A Selective Attention Based Method for Visual Pattern Recognition

Albert Ali Salah (SALAH@Boun.Edu.Tr)
Ethem Alpaydın (ALPAYDIN@Boun.Edu.Tr)
Lale Akarun (AKARUN@Boun.Edu.Tr)
Department of Computer Engineering; Boğaziçi University,
80815 Bebek Istanbul, Turkey

Abstract

Parallel pattern recognition requires great computational resources. It is desirable from an engineering point of view to achieve good performance with limited resources. For this purpose, we develop a serial model for visual pattern recognition based on the primate selective attention mechanism. The idea in selective attention is that not all parts of an image give us information. If we can attend to only the relevant parts, we can recognize the image more quickly and using less resources. We simulate the primitive, bottom-up attentive level of the human visual system with a saliency scheme, and the more complex, top-down, temporally sequential associative level with observable Markov models. In between, there is an artificial neural network that analyses image parts and generates posterior probabilities as observations to the Markov model. We test our model on a well-studied handwritten numeral recognition problem, and show how various performance related factors can be manipulated. Our results indicate the promise of this approach in complicated vision applications.

Introduction

Primates solve the problem of visual object recognition and scene analysis in a serial fashion with *scanpaths* (Noton & Stark, 1971), which is slower but less costly than parallel recognition (Tsotsos, Culhane, Wai, Lai, Davis, Nuflo, 1995). The idea in selective attention is that not all parts of an image give us information and analysing only the relevant parts of the image in detail is sufficient for recognition and classification.

The biological structure of the eye is such that a high-resolution fovea and its low-resolution periphery provide data for recognition purposes. The fovea is not static, but is moved around the visual field in saccades. These sharp, directed movements of the fovea are not random. The periphery provides low-resolution information, which is processed to reveal salient points as targets for the fovea (Koch & Ullman, 1985), and those are inspected with the fovea. The eye movements are a part of overt attention, as opposed to covert attention which is the process of moving an attentional *'spotlight'* around the perceived image without moving the eye.

In the primate brain, information from the retina is routed through the lateral geniculate nucleus (LGN) to the visual area V1 in the occipital lobe. The *'what'* pathway, also known as the *ventral* pathway for anatomical reasons, goes through V4 and inferotemporal cortex (IT). The *'where'* pathway, or the *dorsal* pathway, goes into the posterior parietal areas (PP) (Ungerleider & Mishkin, 1982). The ventral pathway is crucial for recognition and identification of objects, whereas the dorsal pathway mediates the location of those objects. We should note that although recent findings point towards a distinction between perception and guidance of action (Crick & Koch, 1990) instead of a distinction between different types of perception, the issue is not resolved in favour of a specific theory (Milner & Goodale, 1995).

The serial recognition process gathers two types of information from the image: The contents of the fovea window, and the location to which the fovea is directed. We call these *'what'* and *'where'* information, respectively (Ungerleider & Mishkin, 1982). The object is thus represented as a temporal sequence, where at each time step, the content of the fovea window and the fovea position are observed.

Recurrent multi-layer perceptrons were used to simultaneously learn both the fovea features and the class sequences (Alpaydın, 1996). Other techniques are explored in the literature to apply the idea of selective attention to classification and analysis tasks (Itti, Koch, Niebur, 1998; Rimey & Brown, 1990). Our approach is to combine a feature integration scheme (Treisman & Gelade, 1980) with a Markov model (Rimey & Brown, 1990).

We use handwritten numeral recognition to test our scheme. In our database (UCI Machine Learning Repository, Optdigits Database), there are ten classes (numerals from zero to nine) with 1934 training, 946 writer-dependent cross-validation, 943 writer-dependent and 1797 writer-independent test cases. Each sample is a 32×32 binary image which is normalized to fit the bounding box. There are parallel architectures to solve this problem in the literature (Le Cun, Boser, Denker, Henderson, Howard, Hubbard, Jackel, 1989), and they have good performance, but our aim is to design a scalable system which is applicable to problems where the input data is high-dimensional (e.g. face recognition), or not of fixed size (e.g. recognizing words in cursive handwriting). Implementing a parallel scheme with good performance is not trivial in such cases.

This paper is organized as follows: We first describe our model and its three levels. Then we report our simulation results. In the last section we summarize and indi-

cate future directions.

