I.J. Modern Education and Computer Science, **2020, 6, 16-28**
Published Online December 2020 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2020.06.02

A Lightweight Face Recognition Model Using Convolutional Neural Network for Monitoring Students in E-Learning

Duong Thang Long
Hanoi Open University, Viet Nam
Email: duongthanglong@hou.edu.vn

Received: 17 August 2020; Accepted: 25 October 2020; Published: 08 December 2020

Abstract: Using convolution neural network (CNN) for face recognition is being widely research with a promising significant in applications and it is interested by many authors. Moreover, the CNN model has brought successful applications in practice such as detection and identification face of people on Facebook users' photos application, they use DeepFace model. There are many articles which proposed CNN models for face recognition with using some modifications of popular models of large architectures such as VGG, ResNet, OpenFace or FaceNet. However, these models are large complexity for some applications in reality with limitations of computing resources. This paper proposes a design of CNN model with moderate complexity but still ensures the quality and efficiency of face recognition. We run experiments for evaluating the model on some popular datasets, the experiment shows effective results and indicates that the proposed model can be practically used.

Index Terms: Convolutional neural networks, face recognition, online student monitoring

1. Introduction

Computer vision is a very exciting field of research today, with methods based on huge powerful computing systems with valuable practical application problems. Methods underlying physical characteristics or behavior of persons to identify people is strongly researched and applied on identification systems. In particular, facial recognition has been widely research and application. Faces of each person in the world have their own unique and distinctive features. So, it can be considered as one's own identity. Facial recognition uniqueness therefore should be used for identity authentication and control people in various applications such as online learning systems (or e-learning), vehicle driver authentication, etc. [3, 6].

In recent years, with development of e-learning, more and more people have chosen to learn and acquire knowledge using online learning systems (LMS). In E-learning, people can learn many things what they need at anytime and anywhere. It is flexible and can be expanded, using fast learning, inexpensive learning methods and proven to be more effective than traditional education. Therefore, e-learning is becoming more and more popular. However, assessing quality of learning activities is definitely essential in educational process. If there is any fraud or cheating in learning that is not acceptable, it will greatly affect learning outcomes of learners and quality of educational systems. Therefore, educational systems and a specific of e-learning systems need to provide ability of identifying and monitoring learners' activities [24].

Clearly, it is difficult to have a perfect biometric system suitable for all applications. Several studies focus on improving security in online learning using biometric authentic systems such as keystroke motivation in [7], but some of them have to face continuously authentications of learners. Researchers are currently looking for better ways to determine biometrics that will help identification and monitoring during online learning and examinations. Face recognition (FR) systems are very friendly to humans because they do not require contact and no additional hardware is needed (provided that most computers or user devices now have built-in cameras). More importantly, FR systems can be used for continuous authentication of learners over the entire period of learning or examination.

There are many proposed FR systems with using large CNN models [2-16, 18, 21, 23], some of them are designed for real time monitoring vehicle drivers [6], or applications in radiology [18]. They are rarely designed for e-learning applications, a method for monitoring learners in taking online examination in [7], but it just used images processing as PCA for facial feature extracting. For utilizing advantages of CNN models, and instead of using very complexity FR models such as VGG, ResNet, OpenFace or FaceNet [17] directly, we design a lightweight face recognition model to fit limited computation systems, then we proposed a system using this model for continuously monitoring online learning students with high effective and suitable for practical applications. We also integrate this authentication system with a current online learning management system (LMS) by using some cases of connections techniques.

This paper is structed as 5 parts. After this Introduction part, summarization relevant research is in Related work part. Methodology part introduces our proposed CNN model with highlighting some advantages and limitations, then we design a system for applying the model to monitor online learning students with integrated LMS. Part 4 presents experiments of proposed model and analysis results. Part 5 is conclusion and some further research directions.

2. Related Work

In recent years, strong developments of deep learning technology with convolutional neural network (CNN) has been successfully applied in many practical problems [17, 18]. CNN generally consists of three types of neuron layers (illustrated in Fig.1): convolution neuron layer, generalized neuron layer (pooling layer) and fully connection neuron layer. The first two types of neuron layers (convolution and pooling) perform the role of extracting characteristics of face image, while the third layer (fully connection) performs the role of mapping extracted features into outputs to identify people. The convolution layer plays an important role in CNN, including a stack of convolutional operations, which is a specialized linear operation. The pooling layer acts to reduce dimension of extracted features space (also called subsampling), this is for speeding up of recognition process. In learning phase of CNN, network parameters (trainable parameters) will be updated for getting better models, the trainable parameters include link weights of convolutional neurons and fully connected neurons. A typical learning algorithm of this kind neural network is backward propagation of errors, with a goal of minimizing errors of models. In addition, this model kind also has parameters which need to be set before applying such as number and size of kernels in convolutions, slide of convolutional operations, activation functions, calculation method of pooling layer and other parameters. Detail of all parameters is mentioned in [18].

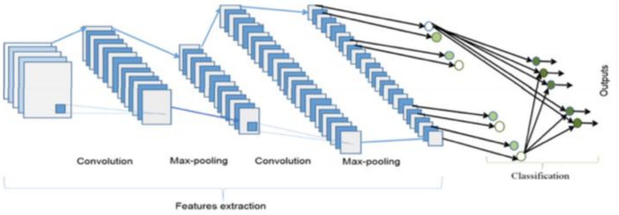


Fig.1. An example of a CNN architecture [19]

Authors in [9] analyze effectiveness of CNN compared to classical recognition methods including principal component analysis (PCA), local binary pattern histogram (LBPH) and k-nearest neighbor (KNN). They take experiments on ORL (A&T) dataset and their results show that the LBPH model is better than PCA and KNN, but CNN model get the best accuracy of recognition (98.3% compared to other methods, they all less than 90%). It so may confirm that CNN model is superior to other methods in face recognitions. Some innovative CNN architectures for face recognition are analyzed and evaluated in [12]. They proposed an architecture contains 22 neuron layers with 140 million trainable parameters. We can also use the “triplet loss” technique in training models for getting better accuracy of classifications, it is mentioned in [2, 8, 11, 12, 13].

