International Journal of Electrical and Computer Engineering (IJECE)
Vol. 10, No. 6, December 2020, pp. 5758~5771
ISSN: 2088-8708, DOI: 10.11591/ijece.v10i6.pp5758-5771

Gender classification using custom convolutional neural networks architecture

Fadhlan Hafizhelmi Kamaru Zaman
Faculty of Electrical Engineering, Universiti Teknologi MARA, Malaysia

Article Info

Article history:

Received Sep 9, 2019
Revised May 20, 2020
Accepted Jun 11, 2020

Keywords:

CNN architecture
Convolutional neural network
Cross-dataset inference
Deep learning
Gender classification

ABSTRACT

Gender classification demonstrates high accuracy in many previous works. However, it does not generalize very well in unconstrained settings and environments. Furthermore, many proposed convolutional neural network (CNN) based solutions vary significantly in their characteristics and architectures, which calls for optimal CNN architecture for this specific task. In this work, a hand-crafted, custom CNN architecture is proposed to distinguish between male and female facial images. This custom CNN requires smaller input image resolutions and significantly fewer trainable parameters than some popular state-of-the-arts such as GoogleNet and AlexNet. It also employs batch normalization layers which results in better computation efficiency. Based on experiments using publicly available datasets such as LFW, CelebA and IMDB-WIKI datasets, the proposed custom CNN delivered the fastest inference time in all tests, where it needs only 0.92ms to classify 1200 images on GPU, 1.79ms on CPU, and 2.51ms on VPU. The custom CNN also delivers performance on-par with state-of-the-arts and even surpassed these methods in CelebA gender classification where it delivered the best result at 96% accuracy. Moreover, in a more challenging cross-dataset inference, custom CNN trained using CelebA dataset gives the best gender classification accuracy for tests on IMDB and WIKI datasets at 97% and 96% accuracy respectively.

Corresponding Author:

Fadhlan Hafizhelmi Kamaru Zaman,
Faculty of Electrical Engineering,
Universiti Teknologi MARA,
40450, Shah Alam, Selangor, Malaysia.
Email: fadhlan@uitm.edu.my

1. INTRODUCTION

Human facial analysis has become one of the most significant tasks in computer vision, since it plays a vital role in social interactions. Like other tasks such as the characterization of age, gender, facial attributes, expressions, and personality, automatic gender classification has various important applications such as intelligent user interfaces, user identification, social interaction, visual surveillance, collecting demographic statistics for marketing, behaviour recognition and so on. Therefore, many research efforts have been devoted to design automated system which can classify genders [1-4]. Although this task has been largely addressed in the past, the reported performances are far from optimal especially under unconstrained conditions [5, 6]. Moreover, the complexity of this task largely depends on the context of the application and training protocols. Gender classification model can be trained and tested from the same dataset, also known as in-dataset inference, or from different dataset, also known as cross-dataset inference. Besides, facial images used these datasets can be captured under controlled or uncontrolled/unconstrained environment which will increase the complexity of the task. One of state-of-the-art in gender classification is obtained by Jia and Cristianini where they used 4 million images to train their method called C-Pegasos [7] and tested it using cross-dataset inference strategy on unconstrained LFW dataset. However, more recently, Afifi and

Abdelhamed [8] performed similar cross-dataset tests and based on their results, it can be observed that poorer classification performance may be obtained as a result, in which according to them is due to different conditions of collecting images in different datasets, such as occlusions, illumination changes, backgrounds, etc.

Recently, deep neural networks, more specifically convolutional neural network (CNN) [9] has become the golden standard for object recognition. CNN have improved nearly all areas of computer vision including human action recognition [10], hand-written digit recognition [11], face verification and classification [12-14] and face detection [15]. However, there are two problems associated with CNN in particular, which is (1) the enormous size of data required to train the network such as in [12], and (2) the memory requirement of the network due to computation of massive parameters often limits the application of CNN on embedded platforms such as in mobile phones, as well as on cloud services. For example, two state-of-the-arts CNN architecture called GoogleNet [16] and AlexNet [17] both contains 6,799,700 and 62,378,344 parameters respectively. Another example, a 16-layer CNN described in [18] has a weight file bigger than 500MB and requires about 3.1×10^{10} floating operations per image. Thus, CNN can be regarded as a high-capacity classifier having very large numbers of trainable parameters that requires CNN to learn from larger datasets [16] due to difficult process of tuning and estimating each parameter from small number of samples. To reduce the effect of these limitations, we can optimize the CNN to reduce the complexity of the layers by employing several approaches such as either by reducing the number of convolutional layers, and/or reduce the number of neurons in fully connected layers and/or reduce the resolution of the input images. However, it must be done carefully as to ensure that the resulting architecture can still learn the task at hand, e.g. gender classification, and generalize well on unseen data. The improvement in computation should not compromise the accuracy of the classification.

In this work, the problem of gender classification and high complexity of exiting deep neural networks is addressed, by focusing on reducing the complexity of the CNN and to improve the memory requirement as well as the time required for network inference. In particular, the goal of this paper is to propose a complete design of a low-complexity hand-crafted CNN, where this network will be tested on gender classification task under a very challenging unconstrained conditions as well as undergoing experiment using cross-dataset inference implementations. This relatively simpler and minimized model achieved state-of-the-arts performance and shows a significant boost in inference time when compared against several existing CNN architectures. However, A hand-crafted architecture is a very challenging, time-consuming and require expert knowledge due to a large number of architectural choices [19]. This proposed simplified CNN can learn from relatively smaller dataset and perform classification on a larger dataset.

One of the important work on gender classification called Face Tracer [20] employs combination of Adaboost and Support Vector Machines that select and train on the optimal set of features for each attribute based on the salient structure of faces. Similarly, PANDA-w and PANDA-1 [2] combines deep learning and part-based models by training pose-normalized CNNs to classify various human attributes such as hair style, gender, expression, clothes style, etc from that works well for images under large variability of pose, appearance, viewpoint, occlusion, and articulation. Liu et al., [21] proposed cascades of dual CNNs, called LNet and ANet. These CNNs are pre-trained in different sessions but jointly fine-tuned with attribute tags. ANet is pre-trained for attribute prediction using huge face identities, whereas LNet is pre-trained for face localization using huge general object categories. This approach surpassed the state-of-the-art by a considerable margin and discloses important facts on learning face representation. By combining the intermediate layers of a deep CNN using a separate CNN followed by a multi-task learning algorithm, Hyperface [22] boosts up their individual performances by exploiting the synergy among the tasks. The authors showed that Hyperface is able to extract both holistic and local information in faces and thus outperforms many competitive algorithms for face detection, pose estimation, gender recognition and landmarks localization. Jia and Cristianini [7] presented a simple yet effective classifier of face images called C-Pegasos which is generated by training a linear classification algorithm on a massive dataset which is automatically assembled and labelled. They used four million images and more than 60,000 features to train these classifiers. By employing linear classifiers ensembles, when tested an LFW dataset, C-Pegasos achieved an accuracy of 96.86%.

