Learning Better Image Representations Using ‘Flobject Analysis’: Supplementary Material

Patrick S. Li*, Inmar E. Givoni*, Brendan J. Frey
University of Toronto
{pli,inmar,frey}@psi.utoronto.ca

Detailed Flobject Pipeline

There are four major stages in the flobject pipeline: preprocessing, flobject analysis, creating image descriptors, and classification. Before any analysis is done, video frame pairs and static images are first preprocessed and reduced to a suitable representation during the preprocessing stage. Next the unsupervised flobject analysis stage takes as input a collection of video frame pairs and the number of topics to learn, and returns as output a distribution over codewords for each topic. After analysis, the learnt per-topic distributions over codewords are used to create an image descriptor for each static image that is useful for classification tasks. Finally, in the last stage, the computed image descriptors for a collection of labelled training images and unlabelled test images are fed into a standard classifier to predict the object class for the test images.

Preprocessing

Before analysis, all video frame pairs and static images are preprocessed and reduced to a suitable representation. Video frame pairs are represented as a collection of appearance features each with a corresponding flow feature. Appearance features are represented as integer indices. Flow features are represented as two dimensional real valued vectors. Static images are represented as a collection of appearance features.

For static images, we convert the image to grayscale by averaging the colour channels, and scale it down to a size of 216x384 pixels. Histogram-of-oriented-gradients (HOG) features for every 16x16 pixel patch lying on a 6x6 pixel grid are extracted from the grayscale image. The HOG features in the entire training image collection are discretized into an integer index between 1 and W, called a ‘codeword’, using online k-means clustering. Thus, a single static image is represented as an array of integers between 1 and W.

For video frame pairs, we extract the appearance features for the first frame in the pair using the same procedure as for the static images. To compute the flow for each feature, we first compute a dense per-pixel optical flow field for the video frame pair using [1]. The flow associated with a particular appearance feature is computed by averaging the per-pixel optical flow over its 16x16 patch with a gaussian weighting window. Thus, a single video frame pair is represented as a collection of appearance features (an array of integers between 1 and W), and a corresponding collection of flow features (an array of two dimensional real valued vectors).

For our experiments, the vocabulary size W for the CityCars dataset was chosen to be 1000 as is standard in the literature. The vocabulary size W for the CityPedestrian dataset was chosen to be 400 using cross validation over W = 200, 400, 1000, and 2000.

Details of HOG Extraction: For computing the HOG features, we use gradient histograms of eight angle bins spanning 0 to 360 degrees. First, the 16x16 pixel patch is subdivided into four 4x4 subpatches. An 8-dimensional gradient histogram is extracted from each subpatch and then concatenated with the 8-dimensional gradient histogram extracted from the whole 16x16 patch to form a 40-dimensional HOG feature for the patch. The gradient histograms for the subpatches are downweighted by a factor of two before the entire HOG feature is L1 normalized to create the final HOG descriptor [2].

Flobject Analysis

Given a collection of images, each represented as a collection of appearance and flow features, and the number of topics to learn, we run FLDA for 32000 Gibbs sampling iterations to learn the topic distributions. For example, learning two topics will result in two topic distributions, each a W dimensional vector which sums to one indicating the likelihood of each codeword under that topic. We use FLDA parameters $\alpha = 20$, $\beta = 20$, $\gamma = 10$, and $L = 4$. For the prior over the flow, we set $\mu_0 = 0$, and $\Lambda_0 = 2I$, $\kappa_0 = 20$, $\nu_0 = 2$, where I is the two dimensional identity matrix.

The number of topics used for analysis as reported in Table 2 and 3 for the CityCars dataset, and in Table 6 for the non-hierarchical descriptors for the CityPedestrians dataset

*These authors contributed equally.

was $T = 2$. The number of topics reported in Table 6 for the hierarchical descriptors was chosen to be $T = 10$ using cross validation over [2-10] topics.

Creating an Image Descriptor

Two different descriptors were experimented with in the paper, standard FLDA descriptors and Hierarchical-FLDA descriptors.

FLDA Descriptors: With a single sample from FLDA, which gives the topic assignments for each feature in each video frame pair in the training collection, we can compute the maximum *a posteriori* (MAP) estimate of ϕ. Given a static image represented as a collection of appearance features, we run LDA for 600 Gibbs iterations with ϕ fixed to its MAP estimate learnt during flobject analysis. This results in a topic assignment for each feature in every image. For each topic in each image, we compute the histogram over codewords of the features assigned to each topic to create topic-specific histograms. The topic-specific histograms are then individually normalized and concatenated together with the histogram over codewords of all features in the image to form the final FLDA descriptor for that image.

For example, if we learnt two topic distributions during flobject analysis, each image will be described by a $3W$-dimensional vector, where dimensions 1 to W is the histogram of all features in the image, dimensions W+1 to $2W$ is the histogram of features assigned to topic 1, and dimensions $2W$+1 to $3W$ is the histogram of features assigned to topic 2.

Hierarchical-FLDA Descriptors (H-FLDA): Similarly, in the case of FLDA Descriptors, we compute the maximum *a posteriori* estimate of ϕ from a single sample from FLDA. For a given image, the H-FLDA descriptor is a T dimensional vector that sums to one, where T is the number of topics obtained during FLDA. This descriptor is created by scanning a 10x10 window over the image, and computing a W-dimensional histogram over codewords for each window. This window is assigned to the topic that is closest in Euclidean distance to its histogram. For each topic, the total number of windows assigned to it is computed and this histogram is normalized to obtain the final descriptor.

Classification

Given a collection of labelled training images and unlabelled test images, where all images are either represented as FLDA descriptors or H-FLDA descriptors, we can use a standard classifier to classify test images. We experiment with a nearest neighbour classifier using L2 distance and the intersection kernel.

References

[1] A Bruhn, J Weickert, C Schnörr. Lucas/kanade meets horn/schunck: combining local and global optic flow methods. *IJCV*, 2005.

[2] A Bosch, A Zisserman, X Munoz. Representing shape with a spatial pyramid kernel. *CIVR*, 2007.