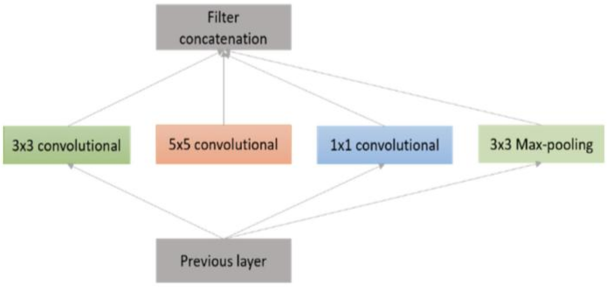


Fig.2. Naive version of interception module [19]

Another architecture is based on GoogleNet's Interception model (Fig.2), which includes some versions of different network input sizes to reduce space of trainable parameters. These architectures are applied to different problems, the larger CNN architecture, the higher accuracy of recognition and more suitable for applications on huge problems. However, smaller CNN architectures will be suitable kinds of applications on portable mobile devices but they still ensure acceptable results by eliminating some less important parts of models. In order to increase efficiency, authors in [11] propose a "very deep" CNN architecture consisting of 11 blocks with 37 neuron layers, 8 first blocks play the role of extracting characteristics and 3 last blocks show classification functions for recognition. This CNN architecture is experimented on very large datasets (LFW and YTF with thousands of identifiers and millions of face images), it results better than those other CNN models (98.95% on LFW and 97.3% on YTF). Authors in [5] have proposed a CNN for facial recognition with some improvements based on VGG's architecture (from Visual Geometry Group - University of Oxford). They use concatenated ReLu (Fig.3) for selecting positive parts and other ReLu for selecting negative parts of activation. It is called CReLu module. This makes double point of nonlinearity of activation functions in CNN model and it has been showed for better quality results. Based on this proposed model, the authors built a real-time face recognition system with “very deep” model of convolutional neural network. They took experiments and analysis with better results compared to the original model.

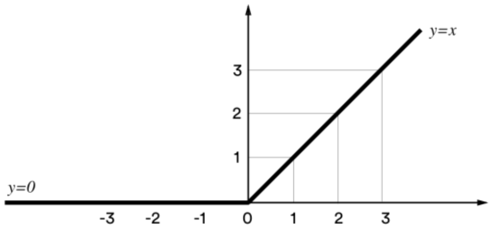


Fig.3. ReLu activation of neurons

In CNN based face recognition models, a pair of faces to compare are independently fed into the CNN model for feature extraction. For both faces, the same kernels of convolutional neurons are applied and therefore representation of a face stays fixed regardless of whom it is compared with. However, for us humans, one generally focuses on varied characteristics of a face when comparing it with others. Therefore, authors in [18] have proposed a new CNN architecture called contrastive convolution. It particularly focuses on differences between two faces for comparison, i.e. contrastive characteristics between them. Their experiments show that this proposed model greatly improves compared with conventional CNN and promises superiority in applications. Contrastive convolution has advantages of automatically generating convolution results based on pairs of faces considered. This contrastive convolution can be incorporated into any type of CNN architecture.

Some useful applications have been developed from biometric methods with face recognition problems. For example, authors in [6] built a CNN-based face recognition system for continuous and real-time authentication of drivers in preventing car theft and process monitoring. In educational field, authors in [7] provided a solution for online examination systems using face recognition to authenticate learners when taking online examinations. These methods are useful, the designed systems are early extended to the real world. They provide early warning for users if suspicious behavior has been noticed by the system. However, the CNN model in [6] is just directly used VGG architecture [11] for extracting features of face. This model is very deep with large number of neural layers, so it is just suitable for large systems in applications. The model in [7] only used original image processing methods for detecting faces and feature points on faces, it is very sensitive with image changing in color or pose. In this paper, we design a CNN-based face recognition model in lightweight version to be suitable for moderate system, then propose an LMS-integrated application for monitoring online learning of learners. Thereby, it is possible to measure the attendance rate in online learning systems and use these results to support evaluation of learning results of learners.

3. Proposed Face Recognition System

In this section, we design a new face recognition model by using CNN in lightweight version. For the lightweight model, we can apply it to widely applications of limited computation in reality such as an integrated system with online learning system to continuously capture images of learners and identify them participating in the system to contribute for assessing learning results of learners, eliminate cheating in online learning and helping to improve quality of educational systems. Our face recognition model is divided into 3 main steps (Fig.4), a preprocessing step to detect a face region from input image, enhance quality if necessary; second step extracts facial features and third step identifies people of the input image based on selected features. Both second and third steps are designed to be integrated in a CNN model.

