

Learning Better Image Representations Using ‘Flobject Analysis’

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

Abstract

Unsupervised learning can be used to extract image representations that are useful for various and diverse vision tasks. After noticing that most biological vision systems for interpreting static images are trained using disparity information, we developed an analogous framework for unsupervised learning. The output of our method is a model that can generate a vector representation or descriptor from any static image. However, the model is trained using pairs of consecutive video frames, which are used to find representations that are consistent with optical flow-derived objects, or ‘flobjects’. To demonstrate the flobject analysis framework, we extend the latent Dirichlet allocation bag-of-words model to account for real-valued word-specific flow vectors and image-specific probabilistic associations between flow clusters and topics. We show that the static image representations extracted using our method can be used to achieve higher classification rates and better generalization than standard topic models, spatial pyramid matching and gist descriptors.

1. Introduction

A promising direction of vision research is to develop unsupervised learning algorithms that can extract concise representations of images, and then use those representations as inputs for supervised learning tasks, such as object classification, scene recognition and image segmentation. The unsupervised step may rely primarily on hand-crafted features (eg [11]) or may make more extensive use of machine learning techniques and hierarchical probability models (eg [13, 8, 10, 14, 5, 9, 15, 16, 2, 18, 7, 6]).

Here, we study how unsupervised learning can be used to obtain good representations of static images. A common approach is to apply increasingly sophisticated machine learning techniques to training sets of static images. Alternatively, one can examine the best vision systems that we know of, namely biological ones, and ask what kinds of

*These authors contributed equally.

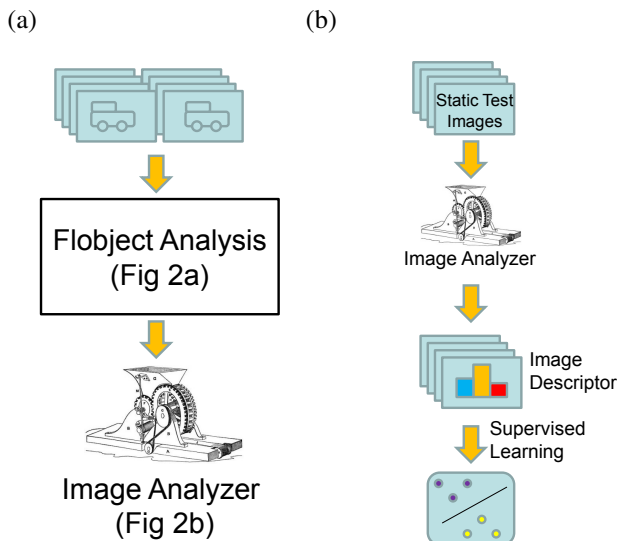


Figure 1: (a) Flobject analysis uses pairs of consecutive video frames to produce an image analyzer. (b) The image analyzer can be used to derive representations, or descriptors, from *static images* for object recognition.

information those systems may be using to learn image representations. While biological vision systems are capable of achieving high classification rates on static images, most of those systems have motion and/or stereo information as additional input during learning. In fact, motion cues have been found to be important for developing static image representations [24]. This led us to ask:

Can motion and/or stereo disparity information be used to train better methods for extracting representations from static images?

To answer this question, we explore a two-stage approach. In the first stage (Fig. 1a), consecutive pairs of video frames are used to train a model that can infer representations from static images so that those representations are consistent with optical flow patterns. We refer to this stage as *flobject analysis*, since the goal is to learn a representation that is consistent with optical flow-derived ob-

jects, or *flobjects*. Once learnt using pairs of frames, the model can be applied to static images *without* using optical flow, so we refer to it as an *image analyzer*. To explore the usefulness of this approach, in the second stage we use the learnt image analyzer to extract representations from labeled static images and examine the performance of supervised classification (Fig. 1b).

Each image pair is preprocessed to obtain a collection of appearance features \mathcal{X} from the first image, and a corresponding collection of flow features \mathcal{V} (flow vectors) extracted from the image pair. Flobject analysis entails inferring a model \mathcal{M} that uses hidden variables \mathcal{H} to explain the appearance and flow features. For example, \mathcal{H} could describe clusters of visual words [11], whole-image transformations [13], transformations of sub-images containing putative object parts [12], colour invariant object appearance [7], visual word topics [17], topics accounting for spatial distributions of visual words [8, 10, 9], or statistical relationships between features in a deep belief network [15].

