action prediction is the one with the highest probability. More formally, given $p^v(a_i)$ and $p^t(a_i)$ (the action prior for the i^{th} action), the inferred action a is calculated as $a = max_i\{p^v(a_i) \times p^t(a_i)\}$. An overview of the system is shown in Figure 1.

3.1 System architecture

Both D_V and D_O, the verb and object detectors, take as input a video $X = \{F_1, F_2, ..., F_n\}$, an ordered list of frames of the video, where $F_i \in R^{224x224x3}$. As the videos have a varying length, we uniformly sample 25 frames from each one. The network architecture is based on the work of Sudhakaran et al. [14], being composed of a Convolutional Neural Network (CNN) with a Recurrent Neural Network (RNN) on top, as the feature extraction part, and a single Fully-Connected (FC) layer as the classifier. Specifically, in this work we use a ResNet50 [6] architecture as the CNN (with a $1x1$ convolution of 256 filters on top to reduce the dimensionality) and a Convolutional Long Short-Term Memory (ConvLSTM) [17] as the RNN. The detector outputs a probability distribution $p = \{p_1, p_2, ..., p_n\}$, where $p_i \in [0, 1]$ is the probability of the class i for a given video such that $\sum_{i=1}^{n} p_i = 1$. Depending on the task of the network, i.e., predicting verbs or active objects, p is defined as $p(v)$ (output of D_V) or $p(o)$ (output of D_O), respectively, and p_i as $p(v_i)$ and $p(o_i)$.

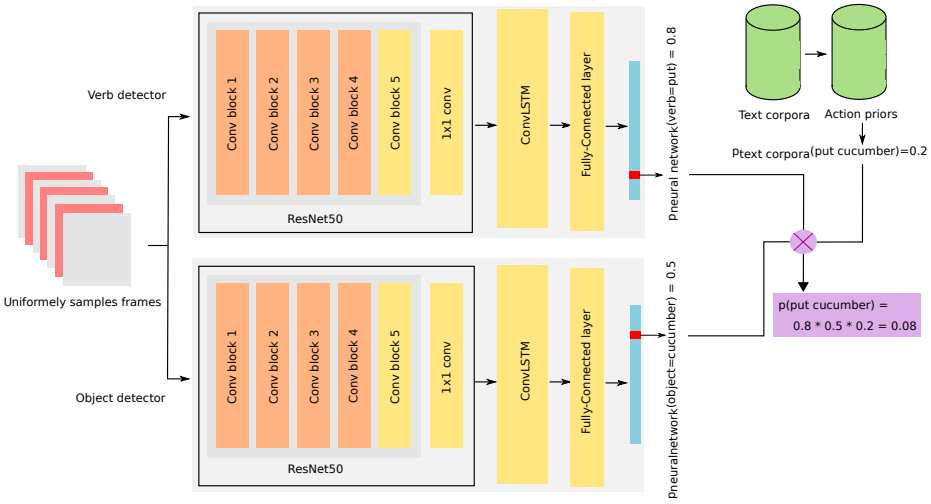

Fig. 1. Architecture overview: two neural networks composed of a ResNet50 and a ConvLSTM take as input a video (uniformly sampled frames) and output two probability distributions (verbs and objects). The resulting probability distributions are combined with an action prior sampled from text corpora to infer the most probable action. The layers or blocks of layers in orange are frozen while the yellow ones are trained.

3.2 Action priors

The action prior $p^t(a)$ is the probability distribution over the set of actions coming from the Cartesian product of the set of verbs and objects from a given dataset, i.e., $\{(v_j, o_k) : \forall j, \forall k | v_j \in V \; and \; o_k \in O\}$ and $a_i = (v_j, o_k)$. The objective of the action prior is to estimate the likelihood of a given combination of verb and object, i.e., $p^t(a_i)$, based on external knowledge completely independent from the action recognition videos. We propose to estimate those priors using different textual corpora.

In our work, the following external knowledge sources are used to estimate action priors:

- Cookbook wiki: using the Cookbook wiki page [6], we extract a corpus containing recipes and, thus, actions related to cooking recipes. We selected this knowledge source to further narrow the domain of the egocentric videos and see how specialised knowledge can help for ZSL.
- Google searcher API[7]: we use the API to search for actions and get the number of results as the number of occurrences. This knowledge source was chosen to have a more general prior estimation which is not focused on a specific domain, in contrast to Cookbook wiki.

