Zero-Shot Learning for Human Action Recognition

Nicole Rafidi
nrafidi@cs.cmu.edu

Yuzuko Nakamura
yuzi@cs.cmu.edu

Danny Zhu
dannyz@cs.cmu.edu

Abstract

We demonstrate the performance of zero-shot learning (ZSL) for the task of human action recognition from video. Instead of training a classifier to map from low-level video-extracted features directly to action labels, we instead define a set of semantic features using knowledge of human actions, and train regressors to map the videos to an intermediate semantic feature space, where each action label is a point in this feature space. We classify by choosing the canonical action point that is closest to the query point. This allows us to classify actions for which we have never seen video examples, a generalization that improves the feasibility of human action recognition. We demonstrate the efficacy of this technique with several experiments, and conduct a preliminary analysis of our semantic features.

1 Introduction

Zero-shot learning (ZSL) enables one to classify input with labels not seen in the training data. In other words, it allows a classifier learned on a limited data set to generalize to other labels. It does so by making use of a semantic knowledge base as an intermediate step between the features and classes [1]. The main objective of this project is to see whether zero-shot learning yields good performance in human action recognition, and enables us to classify actions that were never seen in training. Human action recognition is a domain in which there are many possible class labels (actions), making it intractable to provide examples of each action in a training set [2]. This makes it a good candidate for improvement by zero-shot learning.

2 Related Work

Semantic knowledge bases, both user-defined (e.g. [3] and [4]) and automatically learned (e.g. [5]), have previously been used as tools for learning and classifying actions. [3] recognizes classes of outdoor scenes by segmenting an image into tiles and classifying each into a visual concept such as water, foliage, and rocks, and then using the resulting grids to perform classification. In order to describe motion, [4] defines a set of operation triplets that each correspond to the way that specific body parts can move. A video sequence is translated into a subset of those triplets using domain-specific knowledge. Learning techniques using a bag of visual words approach to the problem such as [5] often use algorithms to learn the semantics of a lower-level, raw visual vocabulary in order to achieve more accurate classification results.

Like our approach, these works all make use of an intermediate step to process low-level visual features into something more meaningful before performing classification. However, these approaches are only applied to the domain of training on a set of actions and classifying video containing those actions. By contrast, our approach makes use of a user-defined semantic knowledge base to enable a classifier to correctly classify actions not present in training data.

3 Methods

3.1 Zero-Shot Learning

Zero-Shot Learning is a form of two-stage learning. In the first stage, a mapping is learned from the training examples to the semantic space. These training examples may not contain examples of all possible labels. The second stage is a nearest neighbor classification — all labels (including those absent at training) are represented as points in the semantic space. Classification is done by applying the learned regressor and then finding the closest label point in the semantic space. Below we describe how we adapted this to the task of human action recognition.

We found a low-dimensional set of features to distinguish the labels (run, walk, etc). We designed the feature set based on our knowledge of human actions. A potentially better way to have selected this set would have been using text corpus statistics or surveys, but our feature set works reasonably well as a proof-of-concept. Thus, in our model, a 12-dimensional vector represents each action. These 12 features are discrete values corresponding to some canonical aspect of an action (e.g. 'Is there horizontal displacement?'). See section 3.6.2 for the complete list of these features.

In the preprocessing stage, we extract low-level features from the videos using the spatio-temporal interest points algorithm, a 3-dimensional generalization of Harris points. The points are selected as the local maxima of a particular function on the video which selects for large local variation in all directions [6]. We then compute an optical flow histogram (OFH) descriptor for each interest point. The x- and y-direction flow values in a neighborhood of a point are separately binned and the counts normalized. The two histograms are then concatenated to form the descriptor for each point. A single descriptor represents a whole video with a bag-of-features model. The OFH descriptors from all training videos are clustered by k-means, and any video then is represented by the normalized histogram of the clusters to which its OFH descriptors belong [7].

