Comparing the Performance of Single-Layer and Two-Layer Support Vector Machines on Face Detection

Ji Wan Han, Peter C. R. Lane, Neil Davey, and Yi Sun
School of Computer Science
University of Hertfordshire
Hatfield, AL10 9AB
UK
j.w.han, p.c.lane, n.davey, y.2.sun@herts.ac.uk

Abstract

Face detection is a vibrant research branch of computer vision. Methods of detecting faces fall into two categories: global and component-based. In this paper, we compare these two approaches by applying a single-layer and a dual-layer support vector machine classifier to detect faces from images. Experiments suggest that the single-layer classifier has better performance on detecting faces with big attitude extremity. But the dual-layer classifier has equivalent performance on detecting frontal faces and has more generality on different databases.

1 Introduction

Face detection is used to determine the locations and sizes of human faces in given images. The introduction of *machine learning* has greatly improved the performance of face detection systems, with global and component-based approaches being popular.

Support Vector Machines (SVMs) [4] are one of the most powerful algorithms for classification. The application of single-layer SVMs is straightforward. It directly takes images as inputs, outputting a classification. Dual-layer SVMs have two layers respectively for detecting features and object structures. They can comprise several different classifiers in each layer and are organized in an hierarchy.

We construct a dual-layer SVM classifier and introduce a feature map to convey relations among features detected in the first layer. For the dual-layer classifier, there are two main challenges when constructing it. One is to create training sets for both the first and second layer. Another is to effectively locate features in the image for the first layer. We compared its performance on face detection with a single-layer SVM classifier.

This paper is organized as follows: Section 2 gives a background on relevant concepts and work. Section 3 introduces the structure of the dual-layer SVM classifier. In section 4, we give a brief introduction to the databases used in the comparison. Section 5 describes the experiments and compares two classifiers. Sections 6 summarizes the conclusions of this comparison and suggests future directions.

2 Background

Generating a correct and effective representation of the real world is an important research task for computer vision. The real world is composed of different concrete objects which are distinguished by features they possess. A key element of many complex systems that allows us to comprehend, analyze, and build such systems is their decomposition into an hierarchy [13]. Such hierarchical representations have become an important method to solve problems in computer vision [2].

Dillon *et al.* [5] 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 an 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.

Object detection is an indispensable process of scene understanding. Face detection is one of the challenging tasks of object detection. Much effort has been done to improve the performance of face detection systems [14]. SVMs offer the chance for real world applications on object detection to deliver high performance. Using single-layer SVMs to detect faces has gained remarkable success. Osuna *et al.* [7] applied SVMs to face detection, later extending this application to a real-time system. El-Naqa *et al.* [6] applied SVMs to detect microcalcifications in mammograms.

Recently, multi-layer SVMs, some of them adopting an hierarchical structure, have become popular. Heisele *et al.* [9] presented a dual-layer SVM algorithm learning discriminative components (features) of objects. In this algorithm, component-based face classifiers were combined in the second stage to yield an hierarchical SVM classifier. On the first layer, the component classifiers independently detected components of the face. On the second layer, the combination classifier performed the detection of the face based on the output of the component classifiers. They also compared the performance between component-based (dual-layer) and global (single-layer) approaches [8]. Huang *et al.* [11] combined component-based detection and 3D morphable models to detect faces. Their experiments showed the potential of dual-layer SVMs on pose and illumination invariance.

3 Dual-Layer SVM Classifier

3.1 Support Vector Machines

SVMs, based on the principle of structural risk minimization, are an excellent machine learning algorithm and give the promise of learning highly accurate models on large feature spaces [4, 7, 12]. The basic idea of SVMs is to find a hyperplane, which can classify two classes of data correctly, by maximizing the distance between two classes of data and the hyperplane. See Figure 1. SVMs have some useful qualities in particular they are guaranteed to find a global minimum solution.

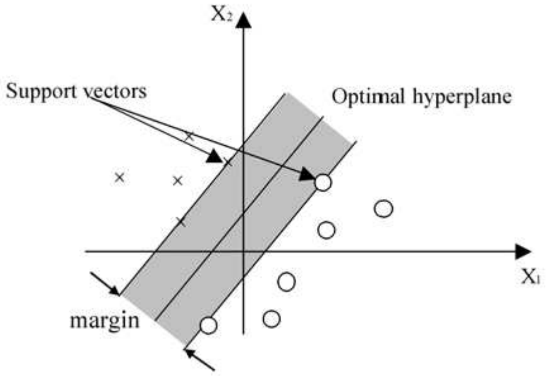

Figure 1: The margin between two classes of data and support vectors (adapted from Figure 2 in [6]).

