symmetry

MDPI

Article

Race Recognition Using Deep Convolutional Neural Networks

Thanh Vo [1], Trang Nguyen [2] and C. T. Le [3,*]

[1] Advanced Program in Computer Science, University of Science, VNU HCMC, Ho Chi Minh 700000, Vietnam; luckyluck379@gmail.com
[2] Faculty of Information Technology, Ho Chi Minh City Open University, Ho Chi Minh 700000, Vietnam; trangntp@grad.uit.edu.vn
[3] Faculty of Information Technology, Ho Chi Minh City University of Technology (HUTECH), Ho Chi Minh 700000, Vietnam
* Correspondence: ct.le@hutech.edu.vn

Received: 7 July 2018; Accepted: 30 October 2018; Published: 1 November 2018

Abstract: Race recognition (RR), which has many applications such as in surveillance systems, image/video understanding, analysis, etc., is a difficult problem to solve completely. To contribute towards solving that problem, this article investigates using a deep learning model. An efficient Race Recognition Framework (RRF) is proposed that includes information collector (IC), face detection and preprocessing (FD&P), and RR modules. For the RR module, this study proposes two independent models. The first model is RR using a deep convolutional neural network (CNN) (the RR-CNN model). The second model (the RR-VGG model) is a fine-tuning model for RR based on VGG, the famous trained model for object recognition. In order to examine the performance of our proposed framework, we perform an experiment on our dataset named VNFaces, composed specifically of images collected from Facebook pages of Vietnamese people, to compare the accuracy between RR-CNN and RR-VGG. The experimental results show that for the VNFaces dataset, the RR-VGG model with augmented input images yields the best accuracy at 88.87% while RR-CNN, an independent and lightweight model, yields 88.64% accuracy. The extension experiments conducted prove that our proposed models could be applied to other race dataset problems such as Japanese, Chinese, or Brazilian with over 90% accuracy; the fine-tuning RR-VGG model achieved the best accuracy and is recommended for most scenarios.

Keywords: race recognition; deep convolutional neural networks; social networks; surveillance system

1. Introduction

Nowadays, surveillance systems contribute vitally to public security. The development of artificial intelligence, especially artificial intelligence for computer vision [1], has made it easier to analyze the resulting videos [2,3]. Several studies have recently addressed the problem of event detection in video surveillance [4] which requires the ability to identify and localize specified spatiotemporal patterns. Another problem in surveillance video analysis, which is attracting much research interest, is the person re-identification problem [5]. Person re-identification describes the task of identifying a person across several images that have been taken using multiple cameras or a single camera. Re-identification is a vital function for surveillance systems as well as human–computer interaction systems in order to facilitate searching of identity from large amounts of videos and images. Likewise, this study addresses the problem of race recognition (RR) in surveillance videos. In several situations, identifying the race of a person may be helpful for surveillance systems to identify the appropriate emergency supporter. Moreover, another application of RR is to video classification/clustering. The race of people

in a video can be an important factor for video classification/clustering. To address this challenge, significant efforts have been made toward RR and categorization. Fu et al. [6] wrote a comprehensive review on the learning of race from the face using many state-of-the-art methods. From their analysis, the problem could be resolved by using two main approaches: single-model RR, which tries to extract both appearance features and local discriminative regions, and multi-model RR, which combines features from both 2D and 3D information, or the fusion of face and gait, etc.

In recent years, social networks have become popular with billions of users around the world, with millions of pieces of information shared daily. Many studies on the application of social networks have been analyzed recently. In 2016, Farnadi et al. [7] gave a detailed analysis of various state-of-the-art methods for personality recognition in many datasets from Facebook, Twitter, and YouTube. In this year, Nguyen et al. [8] used a Deep Neural Network (DNN) to meet two types of information needs of response organizations: (i) informative tweet identification and (ii) topical classes classification. They also provided a new learning algorithm using stochastic gradient descent to train DNNs in online learning during disaster situations. Recently, Carvalhoa et al. [9] presented a smart platform to efficiently collect, manage, mine, and visualize large Twitter corpora.

In recent times, deep learning [10], which tries to learn good representations from raw data automatically with multilayers stacked on each other, has attracted significant attention from researchers due to its various applications in computer vision [11–15], natural language processing [16–18], and speech processing [19]. Convolutional neural networks (CNNs), a type of deep learning model, have recently achieved many promising results in large-scale image and video recognition [20,21]. The VGG model [20], which was first introduced by Simonyan and Zisserman in 2015, achieved very good performance on ImageNet [22] and has been widely used in computer vision studies.

In recent years, many researchers have switched from RR of popular race groups such as African Americans, Caucasians, and Asians to that of sub-ethnic groups such as Koreans, Japanese, and Chinese [23–25]. Inspired by this idea, this paper presents a system that has the ability to detect race using deep learning in realistic environments using social network information for the Vietnamese sub-ethnic group. The main contributions of this work are as follows. (1) We introduce a race dataset of Vietnamese people collected from a social network and published for academic use. (2) We propose an efficient framework including three modules for information collection (IC), face detection and preprocessing (FD&P), and RR. (3) For the RR module, we propose two independent models: an RR model using a CNN (RR-CNN) and a fine-tuning model based on VGG (RR-VGG). Experimental results show that our proposed framework achieves promising results for RR in various race datasets including the Vietnamese sub-ethnic group and others such as Japanese, Chinese, and Brazilian. More specifically, the proposed framework with RR-VGG achieves the best accuracy in most of the scenarios.

The remainder of the paper is organized as follows: Section 2 presents a review of related works including convolutional neural networks and the VGG model. Section 3 proposes the RR framework. The experimental results are shown in Section 4, and Section 5 gives the conclusions of the paper.

2. Related Work

2.1. Deep CNNs in Computer Vision

In recent years, deep CNNs have been used extensively in computer vision especially due to their promising performance. For the problem of image classification, Zhang et al. [15] proposed a novel feature learning method for halftone image classification with very good performance. This method uses stacked sparse auto-encoders (SAE) to extract features of halftone images by using unsupervised learning, and then uses SoftMax regression to fine-tune the deep neural network using supervised learning to classify halftone images. Wei et al. [26] proposed a flexible deep CNN model, called Hypotheses-CNN-Pooling, for multilabel image classification. The input of this

framework is an arbitrary number of object segment hypotheses. Then, a shared CNN is linked to each hypothesis. Finally, the results from different hypotheses are summed with max pooling to generate the final multilabel predictions. In face-related applications, there are several problems such as face detection [27], face alignment [12], facial expression analysis [11], etc. In 2015, Li et al. [27] proposed a cascade model built on a CNN with very powerful discriminative capability while maintaining high performance to deal with the problem of changes in visual properties, such as those due to pose, expression, and lighting, in real-world face detection. In 2017, Park et al. [12] designed deep neural networks for face alignment using facial landmark features and recurrent regression. In addition, because a smile is one of the most common facial expressions in our daily life, Chen et al. [11] proposed an intelligent method for smile detection using CNNs. Facial expression analysis could be applied to other problems such as medical assessment, lie detection, human–computer interface, robotics, etc.