[6] https://en.wikibooks.org/wiki/Cookbook:Recipes
[7] https://developers.google.com/custom-search/v1/overview

– *Phrasefinder* searcher API[8]: similar to the Google API, the *Phrasefinder* source has no specific domain, as it searches through Google Books' N-grams. We chose this as a more controlled alternative to the Google API prior, whose results come from a wilder environment (any site indexed in Google).

In order to create the prior, for the Cookbook source, we scrapped the Wikicook to obtain a corpus and cleaned it. With the raw corpus, we removed non-ascii characters, lowercased the text, eliminated stop words and applied the WordNet lemmatiser[9]. Finally, we experimentally decided to extract N-grams of size 4. To determine that an action appears in an N-gram, both the verb and the object of the action are taken separately and both must appear within the N-gram, not necessarily in adjacent positions. In fact, instead of just taking the verb and the object as they are, we manually defined a list of synonyms for each one and, for each possible combination of synonyms of a verb and an object, their appearance in the N-gram is checked. If at least a synonym of the verb and a synonym of the object are contained in the N-gram, it is considered that the action is contained in it. The number of N-grams where the action is found divided by the total number of N-grams is the final prior of the action.

In the case of the Google and *Phrasefinder* sources, for a given action, the API returns the number of results given by the query. This query is created with the expression "verb * object" with the Google API and "verb ? object" with the *Phrasefinder* API. The symbol "*" and "?" are wildcards, placeholders for strings such as "a", "the", and so on; the use of wildcards allows for searches that are more natural than just searching for "verb object". The query returns the number of results, which is used as the frequency of the action. Again, we use synonyms for verbs and objects, using the mean of all the non-zero results as the final frequency of the action. The latter is normalised by the sum of all actions' frequencies to obtain the action prior.

4 Experimental setup

4.1 Dataset and evaluation metrics

We chose the EGTEA Gaze+[10] dataset for our experiments. Launched in 2017, this dataset contains 28 hours of egocentric videos with 32 subjects performing cooking related actions. It is composed of 10,325 action segments, with 19 verbs, 53 nouns and 106 actions.

To evaluate the performance of the tested systems, apart from reporting the accuracy over the predicted actions, we also chose to present the F1 score in its macro variant, due to the unbalanced nature of the EGTEA Gaze+ dataset: classes with few samples that may not be learnt have low impact in the accuracy,

[8] https://phrasefinder.io/api

[9] http://www.nltk.org/_modules/nltk/stem/wordnet.htmlWordNetLemmatizer

[10] Georgia Tech: Extended GTEA Gaze+. http://webshare.ipat.gatech.edu/coc-rim-wall-lab/web/yli440/egteagp

in contrast, they have a significant impact on the F1 score. In addition, in the F1 computation, we include an artificial class in which all predictions out of the set of test actions are included. This class is taken into account to compute the F1 score.

4.2 Zero-shot splits

EGTEA Gaze+ consists of three official training and test splits that provide a common ground to evaluate action recognition systems. However, those official splits are not suitable for ZSL, since the actions in the test set are also represented in the train set. Therefore, in our experiments, we employ new splits. Using all the data in EGTEA, we followed the guidelines given by Shen et al. [13] for a similar problem. First, we removed action videos containing verbs and objects that only appear once, as they are not appropriate for the zero-shot task, as formulated in this paper. This left us with 9 verbs and 29 objects. Second, to generate the test set, we randomly took 20% of the action classes under the condition that any verb and object contained in that test set must appear in the training set (in any action). That is, all the verbs and objects must appear in the training set. The validation set is created taking a stratified subset from the resulting training set, using the 10% of the videos in train. Note that the validation set is important not for the ZSL task itself, but to train and tune the detectors.

Since we aim at measuring the effects of specialised and generic knowledge in the system, we propose two types of splits: the first one, denoted as the Recipe split (R split), is built explicitly discarding some verbs, objects and actions which have nothing to do with recipes. Specifically, we banned the verb *Inspect/Read*, the objects *cabinet*, *sponge*, *grocery bag*, *eating utensil*, *drawer*, and *fridge drawer* and the action *wash pan*. The split created with this rules has 6121 training videos and 1464 test videos. The second one, called the No Recipe split (NR split), avoids any bias. To create the test set we do not impose any other condition apart from the ones given by Shen et al. [13]. We assume that the Cookbook prior will not be as effective in this type of split as in the R split, as such prior produces a specific type of probability distribution focused on actions related to recipes. In this case, the split has 6277 training videos and 1308 test videos.

