DOI: http://dx.doi.org/10.26483/ijarcs.v17i4.7559

ISSN No. 0976-5697

Volume 17, No. 4, July-August 2026

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

GENDER RECOGNITION THROUGH FACE IMAGES USING CONVOLUTIONAL NEURAL NETWORKS

Priya Laxmi, Rahul Gautam
Sant Longowal Institute of Engineering & Technology,
Longowal, Punjab (India)

Abstract: Gender recognition from facial images is widely used in the field of computer vision for surveillance which is easy for a human but challenging for machines. In computer vision, some factors such as differences in illumination, angle, occlusion, and expressions cause the problem of low accuracy. So, a Convolutional Neural Network (CNN) based transfer learning method is being applied in this work since CNN performs automatic feature extraction and learns complex high-level features in image classification applications. The proposed method efficiently classifies gender classes (male and female) on 'Adience': a benchmark face image dataset publicly available on the Kaggle website. This work interrogates seven different pre-trained CNNs used as base networks in the proposed transfer learning method. For the transfer of the pre-acquired knowledge, five different sets of layers (named experiments-1 to 5) have been tested with every model, and their performance is evaluated. Among these, experiment-4 has performed best with all models. However, DenseNet-121 has achieved the highest mean accuracy of 92.83% with a set of layers in experiment-4. We evaluate the performance of the proposed model using five-fold cross-validation, and the final mean accuracy is compared with previous methods in the literature.

Keywords: Face images, Gender classification, Deep Learning, Convolutional Neural Networks.

1 INTRODUCTION

The human face reveals valuable information about identity, gender, age, expressions, and emotions. Gender recognition from facial images is one of the basic human capabilities. Extending it to the machine is a crucial task in computer vision that has received much attention [1]. Various real-world applications such as biometrics, visual surveillance, demographic collection, content-based searching, human-computer interaction (HCI) systems, and advertising use gender information [2]. Gender recognition is complex due to changes in face expressions, pose, face-image appearance, background, and visual angles. In unconstrained imaging conditions, it is even more difficult.

The most common deep learning-based methods used in computer vision are Convolutional Neural Networks (CNNs). CNNs have been used in many applications such as object classification, speech recognition [3], hand-written digit recognition [4], human action recognition [5], and face detection [6]. Recently, CNNs have also been used for soft biometrics tasks such as gender classification [7]. Presently, the transfer of knowledge from pre-trained CNNs is the current state-of-the-art. Transfer learning is a method of reusing pre-trained deep learning models (trained on a comparatively large-sized dataset) for the target problem, which is under investigation. It helps in reducing the size of the training data, training time, and computational expenses [8]. CNN-TL (Convolutional Neural Network based-Transfer Learning) shows an improvement in the performance of standard CNN variants of VGG, ResNets, DenseNet, and Inception-v3 models. Applying on to the Adience dataset for the task of gender recognition shows significant improvement. The Adience dataset consists of unconstrained facial images acquired in the wild conditions, showing that the proposed model is capable of detecting gender from facial images acquired in uncontrolled environments.

The main contributions of this work are summarized as follows:
In this work, we propose a Transfer Learned-based Deep CNN model to perform gender recognition. In this work, we use the seven different pre-trained deep learning models such as VGG-16, VGG-19, ResNet-50, ResNet-152, Inception-v3, DenseNet-121, and DenseNet-201. All these models are pre-trained on the ImageNet dataset. After that, we fine-tune the model on the Adience dataset. For this work, we developed five different head network architectures for all deep learning models. After that, we apply head network architecture for the experiment. Finally, the results obtained from these models are compared with some existing methods performed on gender recognition using the Adience dataset.

The rest of the paper is organized as follows. Section 2 describes brief information about the literature work. Section 3 describes the material and methods, the dataset used for gender recognition, and various CNN models used in the proposed work. Section 4 shows the results of different experiments for different deep learning models and compares the results with previous methods study for gender recognition. Section 1 presents the conclusion and future work of this study.

2 LITERATURE SURVEY

In literature, many researchers have worked on gender recognition from facial images. Initial studies were based on the Local Binary Pattern (LBP) centered feature extraction methods that were proposed by Ojala et al. [9] for texture classification. LBP features can detect microstructures such as corners, edges, and spots. The following methods by researchers have used LBPs for gender recognition. To extract features from face images, Lian and Lu [10] used LBP with SVM classifier for gender classification. Alexandre [11] performed gender classification based on shape, texture, and intensity features, where histograms have been used for shape features and LBPs for texture features. Tapia and Perez [12] applied mutual information-based feature selection and fused LBP features to improve the accuracy of gender classification. Shan [13] applied Adaboost to select the LBP features. They obtained better results by applying SVM with boosted LBPs. Shih [14] used a precise patch histogram to improve the robustness and accuracy of gender classification. They used the active appearance model (AAM) for face alignment. Patel et al. [15] used a variant of LBP, compass LBP (coLBP) feature extraction method for gender classification. Their method extracts the discriminative information from four different levels, gradient, regional, global, and directional levels. Similarly, Bhattacharyya et al. [2] also used the coLBP method for gender recognition. They divide the face image into different regions such as the forehead, left eye, right eye, and lips.

Then SVM classifier is used to evaluate the probability scores of each region. After that, to improve the overall classification accuracy, the classification score obtained from each region is combined with a genetic algorithm. Although much work has been done on LBP-based gender recognition methods, LBPs are mainly gray-level descriptors that are generally used for the extraction of texture-based features. Other than this, LBP does not represent spatial information well. For their efficient implementation, they are needed to apply to different regions of the images. Therefore, for facial images that are mainly in RGB color space and have linked spatial features, researchers have proposed more sophisticated methods based on CNN and SVM-based classifiers.

