Advances in Science, Technology and Engineering Systems Journal Vol. 6, No. 2, 956-964 (2021)

www.astesj.com

Special Issue on Innovation in Computing, Engineering Science & Technology

ASTESJ

ISSN: 2415-6698

Performance Evaluation of Convolutional Neural Networks (CNNs) And VGG on Real Time Face Recognition System

Showkat Ahmad Dar*, S Palanivel

Department of Computer Science & Engineering, Annamalai University, Annamalainagar, Chidambaram, 608002, India

ARTICLE INFO

Article history:
Received: 18 January, 2021
Accepted: 27 March, 2021
Online: 15 April, 2021

Keywords:
Deep learning
Convolutional Neural Network (CNNs)
VGG16
Face authentication
Real time face images
Classifiers

ABSTRACT

*Face Recognition (FR) is considered as a heavily studied topic in computer vision field. The capability to automatically identify and authenticate human's faces using real-time images is an important aspect in surveillance, security, and other related domains. There are separate applications that help in identifying individuals at specific locations which help in detecting intruders. The real-time recognition is mandatory for surveillance purposes. A number of machine learning methods along with classifiers are used for the recognition of faces. Existing Machine Learning (ML) methods are failed to achieve optimal performance due to their inability to accurately extract the features from the face image, and enhancing system's recognition accuracy system becomes very difficult task. Majority of designed FR systems has two major steps: extraction of feature and classifier. Accurate FR system is still a challenge, primarily due to the higher computational time and separate feature extraction. In general, for various applications using images, deep learning algorithms are mostly recommended for solving these problems because it performs combined feature extraction and classification task. Deep learning algorithm reduces the computation time and enhances the recognition accuracy because of automatic extraction of feature. The major novelty of the work is to design a new VGG-16 with Transfer Learning algorithm for face recognition by varying active layers with three levels (3, 4, and 7). It also designs the Convolutional Neural Network (CNN) for FR system. The proposed work introduced a new Real Time Face Recognition (RTFR) system. The process is broken into three major steps: (1) database collection, (2) FR to identify particular persons and (3) Performance evaluation. For the first step, the system collects 1056 faces in real time for 24 persons using a camera with resolution of 112*92. Second step, efficient RTFR algorithm is then used to recognize faces with a known database. Here two different deep learning algorithms such as CNN and VGG-16 with Transfer Learning are introduced for RTFR system. This proposed system is implemented using Keras. Thirdly the performance of these two classifiers is measured using of precision, recall, F1-score, accuracy and k-fold cross validation. From results it concludes that proposed algorithm produces higher accuracy results of 99.37%, whereas the other existing classifiers such as VGG3, VGG7, and CNN gives the accuracy results of 75.71%, 96.53%, and 69.09% values respectively.*

1. Introduction

In recent years, an active research area is Face Recognition (FR) technology because of technology's potential in commercial use and law enforcement as well as rise in security demands [1]. The issues and developments in face recognition have been alluring a lot of scientists working in computer vision, pattern recognition, and biometrics domain. Various face recognition algorithms are used in diverse applications like indexing and video compression that comes under the domain of biometrics. Face recognition concepts can be used in classifying multimedia content and to help in the quick searching of materials that interests the end user. A comprehensive face recognition mechanism can be of assistance in domains like surveillance and forensic sciences. It can also be used in the areas of law enforcement and to authenticate security and banking systems. In addition to that, it also gives control and preferential access to

*Corresponding Author: Showkat Ahmad Dar, showkatme2009@gmail.com

secured areas and authorized users. The issues in face recognition have garnered even more significance after the spike in terrorism in the recent years. It largely decreases the need for passwords and can offer enhanced security. For this, FR should be used with additional security mechanisms.

In spite of facial recognition's rapid growth and acclaim as a critical authentication mechanism, the algorithms used for facial recognition have not been developed significantly. It has been close to two decades since facial recognition has come to the fore but a comprehensive system that is capable of producing positive results in real-time conditions has not been developed yet. The Face Recognition Vendor Test (FRVT) test carried out by National Institute of Standards and Technology (NIST) is demonstrated that modern face recognition mechanisms will not be able to perform optimally under certain circumstances.

Modern FR systems intended for complex environments has attracted a huge attention in recent years. FR systems that are automated is a developing technology that has garnered a lot of interest. There are a number of conventional algorithms that are used in developing color images and still-face images. The data complexity is increased in color images as the pixels are mapped to a high-dimensional space. This significantly decreases the accuracy and processing efficiency of FR [2].

There are four stages in FR technology namely, classification, representation (extraction of facial feature), alignment and detection [3]. Feature representation technique used for extracting features is the major challenge of FR system. For a specified biometric trait, representations are done using better techniques. In image classification, highly important step is feature extraction. Highly important information is retained in feature extraction, which is used in classification. Feature extraction methods improved FR accuracy slowly.

FR systems are reported with failures or unstable performance often with different false rates in real-world environment due to technical insufficiency. Making it formal and complete use of its performance is being yet a final solution. In the recent years, it has been inferred that deep learning works a lot better for face recognition [4]. For classification and feature extraction, processing unit's multiple layer cascading is used in deep learning techniques. Facial image's very high recognition rate is achieved by this.

The proposed study processes color images to recognize and detect faces with a good deal of accuracy in a real-time scenario. This work aims to apply pre-trained Convolutional Neural Network (CNN) with VGG-16 algorithm for FR and classification accuracy using analysis. These methods have been used for enhancing recognition accuracy. Most relevant challenge for such a system is recognition and feature extraction. A system has been proposed here that makes use of deep learning techniques to extract features. System uses deep learning to recognize faces in an accurate manner. The proposed system will be capable of recognizing more number of faces which can be used for searching suspects as the errors are reduced significantly.