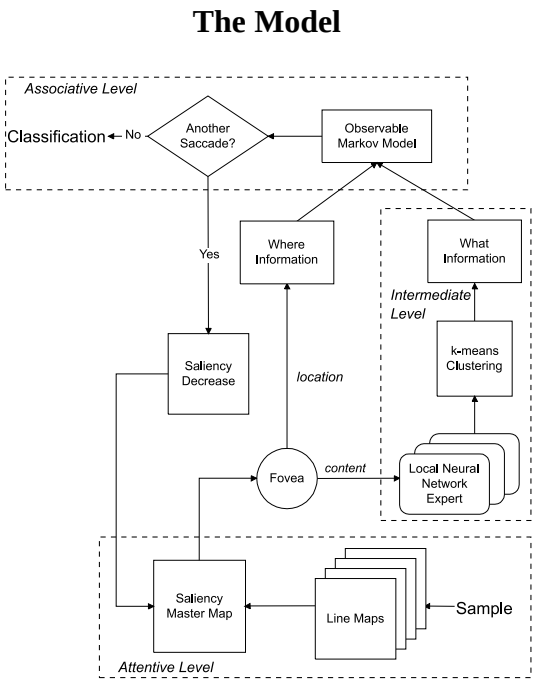

Figure 1: The selective attention model for visual recognition.

The block diagram of the system we propose is given in Fig. 1. It is composed of the *attentive level* that decides on where to look, the *intermediate level* that analyses the content of the fovea, and the *associative level* that integrates the information in time.

Attentive Level

In the first step of the model, the bottom-up part of the visual system is simulated. We work on 12×12 downsampled images to simulate a low-resolution resource. This slightly decreases the classification accuracy, but speeds up the computation considerably. Convolving the digit image with 3×3 line orientation kernels produces four line orientation maps in 0°, 45°, 90° and 135° angles. These are combined in a saliency master map, which indicates the presence of aligned lines on the image.

Line orientations are detected by different primitive mechanisms in the visual cortex, operating in coarse, intermediate and fine scales (Foster & Westland, 1998). We can also talk about simple, complex and hypercomplex cell structures in the visual cortex, that deal with increasing levels of complexity and decreasing levels of resolution. In constructing the saliency map, we use the simplest set of features to decrease the computational cost. Our experiments showed that adding other feature detectors like corner maps, Canny edge detector, and further line orientation maps in higher resolutions increased the classification accuracy only slightly, whereas the increase in the computational cost was significant.

The saliency map indicates the interesting spots on the image. We simulate the fovea by moving a 4×4 window over the 12×12 downsampled image. The saliency values of the visited spots and their periphery are decreased, and these spots are not visited again. This process has a biological counterpart: Once neurons attuned to detect a specific feature fire in the brain, they are temporarily inhibited. Subsequently, subjects respond slower to previously cued locations (Klein, 2000).

Intermediate Level

The simulation of shifts of attention should provide us with *'what'* and *'where'* information, but we want them to be sufficiently quantized to be used in the associative level. We divide the image space into uniform regions, in effect, performing a quantization on the location information. We use a second set of overlappping windows to reduce the effect of window boundaries, as shown in Fig. 2. We obtain a time-ordered sequence of visited regions after the simulation of shifts. This constitutes the *'where'* stream for the particular sample (Fig 3).

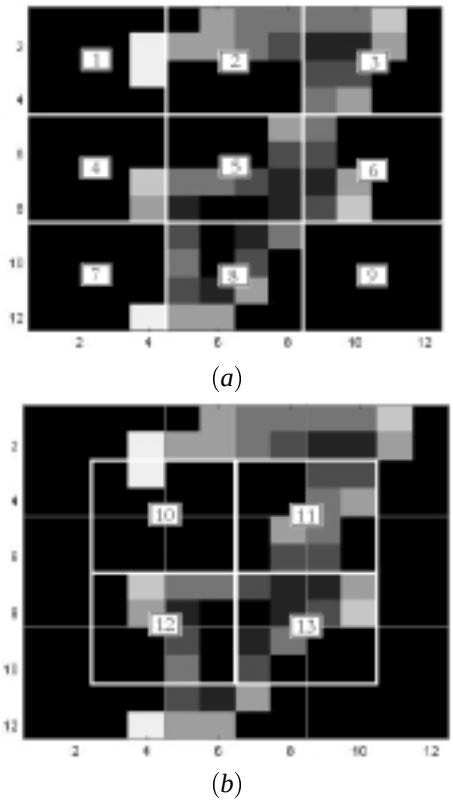

Figure 2: Regions of the downsampled image. (a) The uniform regions. (b) The additional, overlapping regions. Notice how the corner at the intersection of 5^{th}, 6^{th}, 8^{th}, and 9^{th} regions are missed in those regions, but captured clearly in the 13^{th} region.