Recently, Afifi and Abdelhamed [8] proposed an approach based on the behaviour of humans in gender classification. They rely on foggy face which combined isolated facial features and a holistic feature, instead of dealing with the face image as a sole feature. Then, they use foggy face to train a CNN followed by score fusion based on AdaBoost to classify the gender class. Antipov et al., [23] suggested an ensemble model of CNN to enhance the gender classification accuracy from facial images in LFW dataset. Their ensemble model is purposely designed in such a way that minimized the memory requirements and computation time. Likewise, local deep neural network (Local-DNN) [3] is proposed for gender classification where it is relies on two fundamental ideas: deep architectures and local features. overlapping regions in

the visual field is used to train the model by discriminative feed-forward networks built using multiple layers. The authors showed that Local-DNN outperformed other deep-learning-based methods and attains state-of-the-art results in multiple benchmarks.

On the other hand, several works are more focused on designing custom or hand-crafted architectural design of CNN. Most of these works discussed on CNN design issues such as hyperparameters [19, 24], new optimization of objective function [19], improved triplet loss function [25], structure compression of CNN [26] and estimating CNN architectures complexity [27]. These methods share similar objective which is to find the most optimized hand-crafted CNN architecture that produces a high recognition performance while maintaining its complexity to be as low as possible to allow faster computations and less memory requirement.

Additionally, the research community is also addressing a much more realistic general problem for gender classification which is gender classification from facial images in the wild. Many researchers are now focused on experiments involving recent and bigger databases that encompass more variations including identity, ethnicity, age, illuminations, image resolutions and pose variations. This has called for solutions on how to acquire a model that can generalize well on a dataset and gives good inference performance on a completely new unseen dataset. In doing so, is very important to ensure the model did not require constant retraining and can work well in challenging, unconstrained conditions. Thus, a cross-dataset tests have been adopted previously in [1, 7, 8, 23] to measure the performance of gender classifier on new datasets that present this type of challenge. The rest of this paper is arranged as follows. In Section 2 the proposed custom CNN is explained, which can be used to automatically classify genders. Section 3 describes and analyzes the results on the publicly available datasets such as LFW, CelebA and WIKI-IMDB datasets. Finally, Section 4 draws some conclusions and discusses future work.

2. RESEARCH METHOD

The convolutional neural network architecture adopted in this work intends to reduce the complexity by optimizing the convolutional layers, reducing the number of neurons in fully connected layers and reducing the resolution of the input images. In designing this architecture, the network should still be able to deliver state-of-the-arts performance while maintaining a very optimal size. In doing so, AlexNet architecture is used as the starting network template. AlexNet architecture had a very similar architecture as LeNet by LeCun et al., [28] and it contains eight main layers where the first five are convolutional layers. Some of the convolutional layers are followed by Max Pooling layers, and the final three layers are fully connected layers. Firstly, the input is reduced into smaller input image resolution from the original 227×277 to 64×64. Some previous works on face recognition showed that this image resolution is sufficient to achieve good results [29]. By changing the size of image input layer, the size of subsequent convolution layers need to be altered too. For more stable and improved learning, 2 layers of convolution layer have been added to this network. Finally, to reduce the trainable parameters further, the number of neurons are reduced in all fully connected (FC) layers. The number of neurons is later reduced in this custom CNN from 4096, 4096, and 1000 neurons to 100, 50 and 2 neurons at each FC layer, respectively. It is important to note that the final FC layer acts as softmax layer.

To speed up the training, the normalization layers in the network which is based on local response normalization (LRN) are replaced to batch normalization layers [30]. The batch normalization layer makes normalization a part of the model architecture and it performs the normalization for each training mini batch. It allows the network to be trained a higher learning rates that is otherwise difficult and highly unstable with normal LRN layers. Batch normalization has been shown to achieve the same state-of-the-arts performance with 14 times fewer training iterations [30]. The overview of the architecture of the proposed custom CNN is shown in Figure 1.

Subsequently, the layers, size of kernels, number of kernels, and strides for each layer used in custom CNN architecture is given in Table 1. According to Table 1, in total, the custom CNN has 7 convolution layers and 6 batch normalization layers in between of each convolution layers. In between of each convolution layers, Max Pooling layer is added to reduce the spatial dimension of the input volume for next layers. The activation function used in this network is rectified linear unit (ReLu). Several dropout layers are also added with the ReLu to prevent overfitting. To prove that this custom CNN possesses smaller number of parameters as compared to several existing CNN, the parameters in custom CNN design is computed. Here it is shown in detail how to compute the size of output features of each convolution layer and how to compute the number of parameters associated with convolution layers and fully connected layers of a CNN.

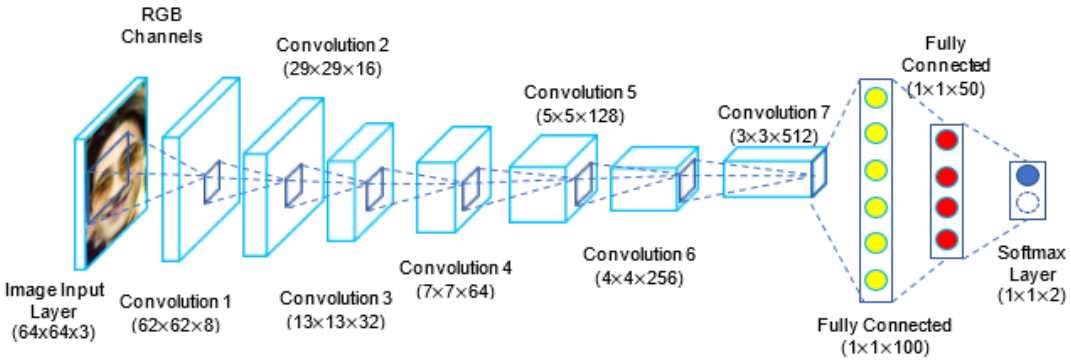


Figure 1. The architecture of the proposed custom CNN, showing 7 convolution layers, 2 fully connected layers and a softmax layer. For clarity, ReLu, Max Pooling, dropout and batch normalization layers are not shown. Size of output features and channels are indicated in the figure using this format: *output width* × *output height* × *output channels*

Table 1. The layers, size of kernels, number of kernels, strides and padding for each layer