Moghaddam and Yang [16] recognized gender from low-resolution face images using SVM. Similarly, Baluja and Rowley [17] used the Adaboost classifier to recognize gender from low-resolution greyscale images. Jain et al. [18] recognized gender from frontal facial images. They used Independent Component Analysis (ICA) feature extraction method to represent each image. Berbar [19] used three feature extraction methods such as discrete cosine transform (DCT), gray-level co-occurrence matrix (GLCM), and 2D-wavelet transform feature extraction methods for gender classification. Jabid et al. [20] used Local Directional Pattern (LDP) to represent facial images for gender classification. The face image is divided into sub-regions to represent the face image. Lu and Shi [21] proposed a fusion-based method for gender classification using support vector machines based on features extracted using 2D-PCA(Two Dimensional Principal Component Analysis). Bui et al. [22] also used 2D-PCA with a support vector machines classifier for face gender classification. Li et al. [23] used both facial features and other external information (hair and clothing) to identify the gender of a person. They used five facial features such as forehead, eyes, nose, mouth, and chin instead of using the whole face. For each feature, they used shows the number of male and female images in each fold used in the proposed work. a support vector machine classifier to classify the gender of a person.

Recently, CNNs based methods have been mostly used for the gender recognition task. The CNN-based methods are applied to both feature extraction and classification for automatic recognition. Mansanet et al. [24] proposed a new model called Local Deep Neural Network (LDNN), which is based on local features and deep architecture and shows that LDNN performs better than other deep learning methods. Dhomne et al. [25] used VGGNet architecture for automatic gender recognition using frontal facial images. Lin et al. [26] proposed a Taguchi-based AlexNet network for face gender classification, which is used to optimize the parameters of AlexNet architecture. Janahiraman and Subramaniam [27] used a convolutional neural network-based deep learning model for gender recognition. They compared various types of deep learning models, including VGG-16, ResNet-50, and MobileNet. To evaluate the performance of these models, they used their dataset containing Asian Faces, a variety of Malaysians, and some Caucasians.

The Adience dataset is created in an uncontrolled environment, proposed by Eidinger et al. [28] to perform the face attribute estimation such as age and gender. They used a dropout-support vector machine (SVM) classifier for age and gender estimation to avoid over-fitting. Levi & Hassner [7] showed learning representation through deep CNNs to improve the accuracy of gender classification. Rodriguez et al. [29] proposed feed-forward attention mechanisms to discover the most discriminative information about the face.

Through the use of attention mechanisms, they improved the performance of CNNs in terms of accuracy and robustness. Zhang et al. [30] proposed a residual of residual networks (ROR) for gender recognition. The Hybrid architecture of CNN and Extreme Learning Machine (ELM) is proposed by Duan et al. [31] to integrate the synergy of two classifiers for gender classification. Afifi and Abdelhamed [32] used isolated facial features and a contextual feature called foggy face to train deep CNNs. Khan et al. [33] divide the face image into six different parts such as hair, eyes, nose, mouth, skin, and back using conditional random fields (CRFs), and then a random decision forest (RDF) classifier is used to classify the face image.

3 MATERIAL AND METHODS

3.1 Dataset Description

The Adience dataset was provided by Eidinger et al. in 2014 [28]. The dataset consists of 26580 images of 2284 individuals with frontal and non-frontal facial images of people from different age groups, races, and countries. The dataset is collected in unconstrained environments and used for age and gender classification. The images in the dataset are captured from smartphone devices. All images contain varying in pose, expression, background, illumination, or lighting, which makes the dataset more challenging. The gender information is given along with the dataset.

Table 1 Folder-wise details of images in the dataset

	Male Images	Female Images	Total Images
Fold_0	2047	1948	3995
Fold_1	1611	1998	3609
Fold_2	1363	1774	3137
Fold_3	1502	1804	3306
Fold_4	1597	1848	3445
Total Image	**8120**	**9372**	**17492**

Figure 1 shows the sample images of the dataset. The image shows many variations in visual angles, expressions, image quality, and image background, which make the dataset more challenging.