2. Literature Review

FR has become a popular topic of research recently due to increases in demand for security. Highly attractive biometric technology is FR and its accuracy is highly enhanced using recent advancements in technology. According to Deep Learning (DL) and Machine Learning (ML) techniques, FR techniques are performed.

In [5] proposed Haar classifier that made use of a surveillance camera for face recognition. The system had four sequential steps that included (1) real-time image training (2) face recognition with the help of Haar-classifier (3) comparing the real-time images having images that were captured from camera (4) generation of the result as per the comparison. Haar is used to detect the faces in a robust manner in real-time scenarios. Face detection uses an algorithm called as Haar cascading. A system called as Aadhar has been adopted by India for recognizing the citizens. If this is used as a database for the citizens, a local citizen and a foreigner can be easily recognized. This information can be eventually used for identifying if the person is a criminal or not. The major advantage of this work is to apply this system to citizenship database. This has less computational cost, better discriminatory power and high recognition precision. Least processing is required by main features in small images for training Haar wavelets. If number of features becomes more it requires increased computation time for FR system. It is left as scope of the work.

In [6] proposed a Local Binary Pattern (LBP) for identifying faces. This was used along with other image processing methods like Histogram Equalization (HE), Bilateral Filter, Contrast Adjustment and Image Blending to improve overall system accuracy. The LBP codes are improved here due to which the performance of the system is enhanced. The major advantage of this work is that it is reliable, robust and accurate. In real-life setting, this may be used as an attendance management system. But this solution limits only the noise. In facial recognition, mask faces and occlusion issues are not addressed by this system. But, in future, this work can be extended for addressing these issues.

For face feature extraction, combination of Local Binary Pattern (LBP) and Histograms of Oriented Gradients (HOG) descriptors [7]. Low computation power is required by these descriptors. For face classification, Random Forest (RF) classifier based accurate and latest methods are applied. Under a controlled environment, from video broadcast, identification and verification of one or more person can be done accurately using this proposed algorithm. For FR system. Still there is a need to have separate feature extraction technique.

For facial expression recognition, a real-time system is presented [8]. Student's 8 basic facial expressions can be recognized using this proposed system and expressions includes natural, surprise, sad, nervous, happy, feat, disgust and anger inside E-learning environment. Proposed system's efficiency is tested by using Support Vector Machine (SVM), k-Nearest Neighbor (k-NN) classifiers and their results are compared. Techniques used in this study for recognizing facial expressions includes SVM and k-NN classifiers for expression recognition, feature selection based on Principal Component Analysis (PCA), feature extraction based on Gabor Feature approach, Viola-Jones technique based face detection. In a real-time system, for facial expression recognition, it can be used. However the k-NN and

SVM classifier needs more time complete the FR process due to feature extraction and feature selection steps.

For addressing partial face images irrespective of its size, Fully Convolutional Network (FCN) with Sparse Representation Classification (SRC) [9] and it is termed as Dynamic Feature Matching (DFM). For optimizing FCN, introduced a sliding loss based on DFM. Between subject's face image and face patch, an intra-variation is reduced for enhancing DFM performance. For other visual recognition tasks like partial person re-identification, this technique can be extended very easily. This solution limits the noise alone and image count is also restricted.

For suppressing unreliable local features from occluded regions, a fuzzy max-pooling scheme [10]. On every subclass, automatic weighting is done for enhancing robustness in final average-pooling. While dealing with data sufficiency and occlusion problem simultaneously, better performance enhancement is shown by this technique, which is a major advantage of this technique. Every subclass is treated as an individual classifier, where ensemble late fusion framework is used for obtaining final decision. Remarkable enhancement in performance can be achieved as shown in results.

Under various conditions, for face recognition, a framework called Optimized Symmetric Partial Face graph (OSPE) [11]. For instance, light variation, facial expression; occluded faces are used in their experimentation. Partial facial data are introduced for enhancing recognition rate as shown in their experimental results. Local spatial information is not explored fully in these techniques, which is a major drawback of it.

For dealing with FR, a Principal Component Analysis (PCA) technique based on patch [12]. Total scatter is computed for computing divided patches correlation directly. For feature extraction, projection matrix is obtained by optimizing projected samples total scatter. Nearest Neighbor (NN) classifier is used at last. For this large sized covariance matrix, eigenvectors evaluation consumes more time.

For real time face recognition, a LBP [13]. The image of the face is represented by using information about the texture and shape. For representing the face comprehensively, the facial area is divided into different sections. LBP histograms are then extorted which are combined to a single histogram. Facial recognition is then using the Nearest Neighbor (NN) classifier. The validation of the algorithm is carried out by devising a prototype model that makes use of the raspberry Pi single-board computer, Matrix Laboratory (MATLAB). The results indicate that LBP algorithm's recognition rate is relatively higher when compared to other approaches.

In [14], four various algorithms are combined with Discrete Wavelet Transform (DWT). Algorithm includes Convolutional Neural Network (CNN), Linear Discriminant Analysis (LDA) Eigen vector, PCA Eigen vector and Principal Component Analysis (PCA) error vector. Then Fuzzy system and detection probability's entropy are used for combining these four results. Database diversity and image defines the recognition accuracy as indicated in results. For best case, 93.34% recognition rate and for worst case, 89.56% recognition rate are provided by this technique. This technique is better than other techniques, where individual technique is implemented on specific images set. In human face recognition, illumination impact is ignored, which is a only limitation in this work.

All these techniques, reviewed don't completely address issues affecting facial recognition accuracy like feature extraction and noise variation. Unfortunately, the processing time and training period for these algorithms are considerably large. In the recent times, Face Recognition (FR) techniques have been replaced by deep learning. Deep learning has been observed to perform better for large datasets. On the contrary, traditional ML algorithms at an optimum level with comparatively smaller datasets. In conventional ML algorithms, a difficulty needs to be broken down into individual steps. FR based on CNN is trained with large datasets and has attained a high level of accuracy. The increased use of deep learning has accelerated the research involved in FR. Recently, with deep learning emergence, impressive results are achieved in face recognition. In computer vision applications, most popular deep neural network is CNN and it has automatic visual feature extraction which is a major advantage [15].

