Patch-based Within-Object Classification*

Jania Aghajanian[1], Jonathan Warrell[1], Simon J.D. Prince[1], Peng Li[1], Jennifer L. Rohn[2], Buzz Baum[2]
[1] Department of Computer Science, University College London
[2] MRC Laboratory For Molecular Cell Biology, University College London
[1]{j.aghajanian, j.warrell, s.prince, p.li}@cs.ucl.ac.uk [2]{j.rohn, b.baum}@ucl.ac.uk

Abstract

Advances in object detection have made it possible to collect large databases of certain objects. In this paper we exploit these datasets for within-object classification. For example, we classify gender in face images, pose in pedestrian images and phenotype in cell images. Previous work has mainly targeted the above tasks individually using object specific representations. Here, we propose a general Bayesian framework for within-object classification. Images are represented as a regular grid of non-overlapping patches. In training, these patches are approximated by a predefined library. In inference, the choice of approximating patch determines the classification decision. We propose a Bayesian framework in which we marginalize over the patch frequency parameters to provide a posterior probability for the class. We test our algorithm on several challenging "real world" databases.

1. Introduction

Recent advances in computer vision have allowed us to reliably detect objects with limited variation in structure such as faces, pedestrians and cars in real time. A typical approach is to use *sliding window* object detectors such as the work of Viola and Jones [19] and [14, 10, 8]. Sliding window object detectors consider small image windows at all locations and scales and perform a binary detection for each. The output of a sliding window object detector is a bounding box around the object of interest.

The success of these techniques allows us to collect large databases of such objects and it would be useful to subsequently describe their characteristics (attributes). For example we might classify gender in face images or phenotype in cell images. This "within-object" classification task has quite different characteristics to other forms of object recognition. All of the examples have a great deal in common and we aim to classify quite subtle differences (see figure 1).

*J.A. and S.P. acknowledge the support of the EPSRC ref: EP/E013309/1. B.B. acknowledges the support of the AICR ref: 05-341.

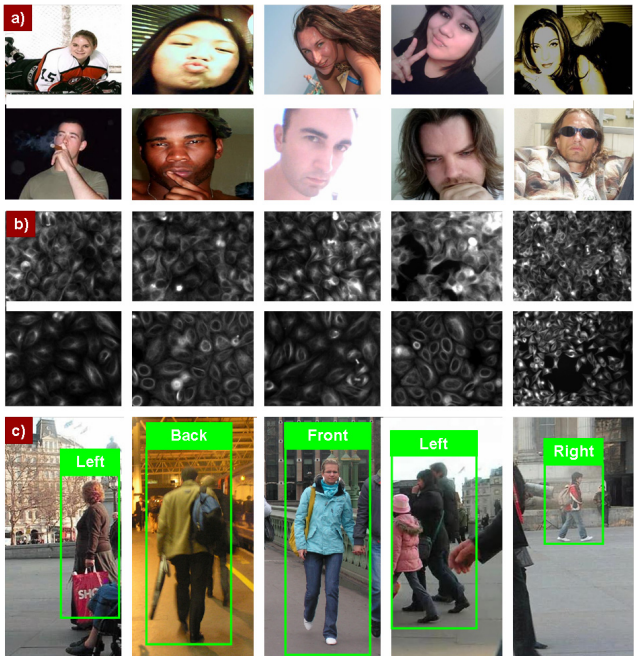

Figure 1. We address the problem of within-object classification on images captured in uncontrolled environments, where large within-class variations are present. For example, we classify (a) gender in face images, (b) phenotype in cell images and (c) pose in pedestrian images. These examples were all correctly classified.

Within-object classification has widespread applications including targeted advertising, consumer analysis, and medical image analysis. Examples include biological classification, where we might automatically screen cell cultures for diseases and gender classification which could be used as a preprocessing step in face recognition.

A large body of research has investigated the choice of the learning algorithm for particular within-object classification tasks including neural networks [9, 3], support vector machines [17, 16] and adaboost [18, 1]. Most current methods use tailor-made representations specific to the object of interest. For example Brunelli and Poggio [3], extract geometric features from faces such as pupil to eyebrow ratio, eyebrow thickness and nose width as input to a neural net-

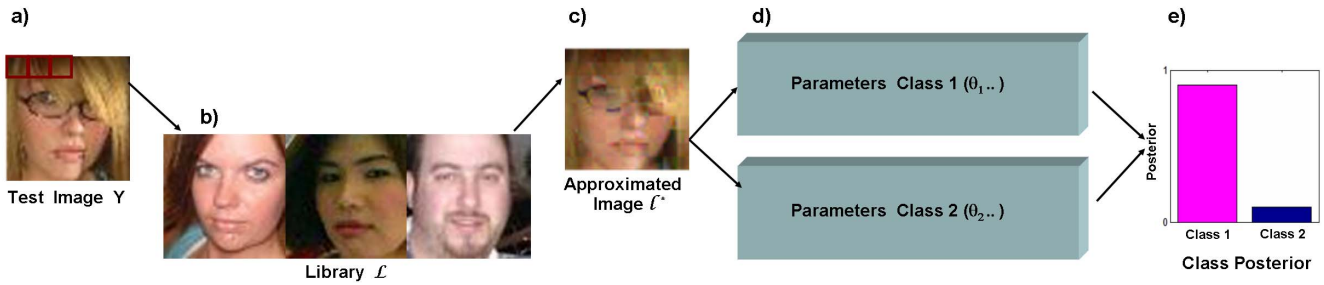

Figure 2. Inference. (a) A test image $\mathbf{Y}$ is decomposed into a regular patch grid. (b) A large library $\mathcal{L}$ is used to approximate each test image patch. (c) The choice of library patch provides information about the class. (d) Parameters θ associated with each class are used to interpret these patch choices and (e) used in a Bayesian framework to calculate a posterior over classes.

work to perform gender detection. Saatci and Town use Active Shape Models [17] to represent faces for gender classification. Similarly 2D contours and stick figures have been used to represent human body for motion analysis [13] and action recognition [6] . Domain specific features are also used in cell screening. These include aspects like the size, perimeter and convexity of cells [11] as well as the size and shape of the nuclei [4].

