Joint Feature Selection for Object Detection and Recognition

Jerod J. Weinman, Allen Hanson, and Erik Learned-Miller
Department of Computer Science
University of Massachusetts, Amherst, MA 01003
{weinman,hanson,elm}@cs.umass.edu

Technical Report UM-CS-2006-54
October 19, 2006
Updated: April 6, 2007

Abstract

Classifiers for object categorization and identification, such as face detectors and face recognizers, are often trained separately and operated in a feed-forward fashion. Selecting a small number of features for these tasks is important to prevent over-fitting and reduce computation. However, when a system has such related or sequential tasks, training and selecting features for these tasks independently may not be optimal. We propose a framework for choosing features to be shared between categorization and identification tasks. The result is a system that achieves better performance with a given number of features. We demonstrate with experiments using text and car detection as categorization tasks, and character and vehicle type recognition as identification tasks.

1. Introduction

Many real-world problems must solve multiple classification tasks simultaneously or sequentially. For example, a vision system may need to discriminate between cars, people, text, and background as high-level categories, while also recognizing particular cars, people, and letters. The categorization/detection task is to determine whether an image region corresponds to an object from a class of interest (e.g., text) or not. The identification/recognition task discriminates among members of that category (e.g., if this is text, is the character a `p` or a `q`?). Often the categorization and identification tasks are treated in a hierarchical or sequential manner by first running a category-level detector and then feeding detections into a category-specific recognizer. Moreover, although the classifiers for the two tasks are related, they are usually trained independently. This work seeks to knit these processes more tightly by considering them jointly during model training.

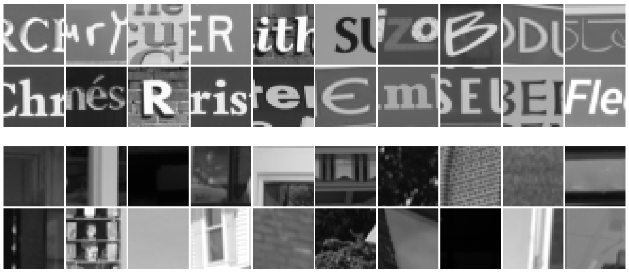


Figure 1. The categorization/detection task must only discriminate characters (top) from background patches (bottom), while the identification/recognition task must identify the centered character.

Constructing a model for a classification task involves many issues, including deciding which features or observations are relevant to the decision. Two reasons for limiting the number of features involved in classification include preventing over-fitting and reducing the amount of computation needed to reach a decision. Models with too many irrelevant features are prone to poor generalization since they are fit to unnecessary constraints. Even when there is no over-fitting, if certain features are redundant or unnecessary, the classification process can be expedited by eliminating the need to compute them.

Feature selection may be important for both categorization and identification, the primary difference being the generality of the classification tasks. However, if these problems are treated in isolation, we may not achieve a feature selection that is optimal—in computational or accuracy terms—for the *joint* categorization-identification problem.

While some features will undoubtedly be useful primarily for detecting object categories and others will have the greatest utility for recognizing objects from a particular category, there may be some features with utility for both tasks.

When this is the case, a method accounting for overlap in utility may have two advantages. First, a feature useful for object identification may boost category detection rates for the class by incorporating more object-specific information in the search. Second, if such dual-use features have already been computed for the purposes of category detection, they may subsequently be utilized for identification, effectively reducing the amount of computation necessary to make a classification.

We propose a framework for jointly considering the general categorization and specific identification tasks when selecting features and compare it to two other approaches. The first attempts only to predict identities and has no explicit representation of categories. The second approach has a category model and several category-specific recognizers, all of which are trained independently. Our approach is to jointly learn and select features for the category and identity models. The three methods are compared in experiments on text and vehicle detection and recognition, examining both the overall and category-level classification accuracy as the number of features increases. We find that the joint approach reduces error by 50% over the independent approach when the number of features is small and 20% as the number increases.

2. Related Work

2.1. Feature Selection

Several general frameworks exist for selecting features. The two most basic are greedy *forward* and *backward* schemes. Forward schemes incrementally add features to a model based on some criterion of feature utility. Examples of this include work by Viola and Jones [16], who use single-feature linear classifiers as weak learners in a boosting framework, adding features with the lowest weighted error to the ensemble. Backward schemes, by contrast, selectively prune features from a model. Many other variants for selecting a subset of features are possible; see Blum and Langley [3] for a more thorough review.

