Biologically Inspired Object Recognition using Gabor Filters

William Hamilton

April 25, 2013

1 Introduction

Recent advancements in the understanding of the computational processes underlying early vision have provided novel opportunities for the creation of biologically inspired vision models. Hubel and Wiesel (1968) empirically demonstrated the existence of receptive fields as a fundamental aspect of early visual processing in mammalian vision systems. Further work demonstrated that these receptive fields can be modelled as the application of linear Gabor filters on visual input data (Jones and Palmer, 1987). These advancements, and others, have led to the creation of biologically inspired feature extraction and selection techniques that emulate the early stages of mammalian visual processing (Fei-Fei et al., 2004; Jarrett et al., 2009; Pinto et al., 2008; Serre et al., 2005).

In addition to these advancements in the understanding of early visual processing, there have also been a number of advancements in biologically realistic classification (LeCun et al., 2010; Woodbeck et al., 2008). There are now a large number of classifiers, the majority of which are based upon neural network architectures, that attempt to perform classification and categorization in a biologically plausible manner.

In this section, I will describe the importance of biologically inspired object recognition models. I will also provide an overview of the previous work on biologically plausible object recognition and will outline some interesting questions that remain open. Lastly, I will introduce the approach I take to address some of these open questions.

1.1 Importance of Biologically Inspired Object Recognition

Biologically realistic models are important for many reasons. First, there is the fact that biological vision systems are still far superior to the current state-of-the-art computer vision systems. Humans have the ability to learn object categories using only small, often unlabelled, noisy examples. Moreover, in natural settings, the pose and lighting conditions of object examples are far from homogenous. Nevertheless,

humans still manage to learn object categories using only a small number of positive training examples (Fei-Fei et al., 2004). Thus, it is practical to emulate mammalian visual systems for object recognition.

On the other hand, computer vision models that emulate mammalian visual systems also provide novel insights into how the brain processes visual input. Often there are competing models attempting to describe the computational processes underlying mammalian vision. Implementing these models in object recognition systems allows an empirical evaluation of their plausibility. For instance, a model that performs better in a parallel implementation is more plausible, as the architecture of the brain is highly parallel (Woodbeck et al., 2008).

Lastly, biological plausibility refines performance goals. That is, computer vision systems inspired by biology should be evaluated according to biologically inspired measures. At the very least, biologically plausible recognition systems should be able perform reasonably well given only a small number of positive training examples and should be robust to variance in pose and lighting.

1.2 Previous Work

There are a number of successful biologically inspired recognition systems in the literature. Three general approaches that inspired the model introduced here are (1) filter-banks (2) multi-stage architectures, and (3) neural-network based approaches.

Filter-banks underly the majority of biologically inspired object recognition systems. In this approach, one applies a battery of linear filters to an image and uses the filtered images as primary features (Jarrett et al., 2009; Jones and Palmer, 1987; Pinto et al., 2008; Serre and Riesenhuber, 2004; Serre et al., 2005). Often some sort of post-processing is done on these features, such as non-linear pooling or dimensionality reduction (Pinto et al., 2008; Serre et al., 2005). However, the main aspect of all these models is that they use linear filters (often Gabor filters, which are described below) in order to mimic the feature processing of early mammalian vision.

Multi-stage architectures build upon the filter-bank approach. These recognition systems attempt to emulate the later stages of visual processing, such as the non-linear pooling of inputs from different simple cells into complex cells (Hubel and Wiesel, 1968). These systems often have multiple layers, where the outputs from applying filter-banks are processed and combined from layer to layer (Serre and Riesenhuber, 2004; Serre et al., 2005; Woodbeck et al., 2008). Some of the most successful architectures use max-pooling techniques, where the max output from a set of filters in a specific region of the visual field is propagated (Serre and Riesenhuber, 2004), while others use neural-networks to learn non-linear pooling structures (Woodbeck et al., 2008).

Neural network based techniques are closely related to, and are often involved in, the above approaches. For example, different types of neural networks can be used to perform classification using features generated from a filter bank (the approach taken

in this work). Neural networks can also be used to perform non-linear pooling during some stage of a multi-layer system (Jarrett et al., 2009), or carefully crafted deep-learning networks can be used to perform both feature selection (via convolution) and classification (LeCun et al., 2010; Scherer and Behnke, 2009).

