Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 192 (2021) 4093–4102

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Face Recognition Smart Attendance System using Deep Transfer Learning

Khawla Alhanaee[a], Mitha Alhammadi[a], Nahla Almenhali[a], Maad Shatnawi[a]*

[a]*Deprtment of Electrical Engineering Technology, Higher Colleges of Technology, Abu Dhabi, UAE*

Abstract

Face identification has been considered an interesting research domain in the past few years as it plays a major biometric authentication role in several applications including attendance management and access control systems. Attendance management systems are very important to all organization though they are complex and time-consuming for managing regular attendance log. There are many automated human identification techniques such as biometrics, RFID, eye tracking, voice recognition. Face is one of the most broadly used biometrics for human identity authentication. This paper presents a facial recognition attendance system based on deep learning convolutional neural networks. We utilize transfer learning by using three pre-trained convolutional neural networks and trained them on our data. The three networks showed very high performance in terms of high prediction accuracy and reasonable training time.

Keywords: Type your keywords here, separated by semicolons ;

1. Introduction

All organizations need an attendance management system to maintain a record of their staff attendance either manually or automatically. Students' daily attendance in class is essential for performance evaluation and quality

* Corresponding author. Tel.: +971506151987.
E-mail address: maad.shatnawi@hct.ac.ae

1877-0509
10.1016/j.procs.2021.09.184

monitoring. Calling names or signing on papers are the traditional methods used in most organizations, which are both time consuming and insecure [1]. On the other hand, most automatic human identification systems are based on traditional methods such as fingerprints, passwords, and ID scans. However, all these methods have several limitations such as forgetting a password or losing an ID card. Therefore, the most suitable method to ensure full security and to save history records is through a smart face recognition system [2]. It is a rapidly growing field in the recent time, and it plays an important role in security as it is a very precise technique to identify and verify people [3] [4].

Transfer learning is a form of machine learning where a model is built for a specific task and then reused on a second task as the starting point to be modified. It is used in deep learning as a pre-trained model in computer vision and natural language processing tasks to develop neural network models on these problems [5]. Transfer learning is very useful in deep learning problems because most real-world problems usually have billions of labeled data, and this requires complex models [6]. It is a perfect technique for optimization, time saving and achieving better performance. Developers can use transfer learning to merge different applications into one. They can quickly train new models for complex applications. Moreover, transfer learning is a good tool to improve the accuracy of computer vision models [5].

In this work we present a facial recognition attendance system based on deep learning convolutional neural networks (CNN). We utilize transfer learning by using three pre-trained convolutional neural networks and trained them on our data which contains 10 different classes where each class includes 20 facial images. The three networks showed very high performance in terms of high prediction accuracy and reasonable training time.

2. Pre-Trained Networks

Pre-trained CNN models have distinct features that matter when selecting a network to deal with a certain issue. Network precision, speed, and size are the most significant features. Generally, selecting a network is alternated between these functions. For people who want to learn an algorithm or test out an established system, pre-trained models are an excellent source of support [7]. It is not always feasible to construct a model from scratch due to time constraints or computational limitations, which is why pre-trained models exist. Several pre-trained CNN models are publicly available [8]. In this work, we investigated three pre-trained networks; AlexNet, GoogleNet and SqueezeNet.

2.1. AlexNet

One of the most common architectures of neural networks to date is AlexNet. It has been used to train millions of images and classify them into object categories such as faces, fruits, cups, pencils, and animals. As an input, the network takes an image and outputs a label for that object in the image. Also, the probabilities for each of the object categories. The input dimensions of the network are 227 × 227 × 3 RGB images [9] [10] [11] [12]. AlexNet architecture is shown in Fig. 1.

2.2. GoogleNet

The GoogleNet structure have 22 layers deep in addition to 5 pooling layers [13]. In total, there are 9 initiation modules stacked linearly. It uses 1×1 convolution filter as well. The net has a very good computation and memory efficiency because of the parallel network implementation and layer reduction, the model size is smaller than others [9] [12]. GoogleNet architecture is presented in Fig. 2.

