The 4th International Conference on Digital Arts, Media and Technology
and 2nd ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering

Face Recognition using Transferred Deep Learning for Feature Extraction

Amornpan Phornchaicharoen
School of Information Technology
King Mongkut's University of Technology Thonburi
Bangkok, Thailand, amornpan@gmail.com

Praisan Padungweang
School of Information Technology
King Mongkut's University of Technology Thonburi
Bangkok, Thailand, praisan@gmail.com

***Abstract*—Face recognition systems are a challenging field in computer vision. An important process and key to success is the feature extraction which requires a lot of data and time for learning. Deep learning has proven to be an outstanding method for extracting relevant features of image classification when a huge amount of data is available. However, it is not an easy task for face recognition, which consists of a small number of images per person considered as classes. This paper applies the idea of transferred learning for feature extraction to a face recognition application. The feature extraction part of the trained deep learning model from a different domain is transferred for extracting face features. Then, the multilayer perceptron neural network is used for model evaluation. Experimental results on public face databases show that the proposed method is highly efficient.**

***Index Terms*—classification, convolutional neural networks, deep learning, face recognition, feature extraction, machine learning, transferred learning, supervised learning**

I. Introduction

A face recognition system can make a machine distinguish and recognize the human face. It is a branch of pattern recognition and computer vision. The system was created in the late 19th-century [1], [2]. It can be used with both pictures and videos. Machine learning can learn to detect and recognize people's identity in a supervised manner. The process of face recognition can be divided into three main stages.

1) Face detection discovers essential components of the desired face section including face shape, eyes, nose, mouth, etc. There are four techniques for face detection, i.e. Knowledge-Based, Feature invariant approaches, Template matching methods, and Appearance-based methods [3], [4].
2) Feature extraction takes out important structures and information from the first stage. It transforms the original image into an understandable computer format in an array of numerical values. There are two main methods [1] which are Holistic template-based methods and Geometric feature-based methods. The former includes Principal-Component analysis (PCA), Eigenfaces, Fisherfaces and subspace LDA. It generates a feature vector based on the structure of the acquired data. The latter includes the Pure Geometry Methods, Dynamic Architecture, Hidden Markov Model and Convolution Neural Network (CNN).
3) Face Recognition / Classification recognizes a person's identity, with the discovered features that appear on a person's face. This stage can be done using machine learning and include k-Nearest Neighbor(kNN), Support vector machine (SVM) and Multi-Layer Perceptron (MLP).

The deep neural network is a path of machine learning, which usually combines the second and third stages into a single machine learning model. There are several structures but CNN is an outstanding model for image classification. It consists of many hidden layers such as convolutional layers, pooling layers, and fully connected layers. Each layer performs different functions such as feature extractor, image reducer, and classifier. The lower-level acts on the separation of the data components and to define a higher level [5]–[8]

Deep CNN has a lot of feature extraction layers, which are the layers between input layers and the last fully connected layer. Training the model needs a huge number of the images. This is because it consists of many parameters called weights that need to be adjusted. It easily overfits a small sample size of training images. It also needs a lot of computation power and time-consuming. There is an approach that avoids training the model from scratch called Transferred Learning. Transfer learning uses a pre-trained model where weights and bias values are already learned in a knowledge domain to be applied to another specific domain [6]–[8].

This paper applies deep transfer learning as the feature extraction step of face recognition. The pre-trained CNN is used in this domain which is a many-class dataset but has small sample size per class. The MultiLayer Perceptron neural network (MLP) is used as the face recognition step. The MLP are trained and evaluated on public face databases for performance measurement.

This paper is organized as follows. First, Section I introduces the face recognition process, deep learning, transfer learning and the main contributions of this work. Second, Section II explains the structure of Convolutional Neural Networks (CNN), the transfer learning and the pre-trained CNN. Third, Section III explains the experimental setup. Forth, Section IV reports the experimental results and discussion.

978-1-5386-8072-8/19/$31.00 ©2019 IEEE

Finally, Section V concludes this paper and outlines future work.

II. RELATED WORKS

Convolutional Neural Networks have proven to be an outstanding model for image classification. One of the well-known models is Inception version 3 (InceptionV3) training for large visual recognition challenge called ImageNet. It is one of the most often used transferred learning models.

A. Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) are a type of deep feedforward neural network. It is a proficient recognition model which is commonly used in pattern and face recognition [5], [7]–[9]. It consists of many layers including convolutional and pooling layers. They have great performance in terms of feature extraction from images [6], [10] in the cases where there are enough images for training.

1) Input Layers: The input layers refer to the layers of loading image data which have already been pre-processed. The input image must be equal in height, width and channels as it affects the calculation in the next layers.

2) Convolutional Layers: The convolutional layer uses many of fix sized filter masks, called a kernel, and performs the convolution operation to extract important patterns. The convolution output at position (i^{th}, j^{th}), that denotes $F(i,j)$, can be computed by equation 1. It is a discrete convolution operator, $\circledast$, of the input image $\mathbf{I}$ and convolution filter $\mathbf{K}$.

$$\mathbf{F} = \mathbf{I} \circledast \mathbf{K} \tag{1}$$

where

$$F(i,j) = \sum_m \sum_n I(i-m, j-n) K(m,n)$$

$I(i,j)$ is the intensity in the image at position (i^{th}, j^{th}) and $K(m,n)$ is the kernel value in the kernel at position (m^{th}, n^{th}).

3) Activation function: The activation function is used to define the output of any given neuron. In this work, we use two activation functions, i.e., Rectified linear unit (ReLU) and Softmax.

Firstly, *the ReLU* activation function is a discrete nonlinear operator with a max output indicative function as shown in equation 2. The output is zero for negative values and positive linearly for positive values.

$$\begin{aligned} F^*(i,j) &= \text{ReLU}(F(i,j)) \\ &= \max(0, F(i,j)) \end{aligned} \tag{2}$$

Secondly, *the Softmax* activation function is a generalization of the logistic function for multiclass classification. It is applied on the output layer where the decision is made. All output values defined by Softmax are real values in the range from 0 to 1 that sum to 1 to characterize the probabilities of v different output classes as shown in equation 3.

$$P_q = \frac{\exp(y_q)}{\sum_{v=1}^{c} \exp(y_v)} \tag{3}$$

where P_q is the defined output by the activation function corresponding to the q^{th} class, y_q is the output from the neuron q^{th} in the last layer of the neuron network and v is number of nodes in the last layer corresponding to the number of classes.

Note that, the label of all samples needs to be encoded by one-hot encoder. This produces a label vector for each sample, which contains 1 at the position corresponding the sample's class and zeros elsewhere. The label vector can be compared with the model output in the equation 3. It is used by the machine learning algorithm as the target for learning.

B. Transfer learning

Transfer learning extracts existing knowledge from a specific domain using knowledge learning from one or more auxiliary domains [11]. Transfer learning is commonly used to extract knowledge exploiting a set of meticulously manufactured features [9], [12].

In this work, we use the transfer learning concept for extracting image features using trained CNN. The result of transfer learning can be retrained in the classification layers. One of the benefits of this approach is not wasting time to building a model with large data training the CNN.

C. Pre-Trained CNN

There are academic benchmarks for computer vision and pattern recognition for validating the image recognition model. ImageNet [13] is one of the well-know image databases uses Large Visual Recognition Challenge. Several popular deep convolutional neural network architectures are trained with the ImageNet database, e.g., QuocNet [14], AlexNet [15], Inception (GoogLeNet) [16] BN-Inception-v2 [17] and Inception-v3 [18].

The trained Inception-v3 is widely used and consists of 22 deep CNN layers. The number of parameters is less than that from the AlexNet model. Inception-v3 is more accurate and smaller than AlexNet. This model classifies all images into 1000 classes. In this paper we transfer the trained CNN layers from Inception-v3 for extracting face features.

III. EXPERIMENTAL DESIGN

Dataset details and the experiments designed for evaluating the idea of transferred learning are presented in this section.

A. Facial Image Datasets

This experiment selects popular image databases that has been used in face recognition research. To cover the most related to real-life problems, we selected databases with the following characteristics and conditions: quantitative and qualitative, different lighting conditions, different facial expressions and high number of classes. The image datasets which use in this work are the Extended Yale Face Database B (Cropped) [19] and the Extended Cohn-Kanade Dataset (CK+) Dataset [20]. The images are divided into two sets the training

and the test set. The training set contains known labels for the training model, the test set is used to evaluate the accuracy of models.

