An overview of methods for image classification

Raymond Ahn Daradipo Bou Augusto Partida Justin Sunu

May 18, 2012

Abstract

An ongoing research in modern mathematics to find the most optimal way to classify images. We focus on 5 classic methods of classifying images: Wavelet Analysis, Principal Angles, Kohonen's Novelty Filter, Linear Discriminant Analysis, and Kernel Linear Discriminant Analysis. Since these methods have been thoroughly studied, we focused on identifying best bases for a training set in order to classify images of cats and dogs. Our research shows that the quality of the training set is more important than the quantity of images in the training set. Furthermore, despite there being more complicated methods, sometimes the simplest methods yield the best results.

1 Introduction

In order to understand how computers can be used in image processing, we must first understand how our minds work. According to psychologists, the process in which we humans process information can be described with the following diagram:

Figure 1: Diagram of information processing for humans [3]

From this figure, we see that an image we look at (stimulus) goes to the input processing stage, then it gets compared to images we already have stored (memory), and finally when a comparison is made, the results are given as a response. Computer based image processing follows the same pattern. First we input the images we wish to identify (test images), we process this information, compare it against what is stored (training set), and finally output a result.

Furthermore, we have the ability to identify images without all the detail. As long as the minimum amount of necessary information is available, we can make guesses as to what a certain image is. Computer based image processing also follows a similar idea. Since it is too time consuming to use an entire image, it always wise to reduce the dimension and size of the image. By reducing the dimension to where only important information is left, we can still use algorithms to make comparisons. These ideas are the basis for our image processing methods. Wavelet analysis allows us to extract important features from an image, Singular Value Decomposition (used in all the other methods) allow us to reduce the dimension of the image.

2 DATA

Our data set consists of 80 cats and 80 dogs provided by Dr. Chang. As for our testing set, it consists of 38 images.

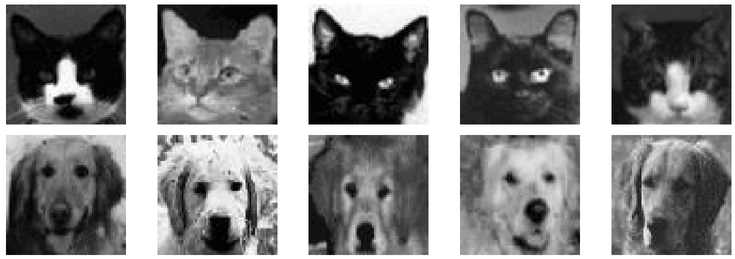

Figure 2: Images of cats and dogs

By looking at the images, we can see that there are key differences between cats and dogs. For example, most of the cats have pointy ears and rounder faces, while dogs have a more oval face and their ears drop to the sides. Furthermore, we can see that several dogs share the same pointy ear feature as a cat, so we expect that any incorrectly identified dogs will be those. Our goal is to identify these differences through wavelet analysis (section 3.1) and singular value decomposition (section 2.1). The results can be seen in Section 4.

2.1 Singular Value Decomposition

Since SVD is an integral part of all our methods, we feel it is important to study the eigen-images produced as a result of using SVD on the data set. We refer the reader to [2] for more information about SVD.