1.3 Some Unaddressed Questions

Despite the successes of the approaches mentioned above, there still remain some interesting and important questions. For instance, many of the previously mentioned models focus solely on biologically realistic feature selection and do not address the importance of biologically realistic classification. This is extremely important, as one would expect biologically realistic classifiers to perform well using biologically realistic features. Moreover, many interesting interactions between biologically realistic classifiers and feature selection are possible. For example, since many neural-network based classifiers perform implicit non-linear operations on their inputs, it is possible that these types of classifiers may perform better when the inputs have not already been pooled in a non-linear manner. Or, at the very least, it is possible that non-linear pooling is simply redundant when used in combination with a neural-net classifier.

A second often unaddressed question is the impact of biologically realistic experimental set-ups. There has been work on recognition systems that are robust to variance in pose and lighting (Serre and Riesenhuber, 2004), and there are also some results on learning from a small number of training examples (Fei-Fei et al., 2004). However, biologically realistic performance tests, specifically the use of a small number of training examples, is rarely the focus of research on biologicaly inspired models. Indeed, many of the most successful object recognition systems require thousands of training examples, something that is unrealistic in natural setting (Fei-Fei et al., 2004).

1.4 Towards Some Answers

The object recognition system presented here is intended to address these questions. The biologically inspired recognition system uses a filter-bank of Gabor filters to obtain features, which are then post-processed using principal component analysis (PCA). An Adaptive Resonance Theory neural network termed Fuzzy ART-MAP, which is specifically designed to emulate the computational processes underlying object recognition in humans, is used for classification (Carpenter et al., 1992).

The system does not perform non-linear pooling during feature extraction. Thus, it is interesting to compare results obtained using the Fuzzy ART-MAP classifier, which performs implicit non-linear pooling, with results obtained using a simple out-of-the-box linear support vector machine (SVM) classifier, which performs only linear transformations on its inputs. I also test the system using a small number of positive training examples in order to assess the performance in a biologically plausible setting.

2 Technical Background

In this section I describe the technical aspects of Gabor filters and Fuzzy ART-MAP networks, the two main components of my object recognition system.

2.1 Gabor Filters

Gabor filters are a powerful type of linear filter that are used for edge detection and texture segmentation. In two-dimensions, Gabor filters are an elliptic Gaussian kernel modulated by a sinusoidal wave (Jones and Palmer, 1987; Movellan, 2002).

Mathematically, a generic elliptic Gaussian kernel centred at the origin can be represented as

$$G(x,y) = K \times exp(-\frac{1}{2}\left(\frac{x^2}{a^2} + \frac{y^2}{b^2}\right)) \tag{1}$$

where the parameters a and b determine the spatial extent in the x and y axis directions respectively and K is a constant determining scale. The orientation of the Gaussian envelope can then be modified by a rotation transformation using an appropriate change of variables. That is,the Gaussian envelope can be rotated by a clockwise angle θ by changing to the new variables $X = xcos\theta + ysin\theta$, $Y = -xsin\theta + ycos\theta$, which transforms equation (1) to

$$G(x,y) = K \times exp(-\frac{1}{2}\left(\frac{X^2}{a^2} + \frac{Y^2}{b^2}\right)) \tag{2}$$

Next, the a and b parameters can be transformed into two new parameters: σ, which controls the global extent of the filter and γ, a spatial aspect-ratio parameter. To do this, the substitutions $\sigma := a$ and $\gamma := \frac{\sigma^2}{b^2}$ are performed. Thus, γ gives control over the relative sizes of the envelope dimensions across the two orthogonal axes, and σ controls the total extent of the envelope. Together, they completely determine the envelope's shape. Finally, the constant K is assumed be the unit value and is dropped for notational simplicity, allowing the the equation for the Gaussian portion of the Gabor filter to be written as:

$$g(x,y) = exp(-\frac{X^2 + Y^2\gamma}{2\sigma^2}) \tag{3}$$

For the sinusoidal part of the Gabor filter only the real portion is used, as this real and observable part is what corresponds to a mammalian receptive field (Jones and Palmer, 1987; Movellan, 2002). For a sinusoidal plane wave $s(x,y)$ this real portion can be written as (Movellan, 2002):

$$Re(s(x,y)) = cos(2\pi(u_0x + v_0y) + \Phi) \tag{4}$$

where (u_0, v_0) specifies the spatial frequency of the wave (in the x and y directions,

respectively) and Φ is a phase shift. However, in the work here, Φ will always be set to 0; that is, unshifted sinusoids are used since they best model mammalian receptive fields (Jones and Palmer, 1987). Thus, (4) can be rewritten as:

$$Re(s(x,y)) = cos(2\pi(u_0 x + v_0 y)) \tag{5}$$

and by combining (u_0, v_0) into a single spatial frequency variable λ and using the same change of variables as above for rotation, it is possible to rewrite (5) as:

$$Re(s(x,y)) = cos\left(\frac{2\pi}{\lambda}X\right) \tag{6}$$

where λ is now a single variable that controls the spatial frequency (Serre and Riesenhuber, 2004).