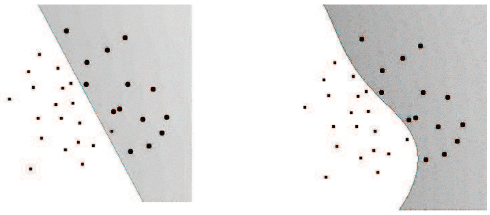

Figure 2: Classification using linear kernel (left) and RBF kernel (right).

The kernel function plays a key role for SVMs to solve real world applications because many such applications are not linearly separable. It maps input data into a high-dimensional feature space. In this space, the mapped data could be linearly separable or have better separability. RBF(radial basis function) kernel is commonly considered as the most powerful one. Linear kernel is best understood and simplest to apply. Figure 2 shows the difference between them on generating the classification boundaries.

3.2 System Overview

The hierarchical structure of objects is shown at the left top in Figure 3. The face hierarchy can be expanded to other scenes, such as the office or the natural environment. Figure 4 is the basic structure of dual-layer SVM classifier for detecting faces.

The first layer detects features from the target object. It comprises four feature detection experts for two eyes, nose, and mouth. Two eye experts are trained with the same training sets. This is because the difference between left and right eyes is very small. They are discriminated by different searching sequences in the potential object area. For example, the classifier searched for the left eye from right to left and top to bottom, but for the right eye it searched from left to right and top to bottom. See Figure 5.

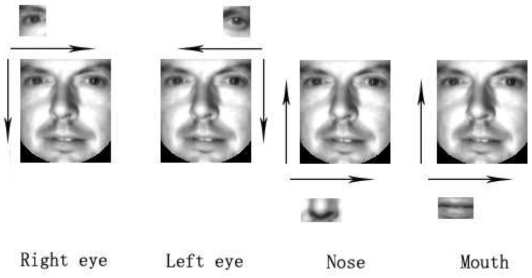

Figure 5: Grid movement directions.

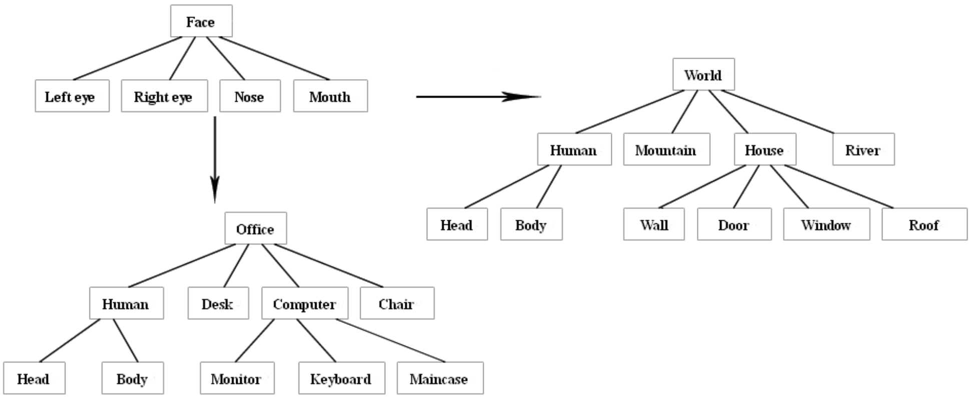

Figure 3: Hierarchical structure of objects.

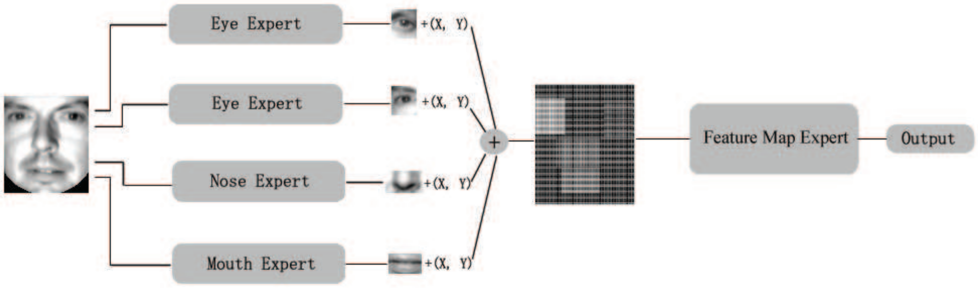

Figure 4: Structure of the dual-layer SVMs.

