Face Recognition Based on Deep Convolutional Support Vector Machine with Bottleneck Attention

Chi Jing, Zhang Haopeng, Chin Kim On*, Ervin Gubin Moung, Patricia Anthony

***Abstract*—A Deep Convolutional Support Vector Machine with Bottleneck Attention module (DCSVM-BAM) to improve the accuracy of recognising face images is proposed and discussed in this article. Although Support Vector Machine (SVM) has achieved excellent results in face recognition, its performance is still limited by its feature extraction ability. The proposed DCSVM-BAM method consists of two processes, feature extraction and feature classification. The face images' features are first extracted with Bottleneck Attention-based Deep Convolutional Neural Network. Then these images are mapped from low-dimensional to high-dimensional spaces. We utilised Soft-Margin SVM to perform classification in the high-dimensional space. Experimental results on eight public face datasets demonstrated that the DCSVM-BAM had achieved better results than Convolutional Neural Networks and SVM. The DCSVM-BAM has achieved the best accuracy precision, recall, specificity, FPR, and F1_measure in six of the eight public face datasets. Meanwhile, the proposed DCSVM-BAM was compared with DeepID in accuracy, precision, recall, specificity, FPR, and F1_measure. The results have shown that the proposed DCSVM-BAM performed better than DeepID.**

***Index Terms*—Bottleneck Attention, Convolutional Neural Network, Deep Learning, Face Recognition, Support Vector Machine**

I. Introduction

Research interest in face recognition has increased over the years. Face recognition is a process of recognising various faces of people via a visual system. It is a valuable technique in biometric security systems for access control and video surveillance. It has become an essential human-computer interaction tool that can be used in social networks, such as Facebook [1][2][3]. A face recognition system should be capable of handling various changes in face images. However, the variations in the same-face photographs brought on by lighting and viewing angle are virtually always more significant than those brought on by variations in the face's identity [4]. This raises two questions related to face recognition. How can facial features be extracted when a face's image changes due to changes in illumination, angle of view, and expression? How can we select useful features that can manage all potential alterations to categorise the brand-new face images?

Conventional face recognition methods are based on geometric features [5]. Usually, the facial features such as eyes, nose, mouth, and chin are involved [6][7][8]. These methods have two advantages; efficiently reduces the number of features, and insensitive to changes in lighting and viewing angle. Unfortunately, these methods rely heavily on existing feature extraction algorithms that are not reliable [9].

Relying on a good theoretical foundation and sophisticated solving procedure, Support Vector Machine (SVM) has achieved good results in the field of face recognition [10]. For example, [11] has proposed SVM-based algorithm for face recognition application and the accuracy has achieved 97% on the FERET dataset. [12] used eigenfaces to extract features, and then performed face recognition using linear SVM and binary tree classification, which has achieved an average accuracy of 97% on the ORL dataset. [13] has presented a local face recognition algorithm which is based on facial features. The facial components are initially located, their features are extracted, and then they are combined into a single eigenvector, which is subsequently subjected to a final SVM classification. The performance of SVMs is influenced by the kernel function, despite the increased accuracy of face recognition with the aforementioned techniques. Choosing the right kernel function might be challenging because they typically produce varied results.

The Convolutional Neural Networks (CNNs) are a type of neural networks, whose effectiveness has been proven in the fields of image recognition and classification [14][15][16]. CNNs, which take into consideration the two-dimensional structure of images and adopt the weight sharing and local receptive field strategies, obtain learnable functions by combining several linear and nonlinear operators [17][18]. The hidden layer in a CNN can be regarded as the mapping of original input into different dimensional spaces. By

Manuscript received November 11, 2021; revised September 19, 2022. This work was supported in part by the University Malaysia Sabah under SDN Grant SDN51-2019, as well as the Handan Municipal Bureau of Science and Technology, Hebei Province, China under Grant 21422031252.

Chi Jing is an Associate Professor in School Information and Electrical Engineering, Hebei University of Engineering, Handan, Hebei 056038, China, and a PhD candidate of University Malaysia Sabah as well, Kinabalu, Sabha 88400, Malaysia (e-mail: chijing@hebeu.edu.cn).

Zhang Haopeng is a postgraduate student at Hebei University of Engineering, Handan, Hebei 056038, China (e-mail: zhanghaopengyyds@163.com).

Chin Kim On is an Associate Professor in Faculty of Computing and Informatics, University Malaysia Sabah, Kinabalu, Sabah 88400, Malaysia (phone: 60-168301621; e-mail: kimonchin@ums.edu.my).

Ervin Gubin Moung is a senior lecturer in the Faculty of Computing and Informatics, University Malaysia Sabah, Kinabalu, Sabha 88400, Malaysia (e-mail: ervin@ums.edu.my).

Patricia Anthony is an Associate Professor in the Faculty of Environment, Society and Design at Lincoln University, New Zealand (e-mail: Patricia.Anthony@lincoln.ac.nz).

mapping face data from low to high-dimensional spaces, features used for recognising faces can be learned [19]. With good feature extraction ability, the CNN was able to achieve remarkable results in the face recognition field [20][21][22]. Although CNNs are efficient in processing face data, they cannot effectively remove irrelevant features. Hence, a method for filtering features in convolution layer is needed to find the most appropriate feature location for face recognition, thereby achieving effective recognition of face data.

The use of CNNs allows automatic extraction of face image features, which avoids the selection of kernel function. However, the feature information in images not only contains content needing recognition, but also includes content that is detrimental to image classification, such as background and noise. Hence, accurately distinguishing the content from images is particularly important. The Attention Mechanism (AM) addresses this issue by [23][24][25] has enabled the model to focus on the information needing recognition in images while reducing information that plays a minor role in image recognition. During face recognition, AM helps the model emphasise the distinguishable parts of face, such as human eyes, nose, and mouth, so that it can extract more distinguishing features to improve the recognition accuracy. Among the Attention methods, the Bottleneck Attention (BA) is an effective one. It first infers an intermediate feature map in both the channel and spatial branches, and then combines these two feature maps to get the attention map. Using this approach, it can generate a hierarchical attention at bottlenecks [26].

Based on the foregoing analysis, this study proposes a Deep Convolutional Support Vector Machine with Bottleneck Attention Module (DCSVM-BAM). This method initially uses the attention-based CNN to filter the facial features and extracts the useful features for face recognition. Then, SVM is used to recognise the filtered features, followed by a final optimisation of the entire method using mini-batch gradient descent. When compared to conventional methods, the proposed method can retain as many distinguishing features as possible while removing features that are unnecessary for face recognition. Moreover, SVM has enabled more effective classification of these features, thereby improving the accuracy of face recognition.

The rest of this paper is organised as follows. In Section II, we describe CNN, SVM, and the Bottleneck Attention Module (BAM). The DCSVM-BAM and its learning algorithm are explained in Section III. The experimental data are presented and discussed in Section IV. Section V makes conclusions of the writing.

II. Related Works

Three types of models, namely CNN, SVM and BAM works are discussed.

A. *Convolutional Neural Networks*

CNN, proposed by [27], is a special type of neural network used in image recognition. It is made up of multiple neurons that process input, use learnable weights and biases to different picture features, and perform convolutional operations [28]. A standard CNN architecture generally comprises of convolution, pooling, nonlinear and fully connected layers [29]. Convolution layers are responsible for implementing the core building blocks of CNNs, which perform most of the computationally heavy tasks. Their main objective is to extract features from the input image data. Convolution maintains the inter-pixel spatial relationship by getting the features from small sizes of input images. Usually, we convolve the input image with a group of learnable neurons, thereby producing a feature mapping or activation mapping in the output image. Afterward, the feature mapping is fed into the next convolution layer as input data. Although the pooling layer reduces the dimensionality of each feature map, it still contains the most important information. This layer achieves both better generalisation and faster convergence, as well as good robustness to translation and distortion, which is usually placed between convolution layers. CNN allows a choice of whether a nonlinear layer is added after each convolution layer for nonlinear operation. The nonlinear layer implements an element-by-element operation, signifying that it operates on a pixel-by-pixel basis. It aims to add nonlinear operations for adapting to more complex scenarios. Meanwhile, the fully connected layer achieves the classification function by classifying high-dimensional features into various classes.