Finally, combining (3) and (6) through multiplication gives the final form of the 2D Gabor filter that is used here:

$$G(x,y) = exp(-\frac{X^2 + \gamma Y^2}{2\sigma^2}) \times cos(\frac{2\pi X}{\lambda}) \tag{7}$$

where again $X = xcos\theta + ysin\theta$, $Y = -xsin\theta + ycos\theta$.

In this form, four parameters of the Gabor filter determine its orientation, extent, shape, and spatial frequency. Specifically, θ determines the orientation of the filter, σ the extent (i.e. the width of the Guassian), λ the wave-length of the sinusoidal component (i.e. the spatial frequency), and γ the aspect ratio (i.e. the extent in the X-direction relative to the extent in the Y-direction). Figure 1 shows a classic example of Gabor filters with some different parameters.

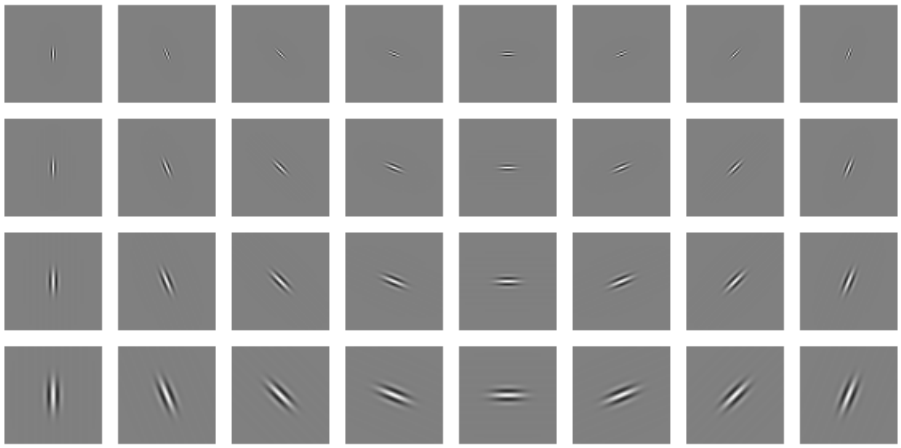

Figure 1: Example Gabor filters with varying θ parameters and σ parameters (Yang, 2010).

Qualitatively, the shape of the Gabor filters resembles the shape of the receptive fields of the simple-cells in early visual processing. Moreover, the edge detecting attributes of the filters are apparent from their spatial structure, as they contain a long elliptical interior that has opposite sign compared to its surrounding boundary. Such a filter will have maximal response when there is a stark contrast in intensity between an interior elongated elliptical section its surrounding boundary, and this is the type of spatial intensity contrast that most often appears at edges in scenes (Movellan, 2002).

Theoretical work has also demonstrated that Gabor filters are solutions to optimization problems that require optimization in both spatial frequency and location (Serre and Riesenhuber, 2004; Daugman et al., 1985). That is, Daugman et al. (1985) show that Gabor filters belong to the 2D filter family which obtains the theoretical lower limit on the combined uncertainty of spatial location and frequency.

In addition to these desirable filter properties, it has been shown that with certain parameter settings Gabor filters are a near-optimal model of the response profile of simple-cell's receptive fields. Specifically, Jones and Palmer (1987) show that Gabor-filters are the best model for the receptive field profiles of simple cells in the cat's striate cortex when compared to other proposed models using a sum-squared error measure. In addition, they show that the residual error is indistinguishable from random noise. Thus, Gabor filters both have powerful edge and texture detection properties, and experimental work has shown that they are optimal models for emulating biological early vision.