4.3 Experiments

We performed several experiments to validate the hypothesis posed in this paper, i.e., that we can improve zero-shot EAR using external knowledge. We compare our proposed system with a baseline system which only relies on D_V and D_O. For this baseline, we infer the action of a given video computing $a = max_i\{p^v(a_i)\}$. Moreover, we test our system with the proposed three action priors (provided by Cookbook, Google and *Phrasefinder*) on both ZSL splits (R and NR).

For each split, both D_V and D_O, as in Section 3.1, are trained for 100 epochs with early stopping with a patience of 10 epochs (the macro-F1 metric in the validation set is used to stop). The CNN weights are initialised with an Imagenet

pre-training and are frozen up to the the 4^{th} convolutional block, being the 5^{th} fine-tuned (see Figure 1). We use Adam [8] optimiser with a batch size of 16, initial learning rate of $1e^{-4}$ and 25 timesteps per video. To avoid overfitting as much as possible we use class weights in training (for the loss function, they penalise errors in classes with fewer samples) and data augmentation: (i) standard random horizontal flipping and (ii) multi-scale random corner cropping, i.e., one of the four corners or the centre position are randomly selected as a possible crop, the initial crop size is set to 224×224, but is scaled with a factor randomly chosen among $1, 0.875, 0.75, 0.65625$ and then re-scaled again to 224×224.

The code of our proposed approach is publicly available[11].

5 Results

The results for verb and object classification per split, using D_V and D_O independently, are shown in Tables 1 and 2, respectively. As it can be observed, the performance in the test set with respect to the validation set suffers a significant drop, specially for the case of D_O. We believe a possible explanation why D_V does not deteriorate as much is the number of classes that must be learnt (9 for verbs and 29 for objects). Besides, we hypothesise that this drop may be a consequence of the different shapes and poses that objects have in the test set compared to the training or the validation set. For instance, a tomato observed during the action "take" may look different from a tomato which is being "cut", especially since the tomato may be partially occluded or even sliced during the latter. This observation suggests that the active object detection is highly correlated with the verb and thus, active object detectors specially suffer in ZSL conditions.

Table 1. Verb classification results with verb detector. The results are given as the mean of 3 runs, with the standard deviation.

Verb detector	Split	Train		Validation		Test	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
	R	99.37% (±0.10)	99.22 (±0.19)	75.69% (±0.35)	67.93 (±0.58)	60.31% (±1.10)	31.08 (±0.06)
	NR	98.44% (±0.61)	97.64 (±0.80)	76.65% (±0.42)	66.42 (±3.11)	53.49% (±1.44)	42.90 (±1.46)

Using the presented D_V and D_O, we carried out all the experiments of Section 4.3 and show the results in Table 3. To analyse the results, paying attention to the type of split is pivotal.

On the one hand, we have the R split, with a test set created specifically with actions related to recipes. The baseline result for this split in Table 3 is

[11] https://github.com/AdrianNunez/zeroshot-action-recognition-action-priors

Table 2. Active object classification results with object detector. The results are given as the mean of 3 runs, with the standard deviation.

Object detector	Split	Train		Validation		Test	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
	R	99.37% (±0.10)	99.22 (±0.19)	75.69% (±0.35)	67.93 (±0.58)	27.14% (±0.74)	13.48 (±0.73)
	NR	98.70% (±0.56)	98.35 (±0.76)	76.43% (±0.34)	68.64 (±0.39)	31.75% (±0.81)	15.34 (±0.51)

higher than for the NR split. Apart from that, the R split benefits the most from the prior built from a corpus of recipes (Cookbook), having an improvement of 5.47 points in accuracy compared to the baseline. However, it is also important to point out that the Google prior grants a slight improvement of 1.73 points, even though it is not as appropriate as the Cookbook prior for this type of split. The reason may well be that the Google prior helps discarding non-existing actions and promoting actions that are more common. We gratefully acknowledge the support of the Basque Government's Department of Education for the predoctoral funding of the first author. This work has been supported by the Spanish Government under the FuturAAL-Ego project (RTI2018-101045-A-C22) and the FuturAAL-Context project (RTI2018-101045-B-C21) and by the Basque Government under the Deustek project (IT-1078-16-D).