1) The Extended Yale Face Database B (Cropped): The Extended Yale Face Database B (Cropped) [19] consists of different lighting conditions. Images are preprocessed by being manually aligned, cropped, and then resizing only the facial shape to 168x192 pixels. Figure 1 illustrates the sample of The Extended Yale Face Database B (Cropped) in different lighting conditions.

Fig. 1. The cropped version of "The Extended Yale Face Database B"

After removal of the damaged images, there were 2404 images from 38 individuals. Subsequently, they were divided into a training set for 1910 images (80%) and the test set for 494 images (20%).

2) The Extended Cohn-Kanade Dataset (CK+): The Extended Cohn-Kanade Dataset (CK+) [20] is a facial image database that displays facial expressing various emotions. It was developed by the Cohn-Kanade (CK) database in 2000. It consists of 593 sequences from 123 subjects which validated emotional labeling from the image sequences. The labels for sequences consists of six types of human emotions: anger, disgust, fear, happiness, sadness and surprise as shown in Figure 2.

Fig. 2. The Extended Cohn-Kanade Dataset (CK+)

Although there are six emotions, some people did not include photos of all emotions. We selected top four emotions of the highest number of individuals. Then only individuals having all four emotions were selected. The four emotions were happy faces having 2043 images from 107 individuals, surprised faces having 1946 images from 115 individuals, fear faces having 1716 images from 94 individuals and disgusted faces having 1710 images from 92 individuals. There were 81 individuals having the images of all the four facial expressions.

Three facial expressions were used for training while a facial expression was withheld for testing. Therefore, the training set consisted of 4205 images showing emotions of happiness, surprise and fear. The test set consisted of 1426 images showing the emotion of disgust.

B. Methodology

We separated the experiments into two approaches: approach 1 created an MLP model for each dataset. Approach 2 extracts feature using transferred learning and then created an MLP model for each dataset.

*1) **Approach 1:** Using only multi-layer perceptron neural network (MLP):* Each dataset was divided into a training set and test set. The training set was used for model training. The test set was withheld for evaluation as shown in Figure 3.

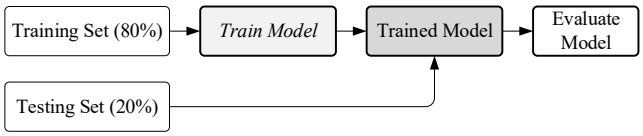

Fig. 3. The Experimental of the Multi-Layer perceptron neural networks.

*2) **Approach 2:** Using deep neural network to transfer feature extraction: CNN+MLP:* This approach used the same dataset in III-B1, but it added the feature extraction procedure using pre-trained CNN (Inception-V3). The features of the training set were extracted using the pre-trained CNN. The last layer of pre-trained CNN flats images into a feature vector. Then, all training feature vectors were used to train the classification model. Finally, the model was evaluated by using the test set which was extracted in the same way as in Figure 4.

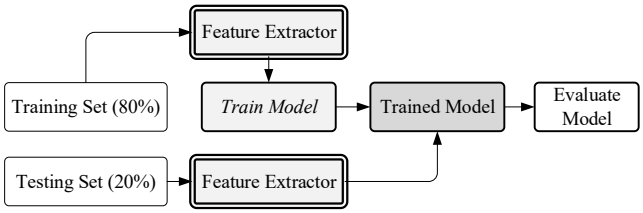

Fig. 4. The Experimental of Feature extraction of the Pre-trained CNN (Inception-V3).

3) Learning rate: The learning rate is a constant measurement in a trend of loss. If the learning rate is bigger than the best value, it means that it cannot find the minimum loss because of its overshot. However, if it is smaller than the suitable value, it will take a long time to go to the minimum. In this experiment, it demonstrates the best value of learning rate which is suitable for the classification model. The parameter uses to test the loss function which divides to five values including 0.00001, 0.0001, 0.001, 0.01, and 0.1. We can get the experimental result of loss values in figure 5.

Table I shows the loss, cross-entropy, of test dataset using different learning rate values. We found that the appropriated learning rate equal is 0.0001, since loss value of test dataset significantly decease after the few epochs of model training.