2.3. SqueezeNet

SqueezeNet is an 18 layers deep convolutional neural network. The pre-trained network will categorize images into 1000 categories of objects. For a wide range of images, the network has learned complex function representations. The purpose of using SqueezeNet is to build a smaller neural network with small datasets that can

integrate into computer memory more easily and can be communicated over a computer network more easily [9] [14] [12]. SqueezeNet Net architecture is shown in Fig. 3.

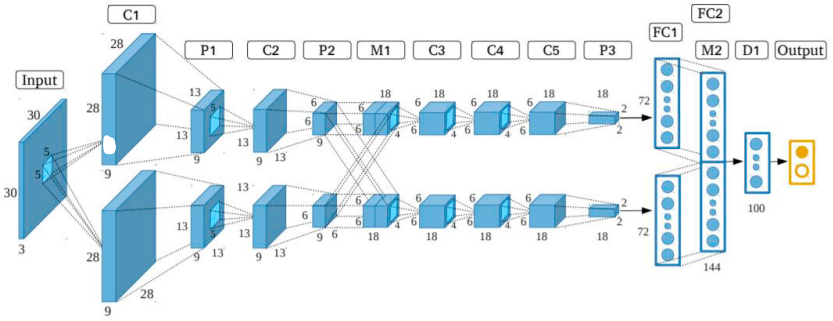


Fig. 1. AlexNet architecture [12].

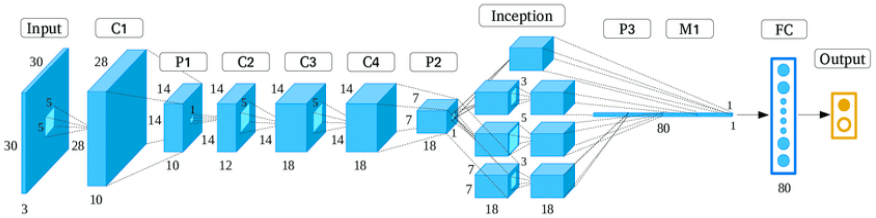


Fig. 2. GoogleNet architecture [12].

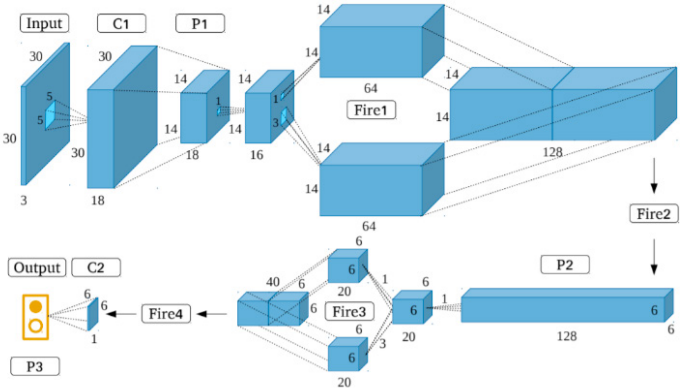


Fig. 3. SqueezeNet architecture [12].

3. Related Work

Face identification has been investigated by several researchers. One existing solution [15] used a technique of face recognition that uses features derived from Discrete Cosine Transform (DCT) coefficients, along with a

classifier based on Self Organizing Map (SOM) classifier. In MATLAB, the method is tested and contains subjects with various facial expressions. The device can achieve a recognition rate of 81.36% for 10 consecutive trials by preparing for this program for approximately 850 epochs. A decreased space of features, specified for this system. This makes the scheme well adapted for the implementation of low-cost, real-time hardware.