Results per class of the test set of the R split are shown in Table 4. It can be seen that the Cookbook experiments are the ones that show the largest improvement on the majority of the classes, although the Google and the *Phrasefinder* priors have also some classes where they can surpass the Cookbook priors. In fact, this is the expected behaviour given the prior of each class. Classes where the Google or the *Phrasefinder* prior is the highest among these three are also the ones where they have the best accuracy.

On the other hand, the NR split shows an accuracy improvement on every experiment, having a higher accuracy with the Cookbook prior but higher F1 with the Google and the *Phrasefinder* priors. Observing Table 5, it is clear that the Cookbook prior has the potential to improve a few classes to a high accuracy, but this effect is localised in some classes and zeroes out others. Meanwhile, the Google prior has a higher F1 due to the balanced effect it has, i.e., it only zeroes out a class with a baseline low accuracy (divide/pull apart lettuce) while it obtains the best accuracy in 5 classes. In fact, this is the expected behaviour, as the Cookbook prior has a few non-zero action probabilities due to the constrained domain used and the Google prior has broader knowledge, thus including more actions and a more balanced distribution.

Moreover, half of the classes do not have any performance gain in any experiment (those in which the baseline is highlighted in bold). There may be some reasons why this can happen in any split: (i) the presence of meta-objects (such as *eating* or *cooking utensil*), as discussed by [14], can affect the performance, as a single label (hyperonym) covers various objects (hyponyms) and learning them is more difficult; and (ii) difficult to learn verbs and objects whose performance

affect the learning of the action, a problem caused by the detectors, because of the few samples in training or their intrinsic variance.

We can conclude that specific domain knowledge applied in the same domain can be beneficial, as in the case of the Wikicook prior in the R split. In fact, not only is it helpful to be in the same domain, approximating the prior to the distribution of actions is very promising too, as seen with the perfect prior. In the case of this dataset, the actor had controlled actions but, different people usually have different routines and, thus, a different action distribution. Adjusting the prior to each one could potentially be a huge improvement. In the opposite side, we have the Google prior, whose generic knowledge seems to be more balanced and helpful in almost all the classes but not as beneficial as a prior specific to a domain of actions.

Table 3. Table of zero-shot action classification results: comparison between the baseline and the experiments using the Cookbook, the Google, the *Phrasefinder* and the perfect priors. Results in bold highlight the best result.

	Baseline		Cookbook prior		Google prior		*Phrasefinder* prior		Perfect prior	
Split	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
R	12.61% (±0.56)	16.52 (±0.23)	**18.08% (±0.99)**	**22.65 (±0.80)**	14.89% (±0.93)	18.89 (±0.64)	14.34% (±0.90)	18.29 (±1.05)	51.46% (±1.45)	44.14 (±0.97)
NR	8.03% (±0.54)	11.46 (±1.21)	**11.47% (±1.41)**	9.73 (±1.25)	9.17% (±0.91)	**12.58 (±1.46)**	10.37% (±1.01)	11.48 (±1.16)	54.31% (±0.42)	45.51 (±1.68)

Table 4. Table of zero-shot action classification results by class in the R split using the accuracy: comparison between the baseline and the experiments using the Cookbook, the Google, the *Phrasefinder* and the perfect priors. Results in bold highlight the best result (not taking into account the perfect prior experiments).