In video analysis, the problem of person re-identification—identification of people across images that have been taken using multiple cameras, or over time using a single camera—is an important one. In 2015, Ahmed et al. [5] proposed a method for parallelly learning features and a corresponding similarity metric for person re-identification. The authors also present a deep convolutional model with layers specially designed to address the problem of re-identification, and they achieve good results. Another problem in video analysis is visual target tracking, which has a wide range of applications such as vehicle navigation, augmented reality, video surveillance, etc. Pang et al. [13] proposed an approach to deal with visual target tracking tasks using a CNN with very good performance that is useful for real-time visual tracking. Human activity recognition is another problem in video analysis which has attracted a lot of attention in recent years. Ronao and Cho [14] proposed a CNN to perform efficient and effective human activity recognition using smartphone sensors by exploiting the inherent characteristics of activities and 1D time-series signals. This method achieves very good performance on several experimental datasets.

2.2. Transfer Learning

In practice, due to insufficiently sized datasets, modern deep CNNs rarely train an entire convolutional network from scratch. Instead, they usually use a network pretrained on a very large dataset (e.g., ImageNet), such as VGG, and then use it as an initialization or a fixed feature extractor for new tasks. The three major transfer learning scenarios are CNNs as fixed feature extractors, fine-tuning CNNs, and pretrained models. Among these, fine-tuning CNNs have been widely used with the various models such as VGG [28–30] and GoogLeNet [31,32].

VGG is a convolutional neural network model proposed by Simonyan and Zisserman [20] which achieves 92.7% accuracy in ImageNet [22], a dataset of over 14 million images belonging to 1000 classes. The trained VGG model has two different forms—VGG-16 and VGG-19—the structure and parameters of which are freely available online. In this study, VGG-16 will be used in the RR-VGG model presented in Section 3.3. Its macroarchitecture is shown in Figure 1. Many works have leveraged the structure of VGG to perform transfer learning in different problems. Paul et al. [33] applied the pretrained VGG model to the lung adenocarcinoma detection problem to extract useful features and classify images under various classifiers. In another work by Long et al. [34], many pretrained CNN models (VGG, AlexNet, and ResNet) were adopted into a fully convolutional network and fine-tuned for segmentation tasks. Hoo-Chang et al. [35] extensively studied the application of pretrained VGG models to computer-aided detection problems and achieved some promising results. The results detailed above make transferring a pretrained network such as VGG more applicable than other methods.

VGG-16 consists of 13 convolutional layers and three fully connected layers. In this model, larger filters (e.g., 5×5) are built from multiple smaller filters (e.g., 3×3) (Figure 2). Therefore, all convolutional layers have the same filter size of 3×3. In total, VGG-16 requires 138 M weights and 15.5 G multiply-and-accumulates to process one 224×224 input image [36]. The VGG model has been used in many studies [28–30] so far.

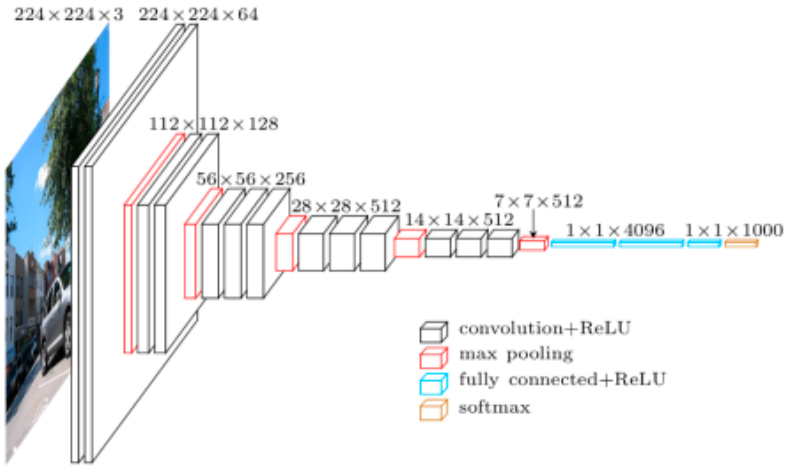

Figure 1. Macroarchitecture of VGG-16.

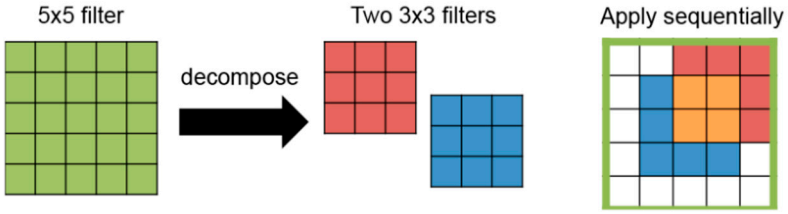

Figure 2. Decomposing larger filters into the smaller filters used in VGG-16 [36].

3. Materials and Methods

3.1. System Architecture

RR is a difficult problem and is almost impossible to solve completely with computer vision. However, with the integration of social network information and computer vision, this problem can be partially solved. First, we introduce the Vietnamese Faces (VNFaces) dataset for academic research which contains only two classes—"Vietnamese" and "other"—for classification. Secondly, we proposed a deep learning framework for the problem of RR. Although the model initially uses only the VNFaces dataset, later experiments have shown that it can be applied to other race datasets.

The proposed framework shown in Figure 3 consists of three main modules as follows. The first module, named the IC module, collects Vietnamese user information, including profile pictures and race, from the social network. Then, the information from the IC module is processed by the FD&P module. Firstly, this module detects faces in users' profile pictures. If faces are detected, the module will crop the face frames, resize them to 64 × 64, and automatically label these faces as "Vietnamese" or "other" based on their information. The resulting dataset of labeled face images is called VNFaces. The third module is RR, which consists of two independent models. A race detection model using a CNN (RR-CNN) is proposed in Section 3.2. In addition, Section 3.3 presents a fine-tuning approach for race detection based on the VGG-16 model [20], named RR-VGG. A comparison between RR-CNN and RR-VGG will be presented in Section 4 using many race datasets from different sub-ethnic groups.