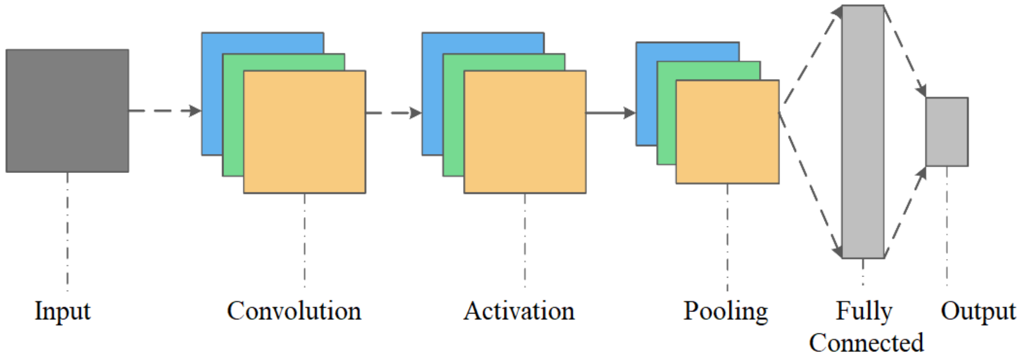

Fig. 1. The structure of generic CNN model. It shows general CNN structure that consists of input image data, convolution layers, activation function layers, such as the nonlinear layer, pooling layer, fully connected layer, and the output class. The input is subjected to convolution operation to extract features, and then nonlinear factors are added through the activation function, so that the model can handle nonlinear problems. Afterwards, important feature information is retained through the pooling layer, and the dimensionality is reduced. The input of the final fully connected layer comes from the previous expansion of feature maps, and the classified classes are obtained through the output of this step.

B. Support Vector Machine

SVM was proposed by [30], based on the Vapnik–Chervonenkis (VC) dimension theory and the structural risk minimisation principle [31]. A SVM can perform very well in solving nonlinear and high dimensional pattern recognition problems. This powerful method has been widely applied in a diversity of fields, such as image retrieval [32], character recognition [33], object recognition [34], etc.

Compared to other types of classifiers, the SVM is considered a machine learning classifier that performs well in high-dimensional spaces. The main idea behind using SVM is to find the optimal hyperplane separating the feature space with a supervised learning algorithm [35]. The SVM can generate a high-dimensional space during training, in order to classify the training dataset into different classes. Although SVM was originally developed for dichotomous classification, it can be easily extended to multiclass classification problems. This is achieved by dividing a multiclass problem into multiple dichotomous problems where the outputs of all sub-binary classifiers are combined to generate the predicted classification of samples. In multiclass SVM, there are two main methods. The first method is called "one against one" [36], which involves creating a classifier for each pair of classes and combining binary classifiers by selecting the class with the most votes to form a multiclass classifier. Thus, the n(n-1)/2 binary SVM classifiers are required, each of which is trained on two samples of corresponding categories. The second method is called "one against all" [37], which takes into consideration all data classes in one optimisation problem.

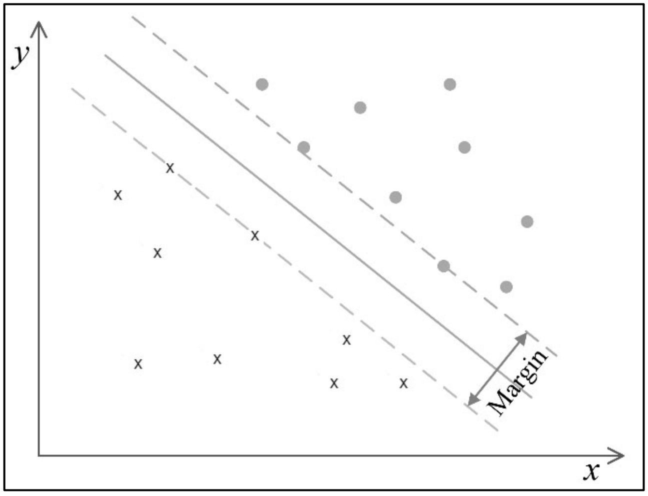

Fig. 2. The structure of SVM model. The cross symbols represent the negative classes, the dots represent the positive classes, and the straight-line segment indicates the separating hyperplane used to classify data into two classes. The dotted lines indicate the margin boundary of the SVM, whereas Margin represents the distance between the two imaginary lines. The soft-margin SVM allows a few sample points to be within the margin.

SVM is a dichotomous model. Its learning objective is to find the hyperplane with the largest margin in the feature space, and divide the data into two classes, i.e., positive and negative classes. The original SVM model is a quadratic optimisation problem. Given a training dataset as

$$S = \{(x^l, y^l) | x^l \in R^p, y^l \in \{+1, -1\}, 1 \le l \le N\} \quad (1)$$

where S stands for the training set, $\boldsymbol{x}^l$ is the l-th training sample, and y^l is its label, R represents the sample space, and N means the number of samples.

The hyperplane is defined as $Wx+b=0$, where W is a vector of parameters and b represents the bias parameter. The hard-margin maximisation kernel-free learning SVM is described as follows

$$\begin{gathered}\min_{\boldsymbol{W},\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{W}\|^2 \\ s.t.\ y^l(\boldsymbol{W}\boldsymbol{x}^l + \boldsymbol{b}) \ge 1,\ 1 \le l \le N.\end{gathered} \quad (2)$$

Further, the linear loss soft-margin SVM can be defined as:

$$\begin{gathered}\min_{\boldsymbol{W},\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{W}\|^2 + C\sum_{l=1}^{N}\xi \\ s.t.\ y^l(\boldsymbol{W}\boldsymbol{x}^l + \boldsymbol{b}) \ge 1-\xi,\ \xi \ge 0,\ 1 \le l \le N\end{gathered} \quad (3)$$

where ξ denotes the non-negative slack variable, and C denotes the penalty factor.

Similarly, the quadratic loss soft-margin model of SVM can be expressed as:

$$\begin{gathered}\min_{\boldsymbol{W},\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{W}\|^2 + C\sum_{l=1}^{N}\xi^2 \\ s.t.\ y^l(\boldsymbol{W}\boldsymbol{x}^l + \boldsymbol{b}) \ge 1-\xi,\ \xi \ge 0,\ 1 \le l \le N.\end{gathered} \quad (4)$$

It can be easily proven that the quadratic loss soft-margin of SVM is equivalent to the following unconstrained minimisation problem:

$$\min_{w,b} \frac{1}{2}\|\boldsymbol{W}\|^2 + C\sum_{l=1}^{N}[max(1 - y^l(\mathbf{W}x^l + \boldsymbol{b}), 0)]^2 . \quad (5)$$

C. Bottleneck Attention Mechanism

AM originates from the study of human vision and applies greater weight of attention to the more interesting aspects of features. It is initially used in machine translation, and some studies show that it has been used successfully in various fields including natural language processing, statistical learning, speech, computer vision, etc [38,39,40]. In face recognition tasks, AM helps the model emphasise on the distinguishable parts of face, so that it can extract more distinguishing features to improve the recognition accuracy. Among the AM, the BAM is an effective one. BAM was proposed by [26]. It is a simple and effective attention model, which can be integrated with any CNNs and can be placed at every bottleneck of models where the down-sampling of feature maps occurs. The structure of BAM model is depicted in Fig. 3.

Based on Fig. 3, the channel attention generates a channel attention map $M_C(F) \in \mathbb{R}^C$, where F represents an intermediate feature map as input. To extract the specific facial features contained in each channel, the interrelation between various channels is exploited. Through the global average pooling of feature map $F \in \mathbb{R}^{C\times H\times W}$, where C means the number of channels, H and W respectively represent the height and width of F, the feature maps in each channel are aggregated and the channel vector $F_C \in \mathbb{R}^{C\times 1\times 1}$ is generated. This vector soft-encodes the global information in each channel. In this study, a multilayer perceptron with a single hidden layer is used to calculate the attention between channels from the channel vector F_C. To save parameter overhead, the hidden activation size is set to $\mathbb{R}^{C/r\times1\times1}$, where r denotes the reduction rate. A batch normalization (BN) layer [31] is added after the multilayer perceptron (MLP) to adjust the proportion of spatial branch output. The channel attention is calculated using formula shown in (6).

$$\begin{aligned} M_c(F) &= BN\left(MLP\big(AvgPool(F)\big)\right) \\ &= BN(W_1(W_0AvgPool(F)+b_0)+b_1) \end{aligned} \tag{6}$$

where $W_0\in R^{C/r\times C}, W_1\in R^{C\times C/r}$ and $b_0\in R^{C/r}, b_1\in R^C$ respectively denote the weights and biases of single-hidden-layer multilayer perceptron.

The attention part of spatial channel generates a spatial attention map $M_s(F)\in\mathbb{R}^{H\times W}$, which enhances or inhibits the features at different spatial positions. Extensive literature has demonstrated that the utilisation of contextual information is crucial to identifying spatial locations [41][42][43]. The receptive field is efficiently expanded through hole convolution [37]. The "bottleneck architecture" proposed by ResNet [15] is adopted for spatial branches, which saves not only the number of parameters, but also the computational overhead. Specifically, 1×1 convolution is performed on the feature map $F\in\mathbb{R}^{C\times H\times W}$ to reduce the dimensionality to $\mathbb{R}^{C/r\times H\times W}$. For the sake of simplicity, the reduction rate r used in the spatial attention part is consistent with the channel attention part. Next, the context information is effectively utilised through two 3×3 dilated convolutions. As a last step, 1×1 convolution is used to again transform the feature into a spatial attention feature map with $\mathbb{R}^{1\times H\times W}$, followed by addition of a BN layer. In brief, the computational formula for spatial attention is shown as in (7).

$$M_s(F) = BN\left(f_3^{1\times1}\left(f_2^{3\times3}\left(f_1^{3\times3}\left(f_0^{1\times1}(F)\right)\right)\right)\right) \tag{7}$$

where *f* means convolution operation, 1×1 denotes filter size, and so on.

