International Conference on Methods and Models in Computer Science, 2009

# Attention Mechanisms and Component-Based Face Detection

Ji Wan Han
School of Computer Science, University of Hertfordshire,
College Lane, Hatfield, AL10 9AB, UK
Email: j.w.han@herts.ac.uk

Peter C. R. Lane
School of Computer Science, University of Hertfordshire,
College Lane, Hatfield, AL10 9AB, UK
Email: peter.lane@bcs.org.uk

Neil Davey
School of Computer Science, University of Hertfordshire,
College Lane, Hatfield, AL10 9AB, UK
Email: n.davey@herts.ac.uk

Yi Sun
School of Computer Science, University of Hertfordshire,
College Lane, Hatfield, AL10 9AB, UK
Email: y.2.sun@herts.ac.uk

***Abstract*—Locating and identifying complex objects in a visual scene is a typical problem within the areas of computer vision and image analysis. One technique to minimise the size of image to be identified is to base the classification on smaller *features* of the image, which are combined into a more complex structure to identify the complete object. For example, locating two eyes, a nose and a mouth can enable us to identify a face without paying attention to the hair, chin or cheeks. In this paper, we present a system and training technique for learning to recognise an object from its component features. Our system incorporates an attention-based mechanism to predict the location of features. We demonstrate the effectiveness of our system with an experiment in face detection; the attention mechanism is shown to improve the overall classification speed and accuracy of feature location.**

## I. Introduction

The human visual system understands a complex scene by separating it into components and relations between those components (e.g. [3]). The visual process must attend to pieces of the scene in turn, and the way in which each fixation point is selected is based on a combination of information from different sources, including lower-level responses to visual stimuli [17], [18], and higher-level expectations of what objects may be found in the scene [2]. Although recent computer-vision systems have incorporated component-based recognition processes, there are still few which integrate the learning of compound objects with the attention mechanisms which guide the system to areas likely to contain relevant features.

We have developed a system, known as *Cengji* [9] (from the Chinese word for hierarchy), which learns to classify objects using a component-based approach and uses an attention mechanism to improve feature detection. Our system has two layers [10]. The first layer is responsible for locating features within the input scene, and these features are stored in an intermediate representation, which we call the *feature map.* The feature map stores the name of the feature identified, and places it in a 2D array, thus retaining information on the relative positioning of features in the scene. The feature map is used as input to the second layer, which classifies the compound object.

One problem in developing such a two-layer system is in training the first layer to construct an intermediate representation suitable for classifying the complete object. We have developed an iterative training algorithm which automatically constructs the intermediate representation, and augments the training set for the first layer to improve the quality of the feature detectors.

The intermediate representation is used both as a source for the second-layer classifier and as a means to improve the feature detection process. An autoassociative mechanism takes a partially completed feature map and predicts likely positions for the remaining features. These likely positions are then used to guide the feature detectors to areas of the image more likely to contain relevant features. In this manner, Cengji combines information from top-down and bottom-up sources, employing an attention-based process to guide its analysis of a scene. The main aim of this paper is to explore the effectiveness of the attention mechanism.

## II. Component-Based Image Classification

Within the field of computer vision, an important task is to make useful decisions about real physical objects and scenes based on sensed images, and generate a correct and effective representation of the real world [16]. Most of these systems employ a hierachical representation, with features being composed into larger objects, an approach which reflects some of the natural structure in the external world [5]. For example, Dillon *et al* [7] developed *Cite*, a scene understanding and object recognition system, which can generate hierarchical descriptions of visually sensed scenes based on an incrementally learnt hierarchical knowledge base. Behnke *et al* [1] proposed a hierarchical neural architecture for image interpretation, which was based on image pyramids and cellular neural networks inspired by the principles of information processing found in the visual cortex. The identification of meaningful components of an image as an aide to recognising or otherwise interpreting that image has also long formed a core topic in models of human thinking within cognitive science. The CHREST architecture [8], [15] illustrates how attention is strongly determined by expectations and information held in long-term memory.

