IJARIIT

INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume3, Issue5)
Available online at www.ijariit.com

Comparison Of Hog (Histogram of Oriented Gradients) and Haar Cascade Algorithms with a Convolutional Neural Network Based Face Detection Approaches

Emine Cengil
Computer Engineering Department
Firat University
ecengil@firat.edu.tr

Ahmet Cinars
Computer Engineering Department
Firat University
acinar@firat.edu.tr

***Abstract:** Face detection is an important computer vision problem that has been working for years. Security and market research are the areas where face detection is used. Face detection is the first step in some problems such as face recognition, age estimation, and face expression detection. Several face detection algorithms have been developed up to now. CNN-based algorithms are the state-of-the-art technology in image processing problems, as well as other methods in terms of accuracy rates and speed criteria in face detection problems. In this paper, we propose a face detection algorithm having a model like Alexnet. The method is implemented on images from MUCT and FDDB public datasets. In addition, in the study, the Hog feature descriptor and the methods developed with the haar cascade features are compared with the CNN based method using images of the same dataset. Tests show that the proposed method produces better results than the other two methods.*

***Keywords:** Deep Learning, caffe, Convolutional Neural Network, HOG, Haar Cascade.*

I. INTRODUCTION

The use of intelligent systems in all areas with the development of information technology in recent years has become inevitable. The concept of artificial intelligence begins with the appeal of the idea of teaching intelligence, the most precious thing possessed, that separates man from other living things, to machines. It has evolved through new techniques introduced in science circles. Deep learning, which has been mentioned a lot in recent times, has been widely used.

Deep learning is the general name of techniques developed at the point where traditional artificial neural networks are inadequate to solve the problem. It's named with a "deep" prefix, indicating the number of hidden layers is too high. Deep Learning algorithms consist of artificial neural network based and energy-based models [1]. The architecture comes in many layers and variants. Convolutional neural networks are a specialized architecture of deep learning and are particularly successful in classification. In this regard, CNN has been used and used in many academic studies.

Deep learning provides non-linear transformation of the data. Instead of shallow structures, it can model complex relations with gauss mixture models, hidden Markov models, conditional random fields, multilayer structure. The most commonly used deep learning architects are; Automatic Encoders, Restricted Boltzman Machines and Convolutional Neural Networks [2].

In recent times, almost in every area is used for deep learning which produces in fast and nearly error-free solutions. There are many reasons for the widespread use of deep learning. Other machine learning algorithms only classify by way of self-taught examples. The machine cannot recognize classes that are not taught and cannot make comparisons. Deep learning algorithms divide the entered samples into layers. Starting from the lowest layer, however, it attempts to identify the pattern to form a prototype.

Every successful new layer, which the deep learning algorithm abstracts in the previous layer, increases the recognition and classification power of the deep learning algorithm. This feature makes deep learning algorithms superior to other algorithms. It allows more recognition and classification with fewer examples taught in the algorithm.

Deep learning does not require any pre-built feature extractor manually. The feature is extracted automatically [3]. Therefore, there is no need finding an answer to the question of which feature to learn for the data. The important question here is that how many layers of the network are better suited to solve the problem and which operations need to be applied. The face recognition problem attracts attention due to the necessity of using it especially in areas where safety is important. Although there are different ways of recognizing the person, such as fingerprint, eye recognition, iris recognition, face recognition is the most preferred because of its advantages [4].

In this study, the solution of facial finding problem is presented with the convolutional neural networks, which is a deep learning algorithm. Using the images called AFLW [5] (Annotated Faces Landmarks in the Wild)", the network model was trained and classified as a face, not face. Subsequently, testing has performed using data from FDDB [6], MUCT [7] and Google.com that were not included in the training set, and the results were reported. The same images are also run with applications in the python environment with Haaarcascade-like features and HOG attribute descriptors, and the results are compared. It is seen that the developed model is better than the other two methods in terms of speed and accuracy rate.

The structure of this article is as follows; Chapter 2 defines related work. Chapter 3 contains theoretical information about the convolutional neural networks, which is the architecture of deep learning. Section 4 gives the proposed method. Section 5 shows the experimental results at finally section 6 presents the conclusions.

II.RELATED WORKS

From past to present, many methods have been developed to deal with the problem of face detection. The most used of academic and scientific studies to solve the problem are; Skin color method, Haar classification algorithm and artificial intelligence algorithms (Artificial Neural Network, Fuzzy Logic, Deep Learning ...). Apart from these, face detection is also performed using SIFT and HOG feature descriptors.

D. Ghimire et al. [8] have provided an illumination-insensitive face detection method based on edge and skin tone information of the input color image. Image segmentation is enhanced after development using skin tone and edge information to separate the face components from the background. The connected components are analysed the primary shape properties and the standard deviation of the candidate face region in the respective grayscale image of the edge and bound image.

Sanjay et al. [9] have examined the dynamics of different color models in a database of five videotapes. These videos contain more than 93,000 hand-drawn face images. In addition, Adaboost has proposed an adaptable skin color model to reduce the false acceptance of the face detector. Since the face color distribution model is updated regularly using the previous Adaboost replies, we have found the system more effective for environmental variables in the real world.

Q. Lan et al. [10] have developed traditional Adaboost face detection algorithm, Haar-like features, use training classifiers with a low detection error rate in the face region. In this study, we propose Adaboost face detection algorithm based on YCgCr skin color model combined with the Adaboost algorithm and skin color detection algorithm. S. Kang et al. [11] have proposed a method to increase the speed of a sliding window type face detector by perceiving the skin color region. S. Zhou, et al. [12] have proposed a four-layer face finder that combines tree structure classifiers with Support Vector Machine, introducing new Multi-Block Local Gradient Patterns (MB-LGP) as feature set. J. Ruan et al. [13] have proposed a facial recognition algorithm based on facial features and LSVM. Face candidate regions are identified by detecting eyes and mouth.

X. Zhang et al. [14] have proposed an effective face detection method based on Two-Dimensional Basic Component Analysis (PCA) combined with Support Vector Machine (SVM). H. M. El-Bakry et al. [15] have introduced a faster PCA approach to describe human faces in a given image. Such an approach has divided the input image into very small sub-images. Y. Li et al. [16] provide human face detection, identification, and tracking techniques and techniques used for a human-robot interaction system. With a standard image processing algorithm, a blurred skin tint adjuster is recommended to detect human faces and then identify them with nonlinear support vector machine (SVM) and Euclidean distance measurement. H. C. Yang et al. [17] have proposed a cascaded face detection method based on the histograms of oriented gradients (HOG), using different features and classifiers to step-out the non-faces. The candidate feature set has created with the HOG feature of different piece size; Support Vector Machine (SVM) was used at different stages as a weak parameter classifier with different parameters.

V. Jones et al. [18] have used haar classifiers to find faces. They have presented a new image presentation called "Integral Image" which allows to quickly calculate the properties used by the detector. J. Coster et al. [19] investigated whether it is possible to measure one's attention in a controlled environment using the OpenCV programming library and the Viola-Jones algorithm.