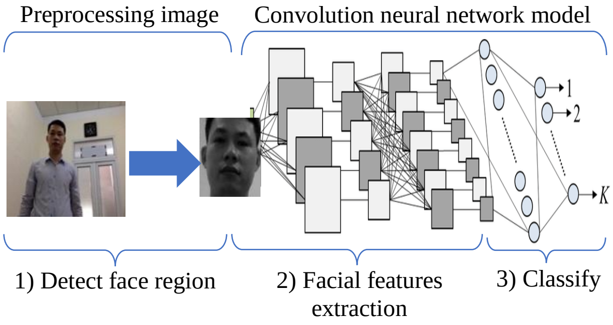

Preprocessing image
Convolution neural network model
1
2
K
1) Detect face region
2) Facial features extraction
3) Classify

Fig.4. Diagram of our face recognition model

A. Input image preprocessing

In this section, we apply a number of pre-processing methods on input images including detection and cropping to get the image area containing the face, then improving image quality. In practical applications, input images are usually captured from a camera, they include background with any object inside. So, we have to perform a face detection method to determine an image area containing the face and then cut off to remove background of the face. To do this, we use a well-known CNN based model called MTCNN as in [4]. For overfit avoiding in training, we augment face images by using some image processing such as noise addition, rotation, shearing and shifting, making lighter or darker images. With an input image a, we so get list of pre-processed face images as,

$$\{\mathfrak{I}^{\alpha}(f^{D}(a), p^{\alpha})\},$$

where, f^D is a face detector such as MTCNN, p^α is parameters for image augmenting operation with kind of augmentation $\alpha = \{noise,\ rotation,\ shearing,\ shifting,\ \dots\}$, $\mathfrak{I}^\alpha$ denotes an transformation of images in augmenting operation. For instance, by applying Gaussian-distributed additive noise to a face image with variance of random distributions in [0.01, 0.02, 0.03, 0.04, 0.05] we get augmented face images in Fig.5.

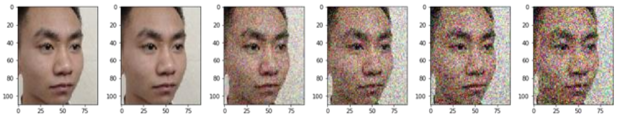


Fig.5. Augmented faces by Gaussian noise

Another augmentation is rotation, it may rotate left or right with a limited angle. The following images are rotated results by angles of [-15, -10, -5, 0, 5, 10, 15] in degree.

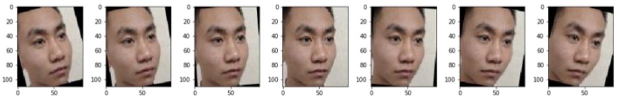


Fig.6. Augmented faces by rotation

Therefore, in training phase of CNN model, we can make large number of face images from a small dataset by using these kinds of augmentation. However, parameters of each augmenting operation should be limited in a propriate range in order to training convergence.

B. Design CNN model

The CNN model is designed with two main functions: features extraction of face and object classification based on extracted features. Number of layers and magnitude (number of neurons) of each layer affect to quality and complexity in computation. Authors often adjust these two factors according to application problems for achieving expected quality and acceptable computational complexity at the same time. So, we design this model with moderate number of layers for fitting to our computation systems.

Each neuron layer in the CNN model takes a multidimensional array of numbers as inputs and generates another multidimensional array at the output which becomes input of next layer. For recognizing faces, input size of the first layer is input image size, output size of the last layer is number of people in the problem. We use all three types of neuron layers to design architecture of this CNN model including 5 convolution layers (CONV), 4 pooling layers (POOL) and 2 fully-connected layers to classify inputs. Each CONV layer is connected following by a POOL layer, we use ReLu activation function (Rectified Linear Unit) for CONV neurons. The ReLu is linear activation function for fast training of model, its default is $f(x) = \max(x, 0)$. This is ensuring that non-negative input will be input of next layers.

According to principles of stacking neuron layers and down sampling in outputs, CNN models perform extracting more and more abstract and complex features by convolutional neuron layers, so it is invariant to transformations [9]. Furthermore, to overcome overfit in training, this model uses Dropout technique following each POOL layer. The Dropout technique was recently introduced and used mainly, it randomly selects activation functions of neuron outputs with a proportional number of neurons and sets them to 0 (i.e. outputs of such neurons are zero) during CNN training, so this model becomes less sensitive to specific weights in the network. The proportional value of Dropout in this model is set by heuristic method and based on practicing. The architecture of CNN model is divided into 8 blocks as in Fig.7.

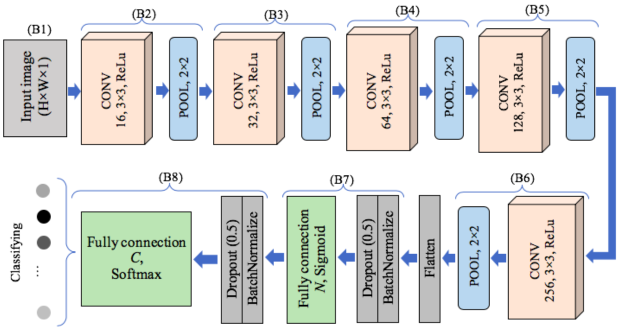
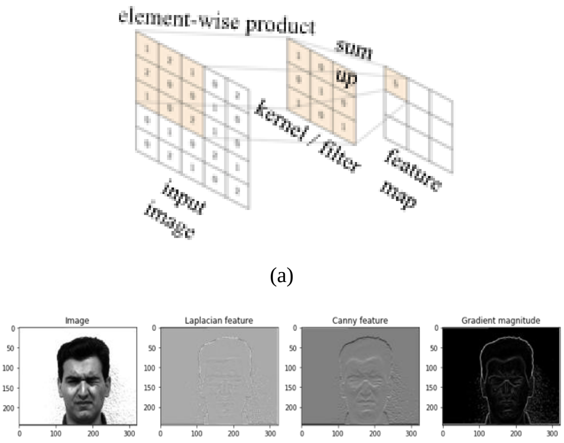


Fig.7. Architecture of proposed CNN model

• B1 block is an input image of H × W × 1 size (high × wide × deep), it is 110×90×1 in our case. In order to reduce memory space of CNN model computation, we can use gray-level input images (the third dimension (depth) in image size is 1 instead of three channels in color images (red, green, blue). So, input images are converted to gray-scale if they are color in the preprocessing image phase.