By detecting these features, the dual-layer SVMs get positions for each feature and give each of them an index. Using feature coordinates and indices, a feature map is created. Figure 6 is a simplified feature map for a face. Each feature in the face map has a label: left eye is 4, right eye 1, nose 3 and mouth 2. The background is 0. Each number in the map corresponds to a pixel. So the map matrix has the same size as the image. There can be overlap between two different features. The SVMs can learn this because the feature detection is always in the same sequence: right eye → mouth → nose → left eye. This ensures that the overlap between two features is always consistent.

The second layer is used to identify the structure of how these features compose the target object. Its input is the feature map and the output is a class index, 1 for face, 0 for non face. The dual-layer SVMs have a similar structure to that introduced in [8, 11], with an important difference being the information given to the second layer. Huang *et al.* provided the second layer with details of the features and their *coordinates*. Instead, we introduce a *feature map* to act as a medium to convey object structure directly; the feature map has the advantage of preserving all relative positions between features explicitly.

```
00000000      Key:
01100440
01100440      0 is no feature
00000000      1 for right eye
00033000      2 for mouth
00033000      3 for nose
00000000      4 for left eye
00022000
```

Figure 6: Simplified feature map for a face; actual size is 84x96 pixels.

4 Data Used

We used the following databases to create positive samples: Database of Faces from AT&T Laboratories, Cambridge; Japanese Female Facial Expression (JAFFE) Database; and The Psychological Image Collection at Stirling (PICS). The following databases were used to create negative samples: BEV1 Dataset and Caltech Database. Harvard Face Database was used to test the generality of classifiers.

We cut faces from these datasets and adjusted the face image size to a consistent size of 84×96 pixels. The proportion of faces to the image size is consistent. We randomly selected 201 face images for training, 178 for test, and 451 non-face images for training, 227 for test. Especially, we selected 78 frontal faces in test set, which have attitude extremity no more than 15^0 in all directions.

For the training of feature experts, we located the features image by image manually from the training images described above. Feature sizes were decided by their geometry characteristics. For the eye, every eyebrow and eye from all images should be contained within the feature area. So 25×29 pixels is the size which contains the biggest eyebrow and eye pair. The size of nose, 38×22 pixels, is decided by the nose width and the distance from bridge of a nose to the very bottom of nose. The width and the height of mouth decide the size, 33×20 pixels, of mouth samples.

Creating a negative dataset is challenging, because it is difficult to get key negative images, which are close to positive images; learning is more efficient when the negative images are 'close' to the positive images. We used an iterative process to create a negative dataset, allowing the learning algorithm itself to locate the 'near-misses':

1. Select some non-feature images, which form the basic negative samples. These have a similar average grey scale to the faces in the training set.

2. Select some negative examples from the feature images to act as negative samples for other features. For example, noses and mouths as negative samples for eyes.

3. Use the dataset to train classifier. Then use the trained system to recognize features from face dataset. Find the falsely detected features and add them to the negative dataset. At the same time, add those not detected features to positive training set. We may delete some feature images or repeatedly cut features from one face image.

4. Repeat step 3 until performance acceptable on training set.

Table 1 shows details of feature training sets. Figures 7 and Figures 8 show some samples in feature datasets.

Feature	Number of Images		Image size
	Positive	Negative	
Eye	344	1865	25×29
Nose	203	1250	38×22
Mouth	201	1634	33×20

Table 1: Feature Datasets details

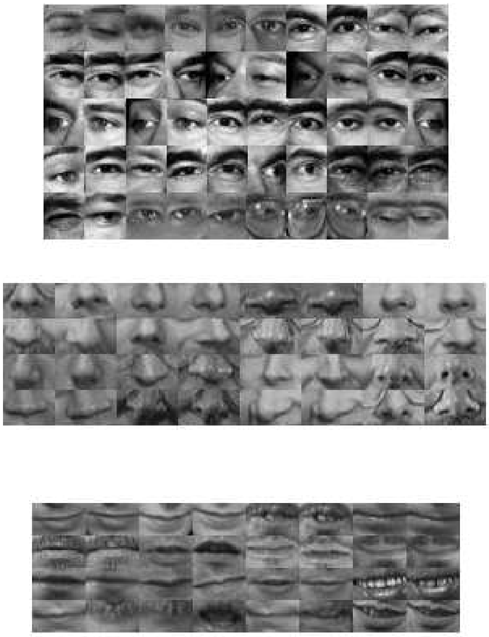

Figure 7: Part of feature datasets

For the second layer, positive and negative feature maps were created artificially by giving each feature different situations in the feature map. In order to create positive dataset, different tolerance was given to each feature which allowed each feature to change its situation in a certain area. The feature map was treated as a negative sample if some features were outside of its positive area.