In the first stage of the classification process, a linear regressor is trained for each semantic feature to map from the low-level features to that semantic feature, using L2 regularized linear regression. The regularization parameters are chosen independently for each feature, using cross-validation. In the second stage, each action is represented by a point in this 12-dimensional feature space. To classify a video, the corresponding point is found, and the closest action point, by the Euclidean metric, is the label. See figure 1 for a summary. While the semantic space is not necessarily Euclidean, using Euclidean distance works in practice.

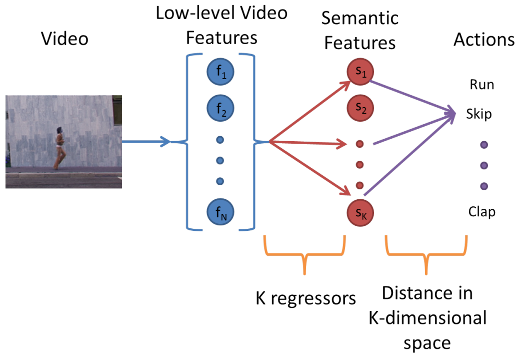

Figure 1: A summary of the two-stage learning process. The low-level video features are optical flow hisogram descriptor clusters. We learn one regressors that map the low-level features to a dimension in the semantic feature space. Each action label is represented by a canonical point in that space. To test a query point, we apply the regression and then perform nearest-neighbor classification in semantic space.

3.2 Leave-Two-Out Cross-Validation

To test the effectiveness of this two-step method over direct video-to-label mappings, we can perform leave-two-out cross-validation, and demonstrate the ZSL classifiers ability to distinguish between two previously unseen class labels based on their video data, a task that direct mappings are unable to perform.

First, the 12 regressors are trained on all but two of the actions. Then, a video of a previously unseen action is presented. This produces a query point in semantic space. The distance between the query point and the actual held-out action points is measured, and the point is classified. See figure 2 for a summary.

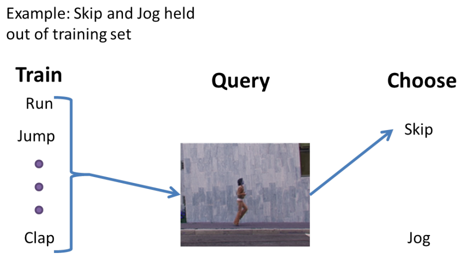

Figure 2: A diagram summarizing the leave-two-out cross-validation used to test zero-shot-learning. The classifier is able to tell whether a video is of skipping or jogging, despite having never seen an example of either action. We rate performance on a per-action basis as the percent of videos correctly classified for every possible two vs two pairing for that action.

For each action label, we were able to produce a distribution of percent of videos correctly classified versus opposing action. So, the results have this structure (numbers invented for the sake of example):

Action/Action	Run	Walk	Wave	...
Run	—-	.70	.90	...
Walk	.80	—-	.95	...
Wave	.90	.99	—-	...

This distribution can be compared to a simulated null distribution (described in Section 3.5) to determine significance of the results via paired t-test.

3.3 Rank Test

A more difficult task than leave-two-out cross-validation would be to compare the query point to all possible points in the semantic space. This allows us to determine if the ZSL classifier is biased toward items it has seen during training. In this experiment, we held one action out during training, and then had the classifier output the rank of the true action label when given a query action. All actions, even those held out in training, are still present in the semantic space as points. We calculated from this a rank accuracy with the following equation:

$$accuracy = 1 - \frac{r}{T} \tag{1}$$

Where r is the rank of the true label (with 0 representing the first rank) and T is the total number of action labels. This can be thought of as a measurement of the fraction of the labels that are ranked below the true label. A score of 1 is perfect. The experiment is summarized in Figure 3.

For each held-out action, we calculated the rank accuracy of each video of that action. This created a distribution that we compared against a simulated null distribution (see Section 3.5 for details) for statistical significance.