For N training cases $(\mathcal{X}^1, \mathcal{V}^1), \dots, (\mathcal{X}^N, \mathcal{V}^N)$, the posterior distribution over models is obtained by integrating over the hidden variables:

$$P(\mathcal{M}|\{\mathcal{X}^i, \mathcal{V}^i\}) \propto P(\mathcal{M}) \prod_{i=1}^N \int_{\mathcal{H}_i} P(\mathcal{X}^i, \mathcal{V}^i, \mathcal{H}^i|\mathcal{M}) \quad (1)$$

where $P(\mathcal{M})$ is the prior distribution over models. The output of flobject analysis is a maximum *a posteriori* (MAP) model $\mathcal{M}^{\text{MAP}} = \arg \max_{\mathcal{M}} P(\mathcal{M}|\{\mathcal{X}^i, \mathcal{V}^i\})$, or a sample from the posterior distribution over models $\mathcal{M} \sim P(\mathcal{M}|\{\mathcal{X}^i, \mathcal{V}^i\})$, which can be used to hedge bets depending on model uncertainty.

The static image analyzer is obtained by integrating over all possible flow patterns that are consistent with the model's explanation of the static image appearance features \mathcal{X} : $P(\mathcal{X}, \mathcal{H}|\mathcal{M}) = \int_{\mathcal{V}} P(\mathcal{X}, \mathcal{V}, \mathcal{H}|\mathcal{M})$. A particular form of appearance-flow model significantly simplifies this integral. If we assume that the hidden variables entirely capture the dependencies between the appearance features and flow features, we can factorize the model: $P(\mathcal{X}, \mathcal{V}, \mathcal{H}|\mathcal{M}) = P(\mathcal{X}|\mathcal{H}, \mathcal{M}^{\mathcal{X}})P(\mathcal{V}|\mathcal{H}, \mathcal{M}^{\mathcal{V}})P(\mathcal{H}|\mathcal{M}^{\mathcal{H}})$. The above integral becomes trivial and we obtain

$$P(\mathcal{X}, \mathcal{H}|\mathcal{M}) = P(\mathcal{X}|\mathcal{H}, \mathcal{M}^{\mathcal{X}})P(\mathcal{H}|\mathcal{M}^{\mathcal{H}}). \quad (2)$$

That is, to evaluate a static image, we simply remove the flow-based part of the model without needing to perform any correctional computations.

Depending on the vision task, the distribution over the hidden variables is converted to an image representation, or descriptor d , $P(\mathcal{H}|\mathcal{X}, \mathcal{M}) \rightarrow d$. For a supervised classification task, the descriptor is extracted for each image in a labeled training set and the descriptor-label pairs are used for supervised training as in Fig. 1b. The MAP estimate of the hidden variables can be used as the descriptor, $d = \arg \max_{\mathcal{H}} P(\mathcal{H}|\mathcal{X}, \mathcal{M})$, in which case d has the same

length as \mathcal{H} . Instead, it may be desirable to use only part of the hidden representation as a descriptor or to further compute summary statistics.

2. A topic model for flobject analysis

There are many different models and algorithms that can be used for flobject analysis. Here we show how a novel extension of the standard latent Dirichlet allocation topic model [1] can be applied. We refer to the extended model as flow-based LDA (FLDA).

Each input image pair is preprocessed by computing an optical flow field and extracting regularly sampled HOG appearance features from the first of the two images (Fig. 2a). Each HOG appearance feature is mapped to a discrete codeword from a codebook that was obtained using k -means clustering, resulting in a bag of appearance and flow features \mathcal{X}, \mathcal{V} . Unsupervised learning takes as input the appearance and flow features for N training cases and produces a set of topics \mathcal{M} described by codeword distributions, and, for each training case, assigns every appearance feature to one of those topics in such a way that appearance features assigned to the same topic tend to have similar flow. These assignments along with the image-specific topic mixing proportions form the hidden representation \mathcal{H} .

One property of FLDA is that the flow and its associated hidden variables can be easily integrated out of the model, leading in a straightforward fashion to the image analyzer. In fact, the image analyzer is a standard LDA model with the topic distributions set using FLDA. Application of the image analyzer (Fig. 2b) entails extracting a bag of appearance features (without flow) and applying LDA with the FLDA-learnt topics to obtain a factorized set of topic-specific appearance feature histograms.

2.1. The flow-based latent Dirichlet allocation (FLDA) model

A good flobject analysis technique needs to take into account (1) that an object may have different motion patterns in different images; (2) that there may be multiple moving objects in the same image, including different objects moving with different velocities or with the same velocity; (3) that a single object may have multiple flow patterns within an image, *eg*, due to articulated parts; (4) that the number of objects in an image may vary; (5) that building a good object model requires aggregating information across images; and (6) that optical flow is often highly erroneous or noisy.

