

Object Class Recognition with Many Local Features

Scott Helmer and David G. Lowe

Department of Computer Science
University of British Columbia
Vancouver, BC
V6T 1Z4, Canada

Abstract

In this paper we present a method to recognize an object class by learning a statistical model of the class. The probabilistic model decomposes the appearance of an object class into a set of local parts and models the appearance, relative location, co-occurrence, and scale of these parts. However, in many object classification approaches that use local features, learning the parameters is exponential in the number of parts because of the problem of matching local features in the image to parts in the model. In this paper we present a learning method that overcomes this difficulty by adding new parts to the model incrementally, using the Maximum-Likelihood framework. When we add a part to the model, a set of candidate parts are selected and the part that increases the likelihood of the data the most is added to the model. Once this part is added to the model, the parameters for all parts up to this point are updated using EM. The learning and recognition in this approach are translation and scale invariant, robust to background clutter, and has less restriction on the number of parts in the model. The validity of the approach is demonstrated on a real world dataset, where the approach is competitive with others, and where the learning for a rich model is much faster than previous approaches.

1. Introduction

Recognizing object categories is one of the oldest problems in computer vision and remains one of its most challenging. Conceptually the problem is difficult, regardless of the implementation, due to the undefined nature of similarity. Given the sheer variety of object classes, defining a similarity metric apriori that can separate all object classes is nearly impossible. A working assumption in visual recognition is that an object category has a set of features that are shared by objects in the class, but are shared less often by objects outside of the class. The task is thus to identify or to learn through the visual appearance some aspect of these features. In this paper we are concerned particu-

larly with learning models to recognize semi-rigid objects from a single viewpoint such as faces, cars, and motorbikes and where the recognition is robust to clutter, occlusion and certain transformations on the image coordinates.

One paradigm that has recently emerged with some success tries to solve the problem by modelling objects as a collection of parts [1,3,4,5,7,8,9]. In this case, the object classification system typically models the appearance, co-occurrence, and spatial relations of these parts. The advantage of this type of approach is that it is robust to clutter and occlusion, and in many cases can be made to be robust to affine transformations of image coordinates. Moreover, it has the attractive possibility of capturing the actual underlying parts of the object. The difficulty, though, lies in learning the parameters for the model. Given a dataset containing images with instances of the object class, any region of an image of the object could be a potential part. Clearly one does not want to explore this huge space to learn which parts are best in the context of recognition. Instead, most approaches select interesting regions of an image, characterize the local region in some way that includes information about scale and location, and then represent the image only by these local region descriptions. From these collections of features, the task is then to learn parts that presumably account for the appearance and spatial relations of the features, or some subset of these features.

The approach taken in this paper is similar to the one taken by Fergus *et al.* [3], in which the appearance, spatial relations, and co-occurrence of parts are learned simultaneously. They define for each part its appearance density, a shape density, a density on the scale of the part, and a joint density for the co-occurrence of parts. However, no initial matching is available between features in the image to parts in the model, so they define the probability of a set of features from an image containing an instance of an object as the sum of the probability of the data over all possible matchings. Evaluating this probability is exponential in the number of parts in the model. Intuitively though, if the model is accurate, then only a few matchings should have a

high likelihood, so the probability can be efficiently approximated. However, this only holds for the recognition phase. In the approach of Fergus *et al.*, they learn the parameters in a Maximum Likelihood framework, using the Expectation-Maximization [2] algorithm and initializing randomly. By doing this they are forced to evaluate an exponential number of matchings for the expectation step, and this is primarily because the likelihood of different matchings is much more uniform at this stage with no dominant peaks. As a result, their approach is limited to just 6 or 7 parts, which typically requires a day to learn.

The focus of this paper is to extend the approach to allow learning a model with many parts by overcoming the exponential nature of the matching problem. Moreover, unlike the approach of Fergus *et al.* the approach presented in this paper is also translation and scale invariant, and relatively fast to learn. We use a similar probabilistic model, except instead of learning all of the parts simultaneously, we learn them incrementally. We begin with a model which has a few parts that are accurate, that is, they have tight distributions on scale, appearance, and location. Each time a part is added to the model, the feature locations in every image are transformed into the model coordinate space using the most likely matches from features to parts. From here we sample a number of possible parts, try adding each part to the model, and measure the increase in the likelihood. We can do this efficiently because we store the information collected during previous iterations. The part that increases the likelihood the most is then added to the model. We then update the model using EM, but since there is a dominant matching for the previous parts, we can use this information to reduce the number of computations in the expectation step. As a result, the learning is fast, the model can be complex, and the variances on the appearance and locations are kept tighter.