Class (R split)	Baseline	Cookbook prior	Google Prior	*Phrasefinder* prior	Perfect prior
cut bell pepper	14.22% (±6.04)	**23.28% (±6.64)**	14.46% (±6.36)	6.86% (±1.93)	63.97% (±9.92)
cut onion	0.57% (±0.00)	**8.05% (±0.47)**	1.92% (±0.27)	2.87% (±0.47)	57.47% (±2.15)
put bread	18.09% (±4.51)	25.89% (±7.39)	25.53% (±4.84)	**29.79% (±6.26)**	64.18% (±8.74)
put cup	7.92% (±3.58)	14.17% (±5.62)	14.58% (±5.62)	**15.42% (±4.60)**	41.67% (±10.27)
put lettuce	21.36% (±3.46)	**41.75% (±3.46)**	25.57% (±2.29)	37.86% (±1.59)	77.67% (±1.37)
put onion	2.56% (±3.63)	**10.26% (±5.54)**	2.56% (±2.09)	5.98% (±3.20)	5.13% (±2.09)
put plate	21.32% (±5.73)	29.41% (±4.33)	25.25% (±5.58)	**29.66% (±5.71)**	61.52% (±10.54)
put pot	14.52% (±3.99)	**28.05% (±1.23)**	16.50% (±2.60)	25.08% (±1.23)	50.17% (±1.23)
put tomato	4.37% (±1.48)	3.17% (±1.48)	**5.95% (±0.97)**	1.59% (±1.12)	13.49% (±0.56)
take bowl	30.00% (±7.08)	18.00% (±5.19)	**32.22% (±7.31)**	18.44% (±6.02)	75.11% (±6.19)
take egg	0.00% (±0.00)	0.98% (±1.39)	**2.94% (±4.16)**	0.98% (±1.39)	6.86% (±3.67)
take onion	6.11% (±1.57)	**17.22% (±1.57)**	8.33% (±2.72)	7.78% (±2.08)	39.44% (±2.08)
take pan	17.11% (±1.07)	**17.98% (±3.28)**	17.54% (±3.10)	8.33% (±4.34)	73.25% (±4.07)
take pot	2.99% (±1.60)	**7.26% (±1.60)**	2.99% (±2.63)	2.99% (±1.60)	16.24% (±1.21)
take tomato	2.63% (±2.15)	**3.07% (±1.24)**	2.63% (±1.07)	0.88% (±1.24)	17.54% (±1.64)
wash pot	10.85% (±1.10)	**13.95% (±3.29)**	9.30% (±1.90)	11.63% (±1.90)	57.36% (±3.95)

Table 5. Table of zero-shot action classification results by class in the NR split using the accuracy: comparison between the baseline and the experiments using the Cookbook, the Google, the *Phrasefinder* and the perfect priors. Results in bold highlight the best result (not taking into account the perfect prior experiments).

Class (NR split)	Baseline	Cookbook prior	Google Prior	*Phrasefinder* prior	Perfect prior
close drawer	7.41% (±5.24)	0.00% (±0.00)	**15.56% (±6.54)**	8.89% (±5.44)	53.33% (±10.10)
cut cucumber	6.41% (±2.27)	0.96% (±0.00)	**8.65% (±3.07)**	3.21% (±0.60)	84.78% (±9.50)
cut lettuce	**5.80% (±3.69)**	1.45% (±1.02)	3.62% (±1.02)	1.45% (±2.05)	38.41% (±9.11)
divide/pull apart lettuce	**0.42% (±0.59)**	0.00% (±0.00)	0.00% (±0.00)	0.42% (±0.59)	16.25% (±3.06)
divide/pull apart onion	**0.00% (±0.00)**	0.00% (±0.00)	0.00% (±0.00)	0.00% (±0.00)	22.44% (±7.08)
open fridge drawer	**0.00% (±0.00)**	0.00% (±0.00)	0.00% (±0.00)	0.00% (±0.00)	42.32% (±7.42)
put bell pepper	5.67% (±5.31)	0.00% (±0.00)	**17.02% (±6.95)**	0.71% (±1.00)	24.82% (±14.15)
put bowl	16.55% (±3.65)	**48.55% (±4.66)**	22.82% (±3.05)	31.77% (±3.12)	77.85% (±3.05)
put cheese container	0.00% (±0.00)	0.00% (±0.00)	0.00% (±0.00)	0.00% (±0.00)	9.40% (±7.93)
put cup	5.83% (±1.56)	8.33% (±0.59)	**10.42% (±0.59)**	**10.42% (±0.59)**	30.42% (±4.71)
put cutting board	28.67% (±3.40)	29.33% (±12.26)	**34.00% (±5.89)**	25.33% (±12.26)	63.33% (±8.22)
put plate	14.95% (±0.92)	**37.50% (±6.24)**	22.30% (±2.27)	26.47% (±2.40)	70.83% (±3.81)
take bell pepper	**11.95% (±3.21)**	0.00% (±0.00)	6.92% (±4.71)	0.00% (±0.00)	70.83% (±6.23)
take cheese container	**2.38% (±2.23)**	0.00% (±0.00)	0.60% (±0.84)	0.00% (±0.00)	48.81% (±3.04)
take sponge	**8.33% (±3.90)**	0.00% (±0.00)	4.17% (±1.95)	4.69% (±1.28)	56.25% (±9.20)
wash eating utensil	**4.97% (±1.09)**	2.34% (±0.41)	2.92% (±1.09)	1.75% (±0.72)	49.42% (±11.42)

We gratefully acknowledge the support of the Basque Government's Department of Education for the predoctoral funding of the first author. This work has been supported by the Spanish Government under the FuturAAL-Ego project (RTI2018-101045-A-C22) and the FuturAAL-Context project (RTI2018-101045-B-C21) and by the Basque Government under the Deustek project (IT-1078-16-D).