After RR is trained on the VNFaces dataset, any image from different environments, e.g., other social networks or surveillance videos, can be put into the FD&P module to extract and preprocess the face. Next, the RR module can identify whether or not the person that appears in this image is Vietnamese.

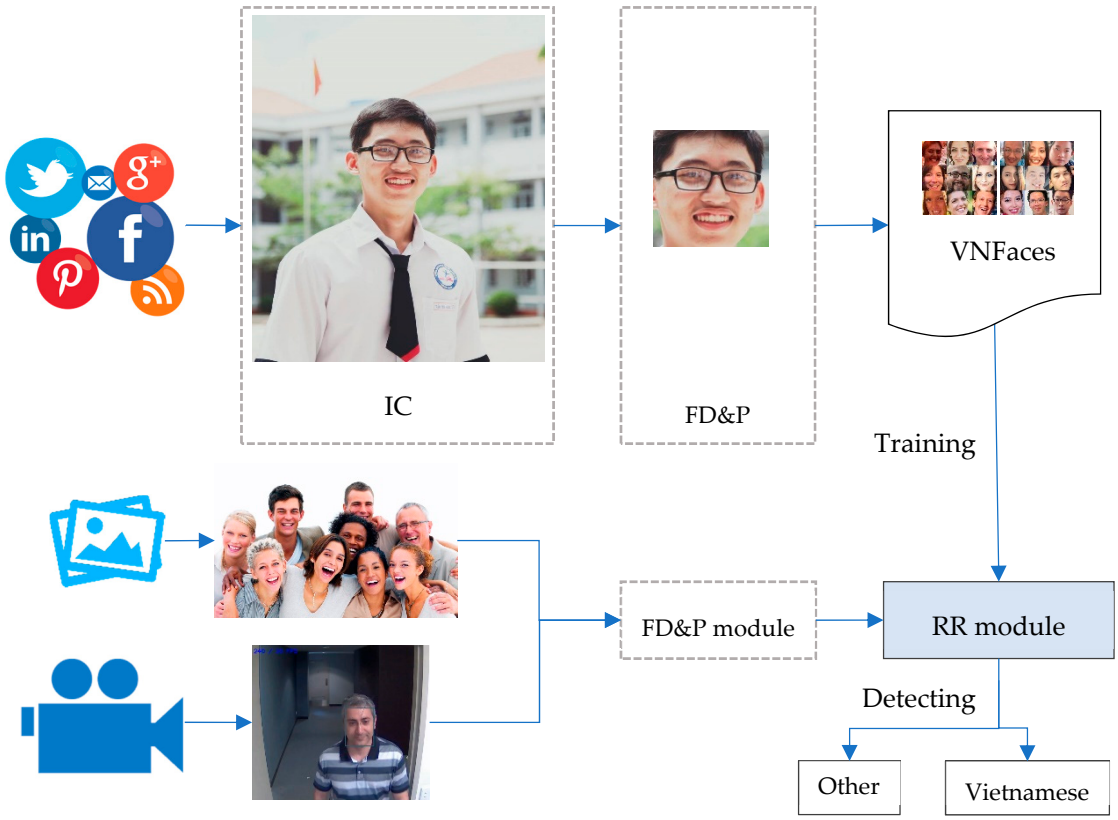

Figure 3. System architecture of the Race Recognition Framework (RRF).

3.2. Race Recognition Using CNN

The architecture of the RR-CNN model is shown in Figure 4. The input for this model is grayscale 64 × 64 images and there are two output classifications of one class, indicating Vietnamese or Other. There are four convolutional (C1–C4) layers and three max-pooling (P1–P3) layers, followed a dropout layer, and two fully connected layers between the input and the output (see Figure 4). The first convolutional layer (C1) filters the input image (64 × 64) with 32 learnable kernels of size 3 × 3 to give 32 matrices of size 62 × 62. The results of C1 are passed to the max-pooling layer P1, with 32 learnable kernels of size 2 × 2. The results of P1 will be 32 matrices of size 31 × 31; these are passed through the second convolutional layer C2, which has 32 learnable kernels of size 3 × 3, to get 32 matrices of size 29 × 29. Then, P2, with 32 learnable kernels of size 2 × 2, is used to process the previous results to get 32 matrices of size 14 × 14. Next, C3 and C4 are contiguous with 64 learnable kernels of size 3 × 3 for each layer. The results of C4 are 64 matrices of size 10 × 10, which are passed to P3, with 64 learnable kernels of size 2 × 2, to give 64 matrices of size 5 × 5. They are passed to the flatten layer to give 1600 values. Two fully connected layers, one with 1024 hidden units and the other with 512, follow. Finally, the output layer applies two classes, "Vietnamese" and "other", to the VNFaces dataset.

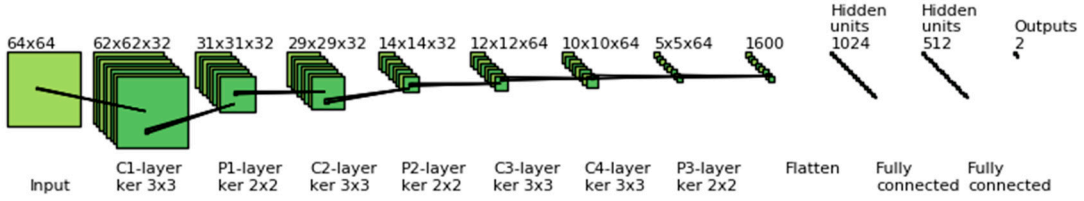

Figure 4. The RR-convolutional neural network (CNN) model.

The activation function of the output layer is SoftMax which produces a distribution over the two class labels in RR. However, to generalize for the problem of RR, N class labels will be used. The SoftMax function is shown below.

$$a_i = \frac{exp(z_i)}{\sum_{j=1}^{N} exp(z_j)}, \forall i = 1, 2, \ldots, N \tag{1}$$

where z is a vector of the inputs to the output layer and a_i indicates the probability of the ith class. Using this function, the total sum of the outputs is $\sum_{i=1}^{N} a_i = 1$. The loss value based on the cross-entropy of image x with parameter w is computed according to the following formula

$$J(w, x_i, y_i) = -\frac{1}{C} \sum_{j=1}^{C} \left(y_{ji} \log(a_{ji}) + (1 - y_{ji}) \log(a_{ji}) \right) \tag{2}$$

where y_{ji} and a_{ji} are the jth values in the y_i and a_i vectors. To minimize this value, the Adaptive Moment Estimation (Adam) optimizer [37], which computes adaptive learning rates for each parameter, was used. To store an exponentially decaying average of past squared gradients v_t, Adam also keeps an exponentially decaying average of past gradients m_t, like momentum. Let $g_t = \nabla_w J(w)$ be the gradient of the objective function with respect to the parameter w at time step t. The averages are computed using the following formulae

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{aligned} \tag{3}$$

where m_t and v_t are estimates of the first moment (the mean) and the second moment (the uncentered variance) of the gradients, respectively. m_t and v_t are initialized as zero vectors. Biases are counteracted by computing bias-corrected first and second moment estimates:

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1 - \beta_2^t}. \end{aligned} \tag{4}$$

Adam's update rule is determined by the following formula

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t. \tag{5}$$

The authors of a past paper [37] propose default values of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. Besides this, RR-CNN model uses a learning rate of $\eta = 0.001$ to minimize the cross-entropy loss value.