3.4 Feature Comparison

To see which semantic features yield the biggest advantage in decoding the action, we can test the leave-two-out cross-validation performance using only a subset of the semantic features in the intermediate set. This wil allow us to see if particular features are stronger predictors than others.

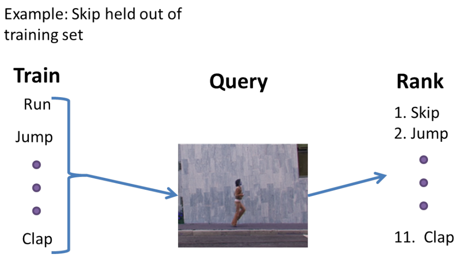

Figure 3: A diagram summarizing the rank test. The classifier ranks the potential action labels in order of nearness to the query point. Performance is measured with respect to the position of the true label.

In the first version of the experiment, we tried leaving each semantic feature out in turn and measuring the performance. See Figure 4 for an illustration.

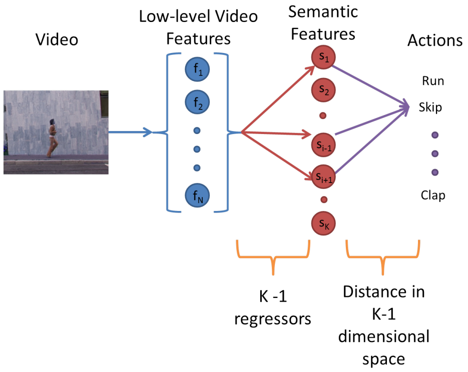

Figure 4: A diagram summarizing the feature comparison test. Each semantic feature was held out in turn and performance re-assessed using the method in Section 2.

We also tried leaving out subsets of features that seemed highly correlated. We chose the following groups:

Feature Group	Features Contained
Coordination	Arm and leg coordination
Center of Mass	Torso motion and orientation, Horizontal and vertical displacement
Hand	Hand location
Speed	Speed of center of mass, and speed of each limb

Table 1: Semantic feature groups left out, and features contained in those groups.

For each held out feature or group of features, we had a distribution of average leave-two-out cross-validation performance across actions. We compared this to the distribution found when using all features.

3.5 Statistical Significance

Each experiment generates a distribution of performance scores. The set of scores can be compared to chance with a paired t-test to establish significance.

We establish chance with a permutation test. We sever the connection between the videos and their semantic feature labels and shuffling the data. The semantic feature space points representing the acitons remained tied to their actions. See Figure 5 for an illustration.

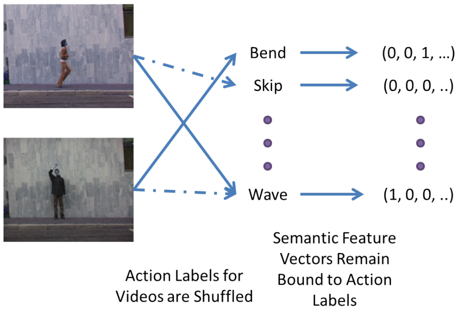
Bend
(0, 0, 1, ...)
Skip
(0, 0, 0, ..)
Wave
(1, 0, 0, ..)
Action Labels for Videos are Shuffled
Semantic Feature Vectors Remain Bound to Action Labels

Figure 5: Illustration of the permuting process. The action labels for each video in the training set were shuffled. Crucially, the action label semantic point representations were maintained so as to make a fair assessment.

Performance of our two-stage algorithm in the leave-two-out cross-validation and rank experiments on this shuffled set approximates chance performance. Performance was measured for 250 trials in the leave-two-out cross-validation performance experiment, and 1000 in the rank experiment.

