



# A hybrid approach to building face shape classifier for hairstyle recommender system



Kitsuchart Pasupa<sup>a,\*</sup>, Wisuwat Sunhem<sup>a</sup>, Chu Kiong Loo<sup>b</sup>

<sup>a</sup> Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

<sup>b</sup> Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia

## ARTICLE INFO

### Article history:

Received 12 March 2018

Revised 12 September 2018

Accepted 6 November 2018

Available online 6 November 2018

### Keywords:

Face shape classification

Deep-learned feature

Hand-crafted feature

Hybrid feature-based approach

Feature combination

## ABSTRACT

Identifying human face shape is the first and the most vital process prior to choosing the right hairstyle to wear on according to guidelines from hairstyle experts, especially for women. This work presents a novel framework for a hairstyle recommender system that is based on face shape classifier. This framework enables an automatic hairstyle recommendation with a single face image. This has a direct impact on beauty industry service providers. It can simulate how the user looks like when she is wearing the chosen hairstyle recommended by the expert system. The model used in this framework is based on Support Vector Machine. The framework is evaluated on hand-crafted, deep-learned (VGG-face) features and VGG-face fine-tuned version for the face shape classification task. In addition to evaluating these individual features by a well-designed framework, we attempted to fuse these three descriptors together in order to improve the performance of the classification task. Two combination techniques were employed, namely: Vector Concatenation and Multiple Kernel Learning (MKL) techniques. All the hyper-parameters of the model were optimised by using Particle Swarm Optimisation. The results show that combining hand-crafted and VGG-face descriptors with MKL yielded the best results at 70.3% of accuracy which was statistically significantly better than using individual features. Thus, combining multiple representations of the data with MKL can improve the overall performance of the expert system. In addition, this proves that hand-crafted descriptor can be complementary to deep-learned descriptor.

© 2018 Elsevier Ltd. All rights reserved.

## 1. Introduction

Which hairstyles should we wear? It is one of the toughest questions that we usually ask ourselves whenever we want to change to a new style or enhance our existing look at a hair salon. In altering hairstyle, we have to bear in mind that there is no perfect cut for everyone. You may experience a disastrous situation if you choose a very nice hairstyle that you like from a magazine and then try it without consulting with a beauty expert (ZergNet, 2017). According to a beauty expert's guidelines, the face shape can be divided into five groups, namely: round, oval, oblong, square, and heart shapes, each of which has a set of suitable hairstyles. To enable automated recommendation of hairstyle, we first need to define these human face shapes by basic geometry together with some decision rules, then a set of suitable hairstyles

can be listed for a specific shape. There are clear guidelines that match an individual face shape with the characteristics of some suitable hairstyles (Derrick, 2017; Hong, 2013).

According to the aforementioned process, the primary issue is to invent an effective face shape classifier. Recently, Sunhem & Pasupa have proposed an approach to solve the issue (Sunhem & Pasupa, 2016). Their approach was inspired by many classifying methods with a similar root concept but for different purposes (Bansode & Sinha, 2016; Li, So, Shin, & Han, 2013; Sarakorn, Charoenpong, & Charoensiriwath, 2014). Our recent technique has an ability to classify women's face shapes into five groups according to expert guidelines. The approach mainly relies on representative features that can be calculated from a set of key landmark points on a human face. These points can be obtained by an Active Appearance Model (AAM), and a Colour-based Skin Segmentation. In this paper, we refer to these representative features as "Geometric feature". This feature is a type of hand-crafted feature that is extracted by particular image processing techniques such as AAM and object detection. In our previous experiment, the geometric feature was evaluated by Linear Discriminant Analysis,

\* Corresponding author.

E-mail addresses: [kitsuchart@it.kmitl.ac.th](mailto:kitsuchart@it.kmitl.ac.th) (K. Pasupa), [wisuwat.sun@gmail.com](mailto:wisuwat.sun@gmail.com) (W. Sunhem), [ckloo.um@um.edu.my](mailto:ckloo.um@um.edu.my) (C.K. Loo).

Support Vector Machine (SVM), and Artificial Neural Network. The results show that SVM in conjunction with Radial Basis Function (RBF) kernel was the best contender.

In addition, there is another type of feature adopted in a wide range of research called “deep-learned feature” which is generated by deep machine learning algorithms such as Convolutional Neural Network (CNN) in the field of image analysis (Fukushima & Miyake, 1982). CNN is one of the most popular deep learning models because of its ability to automatically extract image features from a raw data set. It is usually used as an image classifier. Notwithstanding, it can also perform as a feature extractor by ignoring the inference layers. In Deng et al. (2009), Sharif Razavian, Azizpour, Sullivan, and Carlsson (2014), Girshick, Donahue, Darrell, and Malik (2014), they utilised a deep feature extractor based on the effective transfer learning practice—a use of the model’s architecture with its pre-trained parameters from other similar tasks—that has performed well. In general, a customised feature extractor is likely to yield a better performance. Therefore, rather than using just the pre-trained model, some of these studies put more effort into developing the model by further exploiting a fine-tuning strategy on the pre-trained model. This can be done by tuning the model with their data set with a small number of iterations. There was an attempt to evaluate CNN-training from scratch—for face shape classification in our previous work (Pasupa & Sunhem, 2016). Unfortunately, the deep architecture performed poorly without supplementing with regularisation techniques compared to shallow models, i.e. ANN and SVM, due to the small number of training samples. However, when a regularisation technique—dropout—was used, the deep architecture was competitive to the shallow models.

Although deep learning algorithms enable us to automatically obtain most of the possible discriminative features, a hand-designed one built from a specific-domain expert can provide more generalised features and avoid the over-fitting problem of the model. In order to preserve the advantages of both features, researchers have proposed approaches to combining them. Recently, Majtner et al. attempted to combine deep-learned feature with RSurf—a hand-crafted feature—on a Melanoma classification task (Majtner, Yildirim-Yayilgan, & Hardeberg, 2016). Each type of features was used to build an individual SVM model. The predicted label would be the label which had the highest probability. Furthermore, there was an attempt to combine these features with the sequential structure of human body movement information from RBG-D (Red, Green, Blue, and Depth) data by Wang, Li, Hou, and Li (2016). The deep-learned feature was extracted by CNN from the depth modality while the hand-crafted features were extracted from the skeleton-motion histogram. In order to classify this sequential data, the combined features were fed into a Hidden Markov Model (HMM) in order to explore the global temporal information prior to the final classification by the SVM. In addition, the feature combination technique was recently applied in person re-identification task (Wu et al., 2016) called Feature Fusion Net. The study demonstrates that the hand-crafted histogram feature is useful and can complement the deep-learned feature. The proposed technique clearly outperformed the other conventional methods.

A widely used and naïve fusion technique is feature concatenation. However, there are a lot of evidence showing that fusing features in a principle way by using Multiple Kernel Learning (MKL) techniques can improve the predictive performance of the model, see a well-documented description of MKL and the reviews on it in Gönen and Alpaydin (2011). Recently, an Elastic MKL was proposed by Pasupa, Hussain, Shawe-Taylor, and Willett (2013) and was applied to a drug screening problem with four different types of descriptors. The proposed method was able to yield a better overall drug-screening performance than conventional methods was, es-

specially for the heterogeneous data set that had a high diversity among the substructures of the molecules in the database.

After undertaking a literature search, we found a recent work related to hairstyle decision making for customers, i.e. Liang and Wang (2016). Their approach to hairstyle recommendation was based on customer ratings on hairstyles and expert’s recommendation based on gender, age, occupation, feature, and skin colour. The system suggests a set of hairstyles that matches the users’ profile and their preferences. However, there is a high possibility that the hairstyles might not be truly suitable to the users. A similar system which is related to fashion is a personalised fashion recommendation system by Kang, Fang, Wang, and McAuley (2017). This system lists a set of recommended items after getting detailed information on the user and the product category. It utilises user’s click as an implicit feedback to the system. The paper concluded that the personalised models—taking into account user’s background—showed a capable performance in user’s satisfaction. The system not only suggests existing items to the user, but also generates new fashion images which may be able to satisfy the user too. They employed Generative Adversarial Network to handle this task. This framework benefits manufacturer in modification of their existing items to match users’ requirements. However, it requires more effort to collect data. Recently, there has been an attempt to build a system to assist consumer to locate online stores—fashion products—on Instagram (Lodkaew, Supsohboon, Pasupa, & Loo, 2018). It performs a product image retrieval task. The system employs a deep learning model to perform product image segmentation. Then, deep-learned features are extracted to characterise the segmented images. The system still has a few limitations, e.g. unable to identify the fabric or material of fashion products. Moreover, there was an attempt to model a facial beauty based on facial geometric representation (Zhang, Zhang, Sun, & Chen, 2017). Other facial appearances, e.g. skin colour, texture, were not considered because it can be changed by facial make-up. The problem was solved by a semi-supervised learning technique because there is a limited number of attractive faces and a huge gap between the number of attractive and non-attractive faces. Moreover, no controlling of unlabelled sample distribution in the training set could result in high variance of the outcomes which will lead to instability of the model (Lu, 2009). To solve the scarcity of labelled data, a sufficient degree of prior knowledge on the data set should be put into consideration.

This paper aims to construct a hairstyle recommendation system that can expand the opportunities in fashion business. As our literature survey has found, there are a number of research gaps that has been addressed as the contributions from this paper. These are as follows.

- (i) In beauty industrial perspective, a hairstyle recommender system by Liang and Wang (2016) aims to construct a beauty expert system that is based on user’s preferences (Liang & Wang, 2016). The approach would concern only personal preferences without consideration of the suitability of the recommendation to their actual appearance. The recommended hairstyle might not suit other users as it should be related only to their own looks. Thus, we propose an approach that provides both suitability and reliability as it follows an experts guidelines in using face shape to determine the hairstyle.
- (ii) In our recent study (Sunhem & Pasupa, 2016), the geometric feature was introduced for the first time with a new face shape data set and its preliminary results. In this work, we further investigated and analysed these features on a more complete experimental framework.
- (iii) According to Liang and Wang (2016) and Kang et al. (2017), none of them illustrated technique that deals with small

sample-sized data set, while (Zhang et al., 2017) proposed an approach to handle label scarcity problem with semi-supervised learning. Due to the small number of samples of the data set, pre-trained models from similar tasks, e.g. face recognition, should be used to extract visual feature. This is because the model can be easily over-fitted if it is trained from scratch due to the small number of training samples. To the best of our knowledge, there has been no attempt on applying transfer learning techniques to face shape classification. Here, VGG-face descriptor for face recognition task—a widely used pre-trained model proposed by the University of Oxford—is applied (Parkhi, Vedaldi, & Zisserman, 2015). As the fine-tuning technique can enhance the performance of the model, we further perform this technique on the VGG-face descriptor with our data set.

(iv) Our recent study reports that the shallow and deep models are statistically comparable methods for small sample-sized data set (Pasupa & Sunhem, 2016). Both methods utilise different types of features including distinct discriminative features obtained from the same raw data. Therefore, fusing both types of features—hand-crafted and deep-learned features—may be beneficial to the classification task<sup>1</sup>. Majtner et al. (2016) employed two models with two different types of feature, hand-crafted and deep-learned features, the predicted label was obtained based on whose confidence score is higher. Wang et al. (2016) demonstrated that deep network cannot handle global translation while hand-crafted feature, Histogram of Oriented Displacement is more robust. They clustered training data based on the hand-crafted features. Then, each cluster was given a label. After that, a deep-learned model was used to encode raw data before feeding to HMM-SVM. They concluded that deep representation was then more invariant. However, this technique is useful only if raw data are encoded with CNN that is trained from scratch according to cluster labels of hand crafted feature. Therefore, this technique is not applicable to a data set with a small sample size. In our work, all of the individual features—the geometric, VGG-face, fine-tuned VGG-face features—are compared with different combinations of these features that are fused by either conventional feature concatenation or MKL techniques. We also demonstrate which features are the most prominent features in the face shape classification task—one of the benefits of MKL. Unlike previous papers in the literature (Majtner et al., 2016; Wang et al., 2016), we directly utilised all features as input to a single algorithm.

According to the above contributions, the model involves many hyper-parameters. In order to obtain the optimal performance from each method, the optimal hyper-parameters can be searched by employing one of the most powerful optimisation techniques called Particle Swarm Optimisation (PSO).

This paper is organised as follows. The overview of the designed system starting from data collection for generating a face shape classification model; methodologies used to accomplish the task, i.e. hand-crafted and deep-learned feature extraction; and the work-flow of the application including decision rules for selecting appropriate hairstyles to a particular face shape is explained in Section 2. Section 3 describes the experimental framework for evaluating models. This includes the hybrid feature approaches—feature vector concatenation and MKL—to enhancing the model's performances over using individual features explained in Section 2. This section also includes the model parameter tuning process op-

timised by PSO. Next, the experimental results are discussed and statistically evaluated in Section 4. Lastly, the conclusion and future works are presented in Section 5.

## 2. Development of a hairstyle recommendation system

### 2.1. The face shape data set

In order to construct a generalised face shape classifier, a data set is required to enable us to create the model. By undertaking a literature review, we have found many resources that are related to face shape classification task, e.g. Parkhi et al. (2015), Grgic, Delac, and Grgic (2011), Jain and Learned-Miller (2010), Huang, Ramesh, Berg, and Learned-Miller (2007). Unfortunately, none of these are labelled according to the shapes stated in the hairstyle selection guideline mentioned in the introduction—round, oval, oblong, square, and heart shapes. Moreover, subjects should look straight towards the camera; their heads should neither turned nor at an angle to it; and the faces must be clearly visible. Here, we focus only on women's face shapes. These are the reasons why we decided to start collecting from scratch instead of utilising the existing and available public data set related to our work.

