Classification of Hand-Drawn Basic Circuit Components Using Convolutional Neural Networks

Mihriban Günay
Dept. of Electrical and Electronics Engineering
Munzur University
Tunceli, Turkey
e-mail: mihribangunay@munzur.edu.tr

Murat Köseoğlu
Dept. of Electrical and Electronics Engineering
Inonu University
Malatya, Turkey
e-mail: murat.koseoglu@inonu.edu.tr

Özal Yıldırım
Dept. of Computer Engineering
Munzur University
Tunceli, Turkey
e-mail: oyildirim@munzur.edu.tr

***Abstract*—In this paper, the Convolutional Neural Network (CNN) architecture, which is one of the deep learning architectures, is used to classify the basic circuit components drawn by hand. During the training and testing stages of the model, a new dataset containing images of 863 circuit components manually drawn by different people is created. The data set contains images of four different classes of circuit components such as resistor, inductor, capacitor and voltage source. All images have been fixed to the same size and converted to grayscale to increase recognition performance and reduce process complexity. In the study, training for four classes is performed with CNN architecture. Based on the CNN architecture, four new CNN models are employed with different the number of layers. The training and validation results of these models are compared separately, the model with the highest training and validation performance is observed with four layer CNN model (CNN-4). This model obtained 84.41% accuracy rate at classification task.**

Keywords-Deep learning; classification, CNN, circuit components.

I. Introduction

Nowadays, with the spread and development of technology, computers are actively used in all areas of engineering. The computers used in the field of Electrical and Electronics Engineering contribute to solving the problems in issues such as circuit drawing, simulation and coding. Although the use of computers is widespread, when electrical and electronic engineers and students studying in this field want to design a circuit, they firstly draw the circuit diagram on a paper. Hence, in Electrical Electronics Engineering; sketches are both easier and more common in the courses of circuit design, circuit analysis, electronics and the other applied courses. In these courses, students draw circuit diagrams manually on paper and perform certain operations and calculations on it. There can be many different circuit components in the circuit diagrams drawn on the paper. Each circuit component has different tasks in the circuit. Depending on the problem to be solved, the circuit is designed by selecting the appropriate circuit components.

It is easy for Electrical and Electronic Engineers to recognize the components in a circuit when they see any hand-drawn circuit diagram, as they are familiar with the circuit components.

However, it is quite difficult for a special software in the field of electrical and electronic engineering to recognize a hand-drawn circuit diagram and components and perform various analyzes directly on the scheme.

For this purpose, there are some studies conducted for computers to recognize and classify hand-drawn electronic circuit diagrams and electronic circuit components.

In [1], hand-drawn circuit diagrams on paper are divided into certain sections and each circuit component is classified by the support vector machines (SVM). In the study [6], hand-drawn circuits with a total of 26 circuit components were tested and 25 components were classified correctly. An accuracy rate over 90% was obtained for each circuit component. In [2], the histogram (HOG) method of directed gradient properties is proposed to recognize hand-drawn electronic components. After the recognition of the electronic components is completed, these components are classified using the SVM method. In [3], to recognize AND, OR, NOT logic gates on the hand-drawn digital logic circuit, at first segmentation process was applied to the circuit drawn, then SVM was used to classify logic gates. In [4], the authors used the finite state machine method to find the type of hand-drawn circuit. Then they used the SVM to classify each electronic component in this circuit. In [5], the authors used the k-nearest neighbor (KNN) method and classification technique to identify components in hand-drawn electrical circuits. In [6], the authors made use of the image deformation model (IDM), which is the feature extraction method, to recognize the symbols in various categories including hand-drawn circuit symbols and many visual features allowing the recognition of symbols in different categories. In another study [7], after preprocessing the scanned images of hand-drawn circuit diagrams, the circuit diagrams were divided into certain sections and the recognition and assignment of nine different electronic components to the classes were carried out by the node recognition method.

In the latest study on hand-drawn electronic component recognition in the literature [8], using the convolutional neural network (CNN) architecture, which is one of the deep learning algorithms, has achieved a high success rate in the classification process, taking into account three classes such as diode, resistance and capacitor.

The aim of this study is to train the images in the dataset using the CNN architecture, which is one of the popular research topics, and to determine successfully which image belongs to which electronic circuit component class. The

978-1-7281-9352-6/20/$31.00 ©2020 IEEE

specific aspects of the study are the using of a new data set created by the authors in the study. The high number of classes, the classification of relatively different circuit components, the design and use of four CNN models with different layer numbers.