Finally, the channel attention is combined with the spatial attention. After adjusting the outputs of channel and spatial attention mechanisms to the input feature map size, the attention part with the sigmoid function on the output activation is then obtained. The computational formula for final output refined feature map F' is as follows:

$$F' = F + F\otimes M(F) \tag{8}$$

$$M(F) = \sigma\big(M_c(F)+M_s(F)\big) \tag{9}$$

where $\otimes$ stand for element-wise multiplication, σ means a sigmoid function.

III. THE PROPOSED DCSVM-BAM

This section describes the architecture of DCSVM-BAM and its objective function.

A. The DCSVM-BAM Architecture

The DCSVM-BAM encompasses two modules: feature extraction and feature classification. The first module consists of five learnable layers and two BAM modules. The five learnable layers are made up of three convolution layers (C1, C4, and C7) and two fully connected layers (F9 and F10). A maximum pooling layer is added after each convolution layer. The two BAM modules are placed at the bottleneck of the model, including channel attention and spatial attention. The channel attention aggregates the feature maps in each channel by global average pooling first, and then learns the attention between channels with single-hidden-layer multilayer perceptron. The spatial attention effectively learns the spatial information based on contextual information by enhancing and inhibiting the features at different positions in the feature maps. The classification module classifies the output of feature extraction module via a SVM classifier. The activation function used in this study is ReLU. Fig. 4 shows the overall architecture of DCSVM-BAM.

It can be seen from Fig. 4 that the first hidden layer of the model is convolutional, with 32 kernels, each of which has a size of 5×5 and a step count of 1. The second hidden layer uses the BAM, which filters the output feature maps of the first layer in both channel and spatial terms. The third hidden layer is a maximum pooling layer, whose pooling kernel size is 2×2, and step size 2. After processing the maximum pooling layer, the length and width of feature maps become half of the original. The fourth hidden layer is convolutional.

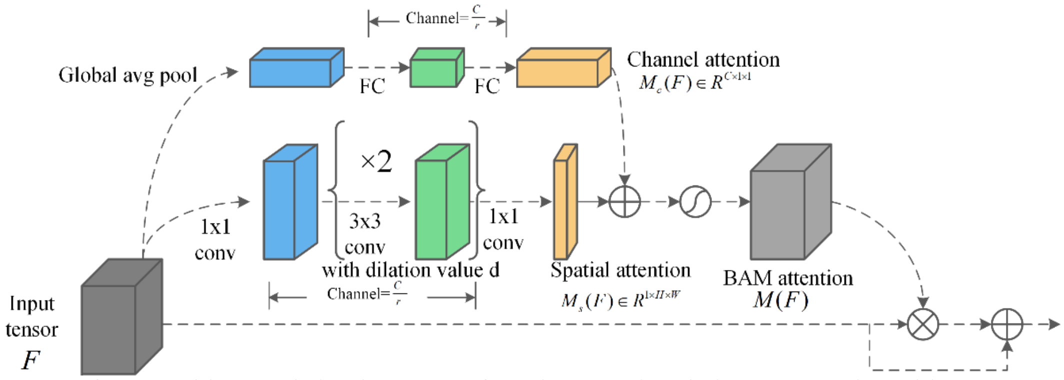

Fig. 3. The structure of BAM model [26]. It depicts the structure of generic BAM. Given the input tensor *F*, the model generates an attention map *M(F)* through the channel attention map $M_c(F)$ and the spatial attention map $M_s(F)$. The final output refined feature map is calculated according to formula (8). In addition, there are two hyper-parameters in the model: dilation value *d* and reduction ratio *r* which respectively control the sizes of receptive fields and the capacity and overhead of the model.

As in the first layer, 32 kernels are used, each of which has a size of 5×5 and a step stride of 1. The fifth hidden layer is a BAM layer, which filters the features of feature maps output by the fourth hidden layer. The sixth hidden layer is a maximum pooling layer, where 2×2 pooling kernels with a step stride of 2 are used as in the case of the third hidden layer. The seventh hidden layer is convolutional, with 32 kernels, each of which has a size of 1×1 and a step stride of 1. The eighth hidden layer is a maximum pooling layer, which has the same configuration as the previous ones. The nonlinear activation function used after each convolution layer is ReLU. After expanding the feature maps obtained in this layer, the neurons are randomly discarded with a probability of 0.5, and the subsequent output is used as the input of the next layer. The ninth hidden layer is a fully connected layer, which maps the 3200-dimensional eigenvector to a 120-dimensional eigenvector. The tenth hidden layer is also a fully connected layer, which further maps the eigenvector of the previous layer to an 84-dimensional eigenvector. The last hidden layer is the SVM layer, which classifies the previously extracted eigenvectors via SVM, and uses the classification result as the final output. DCSVM-BAM optimises the feature extraction and classification modules by mini-batch stochastic gradient descent. Table I presents the configuration information of DCSVM-BAM.

B. *Objective Function of DCSVM-BAM*

The objective function used by the DCSVM-BAM model is based on (3). Its unconstrained form is as in (10). Where C stands for the SVM penalty factor, l denotes the sample serial number, N is the total number of samples, y^l is the true label of the lth sample, o^l is the feature extracted after input model operation, and W, b respectively represent the weight and bias of SVM.

IV. EXPERIMENTAL SETTINGS

A. *Datasets*

In this experiment, we used 8 face datasets comprising of ORL, Faces94, Grimace, Jaffe, Asian, Hispanic, Black, and Multiracial. The ORL Database of Faces (ORL) [44] contains 40 distinct individuals comprising of 10 different images for each individual taken at different time with variation in lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images have a dark homogeneous background, and the subjects are photographed in an upright, frontal position (with tolerance for some side movement).

The Face Recognition Data from the University of Essex [45], is made up of 20 different images of 395 male and female various racial origins. These individuals are mainly first year undergraduate students between 18-20 years old. These images also contain individuals who are wearing glasses and with beards. The data is held in four directories. In this work we used the datasets from two directories (Faces94 and Grimace).

We obtained 213 images of 7 facial expressions from 10 Japanese female models from the Japanese Female Facial Expression (JAFFE) Database [46]. Each class contains at least 20 images. Where the images were greater than 20 in number, only 20 of them were randomly selected for the JAFFE dataset.

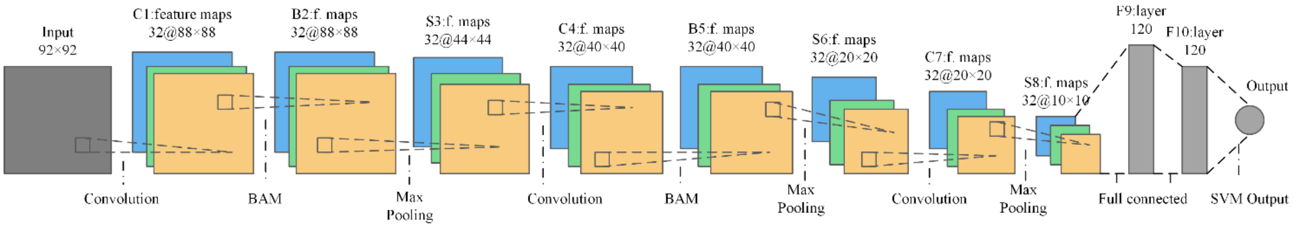

Fig. 4. Structure of the proposed DCSVM-BAM. As illustrated, the model includes three convolution layers (C1, C4, and C7), two fully connected layers (F9 and F10) and three max pooling layers (S3, S6, and S8). The two BAMs (B2 and B5) are placed at the bottleneck of the model. Multiple BAMs construct a hierarchical attention and gradually focus on the exact target which is a high-level semantic. The last layer is the SVM layer, which is used to recognize the filtered features.

TABLE I
CONFIGURATION OF THE PROPOSED DCSVMBAM

Name of layer	Description of layer	Kernel_size	Stride	Kernel_num	Output_size
Input	Input				92×92
C1	Convolution	5×5	1	32	88×88×32
B2	BAM				88×88×32
S3	Max pooling	2×2	2		44×44×32
C4	Convolution	5×5	1	32	40×40×32
B5	BAM				40×40×32
S6	Max pooling	2×2	2		20×20×32
C7	Convolution	1×1	1	32	20×20×32
S8	Max pooling	2×2	2		10×10×32
F9	Full connected				120
F10	Full connected				84
Output	SVM Output				Number of classes

$$Loss = C\sum_{l=1}^{N}[max(1-y^l(Wo^l+b),0)]^2+\frac{1}{2}\parallel W \parallel^2 \tag{10}$$

The remaining four datasets were obtained from The CNBC Face Database [47] which includes Caucasian, Asian, Hispanic, Black, and Multiracial Folders. The CNBC Face Database contains multiple images for more than 200 distinct subjects of many different ethnicities. These images were taken with consistent lighting, multiple views, facial expressions, and disguises. Individuals had varying numbers of images, and those with less than 20 images were not considered. For the remaining individuals, 20 images were reserved per person to maintain the same standard across all the datasets except the ORL.