We gratefully acknowledge the support of the Basque Government's Department of Education for the predoctoral funding of the first author. This work has been supported by the Spanish Government under the FuturAAL-Ego project (RTI2018-101045-A-C22) and the FuturAAL-Context project (RTI2018-101045-B-C21) and by the Basque Government under the Deustek project (IT-1078-16-D).

6 Conclusions

Throughout this manuscript, we have presented our system of Zero-Shot Egocentric Action Recognition, a branched approach composed of a verb detector and an object detector whose results are fused to infer an action. This is further improved by the main contribution of the work: the addition of action priors. We have presented several priors from different sources and made experiments with each of them, highlighting their pros and cons. As future work, we aim to improve the base verb and object detectors and how the action priors are fused with them, as this research path has not been extensively exploited.

Acknowledgments

We gratefully acknowledge the support of the Basque Government's Department of Education for the predoctoral funding of the first author. This work has been supported by the Spanish Government under the FuturAAL-Ego project (RTI2018-101045-A-C22) and the FuturAAL-Context project (RTI2018-101045-B-C21) and by the Basque Government under the Deustek project (IT-1078-16-D).

References

1. Al-Naser, M., Ohashi, H., Ahmed, S., Nakamura, K., Akiyama, T., Sato, T., Nguyen, P.X., Dengel, A.: Hierarchical model for zero-shot activity recognition using wearable sensors. In: ICAART (2). pp. 478–485 (2018)
2. Bambach, S.: A survey on recent advances of computer vision algorithms for egocentric video. arXiv preprint arXiv:1501.02825 (2015)
3. Brezovan, M., Badica, C.: A review on vision surveillance techniques in smart home environments. In: 2013 19th International Conference on Control Systems and Computer Science. pp. 471–478. IEEE (2013)
4. deCampos, T.: A survey on computer vision tools for action recognition, crowd surveillance and suspect retrieval. In: XXXIV Congresso da Sociedade Brasileira de Computacao (CSBC). pp. 1123–1132. Citeseer (2014)
5. Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R., Darrell, T., Saenko, K.: Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 2712–2719 (2013)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
7. Junior, V.L.E., Pedrini, H., Menotti, D.: Zero-shot action recognition in videos: A survey. arXiv preprint arXiv:1909.06423 (2019)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Luo, Z., Hsieh, J.T., Balachandar, N., Yeung, S., Pusiol, G., Luxenberg, J., Li, G., Li, L.J., Downing, N.L., Milstein, A., et al.: Computer vision-based descriptive analytics of seniors' daily activities for long-term health monitoring. Machine Learning for Healthcare (MLHC) (2018)
10. Nguyen, T.H.C., Nebel, J.C., Florez-Revuelta, F., et al.: Recognition of activities of daily living with egocentric vision: A review. Sensors **16**(1), 72 (2016)
11. Rege, A., Mehra, S., Vann, A., Luo, Z.: Vision-based approach to senior healthcare: Depth-based activity recognition with convolutional neural networks. Semantic Scholar (2017)
12. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
13. Shen, L., Yeung, S., Hoffman, J., Mori, G., Fei-Fei, L.: Scaling human-object interaction recognition through zero-shot learning. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1568–1576. IEEE (2018)

14. Sudhakaran, S., Lanz, O.: Attention is all we need: nailing down object-centric attention for egocentric activity recognition. arXiv preprint arXiv:1807.11794 (2018)
15. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
16. Tadesse, G.A., Cavallaro, A.: Visual features for ego-centric activity recognition: A survey. In: Proceedings of the 4th ACM Workshop on Wearable Systems and Applications. pp. 48–53. ACM (2018)
17. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems. pp. 802–810 (2015)
18. Zhang, Y.C., Li, Y., Rehg, J.M.: First-person action decomposition and zero-shot learning. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 121–129. IEEE (2017)