Facial Attributes Classification: a Light Weight Deep Neural Network

1st Ziqing Wang
Research School of Engineering
u6762874@anu.edu.au

1st Suikei Wong
Research School of Engineering
u6624985@anu.edu.au

***Abstract*—Facial attributes is a mid-level feature in computer vision task, which has achieved outstanding performance in classification problems by various deep neural networks (DNN). However, the majority of DNN has complex structure and huge amount of parameters, which is trained and computed costly. In this paper, we implement a small and efficient model called MobileNets, which is based on depthwise separable convolutions. Two simple hyper-parameters are introduced to choose the suitable size of the model based on the actual problem. We also develop a standard DNN model by using transfer learning and present experiments on the accuracy and computational cost of both models. The results show that compared to standard DNN and other popular models on ImageNet classification, MobileNets can achieve similar accuracy with less computation.**

***Index Terms*—Attributes classification, Light weight neural network, Deep neural network**

I. Introduction

As a fundamental research area, face attributes classification has received popularity recently. There is a wide range of applications based on it, such as verification [1] and recognition [2]. In facial attributes classification tasks, the input is a facial image which the model never seen before and it requires the model to classify and predict the attributes in it [3], such as gender, bald, smile, etc.

Deep Neural Network (DNN) has been widely used in recent research. Liu et al. [4] combine three DNN together for face detection, attributes extraction, deep feature learning, and classify the different attributes with Support Vector Machine (SVM). Zhuang et al. [3] use transfer learning adding fully connected layer after pre-trained VGG16 model. In [5], facial landmark points are also introduced to learn the common feature representation of different attributes. These techniques perform well by using DNN.

There are two major methods for facial attributes classification task. The first method is to predict each attribute separately, which is so called single-label learning [1] [6]. In this method, these facial attributes are independent and each attribute is classified by binary classifier [1]. However, some unlabelled attributes such as age and attraction may not be able to classified individually. Therefore, in multi-label learning [5], these attributes can be predicted simultaneously by using labelled data [3]. For most dataset, only common attributes are labelled well which causes failure in multi-label learning.

Besides, the general trend is to make the networks deeper [7] [8], since some deep features are more likely to be extracted and classified through these more complicated networks. However, simply adding the layers of DNN and increasing the numbers of filters in convolutional layers can significantly improve the accuracy of the model, the computational costs also show exponential growth. There are also some approaches proposed for faster training [9] and some small networks for reducing size as well as training time [10] [11].

Inspired by the above observations, we implement a small but efficient network MobileNets [5], which is based on separating the traditional standard convolutional filters. More specifically, this model has the same feature extracting convolutional layers as the standard DNN, but it separates these standard convolutional filters into a light structure with deepwise convolutional filters and 1×1 pointwise convolutional layers. In order to verify the real performance of this light weight network with standard DNN, we also develop a model based on transfer learning by implementing a linear classifier after the convolutional layers of pre-trained ResNet18. Moreover, we consider this task as a single-label learning since all data for training are labelled. MobileNets is able to achieve state-of-the-art performance for large-scale dataset CelebA [4].

The main contributions of this paper are as follows:

- We develop a transfer learning model based on ResNet18 with a linear classifier for facial attributes classification.
- We analyse the depthwise separable convolution and implement MobileNets for facial attributes classification.
- We compare the computational cost between standard convolutional network and MobileNets. The performance of both model and some other popular models on a challenging and large-scale dataset CelebA.
- We achieve state-of-the-art performance on MobileNets for size, speed and accuracy.
- We decrease the computational cost significantly over 8 times comparing to standard deep convolutional neural network.

The remainder of the paper is organized as follows: In Section 2, we do literature review and some related works are briefly discussed. In Section 3, the details of the development of transfer learning model(ResNet18 with linear classifier) and MobileNets for facial attributes classification task are described. In Section 4, the experimental results are reported and compared. In Section 5, conclusions are presented.

II. Related work

There are numbers of proposed work [3], [4], [12]–[14] on deep learning with CNNs, multi-task learning as well as transfer learning. In this section, the relevant literature of these areas is reviewed and briefly discussed.