P. Irgens et al. [20] have presented a Viola-Jones face detection algorithm based on Field Programmable Gate Arrays (FPGAs). X. Zhao et al. [21] proposed a new context modeling method for facial mark detection that integrates context constraints with the local texture model in the stepped AdaBoost framework. Numerous face finding methods developed using artificial neural networks [22, 23, 24, 25] and fuzzy logic [26, 27, 28] are available in the literature. But in recent years, deep architectures have begun to take knowledge of these methods.

Shuo Yang et al. [29] have proposed a deep learning approach by taking advantage of the facial part answers for face finding. In this study, a DNN (Deep Neural Network) design was used for face detection. Faceness-Net is the method of acquiring the facial approach and facial approach. Producing partners maps, sorting candidate windows with faceness scores and refining facial suggestions for face finding. In this study, which carried out face classification and limiting box stretching together, a network containing 7 convolutional and 2 max-pooling layers were used. The study provided a recall of 90.95% in datasets such as FDDB, AFW, and PASCAL face. H. Qin et al. [30] have to recommend step-CNN joint training for face finding. In this study, coupled training for the end-to-end optimization step for the CNN step was proposed.

K. H. Kong et al. [31] have used S-LGP (Symmetry Uniform Local Degrees) and U-LGP (Uniform Local Degrees), which improved symmetry and unchanging by using LGP (Local Degree Descriptor) against external factors such as lighting changes, Pattern) methods. Face expression, background, etc. LBP (Local Binary Pattern) analyzes the sampling and negative face sample in 1: 2 ratio using the input image and the skin color extraction method. Using the step classifier extractor to generate the xml files without collecting and examining the positive face in YCbCr space and uses the CNN deep learning algorithm.

X. Sun et al. [32] have proposed a new method for face detection using deep learning techniques. In particular, it expands the RCNN framework used for generic object detection and features aggregation, multi-scale training, constant negative mining, and so on. , Proposed several strategies to improve the faster RCNN algorithm to solve face recognition tasks, including.

K. Zhang et al. [33] have proposed a deep stepped multitasking framework that uses the natural relationship between sensing and alignment to enhance face finding performance in their work. The frame activates a three-stage hierarchical architecture of carefully designed deep folding nets to predict face and bookmark position in a gradual and precise manner.

In this paper, two existing methods for face detection are compared with the proposed method. The later is a method developed by HOG (Histogram of oriented gradients), which is a property identifier used to detect objects in the field of computer vision and image processing. The main purpose of the HOG method is to identify the image as a group of local histograms. These groups are the histograms of the magnitudes of the gradients in the orientation of the gradients in a local region of the image [34]. The HOG descriptive technique counts the occurrences of gradient guidance in localized portions of a view detection window or area of interest [35].

Another algorithm we compare is the HaarCascade object detection algorithm, which is mainly used for face detection, to find faces, pedestrians, and objects in an image. The Haar cascade system provides several different positive and negative images. Feature selection is done with classifier training using adaboost and integral images [18]. The developed method offers more durable solutions against the factors such as face closure, exposure change, light effect.

III. CONVOLUTIONAL NEURAL NETWORK

The convolutional networks spread from the local receptive field of the visual cortex of the monkey. When monkeys show familiar faces, their brains burn in a certain area. The recipient cells in the monkey visual system are sensitive to the small subregions of the visual field called the "receptive field". However, in convolutional networks, the local receptive fields in the image are connected to the individual neurons in the first hidden layers. CNN's use spatially local interest among neighboring neurons. They learn layers using a local connection model. Thus, the learned filters spatially give the strongest response to the local input pattern [36].

Convolutional neural networks seem to be a combination of biology and mathematics, but these networks are some of the most effective innovations in computer vision. Alex Krizhevsky has been used convolutional neural networks to win the 2012 ImageNet competition, ImageNet, and reduced the classification error from 26% to 15% [37]. Since then, deep learning has attracted the attention of many companies and a number of companies have begun to use deep learning on the basis of their services.

CNN is feeding forward and is a very effective method of finding. The structure is simple; less training parameters and adaptability. Its weight-sharing network structure made it more similar to biological neural networks. This reduces the complexity and weight of the network model. CNN is used in many fields such as signal processing, natural language processing, robotics and sound processing in science and academia. But the most popular area is image processing and pattern finding problems.

Convolutional neural networks are very similar to ordinary neural networks. It consists of neurons with learnable weights and biases. Each neuron takes some inputs, generates a point product, and optionally follows it nonlinearly. The whole network expresses a differentiable scoring function. As it is the case with known neural networks, the convolutional network of neurons contains a loss function, such as softmax, in the final layer [38].

Convolutional networks differ from neural networks in terms of the structure and function of layers. In ordinary neural networks, the layers are one-dimensional and the neurons in the layer are completely connected. On the other hand, the format of CNN layers is usually three-dimensioned whose parameters are the width, height, and depth [39].

In CNN architecture, every neuron of the hidden layer is only connected to the local region of the scene. This area in the input image is called as the "local receptive field". Weight and bias are shared in open areas. All the weights and bias in the first hidden layer are connected by the same shape but different local area pixels.

Convolutional neural networks can be more than one dimension. The one-dimensional network can be used for voice data, and the three-dimensional network can be used if necessary. The neural network will select a kernel (filters) that perceive certain features (ie, shapes). Like neural networks, you can place convolutive nets in layers; Where filters in deeper layers may detect more complex features.

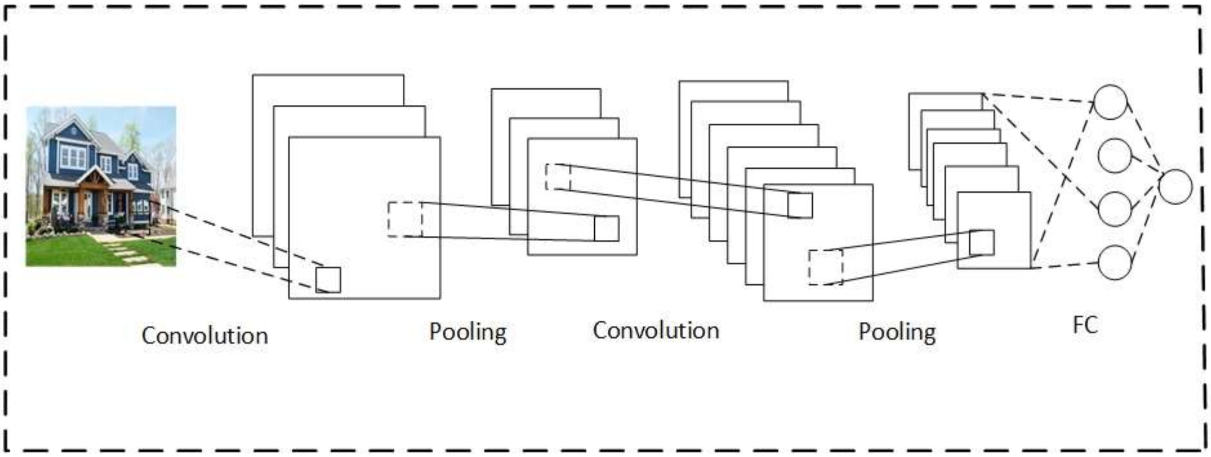


Fig.1. Model of a convolutional neural network with convolution, pooling, and fully connected layers