5 Experiments

The SVM package we are using is *LibSVM*, which was developed by Chang and Lin [3]. Firstly, we trained classifiers and run them on test sets. This will be described in section 5.1 and 5.2. Secondly, we compared two classifiers by using test data, especially frontal faces. Then used Harvard database to test the ability to generalize to a completely different database.

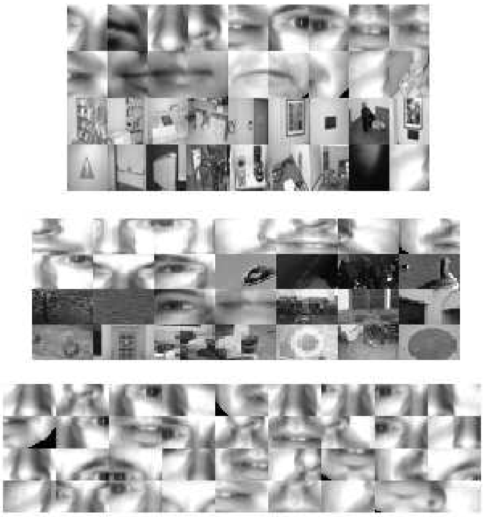

Figure 8: Part of negative datasets. From top to bottom: negative eyes, negative noses, negative mouths

5.1 Performance of Single-Layer SVM Classifier on Test Data

Method. The single-layer SVMs directly take face images as inputs. 201 face images and 451 non-face images in the training dataset were used to train the SVMs. We used grid method to search the best value of two parameters, the cost C and γ [10], to optimize a kernel. This method searches parameters based on a trying all method, which means searching all the possible combinations and all the ranges of different parameters specified by user and the one with the best cross-validation accuracy will be chosen. RBF kernel has better performance than other kernels. Using the grid search, the default parameters of RBF kernel specified by *LibSVM* gave the best performance. The 5-fold cross validation showed that the accuracy was 96.49%.

Results. The single-layer SVMs displayed a good performance. For the 178 test faces, 172 were successfully detected and 6 were missed. The success rate was 96.63%. For the 227 non-faces, 212 were successfully detected as non-faces and 15 were misdetected as faces. The success rate was 93.39%. See Table 2.

Set Name	Classification result			Correct(%)
	C	I	T	
Faces	172	6	178	96.63
Non faces	212	15	227	93.39
Total	384	21	405	94.81

Table 2: Results of experiment with single-layer SVMs (C: Correct, I: Incorrect, T:Total).

Discussion. The single-layer SVMs support the excellence of Support Vector Machines, displaying a good performance on detecting faces. It is convenient to create a single-layer SVM detector. This can be useful in scene understanding system to locate objects.

5.2 Performance of Dual-Layer SVM Classifier on Test Data

The first layer of the dual-layer SVM classifier comprises 4 independent feature experts. The input of the second layer is from the first layer, so the system's performance depends on the first layer deeply. Firstly, we tested the performance of the first layer, then did the joint experiment.

5.2.1 Performance of the First Layer on Feature Detection

The first layer detects independent features from faces. 4 feature SVM experts were trained on datasets manually created from those 201 face images and 451 non-face images.

Method. When detecting different features, a grid with the same size as the feature scanned the whole target image and passed image data in it to the expert. When the expert found a feature, it would give the feature a feature index. The grid moved according to certain rules. See Figure 5 and previous description.

We used grid search to find the best parameters to optimize kernels. We compared several kernels including linear, polynomial, RBF and sigmoid tanh. RBF kernel performed the best with the default parameters provided by *LibSVM*. During 5-fold cross validation, the average accuracy on the eye dataset was 92.13%, nose dataset 92.91% and mouth dataset 91.61%.

Results. Classifiers can find features well from frontal test faces but not so well on the side faces. For non-faces, there were many false detections. Figure 9 is a sample of detecting different features from face. Note that each feature may be recognized several times. The lower right image is a detected face in which only the first detected features were taken to form the face.