As fovea contents, we extract 64-dimensional real-valued vectors. These vectors are produced by concate-

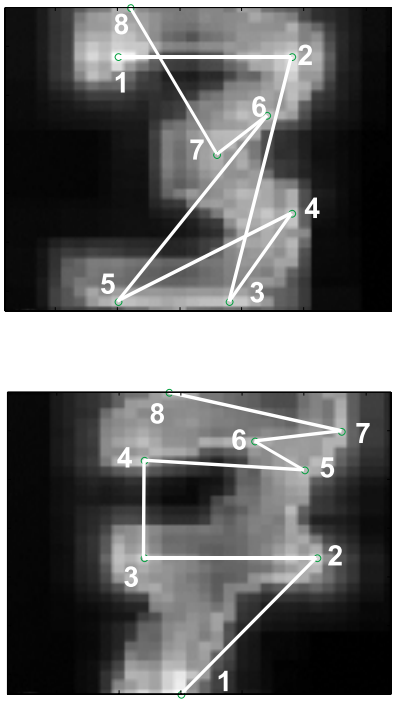

Figure 3: The saliency master maps of two examples from the Optdigits database. Locations with high intensity indicate high saliency values. The locations visited by the fovea are connected with a line, and enumerated in the order they are visited.

nating the corresponding 4×4 windows on the line maps. We prefer using the concatenated line maps to inspecting the original bitmap image, because the line maps indicate the presence of features more precisely. Furthermore, since they were constructed in the attentive level, they come at no additional cost. In any case, we need a vector quantization on the fovea contents before passing them to the associative level.

In order to efficiently quantize this information, we train artificial neural network experts at each region of the image. The experts are single-layer perceptrons (SLP) that are trained in a supervised manner (Bishop, 1995). Their input is the 64-dimensional fovea content vector. The output of the experts are 10-dimensional class posterior probability vectors, which are then clustered with *k*-means clustering (Duda & Hart, 1973) to obtain the *'what'* information stream. We select single-layer perceptrons over multi-layer perceptrons for a number of reasons. Multi-layer perceptrons overlearn the training data quickly, and perform worse on the cross-validation set. The number of parameters we need to store for the multi-layer perceptron is larger, and the training time is significantly higher. These properties make the single-layer perceptron the better choice of expert in the final model.

Associative Level

In the associative level, the two types of quantized information are combined with a discrete, observable Markov model (OMM) (Rabiner 1989). We treat the regions visited by the fovea as the states of a Markov model, and the quantized output of the local artificial neural network experts as the observations from each state. We simulate eight shifts for each sample in the training set, obtain the *'where'* and *'what'* streams, and adjust the probabilities of the single Markov chain of the corresponding class to maximize the likelihood of the training data. Training an observable Markov model is much faster than training a Hidden Markov Model.

In the observable model, the model parameters are directly observed from the data. Since we know the states, we can count the state transitions, and normalize the count to find the state transition probabilities a_{ij}, as well as the initial state distribution probabilities π_i. Similarly, we count the occurrences of the observation symbols (quantized outputs of the local neural networks) at each state, and normalize them to find the observation symbol probability distribution $b_j(k)$:

$$\pi_i = \frac{\#\text{ of times in } S_i \text{ at time } t=1}{\#\text{ of observation sequences}} \quad (1)$$

$$a_{ij} = \frac{\#\text{ of transitions from } S_i \text{ to } S_j}{\#\text{ of transitions from } S_i} \quad (2)$$

$$b_j(k) = \frac{\#\text{ of times in } S_j \text{ observing } v_k}{\#\text{ of times in } S_j} \quad (3)$$

Finding the probability of the observation sequence is much simpler in the observable Markov model, since the states are visible. We just multiply the corresponding state transition probabilities and the observation probabilities:

$$P(O,S|\lambda) = \pi_{S_1} b_{S_1}(v_1) \prod_{i=2}^{n} a_{S_{i-1}S_i} b_{S_i}(v_i), \quad (4)$$

where S is the state sequence, O is the observation sequence, and $\lambda = \{\pi_i, a_{ij}, b_j(k)\}$ stands for the parameters of the Markov model. $i, j = 1..N$ are indices for states, $k = 1..M$ is the index for the observation symbols.

