Slow and steady feature analysis: higher order temporal coherence in video (Supplementary material)

Dinesh Jayaraman
UT Austin
dineshj@cs.utexas.edu

Kristen Grauman
UT Austin
grauman@cs.utexas.edu

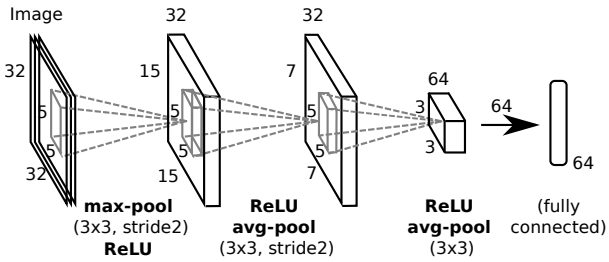


Figure 1: 32×32 CNN architecture used for the KITTI→SUN and HMDB→PASCAL-10 tasks

1. Appendix

We now provide supplementary details on (1) the CNN architecture used in our SUN and PASCAL-10 experiments, (2) the sequence completion task used to quantify steadiness, (3) our experiments with varying sizes of unsupervised training datasets, (4) our experiments with purely unsupervised feature learning, (5) pre-processing steps for the datasets used in our experiments, (6) optimization-related details, and (7) details of the supervised pretraining and finetuning baseline SUP-FT from the paper. We also show samples of all the real image datasets used in our experiments.

32×32 images CNN architecture: The 32×32 CNN architecture [1] representing $\mathbf{z}_{\theta}$, used for the KITTI→SUN and HMDB→PASCAL-10 tasks is shown in Fig 1.

Quantifying steadiness - details As described in the main paper (Sec 4.2), the candidate set $\mathcal{C}$ for NORB was straightforward to construct – the entire NORB test image set was used. For the video datasets KITTI and HMDB though, it would have been practically difficult to include all image frames in the candidate set $\mathcal{C}$. To avoid having to compute features and perform nearest neighbor search over too large a number of frames, we formed a randomly sub-sampled $\mathcal{C}$ instead, as follows. Starting from empty $\mathcal{C}$, we added (1) all the unique images among the query pairs (2) their corresponding ground truth completion images and (3) a minimum number N of randomly chosen frames from each video represented within $\mathcal{C}$ until this point. This ensures that the task is non-trivial by adding distractors from the same video as the ground truth candidate image, which are likely to have similar appearance. We used N=10 for KITTI and N=5 for HMDB to keep the total numbers of images manageable. Finally, we select from $|\mathcal{C}|$ =8100, 5000 and 5000 candidates respectively for NORB, KITTI and HMDB, for each of N =20,000, 1000 and 1,000 query pairs respectively for the three datasets.

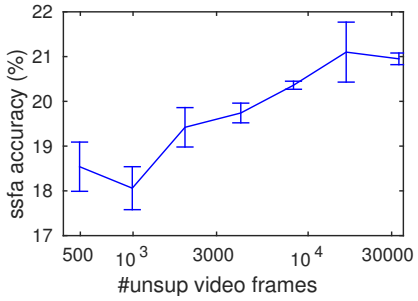


Figure 2: SSFA classification accuracy vs. duration of unsupervised video (mean, standard error over 5 runs).

Varying unsupervised training set size: To observe the effect of unsupervised training set size, we now restrict SSFA to use varying-sized subsets of unlabeled video on the HMDB→PASCAL-10 task. The full HMDB dataset has approximately 1000 videos, for a total of ≈32000 frames. Performance scales roughly log-linearly with the duration of video observed as shown in Fig 2, suggesting that even larger gains may be achieved simply by training SSFA with more freely available unlabeled video.

Purely unsupervised feature learning: We evaluate the usefulness of features trained to optimize the unsupervised SSFA loss L_u (main paper Eq () alone. Features trained on HMDB are evaluated at various stages of training, on

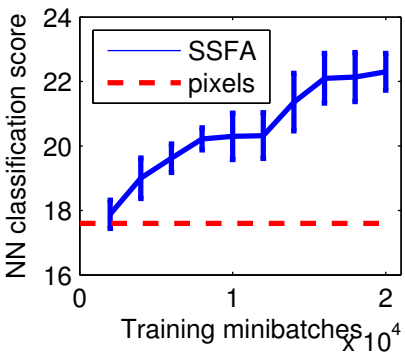


Figure 3: SSFA k-NN accuracy improvement with SSFA training (mean, standard error over 5 runs).

the task of k-nearest neighbor classification on PASCAL-10 (k =5, and 100 training images per action). Fig 3 shows the results. Starting at $\approx$ 17.8% classification accuracy for randomly initialized networks, unsupervised SSFA training steadily improves the discriminative ability of features. This shows that SSFA can train useful image representations even without jointly optimizing a supervised objective.