The content of this study is as follows: Section II presents general information about deep learning and the CNN architecture. In Section III, the used method, data set, pre processing steps and CNN model are described. In Section IV, training and accuracy performances for four different CNN models are compared. In addition, the confusion matrix of the most successful model is calculated. Finally, in Section V, the basic results of the study are given.

II. Deep Learning

Deep learning is currently the most sought-after field of machine learning and still improving. It is a common intersection point among important research topics such as neural networks (NN), artificial intelligence, graphic modeling, optimization, pattern recognition and signal processing [9]. The basis of deep learning is based on artificial neural networks which consist of the input layer, the output layer and the hidden layer. In feed forward neural networks, the number of hidden layers is at most three. In Fig. 1, the model of feed forward neural network consisting of input layer, output layer and one hidden layer is shown. In classification with classical machine learning techniques, feature vector extraction steps are employed by expert engineers, while in deep learning methods feature vector extraction steps are performed by computers with high processor power [10].

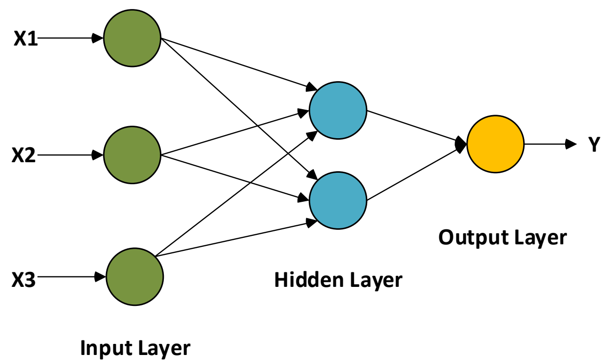


Fig. 1. Feed forward neural network model.

Deep neural network (DNN) is an artificial neural network with multiple layers between input and output layers [11]. A large number of hidden layers between the input layer and the output layer allowed the development of the concept of deep learning. In Fig. 2, the DNN model consisting of the input layer, the output layer and many hidden layers is shown.

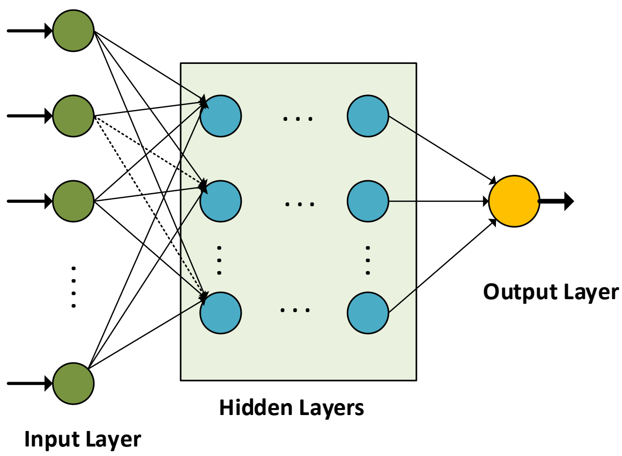


Fig. 2. An illustration of Deep neural network model (DNN).

A. Convolutional Neural Networks (CNN)

The concept of CNN was introduced by Kunihiko Fukushima in 1980 [12]. The first CNN model was created in 1988 to recognize handwritten digit by Yann LeCun [13]. This architecture, called LeNet, has achieved successful results [13]. In the early 1990s, CNN was used for speech recognition [14]. Nowadays, CNN has been successfully applied to handwritten characters [15,16], face recognition [17–19], behavior recognition [20], speech recognition [21] and image classification [22–24].

III. Proposed Method

In this study, different CNN models have been developed to recognize and classify images of hand-drawn circuit components with high accuracy. Image classification method with the proposed CNN model consists of five main steps. These steps are shown in Fig. 3.

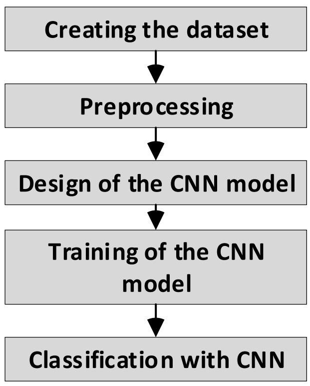


Fig. 3. The steps of the proposed method.

As can be seen from Fig. 3, the first step is the creating of a dataset and the second step is the preprocessing in the dataset. In the next steps, the CNN model is used to classify images. The design of the model, training of the model and classification processes are employed respectively.