These images were resized to a fixed resolution of 92×92. In addition, we did not perform any further pre-processing to these images, aside from subtracting the mean and dividing by the standard deviation over the training set from each pixel. This essentially means that the network was trained using the original values of the pixels.

The eight standard datasets did not provide any recommendation for splitting the data between training set and validation set (test set). In this experiment, we divided the training and validation sets at a ratio of 8:2. Table II provides the detailed information of these datasets whilst Fig. 5 displays some sample images from the eight datasets.

B. *Parameter Setting*

In this experiment, we compared the performance of DCSVM-BAM with three models of CKMSVM, CNN and SVM. CKMSVM is similar to DCSVM-BAM but without the Bottleneck Attention Module, CNN is the convolutional neural network, and the SVM model uses Support Vector Machine. We used recognition accuracy to measure the performance of the four models. The recognition accuracy is the percentage of the number of correctly classified samples in the total number of samples.

Regularisation term is an important way to reduce overfitting [48], and we employed L2 regularisation in the experiments. Through validation on the ORL dataset, weight decay (regularization coefficient) was set to 0.001 from the candidate set $\{10^k|k=-2, -3, -4, -5\}$.

Another approach to reduce overfitting is by using dropout. Dropout removes some of the hidden units with a certain probability and prevents complex co-adaptation between hidden units. [49]. According to [49], dropout regularisation can be added after the last pooling layer where each hidden unit is omitted with a default probability of 0.5. This approach could effectively prevent overfitting the model. The learning rate of neural network is increased using momentum-based technique. This method helps the optimisation process retain speed in flat regions of the loss surface and avoid local optima. A momentum of 0.9 is a typical setting of this meta-parameter and is used in this experiment [50].

We set different candidate sets for the different datasets according to the number of samples. Specifically, {8, 16, 32} was used for ORL, Faces94, Asian and Black; {4, 8, 16} was used for Grimace, Hispanic and Multiracial; and {2, 4, 8} was used for Jaffe. Learning rate is often the single most important hyperparameter [50]. In this experiment, the candidate set of Learning rate (lr) was $\{lr=i\times10^k|i=1,3,5,7,9, k=-1, -2, -3\}$, and $lr\in[0.001,0.1]$. Using cross validation, the optimal combination of mini-batch size and learning rate was selected on each dataset.

When validating the ORL dataset, Margin, the hint loss parameter of CKMSVM, was set to 1 from the candidate set $\{2^k|k=-3, -2, -1, 0, 1, 2, 3, 4, 5\}$. The *C* parameter of SVM was selected from the candidate set $\{2^k|k=-6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$. Table III lists the final hyperparameter settings for the four models.

TABLE II
DATASETS USED IN THE EXPERIMENT

No.	Datasets	The size of per image	Total number of individuals	Number of images per individual	Number of training samples	Number of testing samples
1	ORL	92×112	40	10	280	120
2	Faces94	180×200	152	20	2432	608
3	Grimace	180×200	18	20	288	72
4	Jaffe	256×256	10	20	160	40
5	Asian	250×250	53	20	848	212
6	Hispanic	250×250	19	20	304	76
7	Black	250×250	33	20	528	132
8	Multiracial	250×250	20	20	320	80

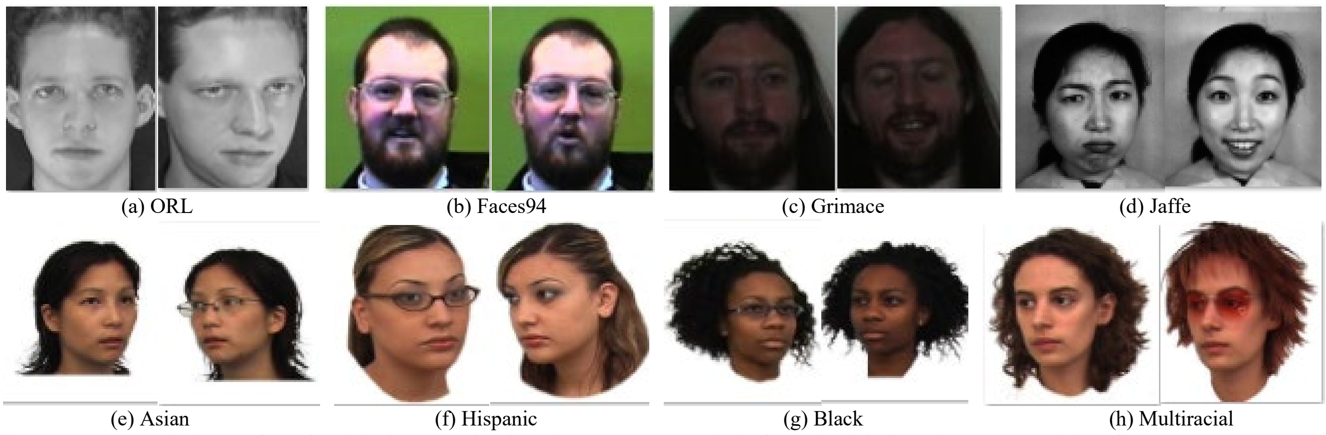
(a) ORL (b) Faces94 (c) Grimace (d) Jaffe
(e) Asian (f) Hispanic (g) Black (h) Multiracial

Fig. 5. Some sample images from the eight datasets. The above images are respectively from the eight face datasets comprising of ORL, Faces94, Grimace, Jaffe, Asian, Hispanic, Black, and Multiracial. Only two images of one individual from each dataset are shown.

TABLE III
HYPER-PARAMETERS VALUES

Datasets	SVM	CNN		CKMSVM			DCSVM-BAM			momentum	Weigh decay	Dropout
	C	Mini-batch size	Learning rate	Mini-batch size	Learning rate	Margin	Mini-batch size	Learning rate	Margin			
ORL	2	8	0.003	32	0.003	1	32	0.003	1	0.9	0.001	0.5
Faces94	0.5	32	0.005	8	0.001		8	0.001				
Grimace	0.25	16	0.001	16	0.001		4	0.001				
Jaffe	1	8	0.001	4	0.001		4	0.001				
Asian	8	8	0.001	16	0.001		16	0.001				
Hispanic	4	8	0.003	16	0.001		8	0.007				
Black	8	16	0.001	16	0.001		8	0.003				
Multiracial	8	16	0.007	8	0.001		16	0.007				

V. RESULTS AND DISCUSSIONS

A. Performance Comparisons of DCSVM-BAM with CKMSVM, CNN and SVM on Face Recognition

Using the hyperparameters obtained through cross validation in the previous section, this subsection reports on the performance of the four models using the eight face recognition datasets. In addition to accuracy, the evaluation metrics also include precision, recall, specificity, FPR, and F1_measure. The calculational formulas for the indexes are as follows:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \tag{11}$$

$$precision = \frac{TP}{TP + FP} \times 100\% \tag{12}$$

$$recall = \frac{TP}{TP + FN} \times 100\% \tag{13}$$

$$specificity = \frac{TN}{TN + FP} \times 100\% \tag{14}$$

$$FPR = \frac{FP}{TN + FP} \times 100\% \tag{15}$$

$$F1\_measure = \frac{2 \times precision \times recall}{precision + recall} \tag{16}$$

where TP (True Positive) means the number of positive samples correctly identified as positive, TN (True Negative) means the number of negative samples correctly identified as negative, FP (False Positive) means the number of negative samples incorrectly identified as positive, and FN (False Negative) means the number of positive samples incorrectly identified as negative. Table IV and Table V show the experimental results.

Based on Table IV and Table V, several observations can be made:

(1) The accuracy, precision and recall of DCSVM-BAM were higher than CKMSVM on seven datasets. DCSVM-BAM showed a relative improvement of 0.1%-4.67% on the seven datasets over CKMSVM. The specificity, FPR and F1_measure of DCSVM-BAM were better than CKMSVM on six datasets.

(2) The DCSVM-BAM achieved better results than CNN on five datasets. The accuracy, precision and recall of DCSVM-BAM showed a great improvement of 7.47%, 5.8%, and 7.57% on the Black dataset over CNN.

The DCSVM-BAM achieved better results than SVM on six datasets except ORL and Faces94 datasets. The accuracy, precision and recall of DCSVM-BAM showed a remarkable improvement of 13.5%, 19.15%, and 13.5% on the Multiracial dataset over SVM.