2. Probabilistic Model

The probabilistic model we use to recognize an object class is almost identical to the one proposed by Fergus *et al.* [3]. We model an object as a set of parts with parameters θ , where each part has a density function for scale, location, and appearance, all of which are Gaussian. We represent an image as a set of features that are extracted from the image using the SIFT operator [5, 6], where each feature contains information on shape, scale, and appearance. To perform object detection, we do matchings between features and parts, evaluate their likelihood given the model, and use this measure to determine if the object is present in the image or not by comparing it to the likelihood that all of the features belong to the background. Note that for clarity we refer to parts only in relation to the model, and features only in regards to images.

A feature i is composed of a data vector that contains appearance \mathbf{A}_i , scale \mathbf{S}_i , and locations \mathbf{X}_i . So an image is represented as a set of extracted features \mathcal{F} with locations \mathbf{X} , scales \mathbf{S} , and appearances \mathbf{A} .

Furthermore, we define \mathbf{h} as a hypothesis vector where $\mathbf{h}(i) = j$ means that feature j is matched to part i in the model. In other words, \mathbf{h} is a potential matching. If there is no feature in the image that matches part i , then $\mathbf{h}(i) = 0$.

So, given a matching \mathbf{h} in the image, we define

$$p(\mathbf{X}, \mathbf{S}, \mathbf{A} | \mathbf{h}, \theta) = p(\mathbf{A} | \mathbf{h}, \theta) p(\mathbf{X} | \mathbf{h}, \theta) p(\mathbf{S} | \mathbf{X}, \mathbf{h}, \theta) \quad (1)$$

which is a probabilistic measure of how good the features from the image match the parts that they were assigned to in \mathbf{h} . It can be seen that we assume that the appearance, and location of parts are independent given the matching. Note, however, that the number of features extracted is not fixed. In order to define a proper probability we have to define the domain as all possible feature combinations in both the number of features present, up to some maximum, and their data vectors. To achieve this we can add a term with a uniform distribution over the number of features. This however has no affect upon the learning nor recognition, so we omit this for clarity of presentation.

In reality we are not given the assignment of parts in the model to features in the image, so to determine the probability of \mathcal{F} given θ , we can simply sum over all possible hypotheses.

$$p(\mathcal{F} | \theta) = \sum_{\mathbf{h} \in \mathcal{H}} p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h} | \theta) = \sum_{\mathbf{h} \in \mathcal{H}} p(\mathbf{A} | \mathbf{h}, \theta) p(\mathbf{X} | \mathbf{h}, \theta) p(\mathbf{S} | \mathbf{X}, \mathbf{h}, \theta) p(\mathbf{h} | \theta) \quad (2)$$

The Achilles heel of this approach is that the cardinality of \mathcal{H} is exponential in the number of parts, P , that is $O(|\mathcal{F}|^P)$. We should note however, that we actually are trying to achieve a model of an object class such that for a typical image of an object there is one dominant matching from features to parts. We want tight distributions on both shape and location, since loose distributions will result in a greater false positive rate. This is especially true in approaches that try to achieve translation invariance. Now if we have a sufficiently accurate model, an image with the object class in it will only have a few matchings that contribute to the value of (2). So, to get an estimate of (2), one can use A* search by first evaluating matches based solely upon appearance, and using bounds provide by these matches to explore the hypothesis space. For an accurate model, an accurate estimate of (2) can be provided relatively quickly.

2.1. Feature Extraction

The features extracted from an image are based upon Lowe's SIFT features [5, 6]. In this approach, candidate

keypoints are identified by finding peaks in the difference-of-Gaussian function convolved with the image in scale space. In essence, keypoints are located in regions and at scales where there is a high amount of variation, which means these locations are likely to contain useful information for matching. Moreover, since they are located at peaks, minor variations in the region surrounding the keypoint won't greatly affect its location. As a result, intra-class variability for a part will be less likely to affect the location of the corresponding image feature. From an image feature identified by the SIFT procedure, we know its location and scale, and we extract a local description of the region around this feature. Rather than represent the local region as the pixel values, or some straightforward compression like PCA, we have chosen to represent them in a manner identical to Lowe's SIFT features. Here a local region is divided into K smaller regions, and each region is described by a histogram of size Q of image gradients in that region. These appearance descriptors are invariant to small changes in position of the keypoint, and also brightness changes. In this paper we have chosen $K = 4$ and $Q = 8$, so the description of a local region has 32 dimensions.