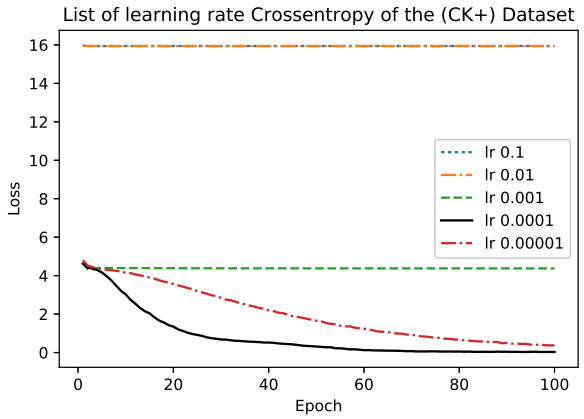

Fig. 5. The list of dropout cross-entropy of the (CK+) Dataset.

TABLE I
THE RESULT OF CROSS-ENTROPY OF DIFFERENCE LEARNING RATE VALUES.

Learning Rate	Training Loss	Testing Loss
0.1	15.9418	15.9033
0.01	15.9418	15.9033
0.001	4.2163	4.1334
0.0001	0.0313	0.0048
0.00001	0.3756	0.0552

4) Dropout: The dropout is a method to prevent the overfitting problem. Figure 6 shows the accuracy results from different dropout values. In this experiment, the learning rate was set to 0.0001. We can find that, the network does not seem to overfit to the training dataset while training. We can achieve high accuracy results using small value of dropout.

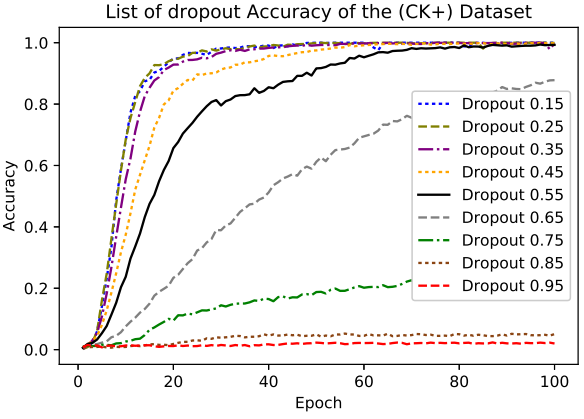

Fig. 6. The list of dropout accuracy of the (CK+) Dataset.

Table II shows accuracy results using different values of dropout. We found many appropriated dropout values having high accuracy such as 0.15, 0.25, 0.35, 0.45, 0.55 and 0.65. In contrast, the value of 0.75, 0.85, and 0.95 are in the underfitting category. Therefore, we average those appropriated values, that equal to 0.4, to represent the dropout value in the experiments.

TABLE II
THE ACCURACY RESULTS USING DIFFERENT VALUES OF DROPOUT.

Dropout	Training Accuracy	Testing Accuracy
0.15	1.0000	0.9993
0.25	0.9998	0.9986
0.35	0.9983	0.9972
0.45	0.9969	1.0000
0.55	0.9941	0.9909
0.65	0.8930	1.0000
0.75	0.2373	0.4804
0.85	0.0490	0.0968
0.95	0.0233	0.0105

The structure of multilayer perceptron (MLP) in the experiments consisted of 3 hidden layers with 1024, 512 and 256 nodes respectively. The Activation function of all nodes in the hidden layer are Rectified Linear Units (ReLU). The batch size value was set to 16, the number of samples that was propagated through the network in each step. Gradual descent algorithm with learning rate equal to 0.0001 was used as the machine learning algorithm. Also, we used the dropout technique for reducing over fitting problem in neural networks, it is set to 0.4 for each hidden layer.

All experiments were performed on a 64-bit Windows Server (10.0) OS, Intel Xeon Gold 5118 CPU @ 2.30GHz and 8 GB of RAM. This work used Inception-V3 [18], the deep convolutional neural network from Google as the pre-trained model. Implementation code in this paper is in Python with Keras [21] and TensorFlow library.

IV. RESULTS AND DISCUSSION

This section discusses the experimental results in 3 parts, i.e., Cross-Entropy, accuracy and computational time.

This paper focused on the benefits of using transferred learning on a high number of classes with a small sample size per class. There were many conditions affecting the results using different neural network models, e.g., initial weights, feature spaces and loss surfaces. Therefore, each experiment used the same environment by sharing parameters as much as possible.

A. The Cross-Entropy results

The Cross-Entropy refers to the loss function of the model. The Cross-Entropy of the test set using the model in approach 2 was far less than the one of the test set using the model in approach 1 as shown in the Table III. The loss value was slightly reduced after training over 200 and 60 epochs for Yale B and CK+ dataset as shown in Figture 7 and Figture 8 respectively.