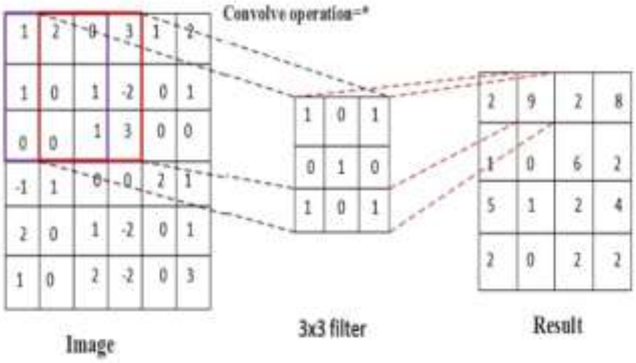
In Fig. 1, a simple convolutional neural network model is given, which includes pooling, convolution, and fully connected layer. CNN's are often made up of many parts. These parts are; Convolution, activation, pooling or subsampling and fully connected layers.

A. Convolution Layer

The primary purpose of the convolution layer is to extract features from the input image. The matrix of convolution is the application of a matrix to another matrix called the "kernel". The convolution matrix filter uses the image to be processed first. Convolution protects the spatial relationship between pixels by learning image properties using small squares of the input data. When convolving, 5x5 or 3x3 matrices are usually used and sufficient for all desired effects.

Fig.2 shows that how to operate the convolution process. In CNN terminology, a 3x3 matrix is called a 'filter' or 'kernel', and a matrix created by scrolling the filter on the image is called a 'Feature Map'. A 6x6 image is passed through 1 stride of 3x3 size filtration. The filtering result is reduced to 4x4 from the feature map (image size - filter size + step size = feature map size) created [40].

A CNN learns the values of the filters alone during the training process, but before the training process parameters like filter number, filter size, network architecture must be specified. As the number of filters, we have increased, more image features are extracted and patterns in images that do not show up in our network are better known. The activation process is followed by the layer of convolutional neural network structure.



1	2	0	3	1	2
1	0	1	-2	0	1
0	0	1	3	0	0
-1	1	0	0	2	1
2	0	1	-2	0	1
1	0	2	-2	0	3

1	0	1
0	1	0
1	0	1

2	9	2	8
1	0	6	2
5	1	2	4
2	0	2	2

Fig.2. Convolution operation

B. Activation Layer

After each convolution layer, it is the case that a non-linear layer (or activation layer) is applied immediately. The purpose of this layer is to introduce non-linearity into a system that computes linear operations during essentially convolutional layers[41]. In the activation layer, the function works on net entries coming into the cell and determines the value that the cell will produce for this cell. Various types of activation functions can be used in the cell models according to the performance of the cell. Activation functions can be selected as a fixed or adaptive parameter.

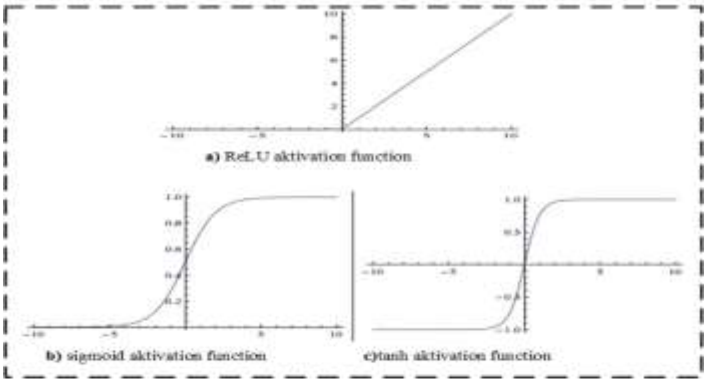


Fig.3. Activation functions most commonly used in the CNN [42]

Fig.3 depicts the most used activation functions. In the models of the neural networks, non-linear functions such as tanh, sigmoid have been used in the past, but researchers have discovered that the ReLu layers work better than others because the network can process faster due to computational efficiency. It also helps to alleviate the problem of the disappearance curve, which is the problem that the sublayers of the network proceed too slowly due to the collapsing of the slope through the layers. The ReLU layer implements the function f (x) = max (0, x) for all values in the input volume [43]. In basic terms, this layer changes all negative activations to 0 only. This layer enhances the nonlinear properties of the model and the public network without affecting the receiving domains of the convolution layer.

C. Pooling Layer

In the architecture of convolutional networks, it is common to add a pooling layer periodically between successive layers of convolution. In this section, there are also models where the sub-sampling layer replaces the pooling layer. The function is to gradually reduce the representative spatial dimension to reduce the amount of parameters and computations on the network and thus control overfitting.

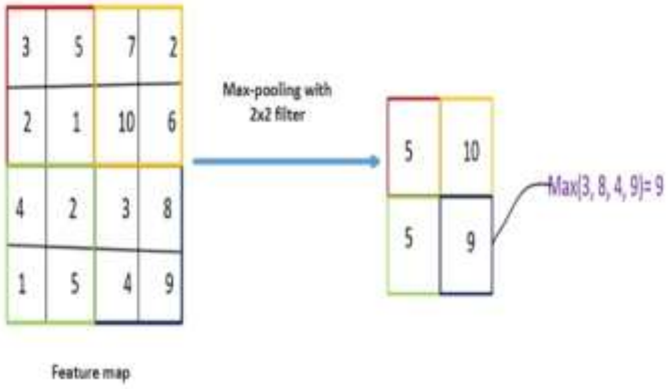


Fig.4. Max-pooling process with 2x2 filter and 2 stride

Fig.4 shows the maximum pooling operation. The Pooling Layer runs independently in each depth slice of the input and resizes it spatially using the maximum process. The most common form is a pool player with 2x2-dimensional filters applied with one step in 2 subsamples of each depth slice along the width and height. The depth dimension remains unchanged [44].

Assuming that the input image in the network model given in Fig. 1 is 64x64 when the 5x5 size filters are applied after the first convolution process, the new size becomes 60x60. The non-linear properties of the post-ReLU layer model and the public network are increased. The 60x60 image in the maximum pooling layer will be pooled with a 2x2 size filter so that the number of steps is 2, and the new size of the image will be reduced to 30x30 by half.

The same process is repeated on the first layers in the next convolution + ReLU + pooling layer. After the second pooling layer, the new size of the image becomes 14x14. The next layer is the fully connected layer where all the neurons are connected.