2.2. Appearance

The appearance of each feature extracted from an image lies in a 32 dimensional space where each dimension is an integer between 0 and 255. Given a matching \mathbf{h} , we evaluate the appearance of a feature f according to the density for part p only if $\mathbf{h}(p) = f$, and if f is assigned to no part then it is evaluated according to the background distribution. Each part p is modelled by a Gaussian with a mean μ_p^{app} and covariance Σ_p^{app} , and we assume that the parts are independent given \mathbf{h} and so Σ_p^{app} is diagonal. We represent the probability of \mathbf{x} under a Gaussian distribution with mean μ and σ as $G(\mathbf{x}|\mu, \sigma)$. The background model is modelled with a uniform distribution, but because keypoints are only found at regions of variability, it isn't a 256^{32} space. Instead we approximate it as $p(\mathbf{A}_{\mathbf{f}}|f \in \text{background}) = \alpha$, which is a global constant that is determined experimentally. So, if we had n features that were not matched to some part in \mathbf{h} , then

$$p(\mathbf{A}|\mathbf{h}, \theta) = \alpha^n \prod_{p|\mathbf{h}(p) \neq 0} G(A_{\mathbf{h}(p)}|\mu_p^{app}, \Sigma_p^{app}) \quad (3)$$

2.3. Location

One of the distinctive differences between our approach and that of Fergus *et al.*, is that ours is translation and scale invariant in both the recognition and learning phases. To achieve this a distinction is drawn between an image coordinate space and a model coordinate space. To evaluate a

hypothesis in regards to the location of features, we linearly project the locations of the features to model space and evaluate the locations in the model space. For a proper probability distribution, we must evaluate the hypothesis under all such transformations. A linear transformation, \mathbf{t} , shifts feature locations by x and scales them by s . So we have,

$$\begin{aligned} p(\mathbf{X}|\mathbf{h}, \theta) &= \int_{\mathbf{t}} p(\mathbf{X}, \mathbf{t}|\mathbf{h}, \theta) \\ &= \int_{\mathbf{t}} p(\mathbf{t}(\mathbf{X})|\mathbf{h})p(\mathbf{t}|\mathbf{h}) \end{aligned} \quad (4)$$

Here we assume that $p(\mathbf{t}|\mathbf{h})$ is uniform, so $p(\mathbf{t}|\mathbf{h})$ is constant, and for technical correctness we must restrict the domain of the transformations. Regardless, (4) is difficult to evaluate, so we make the approximation

$$p(\mathbf{X}|\mathbf{h}, \theta) \simeq \underset{\mathbf{t}}{\operatorname{argmax}} p(\mathbf{t}(\mathbf{X})|\mathbf{h}) \quad (5)$$

This means that we are approximating the probability of the feature locations based solely on their best projection into model space. Now, given the optimal transformation \mathbf{t}_{opt} and a hypothesis \mathbf{h} , we assume that the location of parts are independent. As in appearance, for features that are not assigned to a part we simply evaluate their location according to the background distribution, which is uniform with constant β which is dependent only on the size of the image. Each feature that is assigned to a part in the hypothesis has its location evaluated by the distribution of its respective part. Each part p has a Gaussian density with a mean μ_p^{loc} and covariance matrix Σ_p^{loc} , where we assume the Σ_p^{loc} is diagonal. Let n be the number of features not assigned to parts in \mathbf{h} . So,

$$p(\mathbf{t}_{\text{opt}}(\mathbf{X})|\mathbf{h}) = \beta^n \prod_{p|\mathbf{h}(p) \neq 0} G(\mathbf{t}_{\text{opt}}(\mathbf{X}_{\mathbf{h}(p)})|\mu_p^{loc}, \Sigma_p^{loc}) \quad (6)$$

Since

$$p(\mathbf{t}(\mathbf{X})|\mathbf{h}) = G(\mathbf{t}(\mathbf{X})|\mu_p^{loc}, \Sigma_p^{loc}) \quad (7)$$

The transformation, \mathbf{t}_{opt} , that optimizes this is the weighted least squares solution of fitting the matched features to the parts, where the weights are the variances on location.