A. Deep Learning with CNN

A growing number of works are based on deep convolutional neural networks for feature extraction in facial attributes classification task. Comparing with traditional methods, such as Histogram of Oriented Gradients (HOG) [15] and Eigenfaces [16], these methods are based on the analysis and process of images. Recently, more and more methods based on deep learning [17] [18] have shown great improvements. Besides, the introduction of open-source framework and packages such as Torch [19], Tensorflow [20] and Caffee [21] also make it popular. Rudd et al. [13] introduce a novel mixed objective optimization network MOON to solve the imbalance training data problem in tranditional deep CNNs. Zhong et al. [17] consider using mid-level CNN features as representations instead of high-level abstractions, which solve the face recognition task as well as attribute prediction by using a single deep network. Zhang et al. [12] use the part-based models which perform well on various subtle signal in attribute classification, and combine it with pose-normalized CNNs based on deep learning. These deep learning based methods rely on CNN, which is used to extract and classify features. Our works still focus on using CNNs to solve this facial attribute classification problem, which is robust and improves the performance.

B. Multi-task Learning

As a method to earn several different but related labels simultaneously, Multi-task learning (MTL) [5] has gained popularity. For example, Huang et al. [22] propose a multi-task deep neural network (MT-DNN), which generalizes one classification task into multiple binary classification tasks. Hand et al. [23] also build a state-of-the-art model based on multi-task deep convolutional neural network (MCNN) with an auxiliary network at the top (AUX). Pillai et al. [24] develop a classification algorithm which maximizes specific accuracy measures in multi-label problem. In [25], Argyriou et al. present a method based on 1-norm regularization problem sharing across multiple related tasks. Similarly, Hwang et al. [26] adopt convex multi-task feature learning for visual attributes recognition models, which learn a shared lower-dimensional representation. In DNN, features are learned by hierarchical structure of images such as pixels and edges. These low-level features usually have the similar representations [3] and the features of multiple face attributes are learned in high-level part. In our work, we consider all attributes into different binary classification problems, which means we assume that all attributes are independent. Therefore, the feature of each attribute can be extracted separately.

C. Transfer Learning

Since the neural networks often require a large amount of training data to achieve proper learning. The model with limited number of labelled data may not be able to extract all features well. In order to improve the performance of the model, using transfer learning is convenient and effective with limited annotated data. [27] Most computer vision models in transfer learning are based on the model trained on ImageNet [28], which is a large-scale image database. Therefore, for most computer vision tasks, these models can complete the classification or recognition task. Zhuang et al. [3] propose a deep transfer neural network model, which uses Support Vector Machine (SVM) classifier for each class after the face detection network Fast R-CNN [29] and extracts features by using VGG16 [7]. In [30], Long et al. put forward a novel transfer learning approach Joint Distribution Adaptation (JDA) which considers both marginal and conditional distributions in dimensionality reduction procedure. Another work from authors [31] propose a Transfer Joint Matching (TJM) approach since matching the features jointly produces domain difference, so it should be reduced. However, these models for transfer learning are usually very large and deep with millions of parameters [32], which are redundant. There are also some small and efficient neural networks [10], [33], [34], but most of them only focus on size and ignore speed [11]. As far as we know, our work on such a light weight DNN is fast, precise and effective.

III. Approach

A. Transfer Learning Model

ResNet18 is a classic and popular deep convolutional neural network for its outstanding performance by using residual block structure. [35] Its deep and complex convolutional layers are pre-trained on ImageNet, a large-scale hierarchical image database [28], so ResNet18 has strong ability to extract deep features of images well and precisely.

The original ResNet18 only has one fully connected layer for feature classification after the average pooling layer. Considering the problem that this task is a multi-attributes classification (40 attributes in total), we replace the last fully connected layer into a three-layer linear classifier. Fig.1 shows the overall structure of this transfer learning model. The input and output sizes of the three linear classifier layers are (512, 512), (512, 128) and (128, 40) separately.

The input of the first layer of the linear classifier are the features extracted after 17 convolutional layers of ResNet18. The first linear layer takes the size of 512 input features into 512 classes, then the second linear layer classifies them into 128 classes. Finally the last linear layer separates them into 40 classes, which is our ideal output size. As this task is considered as a single-label learning, all the attributes are independent. Therefore, the activation function for the last layer is `Sigmoid`, and for each attribute there are only two prediction results: 0 and 1, standing for True and False separately.