Arsenovic et al. [16] proposed a deep learning-based face recognition attendance system. This model is made up of several key steps developed using the most modern techniques available today such as CNN cascade for face detection and CNN for face embedding generation. On a limited dataset of original face images of employees in the real-time world, the overall accuracy was 95.02%. The proposed face recognition model could be used in other systems also.

Fu et al. [17] proposed a solution that integrates two deep learning algorithms, Multi-Task Cascade Convolution Neural Network (MTCNN) face detection, and Centre-Face recognition, to create a university classroom automated attendance system. The system will report those three violations of classroom discipline for automatic attendance: absence, lateness, and leaving early, according to a significant number of experimental findings. After class, an attendance table with all students' learning status is automatically registered. The system quickly recognizes faces, taking just 100 milliseconds per frame and achieving high accuracy. The model has a 98.87 % accuracy rate, a true positive rate of under 1/1000, and a false positive rate of 93.7 percent.

Zulfiqar et al. [18] proposed a face recognition system based on convolutional neural networks that detect faces in an input image using the Viola-Jones [19] face detector and automatically extract facial features from detected faces using a pre-trained CNN for recognition. For efficient convolutional neural network training, a huge database of facial images of subjects was created, which was augmented to increase the number of images per subject and provide various illumination and noise conditions. Furthermore, for deep face recognition, an optimized pre-trained CNN model and a set of hyper-parameters were experimentally chosen. The efficacy of deep face recognition in automatic biometric authentication systems was demonstrated in promising experimental findings with an overall accuracy of 98.76%.

4. Method

The proposed approach consists of several stages: data collection, data pre-processing, data augmentation, CNN training and validation, and system testing.

4.1. Data Collection

Our dataset is a collection of 200 images that were collected using an iPhone 12 front-facing camera which is a 12-megapixel, f/2.2 lens. The data was classified into 10 classes, each individual class includes 20 images. Those 10 classes represent 10 people from both genders as shown in Fig.4.

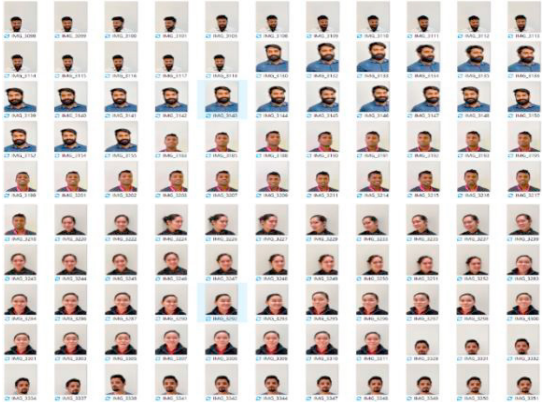


Fig. 4. Dataset.

4.2. Data Formatting

The collected data is used in a JPG file formatting. The image sizes are ranging between 3.00 MB and 4.00 MB. Each net has different input size. Therefore, we had to resize the images to the corresponding input dimensions of the network. SqueezeNet and AlexNet uses 227 × 227, while GoogleNet uses 224 × 224. All the images taken are in RGB colors which is good to extract the right features.

4.3. Data Augmentation

Data augmentation is a tool used to increase the amount of data by inserting slightly changed copies of existing data or newly produced synthetic data from existing data. It regularizes and helps in the training of a machine learning model to minimize overfitting. In deep learning, data augmentation comes in a form of geometric transformations, flipping, color alteration, cropping, rotation, noise injection and random erasing are used to enhance the image *[8]*. In our trained networks, we used data augmentation by taking multiple images from different angles, environments and conditions, orientation, location, and brightness as shown in Fig. 5. After importing our data to the network, we applied two types of data augmentation which are rotation and scaling. A random rotation is performed on each image by an angle in the range of -90 to 90 degrees. A random scaling is performed on each image by a factor in the range of 1 to 2.