Dataset pre-processing details For all tasks, images are mean-subtracted and contrast-normalized before passing to the neural networks. In addition, for KITTI→SUN, full KITTI frames were resized to 32×32 and SUN images were cropped to KITTI aspect ratio before resizing to the same dimensions. Grayscale images were used in this task. Similarly, for HMDB→PASCAL-10, HMDB frames were cropped to centered squares, and PASCAL-10 bounding boxes were expanded to the closest square before resizing to 32×32. Resizing for KITTI→SUN and HMDB→PASCAL-10 was done to allow fast and thorough experimentation with standard CNN architectures known to work well with tiny images [1]. On the SUN dataset apart from resizing, where we also lose information due to KITTI-aspect-ratio cropping, we verified that our baselines were legitimate by running a simple nearest neighbor baseline in the pixel space (standard approach for tiny images). This achieved 0.61% accuracy compared to UNREG's 0.70%, given the same training data.

Optimization details We initialized according to the scheme proposed in [2], and run Nesterov accelerated stochastic gradient descent using the open source Caffe [3] package. The base learning rate and regularization λs are selected with greedy cross-validation.[1] Specifically, for each task, the optimal base learning rate (from 0.1, 0.01, 0.001, 0.0001) was first identified for UNREG. Next λ was set through a logarithmic grid search (steps of $10^{0.5}$), with λ' set to 0 *i.e.* this parameter was optimized for SFA-2. The margin parameter δ of the contrastive loss in $R_2(.)$ was set to 1.0 for all methods – this affects the objective function only up to a feature scaling operation, and so may be set to any positive value. For SSFA, a similar search was then performed over λ' (logarithmic grid search with steps of $10^{0.5}$), and then a small search for the contrastive loss margin δ in $R_3(.)$ (over 0, 0.1 and 1). Setting the margin to $\delta = 0$ in a contrastive loss reduces it to the simple distance loss over positive samples.

On a single Tesla K-40 GPU machine, NORB→NORB training tasks took ≈30 minutes, KITTI→SUN tasks took $\approx$ 90 minutes, and HMDB→PASCAL-10 tasks took ≈60 minutes. SSFA training took about 2x training time and 1.5x training epochs to converge, compared to SFA baselines, because of the more complex loss function.

Supervised pretraining and finetuning - details For the supervised pretraining and finetuning comparison experiments in Sec 4.3, we used the same neural network architecture as used for our approach and other baselines on the SUN scene and PASCAL-10 action recognition tasks (architecture shown in Fig 1). A 100-way softmax classifier was trained on the 64-dimensional final layer features to classify CIFAR-100 classes during pretraining, but these classifier weights are ignored for supervised transfer. All other weights in the network are used to set the corresponding weights on the network to be trained for the target task. For SUN (397 classes x 5 images per class), we found it beneficial to finetune features by reducing the learning rate for the pretrained layers by a factor of 0.1 compared to the full learning rate used to train the 397-way classifier on top. For PASCAL-10 (10 classes x 5 images per class), only the 10-way action classifier was trained starting from random weights, while the weights of lower layers were frozen to their pretrained values, since finetuning was found to adversely impact classification results.

Dataset sample images Some sample images of KITTI, SUN, HMDB-51 and PASCAL-10 are shown at the end of this document.

[1] our validated (λ,λ') values for NORB→NORB, KITTI→SUN, and HMDB→PASCAL respectively are (0.1,0.3),(3,0.1), and (0.3,1)