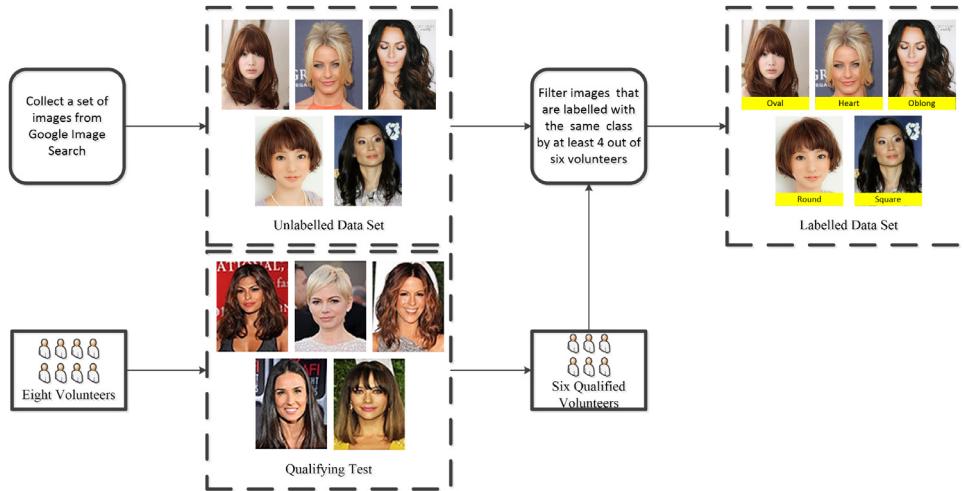
We collected women face images by using Google Image search engine with different relevant search queries according to the expert's guidelines such as 'long women face'. After the searches, 1000 relevant face images were collected. However, these images were not tagged. In addition, the queries used in the searches could not be used as suitable labels because these retrieved images were not labelled by experts and might not be relevant to the queries. It should be noted that there were only a limited number of available resources that contained images labelled by experts. One of the reliable hairstyle recommendation articles in the literature offers face images with labels specified by an expert, Mika Fowler, who is a celebrity hairstylist and resident dry cut expert (Fowler, 2015). Unfortunately, there were only ten images. Due to the small number of labelled images, we certainly would not be able to create a good predictive model. Therefore, these 1000 images were required to be manually labelled. In order to do so, one of the feasible solutions was to recruit a number of participants to manually provide shapes that corresponded to the faces in all 1000 images.

Initially, we recruited eight volunteers to label this set of images. For the volunteers to be qualified to complete the task and able to distinguish women face like an expert, they were required to pass a qualification test. They were first requested to study the expert's guidelines in Derrick (2017) before participating in the test. To leverage the valid set of images available at Fowler (2015), we utilised them as a test for evaluating their performance on the classification task. The test consisted of the images of ten women's faces with two images of each face shape as shown in Fig. 1. The participants would be given a score if he/she was able to identify a label to a face that was the same label that the expert did. Finally, there were only six qualified participants who was able to achieve a high score of at least 8 out of 10. After the data collection task was completed, some images were identified with a large variety of labels given by all participants. This might be because of the ambiguity of the face shapes. Therefore, this led to a noisy label problem. To avoid constructing a model with this problem, we included only the images that were similarly identified by at least 4 out of the 6 participants in the valid data set, while the others were excluded from the experiments. Finally, the data set contained 500 valid samples with 100 images per face shape. The data collection process is illustrated in Fig. 2. This data set have been used in our recent studies (Sunhem & Pasupa, 2016). It is publicly available for downloading at <https://github.com/dsmlr/faceshape>.

<sup>1</sup> Hand-crafted features extracted with prior knowledge can enhance the stability of the model (Zhang et al., 2017).



**Fig. 1.** The ten face shapes labelled by an expert—Mika Fowler—adopted for the qualification test of volunteers (Fowler, 2015).



**Fig. 2.** Face shape data collection process.

## 2.2. Facial feature extraction for face shape classification

According to the previous subsection on the face shape data set, the next step is to train a model by a machine learning algorithm with this data set. In general, most machine learning algorithms require a feature vector as an input to represent the attributes or characteristics of each sample instead of using raw data. As mentioned in the introduction that there are two main types of features namely hand-crafted and deep-learned features, this work tends to combine both types together as they can be a complementary to each other. Hence, combining them can improve the performance of the task.

### 2.2.1. Hand-crafted feature

The face region is required to be identified first in order to determine the face shape. To identify the face region, landmark localisation technique called *Active Appearance Model* (AAM) (Cootes, Edwards, & Taylor, 2001) in conjunction with colour-based skin segmentation (Mustafa, 2007) is utilised. AAM is a technique to obtain vital points on face images. In addition to AAM, colour-based skin segmentation can define the point indicating the hairline that separates the hair and forehead. Then, these points are used to calculate the geometric features to represent face shape as hand-crafted features.

AAM is a statistical model that was built on top a previous model—*Active Shape Model* (ASM). The core of this technique is to

generate a model that can represent the texture and shape of an interesting object that will undergo a feature extraction procedure. The technique requires a set of images with their predefined landmark positions in order to learn and build a landmark localisation model. The represented landmarks for each sample can be defined as a vector  $s$  containing  $L$  landmark positions formulated by the following equation:

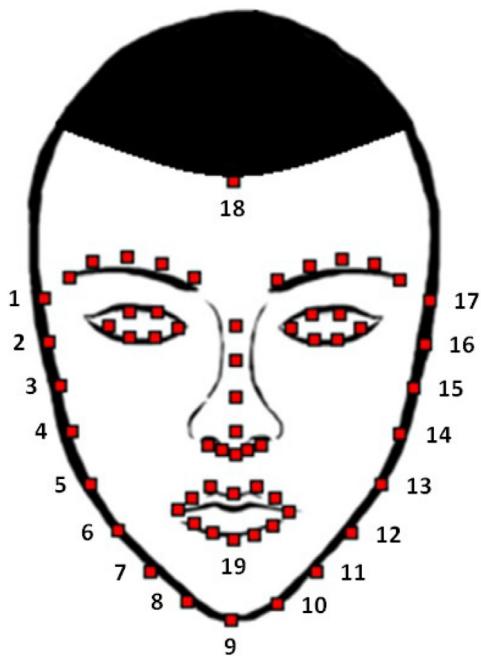
$$\mathbf{x} = [p_x^{(1)}, p_y^{(1)}, \dots, p_x^{(L)}, p_y^{(L)}] \quad (1)$$

Thus, training samples are represented as  $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_N\}$  for  $N$  number of images. Then, a statistical model of shape variation can be generated based on these samples, see (Cootes, Taylor, Cooper, & Graham, 1995) for more details. Any new images can be aligned by the following equation:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{U}_x \mathbf{p}_x, \quad (2)$$

where  $\bar{\mathbf{x}}$  is the mean shape calculated by  $\frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$ ,  $\mathbf{U}_x$  is an orthogonal matrix that represents a set of eigenvectors of point distribution calculated by Principal Component Analysis, and  $\mathbf{p}_x$  is a weight vector of the model.

In addition, the last land mark point—the top of the forehead—is processed by colour-based skin segmentation that has an ability to partition human skin regions. Hence, we can define the hairline and obtain the point that represents the top of the forehead. This point is required because it is a substantial complement to calculating the height of the face—the distance between the top of the



**Fig. 3.** Facial landmark points that are generated by AAM and colour-based skin segmentation.

forehead and the chin—that is an indispensable factor for identifying the face shape. It should be noted that the AAM adopted in our work was not able to obtain the top position of the forehead directly. Unfortunately, this approach cannot identify face images that come with fringe hair cuts. With this limitation, we can instruct a user to clearly show her forehead in practice. In this experiment, this position is manually defined by human.

AAM and skin segmentation techniques were already implemented by Saragih, Lucey, and Cohn (2009) and Mustafa (2007). Their source codes are also publicly available on the Internet<sup>2</sup>. AAM algorithm selected 60 points in an image of a face that outline the features of the face clearly and one additional point identified by colour-based skin segmentation. Therefore, there are 61 points in total as shown in Fig. 3. Therefore, we can compute the 19 geometric features based on 61 points as a feature vector. These features can be described and formulated as follows:

1. Ratio of the height of a face to the width,

$$F_1 = \frac{\|\mathbf{p}^9 - \mathbf{p}^{18}\|}{\|\mathbf{p}^1 - \mathbf{p}^{17}\|} \quad (3)$$

2. Ratio of the distance between both sides of the jaws to the width of the face,

$$F_2 = \frac{\|\mathbf{p}^5 - \mathbf{p}^{13}\|}{\|\mathbf{p}^1 - \mathbf{p}^{17}\|} \quad (4)$$

3. Ratio of the distance between the chin and the bottom of the mouth to the distance between both sides of the jaws,

$$F_3 = \frac{\|\mathbf{p}^9 - \mathbf{p}^{19}\|}{\|\mathbf{p}^5 - \mathbf{p}^{13}\|} \quad (5)$$

4. Angle between the straight line from the boundary of the face at a considered point to the chin point and the horizontal vec-

tor,

$$F_i = \tan^{-1} \left( \begin{cases} \frac{\|\mathbf{p}_y^{(i-3)} - \mathbf{p}_y^9\|}{\|\mathbf{p}_x^{(i-3)} - \mathbf{p}_x^9\|}, & i = 4, \dots, 11 \\ \frac{\|\mathbf{p}_y^{(i-2)} - \mathbf{p}_y^9\|}{\|\mathbf{p}_x^{(i-2)} - \mathbf{p}_x^9\|}, & i = 12, \dots, 19 \end{cases} \right) \quad (6)$$

These 19 feature representations are a kind of hand-crafted features that are prominent characteristics. This type of features with prior knowledge spontaneously contributed to the model has an ability to cope with over-fitting problem especially on small data set. However, lack of domain expertise often leads to a poor representation and prediction performance.

### 2.2.2. Deep-learned feature

On the other hand, deep-learned feature—generated by deep learning algorithm—does not require strong expertise in a particular domain to generate. The same structure of deep learning algorithm can be used to generate features for different tasks. Deep-learned feature can be extracted from instances by a model that consists of many computational layers that transform low-level features or raw data to higher-level or more meaningful features. In general, deep model can be utilised to perform as feature extractor as well.

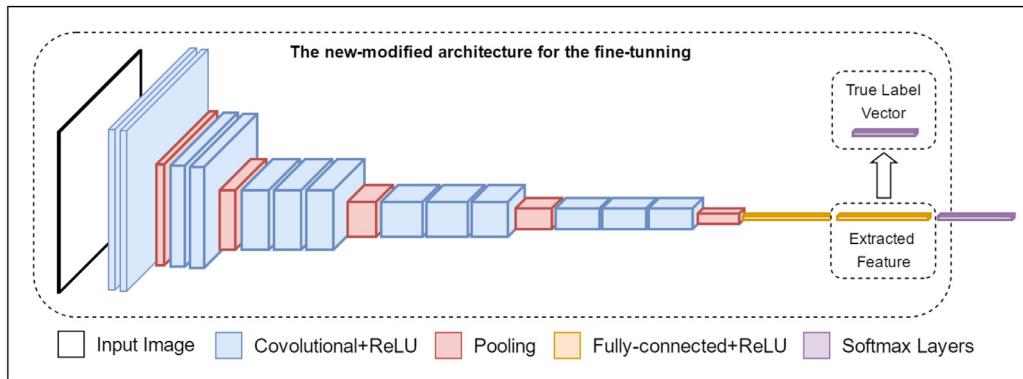
A widely used and powerful deep learning method for image analysis is CNN. To exploit this algorithm as a feature extractor, it is required to start with defining a deep architecture similar to the process of building a deep classifier and training it. Outputs from any layers of CNN can be used as represented features for each instance (Sharif Razavian et al., 2014). However, this procedure is not suitable when the data set contains a small number of samples. The deep model trained with a small-sample-size data set from scratch has a high risk of encountering an over-fitting problem that can lead to poor representation because it is only fit to the training data. Hence, using an existing model that was built to achieve similar task in the past is more practical than training a model from scratch. The pre-trained model could also be used as a generic feature extractor for other kinds of tasks.

In this work, due to a limited number of samples, the VGG-Face CNN descriptor (Parkhi et al., 2015) was exploited. The descriptor was generated by a very deep CNN model—VGG-16 (Simonyan & Zisserman, 2014)—that was previously proposed to accomplish face recognition task. The model contained three fully connected layers as the last three blocks. The output from the second fully connected layer was selected as a 4,096-D representative feature vector<sup>3</sup> (Parkhi et al., 2015). Prior to the feature extraction, face alignment techniques—AAM and skin segmentation—were used for precise detection of face area. In addition, not only the VGG-face descriptor was directly employed, but fine-tuning technique was also incorporated for investigation.

In the fine-tuning process, the last fully connected layer in VGG-16 was removed. Thus, the output of the second fully connected layer was directly connected to a softmax layer to perform five-class classification on the face shape classification task as shown in Fig. 4. After that, the model was re-trained to fine-tune the parameters with stochastic gradient descent to our data set. The fine-tuned framework is explained in more details in Section 3. Using a pre-trained deep architecture to generate feature representation of instances is a fascinating choice for dealing with issues caused by modelling with a small-sample-size data set by deep algorithms. However, it is not every new task on which any available pre-trained models are able to perform, even though they are generic feature extractors. Thus, it is not guaranteed that using a

<sup>2</sup> AAM is available at <https://github.com/kylemcdonald/FaceTracker> and Skin Segmentation is available at <https://www.mathworks.com/matlabcentral/fileexchange/13716-face---eye-detection>.