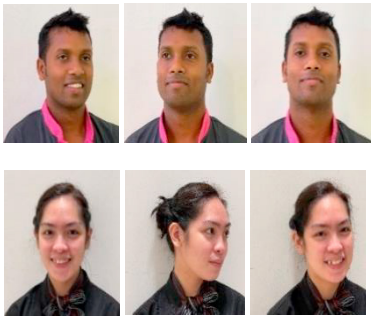


Fig. 5. Examples of data augmentation.

4.4. Choice of pre-trained networks

To train the convolution neural network on our data, we chose 3 nets to work with which are, SqueezeNet, AlexNet, and GoogleNet. SqueezeNet is small CNN and need less communication between servers during distributed training. Smaller CNNs are also easier to implement on hardware with limited memory, such as a field-programmable gate array (FPGA).

AlexNet can easily pass learned features to a special assignment with a smaller number of training images. AlexNet was developed to enhance the performance of the ImageNet challenge. This was one of the first Deep convolutional networks to reach significant accuracy. The overfitting problem is also solved by AlexNet by using drop-out layers where a connection is dropped with a probability of p=0.5 during testing. A probability of 0.5 was chosen because it was the best probability to match the net specifications and training options. This was set after many trials and changes. Although this prevents the overfitting of the network by having it escape from bad local minima, the number of iterations needed for convergence has also doubled.

The inception module in the GoogleNet architecture solved most of the challenges that large networks had. GoogleNet is scoring a 6.67% error rate which is close to human level performance. The architecture consisted of 22 layers of the Deep CNN reducing the number of parameters to 4 million (60 million compared from AlexNet).

4.5. Pre-trained network Tuning

In each of the three nets used, we tuned the pre-trained network parameters which are the convolution 2D layer and the classification output layer. In the deep network designer, we modified the filter size to 1×1, and the number of filters to 10 as we have 10 classes of data as shown in Fig. 6. We modified the classification output layer to suit with our output classification and labels.

Fig. 6. Convolution Layer Tuning.

4.6. Training

The generic procedure when training any network using transfer learning starts with modifying the parameters belonging to the base architecture. This includes choosing an appropriate learning rate, training time, number of epochs, and the validation frequency. When the initial rate is too low, the training process may become stalled, and when the rate is too high, the training process may become unstable or learn a sub-optimal set of weights too quickly. Therefore, we chose the initial learning rate to be set to 0.0001, the validation frequency is 10, and maximum epochs are 30 because it should be as high as possible to eliminate the failure of training pauses based on the error rates. To be specific, an epoch is a single learning cycle during which the learner is exposed to the whole training data set. Also, the minimum batch size is equal to 11, this fits the memory requirements (8.00 GB) of the CPU hardware that is operating using 1.8 GHz. The higher the number of epochs, the higher the accuracy of the network. Therefore, we had to choose a bigger number. The data set was randomly split into two parts: 70% of the data is used for training and 30% of the data is used for validation. The training parameters are illustrated in Fig. 7.

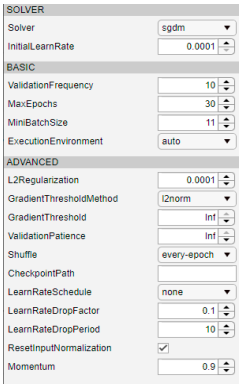


Fig. 7. Training Options.

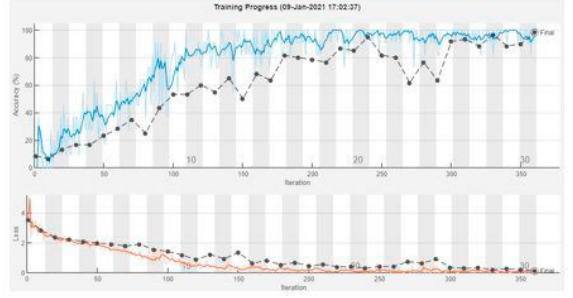
5. Results and Discussion

Training the SqueezeNet required 30 epochs in total with 12 iterations per epoch for the network to train the data very well and validate it. After 360 iterations, it achieved a validation accuracy of 98.33%. The training process took 26 minutes and 53 seconds. In addition, the validation frequency was done in a 10-iteration process to ensure that the system is trained well but not overfitting the data.