• B2, B3, B4, B5 and B6 blocks are convolutional neuron layers with different filters and window size of kernel functions. Normally, 3×3 of window size is often used and number of filters is increased through layers. In our case, they are 16, 32, 64, 128 and 256 in turn. In context of CNN, kernel is also called filter or feature detector. ReLu activation function is used in this neuron layer. In each block, we use max pooling layer with window size 2×2 immediately after the CONV layer. This affects to intend improving the sparse features of whole network and dependency avoiding on passing parameters among neurons. Kernel functions act as features detector based on element-wise production between input images and them, then sum them up to feature map. This operation is usually applied by a stride 1. Fig.8 illustrates convolution operations (a) and results of three popular filters (Laplacian, Canny and Gradient magnitude detections) on an image (b).



(a)

(b)

Fig.8. Convolution operations and examples

In general, the more using convolution layers, the more opportunities to extract complex features, thereby expecting the proposed model tend to learn to identify objects better [9]. For example, in face recognition problems, a CNN model can learn to detect edge features from raw pixels in first CONV layers, then using these edge features to detect shapes like eye, nose or mouth of faces in next CONV layers. Lastly, for these shapes, the model can detect higher-level features, such as face shapes in higher layers.

In Fig.9 (a / b / c), we illustrate results of filters in B2, B4, and B6 blocks. Size of images after processing of each block decreases with coefficient ½ (after B2 is haft of input (i.e. 55×45), after B4 is a fraction 1/8 of input (i.e. 14×12), after B6 is a fraction 1/32 of input (i.e. 4×3). Visualization of results shows that later images becomes blurrier, demonstrating ability to abstraction and representation most common features of a face, regardless of any image view. Or it can be said that these features of faces have highest immutability for any their different images, whether in different forms, brightness, colors, or sizes.

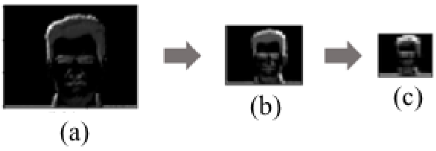


Fig.9. Output images of B2, B4 and B6 blocks

• B7 and B8 blocks are fully connected layers. These layers aim to classify persons from extracted features in previous layers. They act as a classification. So, we design a quite large number of neurons in B7 block, whereby, we set $N = 1024$. For non-linear classification problems and getting potential of good results, we use “sigmoid” activation function in B7 layer instead of ReLu. However, we can use ReLu for reducing computation in particular problems. In B7 block, we set probability of Dropout 0.5.

B8 block is final layer (or output layer), which is a distribution for classifying different objects with “softmax” activation function (1). In CNN model, using “softmax” function of output layer is better way for classification problems and it is the most widely used.

$$O_k = softmax(y_k) = \frac{e^{y_k}}{\sum_{i=1}^{M} e^{y_i}} \tag{1}$$

where, y_k is aggregate of all k^{th} neuron input and weights, M is number of neurons in the layer. Fig.10 illustrates “softmax” output of neurons in last layer, it has the highest value at 1^{st} neuron ($N1$) in $M = 15$ neurons, while the rest is much smaller than it, they are almost near to zero or even some of them are zero. Total of all “softmax” output neurons in this layer is equal to 1, of course.

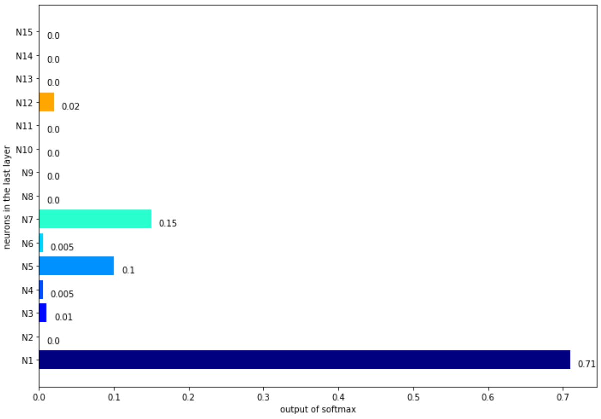


Fig.10. Example of “softmax” output N1-N15

The final output of CNN model is determined based on the maximum value of “softmax” output of neurons in B8, for CNN model having M classes (i.e., there are M neurons in output layer), we have classified output formula as following:

$$\text{out} = argmax_{C_k}\{O_k : k = 1, \ldots, \text{M}\} \tag{2}$$

where, O_k is k^{th} output of final layer and it is corresponding to C_k class. In Figure 3.7, the highest value at 1^{st} neuron means the corresponding input image being classified to C_1 label.

The proposed CNN architecture has 7 neuron layers which is much smaller than FaceNet [12] of 14 layers, VGG [11] of 19 layers. It has 3 more layers in comparison to [9], however, for CNN model with deep learning, too small layers will not be extracted important features from images, it may cause larger number of errors in applications. So, authors in [15] need to apply complex image processing for enhancing before using the CNN model in [9] in order to reduce errors. This proposed model also is smaller than [21], they used 5 CONV layers which same as ours but 1 more

FC layer. CNN model in [8] has 16 CONV layers, model of [12] has 11 CONV and 3 FC layers, model of [13] has 8 CONV and 1 FC layers, model of [15] uses ResNet architecture with great complexity. So, all of these models have much more complexity than our proposed model. In addition, our model uses 2 FC layer for classification where the ‘sigmoid’ FC layer is used for non-linear activations of classifications from extracted features. This can make more flexibility in classifications.