D. Fully-Connected Layer

The fully-connected layer is a multilayer perceptron that uses classifiers such as SoftMax and SVM in the output layer. The term "fully connected" means that each neuron in the previous layer is connected to each neuron in the next layer. It is very similar to multilayer perceptrons [44].

The output at the end of the convolution and pooling layers represents the high-level properties of the input image. There is no way of estimating the classification of the evolution and pool layers. The purpose of the Fully-connected layer is to use these properties to classify the input image into various classes based on the set of training data.

As seen in Fig. 1, there are 4 neurons in the fully connected layer. Each of these neurons represents a class. These; home, bicycle, car, truck. The network looks at the attributes extracted here to see which class the input data is closer to and returns a class as a result. A properly trained network will return home value.

IV. PROPOSED METHOD

In this part of the article, convolutional neural network based face detection approach is proposed. The training process is performed using images in the facial dataset AFLW (Annotated Faces Landmark Wild). IOT (Intersection over Union) metric which used to measure the performance of the HOG-Linear SVM detector and CNN detectors are used for data enhancement. Images with IoT values greater than 50% are accepted as faces and placed in the positive training data file, while the rest are discarded in the negative image file. The resulting images are resized to 227x227.

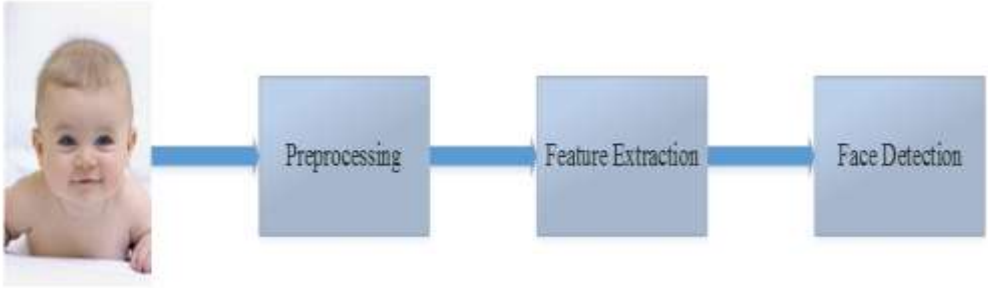


Fig.5. Pipeline of face detection algorithm

Fig. 5 gives a way to get a face detection approach. In order to extract the feature after the image has been passed through the preprocessing steps, the model Alexnet [45] is used with fine-tuning. By using a fine-tuned deep network model, the last face detector is obtained by the sliding window method.

A. Dataset

The features and size of the images used for training a model are important in that the learning rate is acceptable and accurate results can be obtained. A large number of images used for training affects the learning of the model positively.

The availability of many image sources on the Internet and the progress of large data technology allow the creation of many large data sets that are open for access. For the problem of finding faces, many methods are developed from the past. The methods must also be resistant to preventive factors such as obstruction, illumination, and exposure change. In addition to clean images taken from the front, there are many datasets that contain these challenges.

FDDB [46] (Face Detection Dataset and Benchmark): It has 2871 images and 5171 face images in total. Grayscale and color images in the dataset, congestion, closure, difficult exposures and low-resolution images are available.

LFW [47] (Labeled Faces in-the-Wild): It has 13,233 images with 5749 faces. 1680 images contain two or more faces. The only limitation of the commonly used dataset used to measure the accuracy of facial detection methods is that it can be detected by the Viola-Jones detector.

AFW [48] (Annotated Faces in-the-Wild): The dataset created by Zhuo et al. It Contains 468 faces with 250 images. Six hundred facial landmarks are provided for each face.

AFLW (The Annotated Faces Landmark in-the-wild): It has 24,866 faces with 25,993 images downloaded from Flickr. There is wide range of natural exposures. Face marker explanations are available for all data. It consists of 21 facial landmark points.

MUST [7]: The dataset, which contains 2755 people with 3755 images, has 76 landmarks. Images contain differences such as light, age, and race.

FERET [49] (The Facial Recognition Technology Database): It comes from 807 images with 256x 384 image size.

LFPW [50] (Labeled Face Parts in-the-wild): It contains 1,287 images downloaded from Google, flickr, and yahoo. The dataset comes from images that have a wide variety of pose, lighting and clogging effects.

HELEN [51]: It consists of 2330 annotated images downloaded from Flickr. Sometimes face sizes larger than 500x500 are available. The explanations provided are very detailed and have 194 hundred facial landmarks.

TABEL 1
Frequently Used Datasets For Face Detection Problem

Database	**Image number**	**Facial landmarks**	**Size**	**Color**
FDDB	2845 images 5171 face	-	-	Grayscale/Color
LFW	13.233 images 468 face	-	-	Color
AFW	250 images 468 faces	6	-	Color
AFLW	25.933 images	21	-	Color/grayscale
LFPW	1287 images	35	-	Color
HELEN	2.330 images	194	500x500$^+$	Color/grayscale
MUCT	3755 images 276 yüz	76	480x640	Color
FERET	807 images	-	256x384	Color

For the training of the proposed method, the AFLW dataset was used. The dataset containing 25.933 images does not give very accurate results with so many examples.

In order to increase the number of positive (face) data, we obtained two subsets of data, positive and negative, with floating windows using IoT metric. If the IoT ratio is greater than 50%, we considered the bottom window as a face.

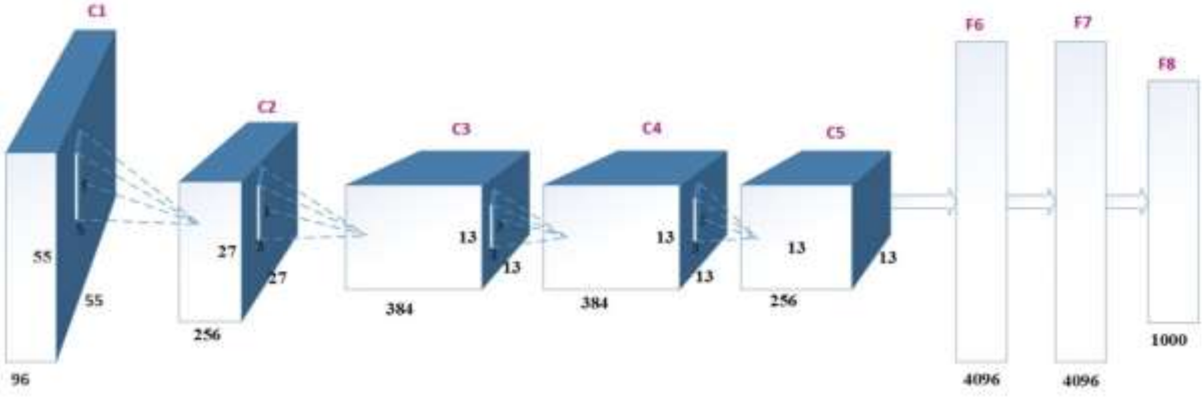


Fig.6. Architecture of AlexNet Model

We did not use all 25,933 images when creating the dataset. As the number of training samples increases, the accuracy of the system increases, but the number of training sessions is increased and the machine we are using does not have enough capacity to process millions of images. The number of positive samples was 56,798 and the number of negative samples was 243,470, and the negative/positive ratio was kept around 1/4.