The model we present here, flow-based latent Dirichlet allocation (FLDA), is an extension of LDA [1] that accounts for the flow information in a principled manner, while addressing the above desiderata. Standard LDA takes a collection of documents (corresponding to images), each represented as a collection of words that take on values from a codebook. LDA infers a set of topics shared across the

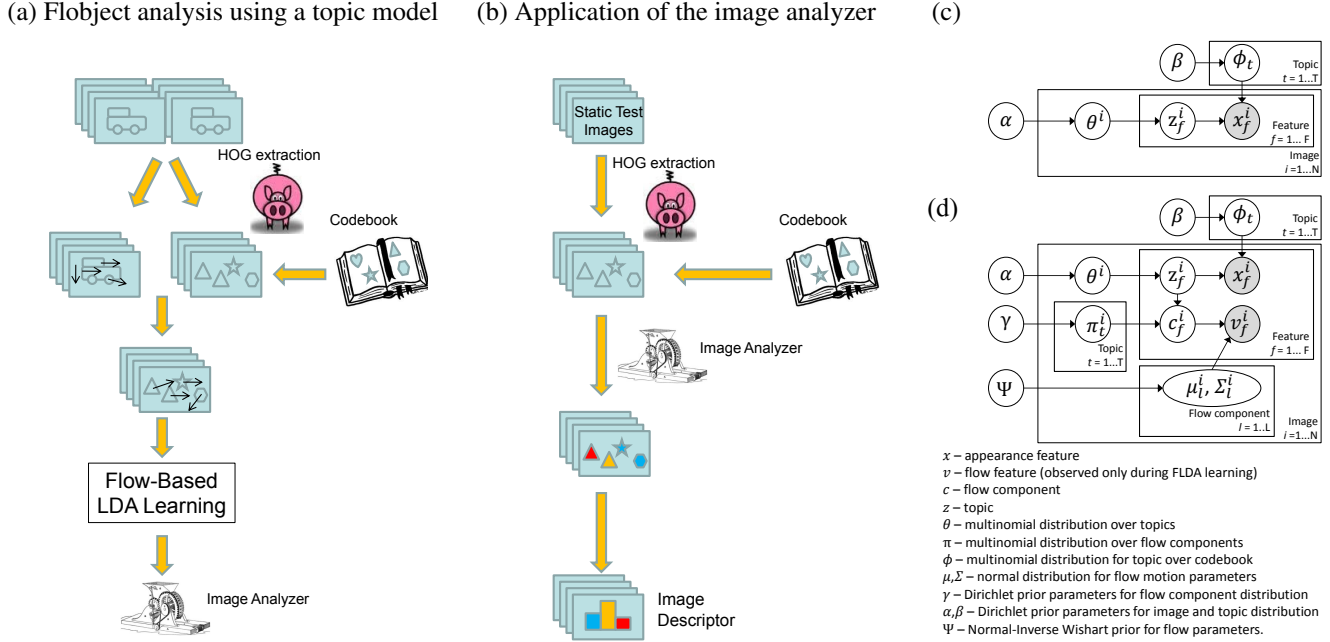


Figure 2: Fobject analysis can be performed using an appropriately extended bag-of-features topic model (a), and the resulting image analyzer can be used to factor the histogram of appearance features for a static image into a set of topic-specific histograms (b). Our method, flow-based LDA (FLDA) can be viewed as an extension of LDA (c) that incorporates real-valued feature-specific flow vectors and image-specific probabilistic associations between flow clusters and topics (d).

documents such that each topic is represented as a distribution over codewords, and each document is associated with a distribution over topics. In LDA (Fig. 2c), the t th topic is represented by a multinomial distribution over codewords ϕ_t , which is assumed *a priori* to have a Dirichlet distribution. For each image, appearance features are generated by first sampling a multinomial distribution θ over topics from another Dirichlet distribution. Next, the topic z_f for the f th feature in the image is sampled from θ . Finally, the codeword x_f for the f th feature is sampled from the multinomial distribution ϕ_{z_f} .

FLDA uses flow information to guide the creation of the topics such that features with similar flow in an image are more likely to have the same topic assignment. As a result, the codewords corresponding to these features are likely to co-occur in the same topic. However, since multiple objects may have similar, or even identical, flow, we do not force features with similar optical flow to have the same topic, but rather allow the algorithm to flexibly account for both alternatives. Each feature f is associated with both the appearance feature codeword x_f and an optical flow vector $v_f \in \mathcal{R}^2$. We augment the LDA model to account for the flow, as shown in Fig. 2d. In order to accommodate the above desiderata, the model allows for a collection of flow components to be available for every image. Each topic is associated with a distribution π_t over these components.

Given the topic assignment z_f for feature f , a component c_f is drawn from π_{z_f} . The flow is then generated from a Gaussian distribution with image specific mean and variance parameters μ_{c_f}, Σ_{c_f} .

We use a fully conjugate model with symmetric Dirichlet priors for all multinomial distributions, and a normal inverse-Wishart prior for the flow component normal bivariate distribution. We use the notation $\Theta = \{\alpha, \beta, \gamma, \Psi\}$ where $\Psi = \{\mu_0, \Lambda_0, \kappa_0, \nu_0\}$ are the normal inverse-Wishart parameters. Bold notation for Greek letters denotes a collection of variables across images, topics, or both.