2.4. Scale

Again, given a hypothesis \mathbf{h} we assumed the scale of the parts are independent, and each part is modelled using a Gaussian with mean μ_p^{scale} and standard deviation σ_p^{scale} . However, we also use the positions \mathbf{X} to determine \mathbf{t}_{opt} to transform the feature scales into model coordinates. In this case we multiply the scales by whatever scaling factor was

used in the linear transformation \mathbf{t}_{opt} . We assume that if a feature is not assigned to a part in \mathbf{h} then it is part of the background, and once again we assume the background is a uniform distribution over possible scales, ψ . So the likelihood of the scales given a hypothesis and feature locations is

$$p(\mathbf{S}|\mathbf{X}, \mathbf{h}) = \psi^n \prod_{p|\mathbf{h}(p) \neq 0} G(\mathbf{t}_{\text{opt}}(\mathbf{S})|\mu_p^{\text{scale}}, \sigma_p^{\text{scale}}) \quad (8)$$

2.5. Part Statistics

Given the model, we simply assume that the presence or absence of parts are independent. Although there are certainly some interesting and valuable co-occurrence patterns present in some object classes, it is an issue we leave for future research. So, the probability that part p is present is ϕ_p . For a particular matching \mathbf{h}

$$p(\mathbf{h}|\theta) = \prod_{p|\mathbf{h}(p) \neq 0} \phi_p \prod_{p|\mathbf{h}(p) = 0} (1 - \phi_p) \quad (9)$$

3. Learning

The primary contribution of this paper is in how the parameters for the model are learned. In the following we present a method to learn the maximum likelihood parameters for the model. That is, for images I , we seek model parameters θ such that

$$\prod_{i \in I} p(\mathcal{F}_i|\theta) \quad (10)$$

is maximized. In the approach of Fergus *et al.* [3], the parameters are learned in a maximum likelihood framework via the EM algorithm, first initializing the model parameters randomly. However, in the Expectation step they must compute

$$p(\mathbf{h}|\mathbf{X}, \mathbf{A}, \mathbf{S}, \theta) = \frac{p(\mathbf{X}, \mathbf{A}, \mathbf{S}|\mathbf{h}, \theta)p(\mathbf{h}|\theta)}{p(\mathbf{X}, \mathbf{A}, \mathbf{S}|\theta)} \quad (11)$$

for all possible hypotheses \mathbf{h} in order to do the parameter updates in the Maximization step. Since the distribution $p(\mathbf{h}|\mathbf{X}, \mathbf{A}, \mathbf{S}, \theta)$ is roughly uniform initially because of the random initialization, \mathbf{A}^* is of little use in making this computation tractable.

An alternative approach is to increase the complexity of the model incrementally. That is, begin with a model with a small number of selected parts, learn the parameters for this model, and then add a new part one at a time.

The basic approach is as follows,

1. Begin with a model of 1 or more parts.

2. Until P parts or desired accuracy:

- (a) Find dominant matchings for every image.
- (b) Sample n features from the dataset as potential new parts
- (c) For each of these features, try adding it as a part to the model.
- (d) Measure the likelihood of the data with this potential part.
- (e) Select the part that increased the likelihood of the data the most and add it to the model.
- (f) Run EM to update parameters.

3.1. Initialization

The first step can be done in a number of ways. One approach would be to select the feature that had the highest density around it in appearance space, where the weightings on each dimension are equal. This is equivalent to selecting the feature that would maximize the likelihood of the data for our model, where it was the only part of the model and we drop terms on location and scale. This is the approach taken in the experiments. This approach only works, however, if there is a feature that is common to many of the images. Another approach would be to match some of the images in the dataset to other images in the dataset, and select the parts that tend to appear in the matchings. This approach and others are currently being investigated.

3.2. Incremental Learning

The second step of the approach avoids constructing a model with loose variances, which will result in an increase in the number of significant matchings \mathbf{h} for each image. To achieve this we utilize what the model has learned so far in order to transform the locations and scales of the features of each image into model coordinate space. With this information we can then sample from the dataset to initialize a new part for which its appearance, and the location and scale in model coordinate space have a high number of matches in the dataset.