Feature types and selection strategies for visual tasks have varied widely. The Viola-Jones object detector [16] employs outputs of simple image difference features, which are similar to wavelets. There are many possible filters and only a few are discriminative, so a selection process is required primarily for computational efficiency. Other methods use image fragments or patches as feature descriptors [15, 1, 12]. These high-dimensional features can be densely sampled and vector quantized to create a discrete codebook representation [17, 6]. Alternatively, LeCun et al. [7] learn (rather than select for) a discriminative intermediate feature representation. These models are related to the Fukushima Neocognitron [4], a model with hierarchical processing for invariant recognition based on successive stages of local template matching and spatial pooling. However, all of these methods are focused on one level of recognition, either categorization or identification.

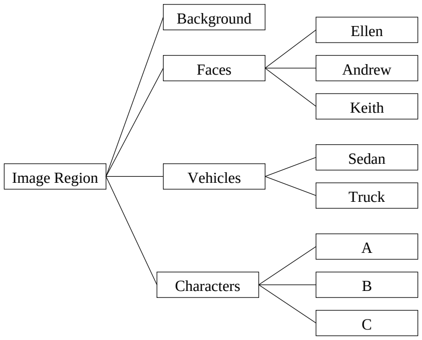


Figure 2. An example object class hierarchy for images. Categorization is finding instances in the center column, while identification is recognizing instances in the right column.

2.2. Feature Sharing

Torralba et al. [14] have shown that jointly selecting features for detecting several object categories generalizes better and reduces the requisite the number of features. Our work synthesizes many of these ideas, adding the object identification task to the competition for feature resources. Performance improves by selecting features to be shared among sibling tasks [14], i.e., the category detectors within the middle column of Figure 2. We propose to select features to be shared by hierarchical or sequential tasks: *between* the middle (category detection) and right (recognition/identification) columns.

Recently, Bar Hillel and Weinshall [5] have shown that learning category level models first and using that representation for identification is better than learning identification models directly. While their framework learns a binary detector for each category, we learn one model that discriminates among categories. Although slightly different, we can compare our approach to a method that selects features for categorization and uses only these same features in all identification models.

3. Categorization and Recognition Model

For every query image or sub-image, our goal is to determine whether the query belongs to some general category of interest, and if so, to recognize it as a particular instance of that category. For instance, given an image region, we may first want to determine whether it is text, and if so, to identify the character. Thus, every query is assigned a category and category-specific identity; if it does not belong to any particular category of interest, we may call it "background."

In this section, we describe three alternatives to modeling and training (Figure 3). First, we describe the com-

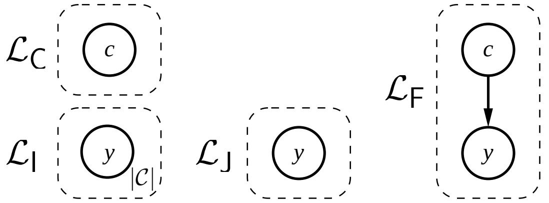

$\mathcal{L}_{\mathsf{C}}$
c
$\mathcal{L}_{\mathsf{I}}$
y
$|\mathcal{C}|$
$\mathcal{L}_{\mathsf{J}}$
y
$\mathcal{L}_{\mathsf{F}}$
c
y

Figure 3. Graphical representation of the models and training strategies. Dashed lines indicate the unknowns considered by a training objective function. LEFT: Independent models (there are $|\mathcal{C}|$ recognition objectives, one for each category). CENTER: Joint model. RIGHT: Factored model.

mon method of treating categorization and identification independently, followed by a simple flat model that forgoes category modeling and only aims to identify each query. Finally, we propose a factored model for jointly learning categorization and identification.

Let $c \in \mathcal{C}$ represent the category (e.g., text, vehicle, etc.) of an image query $\mathbf{x}$. For each category c, there is a set $\mathcal{Y}_c$ of particular objects in that category; e.g., $\mathcal{Y}_{\mathsf{c}}$ might be characters for the text category. If we include a background category $\mathsf{b} \in \mathcal{C}$ for other regions, then every image region takes a category label from $\mathcal{C}$ and an identity from $\mathcal{Y} = \{\mathsf{b}\} \cup \bigcup_{c \in \mathcal{C}} \mathcal{Y}_c$. We assume that objects belong to only one category.

To categorize and identify objects, we will use discriminative maximum entropy probability models [2]. For each model, a set of real-valued features are calculated from the image and multiplied by class-specific weights to yield a probability for that class. We describe in greater detail how these features are chosen in the following section.