In artificial neural networks and deep learning applications, computers with Graphics Processing Units (GPU) are usually used to complete training and testing processes more quickly. Many computers do not have GPUs with sufficient processing power for complex and heavy tasks. For computers with insufficient graphics cards, the Google company created the cloud service called Google Colaboratory (Colab) [25], which provides the opportunity to use the free NVIDIA Tesla K80 GPU graphics card over the Internet. Colab cloud service supporting Python language; It can be used for Keras, TensorFlow, OpenCV and many other libraries without requiring any additional installation. In this study, the Google Colab enviroment is preferred since it is needed a powerful GPU card during the training and testing stages with the CNN model.

A. Dataset and Preprocessing

In this study, a new data set is created from the images of electronic circuit components hand-drawn by the students of the Faculty of Engineering of Munzur University and this data set is used in all processes. Sample images in the data set are given in Table I. Two separate sets are created for the images of hand-drawn electronic circuit components, which many people draw on papers in different styles, taken with the help of cameras. The first one of these set is the training dataset set, and the second one is the test set. In the study, four different sub-sets are created in each training dataset since four different components are classified as resistance, inductor, capacitor and voltage source. These sets are the resistance, inductor, capacitor and DC voltage source sets. The same process is repeated by creating four separate sets in the test data set. In this way, the collected image data of each class is arranged in separate sets to prevent complexity. There are totally 709 images in the training data set. There are 154 images in the test data set. Detailed information about the Data Set is given in Table II.

TABLE I. SAMPLE IMAGES ON THE DATASET

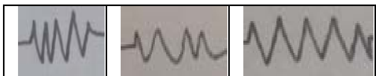
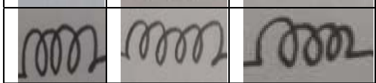
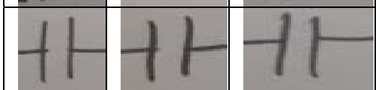
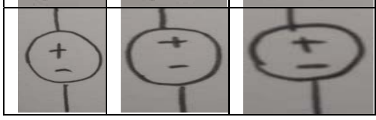
	Resistor samples
	Inductor samples
	Capasitor samples
	DC Voltage Source samples

Image preprocessing can be thought of as a preparatory step that should be done before training the model. The purpose of preprocessing is a set of regulations related to images such as improving the image properties, changing the image size and color properties of the image.

Firstly, since the images of the hand-drawn circuit components are of different sizes, the image data has been resized as 150 * 150. After all the images in the data set are reduced to the same size, they are converted to grayscale. The main reason for converting color images to grayscale is that color information is not required to recognize and interpret an image. As color images are represented by three channels, they contain more information than grayscale images, creating unnecessary complexity in the processor memory and taking up more memory. All images in the dataset have been converted to grayscale in order to reduce the computational complexity on the processor, reduce the calculation area and save time.

TABLE II. DETAILS OF THE DATASET

Files in the dataset	The size of the images	Number of images in the train dataset	Number of images in the test dataset
Resistor	150*150	282	32
Inductor	150*150	170	65
Capasitor	150*150	152	33
DC Voltage Source	150*150	105	24

B. The Proposed CNN Model

During the CNN model design process, four different new CNN models are developed. The model given in Fig. 4 is chosen among the models with different layer numbers. This preferred CNN model has 12 layers. The model is designed with five convolution layers, five pooling layers, one fully connected layer (FC), one softmax layer and ReLu activation function.

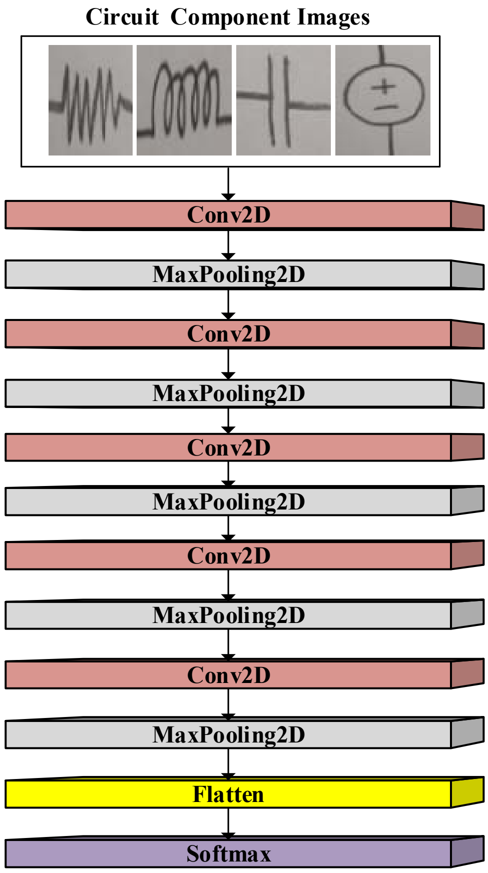


Fig. 4. Architecture of the proposed CNN-4 model.