2.2 Fuzzy ART-MAP

Adaptive Resonance Theory neural networks are a relatively new neural network architecture designed specifically to mimic the way in which humans and other mammals perform object recognition. The network used in this work was not implemented from scratch; instead a previously implemented version was used (Carpenter et al., 1992). For this reason, I provide only a high level description of the Fuzzy ART-MAP network, focusing on the aspects that are relevant to this work.

At a very high level, Fuzzy ART-MAP systems function using an approach that involves learning prototypical examples of object categories. The full system contains two distinct network components, connected by special controller. These two components are used in tandem to learn stable object categories. One network, ART_a, is fed the feature vectors during training and the other, ART_b, is fed the correct labels. The system then uses fuzzy logic at each connection to learn weights and ART_a produces a "bottom-up" classification that is then compared to a "top-down" prediction for the expected output which is produced by ART_b. If the prediction is correct, then learning via reinforcement occurs; otherwise corrections are made to the fuzzy-logic units in order to prevent further misclassifications (Carpenter et al., 1992).

As with the majority of neural networks, at a low-level the connection weights

between different "neuron-like" nodes are tuned during learning. In this case, fuzzy logic operations are used in the weight tuning. Fuzzy ART-MAP networks also employ majority voting at some stages. Thus, through a combination of weight-tuning and voting protocols, the network is able to learn non-linear pooling structures on the input features (Carpenter et al., 1992). This is a critically important aspect of this classifier (and other similar biologically inspired classifiers) and was one of major criteria considered when selecting it.

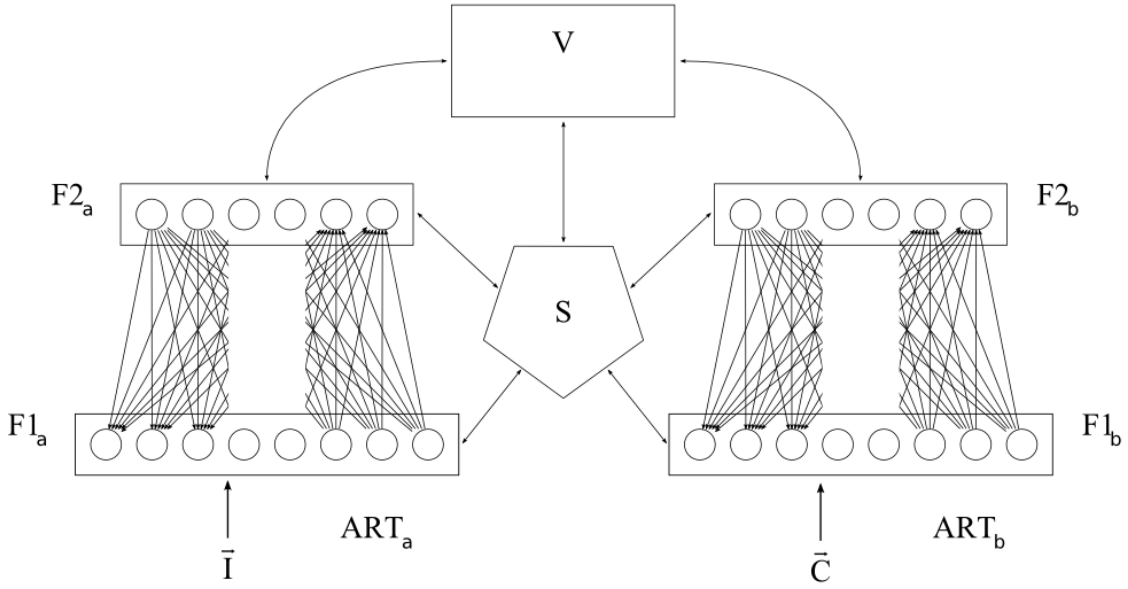

Figure 2: A visualization of the Fuzzy ART-MAP architecture (Muller, 2006).

3 The Object Recognition Algorithm

The object recognition algorithm I implemented is a combination of the two major components described above along with some other processing techniques that are necessary in order to normalize the data and improve efficiency. In this section, I describe the details of the implementation of the algorithm. All aspects of the algorithm, excluding the classifiers, were implemented from scratch using MATLAB (making use of MATLAB's native functions); this includes the implementation of the Gabor filters. A summary the algorithm is provided below.

The Algorithm

Inputs: $\mathcal{T}$ set of training example images (containing positive and negative examples for a category), $\mathcal{Y}$ set of test images, $\mathcal{P}$ set of parameters for the Gabor filters, d the size of the feature vector for classification, $\mathcal{C}$ a classifier.