These techniques have several disadvantages. First, object specific representations cannot be applied to other problems without major alteration: most techniques have only been applied to a single class. Second, most methods do not exploit the large amounts of available training data (there are some exceptions, e.g. [12]). Instead they have mostly been investigated using small databases some of which contain images that are not typical of the real environment. For example, in gender classification, the FERET database is often used, although it does not contain the variations in pose, illumination, occlusion and background clutter seen in figure 1. It has been shown that performance of most methods drops sharply when tested on images captured in uncontrolled environments [15].

In this paper we propose a Bayesian framework for within-object classification that exploits very large databases of objects and can be used for disparate object classes. We build a non-parametric generative model that describes the test image with patches from a library of images of the same object. All the domain specific information is held in this library: we use one set of images of the object to help classify others. We test our algorithm on large real-world databases of faces, pedestrians and human cells.

In section 2 we describe the proposed method. Data collection and parameter selection is described in sections 3.1 and 3.2. In sections 3.3-3.6 we present classification experiments using face, cell and pedestrian images. Method comparison and summary is presented in sections 3.7 and 4.

2. Methods

Our approach breaks the test image into a non-overlapping regular grid of patches. Each is treated separately and provides independent information about the class label. At the core of our algorithm is a predefined library of object instances. The library can be considered as a palette from which image patches can be taken. We exploit the relationship between the patches in the test image and the patches in the library to determine the class. Our algorithm can be understood in terms of either *inference* or *generation* and we will describe each in turn.

In *inference* (see figure 2), the test image patch is approximated by a patch from the library $\mathcal{L}$. The particular library patch chosen can be thought of as having a different affinity with each class label. These affinities are learned during a training period and are embodied in a set of parameters θ associated with each class. The relative affinity of the chosen library patch for each class is used to determine a posterior probability over classes.

Alternatively, we can think about *generation* from this model. For example, consider the generative process for the top-left patch of a test image. The true class label induces a probability distribution over all the patches in the library based on the learned parameters for that class. We choose a particular patch using this probability distribution and add independent Gaussian noise at each pixel to create the observed data. In inference we are inverting this generative process using Bayes' rule to establish which class label was most likely to be responsible for the observed data.

2.1. Inference

Consider the task of assigning a class label $\mathcal{C}$ to a test image, where there are K possible classes so $\mathcal{C} \in \{1 \dots K\}$. The test image $\mathbf{Y}$ is represented as a non-overlapping grid of patches $\mathbf{Y} = [\mathbf{y}_1 ... \mathbf{y}_P]$. The model will be trained from I training examples $\mathbf{X_c}$ from each of the K classes. Each training example is also represented as a non-overlapping grid of patches of the same size as the test data. We denote the p^{th} patch from the i^{th} training example of the c^{th} class by $\mathbf{x}_{icp}$ (see figure 3a).

We also have a library $\mathcal{L}$ of images that are not in the training or test set and would normally contain examples of all classes. We will consider the library as a collection of patches $\mathcal{L}_l$ where $l \in \{1..N\}$ indexes the N possible