Models for the categorization and identification problems are typically learned independently. Formally, a categorization model is a probability

$$p\left(c \mid \mathbf{x}, \boldsymbol{\lambda}, F\right) \equiv \frac{1}{Z} \exp\left(\boldsymbol{\lambda}\left(c\right) \cdot F\left(\mathbf{x}\right)\right), \qquad (1)$$

for $c \in \mathcal{C}$, where $\boldsymbol{\lambda}$ are parameters of the model, F represents the features of $\mathbf{x}$ that are calculated, and Z is a normalizing constant ensuring the expression is a proper probability. Similarly, an identification model is a probability conditioned on the category,

$$p\left(y \mid c, \mathbf{x}, \boldsymbol{\theta}_c, G_c\right) \equiv \frac{1}{Z} \exp\left(\boldsymbol{\theta}_c\left(y\right) \cdot G_c\left(\mathbf{x}\right)\right), \qquad (2)$$

for $c \in \mathcal{C},\ y \in \mathcal{Y}_c$, where G_c are the features for identifying objects in a category c, and $\boldsymbol{\theta}_c$ are the corresponding parameters. Identification models for different categories need not use the same features. We define $p\left(y \mid c, \mathbf{x}, \boldsymbol{\theta}_c, G_c\right) = 0$ for $y \notin \mathcal{Y}_c$. Given a set of examples having category and identity labels $\mathcal{D} = \left\{\left(c^{(i)}, y^{(i)}, \mathbf{x}^{(i)}\right)\right\}_i$, the typical method of training is to independently optimize likelihoods for the categorization and category-specific identification models:

$$\mathcal{L}_{\mathsf{C}}\left(\boldsymbol{\lambda}; F, \mathcal{D}\right) \equiv \log \prod_i p\left(c^{(i)} \mid \mathbf{x}^{(i)}, \boldsymbol{\lambda}, F\right) \qquad (3)$$

$$\begin{aligned} &\mathcal{L}_{\mathsf{I}}^{c}\left(\boldsymbol{\theta}; G, \mathcal{D}\right) \equiv \\ &\quad \log \textstyle\prod_{i: c^{(i)} = c} p\left(y^{(i)} \mid c^{(i)}, \mathbf{x}^{(i)}, \boldsymbol{\theta}_{c^{(i)}}, G_{c^{(i)}}\right). \end{aligned} \qquad (4)$$

Note that there are actually $|\mathcal{C}|$ identification models (2), one for each category, which we can optimize with one likelihood

$$\mathcal{L}_{\mathsf{I}}\left(\boldsymbol{\theta}; G, \mathcal{D}\right) \equiv \sum_{c \in \mathcal{C}} \mathcal{L}_{\mathsf{I}}^{c}\left(\boldsymbol{\theta}; G, \mathcal{D}\right). \qquad (5)$$

Since every object belongs to one category, an alternative is to forgo category modeling altogether, and simply have one model that aims to determine identity from among all possible labels:

$$p\left(y \mid \mathbf{x}, \boldsymbol{\beta}, F\right) \equiv \frac{1}{Z} \exp\left(\boldsymbol{\beta}\left(y\right) \cdot F\left(\mathbf{x}\right)\right), \qquad (6)$$

for $y \in \mathcal{Y}$, where $\boldsymbol{\beta}$ and F are the parameters and features of $\mathbf{x}$, respectively. We call this the joint model. The probability for a category label c can be calculated by summing the probabilities for all $y \in \mathcal{Y}_c$. Training then involves optimizing one likelihood,

$$\mathcal{L}_{\mathsf{J}}\left(\boldsymbol{\beta}; F, \mathcal{D}\right) \equiv \prod_i p\left(y^{(i)} \mid \mathbf{x}^{(i)}, \boldsymbol{\beta}, F\right), \qquad (7)$$

but the label space $\mathcal{Y}$ of the probability (6) is potentially very large.

Alternatively, the joint probability for categorization and identification can be written as the product of two probabilities:

$$p\left(c, y \mid \mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\theta}, F\right) = p\left(y \mid c, \mathbf{x}, \boldsymbol{\theta}_c, F\right) p\left(c \mid \mathbf{x}, \boldsymbol{\lambda}, F\right), \qquad (8)$$

where we assume the same features F are used by both the categorization and all identification models, but different parameters (discriminative feature weights $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}_c$) are used by each. Training the factored model $p\left(c, y \mid \mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\theta}, F\right)$ now involves accounting for *both* the categorization and identification likelihoods (3) and (5) *simultaneously* rather than treating them independently:

$$\mathcal{L}_{\mathsf{F}}\left(\boldsymbol{\lambda}, \boldsymbol{\theta}; F, \mathcal{D}\right) \equiv \mathcal{L}_{\mathsf{C}}\left(\boldsymbol{\lambda}; F, \mathcal{D}\right) + \mathcal{L}_{\mathsf{I}}\left(\boldsymbol{\theta}; F, \mathcal{D}\right). \qquad (9)$$

Whereas the previous joint model (6) is flat, the factored model (8) is hierarchical. The difference between factored training with (9) and the independent training with (3) and (5) is manifest during feature selection.