Returns: Classification predictions P for test set $\mathcal{Y}$.

1: Normalize and down-sample (using bicubic interpolation) all images in $\mathcal{T}, \mathcal{Y}$ so that their largest dimension is 150 pixels.
2: Filter all images using Gabor filters defined by $\mathcal{P}$. Result is $|\mathcal{P}|$ filtered images per original image.
3: Down-sample (using bicubic interpolation) all filtered images to 30×30 pixels.
4: For each image in $\mathcal{T}, \mathcal{Y}$ create a feature vector by flattening out the filtered images (in a fixed order) into vectors and concatenating. Result of this step is two sets of features F_t and F_y (where each feature vector in train set also associated with a label).
5: Perform PCA on F_t to find new d-dimensional basis $\mathcal{B}$ that captures a maximal amount of variance.
6: Project F_t and F_y on to $\mathcal{B}$. Call their projections $F_t^{'}$ and $F_y^{'}$.
7: Train $\mathcal{C}$ on $F_t^{'}$ and obtain output predictions P for $F_y^{'}$.

3.1 Feature Extraction

The first step in feature extraction is to regularize the input images. Prior to feature extraction the images are down-sampled using a bicubic interpolation (a native function of MATLAB) so that their largest dimension is 150 pixels (MATLAB, 2010). The aspect ratio of the images is kept constant at this stage, however, in order to avoid distortions that could effect feature extraction. The images are then normalized so that the mean intensity value is 0 (negative values are permissible) and the standard deviation in intensity is 1.

Following these preprocessing steps, the images are filtered using a battery of Gabor filters. The parameters of these Gabor filters are taken from (Serre and Riesenhuber, 2004). In this work, it is shown that these parameter settings provide responses that agree with experimental neurophysiology results. Table 1 summarizes the parameters used. The four orientation settings are used for each set of the other parameters, and thus, there are $16 \times 4 = 64$ filters in total, giving 64 unique filtered images from a single original image.

To enforce regularity between the features for different images, the filtered images are then down-sampled (again using bicubic interpolation) to 30×30 pixels. Feature vectors for each original image are then formed by flattening out the pixel matrix for each filtered image in a vector and then concatenating these 64 vectors together. This then gives a feature vector of length $30 \times 30 \times 64 = 57600$.

Band	1	2	3	4	5	6	7	8
Size	7 & 9	11 & 13	15 & 17	19 & 21	23 & 25	27 & 29	31 & 33	35 & 37
σ	2.8 & 3.6	4.5 & 5.4	6.4 & 7.3	8.2 & 9.2	10.2 & 11.3	12.3 & 13.4	14.6 & 15.8	17.0 & 18.2
λ	3.5 & 4.6	5.6 & 6.8	7.9 & 9.1	10.3 & 11.5	12.7 & 14.1	15.4 & 16.8	18.2 & 19.7	21.1 & 22.8
θ	$0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}$							

Table 1: Parameters used for Gabor filters. Grouped by the bandwidth of the filters. $\gamma = 0.5$ in all settings.

3.2 Dimensionality Reduction

A practical issue that then arises is that learning using feature vectors of such a large size is extremely inefficient, especially when using neural networks. For this reason, I reduce the dimensionality of the feature vector using PCA (Jolliffe, 1986). Specifically, I determine a new basis that is defined by the eigenvectors of the covariance matrix of the training features. The dimensionality of this new basis, d, was selected through trail and error and 50 was found to be an optimal value. All feature vectors in the training set and testing set are then projected down to this new basis before they are fed to the classifier.

Of course, a technical quirk in this set-up is that the test features are not used in the determination of the reduced basis. This means that the reduced basis may not be optimal given the full set of features (i.e. both training features and test features). However, the use of only the training features is a necessity, as the use of the test features in the determination of the basis would amount to training on the testing data, which would induce over-fitting. Moreover, the model should have no knowledge of the test data prior to its classification, so this immediately precludes using the test data to learn the reduced basis.

3.3 Classification

Finally, the reduced feature vectors for the training data (along with the correct labels) are used to train the classifier. The main classifier of interest here is the Fuzzy ART-MAP classifier; however, any classifier could be used.