For recognizing face images, Support Vector Machine (SVM) and Convolutional Neural Network (CNN) [16]. For automatically acquiring remarkable features, CNN is utilized as feature extractor. Using ancillary data, CNN pre-training is proposed at first and updated weights are computed. Target dataset is used for training CNN, which extracts highly hidden facial features. At last, for recognizing all classes, proposed a SVM classifier. With high accuracy, face images are recognized using SVM where facial features extracted from CNN are given as an input. In experimentation, for pre-training, images in Casia-Webfaces database are used and to test and train, used the Facial Recognition Technology (FERET) database. With less training time and high recognition rate, efficiency is demonstrated in experimentation results. Moreover, it is highly difficult to acquire some facial features manually from face images. But, effective facial features are automatically extracted using CNN. With more optimization techniques, deeper CNN based training time and recognition rate's balance point are computed and larger dataset is left as scope of this work.

A modified CNN architecture [17], where two normalization operations are added to two layers. Batch normalization is used as a normalization operation and network is accelerated using this. Distinctive face features are extracted using CNN architecture and in CNN's fully connected layer, faces are classified using Softmax classifier. With better recognition results, face recognition performance is enhanced using this proposed technique as shown in experiment part and it uses Georgia Tech Database.

A novel technique based on CNN [18] which is termed as Deep Coupled ResNet (DCR) model. It consists of two branch networks and trunk network. For face images with various resolutions, discriminative features are extracted using trunk network. High-Resolution (HR) images are transformed using two branch networks and targeted Low Resolution (LR) images are also transformed using this.

Pre-trained CNN's performance [19] with multi-class Support Vector Machine (SVM) classifier and for performing classification, AlexNet model based transfer learning

performance is also proposed. An excellent performance is achieved using hybrid deep learning as shown in results when compared with other techniques of face verification.

For facial recognition, CNN and Multi-Layer Perceptron (MLP) [20]. For performing facial recognition, it is a system based on open code deep-learning. For fiducial point embedding and extraction, deep learning techniques are used. For classification task, SVM is used. In inference and training, it has fast performance. For facial features detection, 0.12103 error rate is achieved using this system, which is pretty close for state of the art algorithms and for face recognition, it has 0.05. In real-time, it can run.

In [21] proposed a CNN for facial recognition that has been implemented on the embedded GPU system. The method uses facial recognition based on CNN along with face tracking and deep CNN facial recognition algorithm. The results have indicated that the system is capable of recognizing different faces. An estimated 8 faces can be recognized simultaneously within a span of 0.23 seconds. The recognition rate has been above 83.67%. As a result, processing time is enhanced and for real-time multiple face recognition, it can be used with acceptable recognition rate.

For face recognition, a Pyramid-Based Scale-Invariant - CNN model (PSI-CNN) [22]. From image, untrained features are extracted using PSI-CNN model and with original feature maps, these features are fused. With respect to matching accuracy, experimentation results are shown and it outperforms model derived from VGG-Face model. Stable performance is maintained using PSI-CNN during the experimentation with low-resolution images which are acquired from CCTV cameras. Robust performance is exhibited with image quality and resolution change.

For regressing facial landmark coordinates, at CNN's intermediate layers [23]. In specific poses and appearances, for regressing facial landmark coordinates, specialized architecture is designed in novel CNN architecture. For every specialized sub-networks, for providing sufficient training examples, designed a data augmentation techniques and it address training data's shortage specifically in extreme profile poses. On true positive detected faces, accuracy is reflected by this. At last, trained and code models are made publicly available using project webpage, for promoting results reproducibility.

Using CNN, real-time face recognition system's evaluation [24]. Standard AT&T datasets is used for performing proposed design's initial evaluation and for designing a real-time system, same can be extended. For enhancing and assessing proposed system's recognition accuracy, CNN parameters tuning are used. For enhancing system performance, systematic technique for tuning parameters is proposed. From the review it concludes that the deep learning algorithms give lesser computation time and more recognition accuracy than the other methods.

The VGG16 based multi-level information fusion model [25]. On fully connected neural network, enhancement is proposed. Computation time is reduced using this technique. In calculation and propagation process, some useful feature information are lost in CNN model. This work enhances model to be a convolution calculation's multi-level information fusion and discarded feature information are also recovered and it enhances image's recognition rate. Network is divided into five groups using VGG and they are combined as a convolution sequence. The main advantage here is the representational efficiency. Face recognition can be attained using 128 bytes per face.

3. Proposed Methodology

The proposed system is broken into three major steps: (1) database collection, (2) face recognition to identify particular persons, and (3) Performance evaluation. For the first step, the system collects the faces in real time. In the database consists of 24 different persons are in the form of 1056 images having the resolution of 112*92; Olivetti Dataset is also used for implementation. In second step, Convolutional Neural Network (CNN) and VGG-16 Deep Convolutional Neural Network (DCNN) are introduced for improving recognition accuracy. Finally results evaluation of these two classifiers is measured using precision, F1-score, recall and accuracy. These classifiers recognize faces in real-time with high accuracy. The recognition building blocks are shown in the Figure 1.