The Markov model is trained with a limited training set, and if the number of states and observation symbols is large, there will be connections that are not visited at all. Since the model is probabilistic, having a transition or observation probability of zero is not desired. Instead, the transitions that have not occurred within the training set should have a low probability in the model. This is what we do in the post-processing stage. We scan the probabilities of the trained Markov model, and replace all probabilities lower than a threshold (0.001) with the threshold value. Then we normalize the probabilities once more. This is a simple and fast procedure that achieves the desired effect.

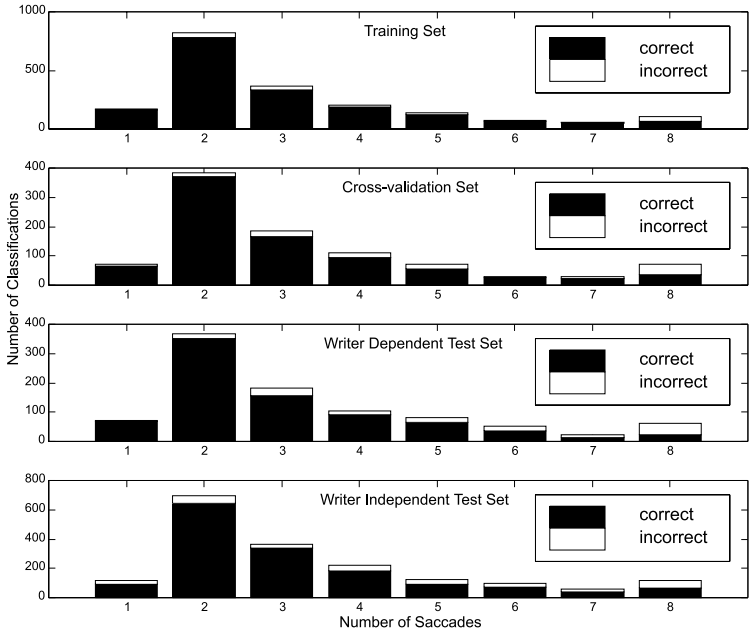

Figure 4: Dynamic fovea simulation results. These are the histograms of the number of correctly and incorrectly classified digits after each shift. See also Figs. 5 and 6.

We have also tried Hidden Markov Models where the states are not visible and where the concatenated *where-what* information is the observation, but this structure performed worse than the observable Markov model.

Dynamic Fovea

One important advantage of using a Markov model is the ease with which we can control the number of shifts necessary for recognition. In the training period, our model simulates eight shifts, which is set as the upper bound for this particular application. After each shift, the Markov model has enough information to give a posterior probability for each class. We may calculate the probability $\alpha_t(c)$ of the partial sequence in the Markov model, which reflects the probability of the sample belonging to a particular class, given the *'where'* and *'what'* information observed so far. Using Eq. 4, we have

$$\alpha_t(c) = P(O_1,...,O_t,S_1,...,S_t|\lambda_c), \tag{5}$$

where $O_1,...,O_t$ is the observation sequence up to time t, $S_1,...,S_t$ is the state sequence, and λ_c are the parameters of the Markov model for class c.

We can use this probability to stop our shifts whenever we reach a sufficient level of confidence in our decision. Let us define $\alpha_t^*(c)$, the posterior probability for class c at time t:

$$\hat{p}(c|O_1,...,O_t,S_1,...,S_t) \equiv \alpha_t^*(c) = \frac{\alpha_t(c)}{\sum_{j=1}^{K}\alpha_t(j)} \tag{6}$$

Let τ be the threshold we use as our stopping criterion:

$$\alpha_t^*(c) \geq \tau, \tag{7}$$

where the value of τ is in the range [0,1]. If we assume that absolute certainty is not reached anywhere in the model and $\alpha_t^*(i)$ is always below 1.0, selecting $\tau = 1.0$ is equivalent to treating all samples as equally difficult and doing eight shifts. Conversely, selecting $\tau = 0$ is equivalent to looking at the first salient spot and classifying the sample.

Selecting a large value for τ trades off speed for accuracy. With a well selected value, we devote more time for difficult samples, but recognize a trivial sample in a few shifts.

Results

In this section we present our simulation results. We give additional information about the techniques we employ in subsections.