<sup>3</sup> The last fully connected layer in the very deep CNN model was to perform 2622-way classification in their task.



**Fig. 4.** Modified VGG-16 architecture which provides a fine-tuned VGG-face CNN descriptor on face shape classification task.

pre-trained model can provide the most suitable features to the task because it has been built for a different purpose. Although employing a fine-tuning strategy is an effective choice but too many re-training iterations might lead to an over-fitting problem of the model similar to training the model from scratch. Therefore, these aforementioned limitations ought to be taken into account prior to using the deep model.

After that, these features including geometric, VGG-face, and fine-tuned VGG-face features would be used and compared. Not only these features were independently formulated, but feature combination approaches would be also applied to enhance the performance such as simple vector concatenation and MKL techniques. SVM was used as a base classifier. Moreover, PSO technique was used to obtain optimal models. More details on the experimental framework are presented in [Section 3](#).

To summarise, the objective of the hand-crafted feature is to illustrate the potential of prior knowledge inspired by the specific domain expertise—face shape identification ([Fowler, 2015](#)). The key characteristics of face shape classification are geometric properties, e.g. ratio of distances between points or angles. These features are constructed with a requirement of explicit programming to extract useful patterns from raw data. Thus, this set of features can be represented as a distinctive set of features for any machine learning methods. We believe that the generic properties of the hand-crafted feature are the primary factor for face shape classification because it is carefully designed based on the information from a domain expert. However, it might not be possible to manually obtain all discriminative features for this task. Therefore, deep-learned feature might contain some important features that are not defined in the hand-crafted feature set. A deep-learned feature—VGG-face descriptor—is originally constructed for face recognition task that is not very different to our task. Thus, it has a high chance to contain some discriminative features that can benefit our task and improve the performance as well. Although, it cannot be guaranteed that hand-crafted and deep-learned features will always be complementary to each other, but there are several pieces of evidence on how well they can be used together, e.g. ([Wang et al., 2016; Wu et al., 2016](#)), as mentioned in the introduction.

### 2.3. The hairstyle recommender system design

One of the goals of this study is to generate a face shape classifier because it is a necessary component for a hairstyle recommendation system. A classifier can identify human face shapes that correspond to a set of suitable hairstyles. These suitable hairstyles will be listed for a user. The operation of our hairstyle recommender system consists of three main steps as follows: (i) Getting an image of the user's from a camera, (ii) Classifying the face into the

right shape, and (iii) Hairstyle retrieval based on the face shape and superimposing it on her face.

#### 2.3.1. Getting an image of a user's face from a camera

The first step is to capture an image of the user's face for further processing. The system starts with displaying her face on the screen while she is in front of the camera. In order to ensure the user that the system can detect her face correctly, a bounding box is displayed around her face as shown in [Fig. 5\(a\)](#). Any one of powerful face detection techniques can be employed to do this part—Haar cascade based classifier was employed ([Lienhart & Maydt, 2002](#)) in this framework. This step is very important as a good input quality is required. Poor face detection can lead to poor face shape classification. We used a face detection technique instead of a localisation technique—AAM—because AAM is too slow to locate facial area in real time monitoring. After the face was detected, the user is required to capture it. It is vital to confirm the quality of localisation by overlaying the landmarks on the captured photo as shown in [Fig. 5\(b\)](#). If the quality was not good, the user shall go back to the previous step and retake a new photo.

#### 2.3.2. Classifying the face into the right shape

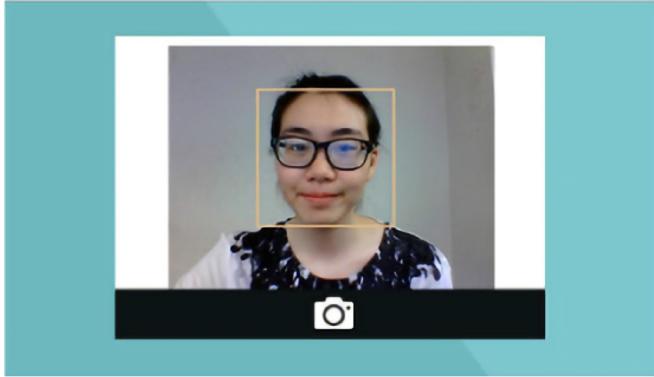
Defining women's face shape is the main purpose of this study. We had to generate a dedicated classifier to undertake this role. In both training and inferencing processes, the classifier required a representation of each instance as a feature vector mentioned in [Subsection 2.2](#). In this work, SVM was used as a base classifier. The proposed framework to define the classifier are explained in the next section. After the model returned the predicted face shape, based on the prediction result, a list of proper hairstyles shall be retrieved according to the expert's guideline.

#### 2.3.3. Performing appropriate hairstyle retrieval and superimposing it on her face

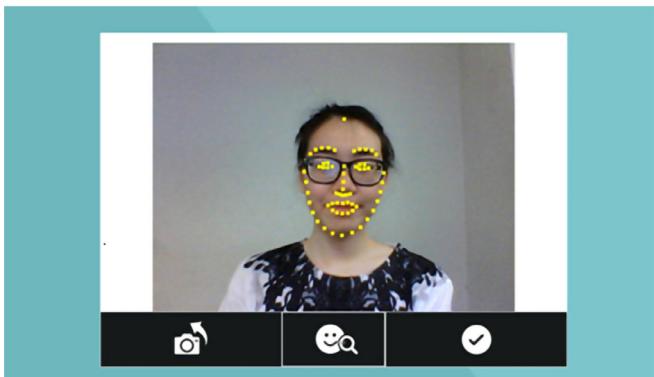
According to Derick's and Hong's proper hairstyle guidelines ([Derrick, 2017; Hong, 2013](#)), any hairstyles can be picked off the shelf but they should be considered according to the guidelines whether or not they should be worn. This process can be done after the face shape was already identified by a classifier based on an input face. According to the guidelines, there are four main attributes which are very important to describing hairstyle as follows:

- i) Length = {pixie, short, mid-length, long};
- ii) Style = {straight, wavy, mix};
- iii) Bang = {none, blunt, side-swept};
- iv) Layered = {non-slide, slide}.

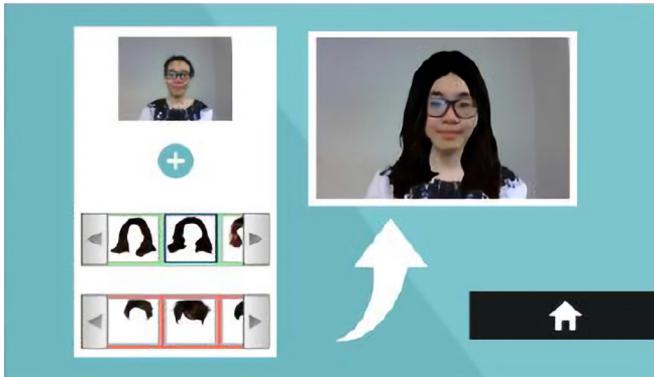
The above attributes can be easily judged by human without ambiguity and are shown in [Fig. 6](#).



(a) An image of the user's face with a bounding box around her face.



(b) Display all the landmarks overlaid on the photo to the user.



(c) Lists of proper (green) and improper (red) hairstyles for the user to consider and simulation of the selected hairstyle on the user's face.

**Fig. 5.** User interface of the proposed system.

The appropriateness of any hairstyles to the face shape can be judged from these attributes based on the recommendation guidelines as decision rules that can be encoded in **Algorithm 1**. The algorithm will return whether an input hairstyle is recommended or not recommended or inconclusive with the face shape. Thus, the

**Algorithm 1** Determine the suitability of one of the hairstyles in the data store to the input face shape.

---

```

1: procedure IsItSUITABLE
2:   faceShape  $\leftarrow$  the shape identified by the classifier according
3:    $\leftarrow$  to the faceimage
4:   hairStyle  $\leftarrow$  a sample of hairstyle with its attributes
5:   [length, style, bang, layered]  $\leftarrow$  hairStyle
6:   if faceShape = round then
7:     if length = midLength OR (length = long AND style = mix)
8:     OR layeres = slide OR bang = sideSwept then
9:       return Recommend
10:    else if bang = blunt OR (layered = slide AND length  $\neq$ 
11:    midLength) OR (length = short AND style = wavy) then
12:      return Not-Recommend
13:    else if faceShape = oval then
14:      if length = midLength OR bang = sideSwept OR style = mix
15:      then
16:        return Recommend
17:      else if length = pixie OR (length = short AND style =
18:      wavy) then
19:        return Not-Recommend
20:      else if faceShape = oblong then
21:        if bang  $\neq$  none OR style = wavy OR (length = short AND
22:        layered = non) then
23:          return Recommend
24:        else if length = pixie OR bang = none OR (length = long
25:        AND style = straight) then
26:          return Not-Recommend
27:        else if faceShape = squared then
28:          if style = wavy OR bang = sideSwept OR layered = slide
29:          OR (length = midLength AND layered = non) then
30:            return Recommend
31:          else if bang = blunt OR length = short OR length = pixie
32:          OR (length = long AND style = straight) then
33:            return Not-Recommend
34:          else if faceShape = heart then
35:            if style = wavy OR bang = sideSwept OR (length = long
36:            AND style = mix) OR (length = short AND style = mix) then
37:              return Recommend
38:            else if bang  $\neq$  sideSwept OR length = pixie OR layered =
39:            slide then
40:              return Not-Recommend
41:            return Inconclusive

```

---

system can simply create a list of proper and improper hairstyles for each face shape. Moreover, it can simulate how the user looks like while she is wearing the chosen hairstyle recommended by the system as shown in Fig. 5(c). Hence, the user can decide whether to go with the recommendation or not based on her look presented on the screen. In order to do so, each hairstyle is manually annotated with four points as reference points for mapping it to the user's face. These points are at the forehead, the left ear, the right ear and the chin as shown in Fig. 7. These points corre-



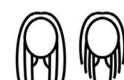
(a) Length: pixie, short, mid, long



(b) Style: straight, wavy, mix, layered



(c) Bang: none, blunt, side swept



(d) Layered: non-slide, slide

**Fig. 6.** Hairstyle's attributes.



**Fig. 7.** Reference points of hairstyle mapping onto the face.

sponded to the points given by AAM in the face image, which were 18, 1, 17 and 9, respectively, as shown in Fig. 3.

### 3. Experimental design

In the previous section, the overall concept for constructing a hairstyle recommendation system is examined including the description of face shape data set, two primary types of features that are likely to have the capability to distinguish women's face shape, and the design for a hairstyle recommender system. However, the staple of this study is to generate a face shape classifier. Therefore, constructing a good performance face shape classifier was our focus in the first place. The proposed framework is shown in

**Fig. 8.** The following subsections describe the framework to achieve a good model in this study.

#### 3.1. Data splitting criteria

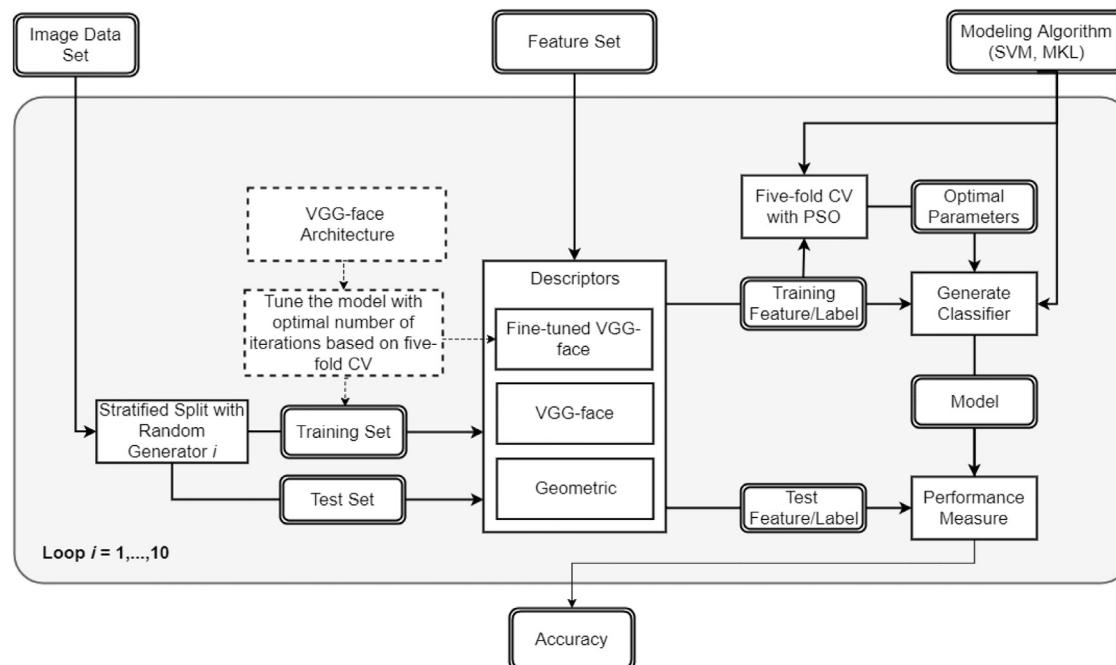
The data were divided into two sets which were training and test sets. The splitting ratio was 70:30 for training to test sets. Machine learning algorithms usually have a number of hyperparameters to be tuned. In order to achieve an optimal performance from them, a technique called "grid search" is usually applied to obtain the optimal parameters. Hence, grid size and number of parameters directly affect computational cost for obtaining optimal results. In other words, if the grid has a high resolution, more computational time is required. This is applicable to the number of parameter as well. Therefore, one of the most popular optimisation algorithms inspired by nature called "Particle Swarm Optimisation" (PSO) was used to search for an optimal solution. The parameter search procedure was operated only on the training set while the test set was used for validating the model. In order to avoid the over-fitting problem and getting a good generalised model, five-fold cross validation was applied. In order to get plausible results, the data set was divided into training and test sets 10 times with a different random seed each time. This led us to be able to statistically confirm the results.