GoogleNet training required 30 epochs in total with 12 iterations per epoch for the network to train the data very well and validate it. After 360 iterations, it reached a validation accuracy of 93.33%. The network took 39 minutes and 21 seconds to complete the training. In addition, the validation was done in a 10-iteration process to ensure that the system is trained well but not overfitting the data.

Training the AlexNet used 60 epochs in total with 12 iterations per epoch for the network to train the data very well and validate it. After 720 iterations, it achieved a validation accuracy of 100% which is perfect accuracy indicating a well-trained network. The network took 76 minutes to complete the training, which is a long processing time compared to the previous two networks. In addition, the validation was done in a 10-iteration process to ensure that the system is trained well but not overfitting the data.

We used a system consisting of a single CPU that is operating using 1.8 GHz with an 8.00 GB RAM. As presented in Fig. 8, we used the same initial learn rate, the max iterations and single CPU for the three networks while the number of epochs was different for each network. It is observed that AlexNet is the best network in terms of validation accuracy but has the longest training time due to the number of parameters. SqueezeNet is the second-best choice which gives an accuracy of 98.33% with minimum training time which is 26 min 53 sec. GoogleNet gives the lowest accuracy among the three networks. The training and validation results of the three networks are summarized in Table 1.

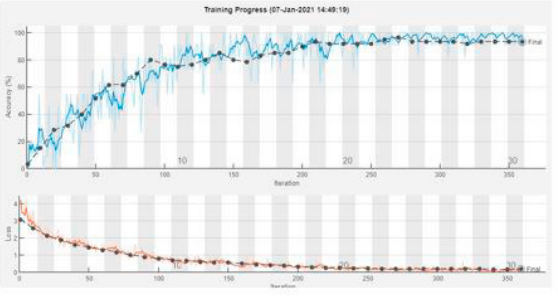


Results	
Validation accuracy:	98.33%
Training finished:	Reached final iteration
Training Time	
Start time:	09-Jan-2021 17:02:37
Elapsed time:	26 min 53 sec
Training Cycle	
Epoch:	30 of 30
Iteration:	360 of 360
Iterations per epoch:	12
Maximum iterations:	360
Validation	
Frequency:	10 iterations
Other Information	
Hardware resource:	Single CPU
Learning rate schedule:	Constant
Learning rate:	0.0001

Accuracy: Training (smoothed), Training, Validation

Loss: Training (smoothed), Training, Validation

(a)