The joint distribution is

$$P(\mathcal{Z}, \mathcal{C}, \mathcal{X}, \mathcal{V}, \boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\phi} | \Theta) = P(\boldsymbol{\phi} | \beta) P(\boldsymbol{\mu}, \boldsymbol{\Sigma} | \Psi) \cdot \prod_i P(\theta^i | \alpha) P(\pi^i | \gamma) P(\mathcal{Z}^i, \mathcal{C}^i, \mathcal{X}^i, \mathcal{V}^i | \boldsymbol{\phi}, \theta^i, \pi^i, \boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i) \quad (3)$$

where

$$P(\boldsymbol{\phi} | \beta) = \prod_t P(\phi_t | \beta), \quad \phi_t | \beta \sim \text{Dir}(\beta) \quad (4)$$

$$P(\boldsymbol{\mu}, \boldsymbol{\Sigma} | \Psi) = \prod_i \prod_c P(\mu_c^i, \Sigma_c^i | \Psi), \quad (5)$$

$$\mu_c^i, \Sigma_c^i | \Psi \sim \mathcal{NW}^{-1}(\mu_0, \frac{\Lambda_0}{\kappa_0}, \nu_0, \Lambda_0) \quad (6)$$

$$\theta^i | \alpha \sim \text{Dir}(\alpha) \quad (7)$$

$$P(\boldsymbol{\pi}^i | \gamma) = \prod_t P(\pi_t^i | \gamma), \quad \pi_t^i | \gamma \sim \text{Dir}(\gamma) \quad (8)$$

$$\begin{aligned} & P(\mathcal{Z}^i, \mathcal{C}^i, \mathcal{X}^i, \mathcal{V}^i | \boldsymbol{\phi}, \boldsymbol{\theta}^i, \boldsymbol{\pi}^i, \boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i) \\ &= \prod_{f,t,l} P(z_f^i=t, c_f^i=l, x_f^i, v_f^i | \boldsymbol{\phi}_t, \boldsymbol{\theta}^i, \boldsymbol{\pi}_t^i, \boldsymbol{\mu}_l^i, \boldsymbol{\Sigma}_l^i)^{[z_f^i=t], [c_f^i=l]} \\ &= \prod_{f,t,l} (\theta_t^i \pi_{t,l}^i \phi_{t,x_f^i} \mathcal{N}(v_f^i; \boldsymbol{\mu}_l^i, \boldsymbol{\Sigma}_l^i))^{[z_f^i=t], [c_f^i=l]}. \end{aligned} \quad (9)$$

The flow model encourages appearance features that have similar flow to come from the same topic. If the model were changed so that each image had its own set of topics, then the model would simply identify topics by clustering flow features and would not learn representations of topics that are shared across images. If the model were changed so that there was just a single flow component per topic, then the model would not be able to account for multiple flows per object. The full FLDA model accounts for these aspects.

2.2. FLDA algorithm

We use a fully collapsed Gibbs sampling scheme where the $\boldsymbol{\theta}$, $\boldsymbol{\pi}$, $\boldsymbol{\phi}$, $\boldsymbol{\Sigma}$, and $\boldsymbol{\mu}$ are analytically integrated out of the model, so that we need only sample values for z and c . The standard LDA updates from [19] are modified to account for the flow component hidden variable c , and for the likelihood of the flow observations v_f^i under the Gaussian model assumption. z is sampled from a multinomial distribution

$$P(z_f^i = t | \mathcal{Z}^{-if}, \mathcal{C}, \mathcal{X}, \Theta) \propto \frac{(N_{wt}^- + \beta)}{(\sum_w N_{wt}^- + W\beta)} \frac{(N_{lit} + \gamma)}{(N_{it} + L\gamma)} (N_{it}^- + \alpha) \quad (10)$$

where W is the codebook size, $\mathcal{Z}^{-if} = \mathcal{Z} \setminus \{z_f^i\}$, N^- indicates a count not using the topic assigned to the variable for which a new value is sampled, $N_{wt} = \sum_{i,f} [x_f^i = w] [z_f^i = t]$, $N_{lit} = \sum_f [z_f^i = t] [c_f^i = l]$, and $N_{it} = \sum_f [z_f^i = t] = \sum_l N_{lit}$.