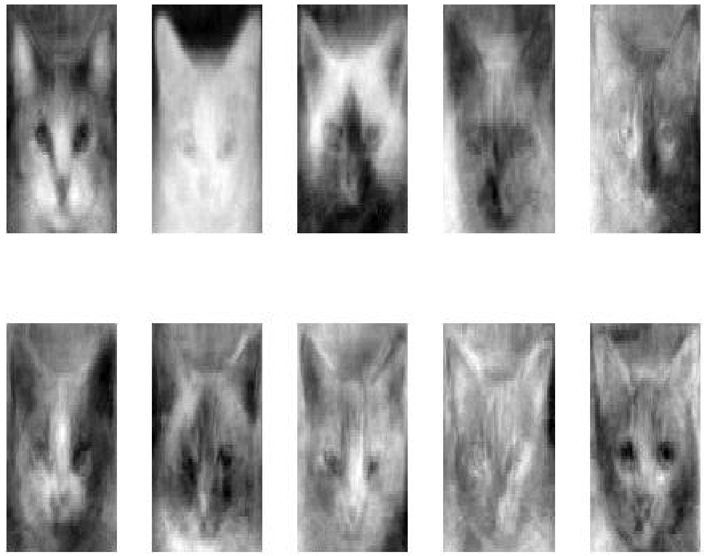

Figure 3: The first 10 eigencats

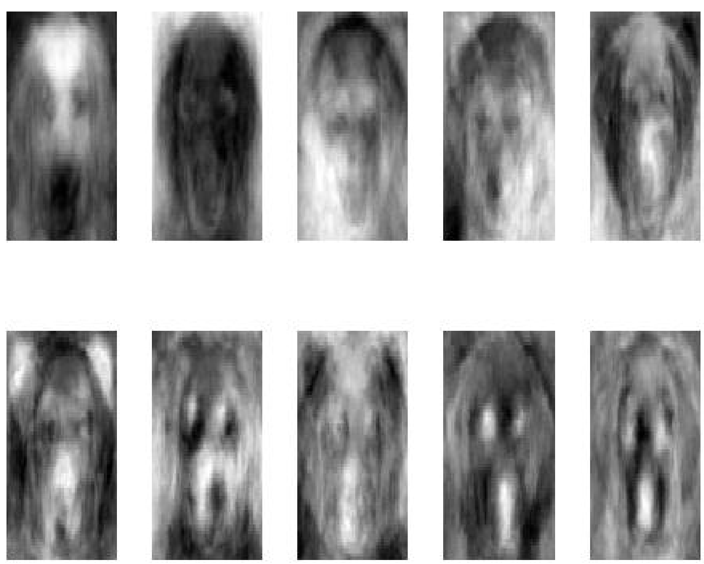

Figure 4: The first 10 eigendogs

By looking at Figure 3, we see that the first eigenvector captures what the average cat should look like. The second eigenvector outlines the basic shape of the cat and the third focuses on the upper half (ears) area of the cat's face. The rest of the eigenvectors capture various features such as texture, markings, and other subtle details of a cat. As for the dogs (Fig. 4), we see, similar to the eigencats, that the first two eigenvectors capture what the average dog should look like and the basic outline of the dog's face, respectively. Unfortunately, since the images of dogs vary more than that of the cats, the rest of the eigenvectors do not provide as much information.

3 Methods

3.1 Wavelet Analysis

Wavelet analysis allows us to extract features from an image in the form of "wavelets". For images, we require a 2-D scaling function $\phi(x,y)$, and three 2-D wavelets, $\psi^H(x,y)$, $\psi^V(x,y)$, and $\psi^D(x,y)$. These wavelets measure functional variations (intensity variations for images) along different dimensions: ψ^H measures along the columns, ψ^V measures along the rows, and ψ^D measures along the diagonals [2].

In order to get these functions, we define a scaled and translated basis function as:

$$\begin{aligned} \phi^j_{m,n}(x,y) &= 2^{-j/2}\phi(2^{-j}x - m, 2^{-j}y - n) \\ (\psi^j_{m,n}(x,y))^i &= 2^{-j/2}\psi(2^{-j}x - m, 2^{-j}y - n). \end{aligned}$$

Then the discrete wavelet transform of an image $f(x,y)$ of size $M \times N$ is then:

$$\begin{aligned} c^j_{m,n} &= \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\phi^j_{m,n}(x,y) \\ (d^j_{m,n})^i &= \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)(\psi^j_{m,n}(x,y))^i, \end{aligned}$$

where $i = \{H, V, D\}$.

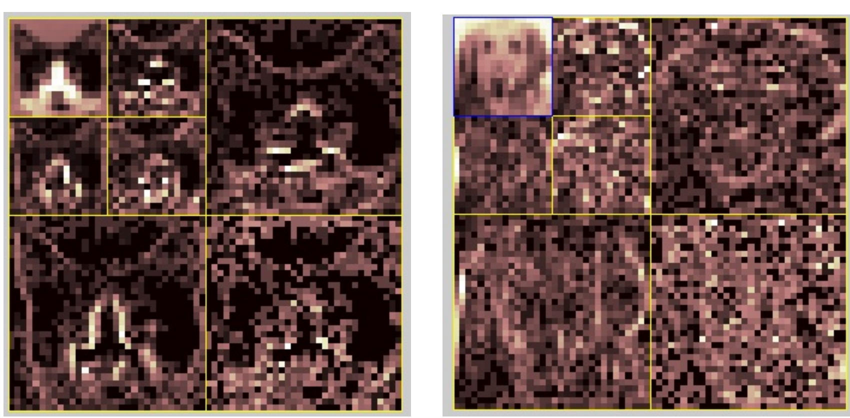

Figure 5: 2 level decomposition of a cat (left) and dog (right).