B. Architecture of Model

Convolutional neural network architectures can be made in a few layers or as deep as in[37]. The architecture to be used is decided by trial and error or by adaptive methods. The fact that the number of layers is small in the architecture used does not usually lead to success in solving the problem. On the other hand, if the number of layers is too great, it will decrease the time efficiency because it increases the number of parameters. Therefore, it is important for the success of the method that the architect has the appropriate number of layers.

As the number of layers is important in network architecture, many values such as properties of used layers, size, and number of filters, activation function, learning parameters are important. Krizhevsky et al. the first Alexnet architecture to be used in the 2012 image competition was a huge success in image classification. Fig.6 gives the architecture of AlexNet model. In the proposed method, Alexnet model is used. Krishevsky's AlexNet is composed of 11 layers. The first two layers are the conv + max

+ norm, the third and fourth layers are convolution, the fifth layer is conv + max, the sixth and seventh layers are fully connected and the final layer is the softmax.

TABLE.2
FİNETUNİNG ALEXNET WİTH CAFFE

Finetuning AlexNet	**Layer**
Input (227x227)	0
conv (55x55)	conv1
Max-pool (27x27)	Pool1
conv(27x27)	Conv2
Max-pool (13x13)	Pool2
conv(13x13)	Conv3
conv(13x13)	Conv4
conv (13x13)	Conv5
Max-pool (6x6)	Pool5
FC (9216- 4096)	Fc6
FC (4096- 4096)	Fc7
FC (4096- 2)	**Fc8-Flickr**

The fine tuning takes a trained model, adapts to the architecture and allows the training to continue with the weight of the model already learned [52]. The AFLW dataset we use for training is Flickr-based and is very similar to the ImageNet dataset on which visually bvlc_reference_caffenet is trained. Since this model works well for classification of object categorization, we use this architecture for face detection. Imagine contains 1 million images. We want to connect many of these images with learned parameters and fine-tune them when necessary.

When we give the weight argument to the caffe [53] open source software, the pre-trained weights are loaded into our model and match the layers according to it. When fine-tuning with AlexNet, there are places in our model that we need to change. Images in ImageNet are classified according to 1000 categories, but we want to do binary classification, face or not face. Therefore, we need to change the classification layer, which is the last layer in the model. We also change the name of the fc8 layer in the prototxt file to fc8_flickr.Since there is no layer in this name in the Bvlc_reference_caffenet model, this layer will begin training with random weights [52]. Tabel.2 gives a fine-tuned model with AlexNet.

C.Caffe Framework

There are several powerful libraries that can be used to design and teach neural networks, including convolutional neural networks such as Theano, Lasagne, Keras, MXNet, Torch, and TensorFlow. Among them, Caffe is a library that can be used to research and develop real-world applications [54].

Caffe is a completely open-source library that gives open access to deep architectures. The code is written in CUDA, Python / Numpy, which is used for GPU computation, and efficient C ++, with nearly complete, well-supported contexts for MATLAB. Caffe offers unit tests for accuracy, experimental rigor and installation speed, depending on the best practices of software engineering. In addition, the code is also well suited for research use due to careful modularity and clean separation of the network definition from the actual implementation [54].

Caffe trains models with fast and standard stochastic gradient descent [55] algorithm. In the proposed method, the optimization is achieved by using a 64 mini-batch by the stochastic gradient descent and a 0.9 momentum coefficient. The model is organized using "dropout" and "weight decay".

During training: A data layer takes images and tags from the disk, passes them over multiple layers, such as convolution, pooling, and rectified linear transformations, and makes the final guess. The data are processed in tiny stacks that pass sequentially through the network. What is vital for training is the snapshot of learning speed reduction timelines, momentum stopping and resuming; all of them are implemented and documented [53].

The learning curve of the proposed method is as shown in fig.7. Test accuracy does not change after 4500th repetition and remains at 91%. Although the loss of training is high in the first repetitions, it does not change much after 5000th and it is around 0.1. This means that the network has learned successfully. The network now knows which images can be faced with 243.470 positive and 56.798 negative images.

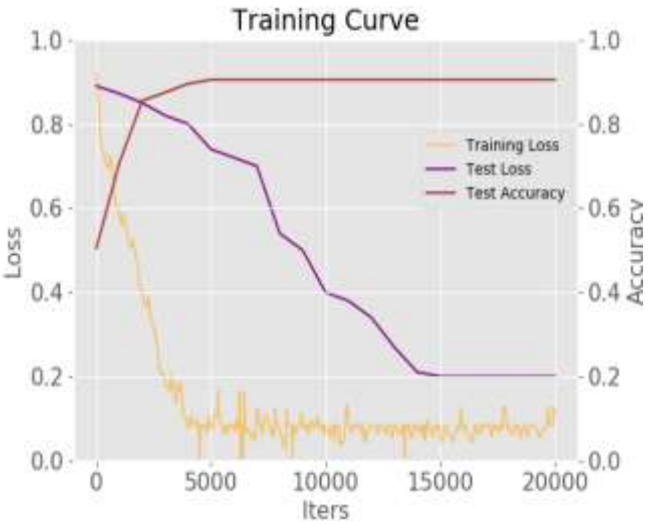


Fig.7. Training curve of model

V. EXPERİMANTAL RESULTS

The suggested method is FDDB, MUCT data sets and Google open access data sets to check whether the desired operation is performed at the end of the long training period. It has been tested with images downloaded from the internet. The MUST used for the test has differences such as enlightenment, age, race, gender. There is only one person in all the images. FDDB contains blockages, closures, difficult exposures and low-resolution images. There are multiple images in the dataset as well as images with only one person.

On the other hand, the methods have developed with the HOG feature descriptor and the haar cascade algorithm have been tested using the images in the FDDB data. These methods find all faces in some images to be correct. But the images that have provided by finding the right ones are those that have no effect of illumination and no change of exposure. The HOG feature descriptor does not find faces that are not particularly flat, and some that are partially closed. Haarcascade also accepted some of the non-facial parts of the imagery as a face. Frontal faces are quite successful in finding. Besides, some of the images did not find the non-frontal faces. Fig.8, fig.9, and fig.10 show the test results have obtained using the images in FDDB dataset. Fig.11 shows the proposed method with some images in the MUCT dataset.