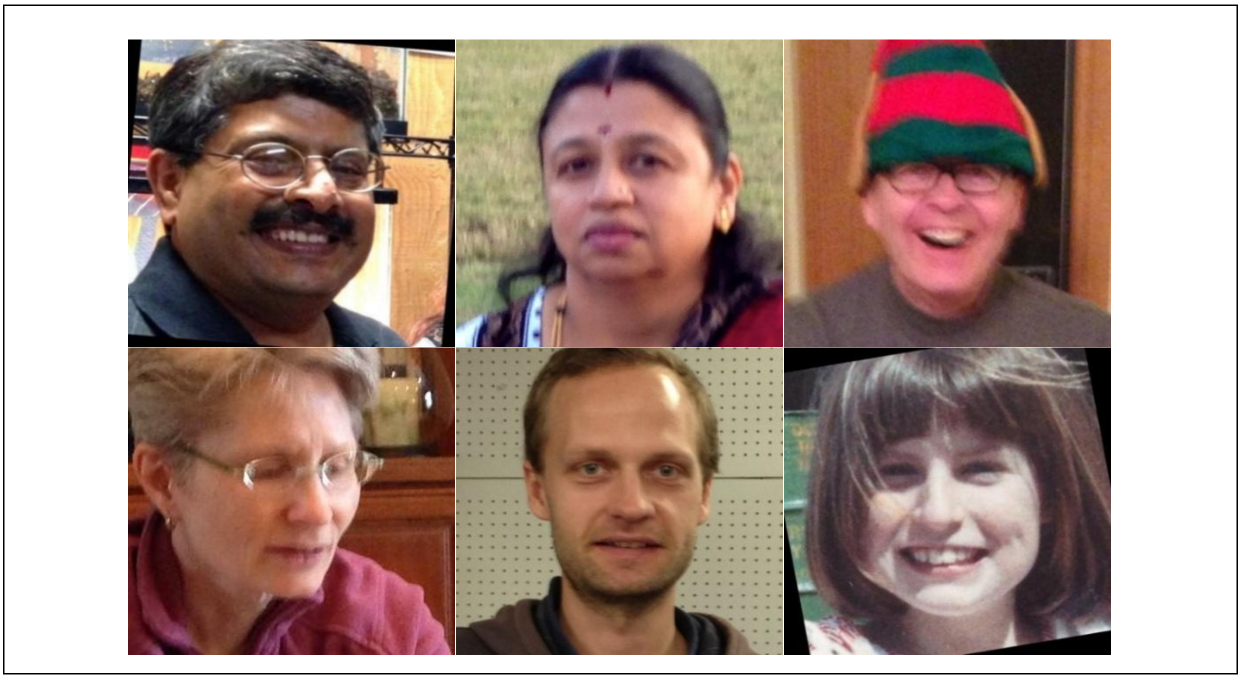

Figure 1 Sample images from the Adience dataset

3.2 Proposed Methodology

The proposed method is a four-stage pipelined process that includes pre-processing the input dataset, training the selected pre-trained CNN model after appending the transfer head network, validation, testing the trained models, calculating considered evaluation metrics, and at last comparison of the different CNN models. The input dataset used in this work is taken from the Kaggle website which is publicly available.

Figure **2** shows the block diagram for the workflow of the model, and Figure 3 shows a flowchart for the proposed model.

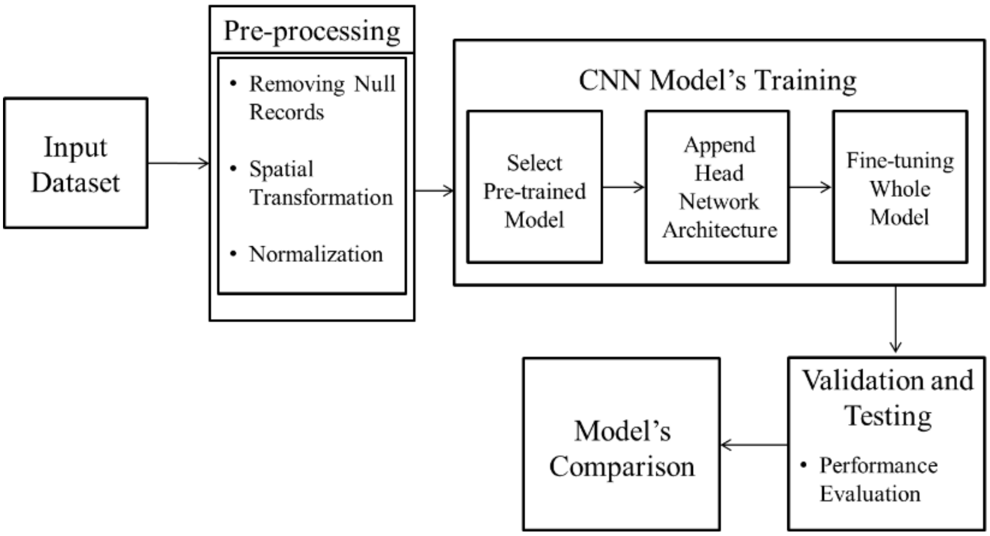

Figure 2 Block diagram of the workflow

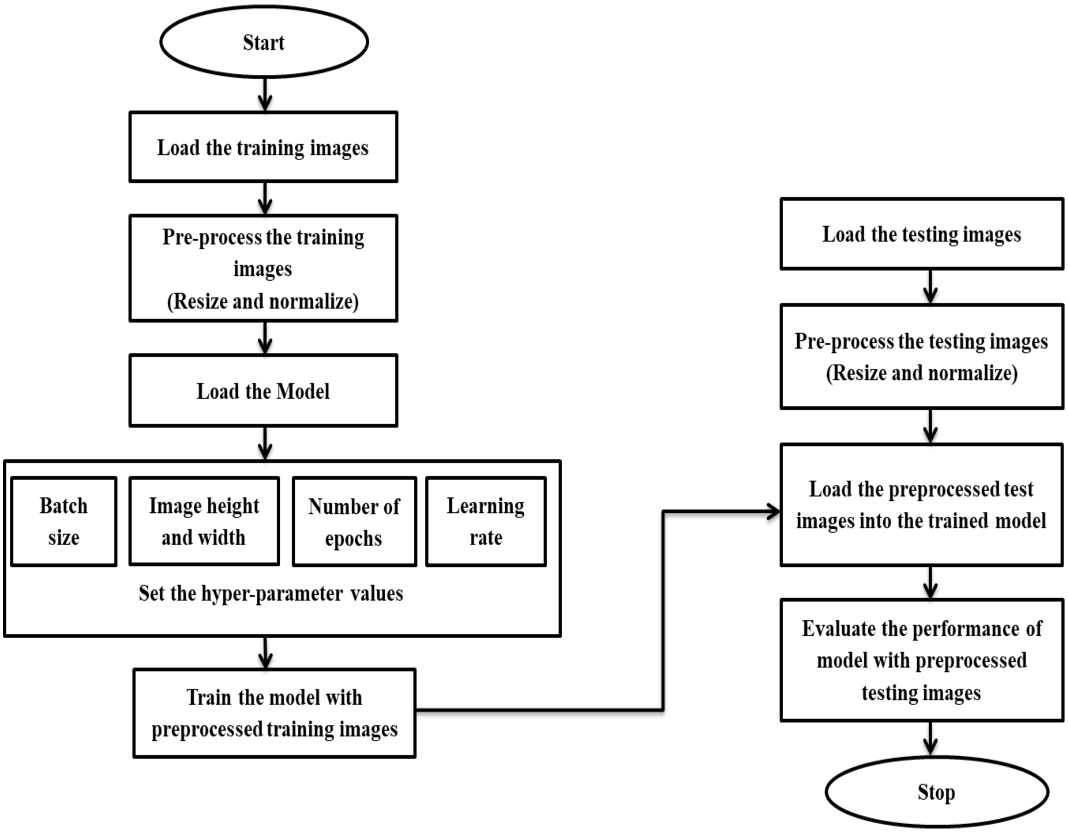

Figure 3 Flow chart for the proposed model

3.2.1 Pre-processing

In the pre-processing step, we remove noise due to factors such as image acquisition, illumination, or lighting conditions. In the dataset, many images are not labeled or do not contain gender information. For this work, we removed these images, and only labeled images are used for the training and testing of models. All the images in the dataset are of different sizes; training and testing of CNN-based methods require the same size of images to be fed. We have performed spatial transformations of the images, and all the images are modified to 224×224 size before processing. The resized dataset is normalized to make it easy and efficient for CNNs to learn better features.