Layer	Name	Type	Kernel size, numbers of kernels, strides and padding
1	'imageinput'	Image Input	64x64x1 images with 'zero centre' normalization
2	'conv_1'	Convolution	8 3x3x1 convolutions with stride [1 1] and padding [0 0 0 0]
3	'batchnorm_1'	Batch Norm	Batch normalization with 8 channels
4	'relu_1'	ReLU	ReLU
5	'maxpool_1'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
6	'conv_2'	Convolution	16 3x3x8 convolutions with stride [1 1] and padding [0 0 0 0]
7	'batchnorm_2'	Batch Norm	Batch normalization with 16 channels
8	'relu_2'	ReLU	ReLU
9	'maxpool_2'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
10	'conv_3'	Convolution	32 3x3x16 convolutions with stride [1 1] and padding [0 0 0 0]
11	'batchnorm_3'	Batch Norm	Batch normalization with 32 channels
12	'relu_3'	ReLU	ReLU
13	'maxpool_3'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
14	'conv_4'	Convolution	64 3x3x32 convolutions with stride [1 1] and padding [1 1 1 1]
15	'batchnorm_4'	Batch Norm	Batch normalization with 64 channels
16	'relu_4'	ReLU	ReLU
17	'maxpool_4'	Max Pooling	2x2 max pooling with stride [2 2] and padding [1 1 1 1]
18	'conv_5'	Convolution	128 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]
19	'batchnorm_5'	Batch Norm	Batch normalization with 128 channels
20	'relu_5'	ReLU	ReLU
21	'maxpool_5'	Max Pooling	2x2 max pooling with stride [2 2] and padding [1 1 1 1]
22	'conv_6'	Convolution	256 3x3x128 convolutions with stride [1 1] and padding [1 1 1 1]
23	'batchnorm_6'	Batch Norm	Batch normalization with 256 channels
24	'relu_6'	ReLU	ReLU
25	'maxpool_6'	Max Pooling	2x2 max pooling with stride [2 2] and padding [1 1 1 1]
26	'conv_7'	Convolution	512 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
27	'batchnorm_7'	Batch Norm	Batch normalization with 512 channels
28	'relu_7'	ReLU	ReLU
29	'maxpool_7'	Max Pooling	2x2 max pooling with stride [2 2] and padding [1 1 1 1]
30	'dropout_1'	Dropout	50% dropout
31	'fc_1'	FC	100 fully connected layer
32	'relu_8'	ReLU	ReLU
33	'dropout_2'	Dropout	50% dropout
34	'fc_2'	FC	50 fully connected layer
35	'relu_9'	ReLU	ReLU
36	'dropout_3'	Dropout	50% dropout
37	'fc_3'	FC	2 fully connected layer
38	'softmax'	Softmax	softmax
39	'classoutput'	Classification	crossentropyex with 'male' and 1 other class

2.1. Size of the output features of a convolution layer and max pooling layer

To compute the number of output features of a CNN, let's denote the width of output image as O_c, the width of input image as I, the width of convolution kernels layer as K, the number of kernels used as N, the stride of the convolution as S, the padding as P and the pool size as P_s. The size of the output of a convolution layer O_c and Max Pooling layer O_p are given by:

$$O_c = \frac{I-K+2P}{S} + 1 \quad (1)$$

$$O_p = \frac{I-P_S}{S} + 1 \quad (2)$$

2.2. Parameters of a convolution layer

In a CNN, there are two parameters for each layer, namely the weights and biases. The total number of parameters is the sum of all weights and biases. Let the number of weights and biases of the convolution layer be $W_c = K^2CN$ and $B_c = N$ respectively, where C is the number of channels of the input image. Thus, we can compute the number of parameters of the convolution layer P_c:

$$P_c = W_c + B_c \quad (3)$$

2.3. Parameters of an FC layer connected to a convolution layer

Let the number of weights and bias of a FC Layer which is connected to a convolution layer denoted as $W_{cf} = O^2NF$ and $B_{cf} = F$ respectively, where F is the number of neurons in the FC Layer and O is the size of the output image of the previous convolution layer. Thus, the number of parameters of the convolution layer P_{cf} can be computed from:

$$P_{cf} = W_{cf} + B_{cf} \quad (4)$$

2.4. Parameters of an FC layer connected to another FC layer

Let the number of weights and biases of a FC Layer which is connected to an FC Layer be $W_{ff} = F_{-1}F$ and $B_{ff} = F$ respectively, where F_{-1} is the number of neurons in the previous FC Layer. Thus, we can compute the number of parameters of the convolution layer P_{ff}:

$$P_{ff} = W_{ff} + B_{ff} \quad (5)$$

Finally, total number of parameters $P_{total} = P_c + P_{cf} + P_{ff}$ can be determined. It is important to note that there are no parameters associated with a pooling, dropout, and ReLu layers. The pool size, stride, and padding are all considered as hyperparameters whose value is set before the learning process begins. Although better results can be achieved when hyperparameters are properly adjusted [24], these hyperparameters are considered non-trainable and are external to the model. Throughout this work, the computer system that is used in the experiments runs on Intel i7-6700 CPU @ 3.40 GHz, with 16GB of RAM and uses GTX1080Ti as the main GPU. The hyperparameters used in training the custom CNN are as follows: momentum = 0.9, mini batch size = 500, L2 regularization = 0.001, initial learning rate = 0.1, learn rate drop factor = 0.9, and learn rate drop period = 10.

3. RESULTS AND DISCUSSION

In this section, the results of gender classification experiments on several datasets are presented. Critical discussions on the performance of proposed custom CNN is presented in terms of its accuracy and inference speed on GPU, CPU and an embedded system. The performance in gender recognition is also compared against state-of-the-arts methods such as AlexNet and GoogleNet to highlight the superiority of the proposed method. Besides, the custom CNN is also tested under a cross-dataset inference constraint, where custom CNN trained using CelebA dataset is tested on IMDB and WIKI datasets.

3.1. Datasets description

In the experiments several publicly available datasets namely the labelled faces in the Wild (LFW) dataset, CelebFaces Attributes Dataset (CelebA) dataset and IMDB-WIKI dataset are used. The LFW dataset [31] contains 13,323 photos of 5,749 celebrities taken under unconstrained environments which are then divided into 10-fold cross validation. Each fold contains both male and female images, as suggested by the restricted protocol. Performance is measured using the restricted protocol, in which only gender labels are available in training. LFW gender labels used in this work are determined by Afifi and Abdelhamed [8], where values of attributes suggested by Kumar et al., [32] are used to label the images based on gender. Subsequently, they remove incorrect labels by manually reviewing each category of male and female images three times. In this paper, a variant of LFW dataset called LFWA [33] which contains the same images available in the original LFW dataset is used, however, images in LFWA dataset are aligned using a commercial face alignment software.

CelebA [34] is a large-scale dataset with large facial diversities, huge quantities, and comprehensive annotations that has more than 202,599 images from 10,177 identities, having 40 binary attributes annotations and 5 landmark locations for each image. The images in this dataset also contains background clutter and pose variations. IMDB-WIKI dataset [35, 36] contains 524,230 images which made it one of the largest public face dataset available. These face images are crawled from IMDB and Wikipedia websites. This dataset in total contains 460,723 face images from 20,284 celebrities from IMDb and 62,328 from Wikipedia with gender and age labels are supplied for training. For IMDB-WIKI dataset, only photos that have the second strongest face detection below a certain threshold are chosen, thus the total number of images used in this work from this dataset are 33,181 images and 3,210 images for IMDB and WIKI respectively. In all experiments, all face images are aligned, cropped and resized to 64×64. Some examples of the images from LFW, CelebA, IMDB-WIKI datasets are shown in Figure 2. Subsequently, the number of male and female images for each dataset are summarized in Table 2.