B. Accuracy results

We found that the accuracy of the test set using the model in approach 2 was also far greater than the accuracy of the test set using the model in approach 1 on both datasets. The accuracy values of the training set for Yale B dataset using

TABLE III
THE RESULT OF CROSS-ENTROPY, ACCURACY AND COMPUTATIONAL TIMES FOR DIFFERENT DATASETS AND CLASSIFICATION METHODS.

Dataset	Model	Training Set		Testing Set		Computational Times (s)
		Cross-Entropy	Accuracy (%)	Cross-Entropy	Accuracy (%)	
Yale	MLP	15.63	3.00	15.68	2.70	15870^{c}
	CNN + MLP	0.135	95.76	0.126	96.56	957^{a}+251^{b}+3770^{c} = 4978
CK+	MLP	15.98	0.88	15.79	2.05	28709^{c}
	CNN + MLP	0.022	99.27	0.001	99.93	2305^{a}+784^{b}+2581^{c} = 5670

[a]This is the computational time of Feature Extraction of *Training Dataset*.
[b]This is the computational time of Feature Extraction of *Testing Dataset*.
[c]This is the computational time of *Model Training*.

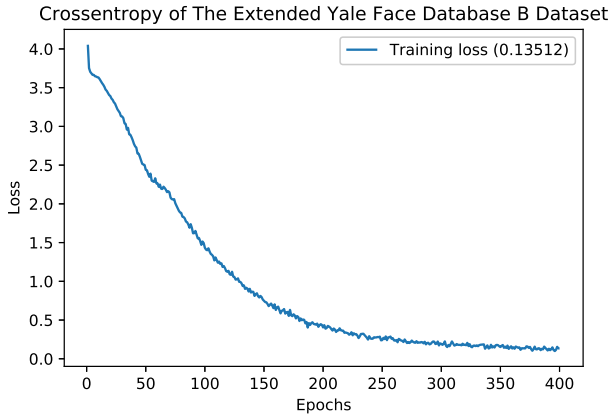

Fig. 7. The loss values on the training set of the Extended Yale Face Database B.

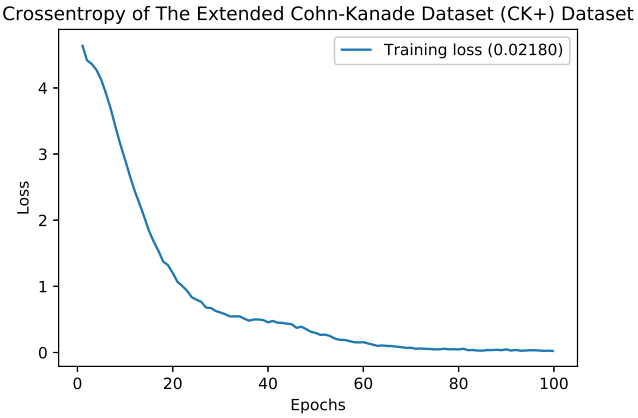

Fig. 8. The result of model loss on the training of the Extended Cohn-Kanade Dataset (CK+) .

the model in approach 2 was higher than 80% after training over 200 epochs as shown in Figure 9. This model needs only 30 epochs for the CK+ dataset as shown in Figure 10.

The accuracy for the training set matches the accuracy for the test set on both datasets. This implies that the model does not overfit the training set as it can predict the unseen data with high accuracy.

C. The computational times results

The total computation time of approach 1 has only one value, that is the model training time. The total computation time of approach 2 has three values: 1.) the computational time of feature extraction for training dataset, 2.) the computational time of feature extraction of testing dataset, and 3.) the computational time of model training. Although the models in approach 2 have many time values, the total time values are less than the one in approach 1.

V. CONCLUSIONS AND FUTURE WORK

Transferred learning using pre-trained convolution neural network as feature extractor has been applied to face recognition. Two public well-known face databases were used for evaluating the performance of the transferred learning model and traditional model. Although there are many classes, they consist of a small sample size per class in the face recognition problem. Based on the experiments, we can conclude that the results of face recognition that use transferred deep feature learning method are satisfactory. The performance of the transferred learning model is far better than the traditional model. In addition, there is no sign that an overfitting problem exists on both datasets.