To achieve this we first, in step 2a, find dominant matchings in each image, which is when an image has a matching \mathbf{h} for which $p(\mathbf{h}|\mathbf{X}, \mathbf{A}, \mathbf{S}, \theta)$ is much greater than other matchings. We then transform the feature locations and scales into model space according to \mathbf{t}_{opt} for this dominant matching. If for a particular image there are a few dominant matchings we use a weighted average to determine locations and scales of the features, where the weights come from $p(\mathbf{h}|\mathbf{X}, \mathbf{A}, \mathbf{S}, \theta)$. For an image where there is no dominant matching, then the feature locations and scales are not transformed into model coordinate space. In this

case, the image is not used as a source of potential parts in this iteration. It should also be noted at this point that dominant matchings also contribute the most to the measure of the likelihood of the data according to the model. This is important because it allows for fast computation of the likelihood of the data which is needed in step 2d and in step 2f.

In the next step, 2b, we could possibility sample every feature in the data as a potential part but this would be too computationally intensive. In order to reduce the computation we sample only n features as possible parts, where this n is generally on the order of the average number of features in an image. To ensure that we are sampling good features, and thus not wasting computation, there are a number of heuristics that can be employed. One very simple tactic is to sample features from images that have dominant matchings, since in this case we have good information on the features locations in model coordinate space. Another approach is to restrict sampling features to regions that are relatively close to the features that are matched to parts in the model. In this case we can be more confident that we are not sampling features that are part of the background.

In order to measure how good a sampled feature would be as a part, we can temporarily add it to the model and measure the increase in the likelihood. We set the temporary part's parameters, μ_p^{app} , μ_p^{scale} , and μ_p^{loc} as the feature's appearance vector, the transformed scale, and the transformed locations. In the experiments we set variances for potential parts as $\Sigma_p^{app} = \text{diag}(2000)$, $\Sigma_p^{loc} = \text{diag}(25)$, and $\sigma_p^{scale} = \mu_p^{scale}/10$, where $\text{diag}(c)$ is a matrix with diagonal entries c . In order to calculate the likelihood, it is not necessary to find the dominant matchings from scratch. From previous computations we already have the dominant matchings for parts 1 to $p-1$, so it is really just a question of finding the features that match to part p to find the new dominant matching for parts 1 to p .

The part that is added to the model is that sampled feature that increased the likelihood of the data the most. At this point, step 2f, we run the Expectation Maximization algorithm on all of the parameters of the model until it converges. As more parts are added to the model, the convergence is fast, since most of the changes to the parameters are occurring on the new part.

3.3. Discussion on Incremental Learning

The important insight as to why this approach avoids the exponential aspect of the matching problem, is that before a new part is added to the model we have only a few hypotheses \mathbf{h} for which $p(\mathbf{h}|\mathbf{X}, \mathbf{A}, \mathbf{S}, \theta)$ is significant, for each image. As a result, when we add a new part to the model, we only have to determine if there are features that match this new part such that it improves our previous matching.

In addition it should be noted that incremental learning

is greedy in the manner in which it selects a new part to add to the model, and thus there is some concern that the final solution is not a globally optimal solution. Recall, however, that the objective function we seek to minimize, (10), with respect to the model parameters θ is such that it has many local minima. So, applying the Expectation Maximization algorithm on a random initialization, as in Fergus *et al.*, results in a solution that is a local minimum, usually arrived at in an exponential amount of time. The approach to learning presented in this paper seeks to both arrive at a solution quickly, but also at a desirable solution. With a high number of parts in the model, EM with random initialization will result in a model that is slowly learned and contains parts that are useless in recognition. In the incremental version, however, a new part is added to the model when there are a number of features in the dataset that match it closely, so the variances remain small.

Another point to note is that in the early stages of model construction, the model may not account for much of the data. The first few parts chosen by the model may only have features in a subset of the images. However, with each part that is added, this subset is expanded. This poses a new problem though, since if the object class is such that there are very distinct subclasses, then the model may only model one of the subclasses. An example of this is the class of vehicles, where cars are visually distinct from motorbikes. One way to deal with this would be to alter the algorithm to detect if it is only modelling a certain subset of the dataset. If this is the case, the algorithm could then split the data, and model the data separately. Research into this scheme is ongoing.