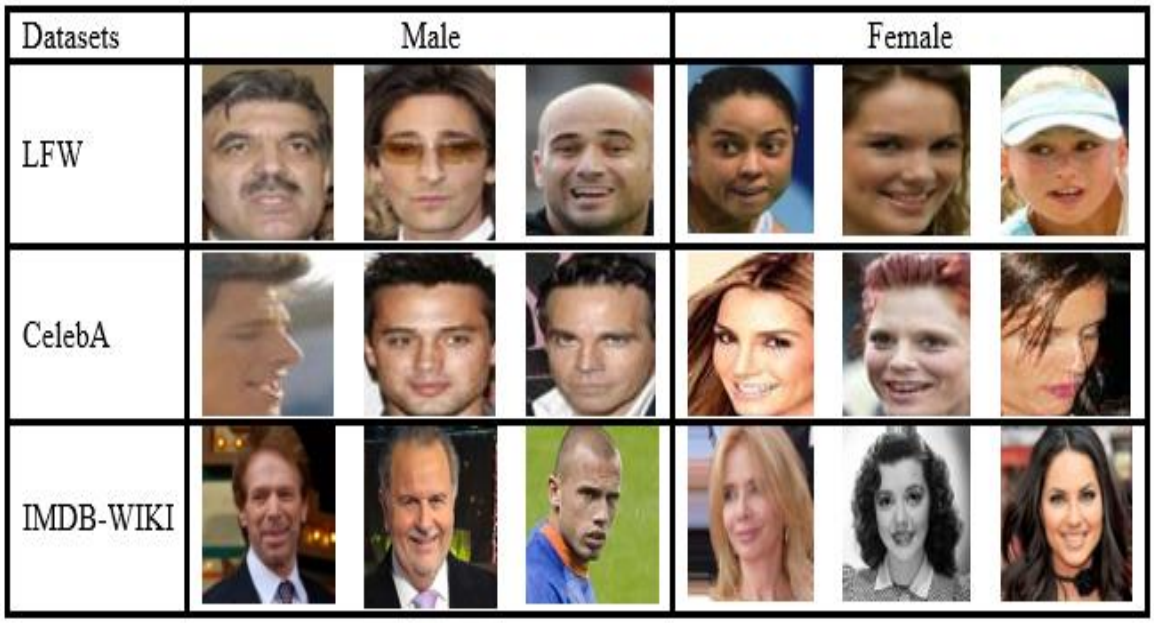


Figure 2. Sample of male (on the left) and female images (on the right) from LFW and CelebA dataset respectively

Table 2. Number of female and male images consist in the datasets used in this work

Datasets	Female images #	Male images #
LFW	2,966	10,268
CelebA	118,165	84,434
IMDB	18,803	14,378
WIKI	1,379	1,831

3.2. Parameters of custom CNN architecture

Firstly, the complexity of the custom CNN architecture are compared against GoogleNet and AlexNet architecture by computing the number of parameters. The feature size (output) of a convolution layer O_c and Max Pooling layer O_p are computed using (1) and (2). Afterward, the number of parameters of a convolution layer P_c, number of parameters of an FC layer connected to a convolution layer P_{cf}, and number of parameters of an FC layer connected to another FC layer P_{ff} are calculated using (3), (4) and (5) respectively. These parameters are given in Table 3.

Based on Table 3, the total number of parameters of custom CNN is 2,041,796 parameters. It is interesting to note that 95% of the parameters comes from the 6th and 7th convolution layer, and the first FC layer. This number of parameters is subsequently compared against the number of parameters of GoogleNet and AlexNet architecture, which is given in Table 4.

Table 3. Number of parameters for each layer in the proposed custom CNN

Layer	Feature Size	Weights	Biases	Parameters
Input Image	64×64×3	0	0	0
Conv-1	62×62×8	216	8	224
BatchNorm	62×62×8	0	16	16
Conv-2	29×29×16	1152	16	1168
BatchNorm	29×29×16	0	32	32
Conv-3	13×13×32	4608	32	4640
BatchNorm	13×13×32	0	64	64
Conv-4	7×7×64	18432	64	18496
BatchNorm	7×7×64	0	128	128
Conv-5	5×5×128	73728	128	73856
BatchNorm	5×5×128	0	256	256
Conv-6	4×4×256	294912	256	295168
BatchNorm	4×4×256	0	512	512
Conv-7	3×3×512	1179648	512	1180160
BatchNorm	3×3×512	0	0	1024
FC-1	100×1	460800	100	460900
FC-2	50×1	5000	50	5050
Softmax	2×1	100	2	102
		Tota1 Parameters #		2041796

Table 4. Comparison on number of parameters of custom CNN against GoogleNet and AlexNet

CNN Models	Image Input Size	Mini Batch Size	Main Layers	Parameters
custom CNN	64 × 64	500	7 conv, 3 FC layers	2,041,796
GoogleNet	224 × 224	30	22 conv with inception layers	6,799,700
AlexNet	227 × 227	30	5 conv, 3 FC layers	62,378,344

According to Table 4, AlexNet has the largest number of parameters with more than 62M parameters while GoogleNet has 6.8M parameters. This renders AlexNet to require larger memory space during training as compared to other methods. The proposed custom CNN possesses lowest number of parameters when compared against the GoogleNet and AlexNet parameters. In fact, custom CNN has 3 times smaller number of parameters than GoogleNet and 30 times smaller than AlexNet. This has a positive impact on the memory requirement for this custom CNN during training, which enable the custom CNN to be trained with mini batch size of 500, compared to GoogleNet and AlexNet which is set to only 30 throughout the experiments. The input image resolution for custom CNN is also significantly smaller at 64×64 pixels, as compared to 224×224 pixels and 227×227 pixels required by GoogleNet and AlexNet respectively. In total, the size of input for custom CNN is 92% smaller than input for GoogleNet and AlexNet. This will allow the custom CNN to be trained much faster and can be trained on relatively cheaper computer system with lesser specifications. Nevertheless, based on results presented in subsequence, the capability of custom CNN to extract and learn highly complex parameters from significantly smaller images in much faster time without sacrificing the accuracy is demonstrated. Since training large datasets on CPU or Visual Processing Unit (VPU) is painfully slow, it is more appropriate to show the computational advantage of custom CNN by measuring the average inference time for 1200 images from 10 folds of LFW dataset using custom CNN, GoogleNet and AlexNet. The inferences are run on GTX1080Ti (GPU), Intel i7-6700 @ 3.40 GHz (CPU) and Movidius Neural Compute Stick (Movidius NCS) (VPU). The average inference time measured in this experiment is given in Figure 3.

According to Figure 3, custom CNN requires only 0.92ms to classify 1200 images on GPU, while GoogleNet and AlexNet requires significantly longer inference time at 4.42ms and 1.95ms respectively. On CPU, the average inference time for custom CNN is just 1.79ms, while GoogleNet and AlexNet requires 62.37ms and 20.41ms respectively. For VPU, custom CNN again requires least amount of inference time of just 2.51ms, while GoogleNet and AlexNet requires 95.78ms and 55.60ms respectively. This put into perspective that custom CNN capitalize on its lesser requirements of image size and parameters which enables custom CNN to be inferred at significantly higher speed on GPU, CPU and VPU environment. The most important takeaway is this will allow custom CNN to be deployed in real-time tasks whether it is run on GPU/CPU-based system or on completely embedded system such as those relying on more energy-efficient VPU such as Movidius NCS. In the following section, it is shown that even though custom CNN works on smaller number of parameters, its performance in gender classification is in fact on-par and occasionally better than state-of-the-arts.