3.3. RR-VGG: A Fine-Tuning Approach

The input images for this model are rescaled to 64×64 to help reduce the complexity of training and testing. To generate more input images as well as to increase the accuracy of this model, we applied random flip, zoom, and shear rotate features for each face. Figure 5 shows the augmented images which were augmented by random flip (both horizontal and vertical), zoom (zoom range is 0.3), and shear rotate (sheer range is 30 degrees) with batch_size = 32.

To create the RR-VGG model, we first loaded all convolutional and max-pooling layers with the existing weights of VGG-16 (see Figure 1). Next, two fully connected layers, one with 1024 hidden units and one with 512, were added. Finally, the SoftMax function was used for classification into the two classes of the output layer. Therefore, the cross-entropy loss value was determined by Equation (2). To minimize this value, a Mini-batch Gradient Descent (GD) optimizer with batch size $n = 32$, learning rate $\eta = 0.0001$, momentum $\gamma = 0.9$, and learning rate decay $t = 0.000001$ was used. The time-based

learning rate schedule was used to anneal the learning rate over time. Therefore, the learning rate for the kth epoch was determined by

$$\eta_{t+1} = \frac{\eta_t}{(1 + k \times t)}. \tag{6}$$

Then, the Mini-batch GD's update rule was calculated by the following formula

$$w_{t+1} = w_t - \eta \nabla_w J(w, x_{i:i+n}; y_{i:i+n}). \tag{7}$$

Figure 5. Augmented images using random flip, zoom, and shear rotate features.

4. Results and Discussion

4.1. VNFaces Dataset and Cross-Validation Setting

To conduct the experiment, the IC module was used to collect user information on Facebook including profile picture and race from users of different ages and races. These profile pictures have varying pose, accessories, illumination, and imaging conditions. The second module, named FD&P, was used to automatically detect (using the Haar Casecade classifier [38]), crop, and label 6100 faces, including 2892 Vietnamese and 3208 Other, forming a collection called the VNFaces dataset (Available at https://goo.gl/M8eww3). Figure 6 shows a sample of Vietnamese (left) and Other (right) faces.

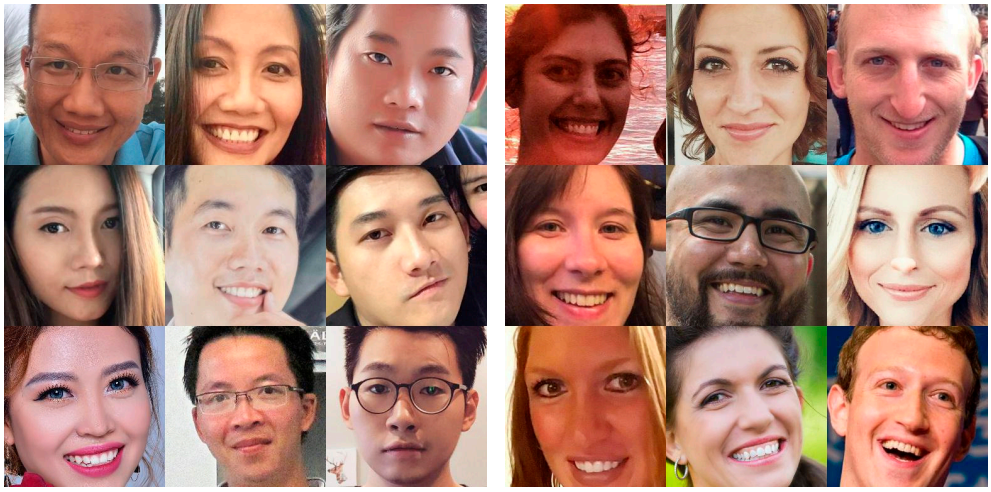

Figure 6. Examples of Vietnamese (**left**) and Other (**right**) faces.

To avoid the overfitting problem and to make the classifier generalizable to independent datasets, we employed 10-fold stratified cross-validation for the VNFaces dataset. We used nine folds for training and the last fold for testing. In the training procedure, the weight parameters of the CNN model were optimized automatically using the training set only. The testing set was hidden from the CNN model and only used after training was complete. Figure 7 shows the numbers of Vietnamese and Other face images in each fold utilized in this experiment.

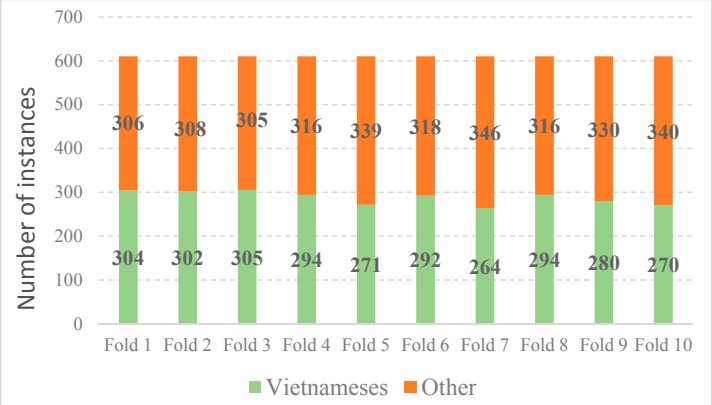

Figure 7. Numbers of Vietnamese and Other face images in each fold.

4.2. Race Recognition for the VNFaces Dataset

For the RR-VGG model, the input images were augmented by random flip, zoom, and shear rotate. To show the effectiveness of this step, we performed RR-VGG in two cases denoted by RR-VGG1 and RR-VGG2. RR-VGG2 used the random flip, zoom, and shear rotate features with batch_size = 32 for each image for training and validation sets, while RR-VGG1 used the original images for training and validation sets.