#### 3.2. Facial feature representations

In this study, there were two types of features that were introduced to differentiate the shape of women's face, i.e. hand-crafted and deep-learned features. In addition to these features, a fine-tuning technique was applied to adjust the deep-learned features for them to be more suitable for the data set. Thus, these choices of features were evaluated on a well-designed framework.

##### 3.2.1. Individual feature representations

i) Geometric feature descriptor consists of hand-designed features constructed from landmark points obtained by AAM and skin segmentation as described in Subsection 2.2.1. One of the prominent attributes of hand-crafted features is its informative representation that can be simply interpreted. It is designed



**Fig. 8.** The overview of the proposed experimental framework.



**Fig. 9.** Examples of pre-processed face images prior to their getting fed into our deep model.

based on prior human-domain knowledge. Therefore, using this type of feature makes more sense to humans than using automatic feature extraction approaches.

- ii) VGG-face feature descriptor is the deep-learned facial feature generated by CNN. To represent each image as a feature vector, all images were resized following the VGG-16's architecture at  $224 \times 224 \times 3$  for width, height and colour-dimensional space, respectively. However, the ratio of the facial dimensions is a vital information that needs to be preserved. Thus, the original image needs to be resized and the aspect ratio of the face needs to be preserved. This can be done by scaling the height of the original image to be 224 pixels while the width was resized to preserve the aspect ratio. After that, to obtain a square image, the image was set at the centre and its left and right sides were filled with black colour pixels as shown in Fig. 9.

Usually, a pre-trained model provides a high-level features resulting from using network-stacked layers. The trained network can be used with any data sets which have some mutual information with the data set that it was trained from. Utilising the pre-trained model as a feature extractor can work well with a small-sample-size data set. In the case that we train a model on a small-sample-size data set without using a pre-trained model, the trained model will suffer from an over-fitting problem.

- iii) Fine-tuned VGG-face feature is a deep-learned feature that was carried out through a fine-tuned strategy by further training the model with our face shape data set. It should be noted that the model was retrained with a small number of epochs only on the training set—the test set was blinded—for each random seed. The model could be over-fitted if it was retrained with many iterations. An optimal number of tuning epochs was obtained by performing a grid-search strategy with a range of {2, 4, 8, 16} on the basis of accuracy of five-fold cross-validation on the training set. Many studies have employed a fine-tuning technique to modify their model parameter (Ng, Nguyen, Vonikakis, & Winkler, 2015; Shin et al., 2016; Wang, Li, Gupta, & Yeung, 2015). As a result, the latter model may be very effective for a new task. Again, this may lead to an over-fitting problem if a small-sample-size data set is used. Thus, it cannot be guaranteed that employing a fine-tuned network will be more effective than using the original trained one.

Prior to the model training phase, data were normalised by using z-normalisation to alter all attributes into the same distribution as shown below:

$$\mathbf{z}_i = \frac{\mathbf{x}_i - \boldsymbol{\mu}}{\sigma}, \quad (7)$$

where  $\mathbf{x}_i$  is a feature vector of the  $i$ -th sample;  $\boldsymbol{\mu}$  and  $\sigma$  are the mean and standard deviation vectors containing the average and the standard deviation for each attribute. Both of them were calculated from only the samples in the training set.

### 3.2.2. Combining feature representations

As mentioned in Subsection 2.2 that hand-crafted and deep-learned features can be complementary to each other, hence combining them are likely to improve the overall performance. This paper proposes the following hybrid feature-based approach.

**Feature vector concatenation.** Feature vector concatenation is the simplest method for combining two or more features from different sources together and treating them in the same feature space. Learning from more than one source usually gains a higher performance than doing so from an individual source. Features from different sources is given equal importance in this technique. This concatenation technique may not perform well with modern data-high dimensionality and rich structural information (Li, Wu, & Ngom, 2016).

**Multiple Kernel Learning (MKL).** MKL is one of the most powerful feature combination approaches based on kernel technique concept. It aims to combine features from multiple sources into one optimal feature in a principal way. It first computes a kernel matrix for each source and generates a new kernel matrix that combines each kernel matrix with a different weight. This research utilised  $p$ -norm MKL based on Elastic-MKL proposed in Pasupa et al. (2013) that allowed us to tune the level of sparsity required for the choice of kernels.

Let  $S = \{(\phi(\mathbf{x}_1), y_1), \dots, (\phi(\mathbf{x}_m), y_m)\}$  denote a set of samples where  $\phi$  is a mapping function that maps points to a higher dimensional feature space,  $\phi(\mathbf{x}) \in \mathcal{R}^n$ ,  $y_i \in \{-1, +1\}$ . These points are compared by using a dot product. Then, the *kernel trick* is used in order to avoid working in the high dimensional feature space. Therefore, the dot product can be evaluated using a non-linear function in the input space.

$$k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle \quad (8)$$

where  $\langle \mathbf{x}, \mathbf{x}' \rangle$  is a dot product between two vectors,  $\mathbf{x}$  and  $\mathbf{x}'$ .

Let  $\kappa$  be a number of feature sets from different sources or a number of considered kernels. In MKL approach, a combined kernel is a linear combination of weighted individual kernels. An individual kernel is weighted by  $\lambda_i$ . Therefore, a combination of these kernel can be formulated as:

$$k_{\text{combine}}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{\kappa} \lambda_i \cdot k_i(\mathbf{x}, \mathbf{x}'), \quad (9)$$

where  $k_i$  is an  $i$ th based kernel.

According to the fundamental principle of SVM optimisation, if  $k$  is a valid kernel function, the SVM classifier can be generated with dual formulation optimization as:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,i'=1}^m \alpha_i \alpha_{i'} y_i y_{i'} k(\mathbf{x}_i, \mathbf{x}_{i'}) \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \quad 0 < \alpha < C, \end{aligned} \quad (10)$$

where  $C$  is a hyper-parameter that controls a trade-off between misclassification rate and model complexity. However, in MKL, there is an addition parameter  $\lambda_i$  that weights each  $i$ th based kernel. Therefore, this parameter shall be considered in the above SVM optimisation problem (Zheng-Peng & Zhang, 2011) resulting in the following optimisation problem:

$$\max_{\alpha, \lambda} \quad \sum_{i=1}^m \alpha_i - \frac{1}{2} \left( \frac{\sum_{i,i'=1}^m \alpha_i \alpha_{i'} y_i y_{i'} (\sum_{j=1}^{\kappa} \lambda_j \cdot k_j(\mathbf{x}_i, \mathbf{x}_{i'}))}{\sqrt[p]{\sum_{i,i'=1}^m \alpha_i \alpha_{i'} y_i y_{i'} (\sum_{j=1}^{\kappa} \lambda_j \cdot k_j(\mathbf{x}_i, \mathbf{x}_{i'}))}} \right)^p$$

$$\text{s.t. } \sum_{i=1}^m \alpha_i y_i = 0, 0 < \alpha < C. \quad (11)$$

where  $\alpha_i$  is a Lagrange multiplier. It is clearly seen that, while the parameter  $C$  controls the trade-off between achieving a low error on the training data and minimising the norm of the weights,  $p$ -norm acts as a hyper-parameter to shape the distribution of a vector which is composed of multiple kernel weights  $\lambda$ . After the training process has finished, we can obtain  $\lambda$  and  $\alpha$  as weights of kernels and support vectors, respectively. A decision function for unseen data can be formulated as:

$$f(\tilde{\mathbf{x}}) = \text{sign} \left( \sum_{i=1}^m \left[ \alpha_i y_i \sum_{j=1}^K \lambda_j \cdot k_j(x_i, \tilde{\mathbf{x}}) \right] + b \right) \quad (12)$$

where  $\tilde{\mathbf{x}}$  is an unseen instance vector and  $b$  is an bias value in SVM. As the SVM is a binary class classification method, to enable it to cope with multi-class classification problem, *one-vs-one* technique was employed.

To sum up, there are three parameters in MKL that need to be tuned: (i)  $p$ -norm that controls  $\lambda$  for each kernel, (ii)  $C$  that controls the trade-off between error rate and norm of the support vectors' weight, and (iii) a kernel parameter  $\gamma$  in RBF function. The following subsection explains on how to optimally select these parameters.

Moreover, there are three types of descriptors available, i.e. geometric, VGG-face and fine-tuned VGG-face descriptors. These primary descriptors can also be combined by two techniques—feature vector concatenation and MKL—into all possible combinations of descriptors. Hence, in addition to the three individual descriptors, there can be three combinations of two-feature representations and one combination of all three-feature representation for each feature combination technique. These were evaluated with SVM.

### 3.3. Data modelling

According to preliminary results reported in our recent work (Sunhem & Pasupa, 2016), SVM in conjunction with RBF kernel is the best contender for face shape classification task that can achieve the highest performance compared to other candidates. It is one of the most powerful machine learning techniques based on instance-based learning concepts. In this work, we focused on evaluating different types of features. Therefore, SVM with RBF kernel were selected as the base classifier.

This paper presents a well-designed experimental framework for evaluating the following types of descriptors:

1. Individual descriptors: Geometric descriptor ( $D_G$ ), VGG-face descriptor ( $D_V$ ), Fine-tuned VGG-face descriptor ( $D_F$ ).
2. Combined descriptors with vector concatenation technique:  $\text{concat}(D_G, D_V)$ ,  $\text{concat}(D_G, D_F)$ ,  $\text{concat}(D_V, D_F)$ , and  $\text{concat}(D_G, D_V, D_F)$ .
3. Combined descriptors with MKL (MKL):  $\text{MKL}(D_G, D_V)$ ,  $\text{MKL}(D_G, D_F)$ ,  $\text{MKL}(D_V, D_F)$ , and  $\text{MKL}(D_G, D_V, D_F)$ .

The experiment was conducted 10 times with different random splits into training and test sets. For each random seed, three primary descriptors and eight combined descriptors were considered together and compared with each other. In order to get an optimal performance with SVM, the hyper-parameters were required to be tuned as follows:

1. Individual features: there were only regularisation parameter  $C$  and RBF function parameter  $\gamma$  that had a range of  $[10^{-6}, \dots, 10^6]$ .
2. Combined features by vector concatenation technique: the settings were similar to the those of the individual feature cases.

	Round	214 14.3%	27 1.8%	0 0.0%	56 3.7%	28 1.9%	65.8% 34.2%
Output Class	Oval	18 1.2%	142 9.5%	28 1.9%	29 1.9%	82 5.5%	47.5% 52.5%
	Oblong	0 0.0%	37 2.5%	237 15.8%	2 0.1%	35 2.3%	76.2% 23.8%
Square	Round	60 4.0%	25 1.7%	7 0.5%	208 13.9%	14 0.9%	66.2% 33.8%
	Heart	8 0.5%	69 4.6%	28 1.9%	5 0.3%	141 9.4%	56.2% 43.8%
	Round	71.3% 28.7%	47.3% 52.7%	79.0% 21.0%	69.3% 30.7%	47.0% 53.0%	62.8% 37.2%
	Oval						
	Oblong						
	Square						
	Heart						

(a) Geometric Descriptor ( $D_G$ )

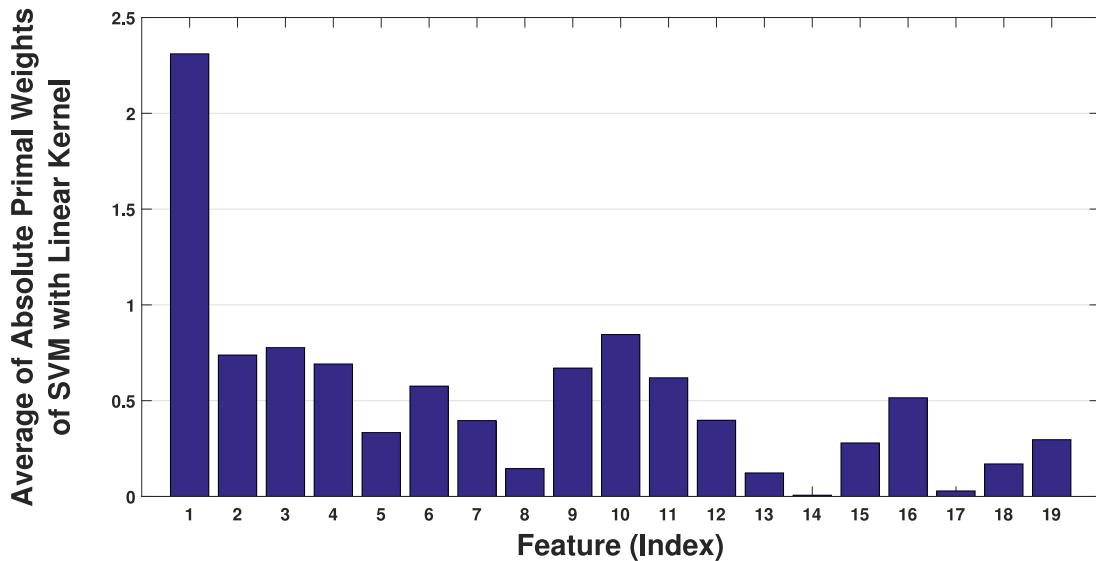
	Round	220 14.7%	53 3.5%	1 0.1%	23 1.5%	36 2.4%	66.1% 33.9%
Output Class	Oval	31 2.1%	142 9.5%	41 2.7%	23 1.5%	29 1.9%	53.4% 46.6%
	Oblong	2 0.1%	37 2.5%	169 11.3%	33 2.2%	35 2.3%	61.2% 38.8%
Square	Round	28 1.9%	23 1.5%	27 1.8%	184 12.3%	21 1.4%	65.0% 35.0%
	Heart	19 1.3%	45 3.0%	62 4.1%	37 2.5%	179 11.9%	52.3% 47.7%
	Round	73.3% 26.7%	47.3% 52.7%	56.3% 43.7%	61.3% 38.7%	59.7% 40.3%	59.6% 40.4%
	Oval						
	Oblong						
	Square						
	Heart						