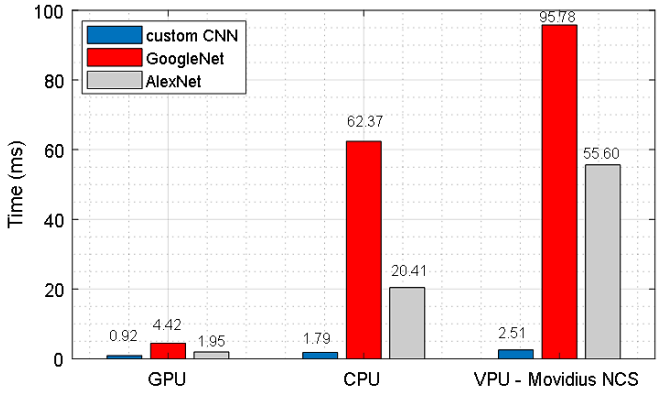


Figure 3. Comparison between the average inference time of 1200 images from LFW dataset for custom CNN, GoogleNet and AlexNet on different inference environments – GPU (GTX 1080Ti), CPU (i7-6700 CPU @ 3.40 GHz, and VPU (Movidius Neural Compute Stick)

3.3. Performance in gender classification

The custom CNN is trained to classify gender from face images in LFWA dataset and its performance is evaluated in terms of accuracy, true positive rate (TPR), false positive rate (FPR) and Precision. These performances are averaged over 10 runs from 10 folds as cross validation as mentioned earlier. Two variants of images are used, namely the grayscale and RGB images. The performance for gender classification using custom CNN is compared against the performance of GoogleNet and AlexNet classifiers doing the same task and is shown in Figures 4 and 5.

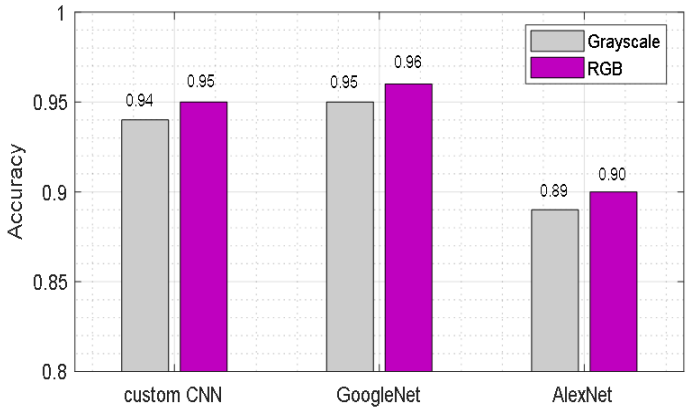
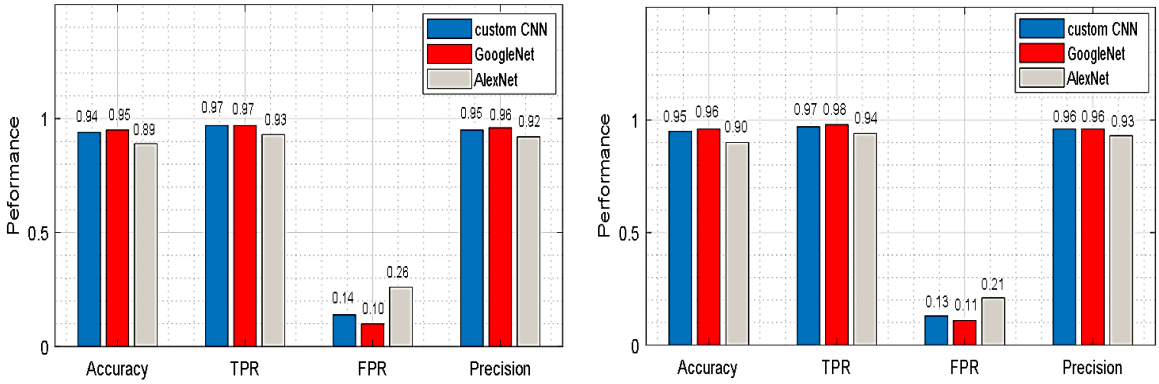


Figure 4. Comparison between the accuracy of custom CNN, GoogleNet and AlexNet with respect to classification using grayscale and RGB images

According to Figure 4, at least slightly better performance is achieved using RGB images compared with grayscale images for all tested classifiers. 0.01 accuracy improvement is obtained by custom CNN, GoogleNet and AlexNet when using RGB images as opposed to using grayscale images. According to Figures 4 and 5, GoogleNet gives the best accuracy for RGB images with an accuracy of 0.96, while custom CNN delivers 0.95 accuracy, better than AlexNet which is at 0.90 accuracy. In terms of TPR, GoogleNet again delivers best performance at 0.98 TPR, while custom CNN and AlexNet delivers 0.97 and 0.93 TPR respectively. Similarly, GoogleNet delivers best performance for FPR and Precision, where custom CNN delivers second-best performance, followed by AlexNet with the worst performance of all three classifiers. A closer look on the accuracy of tested classifiers for each test fold in Figure 6 shows that custom CNN and GoogleNet has comparable performance and more stable with just slight fluctuations in accuracy when compared to AlexNet. These excellent performances of custom CNN are quite impressive considering its significantly less complex architecture which contains fewer parameters and inferred at significantly higher speed than state-of-the-arts such as GoogleNet and AlexNet classifiers.



(a) (b)

Figure 5. Performance measured in terms of average test accuracy, true positive rate, false positive rate, and precision for custom CNN, GoogleNet and AlexNet for (a) grayscale images, and (b) RGB images

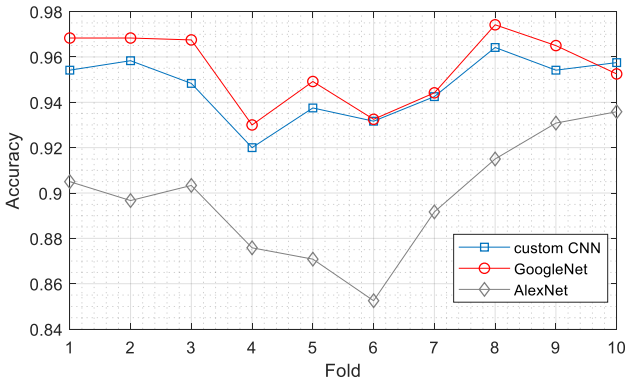
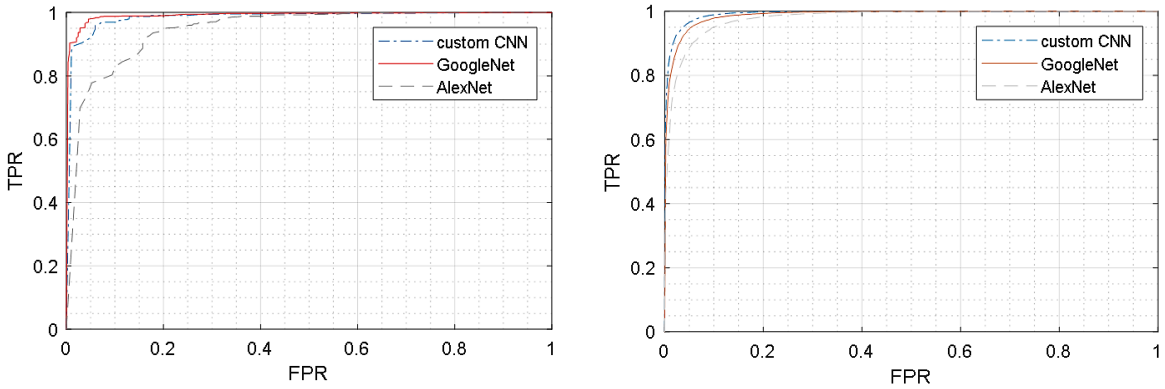