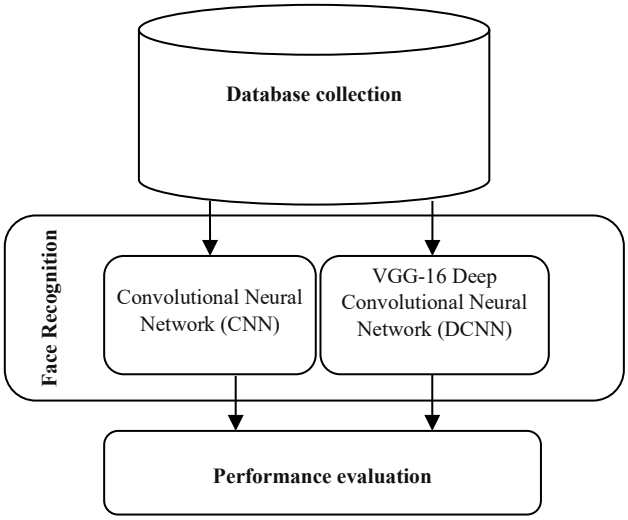

Figure 1: Face Recognition Building Blocks

3.1. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is attracted image processing application's attraction [26, 27, 28] and it is also used for feature extraction capacity from real-time facial images. General CNN model which is used for real time facial recognition has been illustrated in Figure 2. The CNN architecture has three major layers: Convolutional, Fully Connected, and Pooling Layer. The initial layer, facial image samples are given as input. Then in convolution layer, real time face images are converted into facial feature vectors via the kernel with filters=5. It is followed by ReLU activation function for recognition as well as maximum pooling layer. Next in line comes the full connection layer followed by the output layer.

CNN's basic building blocks are explained as follows:

Input layer- This layer has image's raw input having width 112, height 92, depth 3.

Convolution Layer – In CNN**,** a matrix called as the kernel is passed over the input real time face matrix with size of (112*92)

to devise a feature map for subsequent layer. Mathematical operation termed as the convolution is executed via sliding Kernel size (3*3) over input real time face matrix. On each of the locations, real time face matrix multiplication is carried out and adds result set onto final feature map. For example, let use consider a 2-Dimensional kernel filter (K=5, 3*3), and a 2-Dimensional real time facial input image, I. This layer computes output volume via computing dot product between all filters and facial image. Output volume is computed using this layer via computing dot product between facial image and all filters. In this layer, filters are used to produce output volume with 112x 92 x 5 dimension.

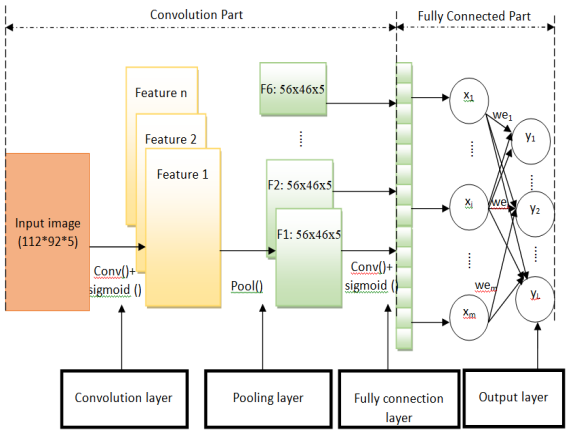

Figure 2: Convolutional Neural Networks Architecture

In layer l, assume In^las neurons input and as neurons output Op^l. Every neuron's activation function and input are used for computing every neurons output. Layer number is represented as l, for instance, for first layer l=1 and for last layer, l=-1. Row number is expressed as i and column number is represented as j. Three-dimensional matrix is produced by CNN' every In^l and Op^l layers data structures, while one dimensional vector is produced by every layer In^l and Op^l in FC.

Assume layer l's weight as we_{ij}^l and bias as ba_j^l. FC last layer's weight is represented as we_{ij}^{-1} and bias is represented as ba_j^{-1}. If layer l is pooling or convolutional layer, pooling windows or convolutional kernel's size is represented as $size^l \times size^l$. If layer l is fully-connected layer, neurons count is given by$size^l$. In layer l, neurons every input value is represented as In_{mn}^l . For convolution computation, convolution(Op^{l-1}, we^l, m, n) function is used. Previous layer's output is given by Op^{l-1}. Between layer l (In^l))'s input and previous layer (Op^{l-1})'s output, weights matrix is given by we^l. Layer l's bias is given by ba^l. The convolutional layer (In^l)'s input is computed as,

$$In_{mn}^l = Convolution(Op^{l-1}, we^l, m, n) + ba^l = \sum_{i=0}^{size^l-1} \sum_{j=0}^{size^l-1} \left(Op_{m+i,n+j}^{l-1} . we_{i,j}^l + ba^l\right) \quad (1)$$

The convolutional layer l (Op_{mn}^l)'s output is calculated as equation (2), where, sigmoid () is activation function

$$Op_{mn}^l = F(net_{mn}^l) = sigmoid(net_{mn}^l) = \frac{1}{1+e^{-In_{mn}^l}} \quad (2)$$

Non-linear activation functions (ReLU) –Once the kernel filter is formed then the next step is to perform FR system. A node that comes next to the convolutional layer is called as activation function. The Rectified Linear Unit (ReLU) can be considered as piecewise linear function which is given as output, if it is positive, or else the output will be given as zero. Total 5 filters for this layer will get output volume 112x 92 x 5 dimension. The expression of ReLu function is R(z)=max(0,z) the function and its derivative image are shown in Figure 3.