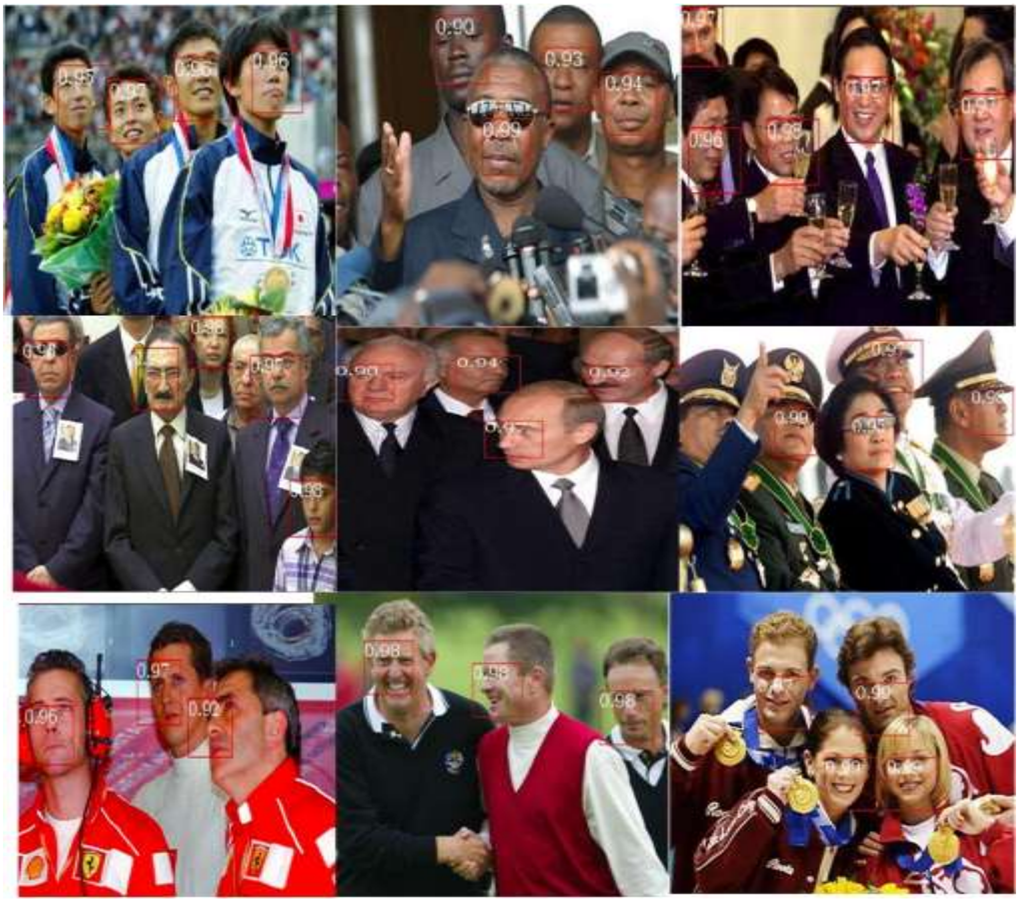


Fig.8. Results of CNN detection on FDDB

Fig.9. Results of haarcascade detection on FDDB

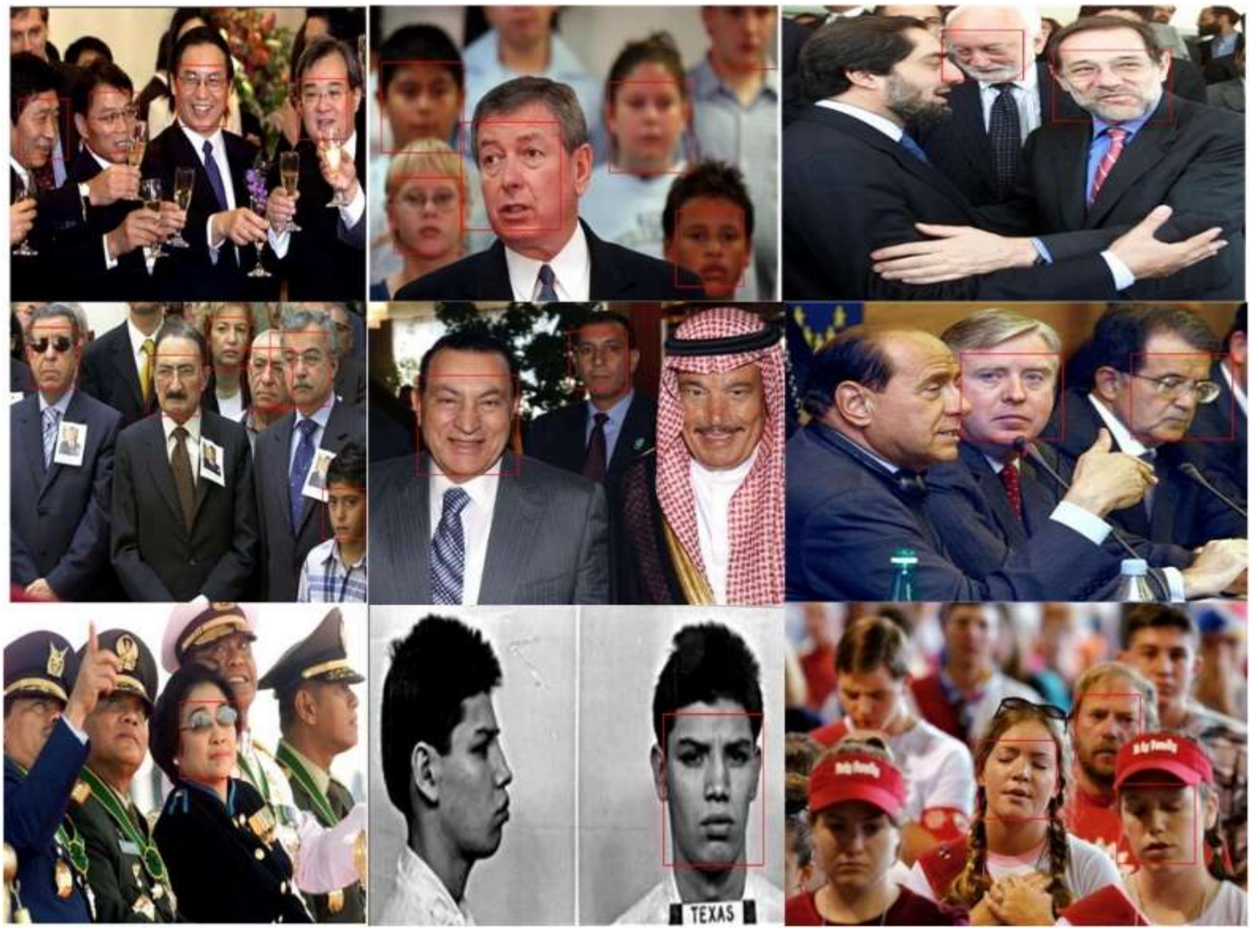


Fig.10. Results of HOG detection on FDDB

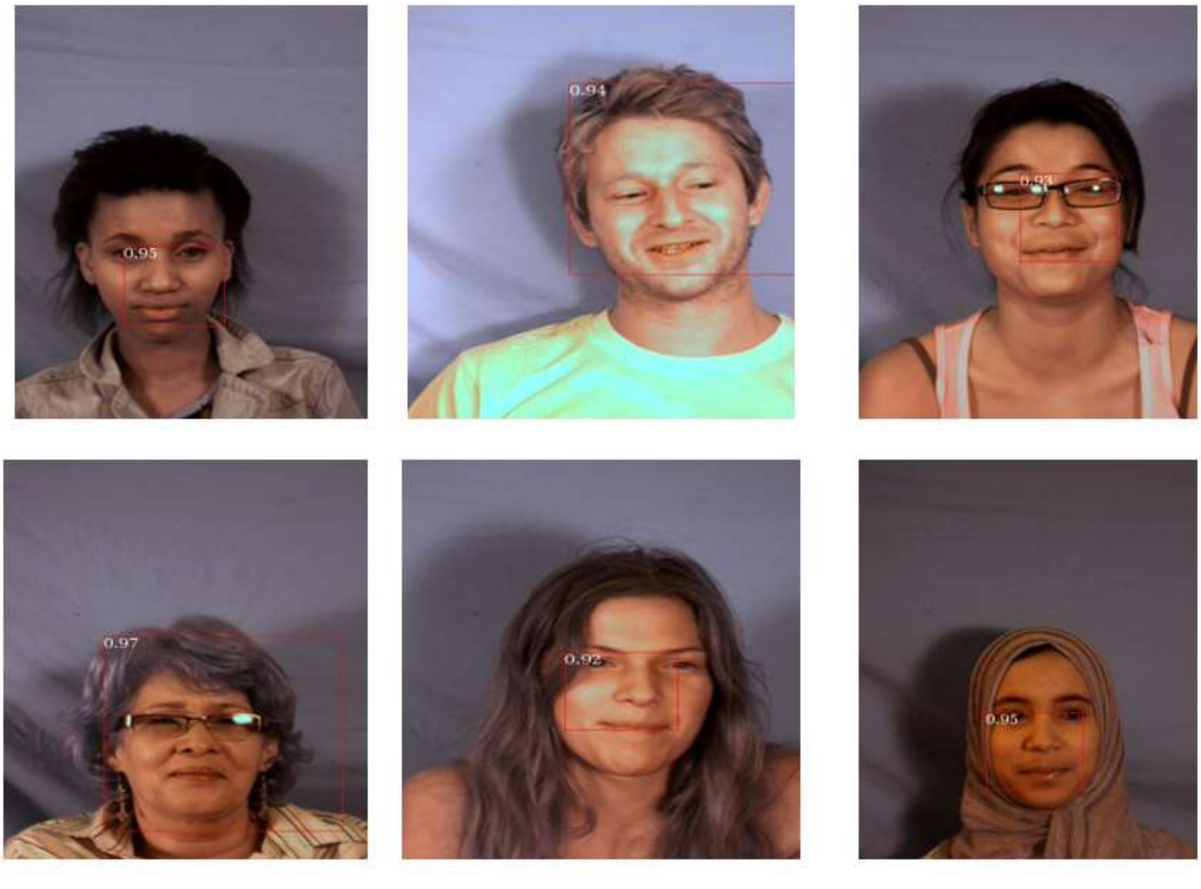


Fig. 11. Results of CNN detection on MUCT

CONCLUSIONS

In this work, we propose a face detector with CNN base. The method is implemented by using the python language and caffe library in ubuntu 16.04 operating system. The architecture we use has many parameters. We also use a lot of data for training. We take advantage of GPU technology to reduce the length of training that will take much longer under normal conditions. Our method of testing with publishing databases MUST and FDDB is successful in finding faces with partially closed faces and pose change in images. In the study, HOG and HaarCascade methods which are frequently used to perform face finding process have been tested with images taken from FDDB. The results show that our CNN-based method is better at finding faces in challenging images.

ACKNOWLEDGEMENT

This work is supported by Fırat University Scientific Research Projects (FÜBAP) with project number MF.16.67 as a research project of the master thesis.

REFERENCES

[1] ssH. Özcan, "Deep Learning Practices in Very Low-Resolution Face Images," Master thesis, Naval War College, Turkey,(2014)

[2] E. Cengil, A. Çınar,"Robust Face Detection with CNN", In *Proceedings of the 4nd ICAT international conference on Advanced Technology & Sciences* (pp. 47-51). ICAT, (2016).

[3] Ş. Karahan, Y.S. Akgül, "*Eye Detection by Using Deep Learning*", Kocaeli, Turkey,(2016).

[4] Varol, A., Cebe, B., "Face Recognition Algorithms", 5 the International Computer & Instructional Technologies Symposium., Fırat University, elazığ- turkey.

[5] Köstinger, M., Wohlhart, P., Roth, P. M., "Bischof, H., Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization", In Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference, pp. 2144-2151, (,November,2011).