Discussion. The first layer SVMs play an important role for the dual-layer SVMs. They can detect all features in the image but there were many false positive detections. The light source extremity and the attitudes of faces had influence on the performance of this layer. The false positive detections were increasing with the light and attitude extremity. This is damaging the detection performance of the second layer SVMs.

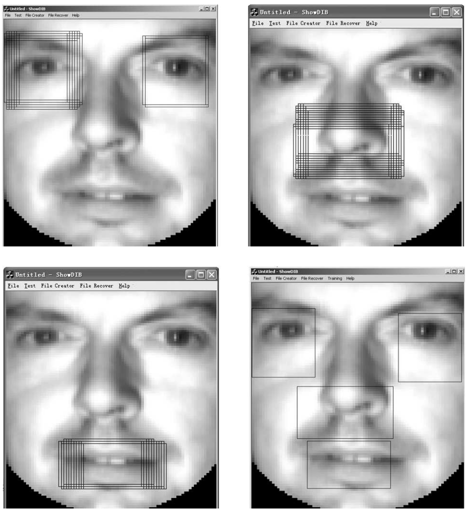

Figure 9: Detection of features

Set Name	Classification result			Correct(%)
	C	I	T	
Faces	147	31	178	82.58
Non faces	176	51	227	77.53
Total	323	82	405	79.75

Table 3: Results of dual-layer SVMs (C: Correct, I: Incorrect, T:Total).

The situation can be improved by improving the datasets, such as adding more key negative samples and adjusting the positive samples.

5.2.2 Joint Performance of Dual-layer SVMs

Method. In this experiment, we connected the first layer and the second layer together to detect faces in images. After the first layer SVMs completes feature detection, they output detected features to form a feature map, which is the input for the second layer. The second layer judges whether the feature map fed to it is a face.

Results. The dual-layer SVM classifier gave an acceptable performance: 323 images were correctly detected from 405 images. The overall correct rate was 79.75%. See Table 3.

Discussion. There were more than one samples detected for each feature in the first layer. Only the first detected sample was taken, so the situations of them have decisive influence on feature maps. See Figure 9. The second layer detection depends strongly on the first layer. The main problem happening in this stage was the structure of face. Figure 10 is the ROC curves of dual-layer SVMs and single-layer SVMs. We got it by giving different cost tradeoffs when training classifiers. The ROC curves, tables 2, and 3 suggest that the dual-layer SVM classifier is not as good as the single-layer one. The reason is there are many false positive detections. When one of these false detections was at the first position to be detected, the feature map went wrong. In order to overcome this problem, we added more negative samples into first level training sets. At the same time, we also improved feature map training set by manually adding some key positive and negative training samples and gave slightly bigger tolerance for positive face map samples.

5.3 Comparison Using Frontal Faces

We selected 78 frontal faces from test face set. They have the least attitude and illumination extremity. The single-layer classifier detected 74 correct out of 78 faces. The correct rate is 94.87%. The dual-layer classifier detected 72 correctly. The correct rate is 92.31%. The single and dual-layer classifiers have equivalent performance on frontal faces.

5.4 Comparison Using Different Database

In the previous experiments, both training sets and test sets were from the same database. So samples in this database have some common ground. In order to compare the generality of two algorithms, we used another database, Harvard database, to test them. There are 5 sets in Harvard database. Set 1 has the smallest illumination extremity. And the extremity is increasing from set 2 to set 5. In set 5, some features are even dark. Table 4 shows the experiment results. The dual-layer classifier clearly gave better performance than single-layer classifier on different database.

Set Name	Correct Rate (%)	
	Single-Layer	Dual-Layer
Set1	70	100
Set2	52.22	81.11
Set3	20	56.92
Set4	8.24	58.24
Set5	1	47.69

Table 4: Experiments on Harvard database.