In this comparison of accuracy, RR-CNN, RR-VGG1, and RR-VGG2 models were used independently for predicting race with 10-fold cross-validation. For better analysis, we also compared with the original VGG (which we called RR-VGG0) where we used a well-trained VGG model without fine-tuning to predict racial identity. The experimental results in terms of accuracy are shown in Table 1. Firstly, it is clear to see that the original RR-VGG0 performs worse than the others with only 72.37% accuracy since it is not trained for face recognition. Secondly, RR-VGG1 lacks stability and has the worst results at 78.70% with an oscillation of 18.46%. In detail, at Fold 2, Fold 5, and Fold 10, RR-VGG1 has bad results of approximately 50% accuracy, while at the other folds, it has good values of accuracy. RR-CNN and RR-VGG2 achieve stable accuracy across the 10 folds with oscillations of 1.76% and 2.22%, respectively. The average accuracy of RR-VGG2 (88.87%) is a little bit better than that of RR-CNN (88.64%).

Table 1. Accuracy of different models for the VNFaces dataset with 10-fold validation.

Folds	Models			
	RR-CNN (%)	**RR-VGG0 (%)**	**RR-VGG1 (%)**	**RR-VGG2 (%)**
Fold 1	87.54	82.15	92.13	90.49
Fold 2	89.34	55.42	51.15	88.03
Fold 3	88.52	75.21	90.66	91.48
Fold 4	88.85	72.46	91.31	86.89
Fold 5	84.92	40.64	46.56	91.48
Fold 6	88.03	73.24	90.98	88.85
Fold 7	87.87	80.15	88.85	90.66
Fold 8	90.98	79.56	89.84	85.08
Fold 9	88.69	84.46	91.31	90.00
Fold 10	91.64	80.46	54.26	85.74
Average accuracy	88.64 ± 1.76	72.37 ± 3.24	78.70 ± 18.46	88.87 ± 2.22

In this experiment, we studied the effect of the number of epochs in the RR-CNN model. We built RR-CNN models with 10, 20, 30, 40, 50, and 60 epochs; the results are shown in Figure 8. The training and validation accuracies of this model are stable from 80% to 90% when the number of epochs increases from 10 to 60. In addition, training losses of this model are also stable. However, validation losses fluctuate upward when the number of epochs increases from 10 to 60. This means that overfitting occurs with a large number of epochs. Therefore, we suggest that only 10 to 20 epochs should be used to train RR-CNN.

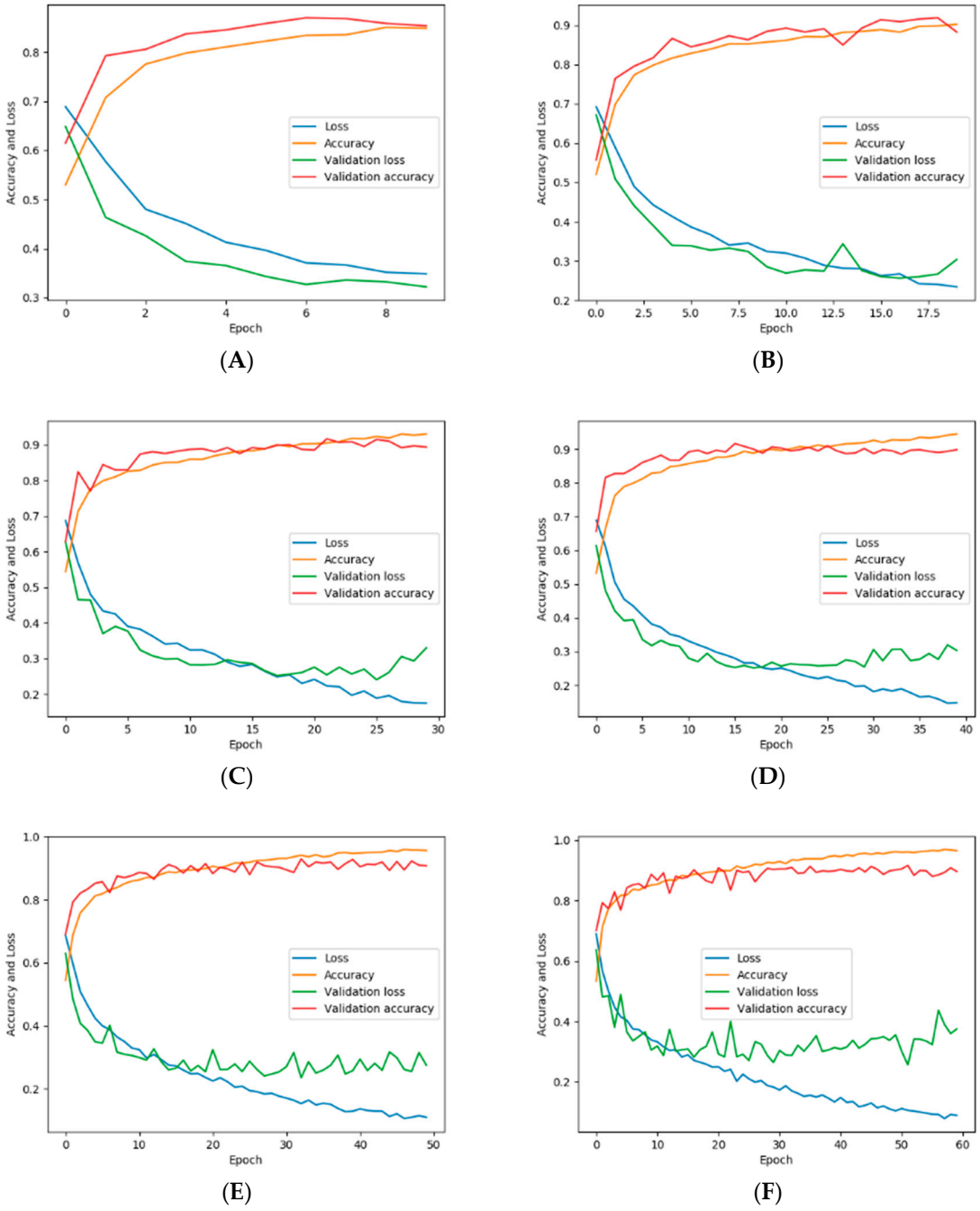

Figure 8. Performance of RR-CNN with 10 (**A**), 20 (**B**), 30 (**C**), 40 (**D**), 50 (**E**), and 60 (**F**) epochs.

Next, we compared the computation time for each fold including training and testing time among RR-CNN, RR-VGG1, and RR-VGG2. The results (Figure 9) show that RR-CNN requires 91.3 s while RR-VGG1 and RR-VGG2 take a much longer at 1623.1 s and 1648.9 s, respectively.