c is similarly sampled from a multinomial distribution, where the flow likelihood is evaluated for the component. Due to the decoupling of the flow from the rest of the model variables given the component assignments and the integration over means and covariances, this term is the Gaussian mixture model predictive distribution for v when the mean and covariances have been integrated out, which is the multivariate Student- t distribution, $t_{\text{dof}}(m, S)$ with dof degrees of freedom, mean m and scale S :

$$P(c_i^d = l | \mathcal{Z}, \mathcal{C}^{-di}, \mathcal{V}, \Theta) \propto \frac{(N_{lit}^- + \gamma)}{(N_{it}^- + L\gamma)} t_{\nu_n - 1} \left(\mu_n, \frac{(\kappa_n + 1)\Lambda_n}{\kappa_n(\nu_n - 1)} \right) \quad (11)$$

where $N_{li} = \sum_i [c_i^d = l] = \sum_t N_{lit}$, $n = N_{li}^-$, and

$$\mu_n = \frac{\kappa_0}{\kappa_0 + n} \mu_0 + \frac{n}{\kappa_0 + n} \hat{\mu}_l^i \quad (12)$$

$$\Lambda_n = \Lambda_0 + \hat{S}_l^i + \frac{\kappa_0 n}{\kappa_0 + n} (\hat{\mu}_l^i - \mu_0)(\hat{\mu}_l^i - \mu_0)^T \quad (13)$$

$$\kappa_n = \kappa_0 + n \quad \nu_n = \nu_0 + n \quad (14)$$

$$\hat{S}_l^i = \sum_{g \neq f} [c_g^i = l] (v_g^i - \hat{\mu}_l^i)(v_g^i - \hat{\mu}_l^i)^T \quad (15)$$

$$\hat{\mu}_l^i = \frac{1}{n} \sum_{g \neq i} [c_g^i = l] v_g^i. \quad (16)$$

3. The CityCars dataset

Since fobject analysis requires a dataset containing consecutive pairs of video frames, we created a new ‘CityCars’ dataset¹ that includes 315 image pairs shot in an urban scene containing moving cars (positive examples). The dataset also includes 338 images shot in the same environment but without cars, which can be used as negative examples for supervised classification. To avoid allowing classifiers to cheat by primarily using background features to classify foreground objects, we ensured that many of the negative examples were recorded at the same locations as the positive examples. See Fig. 3.

We intentionally constructed the CityCars dataset so that it would pose a more realistic and challenging static image classification task. To demonstrate this, we used state-of-the-art methods to compare classification performance on the CityCars dataset with that obtained on a dataset containing 123 Caltech images [20] containing side views of cars and 123 randomly selected images from other categories (we refer to this as the ‘CaltechCars’ dataset). For both datasets, we randomly partitioned the data into one half for training and one half for testing and repeated each experiment 20 times to estimate confidence intervals.

The results for several classifiers that use the spatial pyramid HOG descriptor (see below for details) are summarized in Table 1 and clearly support two observations. First, the CityCars dataset poses a much more difficult classification challenge than the CaltechCars dataset. This may be because the backgrounds, which take up most pixels in each image, are similar in the positive and negative examples. Second, whereas the intersection kernel (IK) SVM clearly outperforms simpler methods on the CaltechCars dataset, on the CityCars dataset the advantages of both the intersection kernel and the SVM disappear: the simple L2 nearest neighbour method outperforms SVM and IK-based methods. This result is explained by the fact that while the feature-AND operation of the intersection kernel enables high SVM accuracy when positive and negative examples do not share many features, it fails when they have many

¹Available for download at <http://www.psi.toronto.edu/fobjectanalysis>



Figure 3: Positive (top) and negative (bottom) training (left) and test (right) images from the CityCars dataset.

	L2 NN	IK NN	IK SVM
CityCars	65% (3%)	55% (1%)	58% (2%)
CaltechCars	93% (3%)	98% (1%)	99% (1%)

Table 1: Comparisons showing that classifying cars in our CityCars dataset is much more difficult than in the Caltech101 dataset (CaltechCars). Classification was performed using spatial pyramid HOG descriptors with nearest neighbour (NN) and SVM classifiers, using the intersection (IK) and L2 kernels. One std. dev. is shown in brackets.

features in common, such as features derived from similar backgrounds. These results further support the conclusions presented in [23] regarding the inherent problems in benchmark datasets such as Caltech.

4. Feature extraction and descriptors

Below, we summarize how we extracted features, created training and testing partitions, and extracted descriptors used for classification (See also Supp. Material).

Training and test sets. Optical flow was extracted from image pairs using the “Lucas/Kanade meets Horn/Schunck” method [22]. HOG appearance features were extracted from the first image in an image pair, and also from a collection of static training images not containing cars. The data was randomly divided into 200 training cases and 200 test cases in a way that avoided having the same car in both the training and testing partitions, and k -means clustering was applied to the training data to obtain a codebook of size 1000 as is standard in the literature (eg [10]). Results reported below were obtained by repeating the above procedure at least 20 times to obtain confidence intervals.