Local Experts

Implementing local artificial neural network experts both increases the classification accuracy and decreases the complexity and classification time. The single-layer perceptron returns a 10-dimensional vector from a 64-dimensional linemap image. Since it is trained in a supervised manner, it provides more useful information for classification to the later Markov model, as our experiments indicated.

Dynamic Fovea Simulation

When we simulate the dynamic fovea with a fixed threshold of $\tau = 0.95$, we get 85.67 per cent classification accuracy with 5.46 per cent standard deviation on the writer-dependent test set. The average number of shifts is 3.33, which corresponds roughly to seeing one thirds of the image in detail. This justifies our claim that analyzing only a small part of the image is enough to recognize it. On the writer independent test set, the classification accuracy is 84.63 per cent, with a standard deviation of 7.58 per cent, and the average number of shifts is 3.37 (See Fig. 4 for the histograms depicting the distribution of classifications over the shifts). We are doing less than half the number of shifts we were doing, but the performance decrease is less than a standard deviation.

The advantages of simulating a dynamic fovea become apparent when we inspect Figs. 5 and 6. The accuracy of classification increases when we increase the threshold, because a higher threshold means making more shifts to get a more confident answer. A lower threshold means that a quick response is accepted. What happens is that the average number of shifts increase sharply if the threshold is set to a value very close to 1.0. In this case, the classifier cannot exceed the threshold probability with eight shifts, and selects the highest probability class, without doing any more shifts. This is the reason behind the relatively high number of correct and incorrect classifications after eight shifts in Fig. 4.

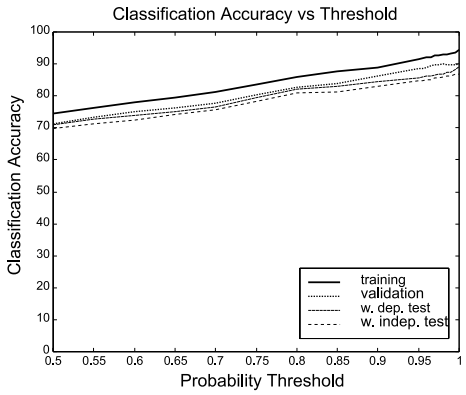

Figure 5: Accuracy vs threshold value in dynamic fovea simulation

Simulation Results

We summarize the results we obtain in Table 1. The first column of the table shows the method employed. The successive columns indicate the classification accuracy and its standard deviation on the training, cross-validation, writer-dependent test and writer-independent test sets.

In the first two rows, we do eight shifts, generate the posterior probabilities of classes by the local artificial neural network experts and take a vote without treating them as a sequence. Soft Voting takes into account the 10-dimensional outputs of the experts instead of a single class code. Comparing the results with the OMM results show that the order information which is lost during voting but used in OMM is useful. Another observation is that the post-processing method we use increases the performance by one standard deviation, which is a significant increase.

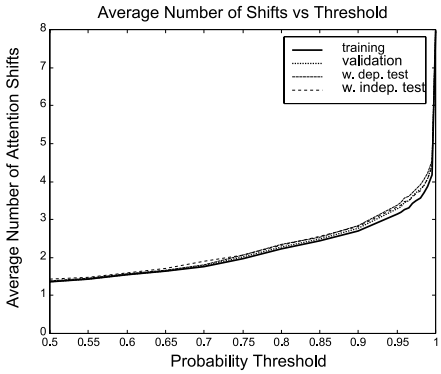

Figure 6: Average number of shifts vs threshold value in dynamic fovea simulation

The dynamic fovea simulation has a lower classification accuracy, but it only needs 3.2 shifts on the average, instead of the previous eight.

Finally, the last row indicates the accuracy of an all-parallel scheme. We use a multi-layer perceptron (MLP) with 32×32 binary input and 10 hidden units. Although the MLP has a good accuracy in this problem, it is not scalable due to the curse of dimensionality.

Conclusions and Future Work

The selective attention mechanism exploits the fact that real images often contain vast areas of data that are insignificant from the perspective of recognition. A low-resolution, downsampled image is scanned in parallel to find interesting locations through a saliency map, and complex features are detected at those locations by means of a high-resolution fovea. Recognition is done serially as the location and feature information is combined in time. By keeping the parallel part of the method simple, we can speed-up the recognition process considerably.