C. Applying CNN model to monitor students in LMS

Our proposed CNN model is now used for integrating with LMS to monitor students whenever they learn. Students have to sign in LMS through their ID and password for authentication before learning. We should verify them immediately signed in LMS by using CNN model, this process as follow:

Step 1) Open client’s camera for capturing images of students. This activity is integrated with LMS in client side for every student.

Step 2) Pre-process captured images to get face images from client’s camera of students.

Step 3) Recognize face images to get ID of students or ‘unknown’, send notifications to LMS for announcing and monitoring whole learning time of students.

This process is repeated in given period of time until logging out LMS by students and they stop learning. Normally, in order to avoiding overload computation systems and decreasing transferred images from client to the system, we set repeatedly time to 60 seconds in our cases. Fig.11 shows overall integrated system LMS and CNN model.
‘

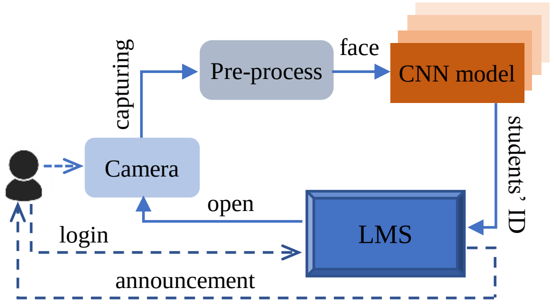


Fig.11. Overall integrated system

We call LMS-integrated system because it does not need much modification of current LMS. Therefore, LMS can run independently and the CNN model for face recognition system (FRS) is an additional running beside LMS. This is easy for integrating FRS into any current LMS.

4. Experimental Results

A. Dataset and parameters

In this experiment, we use three popular datasets, AT&T, Yale and LFW (Labeled Face in the Wild), and of course, our dataset of students. These datasets are published and widely used for face recognition researching [1, 9, 10, 13, 14, 15, 17, 19].

(1) AT&T dataset (also called ORL) created by AT&T Laboratory of Cambridge University, 2002. It includes 400 images of 40 people with 10 different attitudes for each person. All images are taken on a dark, homogeneous background with attitudes of upright position, frontal and slightly tilted left or right, up or down. The face of every person is observed, which is not covered by any objects. All images are multi-level gray. The following figure illustrates images with different attitudes of a person in this dataset.

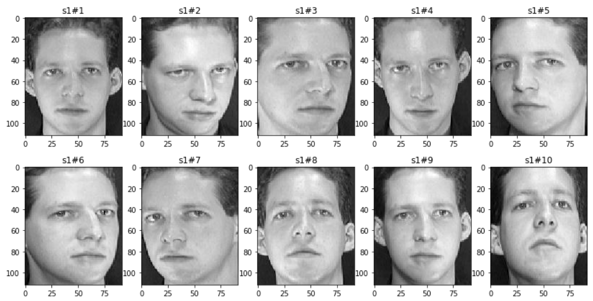


Fig.12. Images of "s1" in AT&T dataset

(2) Yale dataset is created by Computer Control and Visual Center at Yale University. It consists of 165 images taken from the front and in multi-level gray of 15 different people. There are 11 images for each person describing different facial expressions (normal, sad, happy, surprised, sleepy and winking) and conditions such as light (right light, center light and left light), they also include photos with glasses or no glasses.

(3) LFW dataset has diversity number of images from 5 to 530, we just use persons having 20 or more face images, it is so called LFW20. So, LFW20 has 3,023 images of 62 persons. Fig.13 shows number of images with respectively number of persons, e.g., there are 5 persons having 20 face images in the dataset.

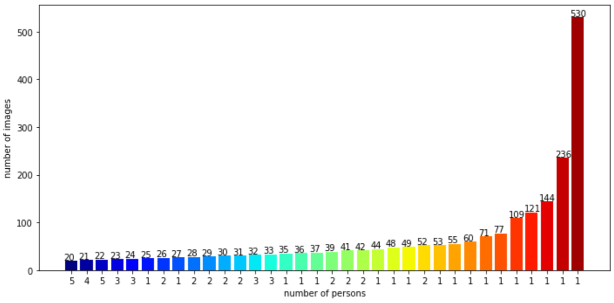


Fig.13. LFW20 distribution images and persons

(4) Our dataset is collected in an online class with 24 students. It has 1,005 face images from lowest number 5 images to largest number 222. Fig.14 shows many different poses and illuminations of our face images. Some images are even captured from almost left side of the face, e.g., they are '~Van Lam.1', '~Van Lam.2' and '~Van Lam.5'.

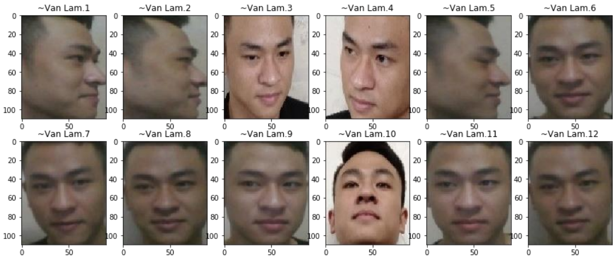


Fig.14. Faces in different poses and illuminations

For running experiment, we randomly divide a dataset D into 5 folds in same size, three folds of them are used for training (D^{tr}), a fold is used for validation (D^{va}), and the remaining fold is D^{te} for testing. D^{te} is unseen data while D^{tr} and D^{va} are seen data for constructing model. This scenario is similar to 5-folds cross validation, we run five times for each dataset, each time uses a fold for testing in turn. Overall results are average of each running. In each running time, D^{tr}, D^{va} are augmented by applying transformations of images including $\alpha = \{noise,\ rotation,\ zoom,\ shifting,\ flip\}$. Parameters for each transforming operation is used in Table 1. For small datasets, we randomly generate more augmented images than larger one in order to get enough diversity for avoiding overfit in training and get high results in application. All datasets except LFW are augmented to minimum 100 images in every person, because of a large-scale LFW dataset, we keep it origin for training.