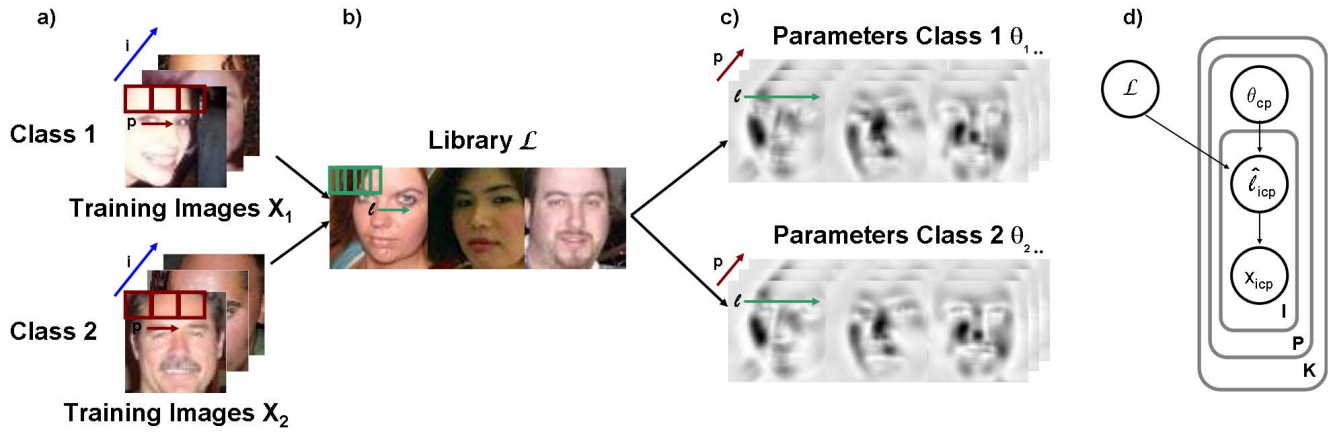

Figure 3. (a) The model is trained from I training examples from each of the K classes. Each training image is represented as a non-overlapping grid of patches denoted by p. (b) We also have a library $\mathcal{L}$ of images that are not in the training or test set and contains examples of all classes. The library is considered as a collection of patches $\mathcal{L}_l$ where $l \in \{1..N\}$ indexes the N possible sites. (c) The parameter θ_{cpl} represents the tendency for the patch from library site l to be picked when considering patch p of an example image of class c. (d) Graphical model representing our method.

sites from which we can take library patches (see figure 3b). These patches are the same size as those in the test and training images but may be taken from anywhere in the library (i.e. they are not constrained to come from a non-overlapping grid). In other words, the sites l denote every possible pixel position in the library images.

The output of our algorithm is a posterior probability over class label $\mathcal{C}$. We calculate this using Bayes' rule

$$Pr(\mathcal{C}=c|\mathbf{Y},\mathbf{X}_\bullet) = \frac{\prod_{p=1}^{P} Pr(\mathbf{y}_p|\mathcal{C}=c,\mathbf{x}_{\bullet cp})Pr(\mathcal{C}=c)}{Pr(\mathbf{Y})} \tag{1}$$

where we have assumed that the test patches $\mathbf{y}_p$ are independent. The notation $\bullet$ indicates all of the values that an index can take, so $\mathbf{X}_\bullet = \{X_1 ... X_K\}$ denotes the training images from all of the K classes and $\mathbf{x}_{\bullet cp}$ denotes the p^{th} patch from all I training images from the c^{th} class.

Although the likelihood in Equation 1 depends on the library, it is not conditioned on the parameters of the model θ. We take a Bayesian approach and marginalize over the model parameters, so the likelihood terms have the form:

$$Pr(\mathbf{y}_p|\mathcal{C}=c,\mathbf{x}_{\bullet cp}) = \int Pr(\mathbf{y}_p|\theta_{cp\bullet})Pr(\theta_{cp\bullet}|\mathbf{x}_{\bullet cp})d\theta_{cp\bullet} \tag{2}$$

where $\theta_{cp\bullet}$ are all of the parameters associated with the p^{th} patch for the c^{th} class.

To calculate the likelihood, we first find the index l^* of the library site that most closely matches the vectorized pixel data from the test patch $\mathbf{y}_p$. We are assuming that the test patch is a Gaussian corruption of the library patch and we can find the most likely site to have been responsible using maximum a posteriori estimation

$$l^* = \arg\max_l \mathcal{G}_{\mathbf{y}_p}[\mathcal{L}_l; \sigma^2\mathbf{I}] \tag{3}$$

where $\mathcal{L}_l$ is the vectorized pixel data from site l of the library $\mathcal{L}$. We define the likelihood to be

$$Pr(\mathbf{y}_p|\theta_{cp\bullet}) = Pr(l^*|\theta_{cp\bullet}) = \theta_{cpl^*} \tag{4}$$

From this it can be seen that the parameter θ_{cpl} represents the tendency for the patch from library site l to be picked when considering patch p of an example image of class c. This can be visualized as in figure 3c. A graphical model relating all of the variables is illustrated in figure 3d.

2.2. Training

In this section, we consider how to use the training data $\mathbf{x}_{\bullet cp}$ from the p^{th} patch of every image belonging to the c^{th} class to learn a posterior distribution $Pr(\theta_{cp\bullet}|\mathbf{x}_{\bullet cp})$ over the relevant parameters $\theta_{cp\bullet}$. In section 2.3 we discuss how to use this distribution to calculate the integral in Equation 2.

We calculate the posterior distribution over the parameters $\theta_{cp\bullet}$ using a second application of Bayes' rule:

$$Pr(\theta_{cp\bullet}|\mathbf{x}_{\bullet cp}) = \frac{Pr(\mathbf{x}_{\bullet cp}|\theta_{cp\bullet})Pr(\theta_{cp\bullet})}{Pr(\mathbf{x}_{\bullet cp})} \tag{5}$$

To simplify notation, we describe this process for just one of the P regular patches and one of the K classes and drop the indices c and p. Equation 5 now becomes:

$$Pr(\theta_\bullet|\mathbf{x}_\bullet) = \frac{Pr(\mathbf{x}_\bullet|\theta_\bullet)Pr(\theta_\bullet)}{Pr(\mathbf{x}_\bullet)} \tag{6}$$

where $\mathbf{x}_\bullet = \mathbf{x}_1 \ldots \mathbf{x}_I$ is all of the training data for this patch and this class and $\theta_\bullet = \theta_1 \ldots \theta_N$ is the vector of N parameters associated with each position in the library for this patch and this class.