3.2.2 Transfer Learning with Pre-trained Models

Because of the high computational requirements of deep neural network models, the training and testing of deep models are time-consuming. So that we use the transfer learning method, through transfer learning, we perform training of models in less time.

Transfer learning is a method used in deep learning to reuse the pre-trained model for a new but related problem. By using this method, we can produce efficient deep-learning models in less time. All the pre-trained models are trained on the ImageNet dataset. In transfer learning, we add a fully connected layer and train on the data. It learns all the necessary parameters from the ImageNet dataset and requires a small amount of training on a fully connected layer.

In this work, we have used seven different pre-trained CNN models. Models include VGG-16, VGG-19, ResNet-50, ResNet-152, Inception-v3, DenseNet-121, and DenseNet-201. All the models are pre-trained on the ImageNet dataset. The pre-learned weight and parameters of all pre-trained models are available in the Tensorflow library.

In this work, we changed the last fully connected layer (classification layer). All the pre-trained models consist of 1000 classes in the fully connected layers from the ImageNet dataset. Instead of 1000 classes, a fully connected layer is added in this work. We only have used the feature extraction layer from the pre-trained model. After that, we add an additional layer for the classification. All the pre-trained models are explained below:

- **VGGNet** stands for Visual Geometric Group Convolution Neural Network, which was proposed by Simonyan and Zisserman in 2014 [34]. The VGGNet was proposed after AlexNet, and the main focus of this model is on the effect of convolutional neural network depth on accuracy. The model replaces large filters with many small filters, which increases the depth of the network using 3×3 convolutional filters, which automatically leads to improvement in recognition accuracy. In this work, we used variants of the VGGNet model named VGG-16 and VGG-19. VGG-16 consists of 16 layers, and VGG-19 consists of 19 layers. There is the only difference between the number of layers in the model. VGG-16 has 138,357, 544 parameters and VGG-19 has 143,667,249 parameters.
- **ResNet** stands for a residual network which was proposed by Microsoft corporation in 2015 [35]. The ResNet uses a residual connection to make the output of the previous layers the input to the next layer. The ResNet model uses skip connections to remove the vanishing gradient problem. Skip connections simplify the neural network, using fewer layers in the initial training. This speeds learning by

reducing the impact of vanishing gradients, as there are fewer layers to propagate through. ResNet-50 and ResNet-152 are used for the proposed work. ResNet-50 consists of 50 deep layers, and ResNet-152 consists of 152 deep layers. ResNet-50 has about 23 million parameters, and ResNet-152 has about 60 million parameters.

- **Inception** models are developed by Google [36]. There are various versions of the inception model available, but for the proposed work, we used the Inception-v3 model. The Inception-v3 uses the Inception modules. The model has fewer parameters and is more efficient than other models like AlexNet, VGGNet, ResNet, etc. It allows the use of multiple types of filter sizes instead of being restricted to single filter size. Inception-v3 has 23, 851, 784 parameters.
- **DenseNet** is a Densely connected convolutional neural network which is proposed in 2017 by Gao Huang et al. [37]. The model uses a dense block to create a dense connection between the layers. In DenseNet, all the layers are directly connected with each other. DenseNet model consists of more than 200 layers; for the proposed work, we have used variants of DenseNet named DenseNet-121 and DenseNet-201.

3.2.3 Head Network Architecture

After selecting the pre-trained models, the head network is appended at the end of the classification model. In the head network, we have added different CNN layers such as Average Pooling, Flatten, Batch-Norm, ReLU layer, Dropout, and a fully connected layer. Average pooling layers are used to calculate the average for each patch of the feature map. It is used to downsample the size of the feature map. Flatten layer converts the data into a 1-D array for inputting it to the next layer. The ReLU layer allows the model to learn faster and perform better. It is used to overcome the vanishing gradient problem. Batch-Norm Layer is used to normalize the output of the previous layers. It allows every layer of the neural network to do learning more independently. Through this layer, learning becomes more efficient, and it is also used as a regularization to avoid overfitting problems. Dropout layers are used to prevent the model from overfitting. This layer randomly set the outgoing edges of hidden layers to 0 at each update of the training phase.

Figure 4 shows the proposed model performed with experiment- 4. In this work, we have appended the head network after the feature extracted by the pre-trained DenseNet-121 model. After the features are extracted by the model, we have added the average pooling layer to downsample the size of the feature map. Then we add the flatten layer to convert the features into a 1-D array. After that, we apply the Batch-Norm layer to normalize the output of the previous layer, and 512 units of a fully connected layer are added. After the fully connected layer, ReLU Layer has been added for faster performance of the model; then again, we added one more Batch-Norm layer. Then Dropout layer is added with a probability of 40%, and this layer is used to prevent the model from overfitting. Finally, we propose to add a fully connected layer for the classification of the model.