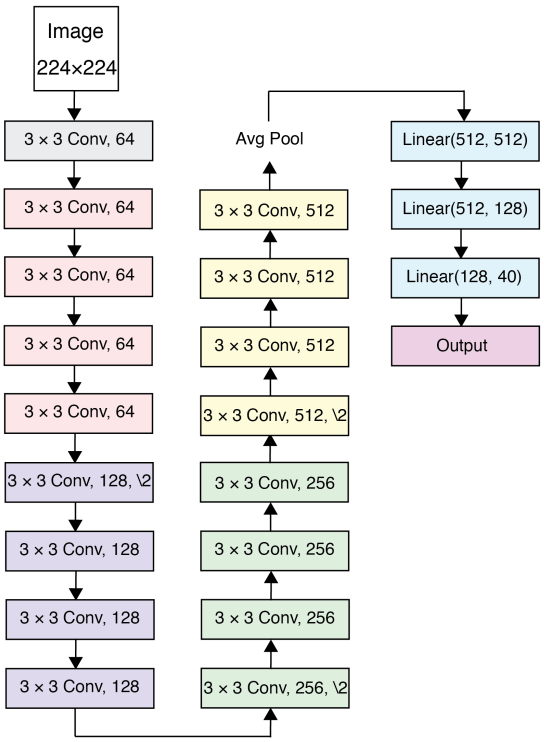
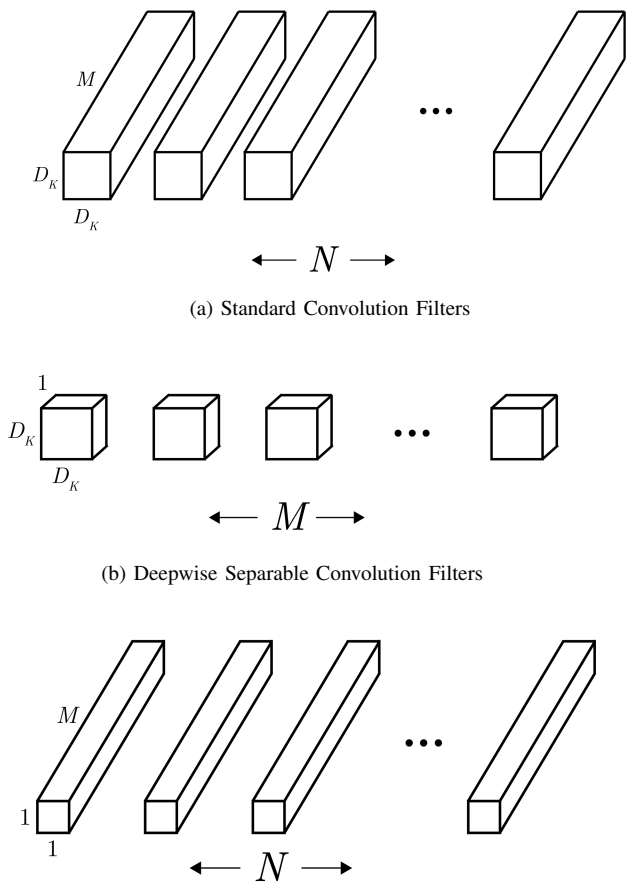


Fig. 1. Architecture of Transfer Learning Model



(a) Standard Convolution Filters

(b) Deepwise Separable Convolution Filters

(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Fig. 2. The standard convolutional filters in (a) are replaced by two layers: depthwise convolution in (b) and pointwise convolution in (c) to build a depthwise separable filter. [11]

B. MobileNets

The core layers of MobileNets are based on depthwise separable convolution, which separate the standard convolution into depthwise convolution and pointwise convolution. In this section, we introduce the principle of depthwise separable convolutions and the architecture of MobileNets.