This is obvious because RR-VGG (including RR-VGG1 and RR-VGG2) has more layers than RR-CNN. Hence, RR-VGG's structure is much more complex than that of RR-CNN. In short, RR-CNN is more efficient than RR-VGG (including RR-VGG1 and RR-VGG2) in terms of the computation time.

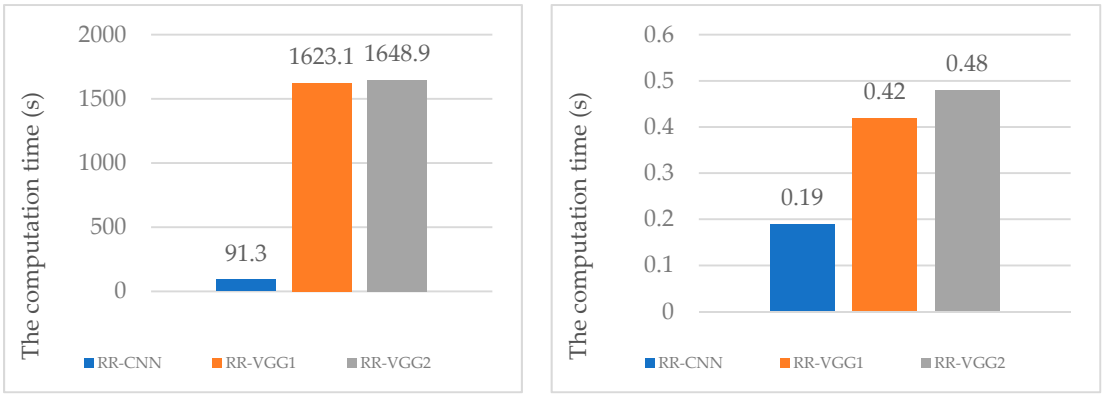

Figure 9. Computation time in (**a**) the training process and (**b**) the testing process of experimental models for the VNFaces dataset.

Table 2 compares the total number of trainable parameters between the three models RR-CNN, RR-VGG1, and RR-VGG2. As we can see in the table, RR-CNN has much fewer parameters in comparison with the two others, which makes it more lightweight and efficient in the training dataset.

Table 2. Number of trainable parameters between RR-CNN, RR-VGG1, and RR-VGG2 models.

Model	Number of Trainable Parameters
RR-CNN	2,230,242
RR-VGG1	17,338,690
RR-VGG2	17,338,690

4.3. Race Recognition: Extension Experiments

In the extension experiments, we aimed to show that our proposed models not only apply for the VNFaces dataset but could also be applied for other race datasets. We conducted two experiments as follows.

In the first experiment, we used the proposed models to perform classification of Japanese, Chinese, and Brazilian datasets; the Japanese Female Facial Expression (JAFFE) [39], Chinese University of Hong Kong Face Sketch (CUFS) [40], and Brazilian face database captured at the Artificial Intelligence Laboratory of FEI (FEI) datasets [41], respectively, were used as the image sources. The JAFFE dataset includes 213 images of seven facial expressions of 10 Japanese female models, as exemplified in Figure 10. The CUFS dataset has 188 images of 188 Hong Kong students, and FEI contains 14 images for each of the 200 individuals, 2800 images in total, with examples from the datasets shown in Figures 11 and 12, respectively. We conducted experiments classifying each race out of the combined dataset produced by adding images of 3208 other people (including Asian, African, and Caucasian) to each dataset as described in Section 4.1. We used 80% of people in each race for training and the remaining 20% for testing, as described in Table 3. These experiment settings make sure that the splitting process is a subject-independent partition where people in the training and testing sets are completely different. The results are reported in Table 4. It is shown that the proposed models both work well in all other race datasets tested. This has proven that our defined CNN model and fine-tuning VGG models could be used in RR problems, with fine-tuning VGG achieving the best accuracy.

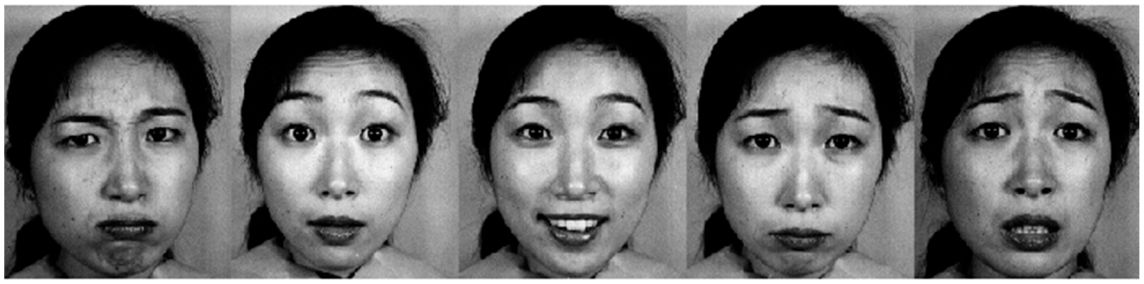

Figure 10. Samples from the JAFFE dataset.

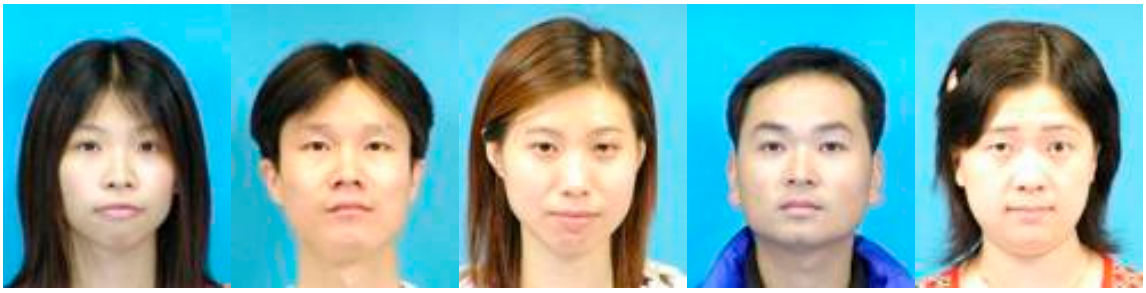

Figure 11. Samples from the Chinese University of Hong Kong Face Sketch (CUFS) dataset.

Figure 12. Samples from the FEI dataset.

Table 3. Number of images in the training and testing datasets of the three combined datasets.

Dataset	Train	Test
JAFFE + Others	8 Japanese + 2566 others	2 Japanese + 642 others
CUFS + Others	150 Chinese + 2566 others	38 Chinese + 642 others
FEI + Others	160 Brazilian + 2566 others	40 Brazilian + 642 others

Table 4. Accuracy of the three proposed methods used on the different race datasets.