Indeed, in order to keep things general, only binary classification is performed. That is, the classifier is fed positive examples for a category and random negative examples. Thus, in a single instantiation of the algorithm only one object category is learned. This protocol is used, despite the fact that Fuzzy ART-MAP can handle multiple categories, since the majority of previous work focuses on binary classification (Pinto et al., 2008; Serre et al., 2005).

Figure 3: Some example images from the object categories used.

4 Results

The object recognition algorithm was tested on the Caltech-101 data set (Fei-Fei et al., 2004). This data-set contains pictures belonging to 101 categories. It is especially interesting since many of the categories do no contain a large number of examples, thus enforcing biologically plausible experiments. The algorithm was tested on three categories: airplanes, motorbikes, and faces (the easier of the two face categories).

For the classification of each category, 10 positive examples and 50 negative examples were used. This set-up is intended to be biologically plausible since in natural settings only a small number of positive training examples are necessary and the number of negative training examples is usually far greater. That is, a human can learn an object category from few examples, and during the learning process they will certainly encounter a large number of negative examples.

Randomized 10-fold cross-validation was also used to ensure that the reported classification accuracies are unbiased. That is, for each category the algorithm was run 10 times using different random subsets of the positive and negative images. The relative orderings of the images were also shuffled, which is important because Fuzzy ART-MAP is sensitive to the ordering of training examples (Carpenter et al., 1992).

Lastly, as mentioned above, the algorithm was tested using both the Fuzzy ART-MAP classifier and an out-of-the-box linear SVM classifier. This allows for an examination of the effect of using a biologically realistic classifier with biologically realistic features. The results using both these classifiers are tabulated below.

The mean classification accuracy refers to the average percentage of images that

Object Category	Mean Accuracy	Standard Deviation
Airplanes	88.9 %	3.45 %
Faces	85.9 %	3.41 %
Motorbikes	90.9 %	2.76 %

Table 2: Results with Fuzzy ART-MAP classifier

Object Category	Mean Accuracy	Standard Deviation
Airplanes	48.0 %	5.01 %
Faces	49.8 %	4.64 %
Motorbikes	48.6 %	3.66 %

Table 3: Results with SVM classifier

were correctly classified over the 10-fold cross-validation. The standard deviation is the deviation of classification accuracy across the 10-fold cross-validation. Interestingly, the mean classification accuracy using the Fuzzy ART-MAP classifier is far higher across all categories. The standard deviations are also relatively small, showing that the classification values obtained are representative of the algorithms' true classification accuracy.

5 Discussion

5.1 Analysis of Results

Firstly, the results obtained point to the importance of using the proper classification algorithm when performing object recognition. With identical features the Fuzzy ART-MAP classifier performed quite well while the SVM classifier performed very poorly. Indeed, the interaction between the features and the classifier has a dramatic impact in this experiment, and interestingly in this experiment the biologically realistic classifier out-performed the non-biologically realistic classifier. Thus, it is clear that using a biologically realistic classifier can have a non-trivial impact on the outcome of object recognition systems.

There is also a plausible explanation for the performance gap between the two classifiers. As mentioned above, biological vision systems (and biologically inspired computer vision systems) rely on non-linear pooling as visual input is filtered and moves through layers of the visual processing hierarchy. For example, the complex cells pool together (in a non-linear manner) the outputs of the simple cells (which can be modelled as Gabor filters). It is possible that the early stages of the ART_a part of the Fuzzy ART-MAP network learn to perform a similar sort of feature pooling during training. That is, it is possible that the neural network is performing some implicit feature extraction prior to classification.

Extensive examination of the neural networks learned weights and voting protocol would be necessary to fully corroborate this explanation, of course. However, the explanation is supported by the fact that the SVM used here is incapable of performing non-linear pooling, since it works solely with linear operations. Moreover, Serre et al. (2005) found that in their model, where they performed non-linear pooling during feature extraction, SVM outperformed a neural network. It is quite possible that in their experiment the non-linear pooling prior to classification made the implicit feature extraction stages of the neural network redundant and possibly even induced over-fitting.

The results obtained are also interesting in that the algorithm was able to produce reasonable accurate predictions given only a small training set. Indeed, the mean classification accuracies are qualitatively quite good, even when compared to state-of-the-art algorithms. For example, using a multi-stage architecture, Serre et al. (2005) were able to obtain mean classification accuracies of 96.7%, 98.0%, 95.9% for the airplane, motorbike, and faces categories respectively. These state-of-the-art results are significantly more accurate than the the results obtained here. However, in that paper they used large training sets (as large as were permitted by the Caltech-101 set). Thus, given that only a small training set was used here, at least in a qualitative manner the results compare favourably.