In the convolution layer, the filter is selected according to the size of the image data. The filter selected in 3 * 3 dimensions is shifted from left to right and top to bottom on each image data, and the numerical values in the filter are multiplied by the pixel values in the image, respectively. The values obtained from the multiplication process are summed and the result is divided by the total number of values in the filter. The obtained value is written on the newly formed image. The size of the newly formed image after the convolution process is smaller than the size of the raw image. In the MaxPooling layer, an empty filter is selected and its size is determined. The empty filter selected in 2 * 2 dimensions is moved from left to right and from top to bottom on the image, the largest value is taken from the 4 remaining values in the filter circulating on the image, and the new image is written on it. In the Flatten layer, image data in matrix format is converted into a flat array format. In the last layer, dense layer (fully connected layer), softmax function expresses which image belongs to which class with a probability value between 0 and 1.

IV. Experimental Results And Analysis

Four CNN models with different layer numbers were trained for 100 epochs using the data set, created by the authors for the study, containing 863 images.

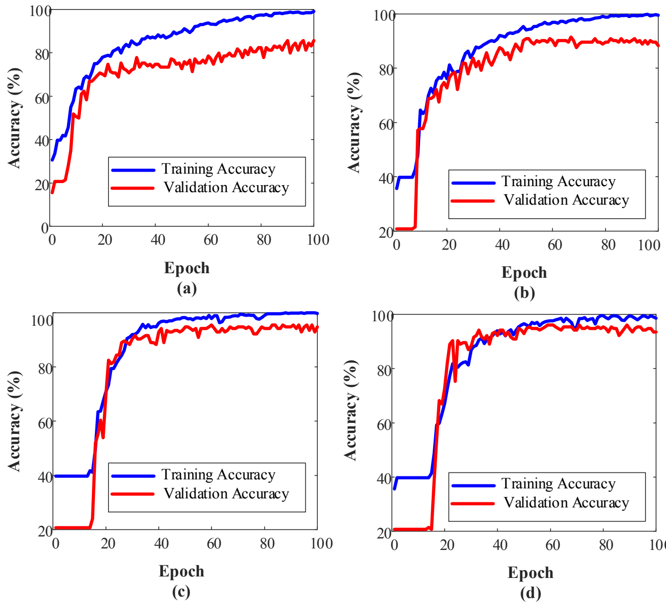


Fig. 5. Training and validation accuracy graphs for a) CNN-1, b) CNN-2, c) CNN-3 and d) CNN-4.

The accuracy graph of CNN-1, model designed with two convolution layers, two maxpooling layers, one fully connected layer and one softmax layer is shown in Fig. 5 (a) while the loss value graph is shown in Fig. 6 (a). The accuracy graph of CNN-2 model designed with three convolution layers, three maxpooling layers, one FC layer and one softmax layer is shown in Fig. 5 (b) while the loss value graph is shown in Fig. 6 (b). The accuracy graph of CNN-3 model designed with four convolution layers, four maxpooling layers, one FC layer and one softmax layer is shown in Fig. 5 (c) while the loss value graph is shown in Fig. 6 (c). The accuracy graph of CNN-4 model designed with five convolution layers, five maxpooling layers, one FC layer and one softmax layer is shown in Fig. 5 (d) while the loss value graph is shown in Fig. 6 (d). When Fig. 5 is examined, the training performance of the graphs in Fig. 5 (c) and Fig. 5 (d) is higher than the other two graphs. The training performances of the graphs in Fig. 5 (c) and Fig. 5 (d) are similar. Considering the validation accuracy, the validation performance in Fig. 5 (d) is higher. When the loss value graphs in Fig. 6 are examined, the validation loss in Fig. 6 (d) is the lowest compared to the other three graphs.