1) Deepwise Separable Convolution: The MobileNets model is on the basis of depthwise separable convolutions. In general, the standard convolutions can do filtering and make the connection between inputs and outputs, which extract the features of inputs. The depthwise separable convolutions break down the standard convolutions into a depthwise convolution and a pointwise convolution with size as 1×1. These two steps can be finished in one step in standard convolution. But in the depthwise separable convolutions, these two steps are decomposed into two separate layers separately, which are the depthwise convolutional layers for filtering and pointwise convolutional layers for combining. The MobileNets model puts one filter into every input channel of the MobileNets, which is for the filtering process. Besides, the 1×1 pointwise convolution combines the outputs of last depthwise convolution layers. In this way, the amount of computation and the size of model can be reduced significantly. In the Fig.2, the N standard convolutional filters with $D_K \times D_K \times M$ size are factorized into M depthwise convolutional filters with $D_K \times D_K \times 1$ size and N pointwise convolutions with $1\times1\times M$ size.

Assuming the square input feature map and the square output feature map have the same spatial dimensions, when a $D_F \times D_F \times M$ feature map F is entered to a standard convolution layer, the output is a $D_F \times D_F \times N$ feature map G. D_F represents the spatial width and height of the input feature map. M represents the input depth, which is the number of input channels. If using stride one and padding, the computation of the output feature map of the standard convolution is shown as following:

$$\mathbf{G}_{k,l,n} = \sum_{i,j,m} \mathbf{K}_{i,j,m,n} \cdot \mathbf{F}_{k+i-1,l+j-1,m} \tag{1}$$

From the formula (1), we can see that the computational cost of the standard convolutions can be calculated as:

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \tag{2}$$

From the formula (2), we can see that the computational cost is influenced by four parameters: the feature map size D_F, the amount of input channels M, the amount of output channels N and the kernel size D_K.

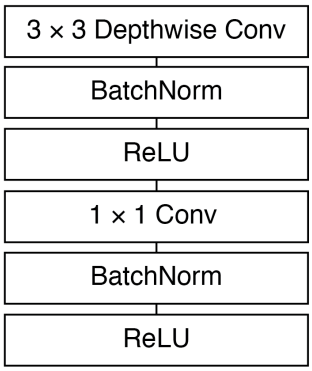


Fig. 3. Structure of Depthwise Separable Convolutions: Depthwise and Pointwise Layer with batch normalization and activation function ReLU.

The MobileNets shrinks the results and generalizes the features with arbitrary sizes and aspect ratios. In MobileNets model, the number of the output channels and the size of the kernel are isolated with each other by the depthwise separable convolutions.

In the standard convolutions, the convolutional kernels are used to filter the features and these features are combined into a new set of outputs. But these operations require large computational cost due to the complicated structure of standard convolutions. The depthwise separable convolutions of MobileNets can reduce the computational cost through separating the two steps into two individual layers, depthwise convolutions and pointwise convolutions.

The depthwise convolutional layers only filter every input channel. So, the operation of depthwise convolution is shown as following:

$$\hat{\mathbf{G}}_{k,l,m} = \sum_{i,j} \hat{\mathbf{K}}_{i,j,m} \cdot \mathbf{F}_{k+i-1,l+j-1,m} \tag{3}$$

From the formula (3), we can see that the computational cost of the depthwise convolutions can be calculated as following:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F \tag{4}$$

The pointwise convolutional layers combine the outputs of depthwise convolution linearly to produce the new features. So the formula to calculate the computational cost of pointwise convolution is shown as following

$$M \cdot N \cdot D_F \cdot D_F \tag{5}$$

Combining formula (4) and (5) together, we can get the computational cost of depthwise separable convolution is:

$$\begin{aligned} &D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F \\ &\qquad = (N + D_K \cdot D_K) \cdot M \cdot D_F \cdot D_F \end{aligned} \tag{6}$$

Comparing the computational cost of depthwise separable convolutions and standard convolution, it is obvious that the computational cost of depthwise convolution is less than standard convolution. And the reduction is shown in following equation:

$$\frac{(N + D_K \cdot D_K) \cdot M \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot N \cdot M \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K \cdot D_K} \tag{7}$$