LDA and FLDA descriptors. After using unsupervised learning to obtain the FLDA topics, we used those topics to analyze static images, which resulted in a topic assignment for every appearance feature in the static image. A histogram of appearance features was constructed separately for each topic and they were appended to the global histogram to make an FLDA descriptor of length $W(T+1)$ for T topics and a codebook of size W . In contrast to

the spatial pyramid approach, which factorizes the global histogram according to spatial regions [21], our approach factorizes the global histogram according to learnt topics. To account for the fact that topics capturing small objects might have a small number of counts, the topic-specific histograms were normalized to sum to 1 before concatenation. For comparison, we also used LDA without flow to learn topics and the same method as described above was used to obtain LDA descriptors.

SPHOG descriptors. The spatial pyramid match kernel based on HOG/SIFT features has been shown to give state-of-the-art classification performance on standard datasets [21]. For our comparison experiments, we used two levels of the pyramid, resulting in a descriptor of length $5W$ (global histogram plus four quadrants).

5. Experiments

The experiments we report investigate the usefulness of fobject analysis in producing an image analyzer that can be used to extract descriptors for object recognition. We report classification results for the CityCars dataset, compare inter-dataset generalization capability, and explore properties of our method. We also examine performance on articulated objects in the context of a spatial hierarchical FLDA based descriptor.

5.1. Comparisons using CityCars data

We compared the FLDA descriptor (3000 dimensional, based on two topics) to four alternatives: a 1000-dimensional HOG descriptor; a 5000-dimensional spatial pyramid HOG (SPHOG) descriptor; a 960-dimensional gist descriptor [4]; and a 3000 dimensional descriptor obtained from standard LDA (see Sec. 4). Using the nearest neighbour classifier (NN), we experimented with both the L2 (Euclidean) and the kernel intersection distance. Furthermore, we explored various normalization schemes for the descriptors. We show results for no normalization, L1 and L2 normalization. Results for Euclidean distance are shown in Table 2 and those for the intersection kernel in Table 3.

Algorithms trained using the FLDA descriptor achieve the best overall classification accuracy, outperforming other

L2 NN	None	L1	L2
HOG	65% (5%)	60% (6%)	54% (2%)
SPHOG	65% (7%)	64% (6%)	54% (4%)
Gist	69% (5%)	69% (4%)	70% (5%)
LDA	62% (5%)	64% (5%)	59% (4%)
FLDA	61% (7%)	82% (4%)	73% (5%)

Table 2: L2 nearest neighbour classification accuracy on the CityCars dataset for various descriptors and normalization schemes.

IK NN	None	L1	L2
HOG	57% (4%)	56% (4%)	67% (5%)
SPM	56% (4%)	57% (4%)	63% (6%)
Gist	61% (9%)	63% (9%)	60% (7%)
LDA	56% (3%)	57% (4%)	70% (6%)
FLDA	56% (3%)	66% (7%)	79% (4%)

Table 3: Intersection kernel nearest neighbour classification accuracy on the CityCars dataset for various descriptors and normalization schemes.



Figure 4: Test images (left column) along with the nearest training image using FLDA descriptors (middle column) and SPHOG descriptors (right column).

descriptors for both the Euclidean and the intersection kernel distances. This indicates that FLDA topic-based factorization of the histograms is beneficial to classification. The LDA descriptor performs similarly to SPHOG, showing that topics inferred without motion coherence do not help with classification. Finally, we note that normalization affects performance, and that the effect is not consistent across different distance metrics used for the NN classifier, indicating that normalization plays an important role. Based on these results, later, we use the Euclidean distance (L2) NN classifier with L1 normalization for both the topic specific and

SPHOG		Testing	
		CityCars	CaltechCars
Training	CityCars	65% (3%)	63% (6%)
	CaltechCars	62% (3%)	93% (3%)

Table 4: Inter-dataset generalization (classification accuracy) using spatial pyramid HOG (SPHOG) descriptors.

FLDA		Testing	
		CityCars	CaltechCars
Training	CityCars	82% (4%)	73% (3%)
	CaltechCars	63% (2%)	93% (2%)

Table 5: Inter-dataset generalization (classification accuracy) using FLDA descriptors.

global histograms for the FLDA descriptor.

We provide a visual representation of the nearest training neighbours for some test cases in Fig. 4, when using FLDA and SPHOG descriptors. When using SPHOG descriptors, the classifier is more likely to confuse background features with object features.

5.2. Inter-dataset generalization

We next investigated to what extent the FLDA model generalizes to other datasets compared to using SPHOG. Using the FLDA based image analyzer obtained from the CityCars data, we generated FLDA descriptors for both CaltechCars and CityCars. We also produced SPHOG descriptors for both datasets. Splitting each set into training and test, we made the full cross comparisons reported in Table 4 and Table 5 for SPHOG and FLDA descriptors, respectively. FLDA descriptors considerably outperform SPHOG on two of the four comparisons, and compare very similarly for the other two, indicating that FLDA descriptors can successfully generalize across datasets. Note that using FLDA descriptors trained on CityCars to classify CaltechCars, compared to that of SPHOG descriptors trained on CaltechCars to classify CityCars yield significantly better results for the FLDA descriptors (73% vs 62%).