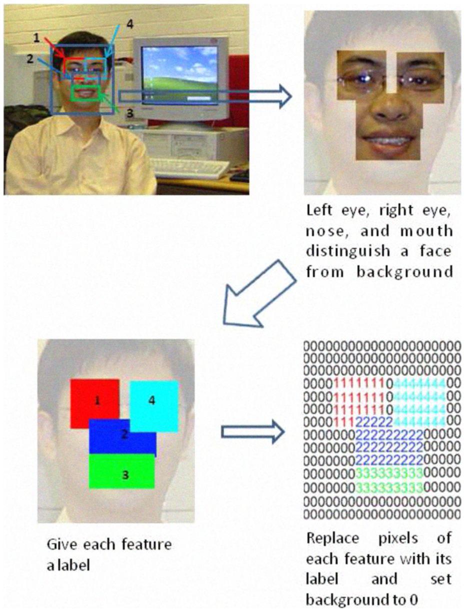
More recent work has focused on using Support Vector Machines (SVM) [4], [6], [11] in multi-layer systems. Heisele *et al* [13] presented a dual-layer SVM algorithm, which learns discriminative components (features) of objects. In this algorithm, component-based face classifiers were combined in the second stage to yield a hierarchical SVM classifier. On the first layer, the component classifiers independently detected components of the face. On the second layer, the combination classifier performed face detection based on the output of the component classifiers. Huang *et al.* [14] have introduced a component-based system for visual classification similar to Cengji. Their system identifies faces, using a collection of 14 features from the face. These features were created automatically from a set of synthetic face images [12]. The features are located on an image by component classifiers within rectangular search regions around the expected positions of the components. Once identified, the features' maximum outputs within a search region and their locations are passed to the top layer in the following format: $(O_1, X_1, Y_1, ..., O_{14}, X_{14}, Y_{14})$. Our system differs from these other approaches in using a *feature map* as an intermediate representation, in order to retain relations between features, and also in employing an active attention mechanism.

## III. Cengji: Attention-Driven Image Analysis

### A. System overview

Fig. 1 provides an overview of our system. Cengji is an image classification system, taking an image as its input and returning a classification of that image as its output. Internally, Cengji is built from two layers. The first consisting of a number of feature detectors, or 'experts', whose job is to recognise specific features within the image; each expert is tailored to recognise one kind of feature, such as an eye within a face. The features identified by all the experts are placed together on a *feature map*, as shown in Fig. 2. The feature map is a grid, of the same size as the input image. The elements of the grid encode the label for the feature which overlaps that position.

The dynamic behaviour of the complete system is captured in Fig. 3. First, the feature detectors are scanned over the image in turn, until one of them identifies their feature; for example, the 'eye expert' will report when it locates an eye in the image. Once the first feature has been located, the feature is placed onto the *feature map*, and passed to the autoassociator. If successful, the autoassociator returns a prediction for the complete feature map, encoding locations for the remaining features. These predicted locations are then passed back to the feature detectors, which attempt to identify the predicted features. When all features have been located, the complete feature map is passed to the second layer, which classifies the feature map. If the autoassociator does not locate further features, the individual detectors will continue to perform a simple scan of the image, adding further information to the feature map if found. This process stops when all the feature detectors have performed a complete scan of the input image, or located a feature based on information from the autoassociator.



Left eye, right eye, nose, and mouth distinguish a face from background

Give each feature a label

```
000000000000000000000
000000000000000000000
000000000000000000000
000011111110444444400
000011111110444444400
000011111110444444400
000011122222444444400
000000022222222200000
000000022222222200000
000000022222222200000
000000033333333300000
000000033333333300000
000000000000000000000
000000000000000000000
```

Replace pixels of each feature with its label and set background to 0

Fig. 2. Composing the feature map: individual features are located and placed onto the feature map.

We have implemented the feature experts and final classifier using Support Vector Machines. The autoassociator is a neural network, trained using backpropagation.