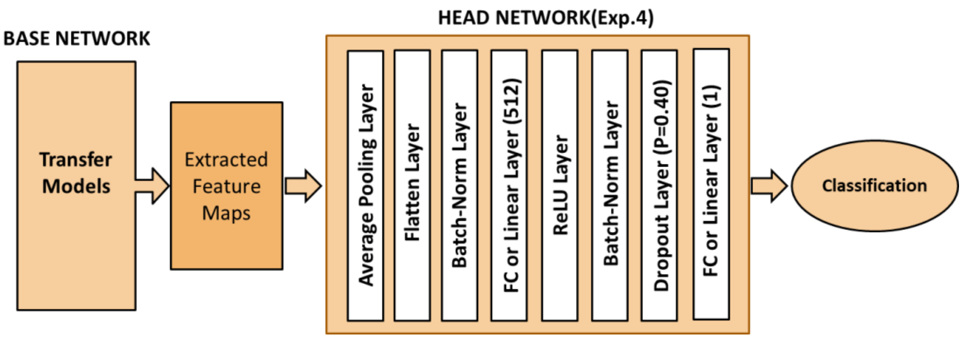

Figure 4 Proposed Model

3.2.4 Fine-Tuning the Model

After appending the head network to the model, the model is fine-tuned on the dataset using the Adam optimizer, and the loss function is calculated using binary cross-entropy. The training process is performed with a batch size of 64. The network includes a dropout layer with a ratio of 0.40 (40% chance of setting a neuron's output value to zero). We perform training on the different learning rates. Initially, the model is trained for 20 epochs with a learning rate of 0.001; after that, the learning rate is set to 0.0001 for 20 epochs, and at the end, the learning rate is set to 0.00001 for ten epochs. After the completion of the training, the model is evaluated on the test dataset.

4 RESULTS AND DISCUSSIONS

This work is implemented using the Tensorflow framework. Training of all models is performed on the GPU instance of the Google Colaboratory service (Colab Pro). It has provided a GPU RAM of 12 GB, CPU RAM, and disk memory, and the graphics card allocated by the service providers is P100.

4.1 Performance Metrics

The performance of the models is estimated with metrics such as accuracy, precision, recall, F1-score, and AUCROC curve.

Accuracy measures the total number of samples that were correctly classified by the model. It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision measures the ratio of the predicted true values that are actually true. It is calculated as follows:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall measures the ratio of the true observations that the model will classify as true. It is calculated as follows:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

F1-score is the weighted average of recall and precision combining both recall and precision, and it is calculated as follows:

$$F1 - score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (4)$$

4.2 Experimental evaluation

In this paper, five different methodologies are used on the Adience dataset with different deep-learning models. In all these methodologies, we have used a different set of layers for the Head Network shown in Figure 5. In two experiments (Experiment 1 and Experiment 2), we did not add the average pooling layer. After the features were extracted from the pre-trained models, we added the Flatten layer to convert the features into a 1-D array. Then we added the fully connected layer of 512 units followed by the ReLU layer. Batch-Norm layer and Dropout layer (with a probability of 40%) are added in Experiment 1, which are not included in Experiment 2. Finally, we added a classification layer in both experiments.

In methodologies 3 to 5 (Experiments 3, 4, and 5), firstly, we add the average pooling layer to downsample the feature maps, then Flatten layer is added to convert the features into a 1-D array. Then the Batch-Norm layer is added in methodologies 4 and 5 (Experiment 4 and 5), which is not included in Experiment 3. In methodology 5 (Experiment 5), after the Batch-Norm layer, one additional dropout layer is added with a probability of 20%. The fully connected layer is added in all the experiments. After adding the fully connected layer ReLU layer, Batch-Norm layer, Dropout layer (with the probability of 40%), and the proposed classification layer are added in all three experiments for the classification.