5.3. Exploration of training conditions

We investigated the sensitivity of FLDA to various training parameters. Fig. 5 shows how classification accuracy depends on the supervised training set size and includes results for SPHOG and HOG. To investigate the sensitivity of FLDA to noise in the input optical flow (eg, due to changing appearance, aperture problems or poor optical flow computation), we artificially added increasing amounts of Gaussian noise to the flow before applying FLDA. Fig. 5 shows that as the noise increases, the performance degrades smoothly towards the level of SPHOG performance.

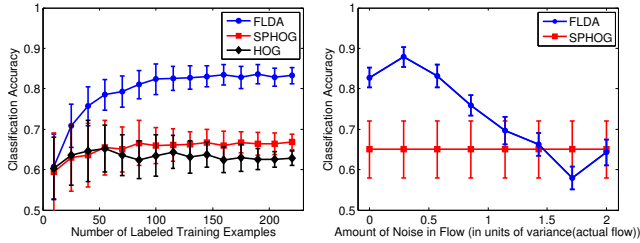


Figure 5: Exploration of training conditions (see Sec. 5.3).

L2 NN	None	L1	L2
HOG	55% (1%)	55% (1%)	51% (1%)
SPHOG	53% (1%)	53% (1%)	53% (1%)
LDA	55% (1%)	53% (1%)	53% (1%)
FLDA	55% (1%)	52% (1%)	52% (1%)

Hierarchical Descriptors	
H-LDA	57% (1%)
H-FLDA	68% (1%)

Table 6: L2 nearest neighbour classification accuracy on the CityPedestrians dataset.

5.4. Hierarchical FLDA descriptors

Many objects are best characterized by combinations of parts, where each part is spatially localized but parts may move, or articulate relative to one another. The FLDA-derived image descriptors described above do not take into account the locations of visual words in the image and consequently do not model spatially localized parts. Here, we show how the FLDA-derived topics can be used to construct hierarchical descriptors that can account for localized parts.

To investigate the use of FLDA for modelling articulated objects containing spatially localized parts, we constructed the ‘CityPedestrians’ dataset² (Fig. 6). It includes 938 image pairs of side views of pedestrians walking in an urban environment, as well as 456 static images without pedestrians for use as negative examples. Similarly to the CityCars dataset, we ensured that many of the negative examples were recorded at the same locations as the positive examples. Since pedestrians contain multiple articulated parts and vary much more in shape and appearance than cars, we expect this dataset to be more difficult than the CityCars dataset. Indeed, the classification accuracies for several previously described descriptors are shown at the top of Table 6 and are comparable to random guessing (50%).

One advantage of fobject analysis is that once the flow-based topics have been learnt, they can be used in many

²Available for download at <http://www.psi.toronto.edu/fobjectanalysis>

different ways to develop descriptors for static images. To account for spatially localized parts when forming the descriptor, we constructed a descriptor that combines visual words in a hierarchical fashion. For a given image, the H-FLDA descriptor is a T dimensional vector that sums to 1, where T is the number of topics obtained during FLDA. This descriptor is created by scanning a 10×10 window over the image, computing the histogram over visual words for each window, and assigning each window to the best topic (using L2). For each topic, the total number of windows assigned to it is computed and the histogram is normalized to obtain the H-FLDA descriptor. For comparison, we also used H-LDA descriptors using the same method, but with LDA-derived topics.

The bottom part of Table 6 shows that the hierarchical H-FLDA descriptor improves performance over the FLDA descriptor, H-LDA descriptor and other descriptors on the CityPedestrians dataset. The vocabulary size for the CityPedestrians dataset was chosen using cross-validation, as well as the number of topics for the hierarchical descriptors (see Supp. Material).

6. Related Work

Methods for motion or activity modelling and video summarization are relevant to our work. However, while the literature in this area is extensive, to our knowledge none of that work is directed toward training methods that can extract good representations for static images, which is the problem that we study here. An interesting avenue for further research is to examine previously described methods for jointly modelling appearance and motion and consider integrating out the motion part of the model after training. This approach could lead to different methods for fobject analysis.