(b) VGG-face Descriptor ( $D_V$ )

	Round	233 15.5%	55 3.7%	2 0.1%	34 2.3%	19 1.3%	67.9% 32.1%
Output Class	Oval	24 1.6%	146 9.7%	46 3.1%	17 1.1%	52 3.5%	51.2% 48.8%
	Oblong	4 0.3%	38 2.5%	187 12.5%	19 1.3%	40 2.7%	64.9% 35.1%
Square	Round	28 1.9%	12 0.8%	10 0.7%	194 12.9%	21 1.4%	73.2% 26.8%
	Heart	11 0.7%	49 3.3%	55 3.7%	36 2.4%	168 11.2%	52.7% 47.3%
	Round	77.7% 22.3%	48.7% 51.3%	62.3% 37.7%	64.7% 35.3%	56.0% 44.0%	61.9% 38.1%
	Oval						
	Oblong						
	Square						
	Heart						

(c) Fine-tuned VGG-face Descriptor ( $D_F$ )

**Fig. 10.** Confusion matrices of face shape predictions on test sets—sum of 10 runs with different random splits performed with three types of features.



**Fig. 11.** The average value—averaged across 10 runs—of the absolute weight vectors of SVM with linear kernel for separating oblong class from round-and-square class.

3. Combined features by MKL: the settings were similar to those of the first two cases but there was an additional parameter  $p$ -norm which was in a range of (0,2]. It should be noted that  $\gamma$  for each RBF kernel can be tuned independently. According to this fact, the number of  $\gamma$  is equal to the number of feature sources.

After the searched range of the parameters were defined, the search could be operated by a grid search technique. It can be seen that there were a large number of tuning parameters, hence this could lead to a high computational cost. Therefore, we utilised PSO to search for the parameters in the search space. The PSO's objective function was an average error rate of five-fold cross-validation in the training set. In addition, there were a few PSO parameters that were needed to be taken into consideration: the number of particles was equal to 100 and the maximum number of iterations was set to 50. After the optimal parameters had been obtained, they were used to train the model again with the entire training set. Then the model was evaluated with the test set.

It can be seen that a considerable number of methods were employed to achieve the ultimate research goal, i.e. SVM, MKL, PSO. Nonetheless, every method comes with strengths and weaknesses which are as follows:

1. SVM is one of the most popular machine learning techniques. It is a non-parametric model. Compared to other popular learning techniques in computer vision such as Artificial Neural Network, it has a distinct ability to prevent over-fitting problem. Moreover, it can be defined by a convex optimisation problem. However, it requires high human intervention to determine the optimum parameter settings such as regularisation parameter, choice of kernel and its parameter.
2. MKL is a technique that combines multiple sources of feature. Unlike a baseline-vector concatenation technique, each feature can obtain their suitable kernel independently. It can automatically assign weight to each source. This approach can benefit the resolution of a problem with heterogeneous data set (Pasupa et al., 2013). However, if the data are considerably homogeneous, this technique is most likely to be too sophisticated.
3. As there are many parameters that are required to be tuned to get an optimal model, PSO is selected for this task. In general, grid search is usually applied to perform the task, however, the technique is time-consuming especially in the case of a large

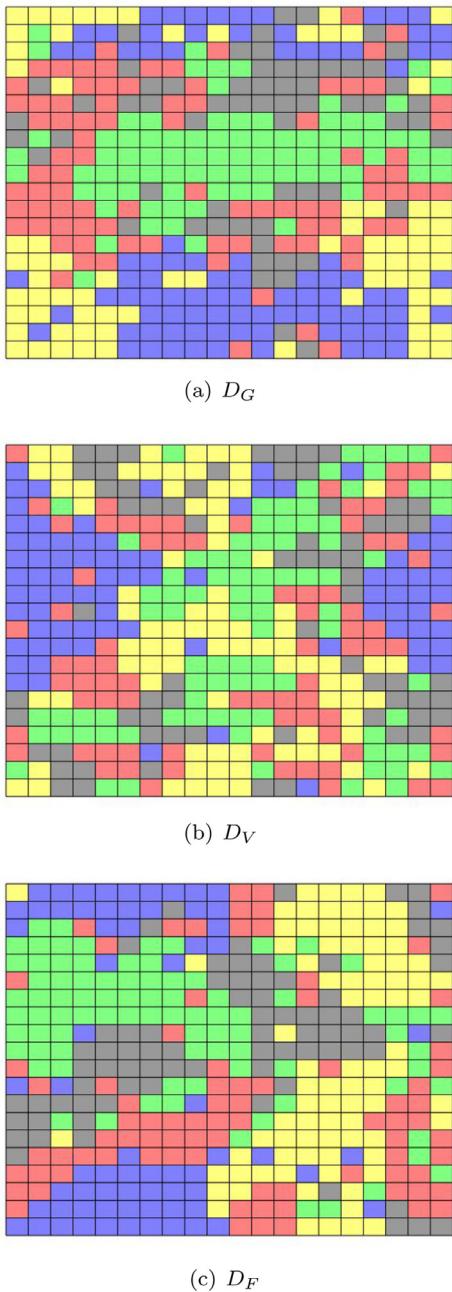
number of parameters and/or a large parameter-search space. Although, utilising PSO can reduce the complexity of the search task, there is no guarantee that global optimum solution will be achieved.

#### 4. Experimental results and discussion

The fact that the data set was perfectly balanced makes the evaluation in terms of accuracy sufficient to describe how effective each classifier performed. Moreover, statistical analysis was applied in order to obtain the significance level of the results.

##### 4.1. Analysis of individual features

Firstly, we investigated three individual features:  $D_G$ ,  $D_V$ , and  $D_F$  results and reported them with confusion matrices shown in Fig. 10. Confusion matrix illustrates the number of correct and incorrect predictions made by a model against actual classes. It is clearly seen that  $D_V$  was the weakest descriptor as it was only able to achieve 59.6% accuracy. This is because this descriptor was extracted from an extractor that was composed by another data set and has never trained with our data set. After performing the fine-tuning strategy, the parameters in the extractor were adjusted based on our data set—resulting in  $D_F$ , the classification performance was enhanced to 61.9%. However, these two deep-learned descriptors were still worse than the hand-crafted feature which was able to perform with 62.8% accuracy. Hence, it can be inferred that coping with a particular task with a specific well hand-crafted feature extraction is also a practical way to obtain good features from a data set that has a small number of samples. We further performed a statistical analysis on the results with *t*-test. It is confirmed that  $D_G$  was able to perform better than  $D_V$  at  $p < 0.005$  but it is inconclusive that the best contender,  $D_G$ , was better than  $D_F$  ( $p = 0.586$ ). As can be seen in Fig. 10(a), there are two pairs of face shapes that mainly affected the misclassification rate of the model, namely, round-vs-square and oval-vs-heart—one shape was often mislabelled as the other. The  $D_G$  might have not performed well or might not be able to distinguish between these shapes. Fortunately, it was able to classify oblong face shape against the other shapes resulting in 79.0% accuracy for oblong class. Moreover, it was able to absolutely separate oblong shape from round shape with no error and from square shape with a relatively small error. Therefore, we further evaluated the importance of the proposed



**Fig. 12.** The Kohonen maps generated by CP-ANN on the training sets with individual descriptors. They illustrate the spatial variation of each class: (i) round (blue), (ii) oval (red), (iii) oblong (green), (iv) square (yellow), and (v) heart (grey). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

features in separating oblong class from round-and-square class as shown in Fig. 11. This was performed by an SVM model with a linear kernel and its optimal parameter. Obviously,  $F_1$ —the ratio of the height of a face to the width—had the highest magnitude, at least 2.5 times larger than those of the other features. This implies that  $F_1$  had the highest power of discriminative feature to classify these shapes. On the other hand,  $D_V$  was able to achieve lower misclassification rates in distinguishing between round shape and square shape as well as oval shape and heart shape compared to  $D_G$  as shown in 10(b). However, the overall performance was still worse than that achieved by  $D_G$  because its ability to correctly predict oblong shape was dramatically reduced from 79.0% with  $D_G$  to 56.3%

with  $D_V$ . This is true of the case of correctly predicted square shape as well—from 69.3% with  $D_G$  to 61.3% with  $D_V$ .

According to Fig. 10(c), using  $D_F$  gave very much the same kind of results as those from  $D_G$ . In addition, using  $D_F$  gave better results in predicting round, oval, and heart shapes than  $D_G$  did.

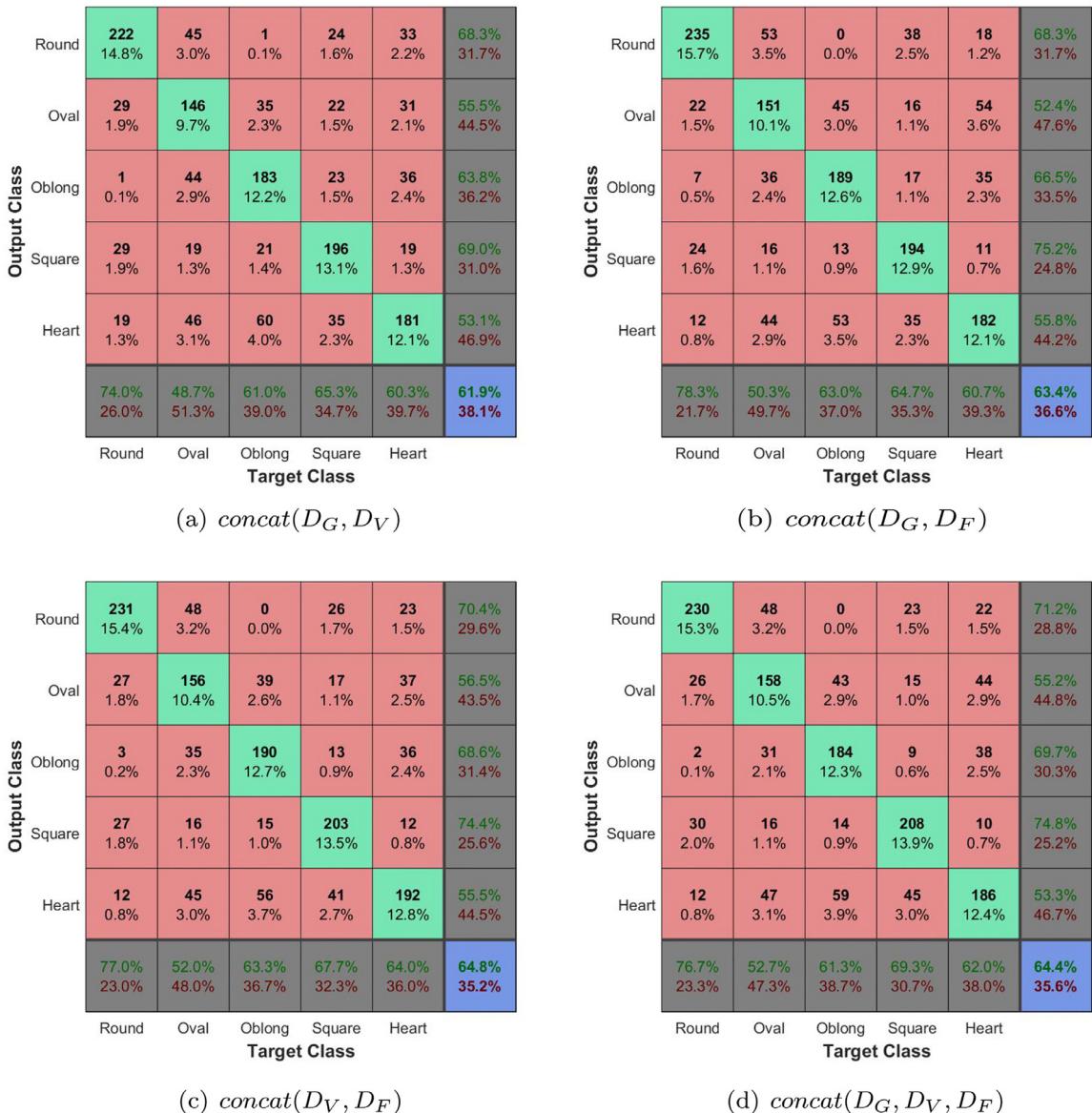
Notwithstanding, we further ran more experiments in order to gain more informative evidence that could verify our assumptions and results. Here, the Counter-propagation Artificial Neural Network (CP-ANN) was employed. It is an extension of Kohonen maps or Self Organising Map (SOM) (Ballabio, Consonni, & Todeschini, 2009) with an ability to cope with supervised learning problems. The algorithm can be used in data visualisation as the substantial spatial variation in SOM can be interpreted clearly. The map represents a mapping from a high dimensional input space into a low dimensional input space which consists of nodes. The distance between nodes quantifies the degree of difference between objects.