As with the SVD case 2.1, we see that the horizontal and vertical details of the first level of decomposition of the cat highlight the edges better than the dogs. Furthermore, the second level highlights some of the eyes, nose, and snout area of the cat. We stopped at the second level because the images created when doing the third level are 8×8 and they only carry color and background information, which are not important in classifying images.

Since wavelet decomposition is only a means of generating detail images, analysis of these images are done with the other methods (Sec. 3.2, 3.3, 3.4).

3.2 Principal Angles

In $\mathbb{R}^2$, we can use the angle between two vectors to determine how similar they are with the following equation:

$$\cos(\theta) = \frac{u \cdot v}{\|u\| \|v\|}.$$

As we go up in dimensions, we start to work with vector spaces, which consists of multiple vectors. To handle this, we use Principal Angles. Principal angles applies the $\cos(\theta)$ formula for all combinations of vectors among the two vector spaces. The smaller the θ the more "similar". Like all methods, we start by reducing the dimension by using SVD to find the optimal KL basis. We use the following algorithm to compute the principal angles.

Algorithm 3.1. Principal Angles
M is the training sets, and N is the test set.

- Perform *SVD* on the training data, $M = U_M \Sigma_M V_M^T$.
- *Project* the test set onto the training set, $U_M N$.
- Compute *principal angles* between the training sets and projected test set.
- The training set is $\Sigma_M V_M^T$.
- The projected test set is $U_M N$.

3.3 Kohonen's Novelty Filter

Kohonen's novelty filter is about using the residual to compare two images. Similar to principal angles, we use SVD to find the optimal basis which is then used as the projection matrix $\mathbb{P}$. For this method, we take the residual to be:

$$(\mathbb{P} - \mathbb{I})X,$$

where $\mathbb{P}$ is the projection matrix, $\mathbb{I}$ is the identity matrix and X is the data set. By comparing the residual with the test image and training images of cats and dogs, we take the minimum residual, a simple two norm, to be the way the test image is labeled. If the residual between test and cats is smaller than test and dogs, then the image is a cat.

3.4 Linear Discriminant Analysis

Linear discriminant analysis (LDA) is a way to find a linear combination of features which characterize or separate two or more classes of objects or events [2]. Essentially, what LDA does is try to find a line that separates two clusters. In order to do this, we want to project the data set onto the real line, where, hopefully, a clear separation of data can be seen.

In order to use LDA, we must first define several concepts. First the within-class and between-class scatter matrices:

$$S_W = \sum_{i=1,2} \sum_{x \in \mathscr{D}_i} (x - m_i)(x - m_i)^T S_B = (m_2 - m_1)(m_2 - m_1)^T,$$

where m_1 and m_2 are the class-wise means. Since we only have two classes (cats and dogs), the two dimensional case is sufficient.
With these matrices, our goal is now to solve the generalized eigenvalue problem:

$$S_B w = \lambda S_W w.$$

We take the eigenvector that corresponds to the largest eigenvalue. Furthermore, we define the separation point as:

$$\alpha = \frac{\tilde{m}_1 + \tilde{m}_2}{2}.$$

By projecting the data set onto w, we can see that points to the left of α are cats and to the right are dogs.

3.5 Kernel Linear Discriminant Analysis

Kernel linear discriminant analysis (KLDA) generalizes LDA since in the transformed space, the principal components are nonlinearly related to the input variables. KLDA maps the input space into a high dimensional, nonlinear feature space. This transformation is carried out by a kernel function $\phi : X \to F$. We used a common kernel function known as the RBF (Gaussian) kernel such that:

$$\phi(\mathbf{x}, \mathbf{y}) = exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right)$$

where the assumption is that the classes have a multivariate Gaussian distribution. ([1],[4],[5]. Figure 6 compares the feature mapping using a linear kernel (i.e. such as that used in LDA) on the left and a nonlinear kernel on the right with the differing optimum projection vectors as a result of this mapping. In particular, the kernel mapping seeks a linear transformation of the original data in a higher dimension.