TABLE I
ARCHITECTURE OF MOBILENETS

Type / Stride	Filter Shape	Input Size
Conv / s2	3×3×3×32	224 × 224 × 3
Conv dw / s1	3×3×32 dw	112 × 112 × 32
Conv / s1	1×1×32×64	112 × 112 × 32
Conv dw / s2	3×3×64 dw	112 × 112 × 64
Conv / s1	1×1×64×28	56 × 56 × 64
Conv dw / s1	3×3×128 dw	56 × 56 × 128
Conv / s1	1×1×128×128	56 × 56 × 128
Conv dw / s2	3×3×128 dw	56 × 56 × 128
Conv / s1	1×1×128×256	28 × 28 × 128
Conv dw / s1	3×3×256 dw	28 × 28 × 256
Conv / s1	1×1×256×256	28 × 28 × 256
Conv dw / s2	3×3×256 dw	28 × 28 × 256
Conv / s1	1×1×256×512	14 × 14 × 256
5× Conv dw / s1	3×3×512 dw	14×14×512
5× Conv / s1	1×1×512×512	14×14×512
Conv dw / s2	3×3×512 dw	14 × 14 × 512
Conv / s1	1×1×512×1024	7 × 7 × 512
Conv dw / s2	3×3×1024 dw	7 × 7 × 1024
Conv / s1	1×1×1024×1024	7 × 7 × 1024
Avg Pool / s1	Pool 7×7	7 × 7 × 1024
FC / s1	Pool 1024×40	1 × 1 × 1024
Sigmoid / s1	Classifier	1 × 1 × 40

2) Network Structure: In the MobileNets model, the core layers are depthwise convolutional layers and pointwise convolutional layers. MobileNets model has 28 layers. The architecture of MobileNets is shown in Table I. Both depthwise convolutional layers and pointwise convolutional layers use batchnorm and ReLU nonlinearities, which is shown in Fig. 3. The last fully connected layer feeds into `Sigmoid` classifier for binary classification without batchnorm and ReLU. Maxpooling, which is a kind of down sampling operation with strided convolution is handled in depthwise convolutions and the first layer. Similar to ResNet [35], a final average pooling layer is applied before the fully connected layer to reduce the spatial resolution.

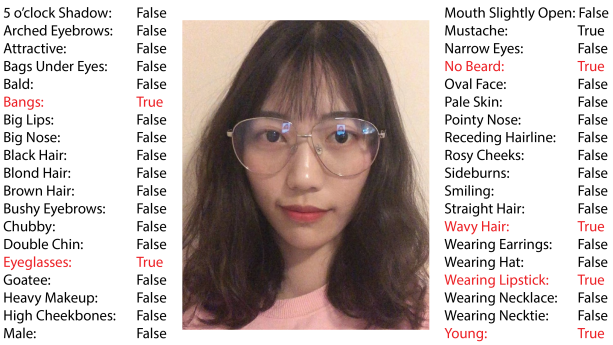
In the original MobileNets model [11], it introduces two hyper-parameters for developing a suitable model which is more smaller and faster and shrinking the structure of the model. The first one, width multiplier, is a scalar for the number of input channels and output channels, which restricts the number of parameters of each layer and makes the model thinner. The second one, resolution multiplier, is a scalar for the resolution of input images, which reduces the complexity of the internal representation of each layer by using the same multiplier. Both multipliers can reduce the size and complexity as well as the computational cost of the model, which makes the model thinner and shallower.

TABLE II
RESULTS FOR CELEBA. THE HIGHEST ACCURACY FOR EACH ATTRIBUTE IS IN BOLD.