### B. Training technique

The two-layer system has two kinds of classifiers. First are those which detect individual features within an image, such as the eye experts. Second is the classifier which takes the output of the first layer, in the form of a feature map, and then outputs a final classification for the compound object. The overall performance will depend both on the quality of the classifiers at the first layer and the performance of the second layer classifier. In training our system, we are only given the initial input images and their classification. There are two issues to address: how best to train the first layer feature experts, and what input data to use to train the second layer classifier and autoassociator.

We have developed a training procedure, described in Table I, which uses performance of the complete system on the training data to automatically refine the training sets of the first-layer feature experts whilst avoiding the problems of overfitting. We provide an initial training set for each expert, and train them. These initial versions are then used to convert the training images into a set of feature maps, which we classify using the appropriate image classification. We then train the second-layer classifier on these constructed feature maps. Errors in the output are used to improve the training

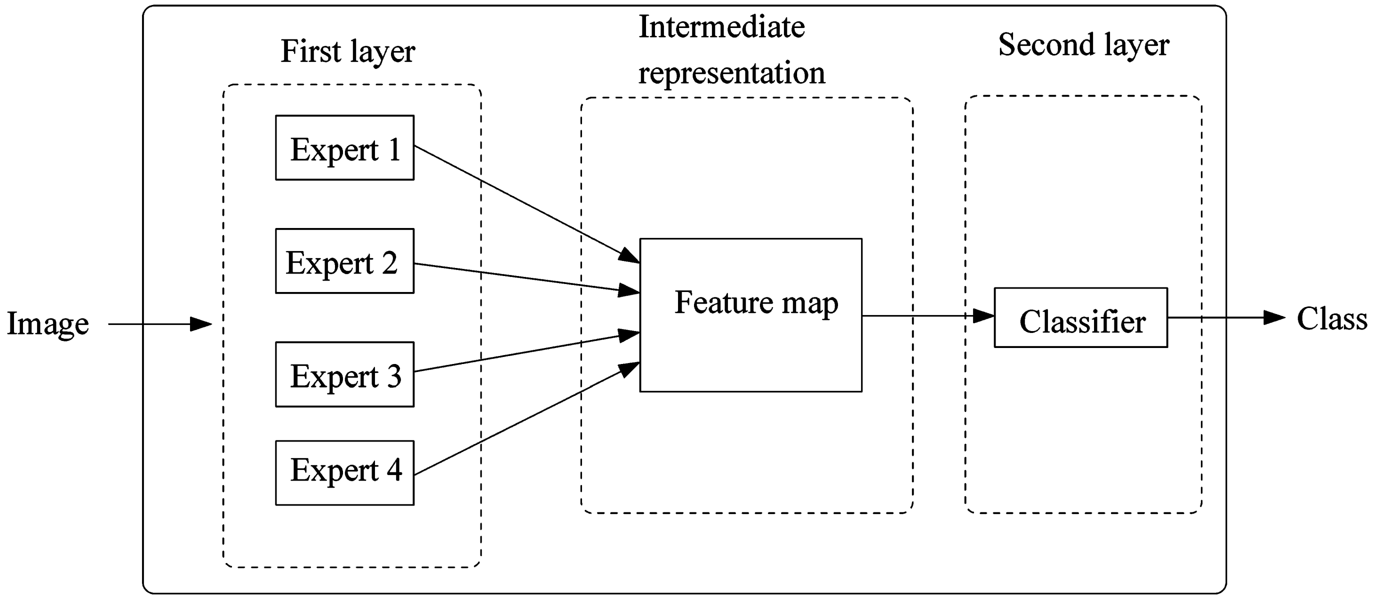


Fig. 1. The Cengji system. The system takes an image as input, returning a classification of that image. Internally, Cengji consists of two layers, communicating through a feature map. The number of experts can be varied, depending on the domain. The autoassociative memory is not shown.