In training the model, we use a method of first-order gradient-based optimization of stochastic objective functions (called Adam [20]). It is based on adaptive estimates of lower-order moments. Learning rate is adjusted in decreasing by cosine annealing schedule [17.Los] as following:

$$\eta_t = \eta^{min} + \frac{1}{2}(\eta^{max} - \eta^{min})\left(1 + cos\left(\pi \frac{T^{cur}}{T^{max}}\right)\right) \tag{3}$$

where, η^{min} and η^{max} are ranges for the learning rate, T^{max} is maximal epochs of learning, T^{cur} is current epoch of learning.

Table 1. Parameters for experiments running

No.	Parameters	Values
Augmentations		
1	noise variance	0 ÷ 0.01, Gaussian
2	rotation angle (left or right)	-15° ÷ +15°, rotation center is image center
3	shift (left, right, up, down)	-10% ÷ +10%, percentage from center
Training CNN model		
4	learning rate	$\eta^{min} = 10^{-1}, \eta^{max} = 10^{-5}$
5	batch size	64
6	epochs	150

Fig.15 shows 11 augmented images from a person at top left of the picture that named '~Van Lam.org', '~' in the title denotes some absent characters for short. Augmented faces are titled ending by 'aug1', 'aug2', ..., 'aug11'. Some of them are operated on almost augmenting operations, e.g., '~Van Lam.aug5' is result of right and down shifting, flipped, rotation and noise.

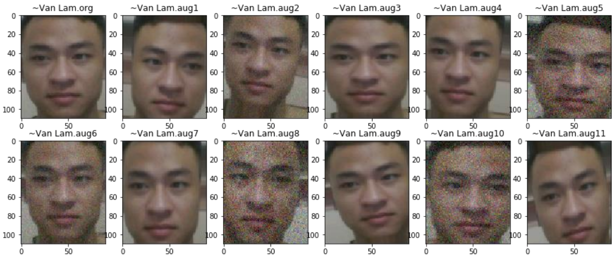


Fig.15. Augmented faces from a person in our dataset

The experiment is operated on a system with Tesla K80 GPU processor configuration, 12Gb RAM. So, we choose moderate-scale datasets for running experiment. This system is installed Python environment, frameworks and basic libraries for machine learning such as numpy, matplotlib, tensorflow, keras, and so on. Accordingly, our program is built on Python language and uses frameworks of tensorflow with keras library interface, these libraries provide powerful features for image processing and modeling CNN.

B. Results of experiments

In the proposed model, convolutional neuron blocks act as modules to extract facial features. Here, we show visual representations of feature extractions at last convolutional neuron layer. One of them is concentration (or interest) of the convolutional neuron layer on images, it is treated as "Gradient-based Localization" method [16], it is also called heat maps of enabled object class. Fig.16 shows heat maps of a convolution layer in the model for persons in AT&T dataset. As seen, all images have heat maps that are almost visible on the face of persons. This implies the convolution layer concentrating on faces where areas of forehead, cheeks, mouth, ears and chin to get individual features of them for recognition. Naturally, this shows that when you are not interested in important areas of the face, it is difficult to identify the person correctly.

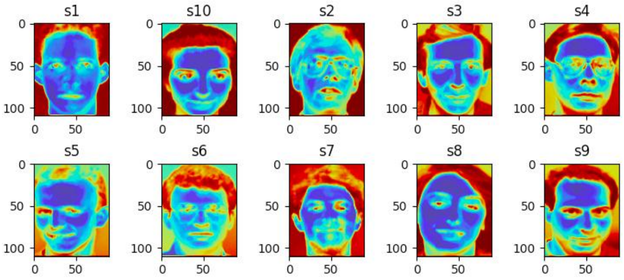
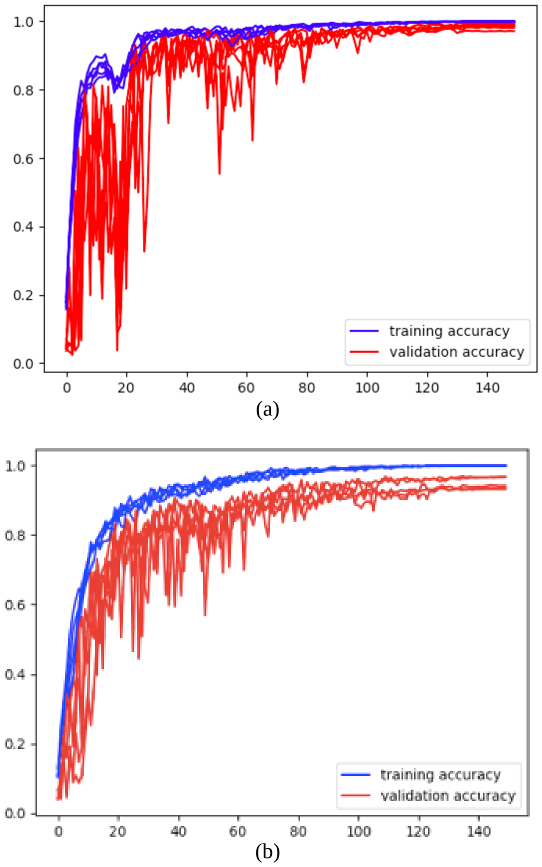