To calculate the likelihood of the i^{th} training example $\mathbf{x}_i$ given the relevant parameters $\theta_\bullet$ we first find the closest matching library patch $\hat{l}_i$ where

$$\hat{l}_i = \arg\max_l \mathcal{G}_{\mathbf{x}_i}[\mathcal{L}_l; \sigma^2\mathbf{I}] \quad (7)$$

The data likelihood is a categorical distribution (one sample from a multinomial) over the library sites so that

$$Pr(\mathbf{x}_i|\theta_\bullet) = Pr(\hat{l}_i|\theta_\bullet) = \theta_{\hat{l}_i}. \quad (8)$$

Now consider the entire training data $\mathbf{x}_\bullet$. The likelihood now takes the form

$$Pr(\mathbf{x}_\bullet|\theta_\bullet) = \prod_{i=1}^{I} Pr(\mathbf{x}_i|\theta_\bullet) = \prod_{i=1}^{I} \theta_{\hat{l}_i} = \prod_{l=1}^{N} (\theta_l)^{f_l} \quad (9)$$

where f_l is defined as

$$f_l = \sum_{i=1}^{I} \delta_{\hat{l}_i=l} \quad (10)$$

and $\delta_{\hat{l}_i=l}$ returns one when the subscripted expression $\hat{l}_i = l$ is true and zero otherwise. In other words, f_l is the total number of times the closest matching patch came from library site l during the training process.

We also need to define the prior over the parameters θ in Equation 6. We choose a Dirichlet prior as it is conjugate to the categorical likelihood so that

$$Pr(\theta_\bullet) = \frac{\Gamma(\sum_l \alpha_l)}{\prod_l \Gamma(\alpha_l)} \prod_{l=1}^{N} (\theta_l)^{\alpha_l - 1} \quad (11)$$

where Γ denotes a Gamma distribution and $\{\alpha_1..\alpha_N\}$ are the parameters of this Dirichlet distribution. These are learned from a validation set.

Substituting the likelihood (Equation 9) and the conjugate prior term (Equation 11) into Bayes' rule (Equation 6) we get an expression for the posterior distribution over parameters which has the form of a Dirichlet distribution:

$$Pr(\theta_\bullet|\mathbf{x}_\bullet) = \frac{\Gamma(\sum_l (\alpha_l + f_l))}{\prod_l \Gamma(\alpha_l + f_l)} \prod_{l=1}^{N} (\theta_l)^{f_l + \alpha_l - 1} \quad (12)$$

We compute one of these distributions for each of the P patches in the regular grid and for each of the K classes.

2.3. Calculation of Likelihood Integral

Finally, we substitute the posterior distribution over the parameters $Pr(\theta_{cp\bullet}|\mathbf{x}_{\bullet cp})$ (now resuming use of the indices c and p) into Equation 2 and integrate over $\theta_{cp\bullet}$ to get an expression for the likelihood[1] of observing test data patch $\mathbf{y}_p$ given that the object class is c :

$$Pr(\mathbf{y}_p|\mathcal{C}{=}c, \mathbf{x}_{\bullet cp}) = \frac{f_{cpl^*} + \alpha_{l^*}}{\sum_l (f_{cpl} + \alpha_l)} \quad (13)$$

[1] See {http://pvl.cs.ucl.ac.uk/j.aghajanian} for derivation details.

3. Experiments

3.1. Databases

Faces: We harvested a large database[2] of images of men and women from the web. These were captured in uncontrolled environments and exhibit wide variation in illumination, scale, expression and pose as well as partial occlusion and background clutter (see figure 1a). Faces were detected using two methods: first, we used a commercial frontal face detector. Second, we manually labelled two landmarks. The former method does not localize the faces accurately, but misses many of the harder non-frontal faces (it detected about 70% of the faces). The latter method localizes the images very accurately but includes all examples in the database regardless of their pose or quality.

For both methods the images were subsequently transformed to a 60x60 template using a Euclidean warp. We band-pass filtered the images and weighted the pixels using a Gaussian function centered on the image. Each image was normalized to have zero mean and unit standard deviation.

Cells: The Baum lab RNAi cell phenotype database [2] contains images of human cancer cells (HeLa-Kyoto) displaying a large variety of morphological phenotypes after the individual knockdown of approximately 500 genes. Of these many morphological changes, we were interested specifically in two phenotypes: (i) when the borders of the cell change significantly in response to the knockdown to produce a 'triangular' phenotype with sharply-edged borders, and (ii) when knocking down a gene had no effect on the cell, leaving its phenotypic appearance as non-triangular/amorphous ('normal').

Each image contains 3 color channels: W1,W2,W3, each of which represents a different fluorescent stain. We use the W1 channel to find the nuclei: we threshold the image and then use morphological opening to remove noise. We find connected regions and take their centroids to represent the nucleus position. We place a 60×60 pixel bounding box around the center and extract the data from the W2 channel for classification. Since cells exhibit radial symmetry we convert the images to a 60×30 polar representation, so that the horizontal coordinate of the new image contains the angle from the nucleus center and the vertical coordinate represents the distance from the nucleus. This allows us to easily constrain patches from the library to only match to patches at similar radii without regard for their polar angle. These radial images were band-pass filtered and normalized to have zero mean and unit standard deviation.

Pedestrians: We collected a large database of urban scenes. Pedestrians were automatically detected using the method of [8]. The images were then manually labeled for pose:

[2] The database can be made available upon request. Please email {j.aghajanian@cs.ucl.ac.uk}.