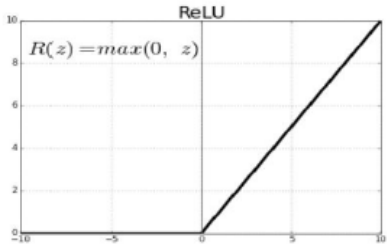

Figure 3: RELU Activation Function

Pooling Layer– In covnets, pooling layer is inserted periodically and its major objective are, minimization of volume size, making fast computation, reducing memory and prevention of overfitting. With stride 2 and 2*2 filters, this work uses a max pool. For face images recognition, resultant volume will have 56x46x5 dimension. Face matrix's average pooling is represented as pool(x) function. Expression (3) gives formula used for computing pool(x). Pooling window size is represented as $size^l$.

$$y_{ij} = pool(x, i, j) = \frac{\sum_{m=1}^{size^l} \sum_{n=1}^{size^l} x_{size^l \times (i-1)+m, size^l \times (j-1)+n}^{l-1}}{size^l \times size^l} \quad (3)$$

As indicated in expression (3), previous layer (Op^{l-1})'s output forms base for pooling layer Op^l's output. In other words, previous layer (Op^{l-1})'s output will be equal topooling layer l (net^l)'s input. As mentioned in above definition, last FC layer's input is represented as t^{-1} , first FC layer's input is represented as net^{-2}, layer before FC's input (pooling layer's last layer) is represented as net^{-3}.

Fully Connected Layer (FC)- Fully Connected (FC) Layer indicates that every node in initial layer is connected with next layer's every node. It computes class scores of facial images and outputs 1-D array of size equal to classes count via softmax function .The CNN model that was proposed was trained with 5 epochs, batch size=32 and using Adam optimizer (adaptive moment estimation). Specify optimizer's learning rate is set at 0.001. The forward propagation's result is$\hat{y}_n$, which is formulated in equation (4) to equation (6).

$$In_j^{-1} = \sum_{m=1}^{size^{-2}} \left(Op_i^{-2} . we_{ij}^{-1} + ba^{-1}\right), j = 1, \ldots size^{-1} \quad (4)$$

$$Op_j^{-1} = F\left(In_j^{-1}\right) = Sigmoid\left(In_j^{-1}\right) = \frac{1}{1+e^{-In_j^{-1}}} \quad (5)$$

$$\hat{y}_n = Op^{-1} \quad (6)$$

The CNN's pseudocodeis listed in Algorithm 1.

Algorithm 1: Training Algorithm of CNN

Input: tr_x, tr_y is considered as the features and labels of training facial images set

te_x, te_y is considered as the features and labels of testing facial images set.

Output: $we_{i,j}^{l}, ba_j^{l}$ weights and bias of convolution and pooling layers respectively

we_{jk}, ba_{jk} weights and bias of Fully Connected (FC) layer respectively

Required parameters: Max_{iter} Maximum number of iterations to complete face recognition task,

Error: when the training error is less than error , the training is finished , η: learning rate of the classifier

Initialization work:

t: t is the current iteration which is initialized as t=1 before training loop, L(t): L(t) is the Mean Square Error (MSE) at iteration t, $L(t)$ is initialized as $L(1) = 1 \geq error$

Begin

Set the required parameters and complete the initialization work

While $t < Max_{iter} \& L(t) > error$

For all tr_x

tr_r (Recognition label of training set tr_x and compute calculation from equation (1-6)

End for

Lt is recalculated as $L(t) = \frac{1}{2}\sum_{n=1}^{N}\left(tr_r(n) - tr_y(n)\right)^2$, N is the total images count in dataset

Increment t=t+1

End

3.2. VGG-16 Deep Convolutional Neural Network (DCNN) with Transfer Learning

VGG16 is considered to be a CNN model that enhances the AlexNet by making replacements of the large kernel-sized filters [29] having various 3×3 kernel-sized filters sequentially. VGG-16 proposed method has four major building blocks namely Softmax classifier, FC-layer, convolution module and attention module.

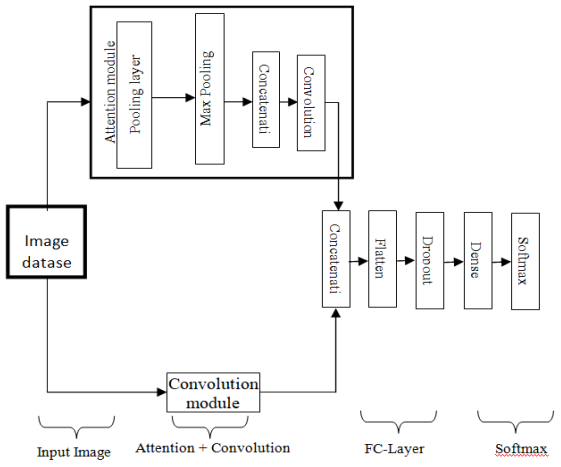

Figure 4: Block Diagram of the Proposed Attention-Based VGG-16 for Face Recognition

3.2.1. Attention Module

In facial image's feature's relationship are captured using this module. On input tensor, average pooling and max pooling are performed and in this technique, VGG-16 technique's 4th pooling layer is formed using this. For performing 7 × 7 filter size (f)'s convolution, max pooled 2D tensor is concatenated with every other layer via sigmoid function (σ). Figure 4 shows attention module's high level diagram. Expression (7) gives concatenated resultant tensor ($M_s(F)$).

$$M_s(F) = \sigma(f^{7\times7}[F^s{}_{max}]) \quad (7)$$

where $F_{avg}^{s} \in \mathbb{R}^{1\times H\times W}$ gives 2D tensors achieved using max pooling operation on input tensor F. Here tensor's height is represented as H and width is represented as W.

3.2.2. Convolutional Module

Convolution module is used which is VGG-16 model's 4th pooling layer. Facial image's features are captured using scale-invariant convolution module. From midlevel layer, extracted the features which are highly needed for real time facial images. For real time facial images, features from other layers like low or high are not appropriate as images are neither more specific nor more general. Thus, attention module is given with 4th pooling layer as first input. Then, that module's result is concatenated using 4th pooling layer.

3.3.3. Fully Connected (FC)-Layers

For representing concatenated features derived from convolution and attention block are converted as one-dimensional (1D) features using fully connected layers. It has three layers namely, dense, dropout and flatten as illustrated in Figure 4. Dropout is fixed as 0.5 and dense layer is limited to 24.