The log-likelihoods (3), (4), (7), and (9) are all convex, so globally optimal parameters can be found via convex optimization. We also follow the typical optimization regularization strategy by adding a parameter prior to the likelihood objective function (details are in §6.1).

If categorization and identification models are learned independently, the features used to make a category decision might not overlap with the features used for identification, possibly increasing the total amount of computation. Furthermore, objects in the same category are visually related, and learning category-level models may improve category detection and the ultimate object identification results by modeling them together. In the next section, we elaborate on feature selection for our chosen class of models.

4. Feature Selection

The algorithm we use for selecting features is a greedy forward method that incrementally adds the feature providing the greatest increase in the likelihood being optimized [2]. Each model has its own likelihood (e.g., $\mathcal{L}_{\mathsf{C}}$, $\mathcal{L}_{\mathsf{I}}^{c}$, $\mathcal{L}_{\mathsf{J}}$, and $\mathcal{L}_{\mathsf{F}}$), and the same algorithm is used on each. As an example, consider the categorization model (1) and corresponding likelihood (3). At some iteration, the model includes set of features F (initially empty) and parameters $\widehat{\boldsymbol{\lambda}}$ that optimize the category likelihood $\mathcal{L}_{\mathsf{C}}(\boldsymbol{\lambda}; F, \mathcal{D})$. Then, some new candidate feature f is added to the feature set F. Let the augmented features be F' and the corresponding augmented parameters be $\boldsymbol{\lambda}'$. After optimizing the likelihood of the augmented model to find the new optimal parameters $\widehat{\boldsymbol{\lambda}}'$, we may calculate the "gain" of the candidate feature by taking the difference of log-likelihoods

$$\mathcal{G}_{\mathsf{C}}(f; \mathcal{D}) = \mathcal{L}_{\mathsf{C}}\left(\widehat{\boldsymbol{\lambda}}'; F', \mathcal{D}\right) - \mathcal{L}_{\mathsf{C}}\left(\widehat{\boldsymbol{\lambda}}; F, \mathcal{D}\right). \quad (10)$$

This is equivalent to a likelihood ratio test of the two models. We calculate the gain of all features from a pool of candidates and add the feature with the highest gain to the model.

Thus, a model is built by iteratively adding the best (highest-gain) feature. The same process may be followed for the joint model likelihood $\mathcal{L}_{\mathsf{J}}$ (7) and the factored model likelihood $\mathcal{L}_{\mathsf{F}}$ (9). For the independent method, each category-specific likelihood $\mathcal{L}_{\mathsf{I}}^{c}$ (4) goes through its own feature selection process to determine the category-specific identification features G_c.

Since many candidate features may need to be examined at each feature selection iteration, approximations are helpful for speeding the process. First, only the augmented parameters for the candidate features are optimized [2], leaving the optimal parameters $\widehat{\boldsymbol{\lambda}}$ for the selected features F fixed in the gain calculation (10). Second, we calculate the gains on a representative subset of the training data $\mathcal{D}' \subset \mathcal{D}$, and then re-calculate the gains of only the top-ranked features using all the training data.

When two separate models are independently trained for a pipelined framework, the gain of a feature is only measured with respect to a particular task, categorization or identification. However, considering the *entire* end-to-end task of categorization and identification yields a different ranking of the features. We show in Section 6 that jointly considering the tasks during feature selection improves performance accuracy and speed.

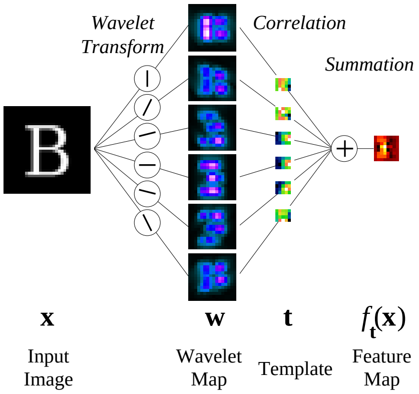


Figure 4. Computation of a single feature map from one template. Scale and orientation selective wavelet features are applied to input image $\mathbf{x}$, followed by a normalization and dilation/downsampling to yield $\mathbf{w}$. Correlations between corresponding channels in a template $\mathbf{t}$ and wavelet map $\mathbf{w}$ are computed and summed, followed by another dilation/downsampling to yield a feature map f. Several additional templates would be used to provide an array of feature maps, which becomes a feature vector for the classifier.