3.6 Materials

3.6.1 Data Sets

Data was taken from the following two data sets: KTH [8] and Weizmann [9]. Both datasets contain videos for walking; these were combined and treated as a single class. The same applies to running. When training and testing, however, we balanced the training set by ensuring that each action was represented by the same number of videos (10). Unfortunately, due to time constraints we were only able to randomize this balancing process for the leave-two-out cross-validation experiment (250 trials), and the permuted portion of the rank experiment (1000 trials). For the feature analysis experiments we worked with the complete set of videos (110).

3.6.2 Semantic Features

The following are the semantic features that we selected, with values and explanations:

1. $Coord\_arms$: Coordination of the arms. (Discrete 0–2) 0 is no coordination, 1 is coordination, 2 is anti-coordination.
2. $Coord\_legs$: Coordination of the legs. (Discrete 0–2) 0 is none, 1 is coordination, 2 is anti-coordination.
3. $Torso$: (Binary) 1 if torso is moving relative to the rest of the body.
4. $Orient\_torso$: : (Discrete 0–2) 0 is that torso orientation is unrelated to the motion, 1 is that the torso is canonically parallel to motion, 2 is that the torso is canonically perpendicular to motion.
5. $Vert$: (Discrete 0–5) amount of vertical displacement of the center of mass.
6. $Horiz$: (Binary) 1 if there is whole body horizontal displacement.
7. $Hand$: Hand location. (Binary) 1 if the hand is lower than the center of mass, 0 if it is higher.
8. $Speed$: (Discrete 0–5) speed of center of mass.
9. $Limbs$: (4 features, each Discrete 1–5) speed of each of the four limbs.

3.6.3 Coding Libraries

This project was coded using Python. Basic libraries used include, SciPy [10], NumPy [11], OpenCV [12], and The Python Image Library [13]. Ridge regression and cross-validation code taken from Scikit-learn [14]. Figures generated with MATLAB [**?**].

4 Results

4.1 Leave-Two-Out Cross-Validation and Rank Performance

Leave-two-out cross-validation performance was significantly above chance for most actions, as summarized in Figure 6. Performance averaged across actions was also statistically significant.

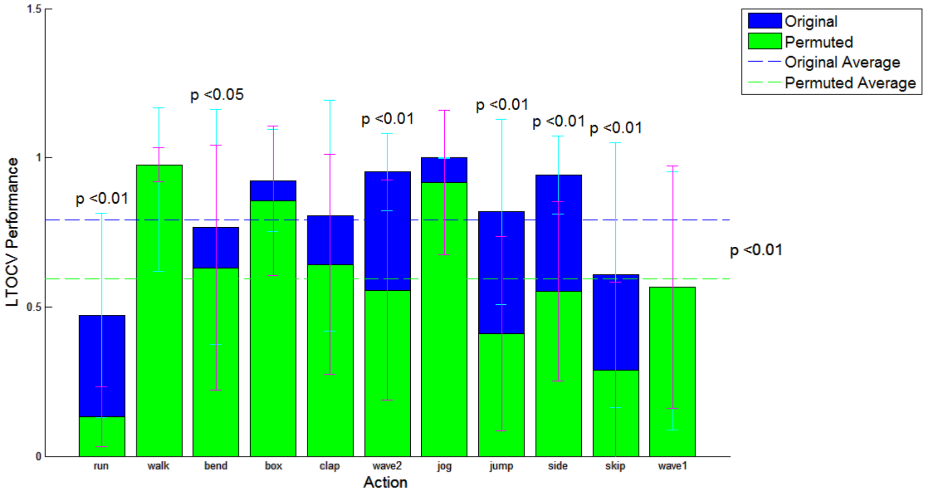

Figure 6: Results when comparing leave-two-out cross-validation (LTOCV) performance on the data set with labels intact (blue), and on the data set with labels permuted (green). Error bars are ± the standard deviation of the distribution of two vs two pairs.

Chance performance may seem high for two of the actions. This can be attributed to flaws in the semantic feature space, as there are some actions that, by nature of their location in the space, have a higher tendency to be selected.

Rank performance was also very significantly above chance for most actions, as well as for performance averaged across action. See Figure 7.