Fig.16. Heat maps of a CONV layer in the model

Training and validation accuracy of two datasets (AT&T (a) and our faces (b)) are showed in Fig.17. Where, accuracy of validation in AT&T is almost nearby accuracy of training from 100th epoch to the end. Although they are nearby 1, the accuracy of testing is still equal to 1 (i.e., it is 100% in Table 2). It is explained that the training and validation data are augmented to be very large (4,000 images for training and 2,000 images for validation) whilst the testing data is kept to be 20% of original dataset (80 images). Validation accuracy of our dataset is not very nearby accuracy of training, because this dataset is much more diversity of poses and illuminations than AT&T. It is also the reason why accuracy of testing is not equal to 1, it is only 98.21%.



(a)

(b)

Fig.17. Accuracy of training and validation in (a) AT&T and (b) our dataset

Overall accuracy of testing in the experiment are shown in Table 2. We output results of each running (from Run.1 to Run.5), they involve accuracy of Top1 recognition and accuracy of Top5 recognition. Top1 means the best 'softmax' output of the model is chose for recognition, Top5 means five largest 'softmax' outputs are chose for recognition. For each dataset, we average testing accuracies of five experiment running, they are bold. There are two datasets (AT&T and Yale) of 100% testing accuracy in both Top1 and Top5 cases, it is because of very small dataset and not too diversity of poses and illuminations of faces. In LFW and our dataset, faces are very different poses, for instance in Figure 4.3. Further, we apply some kind of operations in augmenting that it makes more many different poses and illuminations of faces in datasets. So, LFW has testing accuracy of 99.63% in Top1 and 99.83% in Top5. Although our dataset is small than LFW in both number of persons and among images, our dataset has lower testing accuracy than LFW, they are 98.21% in Top1 and 99.30% in Top5. It is because of ours has more many different poses and illuminations of faces than LFW. So, it is difficult for learning the model and even some images are not learned to extract features for recognition.

Table 2. Overall testing accuracy

No.	#Run	% Accuracy (Top1 / Top5)
Yale		
7	Run.1	100 / 100
8	Run.2	100 / 100
9	Run.3	100 / 100
10	Run.4	100 / 100
11	Run.5	100 / 100
12	**Average**	**100 / 100**
AT&T		
13	Run.1	100 / 100

14	Run.2	100 / 100
15	Run.3	100 / 100
16	Run.4	100 / 100
17	Run.5	100 / 100
18	**Average**	**100 / 100**
LFW		
19	Run.1	99.67 / 99.83
20	Run.2	99.83 / 99.83
21	Run.3	99.00 / 99.67
22	Run.4	99.83 / 100
23	Run.5	99.83 / 99.83
24	**Average**	**99.63 / 99.83**
Our faces		
1	Run.1	100 / 100
2	Run.2	100 / 100
3	Run.3	100 / 100
4	Run.4	95.51 / 98.51
5	Run.5	95.52 / 98.01
6	**Average**	**98.21 / 99.30**

In comparing these results with other articles, we just use testing accuracy of Top1. In Table 3, ‘*’ indicates no using CNN model of methods, ‘#’ means unknown scenario dividing datasets in experiments, ‘-’ is no experiment result. The best accuracy value is bold. On AT&T dataset, results of the proposed method and [21] are the best with testing accuracy of 100% whilst methods in [1, 9, 10] are from 94.0% to 98.3%. Our proposed method is the best with 100% accuracy on Yale dataset, the second highest accuracy is 99% in [10], it is 1% lower than ours. The method in [17] has the best accuracy with 99.86% in LFW dataset, our method reachs to the second highest accuracy with 99.63%. This comparison shows that our proposed CNN model has significant meaning of experiment results, it is potential effectiveness in practical applications.

Table 3. Comparison on testing accuracy

Methods	AT&T	Yale	LFW
[9]	98.30	-	-
[1]*	#98.00	#97.70	-
[10]*	94.00	99.00	84.00
[15]	-	#94.60	#96.70
[21]	**100**	-	-
[17]	-	-	**#99.86**
Our proposed	**100**	**100**	99.63

5. Conclusion

In this paper, we propose a model based on architectures of convolutional neural networks (CNN) to human face recognition problems. This model has 5 convolutional neuron layers (CONV) and 2 fully connected neuron layers (FC). We use 110×90×3 input size, therefore, it has about 3,5 million parameters in total which is smaller than almost others, even it is very much smaller than state-of-the-art models [17] with hundreds of millions of parameters. Thus, it can be affirmed that our model has a low level of complexity which called lightweight version, it is suitable for general computing system in widely used and it also brings high feasibility in practical applications.

Although our model is lightweight version, the experimental results show that quality and effectiveness of face recognition is high, from 99.63% to 100% accuracy. It is good acceptable for applications. Currently, due to computing limitations, we only apply small number of training times and datasets are not too large, if the model is trained at a deeper level with bigger datasets, it is expected to bring higher results.

We also design an LMS-integrated system with using the proposed lightweight CNN model for face recognition. The integrated system is flexible and easy for applying in an LMS, it just needs two connections from LMS to CNN model for sending captured images and receiving recognized faces. Therefore, LMS can connect to CNN model whenever it needs, it makes independently running of these two systems. This is easy for integrating the proposed CNN model into any current LMS.