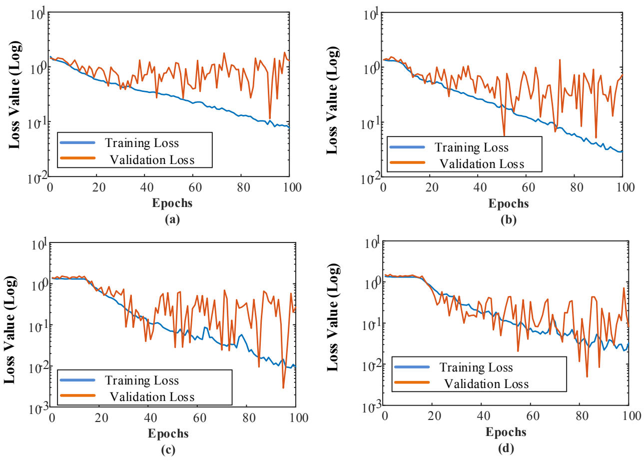


Fig. 6. Loss value graph for a) CNN-1, b) CNN-2, c) CNN-3 and d) CNN-4.

The challenge of over-fitting is when the model performs well during the training phase and performs poorly during the validation phase. As seen from the figures, there is no over fitting problem in the performance graphs of the models. Accuracy rates of CNN-1, CNN-2, CNN-3 and CNN-4 models in the last epoch are 85.71%, 88.31%, 94.81% and 93.51%, respectively. The accuracy rate of the CNN-3 model in the last epoch is higher than that of the CNN-4 model. But the accuracy rate in the last epoch does not exactly show the success of the model. Therefore, when both accuracy graphs and loss value graphs are examined in detail the most successful model is CNN-4. In order to clearly observe this difference, the performance graph on the validation sets of all models is given in Fig.7.

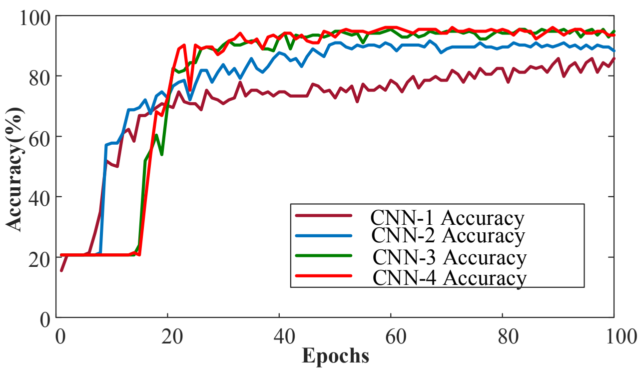


Fig. 7. Graph showing the accuracy rates on the validation set of all CNN models.

As seen in Fig. 7, the validation accuracy performance of the CNN-4 model with the highest number of layers is at the highest rate compared to the other 3 models.

Accuracy: 84.416%

		Predicted Class			
Actual Class		Inductor	Resistor	Capasitor	DC Voltage Source
	Inductor	100.0% 56	11.1% 4	12.2% 5	0.0% 0
	Resistor	0.0% 0	72.2% 26	14.6% 6	0.0% 0
	Capasitor	0.0% 0	16.7% 6	65.9% 27	0.0% 0
	DC Voltage Source	0.0% 0	0.0% 0	7.3% 3	100.0% 21

Fig. 8. Confusion matrix of the CNN-4 model.

The confusion matrix showing the success of the CNN-4 model, which performs best during the training phase, in the classification for each class is given in Fig. 8. Accordingly, the overall success of the CNN-4 model in the classification is 84.41%. As seen from the confusion matrix, the model's success rate of classifying the inductor and DC voltage source is 100%. The resistance classification success rate of the model is 72.2% and the capacitor classification success rate is 65.9%.

V. CONCLUSIONS

In this study, 4 different CNN models are trained to put the resistor, inductor, capacitor and DC voltage source images into the correct classes. These models were tested on the data set created by the authors and the results were examined. Considering the training and validation performances, it was seen that the most successful model among four models was the CNN-4 model and the most unsuccessful model was the CNN-1 model. When the graphics are evaluated, increasing the number of layers in CNN models has a positive effect on performance. In order to increase the performance of the CNN-4 model, which has high training and validation performance, the number of data can be increased and some parameters in the model can be changed to obtain better results.

REFERENCES

[1] Y. Liu and Y. Xiao, "Circuit Sketch Recognition," *Dep. Electr. Eng. Stanford Univ. Stanford, CA*, 2013.

[2] R. Lakshman Naika, R. Dinesh, and Santoshnaik, "Handwritten electronic components recognition: An approach based on HOG + SVM," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 13, pp. 4020–4028, 2018.

[3] M. D. Patare and M. S. Joshi, "Hand-drawn Digital Logic Circuit Component Recognition using SVM," *Int. J. Comput. Appl.*, vol. 143, no. 3, pp. 24–28, 2016.