Dataset	Accuracy (%)		
	RR-CNN	RR-VGG1	RR-VGG2
JAFFE + Others	95.81	99.42	98.72
CUFS + Others	99.94	100.0	100.0
FEI + Others	90.73	95.76	97.25

In the second experiment, we tested the performance of classifying Asian people and Others when we mixed our VNFaces dataset with two other datasets: the JAFFE dataset and the CUFS dataset. The combined dataset has a total 3090 Asian people and 3208 Other people. It was split into training and testing sets with an 80% proportion for training and a 20% proportion for testing, as shown in Figure 13.

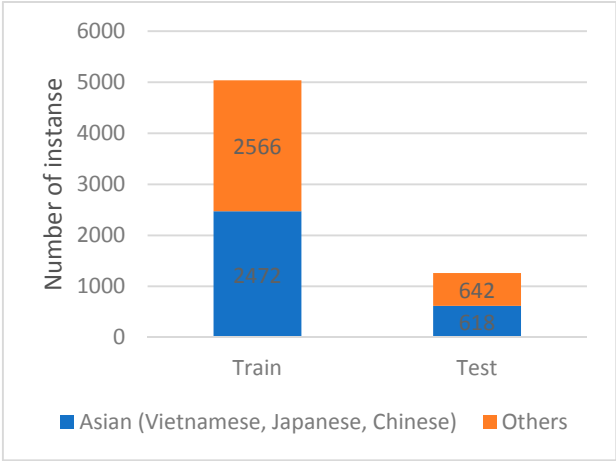

Figure 13. Number of Asian people (including Vietnamese, Japanese, and Chinese) and Others in the training and testing datasets.

The results of the evaluation on the testing set are provided in Table 5. As we can see, all three proposed models performed well with high classification accuracy of over 75%. In detail, RR-CNN achieved 76.51%, while RR-VGG1 and RR-VGG2 achieved 86.55% and 87.24%, respectively. This implies that the fine-tuning VGG model with deeper convolutional layers could be used when dealing with many complicated datasets containing various races.

Table 5. Accuracy of Asian race classification by the three proposed models of a combined dataset of Vietnamese, Japanese, and Chinese with Others.

Model	Accuracy (%)
RR-CNN	76.51
RR-VGG1	86.55
RR-VGG2	87.24

Last, as race classification can be challenging when considering races of similar appearances, so that in the third experiment, we conducted the classifying people in Asian ethnicity. In details, we combined the dataset of Vietnamese, Japanese, and Chinese together named VCJ dataset with 2892 Vietnamese, 10 Japanese, and 188 Chinese. We performed experiment where we classified three races out the dataset. The output layer in RR-CNN, RR-VGG1, and RR-VGG2 models has been transformed to deal with three classes: Vietnamese, Japanese, and Chinese respectively. The experiment was set up identically to the previous with 80% of people for training and 20% of last for testing, as given in Table 6 and the results are given in Table 7. Since the number of Vietnamese is much larger than Japanese and Chinese which makes it become unbalance problem, the default metrics using accuracy is less meaningful. In order to overcome such problem, the receiver operating characteristic (ROC) curve [42] is widely used. However, for multiclass algorithms, we need to use a multiclass AUC method. In this approach, a separate AUC for each class is calculated, such that the AUC of class C_i is calculated by considering all the samples of C_i as positives and the samples of all other classes as negatives. The average AUS is calculated as the mean of AUC of three classes. As we can see in the table, all three models have achieved over 80% in AUC. The best average AUC is 99.08% using RR-VGG1 while RR-VGG2 achieves 94.48% and RR-CNN accuracy is 90.19%.

Table 6. Number of images in the training and testing datasets of the VCJ dataset.

Train	Test
2313 Vietnamese + 8 Japanese + 150 Chinese	579 Vietnamese + 2 Japanese + 38 Chinese

Table 7. AUCs (%) of classifying Vietnamese, Japanese, and Chinese out the VCJ dataset.

Class	RR-CNN	RR-VGG1	RR-VGG2
Vietnamese	89.24	99.13	94.41
Japanese	100.00	100.00	100.00
Chinese	81.35	98.13	89.04
Average	90.19	**99.08**	94.48

5. Conclusions

This study proposed an efficient RR Framework consisting of three modules: an information collector, face detection and preprocessing, and RR. For the RR module, this study proposes two independent models: RR-CNN and RR-VGG. To evaluate the proposed framework, we conducted experiments on our Vietnamese dataset collected from Facebook, named VNFaces, to compare the accuracy between the RR-CNN and RR-VGG models. The experimental results show that for the VNFaces dataset, the RR-VGG model with augmented input images yields the best accuracy at 88.87%, while RR-CNN achieves 88.64%. In other examined cases, our proposed models also achieved high accuracy in the classification of other race datasets such as Japanese, Chinese, and Brazilian. Even in the case of classifying people with similar appearances, our models could perform well with overall results are over 80%. In most of the scenarios, the fine-tuning RR-VGG achieved the best accuracy due to its number of deep layers; this suggests that it could be successfully applied to various RR problems.

In future work, several related issues will be studied. Firstly, a project that collects a race face dataset based on social network data for RR will be opened for contributions from around the world. Secondly, several preprocessing techniques and deep models to improve the accuracy of classification will be studied.

Author Contributions: T.V. proposed the topic and implemented the framework. T.V. wrote the paper. C.T.L. and T.N. improved the paper.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Baidyk, T.; Kussul, E.M.; Monterrosas, Z.C.; Gallardo, A.J.I.; Serrato, K.L.R.; Conde, C.; Serrano, A.; Diego, I.M.; Cabello, E. Face recognition using a permutation coding neural classifier. *Neural Comput. Appl.* **2016**, *27*, 973–987. [CrossRef]
2. Kardas, K.; Cicekli, N.K. SVAS: Surveillance Video Analysis System. *Expert Syst. Appl.* **2017**, *89*, 343–361. [CrossRef]
3. Zhang, Q.; Chen, X.; Zhan, Q.; Yang, T.; Xia, S. Respiration-based emotion recognition with deep learning. *Comput. Ind.* **2017**, *92–93*, 84–90. [CrossRef]
4. Cosar, S.; Donatiello, G.; Bogorny, V.; Gárate, C.; Alvares, L.O.; Brémond, F. Toward abnormal trajectory and event detection in video surveillance. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 683–695. [CrossRef]
5. Ahmed, E.; Jones, M.J.; Marks, T.K. An improved deep learning architecture for person re-identification. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3908–3916.
6. Fu, S.; He, H.; Hou, Z.-G. Learning Race from face: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2483–2509. [CrossRef] [PubMed]
7. Farnadi, G.; Sitaraman, G.; Sushmita, S.; Celli, F.; Kosinski, M.; Stillwell, D.; Davalos, S.; Moens, M.F.; Cock, M.D. Computational personality recognition in social media. *User Model. User-Adapt. Interact.* **2016**, *26*, 109–142. [CrossRef]
8. Nguyen, D.T.; Joty, S.R.; Imran, M.; Sajjad, H.; Mitra, P. Applications of online deep learning for crisis response using social media information. *arXiv* **2016**. Available online: https://arxiv.org/abs/1610.01030 (accessed on 1 July 2018).