Another potential problem with the incremental approach is that in the early stages the approach could pick poor parts and thus construct an unsatisfactory solution. As much as this is a legitimate concern, it is also the case that random initialization will sometimes settle on a unsatisfactory solution, and our experiments indicate that this occurs frequently with random initialization. In the case where the incremental approach has arrived at a poor model in the first few iterations, this can easily be detected by examining the variances on location and scale. In these cases it would be easy to have the system automatically restart.

4. Experiments

The bulk of the experiments presented in this paper were performed on the face data set utilized in the Fergus *et al.*[3]. This dataset is composed of 450 grayscale images, with about 30 subjects, taken under different lighting conditions and background conditions. For training and testing, the data sets were randomly separated into testing and training sets, and the models were trained on the training images. The split was 350 training images to 100 testing images.

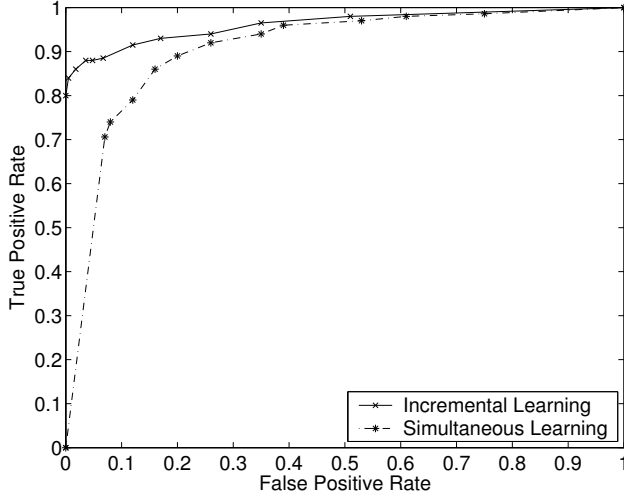


Figure 1: The graph displays the RPC curve for a model with five 5 parts, comparing the incremental approach to simultaneous learning.

For experiments where we report the recall-precision curve (RPC) we train five models, each with a different splitting of the data set, in order to obtain more accurate results. To obtain data on false positive rates, the models were also tested using the same background greyscale images in the Fergus *et al.* Moreover, each of the models trained with the incremental learning were initialized with a single part. This part was the sampled feature for which the likelihood of the data based solely on the appearance was greatest, where the part had μ_1^{app} set as the feature's appearance vector, and $\Sigma_1^{app} = \text{diag}(2000)$. We set $\alpha = \exp(-170)$, the background appearance density, and this value was arrived at by investigating the distribution of appearance for SIFT features in a large database of SIFT features.

In the first set of experiments we compare the approach presented in this paper to an approach that learns all of the parts simultaneously using EM. The simultaneous approach is very similar to Fergus *et al.*, with only slight changes in the probabilistic model. The difficulty in doing the comparison is that our approach is translation invariant, whereas the approach that learns the parts simultaneously is not. To overcome this problem, we transform the locations of the features in each image so that the object is in the centre of the image. We also normalize for scale. In the Fergus *et al.* experiments, they make the same transformation to the data, and train their model in much the same manner. To contrast this with our approach, a model was trained using the incremental approach, except that no preprocessing transformations are made on the data.

The first thing to note is that training the parts simultaneously tended to be very inconsistent, where most of the time the variances on location grew very large, which es-



Figure 2: These 4 figures represent typical images and the features that were in the maximum matching.

entially means that the data is being clustered based on appearance alone. As a result, these models had a high false positive rate. This demonstrates that EM with random initialization suffers as a result of its random initialization. This is certainly a motivation to use an approach that learns a model with many features with tight distributions rather than a model with only a few parts and wide distributions. This result, however, seems to differ somewhat from the results in Fergus *et al.*. The reason for this could be because we detect many more features using SIFT features, on the order of 300, compared to the feature detector used by Fergus, which they limit to just 20. It could also possibly be that the dimensionality of the appearance space is too large. For comparison we report results for the best model for the simultaneously learning approach, which did not have the large variances.

To make the actual comparisons, there is a subtle point to notice. In a translation invariant model, an instance of the object could be found anywhere in the image. In a model that requires the object to be centred, it will only look for the object in that one location. As a result, the number of false matches for the translation invariant model will naturally be higher since it can find a match anywhere in an image. As a result, to make a proper comparison, during classification using the model learned by the simultaneous approach the feature locations are transformed into the model coordinate space using the procedure discussed in section 2.3.