1. *Initialise training sets for each feature expert.* Cut features from face images to form the initial version of the positive training sets. Randomly cut non-features from non-face images and non-feature areas of face images to form the initial version of the negative training sets.
2. *Create first version of feature experts.* Use grid search method to select kernel and optimise kernel parameters for experts using the initial version training sets, then train feature experts on these sets.
3. *Improve feature expert training sets further.* Scan trained experts across both faces and non-faces in face database to generate training sets for feature map expert in the second layer. Train the feature map expert and join the first layer and the second layer together to scan images in face database. Check misclassified images. Add non-detected features to positive feature training sets and false positive detections to negative feature training sets.
4. Use grid search to optimise each expert using new training sets and train feature experts on new training sets.
5. *Get the final version of training sets for all experts.* Repeat step 3 and 4 until the correct detection rate starts to become stable. The creation of training sets is finished.
6. *Use the final system to detect test sets.*

TABLE I
A PROCEDURE TO CREATE THE TRAINING DATABASE.

sets for the feature experts, by adding features that were not detected to the positive training set, and adding image subsets that were mis-classified to the negative training set. The process is then repeated, until the performance of the overall system drops. The final set of constructed feature maps are also used to train the autoassociator.

## IV. EXPERIMENTS

The aim of the following experiments is two-fold: first to demonstrate that the complete Cengji system with its tailored training technique can produce good results in an image classification task; and second to evaluate the impact of the attention mechanism on overall performance and accuracy of feature location.

### A. Method

A dataset of 1000 images was constructed, consisting of 500 faces and 500 non-faces. We used the following sources to construct our face database: the database of faces from AT&T laboratories, Cambridge; the Japanese female facial expression database; the Caltech database; the PIE database and the psychological image collection at Stirling. The BEV1 dataset and Caltech database were used for creating negative samples. Faces were manually cut from these databases and adjusted to a size of $84 \times 96$ pixels. For each face, the individual eyes, mouth and nose were also extracted, manually, to create a training set for the individual feature experts.

The 1000 images were randomly sampled to provide a balanced training set of 700 images and a test set of 300 images. We tested two versions of our system, Cengji with no attention mechanism (CNA), and Cengji with the attention mechanism (CAM). The version with the attention mechanism is as described above. The only difference with the version without the attention mechanism is that the autoassociator is not used, forcing the system to manually scan for all four features, independently.

A 5-fold grid search was used to find the best kernels for the SVM classifiers, and to optimise their parameters. We then used the tailored training procedure described in Table I to train CNA and CAM using the training set of 700 images. Finally, the overall performances of CNA and CAM were compared using the held-out test set of 300 images. Some analysis was performed on the timing and accuracy of feature