5.2 Possible Improvements and Future Directions

There are a number of ways in which the algorithm presented could be improved. First, the fact that PCA had to be performed prior to classification takes away from the biological plausibility. A more efficient implementation of the Fuzzy ART-MAP classifier, perhaps even a parallel GPU-based implementation, would allow for the full feature vector to be used. Moreover, given more time it would have been interesting to run the algorithm using other biologically plausible classifiers in order to see how they compare with the Fuzzy ART-MAP results. For example, it would be interesting to see how a back-propogation based neural net performed.

An interesting continuation of this work would be to analyze the performance of the object recognition system with increasing sizes of training sets and also varying ratios of positive vs. negative examples. Specifically, it would be interesting to run enough experiments using different training set sizes so that the performance of the classifier could be approximately mapped as a function of the training set size. If the object recognition system is truly biologically realistic one would expect this function to have a steep slope for small training set values followed by a plateau. That is, one expects biologically realistic systems to benefit from more training examples only when the training set is quite small (i.e. on the order of tens of examples). Moreover, one would also expect that a biologically realistic system would not be adversely affected by increasing the number of negative examples relative to positive ones. Since humans encounter countless negative examples whilst learning object categories it is

reasonable to expect biologically realistic computer visions to perform well in such a setting. The number of experiments necessary to properly analyze these relationships would be very large; however, the results would likely be extremely informative.

Lastly, perhaps the most interesting direction for future work would be a more systematic analysis of the effect of non-linear pooling prior to classification. Specifically, it would be interesting to test the hypothesis that biologically inspired classifiers, which perform implicit feature selection, perform better when fed features that have not already been been pooled in a non-linear manner. Some work has shown that adding multiple non-linear pooling steps can increase classification accuracy (Jarrett et al., 2009). However, it would be interesting to systematically examine how classification accuracy was effected by increasing the number of non-linear pooling steps and by using different classifiers, some of which perform implicit non-linear pooling.

References

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., and Rosen, D. B. (1992). Fuzzy artmap: A neural network architecture for incremental supervised learning of analog multidimensional maps. *Neural Networks, IEEE Transactions on*, 3(5):698–713.

Daugman, J. G. et al. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Optical Society of America, Journal, A: Optics and Image Science*, 2(7):1160–1169.

Fei-Fei, L., Fergus, R., and Perona, P. (2004). Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 178–178. IEEE.

Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2146–2153.

Jolliffe, I. T. (1986). *Principal component analysis*, volume 487. Springer-Verlag New York.

Jones, J. P. and Palmer, L. A. (1987). An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal Neurophsyiology*, 58(6):1233–1258.

LeCun, Y., Kavukvuoglu, K., and Farabet, C. (2010). Convolutional networks and applications in vision. In *Proc. International Symposium on Circuits and Systems (ISCAS'10)*. IEEE.

MATLAB (2010). *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts.

Movellan, J. R. (2002). Tutorial on gabor filters. *Open Source Document*.

Muller, C. (2006). Artmap structure. `http://en.wikipedia.org/wiki/File:ARTMAP.png`. Accessed: 2013-04-30.

Pinto, N., Cox, D. D., and DiCarlo, J. J. (2008). Why is real-world visual object recognition hard? *PLoS computational biology*, 4(1):e27.

Scherer, D. and Behnke, S. (2009). Accelerating large-scalge convolutional neural networks with parallel graphics mutli-processors. In *Proceedings of Neural Information Processing Systems (NIPS)*.

Serre, T. and Riesenhuber, M. (2004). Realistic modeling of simple and complex cell tuning in the hmax model, and implications for invariant object recognition in cortex. Technical report, DTIC Document.

Serre, T., Wolf, L., and Poggio, T. (2005). Object recognition with features inspired by visual cortex. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 994–1000. IEEE.

Woodbeck, K., Roth, G., and Chen, H. (2008). Visual cortex on the gpu: Biologically inspired classifier and feature descriptor for rapid recognition. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference*, pages 1–8. IEEE.

Yang, G. (2010). 2d gabor filter example. `http://www.mathworks.com/matlabcentral/fileexchange/23253-gabor-filter'`. Accessed: 2013-04-30.