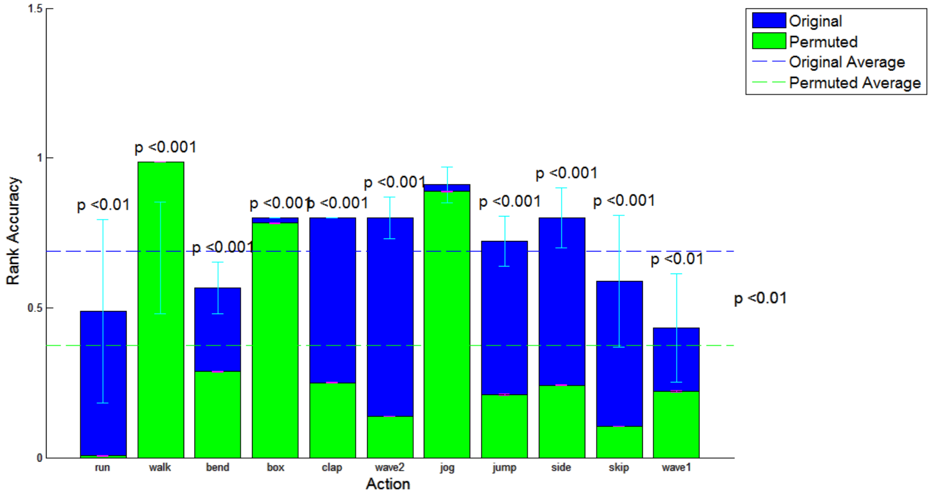

Figure 7: Rank test performance for the original (blue) and permuted (green) versions of the data set. Error bars are ± the standard deviation of the distribution of rank scores for each video of a given action.

The variance of these results are noticeably lower than those of the leave-two-out cross-validation results, perhaps due to the nature of the score (there is a strict bound on the variance of a distribution of rank scores). It also speaks to the consistency of the model that videos from the same action label tend to produce the same rank, even if that rank is not correct.

Unsurprisingly, actions that had high chance performance in leave-two-out cross-validation again had relatively high chance performance in this setting. An additional line of reasoning that can explain these results is the non-Euclidean nature of the semantic space. While using Euclidean distance clearly works reasonably well, more sophisticated notions of classification in this domain might yield better results in the form of greater difference between true and chance performance.

4.2 Feature Comparison

As is shown in Figure 8, no one feature is so important that its absence significantly affects performance, expecting the speed of the right arm. However it is easy to see why, given that waving only involves that feature, it would hurt performance to not have it.

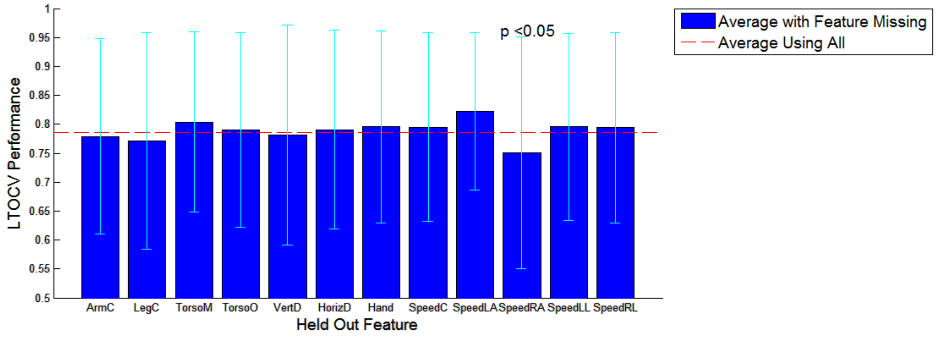

Figure 8: Performance of leave-two-out cross-validation versus held-out semantic feature. Average performance of the complete model is given by the dashed line. Error bars are ± standard deviation of the distribution over two vs two pairs.