5. Detection and Recognition Features

In this section, we describe the two types of candidate features for our model: region-based texture features and local template features.

All features are derived from the steerable pyramid wavelet basis [13], a set of scale and orientation selective filters that loosely resembles the "simple cells" in an initial layer of processing in mammalian visual systems The wavelet coefficients are complex, representing outputs from paired even and odd filters for each scale and orientation (channel). Taking complex magnitudes yields phase invariant responses, similar to complex cells in biological systems.

The first pool of candidate features is a set of image and wavelet statistics originally crafted for texture synthesis [10]. These include image statistics (four central moments plus min and max) at several scales, means of wavelet channel magnitudes, and local auto- and cross-correlation of wavelet channels. Although originally intended to be computed globally over an image of ergodic texture, we compute them "locally" over small image regions, which can be efficiently achieved by convolution.

A character classifier using the wavelet channel magnitudes directly as features is not robust to image deformations. Research in cognitive psychology by Rehling [11] indicates that two mechanisms operate in human character recognition: an initial "flat" recognizer that is fast and a secondary hierarchical, parts-based model like LeCun's convolutional network [7] that is slower but more accurate. To construct a model of this hierarchical framework, template-based feature maps form our second pool of candidates.

Calculating a template-based feature map from the wavelet channel magnitudes involves five steps: normalization, downsampling, correlations, a summation, and another downsampling. First, the wavelet magnitudes are locally normalized by a process similar to that of SIFT [8]: at each location, all the wavelet magnitudes in a local window are normalized to a unit ℓ_2 norm, clipped at a threshold (0.2 in our experiments), and re-normalized to yield the value for that (center) location. Next, to decrease spatial and phase sensitivity, the image's normalized wavelet magnitudes are downsampled after taking the maximum over a small window within each channel (a simple morphological dilation). Let $\mathbf{w}$ be the result of the normalization and downsampling steps, a more compact stack of scale- and orientation-specific images, as shown in Figure 4.

A *template* $\mathbf{t}$ is a small patch extracted from the values of a training example's processed wavelet magnitudes $\mathbf{w}$; each $\mathbf{t}$ is subsequently normalized to have zero mean and unit ℓ_1 norm. The *feature map* $f_{\mathbf{t}}$ for such a template is calculated by first computing the cross-correlation between an input image's wavelet features $\mathbf{w}$ and the corresponding channels from the template $\mathbf{t}$, and then summing the output over all channels (i.e., scales and orientations). Let $\mathbf{t}^c$ represent the normalized wavelet coefficient magnitudes of some channel c for a template, then the corresponding feature map calculation for an image $\mathbf{x}$ having (normalized and downsampled) wavelet coefficient magnitudes $\mathbf{w}$ is

$$f_{\mathbf{t}}\left(\mathbf{x}\right) = \sum_c \mathbf{t}^c \otimes \mathbf{w}^c \qquad (11)$$

where $\otimes$ is the cross-correlation operator. The feature map $f_{\mathbf{t}}$ is then subject to another downsampling operation for even further spatial pooling and dimensionality reduction.

An illustration of the image-to-feature map calculation is given in Figure 4. The resulting template feature map outputs may be transformed into a vector and added to the classification model as entries in $F\left(\mathbf{x}\right)$. The ultimate goal will then be to select the texture statistics or templates most useful for a particular task, be it categorization, identification, or both.

6. Experiments

In this section, we compare four training and feature selection strategies for categorization/detection and identification/recognition: (i) the flat joint classifier, (ii) the factored but jointly trained classifier, (iii) the independently trained classifiers operated sequentially, and (iv) the independent classifiers operated sequentially, but trained using the only the features selected for categorization.

To test our hypothesis that joint feature selection can improve speed and accuracy, we need data with labels for background, as well as both category and identity. We perform experiments on a three category problem involving background, text, and vehicles, with synthetic but difficult data for the latter two categories.

6.1. Data and Procedure

In this section we describe the data from our three categories.

Background A set of 300 images taken from scenes around a downtown area have had text regions manually masked out, and square patches of various scales from the non-text regions were extracted and labeled as background. Examples are in Figure 1 (bottom).