TABLE IV
COMPARISON OF THE DIFFERENT MODELS BASED ON ACCURACY

Datasets	DCSVM-BAM	CKMSVM	CNN	SVM
ORL	98.00	96.00	**98.87**	98.63
Faces94	99.90	99.77	99.90	**100**
Grimace	**100**	**100**	**100**	99.72
Jaffe	**100**	99.5	99.75	99.50
Asian	**70.99**	66.32	66.70	61.84
Hispanic	**74.61**	72.50	72.23	65.92
Black	**82.85**	81.06	75.38	71.74
Multiracial	**75.63**	73.37	71.12	62.13

TABLE V
COMPARISON OF THE DIFFERENT MODELS BASED ON PRECISION, RECALL, SPECIFICITY, FPR, AND F1_MEASURE

Datasets	Precision	Recall	Specificity	FPR	F1_measure
ORL	98.21/96.79/**99.04**/98.63	98.00/96.00/**98.87**/98.63	99.95/99.90/**99.97**/**99.97**	0.05/0.10/**0.03**/**0.03**	**0.98**/0.96/**0.98**/**0.98**
Faces94	99.93/99.83/99.93/**100.00**	99.90/99.77/99.90/**100.00**	99.99/99.99/99.99/**100.00**	**0.00**/**0.00**/**0.00**/**0.00**	0.99/0.99/0.99/**1.00**
Grimace	**100.00**/**100.00**/**100.00**/99.72	**100.00**/**100.00**/**100.00**/99.72	**100.00**/**100.00**/**100.00**/99.97	**0.00**/**0.00**/**0.00**/0.02	**1.00**/**1.00**/**1.00**/0.99
Jaffe	**100.00**/99.6/99.8/99.50	**100.00**/99.5/99.75/99.50	**100.00**/99.94/99.97/99.90	**0.00**/0.05/0.02/0.09	**1.00**/0.99/0.99/0.99
Asian	**75.22**/70.93/72.22/61.84	**70.90**/66.32/66.70/61.84	**99.43**/99.34/99.35/98.62	**0.56**/0.65/0.64/0.71	**0.70**/0.66/0.66/0.62
Hispanic	**80.10**/77.11/78.95/65.92	**74.61**/72.5/72.23/65.92	**98.50**/98.46/98.44/97.13	**1.41**/1.50/1.50/1.62	**0.74**/0.72/0.72/0.66
Black	**86.01**/85.71/80.21/71.74	**82.95**/81.06/75.38/71.74	**99.46**/99.40/99.22/99.16	**0.53**/0.59/0.77/0.80	**0.83**/0.81/0.75/0.71
Multiracial	**81.28**/78.69/77.03/62.13	**75.63**/73.37/71.12/62.13	**98.73**/98.61/98.49/98.16	**1.27**/1.39/1.51/1.74	**0.75**/0.73/0.70/0.62

The table displays the index values of the four models in terms of precision, recall, specificity, FPR, and F1_measure, where the four models are DCSVM-BAM, CKMSVM, CNN and SVM respectively, the index values of four models are separated by "/" in sequence.

(3) All the index values of DCSVM-BAM were better than CKMSVM, CNN and SVM on five datasets (Jaffe, Asian, Hispanic, Black, and Multiracial).

In summary, DCSVM-BAM first used a CNN with BAM to extract features, and then used soft-margin SVM to classify the features, which attained superior results to CKMSVM, CNN and SVM. There are two reasons for this. Firstly, the proposed DCSVM-BAM maps the input to the higher dimensional, and thus allows the kernel mapping to find a more appropriate space to classify. Secondly, the BAM was able to construct the attention map from both the channel and spatial branches, and in turn generating more discriminative features, which were useful for classifying the images.

B. The Effect of Learning Rate on DCSVM-BAM

This subsection explores how the batch size and learning rate affect the classification accuracy of the CNN and DCSVM-BAM models. In this experiment, only one of the datasets, the ORL, was used for discussion, and the number of training rounds was set to 50. The batch size was selected from {8, 16, 32}, whereas the learning rate was selected from {lr=i × 10^k|i=1,3,5,7,9, k=-1, -2, -3}, and lr∈ [0.001,0.1]. Tables VI and VII and Figs. 6-7 display the final experimental result.

TABLE VI
TEST ACCURACY OF CNN WITH DIFFERENT BATCH SIZES AND LEARNING RATES

Learning Rates	Different Batch Sizes		
	8	16	32
0.001	95.00	31.25	97.50
0.003	98.75	96.25	98.75
0.005	98.78	97.50	98.75
0.007	96.25	97.50	96.25
0.009	98.75	97.50	90.00
0.010	98.75	98.75	90.00
0.030	47.50	83.75	25.00
0.050	55.00	36.25	6.25
0.070	26.25	20.00	7.50
0.090	13.75	16.25	3.75
0.100	16.25	12.50	5.00

TABLE VII
TEST ACCURACY OF DCSVM-BAM WITH DIFFERENT BATCH SIZES AND LEARNING RATES

Learning Rates	Different Batch Sizes		
	8	16	32
0.001	96.25	93.75	98.75
0.003	97.50	97.50	100.00
0.005	98.75	98.75	98.75
0.007	97.50	97.50	97.50
0.009	98.75	98.75	98.75
0.010	97.50	97.50	97.50
0.030	95.00	97.50	20.00
0.050	62.50	97.50	7.50
0.070	83.75	17.50	2.50
0.090	2.50	13.75	2.50
0.100	3.75	8.75	2.50

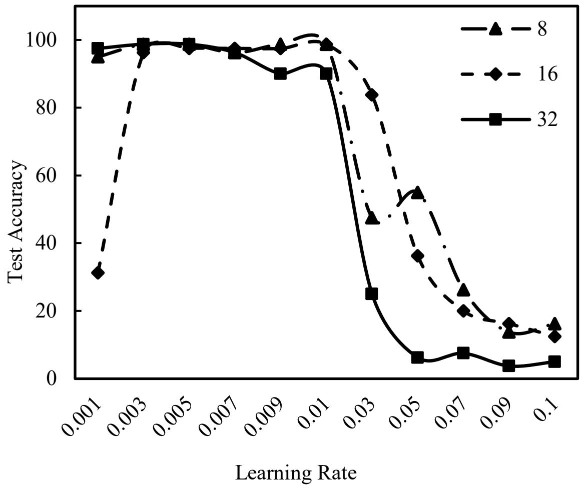

Fig. 6. Test accuracy of CNN. Only the ORL dataset was used in this experiment. The vertical axis is test accuracy, and the horizontal axis is learning rate. The batch size was set to 8, 16, and 32 respectively.

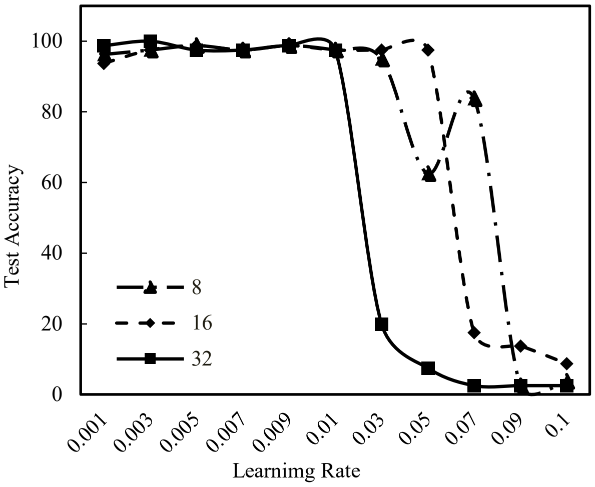

Fig. 7. Test accuracy of DCSVM-BAM. Only the ORL dataset was used in this experiment. The vertical axis is test accuracy, and the horizontal axis is learning rate. The batch size was set to 8, 16, and 32 respectively.

Based on these results, several observations can be made:

(1) When the batch size was 8, 16 and 32, the overall test accuracy of CNN model on the ORL dataset tended to increase first, then fluctuate stably, and eventually decreased as the learning rate increased from 0.001 to 0.1. In Fig. 6, with the increase of learning rate from 0.001 to 0.1 and of batch size from 8 to 32, the test accuracy fluctuated sharply, indicating that the model had a low tolerance for the batch size. The test accuracy was stable at a learning rate range of 0.003–0.007, so the batch size and learning rate were selected corresponding to this range. Hence, the batch size was set to 8, and the learning rate was set to 0.001.

(2) When the batch size was 8, 16 and 32, the test accuracy trend of DCSVM-BAM model on the ORL dataset was identical to that of CNN as the learning rate increased from 0.001 to 0.1. Based on Fig. 7, with the increase of learning rate from 0.001 to 0.1 and batch size from 8 to 32, the test accuracy fluctuated mildly, and the DCSVM-BAM had a higher tolerance for the batch size than the CNN model. Additionally, when the learning rate ranged between 0.003–0.01, the test accuracy was stable, showing a larger

stable range than the CNN model. It is suggested that the DCSVM-BAM has a higher tolerance for the learning rate than the CNN. The batch size and learning rate were selected corresponding to the above range. The batch size was finalised to 32, and the learning rate was set to 0.003.