[6] Jain V, Learned-Miller EG., "Fddb: A benchmark for face detection in unconstrained settings", UMass Amherst Technical Report, 2010.

[7] Milborrow, S., Morkel, J., Nicolls, F. "The MUCT landmarked face database", Pattern Recognition Association of South Africa, *201*(0), (2010).

[8] Ghimire D., Lee, J., A Robust Face Detection Method Based On Skin Color And Edges, J Inf Process Syst, Vol.9, No.1, March 2013.

[9] Jairath, S., Bharadvaj, S., Vatsa, M., Singh, R., Adaptive Skin Color Model to Improve Video Face Detection, Machine Intelligence and Signal Processing, Springer India, p. 131-142, 2016.

[10] Lan, O., Xu, Z., "Adaboost multi-view face detection based on YCgCr Skin Color model", Eight International Symposium sson Advanced Optical Manufacturing and Testing Technology (AOMATT2016), p. 96842D-96842D-5-528, 2016.

[11] Kang, S., Byuoungio C., Donghw, "J, Faces Detection Method Based on Skin Color Modeling, Journal of Systems Architecture", 64, 100-109, 2016

[12] Zhou, S., Yin, J., "Face detection using multi-block local gradient patterns and support vector machine, Journal of Computational Information Systems", 2014, 10(4), 1767-1776.

[13] Jinxin, R., Yin J.," Face detection based on facial features and linear support vector machines, Communication Software and Networks", ICCSN'09 International Conference on IEEE, 2009.

[14] Zhang, X, Pu, J., Huang X.," Face Detection Based on Two-Dimensional Principal Component Analysis and Support Vector Machine", Mechatronics and Automation Proceedings of the 2006 IEEE International Conference on IEEE, 1488-1492, (2006).

[15] El-Bakry, Hazem, M., Hamada, M., "Fast Principal Component Analysis for Face Detection using Cross-correlation and Image Decomposition", Neural Networks, IJCNN 2009, 751-756,(2009).

[16] Li Y.Y., Tsai C.C., Chen Y.Z., "Face Detection, Identification and Tracking using Support Vector Machine and Fuzzy Kalman Filter", Machine Learning and Cybernetics (ICMLC, 2011).