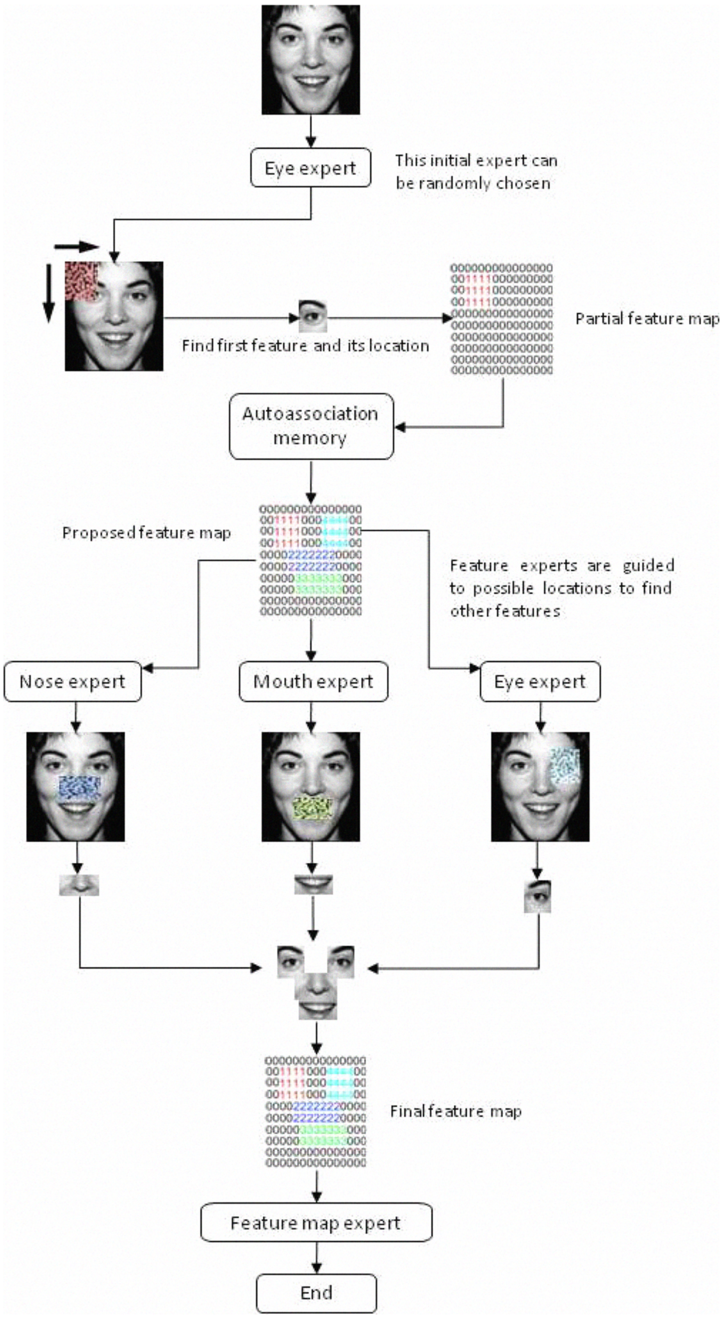

Eye expert
This initial expert can be randomly chosen
Find first feature and its location
Partial feature map
Autoassociation memory
Proposed feature map
Feature experts are guided to possible locations to find other features
Nose expert
Mouth expert
Eye expert
Final feature map
Feature map expert
End

Fig. 3. Flowchart of attention-driven location of features for recognition of a compound object. Each expert is a classifier, identifying the named feature.

location in the two systems.

### *B. Results and analysis*

*1) Overall performance:* Table II shows the overall performance of CNA and CAM in classifying images from the test set. The table shows the overall performance as well as the individual numbers of true and false positives in all cases. We see that CAM had a slightly lower accuracy, 94.33% compared with 95.33%. The precise figures reveal this difference is small (3 images differently classified), with CAM misclassifying 1 face and 2 non-faces.

*2) Training process:* The best kernel for the feature experts was found to be the RBF kernel, but for the second layer a linear kernel outperformed the other kernels. The performance of the individual classifiers, in training, is shown in Table III. The eye and nose experts performed well, but the mouth detector is a little worse.

The training procedure involves a cycle, where the features used in training are added to, including additional false positives as they are encountered by the system. We used cross-

| Images | **Without attention mechanism** | | | | **With attention mechanism** | | | |
|---|---|---|---|---|---|---|---|---|
| | Classification result | | | Correct(%) | Classification result | | | Correct(%) |
| | Correct | Incorrect | Total | | Correct | Incorrect | Total | |
| Faces | 138 | 12 | 150 | 92.00 | 137 | 13 | 150 | 91.33 |
| Non faces | 148 | 2 | 150 | 98.67 | 146 | 4 | 150 | 97.33 |
| Total | 286 | 14 | 300 | **95.33** | 283 | 17 | 300 | **94.33** |

TABLE II
FACE DETECTION TEST RESULTS, COMPARING OUR SYSTEM WITH AND WITHOUT ITS ATTENTION MECHANISM.

| Detector | Accuracy (%) |
|---|---|
| Eye | 96.77 |
| Nose | 95.57 |
| Mouth | 87.26 |

TABLE III
PERFORMANCE IN TRAINING OF INDIVIDUAL FEATURE EXPERTS.