Figure 6. The performance measured as accuracy for custom CNN, GoogleNet and AlexNet for each test fold in LFW dataset

Afterwards, the performance of gender classification for LFWA and CelebA dataset is evaluated in the form of Receiver Operating Characteristics (ROC) curve. It is one of the most important evaluation metrics for checking any classification model's performance. Area Under the Curve (AUC) is used to measure the performance at various thresholds settings used by the final softmax layer. The ROC curve from this experiment is shown Figure 7. Based on Figure 7, for LFWA datasets, GoogleNet has the best AUC, followed custom CNN and AlexNet. Interestingly, the AUC for custom CNN and GoogleNet is not too much different from one another thus proving that custom CNN again can deliver performance on par with GoogleNet classifier despite its much simpler architecture. Moreover, the AUC of custom CNN on CelebA dataset exceed the AUC of GoogleNet and AlexNet, making custom CNN as the best performing classifier for CelebA dataset. This exceptional performance on CelebA dataset also proves that even though custom CNN contains fewer parameters that the other tested classifiers, it can learn to classify the gender of more than 200K images correctly most of the time.

Then, the performance of the custom CNN for gender classification is compared against several other state-of-the-arts. Similar protocols as used by the original publications of these methods are used, and the results are given in Table 5. For LFWA dataset, LNets+ANet [21] delivers the best result with 98% accuracy while custom CNN delivers 95% accuracy. However, customCNN delivers better accuracy compared to several methods such as AlexNet, Face Tracer [20], PANDA-w [2], and RCNN Gender [22]. In fact, custom CNN delivers the same accuracy as [37] +ANet. For CelebA dataset, custom CNN delivers the best performance, surpassing all state-of-the-arts with 96% gender classification accuracy. Custom CNN is even 3% better than GoogleNet and 2% better than the recent HyperFace [22]. The following Figure 8 show the example of female and male samples with incorrect classification for LFW and CelebA dataset.

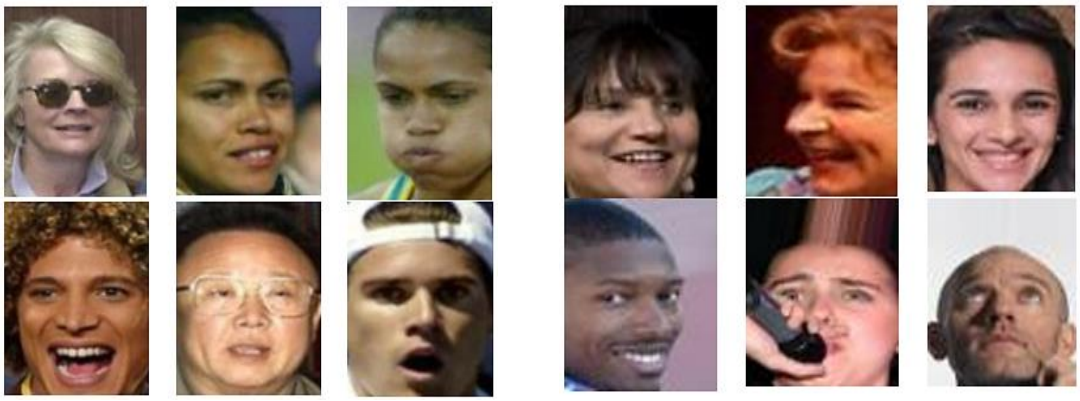


(a) (b)

Figure 7. ROC for (a) LFW and (b) CelebA datasets

Table 5. Accuracy of gender classification for compared against several state-of-the-arts

Method	Classification Accuracy (%)	
	LFWA	CelebA
Face Tracer [20]	91	84
PANDA-w [2]	93	86
PANDA-1 [2]	97	92
[37]+ANet	95	91
LNets+ANet [21]	98	94
Castrillon-Santana et al [1]	98	NA
RCNN Gender [22]	95	91
Multitask Face [22]	97	93
HyperFace [22]	97	94
C-Pegasos [7]	97	NA
AFIF [8]	96	NA
AlexNet	90	91
GoogleNet	96	93
custom CNN	**95**	**96**



(a) (b)

Figure 8. Examples of female (top) and male (bottom) samples with incorrect classification for, (a) LFW dataset and (b) CelebA dataset

According to Figure 8, some of supposedly female subject may have masculine features such as more angular face shape with strong jawline. Likewise, the supposedly male subjects may have feminine features such as rounder face and cheeks. Other factors that may contribute to misclassification is pose variations and objects or occlusions present in facial images, e.g.: sunglasses and microphones. However, these misclassifications can be reduced by having more samples of male and female possessing the pose variations and objects or occlusions in training dataset.

3.4. Cross-dataset inference

To evaluate the robustness and generalization performance of the custom CNN, the accuracy of gender classification in cross-dataset inference fashion is measured. Cross-dataset inference implies that a model trained using a dataset and tested on another completely different dataset. This implementation is similarly used in [1] and it is not transfer learning, since the same classification layer is kept and tested on the new dataset. This is a very challenging test, since the variations and inherent attributes contained in a dataset may not exist in another dataset thus introducing large variability in test dataset. However, it can be used to validate whether the model can generalize well and not overfit to training data. In this experiment, custom CNN, GoogleNet and AlexNet on LFWA and CelebA dataset are trained and subsequently tested on LFWA, CelebA, IMDB and WIKI dataset. IMDB and WIKI datasets are not used at all during training-only LFWA and CelebA dataset is used to train the model. The results of this experiment are given in Table 6, where shaded areas indicate that the results that are obtained from test on same dataset (in-dataset inference).