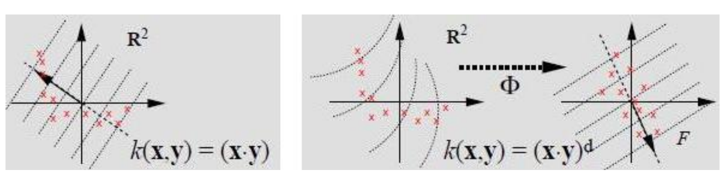

Figure 6: Feature space transformation.

After mapping data to the feature space, then the same procedure as LDA is used to find the optimal direction vector that separates two classes:

- Define and compute the Rayleigh quotient: $J(\mathbf{w}) = \frac{\mathbf{w}^T S_B^\phi \mathbf{w}}{\mathbf{w}^T S_W^\phi \mathbf{w}}$.
- Solve the generalized eigenvalue problem: $S_B^\phi \mathbf{w} = J(\mathbf{w}) S_W^\phi \mathbf{w}$.
- Project a new pattern onto the optimal direction: $\mathbf{w}^T \phi(\mathbf{x})$.

4 Results

A typical approach to measuring the success of an image processing algorithm is to see how well it performed. Although that is the overall goal of our project, we wish to see what the optimal number of training images needs to be. If we can get the same accuracy by using less images in the training set, then the speed of the algorithm improves. Although are images are very small, 64×64 pixels, most images these days significantly larger, which makes not only accuracy important but efficiency as well.

For our project we ran cross-validation with a different amount of images. We used sets of 40, 50, 60, and 70 images are our training and ran 11 iterations to ensure all the images were included in the training set at least once. For each of the methods listed, we used the original images, the first wavelet approximation, horizontal and vertical details, and the second wavelet approximation, horizontal and vertical details. And finally, since all our methods involve SVD, we used 99% cumulative energy.

Method	40	50	60	70
Principal Angles	**.8684**	.8421	**.8684**	**.8684**
Novelty Filter	**.9211**	**.9211**	**.9211**	.8947
LDA	.8158	**.8421**	**.8421**	**.8421**

Table 1: Best results from using original data.

Method	40	50	60	70
Principal Angles	.8421	**.8684**	**.8684**	.8421
Novelty Filter	.8947	**.9211**	.8947	.8947
LDA	.8421	**.8684**	.8421	.8421

Table 2: Best results from using first wavelet approximation.

Method	40	50	60	70
Principal Angles	..8947	.8947	.8947	**.9211**
Novelty Filter	**.9211**	**.9211**	**.9211**	**.9211**
LDA	.7632	**.8158**	.7632	.7632

Table 3: Best results from using first wavelet horizontal detail.

Method	40	50	60	70
Principal Angles	**.9211**	**.9211**	.8947	.8947
Novelty Filter	**.9474**	.9211	.9211	.8947
LDA	.7105	.6579	.6579	**.7368**

Table 4: Best results from using first wavelet vertical detail.

Method	40	50	60	70
Principal Angles	.8421	**.8684**	**.8684**	**.8684**
Novelty Filter	**.9211**	**.9211**	**.9211**	.8947
LDA	**.7105**	.6842	.6579	.6053

Table 5: Best results from using second wavelet approximation.

Method	40	50	60	70
Principal Angles	**.8947**	.8684	.8684	.8684
Novelty Filter	**.8947**	**.8947**	**.8947**	**.8947**
LDA	.7895	**.8684**	.8158	.7895

Table 6: Best results from using second wavelet horizontal detail.

Method	40	50	60	70
Principal Angles	.8421	.8158	.8421	**.8684**
Novelty Filter	**.8421**	.8158	**.8421**	.8158
LDA	.5789	.6579	.6579	**.7368**

Table 7: Best results from using second wavelet vertical detail.