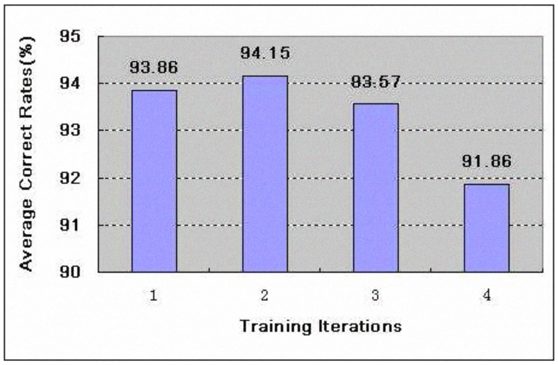


Fig. 4. Relation between training iterations and average correct rates.

validation on the complete training set to control the number of passes through this cycle. Fig. 4 shows the number of iterations (we performed a total of four) against the accuracy of the complete system, on the training data. We find the system performed best after the second iteration, and the training set and parameters created on that cycle were used for evaluating the complete system.

*3) Feature location:* Fig. 5 illustrates the effect of the attention mechanism on the accuracy of feature location. The image on the left shows the location of features obtained with CNA, which uses a simple scanning of the image to locate features. The nose has been located in the wrong place, due to a false positive error by the nose detector. The image on the right indicates the location of features with CAM, which uses the attention mechanism; apart from the use of the attention mechanism, the two systems have otherwise been trained using the same data. As the figure shows, the features are all correctly located. We also found a speed benefit in using CAM: overall the speed improved by 21%, with a greater saving on those images which were faces (of 25%) against those which were not (16%).

### *C. Discussion*

The experimental results demonstrate that our system achieves a good result in face detection, of 94%. This is the accuracy of separating face from non-face images in a held-out test dataset of 300 images. Further, the results show that the attention mechanism improves the time to locate features by an average of 21%, and also the accuracy of located features is improved. We had expected a greater increase in time, because the scanning algorithm requires many comparisons before it reaches the point where a feature may occur. We are currently investigating alternative local search routines to try to improve the search time. Although potentially the training procedure described in Table I could involve a large number of iterations, we were pleased to note our system achieved its best performance on the training set after just two cycles, so the overhead is small. However, the improvement in performance was not so large, indicating that perhaps our initial training sets for the individual feature experts were already comprehensive enough.

## V. CONCLUSION

In this paper, we have described and empirically evaluated an attention-driven, component-based object-recognition system, which we have called Cengji. The complete system achieves a performance of 94% on a held-out test set. The attention mechanism improves the speed of operation by 21%, and also improves the accuracy of locating individual features. We also introduced a training process for iteratively improving the quality of the dataset used for training the system, a training process which can be applied to other component-based recognition systems. These experiments demonstrate that the overall approach of including an attention-based mechanism with component-based image analysis is an effective one. In further work, we shall explore the performance of our system in different domains and attempt to fully exploit the possibilities offered by the feature prediction mechanism.

## REFERENCES

[1] S. Behnke and R. Rojas. Neural abstraction pyramid: A hierarchical image understanding architecture. In *Proceedings of the 1998 IEEE International Joint Conference on Neural Networks*, volume 2, pages 820–825. IEEE World Congress on Computational Intelligence, 1998.

[2] I. Biederman. On the semantics of a glance at a scene. In M. Kubovy and J. R. Pomerantz, editors, *Perceptual Organization*, pages 213–254. Hillsdale, NJ: Lawrence Erlbaum, 1981.