This research uses ‘softmax’ loss to the CNN model for training, however, there are state-of-the-art losses which can be used to get more discriminative between categories. Some well-known forms of CONV blocks such as Inception or residual blocks or "Squeeze-and-Excitation" block [17] may be used for improving ability of feature extracting at deeper and more complexity. So, for further research, we focus on improving accuracy and efficiency by applying some well-known CONV blocks and losses for CNN model. Next, we will design an image data collection system to create training dataset for the model in practical applications, thereby building an application for reality problems such as online students monitoring system.

Acknowledgment

The authors wish to thank our colleagues in Hanoi Open University for collecting data from online students. This work was supported in part by a grant from Ministry of Education and Training, Vietnam.

References

[1] M. A. Abuzneid and A. Mahmood, "Enhanced Human Face Recognition Using LBPH Descriptor, Multi-KNN, and BPNN", IEEE Access, Vol. 6, pp.20641-20651, 2018.

[2] Brandon Amos et al., "OpenFace: A general-purpose face recognition library with mobile applications", CMU School of Computer Science, Tech. Rep., 2016.

[3] Shraddha Arya and Arpit Agrawal, "Face Recognition with Partial Face Recognition and Convolutional Neural Network", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol.7, Iss.1, pp.91-94, ISSN: 2278 – 1323, 2018.

[4] Qiong Cao et al., "VGGFace2 - A dataset for recognising faces across pose and age", IEEE Conference on Automatic Face and Gesture Recognition (http://www.robots.ox.ac.uk/ ~vgg/data/vgg face2), 2018.

[5] Lionel Landry S. Deffo et al., "CNNSFR: A Convolutional Neural Network System for Face Detection and Recognition", International Journal of Advanced Computer Science and Applications, Vol. 9, No. 12, pp.240-244, 2018.

[6] Ekberjan Derman and Albert Ali Salah, "Continuous Real-Time Vehicle Driver Authentication Using Convolutional Neural Network Based Face Recognition", 13th IEEE International Conference on Automatic Face & Gesture Recognition, 2018.

[7] Ayham Fayyoumi and Anis Zarrad, "Novel Solution Based on Face Recognition to Address Identity Theft and Cheating in Online Examination Systems", Advances in Internet of Things, Vol.4, pp.5-12, 2014.

[8] Chunrui Han et al., "Face Recognition with Contrastive Convolution", European Conference on Computer Vision: Computer Vision – ECCV, pp.120-135, 2018.

[9] Patrik Kamencay et al., "A New Method for Face Recognition Using Convolutional Neural Network", Digital Image Processing and Computer Graphics, Vol. 15, No. 4, pp.663-672, 2017.

[10] Hoda Mohammadzade et al., "Pixel-Level Alignment of Facial Images for High Accuracy Recognition Using Ensemble of Patches", Journal of the Optical Society of America, A.35(7), 2018.

[11] Karen Simonyan and Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, University of Oxford, 2015.

[12] James Philbin et al., "FaceNet: A Unified Embedding for Face Recognition and Clustering", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[13] Kevin Santoso et al., "Face Recognition Using Modified OpenFace", 3rd International Conference on Computer Science and Computational Intelligence, Procedia Computer Science, No.135, pp.510–517, 2018.

[14] R. Syafeeza et al., "Convolutional Neural Network for Face Recognition with Pose and Illumination Variation", International Journal of Engineering and Technology (IJET), pp.44-57, 2014.

[15] Muhtahir O. Oloyede et al., "Improving Face Recognition Systems Using a New Image Enhancement Technique, Hybrid Features and the Convolutional Neural Network", IEEE Access, Vol. 6, pp. 75181-75191, 2018.

[16] Ramprasaath R. Selvaraju et al., "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization", IEEE International Conference on Computer Vision (ICCV), Electronic ISSN: 2380-7504, 2017.

[17] Mei Wang et al., Deep Face Recognition: A Survey, School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, 2019.

[18] Rikiya Yamashita et al., Convolutional neural networks: an overview and application in radiology, Insights into Imaging, vol.9, pp.611–629, 2018.

[19] Md Zahangir Alom et al., A State-of-the-Art Survey on Deep Learning Theory and Architectures, Electronics, vol.8, no.292, 2019.

[20] Diederik P. Kingma et al., Adam: A Method for Stochastic Optimization, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015.

[21] Umara Zafar et al., Face recognition with Bayesian convolutional networks for robust surveillance systems, EURASIP Journal on Image and Video Processing 2019:10.

[22] Ni Kadek Ayu Wirdiani, Real-Time Face Recognition with Eigenface Method, I.J. Image, Graphics and Signal Processing, vol.11, pp.1-9, 2019.

[23] Rafflesia Khan and Rameswar Debnath, Human Distraction Detection from Video Stream Using Artificial Emotional Intelligence, I.J. Image, Graphics and Signal Processing, vol.2, pp.19-29, 2020.

[24] Zoran Kotevski et al., On the Technologies and Systems for Student Attendance Tracking, I.J. Information Technology and Computer Science, vol.10, pp.44-52, 2018.

Author’s Profile

Duong T. Long is a lecturer in Information Technology Faculty at Hanoi Open University. He received his PhD degree of Information Technology from Vietnam Academy of Science and Technology (VAST) in 2011. His research interest is Machine Learning, Artificial Intelligence, Deep Learning, Computer Vision, Fuzzy Logic and soft computing with real-world applications.

How to cite this paper: Duong Thang Long, " A Lightweight Face Recognition Model Using Convolutional Neural Network for Monitoring Students in E-Learning", International Journal of Modern Education and Computer Science(IJMECS), Vol.12, No.6, pp. 16-28, 2020.DOI: 10.5815/ijmecs.2020.06.02