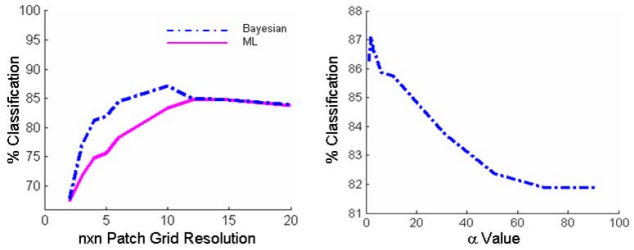

Figure 4. (a) To find the best patch size, gender classification is carried out on the validation set on several patch grid sizes. The performance peaks at the 10x10 grid resolution. (b) The α parameter of the Dirichlet distribution is chosen empirically by testing the performance of the algorithm on the validation. The performance peaks at the value of $\alpha = 2$.

pedestrians facing front, back, left and right. The bounding box around each pedestrian was re-scaled to create a 60x120 image. The pedestrian images were band-pass filtered and normalized as with the face and cell images.

3.2. Experimental Settings

In this section we use a validation set to investigate the effect of patch grid resolution and the Dirichlet parameters $\{\alpha_1..\alpha_N\}$ for gender classification in the face images. We used a training set of 8000 male and 8000 female images, a validation set of 400 male and 400 female images, and a library of 240 images uniformly sampled from both classes.

Figure 4a shows the percentage correct classification as a function of the patch grid resolution. The results show that performance increases as the patch grid gets finer, peaking at a 10×10 grid (6×6 pixels) and then declining. When the patches are very small, they are probably not sufficiently informative. For comparison, we also plot results from a maximum likelihood approach where we form a point estimate of the parameters θ_{cpl}. We note that this approach produces noticeably worse results.

In figure 5 we verify that 6×6 pixel patches are sufficient by reconstructing real images using the closest patches l^* from the library. It is still easy to identify the characteristics of the images using the approximated versions.

Figure 4b shows the percentage correct classification using 6×6 pixel patches as a function of the Dirichlet parameters $\{\alpha_1..\alpha_N\}$ which are constrained to all be the same value. The results show a significant jump in performance when the α value changes from 1 to 2 but then decline. This is also a confirmation of the Bayesian inference being beneficial, since the maximum likelihood solution can be seen as a special case of Bayesian inference when $\alpha = 1$. For the rest of the paper we adopt these optimal parameters: we use a patch resolution of 6×6 and set $\{\alpha_1..\alpha_N\} = 2$.

In Equation 7 we defined a hard (MAP) assignment of training patch $\mathbf{x}_i$ to library index $\hat{l}_i$. In principle it would be better to marginalize over possible values of $\hat{l}_i$, but this

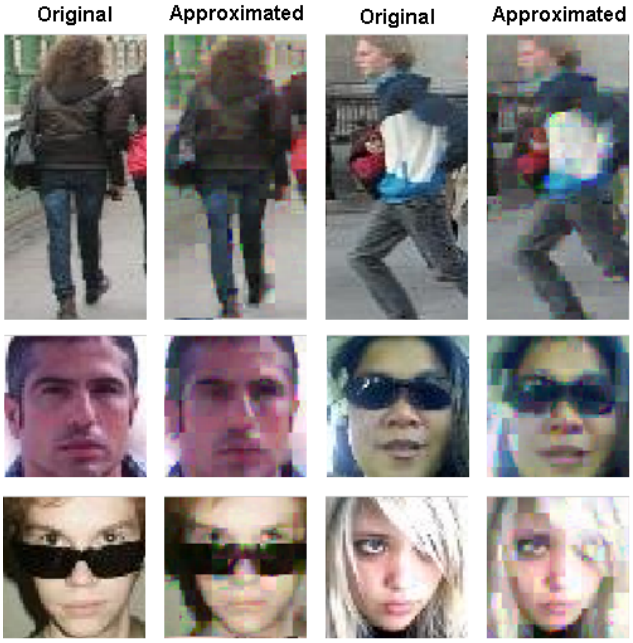

Figure 5. Comparison of original images and best approximations l^* from library patches.

is intractable. We found experimentally that it was possible to slightly improve performance by using a soft assignment of library sites, replacing Equation 10 with

$$f_l = \sum_i \frac{\mathcal{G}_{\mathbf{x}_i}[\mathcal{L}_l; \sigma^2\mathbf{I}]}{\sum_{j=1}^{N} \mathcal{G}_{\mathbf{x}_i}[\mathcal{L}_j; \sigma^2\mathbf{I}]}, \tag{14}$$

and we have done this for all results in the paper. In practice, we also restrict the possible indices l to a subset corresponding to a 6×6 pixel window around the current test patch position in each library image so patches containing eyes are only approximated by other patches containing eyes etc.

3.3. Gender Classification

In this experiment we investigate gender classification for both the manually and automatically detected face datasets. In each case, we use a training set of 16,000 male and 16,000 female images. The test set contains 500 male and 500 female faces and the library is made up of 120 male and 120 female images.