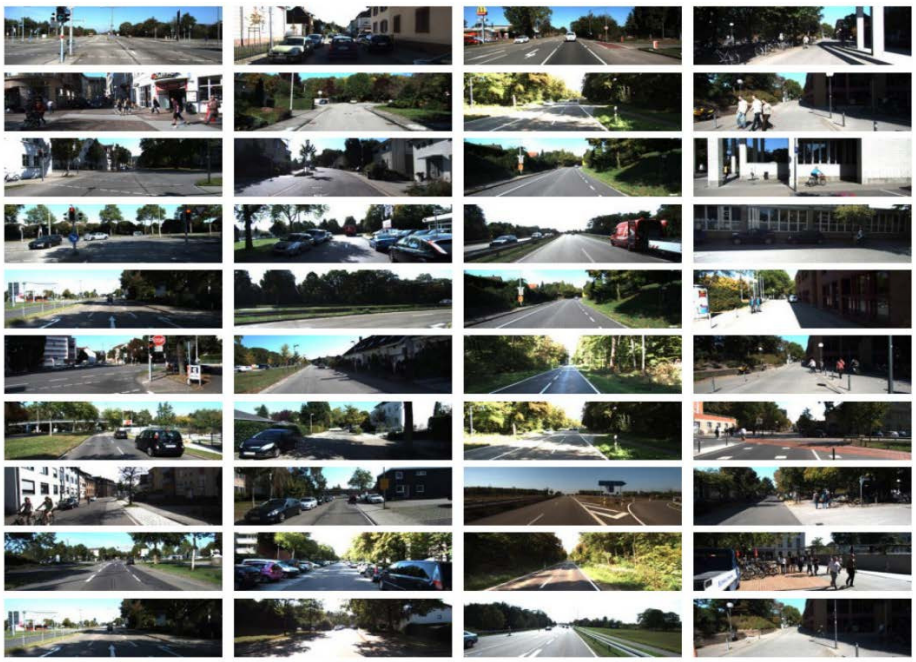
References

[1] Cuda-convnet. https://code.google.com/p/cuda-convnet/. 1, 2

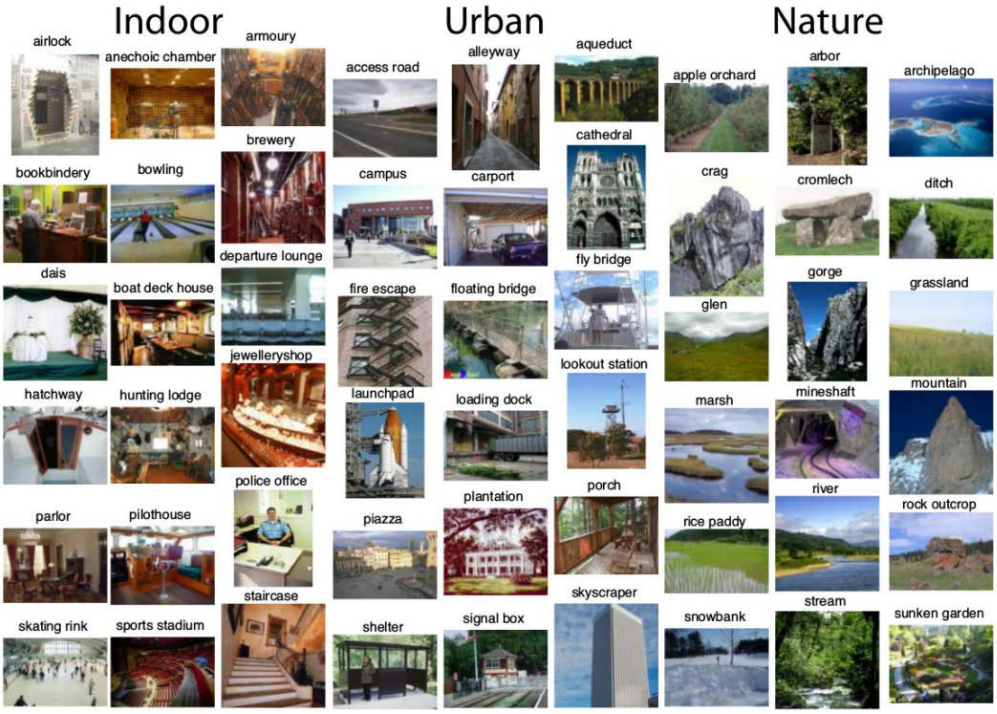
[2] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. *AISTATS*, 2010. 2

[3] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, Sergio S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv*, 2014. 2

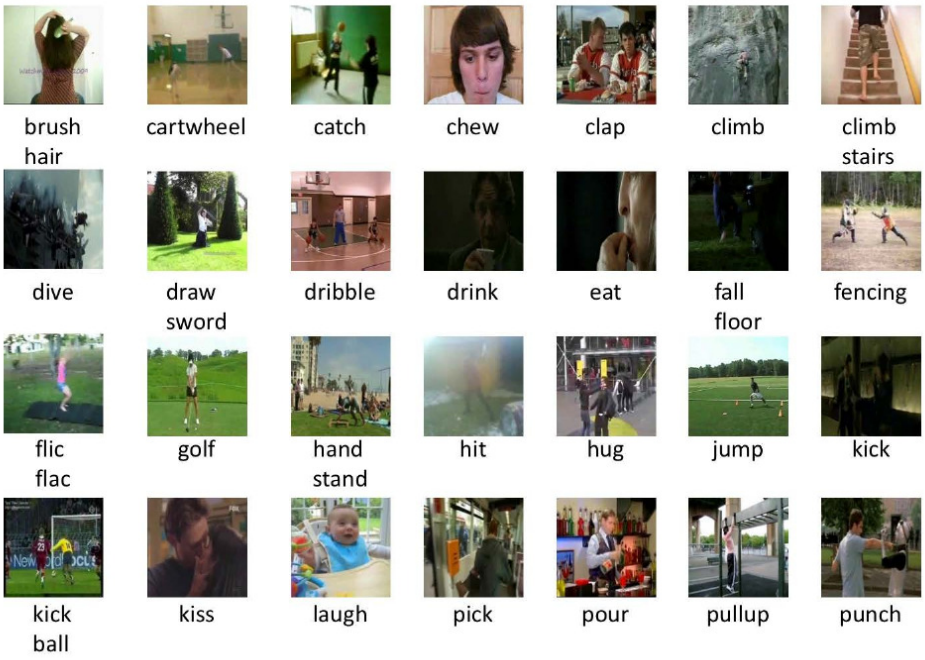
KITTI driving video



SUN scene images



HMDB-51 actions video



PASCAL-10 action still images