[17] Yang H.C, Xu A.W., "Cascade Face Detection Based on Histograms of Gradients and Support Vector Machine", Parallel, Grid, sCloud, and Internet Computing (3PGCIC), 2015 10th International Conference on IEEE, 2015.

[18] Viola, P., Jones M.J., Robust real face detection, International Journal of Computer Vision, 57(2), 137-154, 2004.

[19] Coster, J., Ohilsson, M., Human Attention: The possibility of measuring human attention using OpenCV and the Viola-Jones face detection algorithm, 2015.

[20] Irgens, P., Bader, C., Lé, T., Saxena, D., Ababei, C. An efficient and cost-effective FPGA based implementation of the Viola-Jones face detection algorithm, *hardware*, vol.1 p. 68-75, 2017.

[21] Zhao, X., Chai, X., Niu, Z., Heng, C., & Shan, S., Context modeling for facial landmark detection based on Non-Adjacent Rectangle (NAR) Haar-like feature, Image and Vision Computing, *30*(3), 136-146, 2012.

[22] Aziz, K.A.A., Shahrum, S.A., Face Detection Using Radial Basis Functions Neural Networks With Fixed Spread, *arXiv preprint arXiv:1410.2173*, 2014.

[23] Ali, N. M., Karis, M. S., Aziz, M. A., Abidin, A. F. Z., Analysis of frontal face detection performance by using Artificial Neural Network (ANN) and Speed-Up Robust Features (SURF) technique, M. H. R. O. Abdullah, Y. H. Min, N. A. M. Bashar, & S. M. F. S. A. Nasir (Eds.), AIP Conference Proceedings, Vol. 1774, No. 1, p. 050013, AIP Publishing, 2016

[24] Joseph, S., Sowmiya, R., Thomas, R. A., Sofia, X. Face detection through the neural network. In Current Trends in Engineering and Technology (ICCTET), 2014 2nd International Conference, p. 163-166. IEEE, 2014.

[25] Bulo, S. R., "Face detection with neural networks", Universita Ca'Foscari Venezia DAIS, 2012.

[26] Zhonghua, C., Lichuan, H., Tiejun, W., Fengyi, G., “Modeling Study of the Amount of Wear in Sliding Electric Contact”, 26th International Conference on Electrical Contacts (ICEC 2012), 146 – 150, 2012.

[27] Miry, A. H., “Face Detection Based on Multi Facial Feature using Fuzzy Logic”, Al-Mansour Journal, 21, 2014.

[28] Elbouz, M., Alfalou, A., & Brosseau, C., “Fuzzy logic and optical correlation-based face recognition method for patient monitoring application in home video surveillance”, Optical Engineering, *50*(6), 067003-067003, 2011.

[29] Yang, S., Luo, P., Loy, C. C., Tang, X.,” From facial parts responses to face detection: A deep learning approach”, In Proceedings of the IEEE International Conference on Computer Vision , p. 3676-3684, 2015.

[30] Qin, H., Yan, J., Li, X., Hu, X. “Joint training of cascaded CNN for face detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition”, p. 3456-3465, (2016)

[31] Kong, K. H., Kang, D. S., “ A Study of Face Detection Using CNN and Cascade Based on Symmetry-LGP & Uniform-LGP and the Skin Color”, (2016).

[32] Sun, X., Wu, P., Hoi, S. C., “Face detection using deep learning: An improved faster rcnn approach”, arXiv preprint arXiv:1701.08289, 2017.

[33] Zhang, K., Zhang, Z., Li, Z., Qiao, Y.” Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks”, IEEE Signal Processing Letters, *23*(10), 1499-1503, (2016).

[34] Dalal, N., Triggs, B. “Histograms of oriented gradients for human detection”, In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 1, pp. 886-893). IEEE. (2005, June).

[35] Tomasi, Carlo. "Histograms of Oriented Gradients." (2015).

[36] Eral, M., “Deep Learning Approasch for Laboratory mice grimace Scaling”, master thesis, METU,(Ankara, 2016).

[37] Krizhevsky, A., Sutskever, I. and Hinton, G. E., “ImageNet Classification with Deep Convolutional Neural Networks”, Advances in Neural Information Processing Systems, (2012).

[38] http://cs231n.github.io/convolutional-networks/

[39] Karpathy, A., “Convolutional Neural Networks (CNNs / ConvNets) “, Retrieved August 7, 2016, from CS231n Convolutional Neural Networks for Visual Recognition:

[40] E. Cengil, A. Cinar, A New Approach for Image Classification: Convolutional Neural Network, European Journal of Technic (EJT), 6(2), 96-103, (2016).

[41] Karaatlı, M. A. “Estimation by Artificial Neural Networks Method”, PhD thesis, Süleyman Demirel University, (Isparta, 2011).

[42] http://cs231n.github.io/neural-networks-1/CS231n: Convolutional Neural Networks for Visual Recognition

[43] Nair, V., Hinton, G. E. “Rectified linear units improve restricted boltzmann machines”, In Proceedings of the 27th international conference on machine learning (ICML-10), pp. 807-814, 2010.

[44] An Intuitive Explanation of Convolutional Neural Networks, https://ujjwalkarn.me/2016/08/11/intuitiveexplanation -convnets/

[45] *BVLC AlexNet Model.* https://github.com/BVLC/caffe/tree/master/models/bvlc_alexnet

[46] Jain V, Learned-Miller EG., “Fddb: A benchmark for face detection in unconstrained settings”, UMass Amherst Technical Report, (2010).

[47] Huang, G. B., Ramesh, M., Berg, T., Learned-Miller, E. “Labeled faces in the wild: A database for studying face recognition in unconstrained environments“, (Vol. 1, No. 2, p. 3). Technical Report 07-49, University of Massachusetts, Amherst, (2007).

[48] Zhu, X., Ramanan, D., “Face detection, pose estimation, and landmark localization in the wild”. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on (pp. 2879-2886). IEEE. (2012, June).

[49] Phillips, P. J., Moon, H., Rauss, P. J., Rizvi, S., “The FERET evaluation methodology for face recognition algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence”, Vol. 22, No. 10, (October 2000).

[50] Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., Kumar, N., “Localizing parts of faces using a consensus of exemplars”, IEEE transactions on pattern analysis and machine intelligence, 3*5*(12), 2930-2940, (2013).

[51] Le, V., Brandt, J., Lin, Z., Bourdev, L., Huang, T., Interactive facial feature localization. Computer Vision–ECCV *2012*, 679-692, (2012).

[52] Jia, Y., Shelhamer, E., http://caffe.berkeleyvision.org/gathered/examples/finetune_ flickr_style.html

[53] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... Darrell, T., “Caffe: Convolutional architecture for fast feature embedding”, In Proceedings of the 22nd ACM international conference on Multimedia (pp. 675-678). ACM. (2014, November).

[54] Aghdam, H.H., Heravi, E. J., https://link.springer.com/chapter/10.1007/978-3-319-57550-6_4, (18 mayıs 2017).

[55] Tang, S., Stochastic gradient descent, (2017).