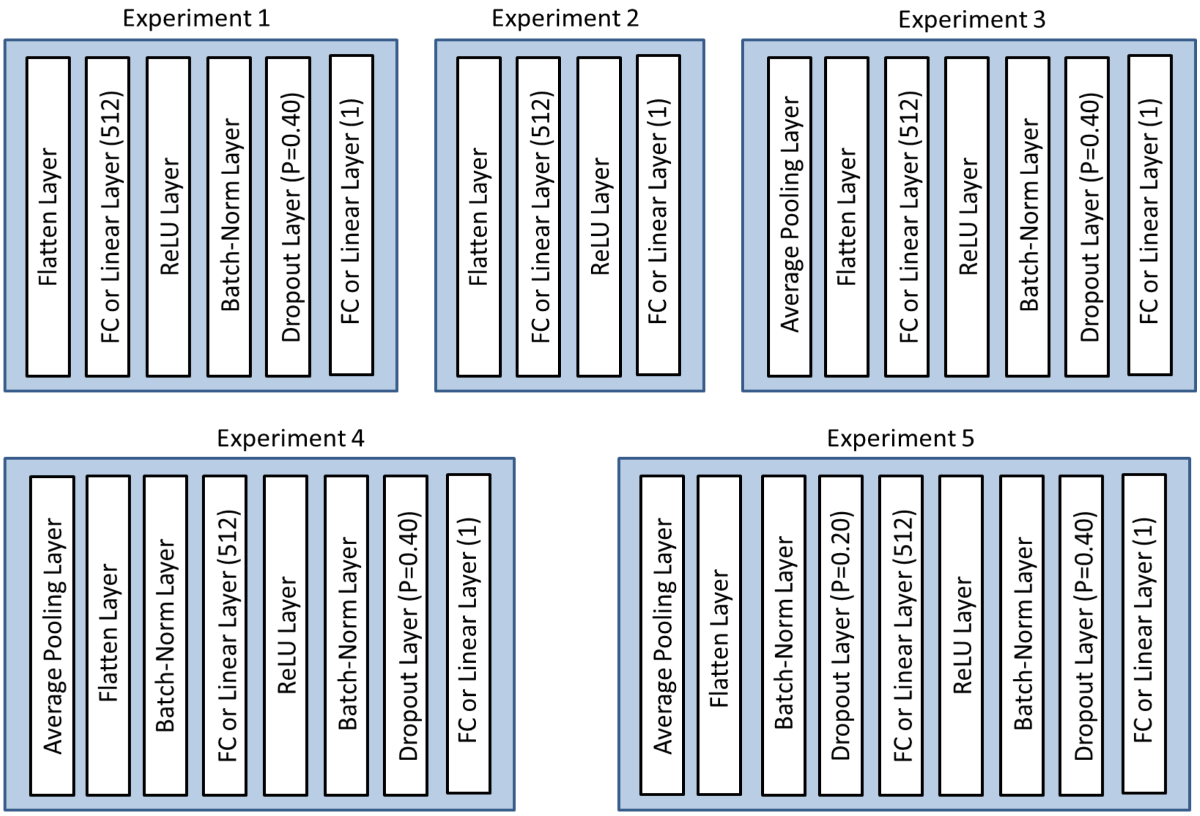

Figure 5 Different sets of layers for Head Network

The experiments are performed using the five-fold cross-validation procedure.

Table 2 shows the mean accuracy of all the methodologies with different experiments and also shows the precision, recall, and F1-score values. From these experiments, we have observed that the DenseNet-121 model with Methodology-4 has performed better than other methodologies.

Table 2 Performance of the Proposed Methodologies

	Model	Precision	Recall	F1-score	Accuracy
Methodology 1	**VGG-16**	0.8514	0.8292	0.8397	**0.8548**
	VGG-19	0.8460	0.8199	0.8324	**0.8480**
	ResNet-50	0.8941	0.8833	0.8884	0.8977
	ResNet-152	0.8622	0.8349	0.8481	0.8631
	Inception-v3	0.9067	0.8968	0.9017	0.9103
	DenseNet-121	0.9187	0.9127	0.9155	0.9227
	DenseNet-201	0.8715	0.8763	0.8757	0.8848
Methodology 2					
	VGG-16	0.8287	0.8201	0.8236	0.8387
	VGG-19	0.7761	0.7772	0.7761	0.7960
	ResNet-50	0.8957	0.8638	0.8791	0.8883
	ResNet-152	0.8741	0.8695	0.8716	0.8828
	Inception-v3	0.9104	0.8985	0.9059	0.9125
	DenseNet-121	0.9254	0.9143	0.9197	0.9267
	DenseNet-201	0.8818	0.8646	0.8726	0.8833
Methodology 3					
	VGG-16	0.8544	0.7771	0.8138	0.8369
	VGG-19	0.8406	0.7690	0.8024	0.8249
	ResNet-50	0.9085	0.9010	0.9010	0.9119
	ResNet-152	0.8816	0.8770	0.8808	0.8891
	Inception-v3	0.9107	0.9088	0.9095	0.9169
	DenseNet-121	0.9213	0.9144	0.9177	0.9248
	DenseNet-201	0.9129	0.8958	0.904	0.9132
Methodology 4					
	VGG-16	0.8452	0.8311	0.8376	0.8519
	VGG-19	0.8309	0.8311	0.8307	0.8445
	ResNet-50	0.9121	0.9064	0.9091	**0.9168**
	ResNet-152	0.8895	0.8741	0.8814	**0.8911**
	Inception-v3	0.9119	0.9151	0.9131	**0.9200**
	DenseNet-121	0.9246	0.9188	0.9234	**0.9283**
	DenseNet-201	0.9143	0.9109	0.9123	**0.9190**
Methodology 5					
	VGG-16	0.8261	0.8212	0.8232	0.8384
	VGG-19	0.8338	0.7978	0.8148	0.8338
	ResNet-50	0.9056	0.9004	0.9027	0.9112
	ResNet-152	0.8595	0.8611	0.8601	0.8720
	Inception-v3	0.9094	0.9076	0.9082	0.9156
	DenseNet-121	0.9225	0.9175	0.9199	0.9269
	DenseNet-201	0.9115	0.9103	0.9101	0.9169

Figure 6 shows the area under the ROC curve for the proposed methodology. We represent the best gender recognition accuracy obtained with the Adience dataset using Methodology-4. The ROC curve shows the accuracy obtained by each fold and also shows the mean accuracy of all folds.

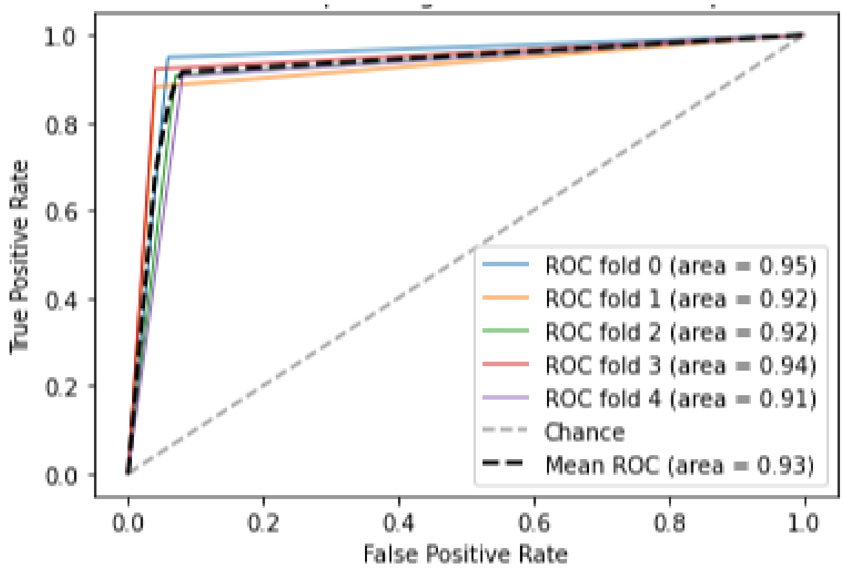

Figure 6 Area under ROC curve for DenseNet-121 model using Methodology-4

Table 3 shows the mean accuracy of models obtained from five different experiments. It clearly shows that the proposed method, the DenseNet-121 model with Experiment-4, performed better than all other models and achieved 92.8% accuracy.