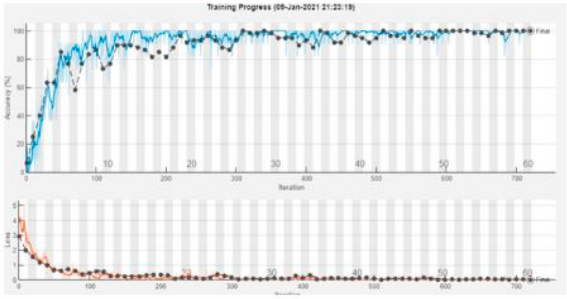


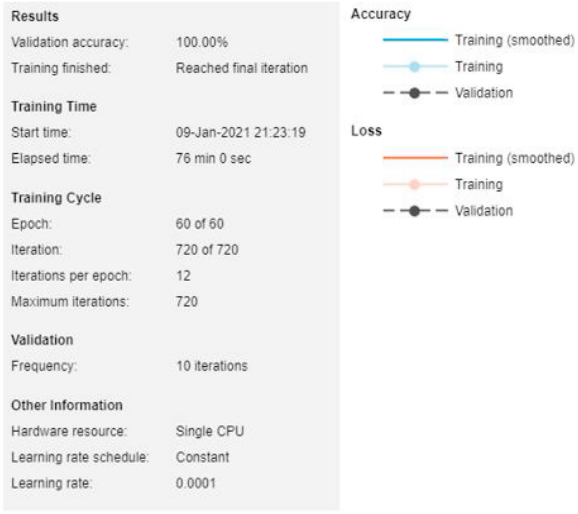
Results	
Validation accuracy:	93.33%
Training finished:	Reached final iteration
Training Time	
Start time:	07-Jan-2021 14:49:19
Elapsed time:	39 min 21 sec
Training Cycle	
Epoch:	30 of 30
Iteration:	360 of 360
Iterations per epoch:	12
Maximum iterations:	360
Validation	
Frequency:	10 iterations
Other Information	
Hardware resource:	Single CPU
Learning rate schedule:	Constant
Learning rate:	0.0001

Accuracy: Training (smoothed), Training, Validation

Loss: Training (smoothed), Training, Validation

(b)





(c)

Fig. 8. The training and validation results of (a) SqueezeNet; (b) GoogleNet; (c) AlexNet.

Table 1. Convolution Neural Network (CNN) Training Results.

Model	Learning Rate	Epochs	Validation Accuracy	Elapsed Time	Hardware Resources	Max Iterations
SqueezeNet	0.0001	30	98.33%	26 min 53 sec	Single CPU	360
GoogleNet	0.0001	30	93.33%	39 min 21 sec	Single CPU	360
AlexNet	0.0001	60	100%	76 min 0 sec	Single CPU	720

When tested on unseen images from the ten different classes, the three CNN models are able to successfully recognize the faces with a very high prediction confidence. The percentage shown on the upper part of the images refer to the confidence level of the trained network in predicting the corresponding labels. The very high confidence level indicated that the model predicts the image successfully. Figures 9a, 9b and 9c illustrate examples of the results achieved by SqueeseNet, GoogleNet and AlexNet, respectively.

To analyze the results, we used the same initial learn rate for the networks, the max iterations, single CPU while the number of epochs was different for each network. We can observe that AlexNet is the best network to use to train the data because it has the highest validation accuracy although it takes the longest training duration. SqueezeNet is the second-best choice which gives an accuracy of 98.33% with minimum elapsing time which is 26 min 53 sec. The third best network is the GoogleNet because it gives the lowest accuracy among the three networks.

(a) (b) (c)

Fig. 9. Testing Results of (a) SqueezeNet; (b) GoogleNet; (c) AlexNet.

A comparison of the proposed method to those obtained from previous methods is demonstrated in Table 2. The results obtained in Table 2 are based on our dataset. It is clearly shown that our approach outperforms other approaches.

Table 2. Comparison of face recognition approaches.

Approach	CNN Model	Accuracy
Fu et al. [17]	ResNet-101	99.7%
Zulfiqara et al. [18]	Resnet50	98.3%
Our Approach	AlexNet	100%

6. Conclusion

This paper presents a deep learning based facial recognition attendance system. We utilize transfer learning by using three pre-trained convolutional neural networks and trained them on our data. When compared to other approaches, the system showed very high performance in terms of high prediction accuracy and reasonable training time. The three networks are SqueezeNet, GoogleNet and AlexNet where they achieved a validation accuracy of 98.33%, 93.33% and 100% respectively. The proposed approach could be used in attendance and door access systems in many organizations such as government and private sectors, airports, schools, and universities.

This work can be extended by investigating more pre-trained CNN models and by including more human facial image data. It is interesting to investigate applying these models to masked face human identification tasks.

References

[1] M. Karunakar, C. A. Sai, K. Chandra and K. A. Kumar, “Smart Attendance Monitoring System (SAMS): A Face Recognition Based Attendance System for Classroom Environment,” *International Journal for Recent Developments in Science and Technology,* vol. 4, no. 5, pp. 194-201, 2020.