Regarding our extension of LDA to model word-specific real-valued optical flow vectors, while there is no previous work in this area, our extended model is most similar in spirit to the work of Sudderth *et al* [8]. They extend LDA hierarchically to allow for variable spatial layouts of visual words. If spatial coordinates in their model were replaced with flow vectors, their model could be used for fobject analysis. However, it is not clear how well this approach would work, since their model was not applied using optical flow and their learnt models were not tested in the absence of spatial information. Others have pre-clustered visual words according to spatial layout and then applied LDA using either subregion-defined words [9] or ‘doublet’ words that encode spatially proximal visual words [10]. A similar approach could be used to pre-cluster visual words according to similar optical flow. However, it is not clear how optical flow should then be integrated out for static image analysis.



Figure 6: Positive (top) and negative (bottom) training (left) and test (right) images from the CityPedestrians dataset.

7. Conclusions

Flobject analysis produces a model that can be used to infer representations of static images that are consistent with optical-flow derived objects, or flobjects. Pairs of consecutive video frames or stereo image pairs are used as input to flobject analysis, but unlike previously described motion analysis techniques, the model obtained by flobject analysis can be applied to static images to extract useful image representations. We examined the CityCars dataset, which includes video frames and is more challenging than an appropriate subset of the Caltech101 dataset (CaltechCars) because both positive and negative examples have urban street backgrounds. We found that the static image representations found by FLDA-based flobject analysis produce significantly higher classification rates than those obtained using other standard descriptors.

The framework we described can be improved in several ways, and altogether different kinds of models can be used for flobject analysis, such as those that decompose the image into a hierarchy of parts or use layers of variables to account for high-order statistics. Importantly, while the framework requires moving objects, this can be achieved for static objects by panning a camera or using stereo data. Currently, we are working on developing a ‘flobject database’ containing hundreds of thousands of image pairs and high-quality optical flow fields extracted from dozens of hours of movies. One important open question is how the framework we described here can be scaled up to handle such a large scale dataset. Also, while here we tested the framework using object classification tasks, future work could include applications to image segmentation and object localization.

Acknowledgments

We thank Andrew Blake and Zoubin Ghahramani for helpful discussions. Funding for IEG was provided by an NSERC Graduate Scholarship and funding for BJF was provided by an NSERC Discovery Grant.

References

- [1] DM Blei, AY Ng, M. Jordan. Latent Dirichlet allocation. *JMLR*, 2003.
- [2] M Ranzato, CS Poultney, S Chopra, Y LeCun. Efficient learning of sparse representations with an energy-based model. *NIPS*, 2006.
- [3] N Jojic, BJ Frey. Learning flexible sprites in video layers. *CVPR*, 2001.
- [4] A Oliva, A Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001.
- [5] T Serre, L Wolf, T Poggio. Object recognition with features inspired by visual cortex. *CVPR*, 2005.
- [6] L Bourdev, J Malik. Poselets: Body part detectors trained using 3D human pose annotations, *ICCV*, 2009.
- [7] N Jojic, A Perina, M Cristani, V Murino, BJ Frey. Stel component analysis, *CVPR*, 2009.
- [8] E Sudderth, A Torralba, WT Freeman, A Willsky. Describing visual scenes using transformed objects and parts. *IJCV*, 2008.
- [9] L Cao, L Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. *ICCV*, 2007.
- [10] J Sivic, BC Russell, AA Efros, A Zisserman, WT Freeman. Discovering objects and their locations in images. *ICCV*, 2005.
- [11] D Lowe. Object recognition from local scale-invariant features. *ICCV*, 1999.
- [12] R Fergus, P Perona, A Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *IJCV*, 2007.
- [13] BJ Frey, N Jojic. Estimating mixture models of images and inferring spatial transformations using the EM algorithm. *CVPR*, 1999.
- [14] J Winn, A Blake. Generative affine localisation and tracking. *NIPS*, 2004.
- [15] GE Hinton, RR Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006.
- [16] A Torralba, R Fergus, Y Weiss. Small codes and large image databases for recognition. *CVPR*, 2008.
- [17] L Fei-Fei, P Perona. A Bayesian hierarchical model for learning natural scene categories. *CVPR*, 2005.
- [18] E Bart, I Porteous, P Perona, M Welling. Unsupervised learning of visual taxonomies. *CVPR*, 2008.
- [19] TL Griffiths, M Steyvers. Finding Scientific Topics. *PNAS*, 2004.
- [20] L Fei-Fei, R Fergus, P Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. *CVPR*, 2004.
- [21] S Lazebnik, C Schmid, J Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2006.
- [22] A Bruhn, J Weickert, C Schnörr. Lucas/kanade meets horn/schunck: combining local and global optic flow methods. *IJCV*, 2005.
- [23] N Pinto, DD Cox, JJ Dicarlo. Why is Real-World Visual Object Recognition Hard? *PLOS Comp. Bio.*, 2008.
- [24] Y Ostrovsky, E Meyers, S Ganesh, U Mathur, P Sinha. Visual Parsing After Recovery From Blindness. *Psychological Science*, 2010.