[3] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

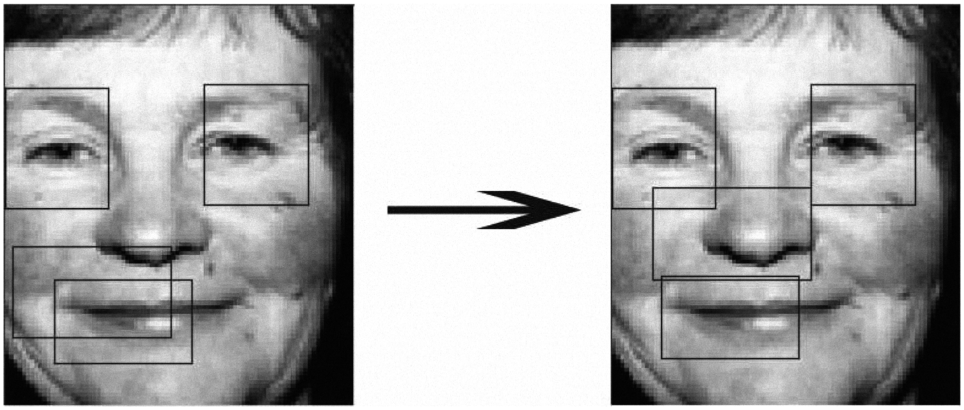


Fig. 5. The image on the left shows location of features obtained with pure scanning, and the image on the right shows location of features using the attention mechanism.

[4] C. J. C. Burges. A tutorial on Support Vector Machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[5] V. Cantoni and L. Lombardi. Hierarchical architectures for computer vision. In *Euromicro Workshop on Parallel and Distributed Processing*, pages 392–398, 1995.

[6] N. Christianini and J. Shawe-Tayor. *An introduction to Support Vector Machines*. The Press Syndicate of the University of Cambridge, 2000.

[7] C. Dillon and T. Caelli. Learning image annotation: The CITE system. *Journal of Computer Vision Research*, 1:89–122, 1998.

[8] F. Gobet, P. C. R. Lane, S. J. Croker, P. C-H. Cheng, G. Jones, I. Oliver, and J. M. Pine. Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, 5:236–243, 2001.

[9] J. W. Han, P. C. R. Lane, N. Davey, and Y. Sun. Comparing the performance of single-layer and two-layer support vector machines on face detection. In *Proceedings of The Seventh UK Workshop on Computational Intelligence*, 2007.

[10] J.W. Han, P.C.R. Lane, N. Davey, and Y. Sun. A dual-layer model of high-level perception. In R.M. French and E. Thomas, editors, *From Associations to Rules: Connectionist Models of Behavior and Cognition, Proceedings of the Tenth Neural Computation and Psychology Workshop*, volume 17, pages 139–149. (World Scientific) Progress in Neural Processing, 2007.

[11] M. A. Hearst. Support vector machines. *IEEE Intelligent Systems*, 28:18–28, 1998.

[12] B. Heisele, T. Serre, M. Pontil, and T. Poggio. Component-based face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 657–662, 2001.

[13] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. Categorization by learning and combining object parts. In *Advances in Neural Information Processing Systems*, volume 2, pages 1239–1245, 2001.

[14] J. Huang, V. Blanz, and B. Heisele. Face recognition using component-based SVM classification and morphable models. In *Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines*, pages 334–341, 2002.

[15] P. C. R. Lane, F. Gobet, and R. Ll. Smith. Attention mechanisms in the CHREST cognitive architecture. In L. Paletta and J. K. Tsotsos, editors, *Proceedings of the Fifth International Workshop on Attention in Cognitive Science*, volume LNAI 5395, pages 183–196. Berlin: Springer-Verlag, GmbH, 2009.

[16] L. G. Shapiro and G. C. Stockman. *Computer Vision*. New Jersey, USA: Prentice Hall, Inc., 2001.

[17] J. K. Tsotsos, S. M. Culhane, W. Y. K. Wai, Y. Lai, N. Davis, and F. Nuflo. Modeling visual attention via selective tuning. *Artificial Intelligence*, 78:507–545, 1995.

[18] J. M. Wolfe. Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review*, 1:202–238, 1994.