Fig. 12 shows a spatial variation in  $20 \times 20$  of SOM resulting from all three individual descriptors. The maps were generated by CP-ANN on the training sets in order to interpret the similarity among the shapes. It is clearly seen that oblong shape area (green) has a low degree of connection to the round (blue) and square (yellow) shapes when  $D_G$  is considered, as shown in Fig. 12(a). In other words, it is easy to separate oblong face from round and square shapes. Apart from this, it is quite complicated to distinguish between round and square faces as well as oval (red) and heart-shaped (grey) faces because there are a high degree of connections between the couples. In addition, the areas in each shape are split into several small areas and scattered around. This confirms the results in the confusion matrix in Fig. 10(a).

Nonetheless, a SOM map with  $D_V$  in Fig. 12(b) illustrates that the areas of a round face shape are formed into two larger areas, while in  $D_G$ , the areas were split into smaller areas. This means that the ability to indicate a round shape in  $D_V$  is better than that in  $D_G$ . For oblong shape, the areas are split into smaller pieces which for the  $D_V$  case, this leads to a drop in the ability to identify it. These findings give a similar picture to that in the confusion matrix given in Fig. 10(b). Finally, using  $D_F$  leads to a better looking SOM map than  $D_V$  does. The areas are grouped into larger areas. Using deep-learned features alone is worse than using hand-crafted feature in this task because the CNN model employed as the feature extractor was not built from our data set. It comes with the knowledge of face information that is related to our problem, but its domain objective is different—face recognition. Although, transfer learning technique has been successfully applied to many tasks that suffer from small number of training data samples, it might fail unless the source domain is extremely near the target domain (Tan, Zhang, Pan, & Yang, 2017). Thus, when it comes to transferring pre-learned knowledge of face recognition to face shape classification, it cannot be guaranteed that some distinctive features, e.g. geometric properties, are still captured in the source feature. The deep-learned features might lead to a poor performance in the task. However, this does not imply that there is no advantage of the deep-learned feature. The deep-learned features might effectively support the hand-crafted features which will be evaluated in the following sections.

#### 4.2. Analysis of concatenated feature

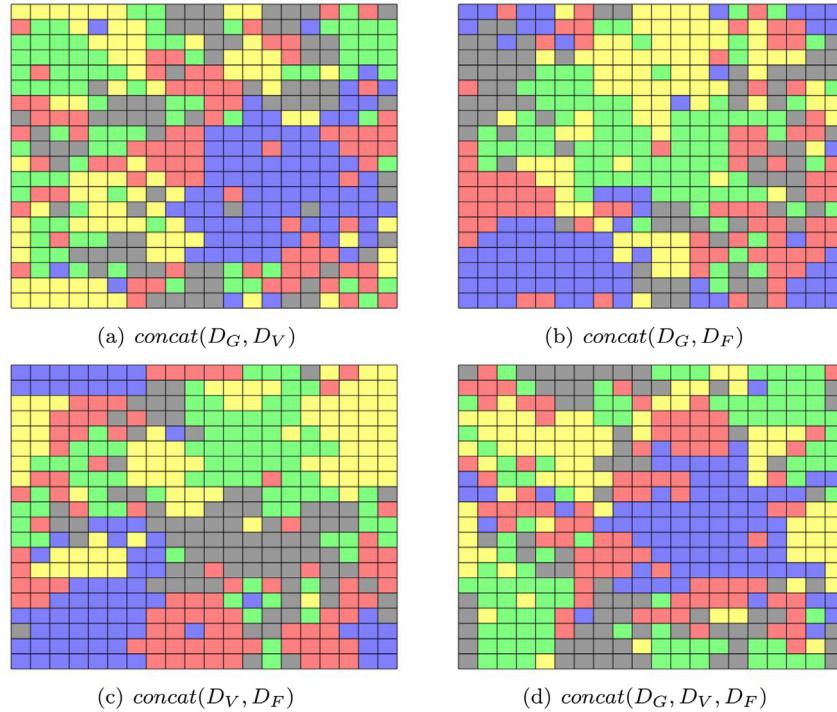
This subsection analyses four combinations of all individual descriptors with vector concatenation technique. Again, confusion matrices are shown in Fig. 13. The worst combination is  $\text{concat}(D_G, D_V)$  which drops the overall performance from using  $D_G$  alone to 61.9% accuracy. This might be because features are represented in different feature spaces and are equally treated.  $D_G$  contains only 19 geometric features while  $D_V$  is in a 4,096-D feature space. Therefore, it is improper to simply merge and consider both fea-



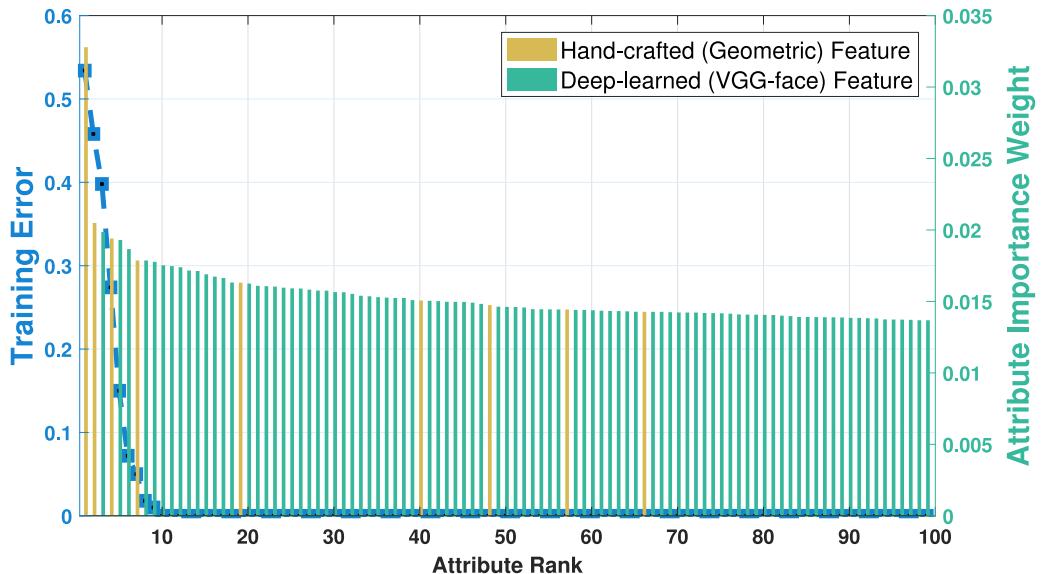
**Fig. 13.** Confusion matrices of face shape predictions on the test sets-sum of 10 runs with different random splits and performed on four combinations of features by vector concatenation technique.

tures in the same space. Besides,  $D_V$  was not optimally generated for our task, and gave a poor performance on its own. However, other combinations of features that contain the fine-tuned VGG-face descriptors illustrate higher performances at 63.4%, 64.8%, and 64.4% accuracy for  $concat(D_G, D_F)$ ,  $concat(D_V, D_F)$ , and  $concat(D_G, D_V, D_F)$ , respectively. It is clear that combining the two best features— $D_G$  and  $D_F$ —can enhance the accuracy, better than using them alone. However, the  $concat(D_V, D_F)$  gave us the best performance. It was composed of the same feature type—deep-learned feature-generated by the same CNN model. Hence, it is sensible to concatenate both features together as they are represented in the same space. Similar to the analysis of individual features, it is still difficult to classify round-against-square and oval-against-heart in all cases resulting in the confusion matrices shown in Fig. 13. CP-ANN was again exploited to visualise the distribution of the data as shown in Fig. 14. The results are very much the same as using the individual features but with clear regions in SOM for all cases—a slightly better performance than using  $D_F$  alone. To

support the assumption that hand-crafted and deep-learned features can be complementary to each other in this task, ReliefF algorithm is employed to rank attributes in  $concat(D_G, D_V)$  to see which attributes are important. ReliefF is a feature selection technique that scores each feature based on its relevance to the class (Robnik-Šikonja & Kononenko, 1997). Each attribute will be given a weight. The attributes with higher weight are considered more relevant as represented by bar graphs shown in Fig. 15. Most of the hand-crafted features are ranked in the top list. We focus on a hundred most important attributes. There are nine out of 19 attributes (47.37%) from the hand-crafted features and 91 out of 4096 attributes (2.22%) from the deep-learned features. These indicate that the hand-crafted features are the primary factor for face shape classification while some attributes of the deep-learned features are ranked after the hand-crafted ones. Therefore, the deep-learned features could be supporting factors. Further analysis is conducted by evaluating the prediction ability of SVM in conjunction with RBF kernel function on these attributes that are incre-



**Fig. 14.** The Kohonen maps generated by CP-ANN on the training sets with different combinations of features by concatenation technique. They illustrate the spatial variation of each class: (i) round (blue), (ii) oval (red), (iii) oblong (green), (iv) square (yellow), and (v) heart (grey). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

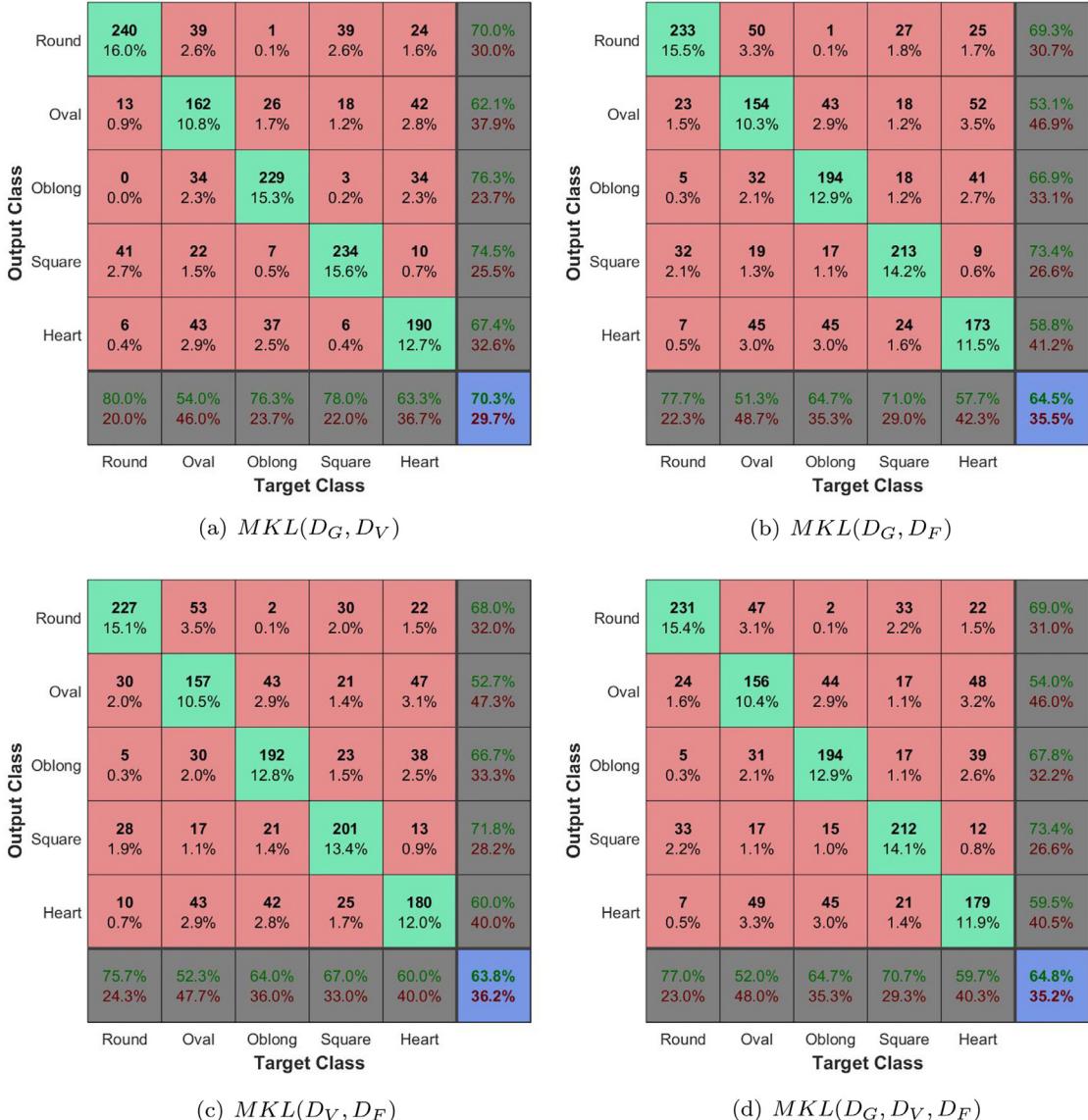


**Fig. 15.** Importance weight of the top hundred attributes of  $\text{concat}(D_G, D_V)$  given by ReliefF (bar graphs) and effect of training error when the number of attribute are incrementally accumulated according to the ranks (line graph).

mentally accumulated according to the ranks. The hyperparameters were tuned based on the training error. The prediction accuracy is represented by a line graph as shown in Fig. 15. There is a dramatic drop in training error to almost zero when only ten most important attributes are considered. These ten attributes consist of four hand-crafted and six deep-learned features. The results lead to the implication that even though the geometric feature strongly influences the predicted outcome, the deep-learned feature should be still taken into account to get a good performance.