Table 6. Gender classification for cross-dataset inference using LFWA and CelebA datasets as training data

Training Dataset	CNN Model	Gender Classification Accuracy on Inference Dataset (%)			
		LFWA	CelebA	IMDB	WIKI
LFWA	custom CNN	95	93	96	94
	GoogleNet	**96**	94	96	94
	AlexNet	90	93	85	81
CelebA	custom CNN	94	**96**	**97**	**96**
	GoogleNet	93	93	96	93
	AlexNet	77	91	70	60

To simplify the results, train dataset – test dataset notation is used. For example, LFWA–LFWA indicates LFWA is used in training and LFWA is used in testing. In this case, different portion of data from the same dataset is used for training and testing. According to Table 6, LFWA–LFWA inference yields best result using GoogleNet at 96% accuracy while LFWA–CelebA inference yields best result using custom CNN at 94% accuracy. CelebA–LFWA inference yields best result using GoogleNet at 94% accuracy, while CelebA–CelebA inference yields best result at 96% accuracy using custom CNN. GoogleNet and custom CNN both deliver best result in IMDB–LFWA inference at 96% accuracy. On the other hand, custom CNN delivers best result in IMDB–CelebA inference at 97% accuracy. Again, for WIKI–LFWA inference, custom CNN and GoogleNet yields best result at 94% accuracy for both, while custom CNN yet again delivers the best result at 96% accuracy for WIKI–CelebA inference. Overall, custom CNN trained using CelebA dataset gives the best gender classification accuracy for CelebA, IMDB and WIKI datasets at 96%, 97% and 96% accuracy respectively which highlight its robustness and ability to generalize trained data on completely different datasets. Another important factor is CelebA is a very large dataset, thus most of the variations that exist in LFWA, IMDB, and WIKI may have been captured by the custom CNN from CelebA images. Several variabilities in terms of 1) identity, age and ethnicity, 2) pose and illumination conditions, and 3) image resolution may be shared across these datasets. On the other hand, AlexNet fails to generalize well in cross-dataset inference experiments where it delivers bad performance in all cross-dataset tests. AlexNet worst performance delivers 77%, 70% and 60% accuracy for LFWA–CelebA, IMDB–CelebA and WIKI–CelebA inferences respectively. From this result, custom CNN also shows that it can generalize well from smaller dataset and perform classification on larger dataset, where custom CNN trained using LFWA dataset can correctly classify gender in larger CelebA dataset with good accuracy of 93%.

One of the challenges of CNN is to comprehend what exactly happen at each layer during training. It is well known that each layer extracts high-level features of the image at earlier layers, while the final layer basically decides on the class of the images. The first layer normally finds edges or corners whereas intermediate layers interpret the basic features to look for overall shapes or components, like a cat or a ball. The final few layers accumulate those features into complete interpretations of the trained class. To learn more about the features learned by the custom CNN in classifying gender, the features learned by the custom CNN at convolution 4 layer and softmax layers are visualized for LFWA and CelebA datasets respectively. These features can be visualized using DeepDream, a dream-like hallucinogenic appearance in the deliberately over-processed images [38, 39]. The visualization is generated images that strongly activate a particular channel of the network layers. The DeepDream visualization from custom CNN which is trained to classify gender is illustrated in Figure 9. At 4th convolution layer, features are more mixed and has different impressions, even though the training samples only contains two classes. This is due to many learned features contained in the images, which belong to different gender classes. At softmax layer, only 2 distinct learned features appear which is male on the left, and female on the right. Comparing softmax features between

LFWA and CelebA, interestingly male learned feature of DeepDream is quite consistent and similar in both datasets, however the female learned feature is somewhat different in both datasets.

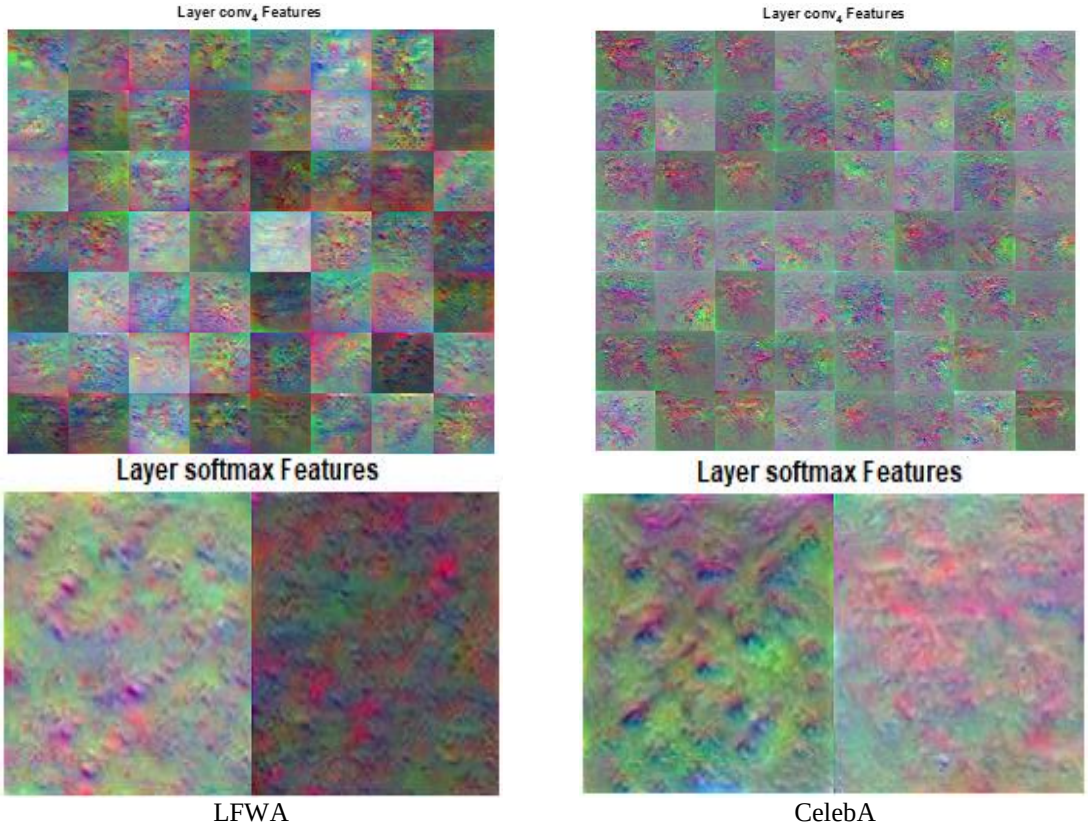


Figure 9. Visualization of features learned by the custom CNN at convolution 4 layer and softmax layers respectively for LFWA and CelebA datasets. Male learned features are shown on the left, while female learned features are shown on the right

4. CONCLUSION

In this work, a hand-crafted, custom CNN architecture is presented which is designed to distinguish between male and female facial images. This custom CNN contains only 7 convolutional layers and 2 fully connected layers, with batch normalization layers used in between the convolutional layers. It requires relatively smaller input image and as a result, it has significantly less parameters than other architecture such as GoogleNet and AlexNet. In fact, custom CNN has 3 times smaller number of parameters than GoogleNet and 30 times smaller than AlexNet. Extensive experiment using various publicly available unconstrained datasets demonstrated the advantages of custom CNN. It delivered the fastest inference speed in all tests where it requires only 0.92ms to classify 1200 images on GPU, 1.79ms required on CPU, and 2.51ms on VPU. The proposed custom CNN also yielded performance on-par with state-of-the-arts and even surpassed these methods in CelebA gender classification where it delivered the best result at 96% accuracy. Moreover, in cross-dataset inference experiment, custom CNN trained using CelebA dataset gives the best gender classification accuracy for IMDB and WIKI datasets at 97% and 96% accuracy respectively which highlight its robustness and ability to generalize trained data on completely different datasets. In future, the performance of the custom CNN will be evaluated on other classification tasks such as classifying people and objects.