Our tests have demonstrated that an observable Markov model may replace an HMM for the two-pathway selective attention model. The observable scheme is easier to train and use, and performs better. The dynamic fovea simulation reveals further benefits of serializing the recognition process. We can control the time we spend on an image, and differentiate between simple and confusing images. This is a desirable property in a classifier, since it allows us to apply more reliable and costly methods to the confusing samples if we wish. It also reduces the average recognition time, but it

Table 1: Summary of Results

Method	Performance			
	Training	**Validation**	**Writer Dep. Test**	**Writer Indep. Test**
SLP+Simple Voting	86.74(± 9.90)	85.92(± 9.39)	64.51(±25.62)	62.66(±25.74)
SLP+Soft Voting	93.85(± 4.47)	91.25(± 7.07)	74.35(±27.66)	73.89(±26.67)
OMM+SLP	95.32(± 3.72)	83.98(±15.37)	84.42(±14.94)	80.92(±16.24)
OMM+SLP + post-processing	94.37(± 3.33)	90.07(± 7.92)	89.73(± 8.68)	87.37(± 8.73)
Dynamic fovea	91.41(± 4.56)	88.47(± 7.98)	85.67(± 5.46)	84.63(± 7.58)
MLP	99.92(± 0.12)	97.45(± 0.28)	97.25(± 0.42)	94.54(± 0.21)

must be remembered that the construction of the saliency map is necessary for all samples. Although we reduce the time complexity of the associative level by half, the overall gain is less than that.

Our attempt to classify digits may be seen as a toy problem, since the ratio of the fovea area to the image is not high enough to demonstrate the benefits of our model. Although the accuracy is lower than the state-of-the-art parallel approaches in the literature (e.g. the MLP result in Table 1), the selective attention mechanism is much more appropriate for applications where parallel processing is too cumbersome to use, and the number of input dimensions is high.

We are planning to employ our model in a more difficult task, such as face recognition, where an all-parallel classifier, like the MLP, would be unnecessarily complex; in a face, small regions of the face like eyes, nose, mouth give us information. The saliency scheme has to be modified for this purpose, as facial features necessitate different and more complex feature detectors. The fovea size also needs to be adjusted for the specific task.

Acknowledgments

This work is supported by Boğaziçi University Research Funds 00A101D.

References

Alpaydın, E. (1996). Selective Attention for Handwritten Digit Recognition. In D.S. Touretzky, M.C. Mozer, & M.E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems 8*, 771-777.

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press.

Crick, F., & Koch, C. (1990). Towards A Neurobiological Theory Of Consciousness. *Seminars in the Neurosciences*, 2, 263-275.

Duda, R., & Hart, P. (1973). *Pattern Classification and Scene Analysis.* New York: John Wiley & Sons.

Foster, D.H., & Westland, S. (1998). Multiple Groups of Orientation-selective Visual Mechanisms Underlying Rapid Oriented-line Detection. *Proc. Royal Society London*, 265, 1605-1613.

Itti, L., Koch, C., & Niebur, E. (1998). A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20, 11.

Klein, R.M. (2000). Inhibition of Return. *Trends in Cognitive Science*, 4(4), 138-147.

Koch C., & , Ullman, S. (1985). Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry. *Human Neurobiology*, 4, 219-227.

Le Cun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., & Jackel, L.D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1, 4, 541-551.

Milner, A. D., & Goodale, M. A. (1995) *The Visual Brain in Action.* Oxford University Press.

Noton, D, & Stark, L. (1971). Eye Movements and Visual Perception. *Scientific American*, 224, 34-43.

Rabiner, L.R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. IEEE*, 17, 2.

Rimey, R.D., & Brown, C.M. (1990). Selective Attention as Sequential Behavior: Modeling Eye Movements with an Augmented Hidden Markov Model. (Tech. Rep. TR-327). Computer Science, University of Rochester.

Treisman, A.M., & Gelade, G. (1980). A Feature Integration Theory of Attention. *Cognitive Pscychology*, 12, 1, 97-136.

Tsotsos, J.K., Culhane, S.M., Wai, W.Y.K., Lai, Y., Davis, N., & Nuflo, F. (1995). Modeling Visual Attention via Selective Tuning. *Artificial Intelligence*, 78, 507-545.

UCI Machine Learning Repository, Optdigits Database, prepared by E. Alpaydın and C. Kaynak. ftp://ftp.ics.uci.edu/pub/machine-learning-databases/optdigits.

Ungerleider, L.G., & Mishkin, M. (1982). Two cortical visual systems. In D.J. Ingle, M.A. Goodale, & R.J.W. Mansfield (Eds.), *Analysis of visual behavior*, MIT Press.