In future work, we plan to fine-tune the existing pre-trained model and experiment with other pre-trained models. Furthermore, the original CK+ dataset that has non-facial parts

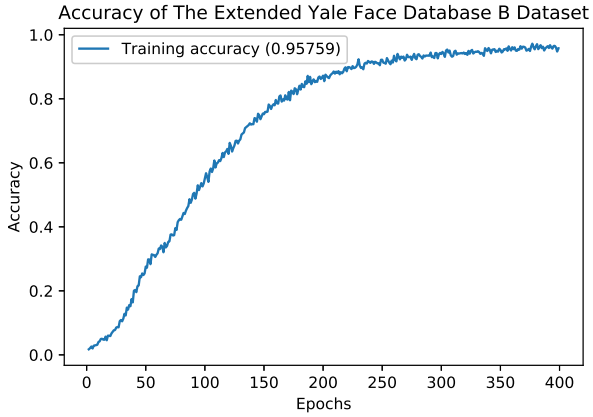

Fig. 9. The accuracy on training set of the Extended Yale Face Database B.

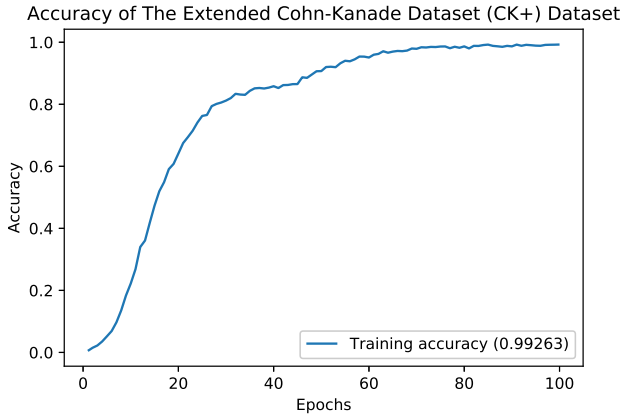

Fig. 10. The accuracy on training set of the Extended Cohn-Kanade Dataset (CK+).

and includes the ears, head or hat, part of the neck, and part of the background image. However, these elements are not required for classification and must be removed and only the portion containing eyes, nose, and mouth should be used.

ACKNOWLEDGEMENT

The authors would like to express the appreciation for King Mongkut's The University of Technology Thonburi for useful ideas, and concrete advice. The School of Information Technology (SIT) Infrastructure, that supports the virtual machine on mainframe for the experiment in this research.

REFERENCES

[1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, Dec. 2003. [Online]. Available: http://doi.acm.org/10.1145/954339.954342

[2] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: a survey," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 705–741, May 1995.

[3] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, Jan 2002.

[4] V. Radhamani and G. Dalin, "A supporting survey to step into a novel approach for providing automated emotion recognition service in mobile phones," in *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, Jan 2018, pp. 35–39.

[5] T. Liu, S. Fang, Y. Zhao, P. Wang, and J. Zhang, "Implementation of training convolutional neural networks," *CoRR*, vol. abs/1506.01195, 2015. [Online]. Available: http://arxiv.org/abs/1506.01195

[6] M. Y. W. Teow, "Understanding convolutional neural networks using a minimal model for handwritten digit recognition," in *2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, Oct 2017, pp. 167–172.

[7] J. R. C. P. de Oliveira and R. A. F. Romero, "Transfer learning based model for classification of cocoa pods," in *2018 International Joint Conference on Neural Networks (IJCNN)*, July 2018, pp. 1–6.

[8] G. Gautam and S. Mukhopadhyay, "Contact lens detection using transfer learning with deep representations," in *2018 International Joint Conference on Neural Networks (IJCNN)*, July 2018, pp. 1–8.

[9] Q. Li, L. Mou, K. Jiang, Q. Liu, Y. Wang, and X. X. Zhu, "Hierarchical region based convolution neural network for multiscale object detection in remote sensing images," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, July 2018, pp. 4355–4358.

[10] S. D., "Understanding convolutional neural networks," *InSeminar Report, Informatik und Naturwissenschaften Lehr-und Forschungsgebiet Informatik VIII Computer Vision.*, 2014.

[11] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. on Knowl. and Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2009.191

[12] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019–1034, May 2015.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[14] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng, "Building high-level features using large scale unsupervised learning," in *International Conference in Machine Learning*, 2012.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[19] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[20] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, June 2010, pp. 94–101.

[21] F. Chollet *et al.*, "Keras," https://keras.io, 2015.