#### 4.3. Analysis of multi-kernel learning approaches

Described in the previous subsection, a kernel trick was applied to the concatenated feature vectors which allowed us to compute the model based on a single kernel function. Since there were three descriptors, different feature characteristics might be suitable with different kernel functions or kernel parameters. Therefore, MKL was applied in order to allow us to combine all features and to use different kernels on different features as well. Moreover, each kernel would be optimally weighted by the algo-



**Fig. 16.** Confusion matrices of face shape predictions on the test sets-sum of 10 runs with different random splits and performed on four combinations of features by MKL technique.

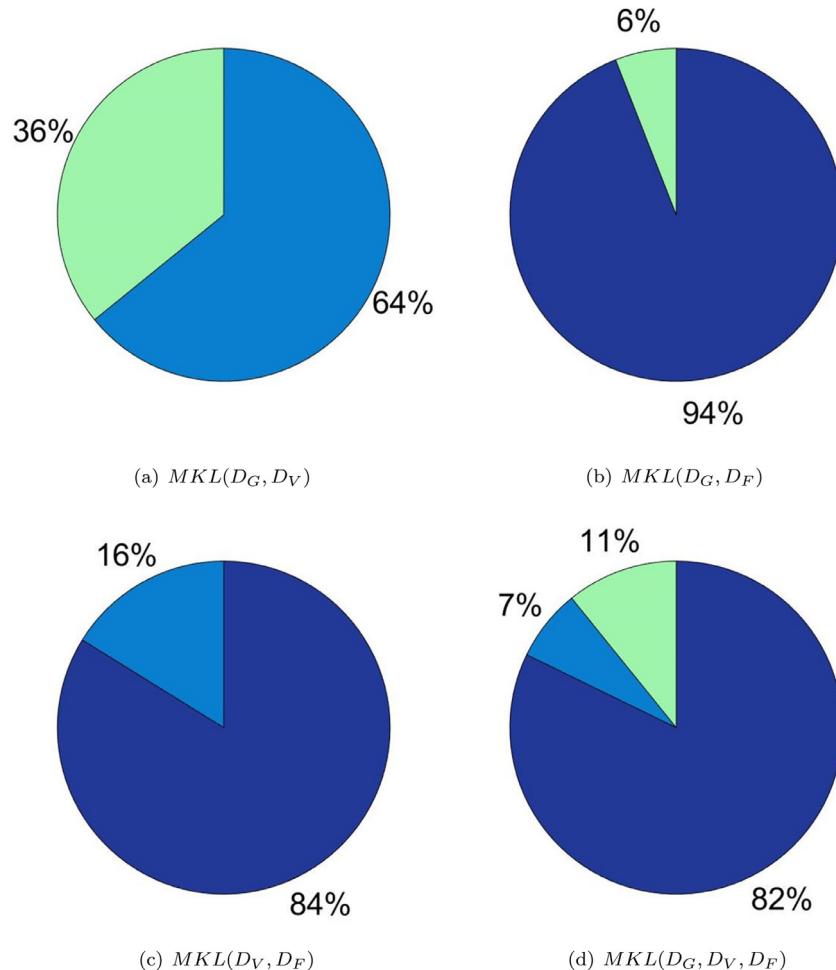
rithm. Fig. 16 illustrates the performance of MKL on different sets of the descriptors. Combined  $D_G$  and  $D_V$  was the most powerful set when they were performed with the MKL approach. It was able to achieve up to 70.3% accuracy. It is followed by  $MKL(D_G, D_V, D_F)$ ,  $MKL(D_G, D_F)$ , and  $MKL(D_V, D_F)$ , at 64.8%, 64.5%, and 63.8% accuracy, respectively. Combining features with MKL is better than using individual features and combining features with feature concatenation technique. Only the  $MKL(D_V, D_F)$  case was slightly worse than  $concat(D_V, D_F)$ . We further illustrate the weight  $\lambda_i$  proportions of each kernel given by MKL in pie charts in Fig. 17. We investigated the influence of each feature in different combinations. The weights were averaged across ten random splits. The pie charts show that MKL gave more weight on the  $D_F$  in all cases except  $MKL(D_G, D_V)$  which did not consider  $D_F$ . The reason that  $D_F$  was given more weight by MKL because  $D_F$  was previously fine-tuned to fit our training data. Hence, MKL allows  $D_F$  to be a dominant descriptor at 94.0%, 84.0%, and 82.0% in  $MKL(D_G, D_F)$ ,  $MKL(D_V, D_F)$ , and  $MKL(D_G, D_V, D_F)$ , respectively. This means that other features

are extremely limited in playing their roles. In contrast, without taking  $D_F$  into consideration, MKL tended to put more weight on  $D_V$  at 64.0% while  $D_G$  was at 36.0%. Therefore, both features played an important role in the task and complemented each other in improving the overall performance.

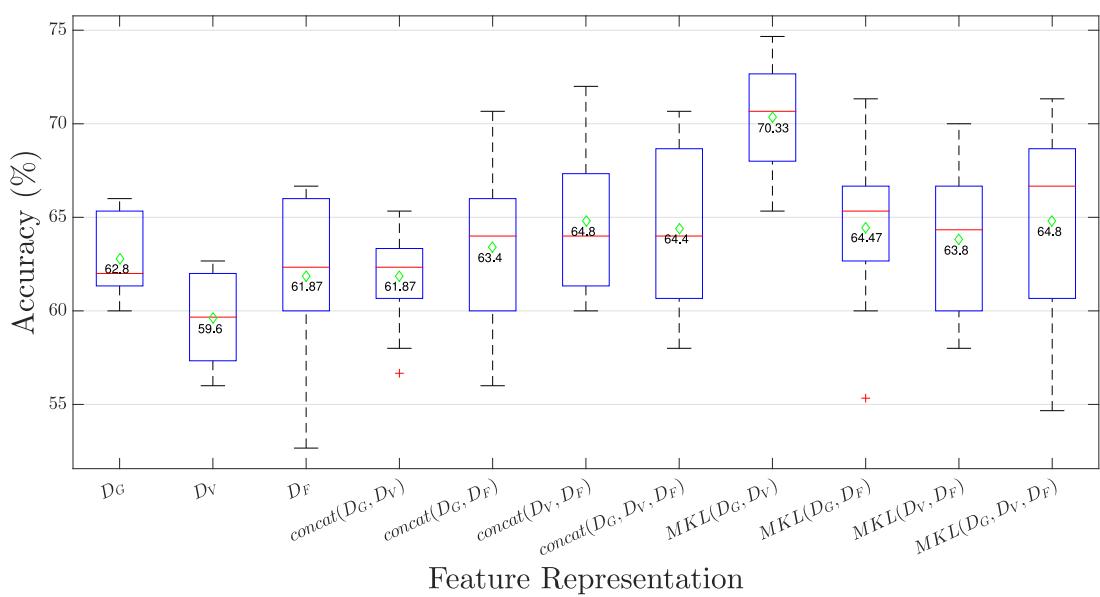
#### 4.4. Comparative analysis

The primary goal of this study is to improve the performance of face shape classification for a hairstyle recommendation system. We compared all of the performances together in a box plot shown in Fig. 18. The figure shows that the best descriptor is  $D_G$  which gave the highest accuracy at 62.8% on average when only a single feature was considered. Moreover, combining  $D_G$  and  $D_V$  with MKL yielded the best accuracy at 70.3% on average.

One-way analysis of variance (ANOVA) was employed to determine whether there are any statistically significant differences between the means of the two or more candidates and the results



**Fig. 17.** Pie charts showing the average weight proportions of each kernel (feature) for different combinations that include (i)  $D_G$ –Green, (ii)  $D_V$ –Blue Sky, and (3)  $D_F$ –Blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 18.** Box plot comparing the accuracies of all types and all combinations of features. Green diamonds and red lines represent the mean and median values of each candidate, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

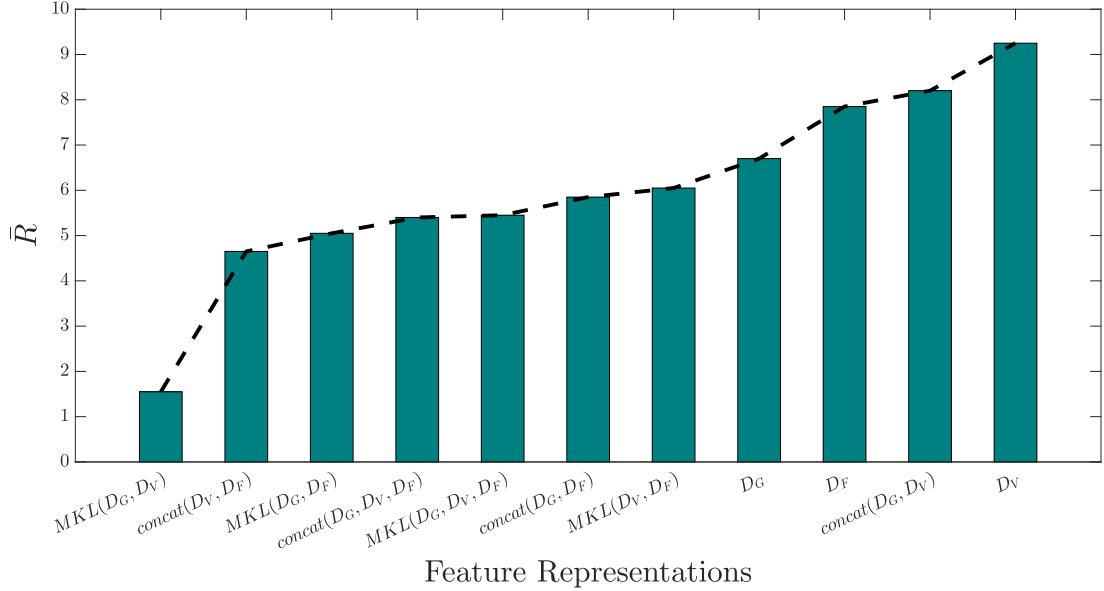


Fig. 19. The graph illustrates  $\bar{R}$  of each candidate-sorted in an ascending order.

Table 1

p-values of multi-comparison based on ANOVA statistical test. Bold face indicates statistically significant differences.

		Individual			concat( · )			MKL( · )			
Individual	<i>D<sub>G</sub></i>	<i>D<sub>G</sub></i>	0.754	1.000	<i>D<sub>G</sub>, D<sub>V</sub></i>	1.000	0.986	<b>0.002</b>	0.997	1.000	0.986
	<i>D<sub>V</sub></i>	0.754		0.966	<i>D<sub>V</sub>, D<sub>F</sub></i>	0.966	0.522	0.112	0.190	<b>0.000</b>	0.175
	<i>D<sub>F</sub></i>	1.000	0.966		<i>D<sub>G</sub>, D<sub>V</sub>, D<sub>F</sub></i>	1.000	0.998	0.839	0.930	<b>0.000</b>	0.918
concat( · )	<i>D<sub>G</sub>, D<sub>V</sub></i>	1.000	0.966	1.000		0.998	0.839	0.930	<b>0.000</b>	0.918	0.989
	<i>D<sub>G</sub>, D<sub>F</sub></i>	1.000	0.522	0.998	<i>D<sub>V</sub>, D<sub>F</sub></i>	0.998	0.999	1.000	<b>0.006</b>	1.000	1.000
	<i>D<sub>V</sub>, D<sub>F</sub></i>	0.986	0.112	0.839	0.839	0.999		1.000	<b>0.069</b>	1.000	1.000
	<i>D<sub>G</sub>, D<sub>V</sub>, D<sub>F</sub></i>	0.998	0.190	0.930	0.930	1.000	1.000		<b>0.036</b>	1.000	1.000
MKL( · )	<i>D<sub>G</sub>, D<sub>V</sub></i>	<b>0.002</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.006</b>	<b>0.069</b>	<b>0.036</b>		<b>0.040</b>	<b>0.013</b>
	<i>D<sub>G</sub>, D<sub>F</sub></i>	0.997	0.175	0.918	0.918	1.000	1.000	1.000	<b>0.040</b>		1.000
	<i>D<sub>V</sub>, D<sub>F</sub></i>	1.000	0.369	0.989	0.989	1.000	1.000	1.000	<b>0.013</b>	1.000	1.000
	<i>D<sub>G</sub>, D<sub>V</sub>, D<sub>F</sub></i>	0.986	0.112	0.839	0.839	0.999	1.000	1.000	<b>0.069</b>	1.000	1.000

Table 2

Ranks assigned to the 11 methods based on their accuracy on ten random splits—agreement is significant at  $p < 0.001$  with Kendall Coefficient of Concordance.