Characters Rather than manually crop and label individual characters from image regions, we synthesize similar character images. There are 62 characters in our alphabet to be recognized (uppercase, lowercase, and digits), rendered in 954 different fonts at a 12.5 pixel x-height and centered in a 32×32 window. Neighboring characters were sampled from bigrams learned on a corpus of English textand placed with uniform random kerning. The resulting trigram image was then subject to a random distortion modelled after the text from our scene images. Adding these factors to the data set allows the classifier to learn them and provides a reasonable test bed without having to manually ground truth individual characters in many images. The label of these character windows is the center character. Examples are shown in Figure 5 (top), and may be compared to actual images of scene text in Figure 1. Note that the identification task involves no character segmentation—the character in the center of the window must be recognized in the presence of neighboring character "clutter."

Vehicles Images of 21 vehicles rendered from three viewing directions and nine lighting conditions are from work on vehicle type identification by Ozcanli et al. [9]. The vehicle identification task then consists of labeling the image as one of SUV, passenger car, pickup truck, or van. Examples of the 32×32 images used are in Figure 5. At this resolution identification is difficult; while published experiments with this data provide the viewing angle and lighting conditions [9], our recognition model achieves similar results (57% accuracy) when the view and lighting are unknown.

Training Data Summary For the background category, our training set has roughly 65,000 windows at multiple

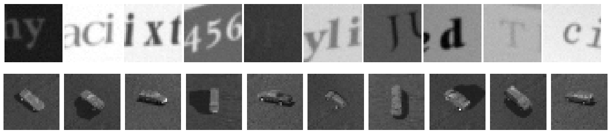


Figure 5. Sample images used in experiments: (top) characters and (bottom) vehicles.

scales from images of outdoor scenes. The character class has nearly 30,000 character windows (each of 62 characters in 467 fonts). The vehicle class has nearly 300 examples. Since text and vehicles are relatively rare in natural scenes, we weight all the data instances in training and test evaluation such that characters and vehicles both have a class prior of 1×10^{-4}; in other words, the ratio of text to background is almost one to ten-thousand. The test set is roughly the same size but comes from a different set of scene images, fonts, and vehicles. Indeed, if we use the same fonts for testing even with different distortions applied, the recognition results are much higher.

Features As shown in Figure 4, the wavelet transform of a given 32×32 patch is downsampled to 16×16 and the resulting feature map is downsampled to 4×4 for a very compact representation of responses for each template. Candidate template patches of various sizes were randomly extracted from the training character images. There were 2,000 template patches and 418 local statistics in the candidate feature pool.

Regularization Procedure In all cases, a Laplacian (ℓ_1) prior was used for regularization, and the relative weight of the prior α was chosen by cross-validation. The training set was split in two, half was used for training, and the value of α that yielded the highest likelihood on the other half was then used on the entire training set. All of the candidate features were included for cross-validation, since we do not a priori know which might be useful. However, a slightly smaller portion of the training data was used since all features for all instances exceeded memory limits.

6.2. Results

In this section, we describe and present the results of our experiments. Section 6.3 contains an analysis and discussion of these results.

Figure 6 shows the results of the selection strategies for the three category problem involving text, vehicles, and background. In the top two plots, the y-axis is the average identification accuracy (mean of a normalized confusion matrix's diagonal for $\mathcal{Y}$). The performance is shown at each round of feature selection for the joint and factored models, as well as the sequential models using only the top categorization features. However, we must decide how to allocate a given number of features between those chosen by the independently trained models. In a two category problem, this involves choosing how many are selected by the categorization model (i.e., text detector), leaving the remaining number for the identification model (i.e., character recognizer). This is a relatively straightforward one-dimensional optimization, but becomes more complex for three or more categories. The problem is simplified by allocating features between categorization and identification, where each category-specific identifier is allotted the same number of features. This one-dimensional strategy may be sub-optimal when some categories are harder to distinguish or are more important than others. This one-dimensional approximate optimization strategy is performed on the *test* data, so the results of the independent method are optimistic.