[2] S. Bhattacharya, G. S. Nainala, P. Das and A. Routray, “Smart Attendance Monitoring System (SAMS): A Face Recognition Based Attendance System for Classroom Environmen,” in *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*, 2018.

[3] G. Hua, M.-H. Yang, E. Learned-Miller, Y. a. T. M. Ma, D. J. Kriegman and T. S. Huang, “Introduction to the special section on real-world face recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 33, pp. 1921--1924, 2011.

[4] F. P. Filippidou and G. A. Papakostas, “Single Sample Face Recognition Using Convolutional Neural Networks for Automated Attendance Systems,” in *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)*, 2020.

[5] J. Brownlee, “A Gentle Introduction to Transfer Learning for Deep Learning,” 20 December 2017. [Online]. Available: https://machinelearningmastery.com/transfer-learning-for-deep-learning/#:~:text=Transfer%20learning%20is%20a%20machine,model%20on%20a%20second%20task.

[6] M. Xu, W. Cheng, Q. Zhao, L. Ma and F. Xu, Facial expression recognition based on transfer learning from deep convolutional networks, IEEE, 2015, pp. 702--708.

[7] P. Marcelino, "Transfer learning from pre-trained models," *Towards Data Science,* 2018.

[8] A. Gandhi, "Data Augmentation | How to use Deep Learning when you have Limited Data — Part 2," 2018. [Online]. Available: https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/.

[9] T. M. Ayyar, "A practical experiment for comparing LeNet, AlexNet, VGG and ResNet models with their advantages and disadvantages.," [Online]. Available: https://tejasmohanayyar.medium.com/a-practical-experiment-for-comparing-lenet-alexnet-vgg-and-resnet-models-with-their-advantages-d932fb7c7d17.

[10] A. Khvostikov, K. Aderghal, J. Benois-Pineau, A. Krylov and G. Catheline, "3D CNN-based classification using sMRI and MD-DTI images for Alzheimer disease studies," *arXiv preprint arXiv:1801.05968,* 2018.

[11] S.-H. Tsang, "Review: AlexNet, CaffeNet--Winner of ILSVRC 2012 (Image Classification)," *A Medium Corporation,* vol. 9, 2018.

[12] Z. Guo, Q. Chen, G. a. X. Y. Wu, R. Shibasaki and X. Shao, "Village building identification based on ensemble convolutional neural networks," *Sensors,* vol. 17, no. 11, p. 2487, 2017.

[13] R. Alake, "Deep Learning: GoogLeNet Explained," 23 December 2020. [Online]. Available: https://towardsdatascience.com/deep-learning-googlenet-explained-de8861c82765.

[14] V. Kurama, "A Review of Popular Deep Learning Architectures: AlexNet, VGG16, and GoogleNet," 1 June 2020. [Online]. Available: https://blog.paperspace.com/popular-deep-learning-architectures-alexnet-vgg-googlenet/.

[15] N. Soni, M. Kumar and G. Mathur, "Face Recognition using SOM Neural Network with Different Facial Feature Extraction Techniques," *International Journal of Computer Applications,* vol. 76, no. 3, pp. 7-11, 2013.

[16] M. Arsenovic, S. a. A. A. Sladojevic and D. Stefanovic, "FaceTime—Deep learning based face recognition attendance system," *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY),* pp. 000053--000058, 2017.

[17] R. Fu, D. Wang, D. Li and Z. Luo, "University classroom attendance based on deep learning," *2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA),* pp. 128--131, 2017.

[18] M. Zulfiqar, F. Syed, M. J. Khan and K. Khurshid, "Deep Face Recognition for Biometric Authentication," in *2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, 2019.

[19] Y.-Q. Wang, "An analysis of the Viola-Jones face detection algorithm," *Image Processing On Line,* vol. 4, pp. 128--148, 2014.