Figure 1 shows the recall-precision curve for a model with 5 parts, which we use for efficiency since a larger number of parts requires a significant amount of training time for the simultaneous learning approach. The incremental approach took just 20 minutes on a 2.5 GHz machine, while the simultaneous approach took 2 hours and failed to converge on a satisfactory solution occasionally. It can be seen that the incremental approach outperforms the simultaneous approach. This result may be caused in part by the fact that

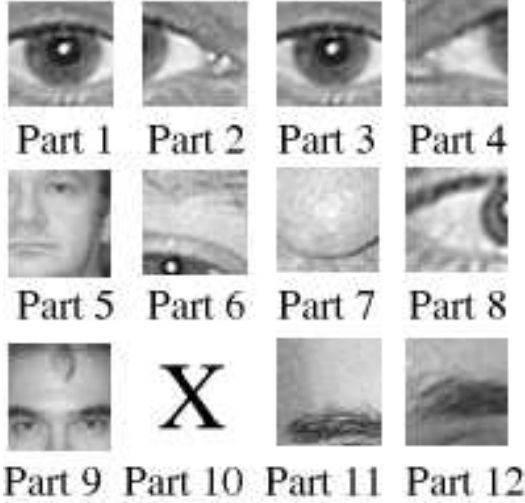


Figure 3: Sample of features from dataset that best match each part for the model trained on with 12 parts. Note that the 10th part is a part with a very wide variance. The parts that correspond to the left side of the face are 2, 3, 8, and 11. The parts that correspond to the right side of the face are 1, 4, 6, and 12. Part 7 corresponds to a nose.

the simultaneous approach to learning relies on human intervention to centre the data. This may introduce additional noise to the data, and thus the model may be less accurate. Regardless, the approach presented in this paper produces a model with better classification accuracy, is much faster to learn, and is translation invariant in both the learning and recognition.

The second experiment is designed more as an illustration of the power of the approach. In this experiment a model for the face dataset with 12 parts was trained. The results are encouraging in regards as to how the approach scales. The time it took for the model to learn 12 parts was 1 hour, where as the time it takes to train a 5 part model is about 20 minutes. Larger models take a similar linear increase in time to learn. Figure 2 shows some example faces with the features that were matched to parts in the model circled in white. On average, about 300 SIFT features were extracted per image.

The model with 12 parts had an impressive RPC curve, achieving a true positive rate of 0.96 and false positive rate of just 0.02. The faces that were not recognized in the dataset were examples of faces taken under lighting conditions that were highly uncommon in the dataset. The features extracted from these images tended to be located in different places and at different scales than most images. As a result it's not that the parts do not exist in the image, but rather that the SIFT feature detector did not extract them from the image.

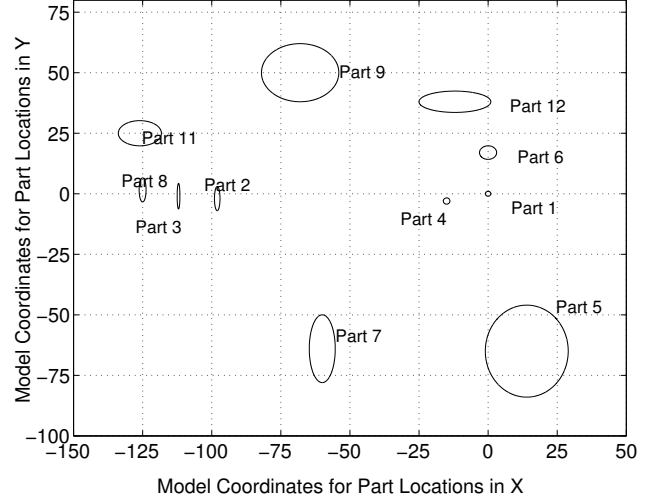


Figure 4: This figure is a representation of the part locations in model coordinate space. The centres are the mean location of each part, and the standard deviation is represented by the ellipse.

There are a number of interesting things to note. First is that the model has a surprisingly tight distribution on location, as shown in Figure 4 which shows the location model for the parts. The centre of the ellipse for part p is μ_p^{loc} , and the ellipse represents the standard deviation σ_p^{loc} . Notice how the variances for the parts corresponding to the eyes are very small. The reason for this is that using an optimal linear transformation to transform feature locations into model coordinate space introduces a bias. If at sometime in the learning a particular part has a very small variance on its location, then the optimal transformation will always place the features matching this part very close to where the part lies in model space. Since the first few parts added to the model were colinear, the variances at the beginning were small for these parts and so this persisted. Future work can overcome this by regularizing the variances so that they do not become too small.