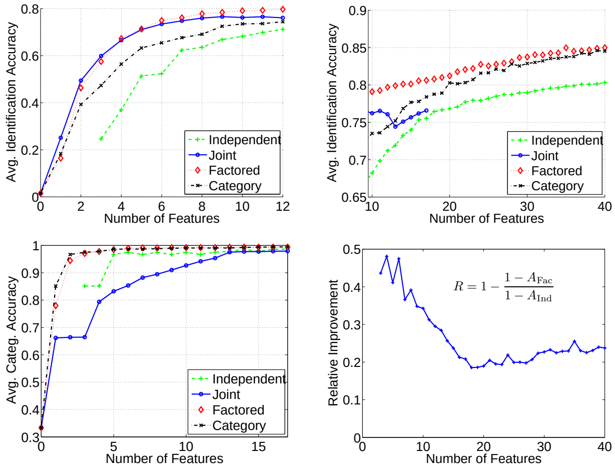


Figure 6. Comparison of the feature selection strategies for the three category (background, text, and vehicle) detection and recognition problem. TOP: Average class identification accuracy (LEFT and RIGHT are different views of the same curve). BOTTOM-LEFT: Average categorization accuracy. BOTTOM-RIGHT: Relative improvement of the factored over independent method on the average identification accuracy (A_{Fac} and A_{Ind}).

The bottom-right of Figure 6 shows the relative improvement (reduction in error on average identification accuracy) of the factored joint model over the optimized independently trained models. We also report the average categorization accuracy from the same classifications. Note that feature selection and optimization in the flat joint model takes much longer than the other training methods, therefore fewer rounds of feature selection were completed.

Figures 7 and 8 demonstrate the models operating on real images of scenes. In a two category framework (text and background) we train a detector (1) and recognizer (2) on all of the synthetic data. Examples of text detection are in Figure 7. Figure 8 gives an example of detection and recognition. Character recognition rate is 0.80, and the average identification accuracy (which includes the background cat-

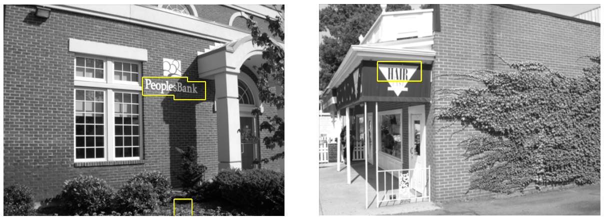
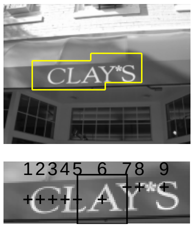


Figure 7. Examples of text detection with the two category model.



Det	Ch	Pr	Ch	Pr	Result
1	K	1.00			FP
2	**C**	0.96	G	0.04	C
3	Q	0.80	2	0.11	FP
4	L	0.85	I	0.15	FP
5	**L**	1.00			C
6	**A**	1.00			C
7	V	0.56	**Y**	0.18	E
8	i	0.78	F	0.21	FP
9	**S**	0.91	6	0.09	C

Figure 8. Recognition example with the two category (text and background) model. Boxes and crosses represent the center region of each detection. For each, the top two predicted characters and their probabilities are given. The evaluation result of each is listed (FP=False Positive, C=Correct, E=Error).

egory) for this image is 0.90.

6.3. Discussion

Our experimental results demonstrate the superiority of a factored joint feature selection over the traditional independent method in several ways. The first and most obvious way is that the average identification accuracy of the independent method is worse than the alternatives for any number of features. Bar Hillel and Weinshall [5] show that with $|\mathcal{C}|$ binary category detectors, using the features selected by the category detector for the category-specific identifiers improves results. However, when we have a $|\mathcal{C}|$-way category-level classifier, our results show that using the best categorization features for identification does not perform as well as jointly choosing features for the overall task of categorization and identification.

Even though the relative improvement becomes more modest as mor features are added, the problem of determining the optimal allotment of features to the independently trained categorization and identification models remains. An issue with the independent method is that when there is a prior feature bandwidth limitation, the optimal feature allotments will undoubtedly depend on the task. To determine the number of features that should be used for categorization requires an additional level of optimization that the joint and factored methods do not.

One of the interesting properties of the flat joint and factored methods' performance is the improvement over the independent method, particularly when there are fewer features available; the accuracy of the joint (flat and factored) feature selection strategies ramp up much more quickly. Even as more features are added the relative improvement of the factored method over the independent remains above 20% (bottom-right, Figure 6).

The reason for the improvement can be seen by considering a feature selected by one model (the best gain for that model) and examining that feature's gain for the other models. In the first round of feature selection, the top categorization feature is also a very good feature for the factored joint model, and vice versa. This happens because most instances are background, and it is important to be able to do categorization early. Furthermore, because any feature will likely discriminate among some characters or vehicles better than no features, the top categorization feature also has modest value (about 70% of the maximum gain) for both character and vehicle identification. However, by the eighth iteration the best categorization feature often has almost no value for identification. With this strategy, by the time a character is detected, the features that have been computed will be of limited help in actually identifying the character. By contrast, the first feature selected by the factored joint model has modest value (also about 70% of maximum gain) for identification, but the second feature selected has high values for both categorization (95% of maximum gain) and character identification (92% of maximum gain). Adding this jointly optimal feature to the factored model thus not only aids in detecting instances of object categories, but very early on the system is also able to identify many more of them as well. When considered independently, however, the best feature for one task (e.g., categorization) is often not as good for another (e.g., character recognition).