3.3.4. Softmax Classifier

From FC layer, features are extracted for classification and for facial image's final recognition, softmax layer is used. The unit number is defined by categories count in softmax layer, which is a last dense layer. According to classification, probability score's multinomial distribution is produced at softmax layer output. This distribution's output is given by,

$$P(a = c|b) = \frac{e^{b_k}}{\sum_j e^{b_j}} \quad (8)$$

where b probabilities that are retrieved from softmax layer and c represents facial image recognition dataset class used in proposed method. Table 1 gives proposed model's architecture details. Here, units in final dense layer (softmax layer) varies from one dataset to another based on categories count. ReLu activation function is applied in all layers except last one.

Table 1: VGG16 Proposed Model's Architecture

Layer (Type)	Output shape
VGG-16 model	90×110×5
conv2d_6 (Conv2D) layer	90×110×5
max_pooling2d_6(MaxPooling2) layer	45×55×5

conv2d_7 (Conv2D) layer	43×53×5
max_pooling2d_7 (MaxPooling2 (None, 21, 26, 5)	21×26×5
flatten_3 (Flatten)	2730
dense_3 (Dense)	24
Total params: 65,914 Trainable params: 65,914 Non-trainable params: 0	

4. Results and Discussion

This section evaluates facial recognitions method's performance. Real time database consists of 1056 images each of size 112*92.Olivetti database [30], dataset has 400 images with grayscale 64×64 pixels. For every person, there are 10 images so there is 40 persons (target) which make it 40×100 equals 400 rows of data. A confusion matrix needs to be computed for each class $g_i \in G = \{1, \dots, K\}$, in such a way that the ith confusion matrix assumes class g_i as the positive class and the remaining classes g_j with $j \neq i$ as negative class. As each confusion matrix pools together the entire observations labelled with a separate class apart from $g_i$as the negative class, this method increases the number of true negatives. This gives us:

- "**True Positive (TN)**" for event values that are correctly analyzed.
- "**False Positive (FP)**" for event values that are incorrectly analyzed.
- "**True Negative (TN)**" for no-event values that are correctly analyzed.

"**False Negative (FN)**" for no-event values that are incorrectly analyzed

Let usTP_i, TN_i, FP_i and $FN_i$to indicate the true positives respectively, false negatives, true negatives, false positives, in the confusion matrix associated with the ithclass. Let their call here be indicated by R and precision by P.

Micro average pools the performance over the least possible unit (the overall facial images):

$$P_{micro} = \frac{\sum_{i=1}^{|G|} TP_i}{\sum_{i=1}^{|G|} TP_i + FP_i} \quad (9)$$

$$R_{micro} = \frac{\sum_{i=1}^{|G|} TP_i}{\sum_{i=1}^{|G|} TP_i + FN_i} \quad (10)$$

The micro-averaged precision, P_{micro}, and recall, R_{micro}, give rise to the micro F1-score:

$$F1_{micro} = 2.\frac{P_{micro} \cdot R_{micro}}{P_{micro} + R_{micro}} \quad (11)$$

Given that a classifier gets a large$F1_{micro}$, it denotes that it performs exceedingly well. Here, micro-average may not be sensitive to the overall predictive performance. Due to this, the micro-average can be misleading when there is an imbalance in the class distribution.

Macro average averages over bigger groups and over the performance of individual classes than observations:

$$P_{macro} = \frac{1}{|G|} \sum_{i=1}^{|G|} TP_i / TP_i + FP_i \quad (12)$$

$$R_{macro} = \frac{1}{|G|} \sum_{i=1}^{|G|} TP_i / TP_i + FN_i \quad (13)$$

The recall and macro-averaged precision leads to the macro F1-score:

$$F1_{macro} = 2.\frac{P_{macro} \cdot R_{macro}}{P_{macro} + R_{macro}} \quad (14)$$

If $F1_{macro}$ has a bigger value, it points out to the fact that a classifier is able to perform well for each of the individual class.

Multi-class accuracy is termed as the average of the correct predictions:

$$accuracy = \frac{1}{N} \sum_{k=1}^{|G|} \sum_{x:g(x)=k} I(g(x) = \hat{g}(x)) \quad (13)$$

where I is defined as the indicator function, which returns 1 when there is a match between the classes and 0 otherwise. For significance testing, cross-validation is a statistical method used for estimating skill of classifiers. k-fold cross validation is a procedure used for estimating skill of model on new data. The value for k is fixed as 10, value that is found using experimentation to generally result in model skill estimate with low bias modest variance.

Table 2: Benchmark Datasets Results Comparison of Metrics Vs. Classifiers

Metrics	Datasets	kNN	SVM	CNN	VGG3	VGG7	VGG16
Macro Average-Precision (%)	Real Time face	65.00	70.00	72.00	78.00	97.00	99.00
	Olivetti	72.00	76.00	79.00	83.00	85.00	90.00
Macro Average-Recall (%)	Real Time face	67.00	71.00	74.00	80.00	97.00	99.00
	Olivetti	75.00	78.00	82.00	87.00	88.00	92.00
Macro Average-F1-score (%)	Real Time face	66.00	70.50	73.00	79.00	97.00	99.00
	Olivetti	73.50	77.00	80.50	85.00	86.50	91.00
Accuracy (%)	Real Time face	64.00	67.00	69.09	75.71	96.53	99.37
	Olivetti	75.00	80.00	84.00	88.00	90.00	94.00
K-fold cross validation Accuracy (%)	Real Time face	65.23	67.86	70.48	76.94	97.23	99.52
	Olivetti	76.10	81.24	84.90	88.66	91.81	95.18

Figure 5 shows the macro average-precision results comparison of six different classifiers like kNN, SVM, CNN, VGG3, VGG7, and VGG16 with two datasets. The proposed VGG16 classifier gives higher macro-average precision results of 99%, the other methods such as kNN, SVM, CNN, VGG3, VGG7 gives 65%, 70%, 72%, 78%, and 97% in real time dataset. The proposed VGG16 classifier gives higher macro-average precision results of 34%, 29%, 27%, 21% and 2% for kNN, SVM, CNN,

VGG3, and VGG7 methods respectively in real time dataset (See Table 2). It can be concluded that VGG16 gives higher macro average-precision, since the proposed work 16 sequential layers are used for recognition.