We also tried leaving out families of features, and the results are shown in Figure 9. This too did not affect performance significantly. It would appear that even removing highly correlated groups of features does not affect the classification results.

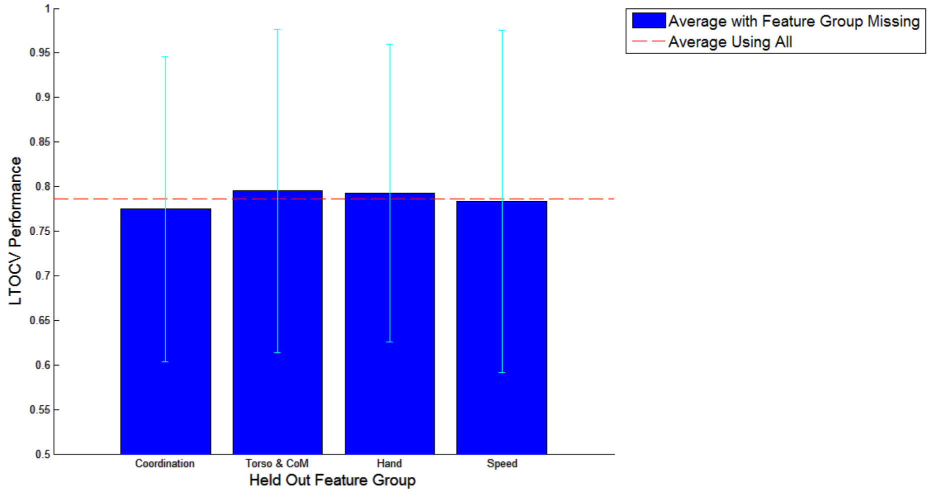

Figure 9: Performance of leave-two-out cross-validation versus held-out semantic feature group (see Table 3.4). Average performance of the complete model is given by the dashed line. Error bars are ± standard deviation of the distribution over two vs two pairs.

5 Conclusion

Our results demonstrate a proof-of-concept for zero-shot learning in the domain of human action recognition from video. This technique allows extrapolation from labels in the training data set, so that we can classify videos with labels we have not yet seen. Performance tended to vary depending on semantic feature and action, but was significantly better than noise in most cases. With a better-developed semantic feature base, we could see an improvement in results.

References

[1] M. Palatucci, D. Pomerleau, G. Hinton, and T. Mitchell, "Zero-shot learning with semantic output codes," in *Neural Information Processing Systems (NIPS)*, December 2009.

[2] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976 – 990, 2010.

[3] J. Vogel and B. Schiele, "Semantic modeling of natural scenes for content-based image retrieval," *Int. J. Comput. Vision*, vol. 72, pp. 133–157, Apr. 2007.

[4] S. Park and J. Aggarwal, "Semantic-level understanding of human actions and interactions using event hierarchy," in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, p. 12, june 2004.

[5] Q. Zhao, Z. Lu, and H. H. S. Ip, "Action recognition based on learnt motion semantic vocabulary," in *Proceedings of the 11th Pacific Rim conference on Advances in multimedia information processing: Part I*, PCM'10, (Berlin, Heidelberg), pp. 193–202, Springer-Verlag, 2010.

[6] I. Laptev, "On space-time interest points," *Int. J. Comput. Vision*, vol. 64, pp. 107–123, Sept. 2005.

[7] I. Laptev and T. Lindeberg, "Local descriptors for spatio-temporal recognition," in *In First International Workshop on Spatial Coherence for Visual Motion Analysis*, 2004.

[8] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *International Conference on Pattern Recognition (ICPR)*, August 2004.

[9] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 2247–2253, December 2007.

[10] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001–.

[11] P. F. Dubois, K. Hinsen, and J. Hugunin, "Numerical python," *Computers in Physics*, vol. 10, May/June 1996.

[12] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[13] PythonWare, "Python Imaging Library (PIL).".

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python ," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.