6 Conclusion and Future Work

We presented a dual-layer SVM classifier and compared it with single-layer SVM classifier on

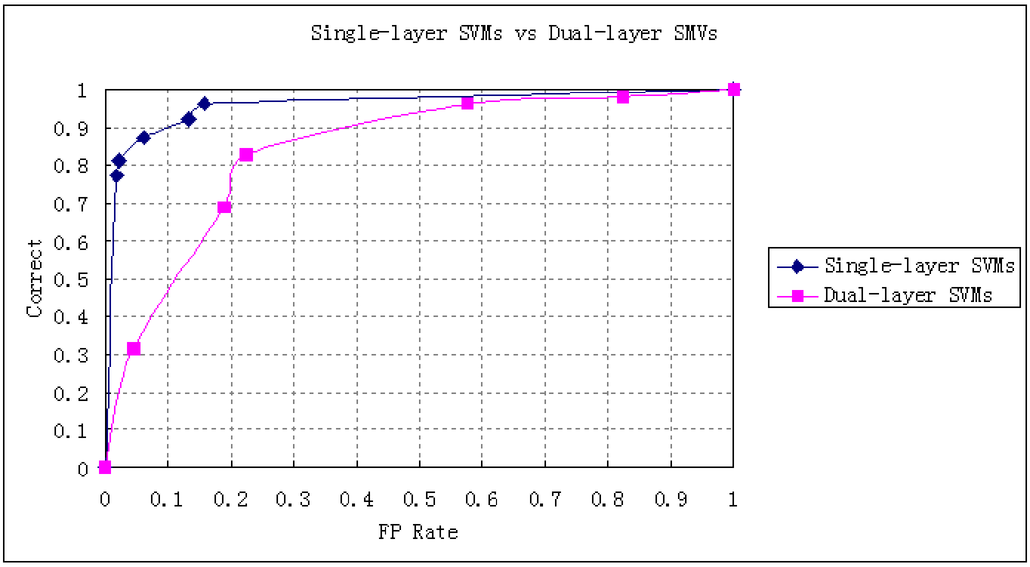

Figure 10: ROC curves of the single-layer SVMs and dual-layer SVMs

face detection. The single-layer SVM classifier has higher performance on the same database from which the training sets and test sets were constructed. The dual-layer SVM classifier has an equivalent performance on detecting frontal faces. It depends deeply on its first layer. This layer gets some false positive detections because it matches many classifications of the background. In order to reduce false detection caused by these false positive detections, we did not use feature maps with difficult attitudes to train the second layer. This influenced the performance of the dual-layer SVM classifier on faces with difficult attitudes.

But the dual-layer SVMs outperformed at detecting faces in different databases. It displayed higher generality. Therefore, the benefits of using dual-layer SVM classifier include separately identified features, structural information of object, a more flexible architecture, and higher generality. The dual-layer SVM classifier uses hierarchical structure and it is an attractive option for scene understanding.

We are improving the dual-layer SVM classifier in the following directions:

Finding features. Currently we find features by scanning through the image. We are adding some algorithms, such as autoassociation, to improve the searching speed by giving indication of probable location of features.

Constructing features. We arbitrarily selected features by specifying the feature size, location and type. It seems to be working well but we can not make sure these features are best for the classification task.

Decreasing uncertainty of detection. The first layer introduces many false detections. This can be improved by enhancing the feature training sets.

References

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

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

[3] C. C. Chang and C. J. Lin. Libsvm – a library for support vector machines. http://www.csie.ntu.edu.tw/ cjlin/libsvm/.

[4] N. Cristianini and J. Shawe-Tayor. *An Introduction to Support Vector Machines.* The Press Syndicate of The University of Cambridge, 2000.

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

[6] I. El-Naqa, Y. Y. Yang, M. N. Wernick, N. P. Galatsanos, and R. Nishikawa. Support vector machine learning for detection of microcalcifications in mammograms. In *IEEE Transactions on Medical Imaging*, volume 21, pages 1552–1563. 2002.

[7] M. A. Hearst. Support vector machines. *IEEE Intelligent Systems*, 13:18–28, 1998.

[8] B. Heisele, P. Ho, J. Wu, and T. Poggio. Face recognition: Component-based versus global approaches. *Computer Vision and Image Understanding*, 91:6–21, 2003.

[9] 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.

[10] C. W. Hsu, C. C. Chang, and C. J. Lin. A practical guide to support vector classification. citeseer.ist.psu.edu/689242.html.

[11] J. Huang, V. Blanz, and B. Heisele. Face recognition using component-based SVM classification and morphable models. In *SVM 2002*, pages 334–341, 2002.

[12] E. Mjolsness and D. DeDoste. Machine learning for science: State of the art and future prospects. *Science*, 293:2051–2055, 2001.

[13] H. A. Simon. *The sciences of the artificial.* Cambridge (Mass.) ; London : M.I.T. Press, 1969.

[14] M. H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 24, pages 34–58. 2002.