We achieve an 89% correct recognition rate on the manually detected dataset. Figures 6a and b show correctly classified female and male examples respectively. Note that the images contain large pose variations ranging from -90^o to +90^o. Figure 6c shows typical examples of male images misclassified as female. Notice that these images have no facial hair and some have long hair. The third person is pulling a face which was seen more often in female training examples. Figure 6d shows typical examples of female

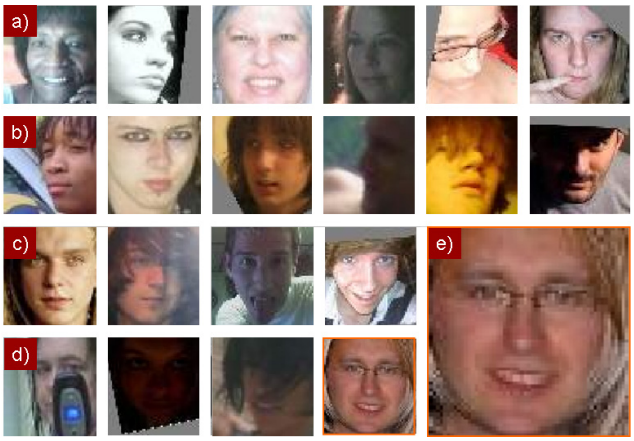

Figure 6. Gender classification performance was 89% on manually detected faces. (a) Correctly classified females. (b) Correctly classified males. (c) Males misclassified as female. (d) Females misclassified as male. (e) Close up: interesting misclassified case.

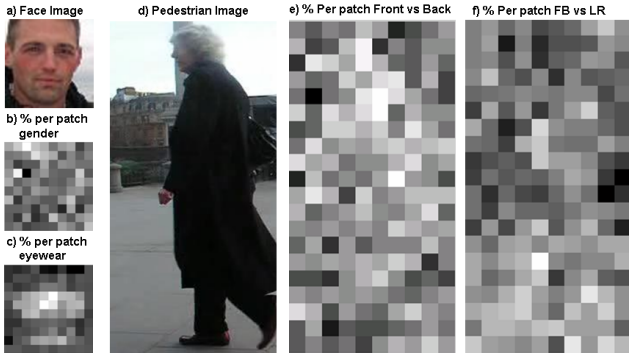

Figure 7. Classification performance was tested per patch. (a) Sample face image. (b) % correct performance per patch for gender (c) % correct performance per patch for eyewear. (d) Sample pedestrian image. (e) % correct performance per patch for classifying pedestrians as facing front vs. back. and (f) % correct performance per patch for classifying pedestrians as either facing front & back vs. facing left & right.

images misclassified as male. Many of these images are partially obscured or low quality. The fourth image (blown up in 6e) is particularly interesting. This was tagged as female but we suspect it is a man in a wig! Gender classification was performed with a similar protocol on automatically detected images. Here, we achieve 90% correct classification. This dataset shows less variation in head orientation but the position of the face varies more in each image. Examples of correctly classified faces were shown in figure 1a.

We also tested the classification ability of each patch individually. Figure 7b shows the percentage correct classification for each patch as a gray level image (the higher the performance, the lighter the patch). Notice that there are no dominant patches with high discriminative power. Instead a collective decision is made for classification of gender.

3.4. Eyewear Classification

We also investigated the task of determining whether people were wearing glasses. The training set for this experiment contains 8,000 images with glasses and 8,000 images without. The library contained 120 images with glasses and 120 without. The algorithm was tested on 400 images with glasses and 400 without. We achieve 84% correct classification. Figure 7c shows the percentage correct achieved based on each patch alone. As expected, there is far more discriminatory power in the top half of the image. We repeated the experiment using only the top half of each image and achieved 91.2% classification.

Figure 8a shows images correctly classified as without glasses despite some images being dark (1,2) or the eye being covered by hair (3-5). Figure 8b shows images correctly classified as wearing glasses, despite the images being very bright (4), blurry (5,6), or non-frontal (3). Misclassified images are shown in figures 8c and d. Many of the images misclassified as wearing glasses (figure 8c) have obscured eyes or were wearing a cap. Most of the images misclassified as being without glasses (figure 8d) were wearing frameless reading glasses which are difficult to distinguish.

We compared our results with the performance of 10 human subjects on the same test set. Their average performance was 96.79% correct classification. We noted that most of the images misclassified by humans as without glasses were also wearing frameless reading glasses.

3.5. Cell Phenotype Classification

In this section, we apply our algorithm to a second object class with completely different properties. We classify human cells from a subset (141 images) of the Baum lab RNAi cell phenotype database [2] as being either 'triangular' or 'normal'. For training 12500 cells from 125 images were used for each class. The library contained 120 cells from each class.

The first experiment tests the ability of our algorithm to classify single cells. We used a test set of 500 normal and 500 triangular cells. The method achieved 70% correct classification. We note that (i) this is a very difficult task as the cell images contain significant within class variation (see figure 9) (ii) not all cells in a given image are affected by the experimental conditions that cause changes in cell shape so we do not expect perfect performance and (iii) biologists are usually interested in classifying entire images (images) each of which contains 50-150 cells.

This motivates our second experiment in which we classify the entire images. Due to limited amount of data there were only 16 images (8 from each class) that were not used either in training or library. We break these images into 4 parts, resulting in 64 subimages which were used in testing. We treat each cell within each subimage as providing independent information about the image class. Under these

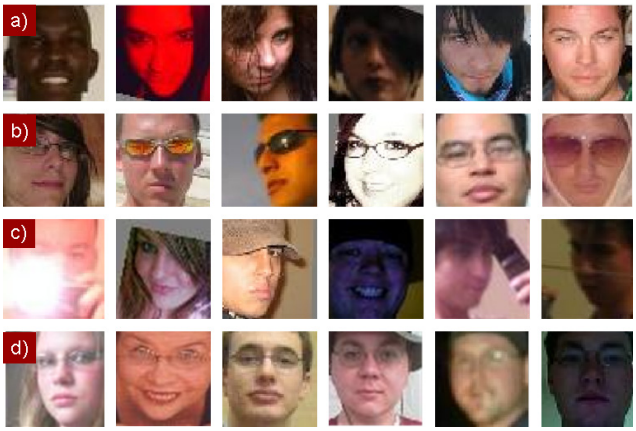

Figure 8. We achieved 91.2% correct classification of the presence of eyewear. (a) Correctly classified as without glasses. (b) Correctly classified as wearing glasses. (c) Without glasses but misclassified as wearing glasses. (d) Faces with glasses but misclassified as not wearing glasses.

conditions, the algorithm managed to achieve 100% correct classification rate by classifying all 64 subimages correctly. Example classifications for cells are shown in figure 9a-b. Subimage classification is shown in Figures 1b and 9c.