Although the flat joint model is competitive with the factored model for small numbers of features, it begins to level off and diverge from the factored model. The flat joint model is quite cumbersome and takes much longer to train since it has to discriminate between every class label. Consequently, this model is not likely to scale well to more categories and identities; so it is not overly important that we do not yet know its performance for more features.

The bottom-left of Figure 6 shows that the flat joint model, which does not explicitly consider categorization, does not necessarily select features that help with the more general task; it performs much worse than the factored model at categorization in the face of limited feature bandwidth or computational time. Also interesting is that the factored model, which must consider the subsequent identification tasks, performs comparably to the model that uses only the best categorization features.

The typical approach to detection/categorization and recognition/identification is sequential. Under such a strategy, the independent detector for the two category (text and background) problem selects 20 features before the model likelihood plateaus, while the independent character recognizer selects 35 *different* features. For any window detected as text, the detector will have calculated 20 features, and then an additional 35 features must be calculated for recognition. Since the prior probability for text is very small, the total additional computation is modest. However, as the number of object categories grows (as in Figure 2), the number of queries to category-specific recognizers gets much larger, and the impact of additional feature computation for recognition becomes non-negligible. Figure 6 demonstrates that for a given performance level, the independently trained classifiers require the computation of more features.

The character-level detection and recognition is interesting, and performs reasonably on real images given the lack of context provided. Adding spatial and/or linguistic context in the form of a more complicated Markov model or even simple post-processing should greatly improve the quantitative results.

7. Conclusions

A typical approach to image understanding involves training system components individually. Unfortunately, errors propagating through sequential systems can have compounded negative effects. Furthermore, if resources (e.g., features) can be shared among the components, training components independently can make their resources too specialized to be useful for other tasks. Therefore, we have proposed to extend the idea of shared feature selection beyond multiclass detection or categorization to include the more specific recognition task.

We have laid out three frameworks for feature selection—the usual, which selects features independently for the detection/categorization and recognition/identification tasks, and two others that jointly select features for the entire detection and recognition task, with one factored or hierarchical and the other flat. Our results show that consideration of the entire end-to-end task yields greater accuracy with the same number of features. Furthermore, in a system with limited computational resources, joint feature selection obviates the need to optimize feature allocation for different tasks. Our factored model explicitly incorporates the categorization task, which we have shown to be important for category accuracy in the face of limited time or feature bandwidth.

In more general systems, there will be many detection and recognition tasks. The benefit of multi-purpose discriminative features for these systems should be even larger than demonstrated here. With more complex object classes to detect, knowledge of individual members can help boost detection rates, but more importantly, having features that are useful for multiple tasks can greatly reduce the necessary amount of computation.

While recent research has focused on developing high accuracy, specialized systems for tasks such as face detection, our results indicate it may be time to consider returning to frameworks that allow joint training of these powerful new models on broader, end-to-end tasks.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.

[2] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[3] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

[4] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 39:139–202, 1980.

[5] A. B. Hillel, T. Hertz, and D. Weinshall. Efficient learning of relational object class models. In *Proc. Intl. Conf. on Computer Vision*, volume 2, pages 1762–1769, 2005.

[6] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proc. Intl. Conf. on Computer Vision*, pages 604–610, 2005.

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, Nov 1998.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[9] O. C. Ozcanli, A. Tamrakar, and B. B. Kimia. Augmenting shape with appearance in vehicle category recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 935–942, 2006.

[10] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–71, 2000.

[11] J. A. Rehling. *Letter Spirit (Part Two): Modeling Creativity in a Visual Domain*. PhD thesis, Indiana University, July 2001.

[12] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 994–1000, 2005.

[13] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *IEEE International Conference on Image Processing*, volume 3, pages 444–447, 23-26 Oct. 1995, Washington, DC, USA, 1995.

[14] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 762–769, 2004.

[15] S. Ullman, E. Sali, and M. Vidal-Naquet. A fragment-based approach to object representation and classification. In *International Workshop on Visual Form*, number 2059 in LNCS, pages 85–102, 2001.

[16] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004.

[17] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Proc. Intl. Conf. on Computer Vision*, pages 1800–1807, 2005.