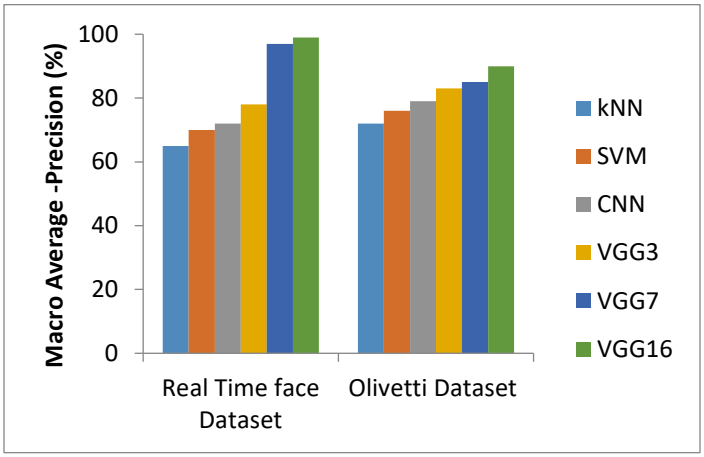

Figure 5: Macro Average-Precision Results Comparison vs. Classifiers

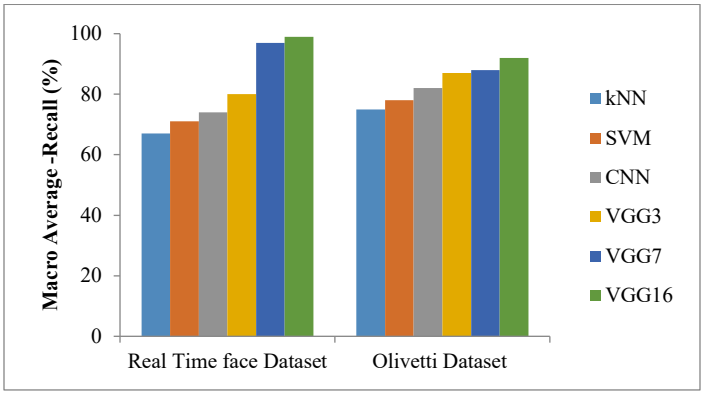

Figure 6: Macro Average-Recall Results Comparison vs. Classifiers

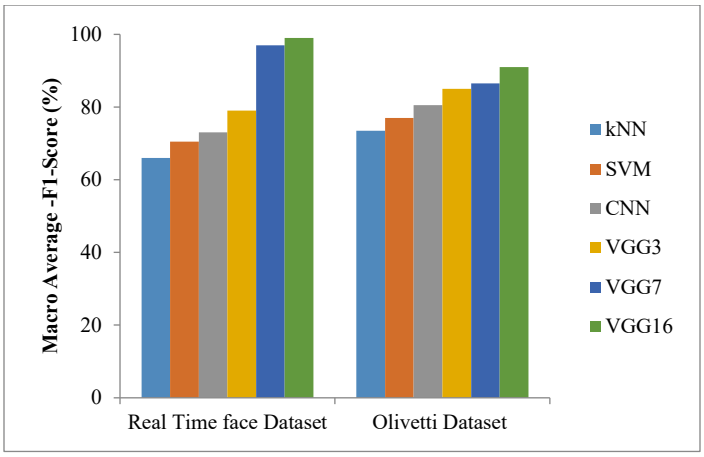

Figure 7 : Macro Average-F1-Score Results Comparison vs. Classifiers

Figure 6 shows macro average-recall results comparison of six different classifiers like kNN, SVM, CNN, VGG3, VGG7, and VGG16 with two datasets. The proposed VGG16 classifier gives higher macro-average recall results of 99%, the other methods such as kNN, SVM, CNN, VGG3, VGG7 gives 67%, 71%, 74%, 80%, and 97% in real time dataset. The proposed VGG16 classifier gives higher macro-average recall results of 32%, 28%, 25%, 19% and 2% for kNN, SVM, CNN, VGG3, and VGG7 methods respectively in real time dataset (See Table 2).

Macro average-F1-score results comparison of four different classifiers such of six different classifiers such as kNN, SVM, CNN, VGG3, VGG7, and VGG16 with two datasets are shown in the figure 7.The proposed VGG16 classifier gives higher macro-average F1-score results of 99%, the other methods such as kNN, SVM, CNN, VGG3, VGG7 gives 66%, 70.5%, 73%, 79%, and 97% in real time dataset. The proposed VGG16 classifier gives higher macro-average F1-score results of 33%, 28.5%, 26%, 20% and 2% for kNN, SVM, CNN, VGG3, and VGG7 methods respectively in real time dataset (See Table 2). On VGG16 based transfer leaning, for getting better results, proposed model is applied with both datasets.