ACKNOWLEDGEMENTS

The author would like to thank Ministry of Education for the FRGS grant (600-IRMI/FRGS 5/3/(081/2019)) and Faculty of Electrical Engineering, Universiti Teknologi MARA for the support given in this work.

REFERENCES

[1] M. Castrillón-Santana, J. Lorenzo-Navarro, and E. Ramón-Balmaseda, "Descriptors and regions of interest fusion for in- and cross-database gender classification in the wild," *Image and Vision Computing*, vol. 57, pp. 15-24, 2017.

[2] N. Zhang, M. Paluri, M. A. Ranzato, T. Darrell, and L. Bourdev, "PANDA: Pose Aligned Networks for Deep Attribute Modeling," [Online]. Available: *arXiv:1311.5591*, 2013

[3] J. Mansanet, A. Albiol, and R. Paredes, "Local Deep Neural Networks for gender recognition," *Pattern Recognition Letters*, vol. 70, pp. 80-86, 2016.

[4] Z. Xie, Z. Guo, and C. Qian, "Palmprint gender classification by convolutional neural network," *IET Computer Vision*, vol. 12, no. 4, pp. 476-483

[5] J. E. Tapia and C. A. Perez, "Gender Classification Based on Fusion of Different Spatial Scale Features Selected by Mutual Information From Histogram of LBP, Intensity, and Shape," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 488-499, 2013.

[6] I. R. P. Selvam and M. Karuppiah, "Gender recognition based on face image using reinforced local binary patterns," *IET Computer Vision*, vol. 11, no. 6, pp. 415-425

[7] S. Jia and N. Cristianini, "Learning to classify gender from four million images," *Pattern Recognition Letters*, vol. 58, pp. 35-41, 2015.

[8] M. Afifi and A. Abdelhamed, "AFIF4: Deep Gender Classification based on AdaBoost-based Fusion of Isolated Facial Features and Foggy Faces," *arXiv:1706.04277*, 2017

[9] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks, A. A. Michael, Ed.: MIT Press*, pp. 255-258, 1998.

[10] S. Ji, W. Xu, M. Yang, and K. Yu, "3D Convolutional Neural Networks for Human Action Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221-231, 2013.

[11] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3642-3649, 2012.

[12] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701-1708, 2014.

[13] M. P. Beham and S. M. M. Roomi, "A review of face recognition methods," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 27, no. 4, 2013.

[14] B. Nassih, A. Amine, M. Ngadi, and N. Hmina, "Combination of feature sets based on binary pattern and oriented gradient for efficient face classification," *International Journal of Artificial Intelligence*, vol. 16, no. 2, pp. 172-193, 2018.

[15] S. Sudhakar Farfade, M. Saberian, and L.-J. Li, "Multi-view Face Detection Using Deep Convolutional Neural Networks," [Online]. Available: *arXiv:1502.02766*, 2015

[16] C. Szegedy et al., "Going deeper with convolutions," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9, 2015.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, 2012.

[18] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," [Online]. Available: *arXiv:1409.1556*, 2014

[19] S. Albelwi and A. Mahmood, "A Framework for Designing the Architectures of Deep Convolutional Neural Networks," *Entropy*, vol. 19, no. 6, 2017.

[20] N. Kumar, P. Belhumeur, and S. Nayar, "FaceTracer: A Search Engine for Large Collections of Images with Faces," *Spriner, Berlin, Heidelberg*, pp. 340-353, 2008.

[21] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild," [Online]. Available: *arXiv:1411.7766*, 2014.

[22] R. Ranjan, V. M. Patel, and R. Chellappa, "HyperFace: A Deep Multi-task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition," [Online]. Available: *arXiv preprint arxiv:1603.01249*, 2016.

[23] G. Antipov, S.-A. Berrani, and J.-L. Dugelay, "Minimalistic CNN-based ensemble model for gender prediction from face images," *Pattern Recognition Letters*, vol. 70, pp. 59-65, 2016.

[24] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *Neural networks: Tricks of the trade. Springer, Berlin, Heidelberg*, pp. 437-478, 2012.

[25] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based CNN with improved triplet loss function," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016, pp. 1335-1344, 2016.

[26] G. Zhong, H. Yao, and H. Zhou, "Merging Neurons for Structure Compression of Deep Networks," *Proceedings International Conference on Pattern Recognition*, vol. 2018, pp. 1462-1467, 2018.

[27] M. D. Ferreira, D. C. Corrêa, L. G. Nonato, and R. F. de Mello, "Designing architectures of convolutional neural networks to solve practical problems," *Expert Systems with Applications*, vol. 94, pp. 205-217, 2018.

[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[29] F. Kamaruzaman and A. A. Shafie, "Recognizing faces with normalized local Gabor features and Spiking Neuron Patterns," *Pattern Recognition*, vol. 53, pp. 102-115, 2016.

[30] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," [Online]. Available: *arXiv preprint arXiv:1502.03167*, 2015

[31] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller., "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," *Technical Report 07-49, University of Massachusetts, Amherst*, 2007.
[32] N. Kumar, A. Berg, P. N. Belhumeur, and S. Nayar, "Describable Visual Attributes for Face Verification and Image Search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 1962-1977, 2011.
[33] L. Wolf, T. Hassner, and Y. Taigman, "Effective Unconstrained Face Recognition by Combining Multiple Descriptors and Learned Background Statistics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 1978-1990, 2011.
[34] S. Yang, P. Luo, C. Change Loy, and X. Tang, "From Facial Parts Responses to Face Detection: A Deep Learning Approach," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3676-3684, 2015.
[35] R. Rothe, R. Timofte, and L. Van Gool, "Deep Expectation of Real and Apparent Age from a Single Image Without Facial Landmarks," *International Journal of Computer Vision*, vol. 126, no. 2, pp. 144-157, 2018.
[36] R. Rothe, R. Timofte, and L. V. Gool, "DEX: Deep EXpectation of Apparent Age from a Single Image," *IEEE International Conference on Computer Vision Workshop*, pp. 252-257, 2015.
[37] J. Li and Y. Zhang, "Learning SURF Cascade for Fast and Accurate Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3468-3475, 2013.
[38] A. Mordvintsev, C. Olah, and M. Tyka., "DeepDream - a code example for visualizing Neural Networks," 2015.
[39] A. Mordvintsev, C. Olah, and M. Tyka., "Inceptionism: Going Deeper into Neural Networks," 2015.

BIOGRAPHY OF AUTHOR

Fadhlan Hafizhelmi Kamaru Zaman received the B.Sc (Hons.) and P.hD. degrees from International Islamic University Malaysia in 2008 and 2015, respectively. He is currently a Senior Lecturer at Department of Computer Engineering, University of Technology MARA, Malaysia. His research interests are in pattern recognition, signal and image processing, artificial intelligence and computer vision. Fadhlan is a Professional Engineer with Board of Engineer Malaysia (BEM) member of IEEE, Professional Technologist with Malaysian Board of Technologist (MBOT), and a Chartered Engineer from the Institution of Engineering and Technology, UK.