3.6. Pedestrian Pose Classification

Finally, we test our algorithm on classifying pose in pedestrian images. In training we use 3000 images of pedestrians from each of the four classes (i) facing front, (ii) facing back, (iii) facing left and (iv) facing right. We use a library of 240 images (60 from each class). We devise four separate experiments.

In the first experiment we do multi-class classification using a test set of 1200 images (300 per class). In this experiment we classify a test image as belonging to one of the four classes. We achieve 67% correct classification overall. Table 1 shows the confusion matrix where each row shows the true label and each column shows the estimated label. It is notable that left facing pedestrians are most confused with right facing ones and front facing pedestrians are most confused with back facing ones. Examples of correct and wrong classifications are shown in Figure 10.

In the second experiment we examine binary classification to distinguish only front-facing from back-facing examples. We tested on 600 images (300 back, 300 front) and we achieve 75% correct classification. This is quite a challenging task as these two classes are largely distinguishable only from the facial area. This is verified when we examine the per patch classification (see figure 7e): patches in the top center of the image are most informative.

In the third experiment we classify test images as either facing left or facing right. We achieve 81.2% correct classification. Finally we test our algorithm on classifying pedestrians as either facing left/right, or facing front/back.

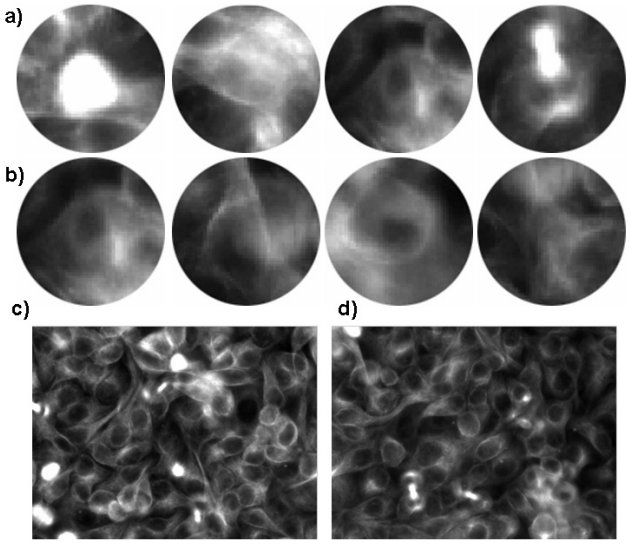

Figure 9. W2 channels of individual cells that were correctly classified as (a) normal and (b) triangular. (c) A correctly classified normal image. (d) A correctly classified triangular image. Interestingly, biologists usually classify cells based primarily on the W3 channel. Our algorithm seems to be exploiting information that is not particularly salient to human experts.

True \ Est.	Back	Front	Left	Right
Back	**77.7**%	10.0%	5.6%	6.7%
Front	35.6%	**53.7**%	6.7%	4.0%
Left	7.3%	6.7%	**71.0**%	15.0%
Right	12.0%	8.3%	15.0%	**64.7**%

Table 1. Confusion matrix for pedestrian pose classification.

In this experiment we achieve 85.3% correct classification. We plotted the per patch classification as a grayscale image in 7f. Unsurprisingly this figure shows that the most discriminative patches for this task are ones towards the bottom of the image: the legs are the most distinctive part of the image to distinguish these classes.

3.7. Comparison to Other Algorithms

For further validation we compare the performance of our gender classification algorithm with the manually registered dataset to that of support vector machines (SVMs). Unfortunately, SVMs were not designed to work with large databases and it is hard to train with the high resolution (60×60) images due to the memory requirements. To get the best out of these methods we have used both (i) the maximum feasible number of training images at high resolution (4000 images per class) and (ii) a larger training set (16000 images per class) of low resolution images which were subsampled to 21×12. This is similar to images used in [16].

For the first case (4000 high resolution images per class) a linear SVM and a non-linear SVM with an RBF kernel

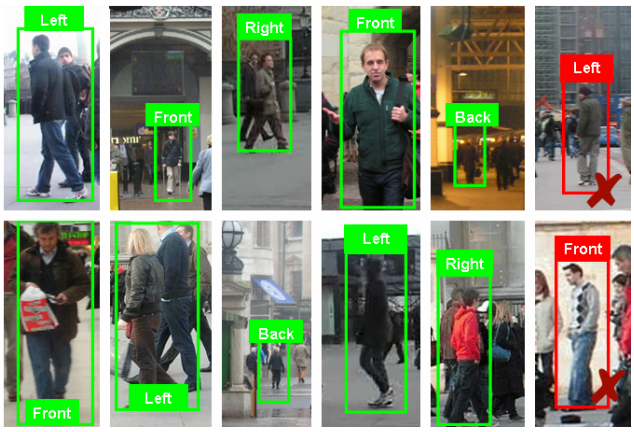

Figure 10. Example results from the pedestrian pose classification experiment. We show the predicted label for each pedestrian. The two images marked by a red cross have been misclassified, but the remaining images show correctly classified pedestrians.

achieved 78.8% and 77.8% performance respectively. The SVMs were trained with libsvm and the parameters selected with 3-fold cross validation. When we tested our method with only these 4000 training images we achieved 84.6% which is considerably better than either SVM method.