Attributes	TNets[a]	AUX[b]	ANet[c]	MNets[d]
5 o'clock Shadow	94.3	94.51	91	94.4
Arched Eyebrows	82.7	83.42	79	84.1
Attractive	82.3	83.06	81	82.5
Bags_Under_Eyes	83.7	84.92	79	85.1
Bald	98.9	98.90	98	99.0
Bangs	95.9	96.05	95	96.0
Big_Lips	71.1	71.47	68	72.1
Big_Nose	83.5	84.53	78	84.3
Black_Hair	88.8	89.78	88	88.8
Blond_Hair	95.7	96.01	95	95.6
Blurry	96.1	96.17	84	95.9
Brown_Hair	87.3	89.15	80	88.4
Bushy_Eyebrows	92.3	92.84	90	92.1
Chubby	95.4	95.67	91	95.6
Double_Chin	96.1	96.32	92	96.2
Eyeglasses	99.7	99.63	99	99.6
Goatee	96.9	97.24	95	97.5
Gray_Hair	97.9	98.20	97	98.1
Heavy_Makeup	91.2	91.55	90	91.2
High_cheekbones	86.7	87.58	87	87.2
Male	98.2	98.17	98	98.0
Mouth_Slightly_Open	93.5	93.74	92	93.4
Mustache	96.5	96.88	95	97.0
Narrow_Eyes	86.2	87.23	81	87.2
No_Beard	96.3	96.05	95	96.1
Oval_Face	75.5	75.84	66	75.9
Pale_Skin	97.1	97.05	91	96.8
Pointy_Nose	75.9	77.47	72	76.5
Receding_Hairline	93.6	93.81	89	93.6
Rosy_Cheeks	94.9	95.16	90	94.9
Sideburns	97.2	97.85	96	97.7
Smiling	92.4	92.73	92	92.2
Straight_Hair	83.8	83.58	73	83.5
Wavy_Hair	84.5	83.91	80	83.9
Wearing_Earrings	90.2	90.43	82	90.0
Wearing_Hat	99.0	99.05	99	99.0
Wearing_Lipstick	93.9	94.11	93	94.0
Wearing_Necklace	86.5	86.63	71	87.0
Wearing_Necktie	96.9	96.51	93	96.8
Young	88.2	88.48	87	88.0
Test accuracy	90.91	91.02	87	91.12

[a]Transfer Learning Model(ResNet18 with Linear Classifier)
[b]MCNN-AUX
[c]LNets+ANet
[d]MobileNets

(a) Test Results on Our Own Facial Images(a)



(b) Test Results on Our Own Facial Images(b)

Fig. 4. Test Results for Each Attributes on Our Own Facial Images

IV. EXPERIMENTAL RESULTS

We implement both transfer learning model (ResNet18 with Linear Classifier) and MobileNets model under the framework of PyTorch [19] and train them under the GPU environment on CelebA dataset [4]. In this section, we describe the details of parameters setting in the training of both models. The experimental results are also discussed and compared.

A. Training Details

CelebA [4] is a large-scale facial images dataset containing 202,599 images with 40 different facial attributes labels. We use the cropped images of this dataset so that we do not need to detect the faces before training. For the image preprocessing, we resize all the data images to 224 $\times$ 224, since the input size of both model we developed is 224 $\times$ 224, transfer them to tensor type and normalize them. The labels of the dataset have 40 attributes and the "-1" represents not having the attribute and "1" represents having the attribute. Since we use `Sigmoid` function as the activation function, we change all the "-1" values in labels to "0". We divide the original 202,599 images of the dataset into training set, validation set and test set according to the ratio of 7:2:1.

For both transfer learning model(ResNet18 with Linear Classifier) and MobileNets model, we set learning rate as 0.001 and train the model for 50 iterations. Since all labels have two different state values, and we assume it as a single-label training. We use the binary cross entropy as the loss function to solve this binary classification problem for each label. During the training process, we keep comparing the accuracy of each epoch and store the one with best accuracy as the best model. Then we use the best model to test on the data of test set and get the final accuracy of each attributes as well as the average one.

B. Results

The numbers of images for each attributes are not balanced in the CelebA [4] dataset. Some attributes only exist in a few images, while some attributes are very common that almost show up in every image. Besides, some attributes have relationship between each other, for example, the attributes of hair color could only have one True, if black hair is

True, the other hair colors will be False according to the labels of CelebA [4] dataset. Therefore, it is not suitable to take the average accuracy into consideration only. To solve this problem, we compute the accuracy of each attribute and compare them among some state-of-the-art models. From the results, we found that some attributes with unapparent features are hard to recognize, like oval face, big-lips and pointy noses, so they have low accuracy around 70% to 85%. In contrast, some attributes, for example, eyeglasses, bald and male, have obvious characters and are easy to recognize, so their accuracy are high at almost 100%. The comparison results of each attributes with different models are shown in Table II.

In Table II, we compare four models, the transfer learning model (ResNet18 with linear classifier), MobileNets [11], MCNN-AUX [23] and LNets+ANet [4]. The LNets+ANet [4] model has the lowest accuracy in each attribute and MobileNets performs even better than other two models with similar performance. The performance of MobileNets in most attributes are good, and it achieves a state-of-the-art performance with reduction of computation and parameters.