C. The Effect of Number of Hidden Layers on DCSVM-BAM

This subsection analyses how the number of hidden layers affects the accuracy of DCSVM-BAM model. The number of hidden layers refers to the number of convolution layers. The number of hidden layers was selected from {2,3,4}, which respectively correspond to DCSVM-BAM2, DCSVM-BAM, and DCSVM-BAM4. Table VIII and Fig. 8 display the result.

TABLE VIII
THE EFFECT OF THE NUMBER OF HIDDEN LAYERS ON THE ACCURACY OF DCSVM-BAM

Data sets	DCSVM-BAM	DCSVM-BAM2	DCSVM-BAM4
ORL	**98.00**	97.25	97.00
Faces94	99.90	**99.93**	99.92
Grimace	**100.00**	99.86	99.72
Jaffe	**100.00**	99.50	99.50
Asian	**70.99**	65.19	67.59
Hispanic	**74.61**	69.08	71.18
Black	**82.85**	79.39	80.68
Multiracial	**75.63**	72.50	64.00

DCSVM-BAM, DCSVM-BAM2 and DCSVM-BAM4 respectively represent models having 3, 2 and 4 hidden layers.

The following observations are made based on Table VIII and Fig. 8.

(1) As the number of hidden layers increased from 2 to 3, the test accuracy of DCSVM-BAM was higher than DCSVM-BAM2 on seven face recognition datasets, especially on four datasets (Asian, Hispanic, Black, and Multiracial), with an improvement of 5.8%, 5.53%, 3.46%, and 3.13%, respectively. This suggests that the increase in the number of hidden layers from 2 to 3 led to improved classification performance of the DCSVM-BAM model.

(2) As the number of hidden layers increased from 3 to 4, the test accuracy of DCSVM-BAM4 was lower than DCSVM-BAM on seven datasets (ORL, Grimace, Jaffe, Asian, Hispanic, Black, and Multiracial), indicating that the continuous increase in the number of hidden layers cannot always improve the model test accuracy.

In summary, the DCSVM-BAM model has the optimal test accuracy on the eight face recognition datasets when the number of hidden layers is 3.

D. The Effect of Activation Functions on DCSVM-BAM

This subsection analyses how the activation function affects the classification accuracy of DCSVM-BAM. These activation functions are ReLU, Softplus and LeakyReLU. Table IX and Fig. 9 display the experimental result.

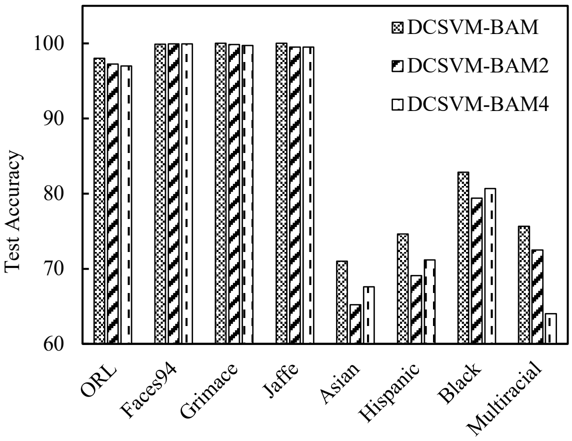

Fig. 8. The effect of the number of hidden layers on the accuracy of DCSVM-BAM. The vertical axis is test accuracy, and the horizontal axis represents the datasets. DCSVM-BAM2, DCSVM-BAM, and DCSVM-BAM4 signified respectively two, three, and four convolution layers were used.

TABLE IX
THE EFFECT OF ACTIVATION FUNCTIONS ON THE ACCURACY OF DCSVM-BAM

Datasets	DCSVM-BAM	DCSVM-BAM_softplus	DCSVM-BAM_leakyrelu
ORL	**98.00**	96.63	97.63
Faces94	99.90	99.87	**99.92**
Grimace	**100.00**	99.86	**100.00**
Jaffe	**100.00**	**100.00**	**100.00**
Asian	70.99	44.01	**72.26**
Hispanic	74.61	57.37	**74.87**
Black	**82.85**	78.03	82.50
Multiracial	**75.63**	69.00	71.25

DCSVM-BAM, DCSVM-BAM_softplus and DCSVM-BAM_leakyrelu respectively represent models using the ReLU, Softplus and LeakyReLU activation functions.

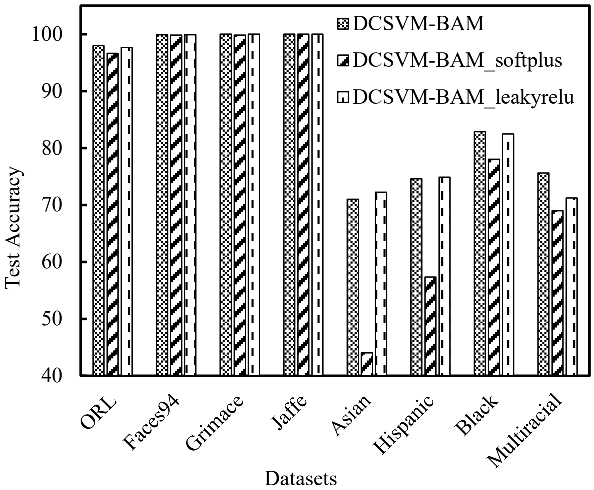

Fig. 9. The effect of activation functions on the accuracy of DCSVM-BAM. The vertical axis is test accuracy, and the horizontal axis represents the datasets. DCSVM-BAM, DCSVM-BAM_softplus and DCSVM-BAM_leakyrelu respectively represent models using the ReLU, Softplus and LeakyReLU activation functions.

Based on Table IX and Fig. 9, we can make the following analysis:

(1) The test accuracy of DCSVM-BAM, which used Softplus as the activation function, was high on one dataset (Jaffe) only, while it was low on the remaining datasets. This suggests that Softplus played an inhibiting role in the feature transfer process.

(2) DCSVM-BAM, which used LeakyReLU as the activation function, showed high test accuracies on three datasets (Face94, Asian, and Hispanic) than DCSVM-BAM, as well as equivalent test accuracies to DCSVM-BAM on two datasets (Jaffe and Grimace). This seems to indicate that LeakyReLU exerted a gain function during the feature transfer process.

E. DCSVM-BAM vs. DeepID on Face Recognition

DeepID [51] was proposed in 2014 and it has achieved outstanding result in face recognition. We have compared the proposed DCSVM-BAM with DeepID with same eight datasets in order to evaluate the DCSVM-BAM model more comprehensively. In addition to accuracy, precision, recall, specificity, FPR, and F1_measure, the testing results were also compared with number of parameters, and recognition time of single image. Table X and Table XI display the experimental results.

TABLE X
COMPARISON OF THE DCSVM-BAM AND DEEPID BASED ON NUMBER OF PARAMETERS AND RECOGNITION TIME OF SINGLE IMAGE

	DCSVM-BAM	DeepID
Number of parameters	**0.42**MB	2.05MB
Recognition time of single image	17ms	**15**ms

Based on Table X and Table XI, several observations can be made:

(1) The number of parameters used of DCSVM-BAM was significantly less compared to DeepID, about one-fifth of that of DeepID. But the recognition time of single image of DCSVM-BAM was slightly higher than the DeepID.

(2) The DCSVM-BAM achieved higher accuracy and recall than DeepID on six datasets namely ORL, Faces94, Asian, Hispanic, Black, and Multiracial. Then, both algorithms achieved 100% accuracy and recall rates for Grimace and Jaffe dataset.

(3) The DCSVM-BAM's precision performance is higher compared to DeepID on Faces94, Asian, Hispanic, Black, and Multiracial dataset. Both algorithms have achieved 100% precision rate on Grimace and Jaffe datasets. But, the DeepID has achieved slightly higher precision result for ORL dataset.

(4) On specificity, FPR and F1_measure, The DCSVM-BAM achieved better results than DeepID on ORL, Asian, Hispanic, Black, and Multiracial datasets. They achieved same results for Face94, Grimace, and Jaffe dataset.