[4] R. Lakshman Naika, R. Dinesh, and S. Prabhanjan, "Handwritten electric circuit diagram recognition: An approach based on finite state machine," *Int. J. Mach. Learn. Comput.*, vol. 9, no. 3, pp. 374–380, 2019, doi: 10.18178/ijmlc.2019.9.3.813.

[5] A. Dewangan and A. Dhole, "KNN based Hand Drawn Electrical Circuit Recognition," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 6, no. 6, pp. 1111–1115, 2018, doi: 10.22214/ijraset.2018.6164.

[6] T. Y. Ouyang, R. Davis, and C. Science, "A Visual Approach to Sketched Symbol Recognition," *Proc. Int. Jt. Conf. Artif. Intell.*, pp. 1463–1468, 2009.

[7] B. Edwards and V. Chandran, "Machine recognition of hand-drawn circuit diagrams," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 6, pp. 3618–3621, 2000, doi: 10.1109/ICASSP.2000.860185.

[8] H. Wang, T. Pan, and M. K. Ahsan, "Hand-drawn electronic component recognition using deep learning algorithm," *Int. J. Comput. Appl. Technol.*, vol. 62, no. 1, pp. 13–19, 2020, doi: 10.1504/IJCAT.2020.103905.

[9] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2013, doi: 10.1561/2000000039.

[10] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[11] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015, doi: 10.1016/j.neunet.2014.09.003.

[12] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980, doi: 10.1007/BF00344251.

[13] Y. Le Cun *et al.*, "Handwritten Digit Recognition: Applications of Neural Network Chips and Automatic Learning," *IEEE Commun. Mag.*, vol. 27, no. 11, pp. 41–46, 1989, doi: 10.1109/35.41400.

[14] *Readings in Speech Recognition*. 1990.

[15] L. Chen, S. Wang, W. Fan, J. Sun, and S. Naoi, "Beyond human recognition: A CNN-based framework for handwritten character recognition," *Proc. - 3rd IAPR Asian Conf. Pattern Recognition, ACPR 2015*, pp. 695–699, 2016, doi: 10.1109/ACPR.2015.7486592.

[16] D. Sen Maitra, U. Bhattacharya, and S. K. Parui, "CNN based common approach to handwritten character recognition of multiple scripts," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 2015-Novem, pp. 1021–1025, 2015, doi: 10.1109/ICDAR.2015.7333916.

[17] X. Yin and X. Liu, "Multi-Task Convolutional Neural Network for Pose-Invariant Face Recognition," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 964–975, 2018, doi: 10.1109/TIP.2017.2765830.

[18] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face Recognition: A Convolutional Neural-Network Approach," *IEEE Trans. NEURAL NETWORKS*, vol. 8, no. 1, pp. 98–113, 1997, doi: 10.1016/j.gene.2017.06.018.

[19] H. Khalajzadeh, M. Mansouri, and M. Teshnehlab, "Face Recognition Using Convolutional Neural Network and Simple Logistic Classifier," *Soft Comput. Ind. Appl. Springer Int. Publ.*, vol. 223, pp. 197–207, 2014, doi: 10.1007/978-3-319-00930-8.

[20] S. Yan, Y. Teng, J. S. Smith, and B. Zhang, "Driver behavior recognition based on deep convolutional neural networks," *2016 12th Int. Conf. Nat. Comput. Fuzzy Syst. Knowl. Discov. ICNC-FSKD 2016*, no. 1, pp. 636–641, 2016, doi: 10.1109/FSKD.2016.7603248.

[21] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Y. Yu, "Convolutional Neural Networks for Speech Recognition," *IEEE/ACM Trans. AUDIO, SPEECH, Lang. Process.*, vol. 22, no. 10, pp. 1533–1545, 2014, doi: 10.1109/ijcnn.1999.835942.

[22] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network," *2014 13th Int. Conf. Control Autom. Robot. Vision, ICARCV 2014*, vol. 2014, no. December, pp. 844–848, 2014, doi: 10.1109/ICARCV.2014.7064414.

[23] Y. Wei *et al.*, "HCP: A flexible CNN framework for multi-label image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1901–1907, 2015, doi: 10.1109/TPAMI.2015.2491929.

[24] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, 2017, doi: 10.1109/TGRS.2016.2612821.

[25] "Google Colaboratory." https://colab.research.google.com (accessed Jun. 13, 2020).