We use our own images to test on the MobileNets model. The results are shown in Fig. 4. In Fig. 4(a), the attributes that have unapparent characters, like attractive and oval face, are True. In Fig. 4(b), the attributes that have unapparent characters, like big lips and wavy hair, are True. We can see that the MobileNets can detect unapparent features successfully. But there still exists some inaccuracy. In Fig. 4, any attributes about hair color did not be detected successfully.

V. Conclusion

Facial attributes classification has been solved in various deep neural networks. In this paper, we implement two models, a transfer learning model, ResNet18 with linear classifier for comparison and MobileNets, which is light weight but efficient, to classify 40 facial attributes of the dataset CelebA [4]. The MobileNets is built on the depthwise separable convolutions that contains depthwise convolutional layers and pointwise convolutional layers to achieve the reduction of network size and parameters. We compare the performance of transfer learning model(ResNet18 with Linear Classifier) and MobileNets model with another two popular models, MCNN-AUX [23] and LNets+ANet [4]. The performance of MobileNets on 40 attributes is similar to the transfer learning model and MCNN-AUX [23]. The MobileNets model has been proved that it is smaller and has less computational cost with state-of-the-art performance.

REFERENCES

[1] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," pp. 365–372, 2009.

[2] B.-C. Chen, C.-S. Chen, and W. H. Hsu, "Cross-age reference coding for age-invariant face recognition and retrieval," pp. 768–783, 2014.

[3] N. Zhuang, Y. Yan, S. Chen, H. Wang, and C. Shen, "Multi-label learning based deep transfer neural network for facial attribute classification," *Pattern Recognition*, vol. 80, pp. 225–240, 2018.

[4] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

[5] M. Ehrlich, T. J. Shields, T. Almaev, and M. R. Amer, "Facial attributes classification using multi-task representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 47–55.

[6] N. Cherniavsky, I. Laptev, J. Sivic, and A. Zisserman, "Semi-supervised learning of facial attributes in video," in *European Conference on Computer Vision*. Springer, 2010, pp. 43–56.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[8] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[9] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," *arXiv preprint arXiv:1706.02677*, 2017.

[10] J. Jin, A. Dundar, and E. Culurciello, "Flattened convolutional neural networks for feedforward acceleration," *arXiv preprint arXiv:1412.5474*, 2014.

[11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[12] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev, "Panda: Pose aligned networks for deep attribute modeling," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1637–1644.

[13] E. M. Rudd, M. Günther, and T. E. Boult, "Moon: A mixed objective optimization network for the recognition of facial attributes," in *European Conference on Computer Vision*. Springer, 2016, pp. 19–35.

[14] P. Luo, Z. Zhu, Z. Liu, X. Wang, and X. Tang, "Face model compression by distilling knowledge from neurons," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005.

[16] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1991, pp. 586–591.

[17] Y. Zhong, J. Sullivan, and H. Li, "Leveraging mid-level deep representations for predicting face attributes in the wild," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3239–3243.

[18] S. Kang, D. Lee, and C. D. Yoo, "Face attribute classification using attribute-aware correlation map and gated convolutional neural networks," in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 4922–4926.

[19] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[20] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[22] Y. Huang, W. Wang, L. Wang, and T. Tan, "Multi-task deep neural network for multi-label learning," in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 2897–2900.

[23] E. M. Hand and R. Chellappa, "Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[24] I. Pillai, G. Fumera, and F. Roli, "Designing multi-label classifiers that maximize f measures: State of the art," *Pattern Recognition*, vol. 61, pp. 394–404, 2017.

[25] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Advances in neural information processing systems*, 2007, pp. 41–48.

[26] S. J. Hwang, F. Sha, and K. Grauman, "Sharing features between objects and their attributes," in *CVPR 2011*. IEEE, 2011, pp. 1761–1768.

[27] W. Wei, C. Tian, S. J. Maybank, and Y. Zhang, "Facial expression transfer method based on frequency analysis," *Pattern Recognition*, vol. 49, pp. 115–128, 2016.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[30] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2200–2207.

[31] ——, "Transfer joint matching for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1410–1417.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[33] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[34] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4820–4828.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.