In summary, the proposed DCSVM-BAM achieved better results compare to DeepID. Firstly, the DCSVM-BAM maps the inputs to the higher dimensional, and thus allows the kernel mapping to find a more appropriate space to classify. Secondly, the BAM was able to construct the attention map from both the channel and spatial branches and in turn generating more discriminative features, which were helpful for classifying the images. Thirdly, the DCSVM-BAM has a shallower and simpler structure than DeepID, which involves a smaller number of parameters compared to the DeepID.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, a DCSVM-BAM method for face recognition is proposed. This method initially learns the facial features through a CNN with BAM, and then maps them to an appropriate dimensional space, followed by feature classification with SVM. The advantages of this model are that it uses a CNN with BAM to explicitly express the kernel mapping and does not require kernel trick parameters. As demonstrated by the experimental results on eight public face datasets, DCSVM-BAM achieves superior recognition accuracy to CKMSVM, CNN, and SVM, and has higher tolerances for batch size and learning rate than the CNN. Moreover, compared with the DeepID model in terms of accuracy, precision, recall, specificity, FPR, and F1_measure on the eight datasets, the proposed DCSVM-BAM achieved overall better results compared to DeepID. The number of parameters used in DCSVM-BAM is far less than that of DeepID.

In order to further enhance the performance of face recognition, we will investigate how to create a CNN architecture that is more effective in the future. The market has seen the release of thousands of embedded systems. Therefore, any real-world application, including an attendance management system, smartphone security, a smart door lock system, and home and office video security, could be tested using the proposed model [52]. The proposed algorithm's efficiency and efficacy will not be known for sure until it is fully implemented. It is likely that the hardware used, and latency could be challenging. Any outperforming face recognition method can be tested for robustness in the agriculture and licence plate recognition cognition, as recommended by [53][54].

TABLE XI
COMPARISON OF DCSVM-BAM AND DEEPID BASED ON ACCURACY, PRECISION, RECALL, SPECIFICITY, FPR AND F1_MEASURE

Datasets	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	FPR (%)	F1_measure
ORL	**98.00**/97.87	98.21/**98.34**	**98.00**/97.50	**99.95**/99.94	**0.05**/0.06	**0.98**/0.97
Faces94	**99.90**/99.73	**99.93**/99.77	**99.90**/99.73	**99.99**/99.99	**0.00**/0.00	**0.99**/0.99
Grimace	**100.00**/100.00	**100.00**/100.00	**100.00**/100.00	**100.00**/100.00	**0.00**/0.00	**1**/1
Jaffe	**100.00**/100.00	**100.00**/100.00	**100.00**/100.00	**100.00**/100.00	**0.00**/0.00	**1**/1
Asian	**70.99**/70.14	**75.22**/74.53	**70.90**/70.14	**99.43**/99.42	**0.56**/0.58	**0.70**/0.69
Hispanic	**74.61**/70.66	**80.10**/76.77	**74.61**/70.66	**98.50**/98.36	**1.41**/1.63	**0.74**/0.70
Black	**82.85**/80.68	**86.01**/84.19	**82.95**/80.68	**99.46**/99.39	**0.53**/0.61	**0.83**/0.80
Multiracial	**75.63**/70.12	**81.28**/73.53	**75.63**/70.12	**98.73**/98.44	**1.27**/1.55	**0.75**/0.69

The table displays the index values of the two models in terms of accuracy, precision, recall, specificity, FPR, and F1_measure, where the values before "/" represent the index values of DCSVM-BAM, and the values after "/" are those of DeepID.

REFERENCES

[1] P. Marasamy, and S. Sumathi, "Automatic recognition and analysis of human faces and facial expression by LDA using wavelet transform," *2012 International Conference on Computer Communication and Informatics. IEEE*, pp. 1-4, 2012 https://doi.org/10.1109/ICCCI.2012.6158798

[2] D. Valentin, H. Abdi, AJ. O'Toole, and Cottrell. GW, "Connectionist models of face processing: a survey," *Pattern Recognition*, vol. 27, no. 9, pp. 1209-1230, 1994. https://doi.org/10.1016/0031-3203(94)90006-X

[3] Y. Adini, Y. Moses, and S. Ullman, "Face recognition: the problem of compensating for changes in illumination direction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no.7, pp. 721-732, 1997. https://doi.org/10.1109/34.598229

[4] R. Brunelli, and T. Poggio, "Face recognition: features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042-1052, 1993. https://doi.org/10.1109/34.254061

[5] W R. Schwartz, H. Guo, J. Choi, and L. S. Davis, "Face identification using large feature sets," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2245-2255, 2011. https://doi.org/10.1109/TIP.2011.2176951

[6] C. Xu, Y. Wang, T. and Tan, L. Q, "Automatic 3D face recognition combining global geometric features with local shape variation information," *The Sixth IEEE International Conference on Automatic Face and Gesture Recognition. IEEE*, pp. 308-313, 2004. https://doi.org/10.1109/AFGR.2004.1301549

[7] Y. Li, Y. Shen, G. Zhang, T. Yuan, X. Xiao and H. Xu, "An efficient 3D face recognition method using geometric features," *The 2nd International Workshop on Intelligent Systems and Applications. IEEE*, pp. 1-4, 2010. https://doi.org/10.1109/IWISA.2010.5473292

[8] L. Ballihi, B. B. Amor, M. Daoudi, A. Srivastava and D. Aboutajdine, "Boosting 3-D-geometric features for efficient face recognition and gender classification," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1766-1779, 2012. https://doi.org/10.1109/TIFS.2012.2209876

[9] I. J. Cox, J. Ghosn, and P. N. Yianilos, "Feature-based face recognition using mixture-distance," *Proceedings of the International Conference on Computer Vision and Pattern Recognition. IEEE*, pp. 209-216, 1996. https://doi.org/10.1109/CVPR.1996.517076

[10] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144-152, 1992. https://doi.org/10.1145/130385.130401

[11] P. J. Phillips, H. Wechsler, J. Huang, and J. R. Patrick, "The FERET database and evaluation procedure for face-recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295-306, 1998. https://doi.org/10.1016/S0262-8856(97)00070-X

[12] S B. Zheng, and G C. Guo, "Efficient scheme for two-atom entanglement and quantum information processing in cavity QED," *Physical Review Letters*, vol. 85, no. 11, pp. 2392, 2000. https://doi.org/10.1103/PhysRevLett.85.2392

[13] B. Heisele, P. Ho, and T. Poggio, "Face recognition with support vector machines: global versus component-based approach," *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. IEEE*, vol. 2, pp. 688-694, 2001. https://doi.org/10.1109/ICCV.2001.937693

[14] A. Krizhevsky, I. Sutskever, G E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Nural Information Processing Systems*, vol. 25, pp. 1097-1105, 2012. https://doi.org/10.5555/2999134.2999257

[15] K. He, X. Zhang, S. Ren, and Sun. J, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016. https://doi.org/10.1109/CVPR.2016.90

[16] Y. Liu, B. Fan, S. Xiang, and Pan. C, "Relation-shape convolutional neural network for point cloud analysis," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8895-8904, 2019. https://doi.org/10.1109/CVPR.2019.00910

[17] A. A. M. Al-Saffar, H. Tao, and M. A. Talab, "Review of deep convolution neural network in image classification," *International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications, IEEE*, pp. 26-31, 2017. https://doi.org/10.1109/ICRAMET.2017.8253139

[18] W. Rawat, and Z. Wang. "Deep convolutional neural networks for image classification: a comprehensive review," *Neural Computation*, vol. 29, no. 9, pp. 2352-2449, 2017. https://doi.org/10.1162/neco_a_00990

[19] S, Ma, X. Zhang, C. Jia, Z. Zhenghui, W. Shiqi, and W. Shanshe, "Image and video compression with neural networks: a review," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. https://doi.org/10.1109/TCSVT.2019.2910119

[20] M. Coşkun, A. Uçar, Ö. Yildirim, and Y. Demir, "Face recognition based on convolutional neural network," *The International Conference on Modern Electrical and Energy Systems (MEES). IEEE*, pp. 376-379, 2017. https://doi.org/10.1109/MEES.2017.8248937

[21] Y. Zhang, D. Zhao, J. Sun, G. Zou and W. Li, "Adaptive convolutional neural network and its application in face recognition," *Neural Processing Letters*, vol. 43, no. 2, pp. 389-399, 2016. https://doi.org/10.1007/s11063-015-9420-y

[22] Y. Li, G. Wang, L. Nie, Q. Wang, and W. Tanc, "Distance metric optimisation driven convolutional neural network for age invariant face recognition," *Pattern Recognition*, vol. 75, pp. 51-62, 2018. https://doi.org/10.1016/j.patcog.2017.10.015

[23] X. Li, W. Zhang, and Q. Ding, "Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism," *Signal Processing*, vol. 161, pp. 136-154, 2019. https://doi.org/10.1016/j.sigpro.2019.03.019

[24] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48-62, 2021. https://doi.org/10.1016/j.neucom.2021.03.091

[25] J. Liu, B. Jin, L. Wang, and L. Xu, "Sea surface height prediction with deep learning based on attention mechanism," *IEEE Geoscience and Remote Sensing Letters*, 2020. https://doi.org/10.1109/LGRS.2020.3039062

[26] J. Park, S. Woo, J. Lee, and S. Kweon, "BAM: Bottleneck Attention Module," arXiv e-prints, 2018:1807.06514. https://arxiv.org/abs/1807.06514

[27] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no.4, pp. 541-551, 1989. https://doi.org/10.1162/neco.1989.1.4.541

[28] Kai Zheng, ZhiGuang Xia, Yi Zhang, Xuan Xu, and Yaqin Fu, "Speech Emotion Recognition based on Multi-Level Residual Convolutional Neural Networks," Engineering Letters, vol. 28, no.2, pp559-565, 2020.