Seed	<i>D<sub>G</sub></i>	<i>D<sub>V</sub></i>	<i>D<sub>F</sub></i>	concat( · )			MKL( · )				
				<i>D<sub>G</sub>, D<sub>V</sub></i>	<i>D<sub>G</sub>, D<sub>F</sub></i>	<i>D<sub>V</sub>, D<sub>F</sub></i>	<i>D<sub>G</sub>, D<sub>V</sub>, D<sub>F</sub></i>	<i>D<sub>G</sub>, D<sub>V</sub></i>	<i>D<sub>G</sub>, D<sub>F</sub></i>	<i>D<sub>V</sub>, D<sub>F</sub></i>	<i>D<sub>G</sub>, D<sub>V</sub>, D<sub>F</sub></i>
1	6	11	9	10	4.5	2	4.5	1	7.5	7.5	3
2	3.5	3.5	8.5	2	8.5	5	6.5	1	10	6.5	11
3	10	11	5.5	8.5	5.5	8.5	3	1	7	2	4
4	4.5	10	11	9	8	6	4.5	2	3	7	1
5	3	11	8.5	8.5	10	4	5.5	1.5	1.5	7	5.5
6	6	3.5	10	6	6	8	10	1	3.5	2	10
7	3.5	11	6.5	10	3.5	2	5	1	6.5	8.5	8.5
8	9.5	11	8	9.5	5	1	5	5	2.5	7	2.5
9	11	9.5	6.5	9.5	5	3	2	1	6.5	8	4
10	10	11	5	9	2.5	7	8	1	2.5	5	5
$\bar{R}$	6.7	9.25	7.85	8.2	5.85	4.65	5.4	1.55	5.05	6.05	5.45

are shown in Table 1. The table shows that only  $MKL(D_G, D_V)$  was better than individual features at  $p < .01$  and better than other combinations with feature concatenation and MKL techniques at  $p < .069$ .

We further ranked each technique based on their accuracy on ten random splits as shown in Table 2.  $\bar{R}$  is the average of the ranks assigned to the 11 objects (methods). The smaller the  $\bar{R}$  is, the better the rank is. Then a non-parametric statistical test called “Kendall Coefficient of Concordance” was performed to evaluate the agreement between the rankings on ten random splits (Siegel

& Castellan, 1988). This led to a significant level of the agreement on ranking at  $p < 0.001$  for  $10^{\circ}$  of freedom which gave the following ranking:

$$MKL(D_G, D_V) < concat(D_V, D_F) < MKL(D_G, D_F) < concat(D_G, D_V, D_F) < MKL(D_G, D_V, D_F) < concat(D_G, D_F) < concat(D_G, D_V) < D_G < D_F < concat(D_G, D_V) < D_V$$

Furthermore,  $\bar{R}$  was sorted and plotted as shown in Fig. 19. This figure shows that  $MKL(D_G, D_V)$  was clearly the best—its  $\bar{R}$  was approximately three times lower than that of the second contender  $concat(D_V, D_F)$ . The worst was using  $D_V$  to represent the data.

Moreover, combining features by using either vector concatenation or MKL could enhance the overall performance.

According to the experimental methodology above, using individual features and concatenated features are simple. The model with hand-crafted features yielded better performance than the one with deep-learned features or fine-tuned deep-learned feature with this data set. Although, the deep-learned model was generated from a very large data set for face recognition task with high accuracy, it still performed worse than the hand-crafted features. This implies that face shape classification is far different from face recognition task. The deep-learned features might possess some characteristics that can effectively distinguish a person's identity but it is not efficient in face shape classification task. The results also imply that prior knowledge is important when there is a small number of samples. Concatenation of features enhanced overall performance, especially when two deep-learned features were concatenated. However, the improvements were not much when some concatenations contained hand-crafted feature. This might be because both deep-learned features had the same structure and number of dimensions while the dimensions of the hand-crafted features were clearly much smaller. On the other hand, combining features with MKL was consistently better than concatenation, especially in the case of combining hand-crafted with deep-learned features. It can yield up to 70.3% accuracy. This shows that the hand-crafted and deep-learned features could work well together. Combining fine-tuned deep-learned feature with the others did not perform well as expected in MKL because MKL tends to put a large weight on the fine-tuned deep-learned feature that is already fit to the data. Thus, a small weight was given to the hand-crafted feature.

## 5. Conclusion

This paper presents an implementation of a hairstyle recommendation system based on face shape classification approach. Once a face shape is identified, the system will recommend a set of hairstyles based on guidelines from an expert. The classification model used in this system is based on SVM-RBF. We propose an approach to improve the performance of the classifier from that of using only individual features. Three different descriptors were employed, namely,  $D_G$ ,  $D_V$ , and  $D_F$ . These descriptors were combined by feature concatenation and MKL techniques. The performances of all candidates were compared and confirmed by statistical analysis techniques.  $D_G$  was the best individual features. Utilising a fine-tuning technique can improve the performance of the pre-trained model. Moreover, performing feature concatenation on VGG face descriptor and its fine tuning model yielded the best accuracy among all combinations by feature concatenation. However, the statistical analysis shows that they are competitive to the usage of individual features. On the other hand, using MKL to combine  $D_G$  and  $D_V$  together was able to achieve the best results statistically confirmed. This shows that hand-designed and deep-learned features can be complementary to each other.

Here are the recommended future research directions to follow this work: (i) In the geometric feature extraction procedure, the current approach utilises the colour-based face segmentation to identify the top point of forehead. This did not work when the user wore fringe as it covered her forehead. It would be better if there is a model that can predict the top point of the forehead with more stability. (ii) In case that more data can be collected, conducting end-to-end feature learning by using hand-crafted feature as a regularisation tool—according to the idea of Wu et al. (2016)—is a reasonable direction to improve the performance. A pre-trained weight of VGG-face descriptor might be used. Then fine-tuned technique with regularised feature learning technique with hand-crafted feature can be applied. (iii) In terms of

feature improvement, investigating several pre-trained models as well as trying different layers in the pre-trained model are keys to explore more appropriate deep-learned features for face shape classification. There is a research that proposes a systematic framework to select an appropriate layer of pre-trained deep CNN-based face recognition for face attribute prediction (Zhong, Sullivan, & Li, 2016). The intermediate layers of the model might keep more physical facial characteristics than the last two layers do. (iv) Although our approach has been proposed and proved from many scientific points of view, it is more interesting to suggest a hairstyle for a face directly without classifying face shape and experts rules. The inference model can be driven from the photos on how models or actresses are wearing their hair. (v) Kang et al. (2017) built a generative model that constructs a new fashion item that can satisfy a customer. This idea can be extended to hairstyle recommendation system too. Creating a new hairstyle that is suitable and satisfy a user would be beneficial to the beauty industrial.

## CRediT authorship contribution statement

**Kitsuchart Pasupa:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Writing - original draft, Writing - review & editing, Visualization, Supervision. **Wisuwat Sunhem:** Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft. **Chu Kiong Loo:** Supervision.

## References

- Ballabio, D., Consonni, V., & Todeschini, R. (2009). The kohonen and CP-ANN toolbox: A collection of MATLAB modules for self organizing maps and counter-propagation artificial neural networks. *Chemometrics and Intelligent Laboratory Systems*, 98(2), 115–122.
- Bansode, N., & Sinha, P. (2016). Face shape classification based on region similarity, correlation and fractal dimensions. *International Journal of Computer Science Issues*, 13(1), 24–31.
- Cootes, T., Taylor, C., Cooper, D., & Graham, J. (1995). Active shape models—their training and application. *Computer Vision and Image Understanding*, 61(1), 38–59.
- Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6), 681–685.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'2009)* (pp. 248–255).
- Derrick, J. (2017). Is your face round, square, long, heart or oval shaped? (Accessed 1 January 2018) <https://www.liveabout.com/is-your-face-round-square-long-heart-or-oval-shaped-345761>.
- Fowler, M. (2015). The best haircut for every face shape. (Accessed 1 January 2018) <https://theladylovescouture.com/the-best-haircut-for-every-face-shape>.
- Fukushima, K., & Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets* (pp. 267–285).
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the ieee conference on computer vision and pattern recognition workshops* (pp. 580–587).
- Gönen, M., & Alpaydin, E. (2011). Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(Jul), 2211–2268.
- Grgic, M., Delac, K., & Grgic, S. (2011). SCFace-surveillance cameras face database. *Multimedia Tools and Applications*, 51(3), 863–879.
- Hong, C. (2013). Powerful hair style (in Thai). Thailand: Nation International Edutainment PCL.
- Huang, G. B., Ramesh, M., Berg, T., & Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *Technical Report*. Amherst: Technical Report 07-49.
- Jain, V., & Learned-Miller, E. (2010). FDDB: A benchmark for face detection in unconstrained settings. *Technical Report*. Amherst: Technical Report UM-CS-2010-009.
- Kang, W.-C., Fang, C., Wang, Z., & McAuley, J. (2017). Visually-aware fashion recommendation and design with generative image models. arXiv:1711.02231..
- Li, L., So, J., Shin, H.-C., & Han, Y. (2013). An AAM-based face shape classification method used for facial expression recognition. *International Journal of Research in Engineering and Technology*, 2, 164–168.
- Li, Y., Wu, F.-X., & Ngom, A. (2016). A review on machine learning principles for multi-view biological data integration. *Briefings in Bioinformatics*, bbw113.
- Liang, Z., & Wang, F. (2016). Research on the personalized recommendation algorithm for hairdressers. *Open Journal of Modelling and Simulation*, 4(3), 102–108.
- Lienhart, R., & Maydt, J. (2002). An extended set of haar-like features for rapid object detection. In *Proceedings of the 2002 international conference on image processing (ICIP'2002)* (pp. 900–903).
- Lodkaew, T., Supsohboon, W., Pasupa, K., & Loo, C. K. (2018). Fashion Finder: A system for locating online stores on Instagram from product images. In *Proceed-*

- ings of the 10th international conference on information technology and electrical engineering (ICITEE'2018) (pp. 288–293).
- Lu, T. T. (2009). *Fundamental limitations of semi-supervised learning*. University of Waterloo Master's thesis.
- Majtner, T., Yildirim-Yayilgan, S., & Hardeberg, J. Y. (2016). Combining deep learning and hand-crafted features for skin lesion classification. In *Proceedings of the 6th international conference on image processing theory tools and applications (IPTA'2016)* (pp. 1–6). IEEE.
- Mustafa, U. (2007). *Face & eye detection*. (Accessed 1 January 2018) <https://www.mathworks.com/matlabcentral/fileexchange/13716-face>.
- Ng, H.-W., Nguyen, V. D., Vonikakis, V., & Winkler, S. (2015). Deep learning for emotion recognition on small datasets using transfer learning. In *Proceedings of the acm international conference on multimodal interaction (ICMI'2015)* (pp. 443–449).
- Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. In *Proceedings of the 2015 british machine vision conference (BMVC'2015)* (pp. 1–12).
- Pasupa, K., Hussain, Z., Shawe-Taylor, J., & Willett, P. (2013). Drug screening with elastic-net multiple kernel learning. In *Proceedings of the 13th international conference on bioinformatics and bioengineering (BIBE'2013)* (pp. 1–5).
- Pasupa, K., & Sunhem, W. (2016). A comparison between shallow and deep architecture classifiers on small dataset. In *Proceedings of the 8th international conference on information technology and electrical engineering (ICITEE'2016)* (pp. 1–6).
- Robnik-Šikonja, M., & Kononenko, I. (1997). An adaptation of relief for attribute estimation in regression. In *Proceedings of the 14th international conference on machine learning (ICML'1997)* (pp. 296–304).
- Saragih, J. M., Lucey, S., & Cohn, J. F. (2009). Face alignment through subspace constrained mean-shifts. In *Proceedings of the 12th international conference on computer vision (ICCV'2009)* (pp. 1034–1041).
- Sarakon, P., Charoenpong, T., & Charoensiriwath, S. (2014). Face shape classification from 3D human data by using SVM. In *Proceedings of the 7th biomedical engineering international conference (bmeicon'2014)* (pp. 1–5).
- Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition workshops* (pp. 806–813).
- Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., et al. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging*, 35(5), 1285–1298.
- Siegel, S., & Castellan, N. J. (1988). *Nonparametric statistics for the behavioral sciences* (2nd). McGraw-Hill.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.
- Sunhem, W., & Pasupa, K. (2016). An approach to face shape classification for hairstyle recommendation. In *Proceedings of the 8th international conference on advanced computational intelligence (ICACI'2016)* (pp. 390–394).
- Tan, B., Zhang, Y., Pan, S. J., & Yang, Q. (2017). Distant domain transfer learning. In *Proceedings of the 31st AAAI conference on artificial intelligence (AAAI'2017)* (pp. 2604–2610).
- Wang, N., Li, S., Gupta, A., & Yeung, D.-Y. (2015). Transferring rich feature hierarchies for robust visual tracking. arXiv:1501.04587.
- Wang, P., Li, Z., Hou, Y., & Li, W. (2016). Combining convnets with hand-crafted features for action recognition based on an HMM-SVM classifier. arXiv:1602.00749.
- Wu, S., Chen, Y.-C., Li, X., Wu, A.-C., You, J.-J., & Zheng, W.-S. (2016). An enhanced deep feature representation for person re-identification. In *Proceeding of the 2016 ieee winter conference on applications of computer vision (WACV'2016)* (pp. 1–8).
- ZergNet (2017). *Find the perfect cut for your face shape*. (Accessed 1 January 2018) <http://www.instyle.com/hair/find-perfect-cut-your-face-shape>.
- Zhang, L., Zhang, D., Sun, M.-M., & Chen, F.-M. (2017). Facial beauty analysis based on geometric feature: toward attractiveness assessment application. *Expert Systems with Applications*, 82, 252–265.
- Zheng-Peng, W., & Zhang, X.-G. (2011). Elastic multiple kernel learning. *Acta Automatica Sinica*, 37(6), 693–699.
- Zhong, Y., Sullivan, J., & Li, H. (2016). Face attribute prediction using off-the-shelf CNN features. In *Proceedings of the international conference on biometrics (ICB'2016)* (pp. 1–7).