Table 3 Mean accuracy using five-fold cross-validation

	Methodology 1	Methodology 2	Methodology 3	Methodology 4	Methodology 5
VGG-16	**0.85476**	0.83870	0.83692	0.85186	0.83836
VGG-19	**0.84802**	0.82600	0.82496	0.84446	0.83380
ResNet-50	0.89766	0.88928	0.91192	**0.91676**	0.91116
ResNet-152	0.86314	0.88284	0.88912	**0.89112**	0.87204
Inception-v3	0.91030	0.91246	0.91698	**0.92000**	0.91556
DenseNet-121	0.92272	0.92674	0.92476	**0.92826**	0.92690
DenseNet-201	0.88480	0.88326	0.91322	**0.91900**	0.91698

Table 4 shows the results available on the Adience dataset for gender recognition. Researchers used different methods for gender recognition. The deep learning-based methods perform better than the other methods. We have compared the proposed methodology with the existing methods using the Adience dataset. The proposed methodology has achieved better results than the state-of-the-art methods.

Table 4 A comparison of the results on the Adience dataset

Authors	**Features**	**Classifier**	**Accuracy**
Eidinger et al. [28]	LBP+FPLBP	SVM	76.10 %
Levi et al. [7]	Learned	CNN (5 layered)	86.80 %
Wolfshaar et al. [38]	Deep Features	CNN-SVM	87.20 %
Ozbulak et al. [39]	Deep Features	CNN-TL (VGG)	92.00 %
Duan et al. [31]	Deep Features	CNN-ELM	88.20 %
Proposed Methodology	**Deep Features**	**CNN (DenseNet-121)**	**92.83 %**

5 CONCLUSIONS AND FUTURE SCOPE

In this work, we have proposed a transfer learning-based classification method using pre-trained CNNs for gender recognition from human-face images. The Adience dataset considered in this study is a large-sized face image dataset of approximately 26K images, among which about 17.5K images are labeled for supervised learning. We have performed a comparative analysis of seven different pre-trained CNN models available in deep learning libraries. We have designed five different head network architectures for knowledge transfer from pre-trained CNNs on target tasks and have compared their performance to find the best transfer network architecture. Experimental analysis suggests that the head network architecture used in methodology 4 has performed best. Among used CNN models, DenseNet-121 with methodology 4 transfer architecture has achieved the best accuracy of 92.83%. Precision, Recall, and F1-Score are 92.46%, 91.88%, and 92.34%, respectively. The performance of the proposed methodology compared with current state-of-the-art methods has shown the superiority of the proposed methodology, where the five-fold mean accuracy surpasses the previous best results available in the literature by 1%. In the future, we are working on age recognition.

Statements And Declarations:

Funding: The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Competing interests: The authors have no financial or non-financial interests to declare that are relevant to the content of this manuscript.

Author Contributions: All authors contributed to the study conception, design, material preparation, data collection, and analysis. All authors read and approved the final manuscript.

Compliance with Ethical Standards: No animal research was conducted by any of the authors in this manuscript.

Data availability*:* The data generated or analyzed during the current study are available from the corresponding author on reasonable request.

REFERENCES

[1] A. Dantcheva, P. Elia, and A. Ross, "What else does your biometric data reveal? A survey on soft biometrics," IEEE Trans. Inf. Forensics Secur., vol. 11, no. 3, pp. 441–467, 2016, doi: 10.1109/TIFS.2015.2480381.

[2] A. Bhattacharyya, R. Saini, P. P. Roy, D. P. Dogra, and S. Kar, "Recognizing gender from human facial regions using genetic algorithm," Soft Comput., vol. 23, no. 17, pp. 8085–8100, Sep. 2019, doi: 10.1007/s00500-018-3446-9.

[3] R. Wason, "Deep learning: Evolution and expansion," Cogn. Syst. Res., vol. 52, pp. 701–708, 2018, doi: 10.1016/j.cogsys.2018.08.023.

[4] D. Cires and U. Meier, "Multi-column Deep Neural Networks for Image Classification," pp. 3642–3649, 2012.

[5] S. Ji, W. Xu, M. Yang, and K. Yu, "3D Convolutional neural networks for human action recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 1, pp. 221–231, 2013, doi: 10.1109/TPAMI.2012.59.

[6] S. S. Farfade, M. Saberian, and L. J. Li, "Multi-view face detection using Deep convolutional neural networks," ICMR 2015 - Proc. 2015 ACM Int. Conf. Multimed. Retr., pp. 643–650, 2015, doi: 10.1145/2671188.2749408.

[7] G. Levi and T. Hassncer, "Age and gender classification using convolutional neural networks," in 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Jun. 2015, vol. 928, no. 3, pp. 34–42, doi: 10.1109/CVPRW.2015.7301352.

[8] M. Shaha and M. Pawar, "Transfer Learning for Image Classification," Proc. 2nd Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2018, no. Iceca, pp. 656–660, 2018, doi: 10.1109/ICECA.2018.8474802.

[9] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 971–987, 2002, doi: 10.1109/TPAMI.2002.1017623.

[10] H.-C. Lian and B.-L. Lu, "Multi-view Gender Classification Using Local Binary Patterns and Support Vector Machines," in International Journal of Neural Systems, vol. 17, no. 6, 2006, pp. 202–209.

[11] L. A. Alexandre, "Gender recognition: A multiscale decision fusion approach," Pattern Recognit. Lett., vol. 31, no. 11, pp. 1422–1427, 2010, doi: 10.1016/j.patrec.2010.02.010.

[12] 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 Trans. Inf. Forensics Secur., vol. 8, no. 3, pp. 488–499, 2013, doi: 10.1109/TIFS.2013.2242063.

[13] C. Shan, "Learning local binary patterns for gender classification on real-world face images," Pattern Recognit. Lett., vol. 33, no. 4, pp. 431–437, 2012, doi: 10.1016/j.patrec.2011.05.016.