9. Carvalhoa, J.P.; Rosa, H.; Brogueirac, G.; Batista, F. MISNIS: An intelligent platform for twitter topic mining. *Expert Syst. Appl.* **2017**, *89*, 374–388. [CrossRef]
10. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
11. Chen, J.; Ou, Q.; Chi, Z.; Fu, H. Smile detection in the wild with deep convolutional neural networks. *Mach. Vis. Appl.* **2017**, *28*, 173–183. [CrossRef]
12. Parka, B.H.; Oha, S.Y.; Kim, I.J. Face alignment using a deep neural network with local feature learning and recurrent regression. *Expert Syst. Appl.* **2017**, *89*, 66–80. [CrossRef]
13. Pang, S.; Coz, J.J.; Yu, Z.; Luaces, O.; Díez, J. Deep learning to frame objects for visual target tracking. *Eng. Appl. Artif. Intell.* **2017**, *65*, 406–420. [CrossRef]
14. Ronao, C.A.; Cho, S.B. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Syst. Appl.* **2016**, *59*, 235–244. [CrossRef]
15. Zhang, Y.; Zhang, E.; Chen, W. Deep neural network for halftone image classification based on sparse auto-encoder. *Eng. Appl. Artif. Intell.* **2016**, *50*, 245–255. [CrossRef]
16. Majumder, N.; Poria, S.; Gelbukh, A.F.; Cambria, E. Deep learning-based document modeling for personality detection from text. *IEEE Intell. Syst.* **2017**, *32*, 74–79. [CrossRef]
17. Poria, S.; Cambria, E.; Gelbukh, A.F. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowl. Based Syst.* **2016**, *108*, 42–49. [CrossRef]
18. Yu, X.; Yu, H.; Tian, X.Y.; Yu, G.; Li, X.M.; Zhang, X.; Wang, J. Recognition of college students from Weibo with deep neural networks. *Int. J. Mach. Learn. Cybern.* **2017**, *8*, 1447–1455. [CrossRef]
19. Qawaqneh, Z.; Mallouh, A.A.; Barkana, B.D. Deep neural network framework and transformed MFCCs for speaker's age and gender classification. *Knowl. Based Syst.* **2017**, *115*, 5–14. [CrossRef]
20. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2015**, arXiv:1409.1556. Available online: https://arxiv.org/abs/1409.1556 (accessed on 1 July 2018).
21. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1106–1114.
22. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
23. Roh, R.C.; Lee, S.W. Performance evaluation of face recognition algorithms on Korean face database. *Int. J. Pattern Recognit. Artif. Intell.* **2007**, *21*, 1017–1033. [CrossRef]
24. Bastanfard, A.; Nik, M.A.; Dehshibi, M.M. Iranian face database with age, pose and expression. In Proceedings of the 2007 International Conference on Machine Vision, Islamabad, Pakistan, 28–29 December 2007; pp. 50–58.
25. Gao, W.; Cao, B.; Shan, S.G.; Chen, X.L.; Zhou, D.L.; Zhang, X.H.; Zhao, D.B. The CAS-PEAL large-scale Chinese face database and baseline evaluations. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2008**, *38*, 149–161.
26. Wei, Y.; Xia, W.; Lin, M.; Huang, J.; Ni, B.; Dong, J.; Zhao, Y.; Yan, S. HCP: A flexible CNN framework for multi-label image classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 1901–1907. [CrossRef] [PubMed]
27. Li, H.; Lin, Z.; Shen, X.; Brandt, J.; Hua, G. A convolutional neural network cascade for face detection. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5325–5334.
28. He, K.; Wang, Y.; Hopcroft, J.E. A powerful generative model using random weights for the deep image representation. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–11 December 2016; pp. 631–639.
29. Yang, W.; Ouyang, W.; Li, H.; Wang, X. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3073–3082.
30. Li, X.; Zhao, L.; Wei, L.; Yang, M.H.; Wu, F.; Zhuang, Y.; Ling, H.; Wang, J. DeepSaliency: Multi-task deep neural network model for salient object detection. *IEEE Trans. Image Process.* **2016**, *25*, 3919–3930. [CrossRef] [PubMed]

31. Carreira, J.; Agrawal, P.; Fragkiadaki, K.; Malik, J. Human pose estimation with iterative error feedback. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4733–4742.
32. Karaoglu, S.; Tao, R.; Gevers, T.; Smeulders, A. Words matter: Scene text for image classification and retrieval. *IEEE Trans. Multimed.* **2017**, *19*, 1063–1076. [CrossRef]
33. Paul, R.; Hawkins, S.H.; Balagurunathan, Y.; Schabath, M.B.; Gillies, R.J.; Hall, L.O.; Goldgof, D.B. Deep feature transfer learning in combination with traditional features predicts survival among patients with lung adenocarcinoma. *Tomogr. J. Imaging Res.* **2016**, *2*, 388–395.
34. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
35. Hoo-Chang, S.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* **2016**, *35*, 1285–1298.
36. Sze, V.; Chen, Y.H.; Yang, T.J.; Emer, J.S. Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *arXiv* **2017**. Available online: https://arxiv.org/abs/1703.09039 (accessed on 1 July 2018). [CrossRef]
37. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**. Available online: https://arxiv.org/abs/1412.6980 (accessed on 1 July 2018).
38. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Kauai, HI, USA, 8–14 December 2001.
39. Lyons, M.; Akamatsu, S.; Kamachi, M.; Gyoba, J. Coding facial expressions with Gabor wavelets. In Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 14–16 April 1998; pp. 200–205.
40. Wang, X.; Tang, X. Face photo-sketch synthesis and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 1955–1967. [CrossRef] [PubMed]
41. Thomaz, C.E.; Giraldi, G.A. A new ranking method for principal components analysis and its application to face image analysis. *Image Vis. Comput.* **2010**, *28*, 902–913. [CrossRef]
42. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284.