As for the KLDA method, we used a slightly different approach. Since the parameter σ is unknown, then it must be estimated from the training data. We performed leave-one-out cross-validation (LOOCV) on the training set to find best parameter $\boldsymbol{\sigma}$ as well as the energy for dimensionality reduction using principal components analysis (PCA).
Furthermore, we only ran this method on the original images (i.e. not filtered). Figure 7 shows the separation of classes given the KLDA method. As shown, the method completely separates the training data. Also, the projected validation data shows that dogs may have more varying features since the majority of test cat images are pulled near the decision boundary.

Data Set	$\boldsymbol{\sigma}$	e	Accuracy
Training	6.05	0.75	0.89375 (143/160)
Validation	5.45	0.75	0.92105 (35/38)

Table 8: KLDA classification performance.

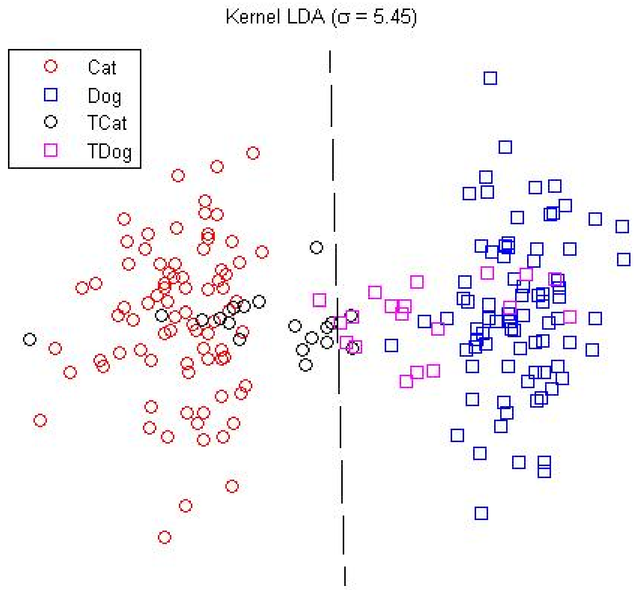

Figure 7: Separation using KLDA

From these results, which are the best results from each iteration, we can see that Kohonen's novelty filter did the best overall. It also gave the best result of 0.9474 (Table 4), or two image out of 38 incorrectly identified, when using the first level wavelet decomposition's vertical detail. As we predicted, the dogs with pointy ears were incorrectly identified as cats, although all the cats were correctly identified. Another interesting trend we see in the results is that using more images typically lowers our accuracy. Our reasoning behind this is that since we are using 99%, the amount of information we are using is too much. By looking at the first few eigenimages (Sec 2.1), we know that only the first few contain relevant information, with the rest mainly being color, texture, pattern, etc. Since cats and dogs do have things in common, using more of the eigenimages is blurring the distinction between them.

Although we can reduce the cumulative energy percentage to 95%, it is also possible that 95% might not capture enough information. With this, we conclude that there is an optimal cumulative energy as well as an optimal number of training images that will yield the best results. Furthermore, we see that by not including dogs with pointy ears in the training set, our accuracy drops significantly since many of the test dogs share that characteristic.

5 Conclusion

As mentioned above, we realized that there are several requirements for an "optimal" training set: quality of the images, number of images, and that cumulative energy percentage is an important factor as well. Since our images are very small and wavelet analysis only provides a few details (Sec. 3.1), if our images were "hi-def", perhaps wavelet analysis would yield better details about the fur type, snout information, distinct eye styles, and other information that these images could not provide. Also since hi-def images are significantly larger, our eigenimages should provide more information for lower order eigenvectors.

Although we were unable to test various cumulative energy percentages and we only ran each iteration once, we feel that by averaging results over various percentages and with more testing, we can improve upon our results and find that best training basis. We would also like to compare our results to some of the more modern techniques and see how they compare in efficiency and accuracy.

References

[1] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Comput.*, October 2000.

[2] Jen-Mei Chang. Matrix methods for geometric data analysis and pattern recognition, 2011.

[3] Saul McLeod. Simple psychology, information processing, 2008.

[4] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Muller. Fisher discriminant analysis with kernels, 1999.

[5] B. Scholkopf, A. Smola, and K. Muller. Kernel principal component analysis, 1999.