For the second case (16000 low resolution images per class) the linear SVM achieved 78.7% performance and the non-linear SVM achieved 82.4%. For this dataset we also tried linear discriminant analysis which achieved a maximum of 78%. None of these results approach the 89% performance achieved by our algorithm.

Finally, we also compared human performance on gender classification. For this purpose 10 subjects were shown the same test images as used for our gender classification experiment. The images were 60×60 in size and grayscale but did not undergo further preprocessing. The average human performance was 95.6%. Although our best performance is 7% lower than this, we conclude that some of the test images are genuinely difficult to classify.

4. Summary and Discussion

In this paper we have proposed a general Bayesian framework for classifying within-object characteristics. Our algorithm uses a generic patch-based representation therefore it can be used on several object classes without major alterations. We demonstrate good performance on 'real world' images of faces, human cells and pedestrians.

The algorithm has a close relationship with non-parametric synthesis algorithms such as image quilting [7] where patches from one image are used to model others. Our algorithm works on exactly the same principles - all the knowledge about the object class is embedded in the library images. This accounts for why the algorithm works so well in different circumstances. If we have enough library images they naturally provide enough information to discriminate the classes. The algorithm also has a close relationship with bag of words models [5] . The library can be thought of as a structured set of textons which are used to quantize the image patches.

In terms of scalability our algorithm is linear with respect to the size of the library and the training data. For a library of size m and a training set of size n it scales as $O(mn)$ during training and $O(m)$ in testing. The Bayesian formulation where we marginalize over the parameters guards against overfitting.

In future work we intend to investigate other visual tasks such as regression on continuous characteristics (e.g. age), localization and segmentation using similar methods that exploit a library of patches and a large database of images.

References

[1] S. Baluja and H. Rowley, "Boosting Sex Identification Performance," *IJCV*, Vol. 71, pp. 711-119, 2007.
[2] J. Rohn and B. Baum, unpublished data.
[3] R. Brunelli and T. Poggio, "HyperBF Networks for Gender Classification," *Image Understanding*, pp. 311-314, 1992.
[4] A. Carpenter, T. Jones, M. Lamprecht, C. Clarke, I. Kang, O. Friman, D. Guertin, J. Chang, R. Lindquist, J. Moffat, et al, "CellProfiler: image analysis software for identifying andquantifying cell phenotypes," *Genome Biology*, Vol. 7, pp. R100, 2006.
[5] G. Csurka, C.R. Dance, L. Fan, J. Willamowski and C. Bray, "Visual categorization with bags of keypoints," *Workshop on Statistical Learning in Computer Vision, ECCV*, pp. 1-22, 2004.
[6] A. Efros, A. Berg, G. Mori and J. Malik, "Recognizing action at a distance," *ICCV*, pp. 726-733, 2003.
[7] A.A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," *Proc. SIGGRAPH*, pp. 341-346, 2000.
[8] P. Felzenszwalb, D. McAllester and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," *CVPR*, pp. 1-8, 2008.
[9] B. Golomb, D. Lawrence and T. Sejnowski, "Sexnet: A neural network identifies sex from human faces," *NIPS*, Vol. 3, pp. 572-577, 1991.
[10] D. Hoiem, A. Efros and M. Hebert, "Putting Objects in Perspective," *CVPR*, Vol. 2, pp. 3-15, 2006.
[11] T. Jones, A. Carpenter, D. Sabatini and P. Golland, "Methods for high-content, high-throughput image-based cell screening," *MIAAB Workshop on Microscopic Image Analysis*, pp. 65-72, 2006.
[12] N. Kumar, P. Belhumeur and S. Nayar, "FaceTracer: A Search Engine for Large Collections of Images with Faces," *ECCV*, pp. 340-353, 2008.
[13] M. Leung and Y. Yang, "First Sight: A Human Body Outline Labeling System," *PAMI*, pp. 359-377, 1995.
[14] S. Li and Z. Zhang, "Floatboost learning and statistical face detection," *PAMI*, Vol. 26, pp. 1112-1123, 2004.
[15] E. Mäkinen and R. Raisamo, "An experimental comparison of gender classification methods," *Pattern Recognition Letters*, Vol. 29, pp. 1544-1556, 2008.
[16] B. Moghaddam and M. Yang, "Learning Gender with Support Faces," *PAMI*, pp. 707–711, 2002.
[17] Y. Saatci and C. Town, "Cascaded Classification of Gender and Facial Expression using Active Appearance Models," *AFGR*, Vol. 80, pp. 393-400, 2006.
[18] G. Shakhnarovich, P. Viola and B. Moghaddam,"A unified learning framework for real time face detection andclassification," *AFGR*, pp. 14-21, 2002.
[19] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *CVPR*,Vol 1, pp. 511-518, 2001.