[29] Longlei Cui, and Ying Tian, "Facial Expression Recognition by Regional Attention and Multi-task Learning," Engineering Letters, vol. 29, no.3, pp919-925, 2021.

[30] C. Cortes, and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995. https://doi.org/10.1007/BF00994018

[31] V. Vapnik, "The support vector method of function estimation," *Nonlinear Modeling*. Springer, Boston, MA, 1998: 55-85. https://doi.org/10.1007/978-1-4615-5703-6_3

[32] E J. Candes, J K, Romberg, and T. Tao. "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 59, no. 8, pp. 1207-1223, 2006. https://doi.org/10.1002/cpa.20124

[33] D H. Walter, K. Rittig, F. H. Bahlmann, R. Kirchmair, M. Silver, T. Murayama, H. Nishimura, DW. Losordo, T. Asahara, dan JM. Isner, "Statin therapy accelerates reendothelialisation: a novel effect involving mobilisation and incorporation of bone marrow-derived endothelial progenitor cells," *Circulation*, vol. 105, no. 25, pp. 3017-3024, 2002. https://doi.org/10.1161/01.cir.0000018166.84319.55

[34] M. Pontil, and A. Verri, "Support vector machines for 3D object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 637-646, 1998. https://doi.org/10.1109/34.683777

[35] Yomna M. Elbarawy, and Wafaa A. Ghonaim, "Hybridized Convolution Neural Network and Multiclass-SVM Model for Writer Identification," Engineering Letters, vol. 29, no.1, pp317-326, 2021.

[36] G. Zheng, Z. Qian, Q. Yang, C. Wei, L. Xie, Y. Zhu and Y. Li, "The combination approach of SVM and ECOC for powerful identification and classification of transcription factor," *BMC Bioinformatics*, vol. 9, no.1, pp. 1-8, 2008. https://doi.org/10.1186/1471-2105-9-282

[37] I. Steinwart, and A. Christmann, "Support vector machines," *Springer Science & Business Media*, 2008.

[38] B. Dzmitry, Cho. K, and B. Yoshua, "Neural Machine Translation by Jointly Learning to Align and Translate," CoRR, vol. abs/1409.0473, 2014. https://doi.org/10.48550/arXiv.1409.0473

[39] Liu Dong, Chen Longxi, Wang Lifeng, Wang Zhiyong, "A multi-modal emotion fusion classification method combined expression and speech based on attention mechanism," Multimedia Tools and Applications, pp. 1-19, 2021. https://doi.org/10.1007/s11042-021-11260-w

[40] Zou Hongyan, Sun Xinyan, "3D Face Recognition Based on an Attention Mechanism and Sparse Loss Function" Electronics, 10 (20), pp: 2539-2539, 2021. https://doi.org/10.3390/electronics10202539

[41] S. Bell, C. L. Zitnick, K. Bala and R. Girshick, "Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2874-2883, 2016. https://doi.org/10.1109/CVPR.2016.314

[42] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localisation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 447-456, 2015. https://doi.org/10.1109/CVPR.2015.7298642

[43] F. Yu, and V. Koltun, "Multi-scale context aggregation by dilated convolutions," arXiv preprint arXiv:1511.07122, 2015. https://doi.org/10.48550/arXiv.1511.07122

[44] S Z. Li, and J. Lu, "Face recognition using the nearest feature line method," *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp. 439-443, 1999. https://doi.org/10.1109/72.750575

[45] F. Ahmad, A. Najam, and Z. Ahmed, "Image-based face detection and recognition: state of the art," arXiv preprint arXiv:1302.6379, 2013. https://doi.org/10.48550/arXiv.1302.6379

[46] M. Abdulrahman, T. R. Gwadabe, F. J. Abdu and A. Eleyan, "Gabor wavelet transform based facial expression recognition using PCA and LBP," *2014 22nd Signal Processing and Communications Applications Conference (SIU)*, pp. 2265-2268, 2014. https://doi.org/10.1109/SIU.2014.6830717

[47] C L. Wilkins, J F. Chan, and C R. Kaiser, "Racial stereotypes and interracial attraction: phenotypic prototypicality and perceived attractiveness of asians," *Cultural Diversity and Ethnic Minority Psychology*, vol. 17, no. 4, pp. 427, 2011. https://doi.org/10.1037/a0024733

[48] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 regularisation for learning kernels," arXiv preprint arXiv:1205.2653, 2012. https://doi.org/10.48550/arXiv.1205.2653

[49] P. Baldi, and P J. Sadowski, "Understanding dropout," *Advances in Neural Information Processing Systems*, vol. 26, pp. 2814-2822, 2013.

[50] M W. Browne, "Cross-validation methods," *Journal of Mathematical Psychology*, vol. 44, no. 1, pp. 108-132, 2000. https://doi.org/10.1006/jmps.1999.1279

[51] Y., S., W. X. and T. X. "Deep learning face representation from predicting 10,000 classes," in 2014 IEEE Conference on Computer Vision and Pattern Recognition. 2014. https://doi.org/10.1109/CVPR.2014.244

[52] CS. Keau, CK. On, MHA. Hijazi, and MM. Singh, "Smart-Hadir-mobile based attendance management system," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 14, 2021. https://doi.org/0.3991/ijim.v15i14.22677

[53] I. K. Witus, , C. K. On, R. Alfred, A.Ag.A. Ibrahim, T. T. Guan, and P. Anthony, "A review of computer vision methods for fruit recognition," *Advanced Science Letters*, vol. 24, no. 2, pp. 1538-1542, 2018. https://doi.org/10.1166/asl.2018.10786

[54] C. K. On, T. K. Yau, R. Alfred, J. Teo, P. Anthony, and W. Cheng, "Backpropagation neural ensemble for localising and recognising non-standardised Malaysia's car plates", *International Journal on Advanced Science, Engineering and Information Technology*, vol.6, no.6, 2016

CHI JING is an Associate Professor in the School Information and Electrical Engineering, Hebei University of Engineering. She received her Bachelor of Computer Application and Master of Computer Technology, from Hebei University Science and Technology and Taiyuan University of Technology in 1996 and 2003. She is now pursuing Ph.D in computer science at University Malaysia Sabah (UMS). Her research interest generally falls under Computer Vision & Pattern Recognition, such as image processing, image classification, object detection, and vision-based learning.

ZHANG HAOPENG is a postgraduate student in the School of Information and Electrical Engineering, Hebei University of Engineering, graduated from Hebei University of Economics and Business with his Bachelor of Computer Science and Technology in June 2020, and is a member of CCF. His research interests are in the field of computer vision and pattern recognition, such as image processing, image classification, target detection, etc. During his undergraduate and graduate studies, he learned the theoretical knowledge of Artificial Intelligence, Machine Learning and Neural Networks, which laid the foundation for completing this topic.

KIM ON, CHIN received his PhD in Artificial Intelligence with the University Malaysia Sabah, Sabah, Malaysia during 2010 and he is currently working as an Associate Professor at the University Malaysia Sabah in the Faculty of Computing and Informatics. His research interests are gaming AI, evolutionary computing, evolutionary robotics, artificial neural networks, image processing, agent technologies, evolutionary data mining, and biometric security system with mainly focused on fingerprint and voice recognition. He has led several projects related to artificial neuro-cognition for solving real world problems such as mobile based number plate detection and recognition, off-line handwriting recognition, item drop mechanism and auto map generation in gaming AI, as named a few. He has authored and co-authored more than 120 articles in the forms of journals, book chapters and conference proceedings. He is a Senior Member of IEEE and IAENG societies.

ERVIN GUBIN MOUNG is a senior lecturer in the Faculty of Computing and Informatics, University Malaysia Sabah. He received his Bachelor of Computer Engineering, Master of (Computer) Engineering, and Ph.D. in Computer Engineering from University Malaysia Sabah (UMS) in 2008, 2013, and 2018. His research interest generally falls under Computer Vision & Pattern Recognition, such as image processing, image segmentation, image classification, object detection, vision-based learning, and big data analytics. His domain of interest includes public health, smart health, agriculture, food security, biodiversity, and environmental sustainability.

PATRICIA ANTHONY is an Associate Professor in the Faculty of Environment, Society and Design at Lincoln University, New Zealand. She received her PhD degree from University of Southampton in 2003. Her research interest is in agents and multi-agent systems, machine learning for natural language processing and text analytics and the applications of artificial intelligence in agriculture and environmental systems. She has published over 100 papers in peer reviewed journals and international conferences.