[14] H. C. Shih, "Robust gender classification using a precise patch histogram," Pattern Recognit., vol. 46, no. 2, pp. 519–528, 2013, doi: 10.1016/j.patcog.2012.08.003.

[15] B. Patel, R. P. Maheshwari, and B. Raman, "Compass local binary patterns for gender recognition of facial photographs and sketches," Neurocomputing, vol. 218, pp. 203–215, 2016, doi: 10.1016/j.neucom.2016.08.055.

[16] B. Moghaddam and Ming-Hsuan Yang, "Learning gender with support faces," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 5, pp. 707–711, May 2002, doi: 10.1109/34.1000244.

[17] S. Baluja and H. A. Rowley, "Boosting Sex Identification Performance," Int. J. Comput. Vis., vol. 71, no. 1, pp. 111–119, Jan. 2007, doi: 10.1007/s11263-006-8910-9.

[18] A. Jain, J. Huang, and Shiaofen Fang, "Gender Identification Using Frontal Facial Images," in 2005 IEEE International Conference on Multimedia and Expo, 2005, vol. 2005, pp. 1082–1085, doi: 10.1109/ICME.2005.1521613.

[19] M. A. Berbar, "Three robust features extraction approaches for facial gender classification," Vis. Comput., vol. 30, no. 1, pp. 19–31, Jan. 2014, doi: 10.1007/s00371-013-0774-8.

[20] T. Jabid, M. H. Kabir, and O. Chae, "Gender classification using local directional pattern (LDP)," Proc. - Int. Conf. Pattern Recognit., pp. 2162–2165, 2010, doi: 10.1109/ICPR.2010.373.

[21] L. Lu and P. Shi, "A novel fusion-based method for expression-invariant gender classification," ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., pp. 1065–1068, 2009, doi: 10.1109/ICASSP.2009.4959771.

[22] L. Bui, D. Tran, X. Huang, and G. Chetty, "Face gender recognition based on 2D Principal Component Analysis and Support Vector Machine," Proc. - 2010 4th Int. Conf. Netw. Syst. Secur. NSS 2010, pp. 579–582, 2010, doi: 10.1109/NSS.2010.19.

[23] B. Li, X. C. Lian, and B. L. Lu, "Gender classification by combining clothing, hair and facial component classifiers," Neurocomputing, vol. 76, no. 1, pp. 18–27, 2012, doi: 10.1016/j.neucom.2011.01.028.

[24] J. Mansanet, A. Albiol, and R. Paredes, "Local Deep Neural Networks for gender recognition," Pattern Recognit. Lett., vol. 70, pp. 80–86, 2016, doi: 10.1016/j.patrec.2015.11.015.

[25] A. Dhomne, R. Kumar, and V. Bhan, "Gender Recognition Through Face Using Deep Learning," Procedia Comput. Sci., vol. 132, pp. 2–10, 2018, doi: 10.1016/j.procs.2018.05.053.

[26] C. J. Lin, Y. C. Li, and H. Y. Lin, "Using convolutional neural networks based on a Taguchi method for face gender recognition," Electron., vol. 9, no. 8, pp. 1–15, 2020, doi:

10.3390/electronics9081227.

[27] T. V. Janahiraman and P. Subramaniam, “Gender Classification Based on Asian Faces using Deep Learning,” in 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), Oct. 2019, vol. 57, no. 5, pp. 84–89, doi: 10.1109/ICSEngT.2019.8906399.

[28] E. Eidinger, R. Enbar, and T. Hassner, “Age and Gender Estimation of Unfiltered Faces,” IEEE Trans. Inf. Forensics Secur., vol. 9, no. 12, pp. 2170–2179, Dec. 2014, doi: 10.1109/TIFS.2014.2359646.

[29] P. Rodríguez, G. Cucurull, J. M. Gonfaus, F. X. Roca, and J. Gonzàlez, “Age and gender recognition in the wild with deep attention,” Pattern Recognit., vol. 72, pp. 563–571, Dec. 2017, doi: 10.1016/j.patcog.2017.06.028.

[30] K. Zhang et al., “Age Group and Gender Estimation in the Wild with Deep RoR Architecture,” IEEE Access, vol. 5, no. X, pp. 22492–22503, 2017, doi: 10.1109/ACCESS.2017.2761849.

[31] M. Duan, K. Li, C. Yang, and K. Li, “A hybrid deep learning CNN–ELM for age and gender classification,” Neurocomputing, vol. 275, pp. 448–461, 2018, doi: 10.1016/j.neucom.2017.08.062.

[32] M. Afifi and A. Abdelhamed, “AFIF4: Deep gender classification based on AdaBoost-based fusion of isolated facial features and foggy faces,” J. Vis. Commun. Image Represent., vol. 62, pp. 77–86, 2019, doi: 10.1016/j.jvcir.2019.05.001.

[33] K. Khan, M. Attique, I. Syed, and A. Gul, “Automatic Gender Classification through Face Segmentation,” Symmetry (Basel)., vol. 11, no. 6, p. 770, Jun. 2019, doi: 10.3390/sym11060770.

[34] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.

[35] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

[37] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

[38] J. Van De Wolfshaar, M. F. Karaaba, and M. A. Wiering, “Deep convolutional neural networks and support vector machines for gender recognition,” Proc. - 2015 IEEE Symp. Ser. Comput. Intell. SSCI 2015, no. October, pp. 188–195, 2015, doi: 10.1109/SSCI.2015.37.

[39] G. Özbulak, Y. Aytar, and H. K. Ekenel, “How transferable are CNN-based features for age and gender classification?,” Lect. Notes Informatics (LNI), Proc. - Ser. Gesellschaft fur Inform., vol. P-260, no. 113, pp. 4–9, 2016, doi: 10.1109/BIOSIG.2016.7736925.