Figure 3 shows an example of features in the data that match each part closely. It can be seen that a variety of local regions are represented. One interesting point is that for part 5, the variances on the appearance are very high in regions where the background is part of the feature. Another interesting point is that part 10 does not clearly match any feature. It is expected that as more parts are added to the model, then more noisy parts like this will occur because of over fitting. Parts such as 10, however, don't affect the false positive rate greatly since the wide variance means any false matches to this part will contribute little the overall likelihood. It is those parts that have tighter distributions on location and scale that contribute the most to the likelihood.

Part #	μ_p^{scale}	σ_p^{scale}	ϕ_p
1	6.15	0.92	0.87
2	3.21	0.53	0.44
3	6.24	0.15	0.88
4	3.27	0.048	0.53
5	25.12	12.52	0.725
6	4.95	0.98	0.56
7	7.03	1.72	0.36
8	3.85	1.37	0.52
9	41.81	17.63	0.41
10	7.35	3.95	0.64
11	4.95	1.22	0.67
12	5.76	2.85	0.41

Table 1: The parameters that were learned for a 12 part model, where μ_p^{scale} , σ_p^{scale} , ϕ_p are mean scale, standard deviation on the scale, and probability of presence for part p .

Table 1 contains other statistics about the model, like the scale parameters and the occurrence parameters. Notice how the parts that are detected initially are, in general, the most common parts detected, as indicated by ϕ_p , and also have the tightest distributions on location and scale.

Other experiments are currently being conducted on the motorcycle and car datasets. Results are encouraging, but temperamental. The reason for this is that this approach relies on the fact that the first few features are good features, even if these features only occur in a subset of the dataset. In some cases, the approach to select the initial model fails to select good initial features, which causes the distributions to become wider, and thus no dominant matchings emerge. We are currently investigating better ways to construct the initial model.

5. Summary and Conclusions

The approach to learning presented in this paper overcomes some of the difficulties of the matching problem that is inherent in approaches that recognize objects using local features. Experiments have shown that the time to learn the model is almost linear in terms of the number of parts in the model, as opposed to exponential as in some methods. This allowance for a greater number of parts will make it possible to learn models with a much richer appearance. Examples of this would be models of faces under many different lighting conditions, with novelties like sunglasses, and possibly an extension to multi-viewpoint models. Moreover, the results have shown that it fares better in terms of recognition performance than a simultaneous approach. An additional contribution is that of presenting an approach that is both scale and translation invariant in both recognition and learning.

As with the approach of Fergus *et al.*, the approach is dependent upon the local feature detector. This is a significant limitation since many object classes are recognized more on the underlying shape than on appearance. Currently most local feature detectors are based on appearance, so further classes of feature detectors would be valuable.

The approach is also dependent upon beginning with a few good parts. For the face data set this did not pose a problem, but for data sets without a clear common part, the problem becomes more difficult. We are currently investigating reliable ways to extract such parts.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Institute for Robotics and Intelligent Systems (IRIS) Network of Centres of Excellence.

References

- [1] M. Burl, M. Weber, and P. Perona, "A probabilistic approach to object recognition using local photometry and global geometry," *European Conference on Computer Vision*, Freiburg, Germany (1998), pp. 628-641.
- [2] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, 39:1-38, 1976.
- [3] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," *Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin (2003), pp. 264-271.
- [4] L. Fei-Fei, R. Fergus, P. Perona, "A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories," *International Conference on Computer Vision*, Nice, France (2003), pp. 1134-1141.
- [5] D.G. Lowe, "Object recognition from local scale-invariant features," *International Conference on Computer Vision*, Corfu, Greece (1999), pp. 1150-1157.
- [6] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, in press (2004).
- [7] A.R. Pope, D.G. Lowe, "Probabilistic Models of Appearance for 3-D Object Recognition," *International Journal of Computer Vision*, 40(2), pp. 149-167.
- [8] S. Ullman, M. Vidal-Naquet, and E. Sali, "Visual features of intermediate complexity and their use in classification," *Nature Neuroscience*, 5(7), (2002), pp. 1-6.
- [9] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," *European Conference on Computer Vision*, Dublin, Ireland (2000), pp. 18-32.