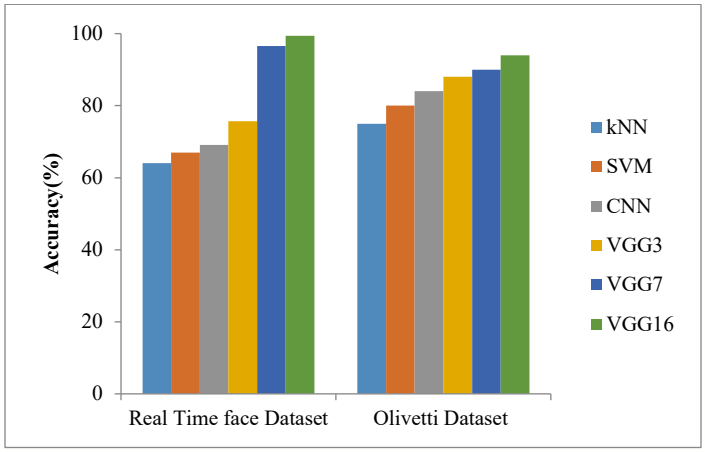

Figure 8: Accuracy Results Comparison VS. Classifiers

Figure 8 shows the accuracy results comparison of six different classifiers such as kNN, SVM, CNN, VGG3, VGG7, and VGG16 with two datasets. The proposed VGG16 classifier gives higher accuracy results of 99.37%, the other methods such as kNN, SVM, CNN, VGG3, VGG7 gives 64%, 67%, 69.09%, 75.71%, and 96.53% in real time dataset. The proposed VGG16 classifier gives higher accuracy results of 35.37%, 32.37%, 30.28%, 23.66% and 2.84% for kNN, SVM, CNN, VGG3, and VGG7 methods respectively in real time dataset (See Table 2). On VGG16 based transfer leaning, for getting better results, proposed model is applied with both datasets. The proposed classifier achieved a very high facial image's recognition rate which approaches human recognition rate.

5. Conclusion and future work

The term Biometrics defines individual's DNA along with other aspects like their facial features, geometry of the hands etc. In addition to that, the behavioral aspects like hand signatures, tone of voice and keystrokes are also taken into consideration. In many circumstances, the recognition of face is becoming more accepted and acclaimed in bio-metric based technologies. This helps in measuring an individual's natural data. This work puts forth a real time face recognition using classification methods. The proposed system contains three major steps that include (1) collection of facial images (2) comparison of trained real time face images via two classifiers such as CNN and VGG16 with transfer learning (3) Results comparison with respect to the metrics like recall, accuracy, precision, F1-score, and precision. The CNN and

VGG16 classifiers recognize faces in real-time with higher accuracy. Both of the classifiers are performed in sequential manner. For VGG16 model is performed based on the transfer learning. Transfer learning intends to extract information from a number of sources tasks and applies it to target task. So VGG16 gives improved accuracy than the CNN classifier. Classifiers are implemented with 1056 face images of24 different persons. The proposed system can successfully recognize 24 different person faces which are which could be useful in searching suspects as its accuracy is much higher than other methods. The proposed VGG16 classifier gives higher values of 99% of macro-average precision, 99% of macro-average recall, 99% of macro-average f1-score and 99.37% accuracy results for real time face images. The major limitation of this work is that it ignores illumination impact in human face recognition which is left as scope of the future work. In future, in various human face sections, this concept can be applied for detecting facial expression for more security.

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgment

There is no funding agency supporting our Research.

References

[1] S. Gupta, T. Gandhi, "Identification of Neural Correlates of Face Recognition Using Machine Learning Approach," in Advances in Intelligent Systems and Computing, Springer Verlag: 13–20, 2020, doi:10.1007/978-981-13-8798-2_2.

[2] B.K. Tripathi, "On the complex domain deep machine learning for face recognition," Applied Intelligence, **47**(2), 382–396, 2017, doi:10.1007/s10489-017-0902-7.

[3] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S.Z. Li, T. Hospedales, "When Face Recognition Meets with Deep Learning: An Evaluation of Convolutional Neural Networks for Face Recognition," in Proceedings of the IEEE International Conference on Computer Vision, Institute of Electrical and Electronics Engineers Inc.: 384–392, 2016, doi:10.1109/ICCVW.2015.58.

[4] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.Y. Chang, T. Sainath, "Deep Learning for Audio Signal Processing," IEEE Journal on Selected Topics in Signal Processing, **13**(2), 206–219, 2019, doi:10.1109/JSTSP.2019.2908700.

[5] P. Apoorva, H.C. Impana, S.L. Siri, M.R. Varshitha, B. Ramesh, "Automated criminal identification by face recognition using open computer vision classifiers," in Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019, Institute of Electrical and Electronics Engineers Inc.: 775–778, 2019, doi:10.1109/ICCMC.2019.8819850.

[6] S.M. Bah, F. Ming, "An improved face recognition algorithm and its application in attendance management system," Array, **5**, 100014, 2020, doi:10.1016/j.array.2019.100014.

[7] H. Mady, S.M.S. Hilles, "Face recognition and detection using Random forest and combination of LBP and HOG features," in 2018 International Conference on Smart Computing and Electronic Enterprise, ICSCEE 2018, Institute of Electrical and Electronics Engineers Inc., 2018, doi:10.1109/ICSCEE.2018.8538377.

[8] H. Ab., A. A., E. E., "A Real-Time System for Facial Expression Recognition using Support Vector Machines and k-Nearest Neighbor Classifier," International Journal of Computer Applications, **159**(8), 23–29, 2017, doi:10.5120/ijca2017913009.

[9] L. He, H. Li, Q. Zhang, Z. Sun, Dynamic Feature Learning for Partial Face Recognition.

[10] Y. Long, F. Zhu, L. Shao, J. Han, "Face recognition with a small occluded training set using spatial and statistical pooling," Information Sciences, **430**–**431**, 634–644, 2018, doi:10.1016/j.ins.2017.10.042.

[11] B. Lahasan, S. Lebai Lutfi, I. Venkat, M.A. Al-Betar, R. San-Segundo, "Optimized symmetric partial facegraphs for face recognition in adverse conditions," Information Sciences, **429**, 194–214, 2018, doi:10.1016/j.ins.2017.11.013.

[12] T.X. Jiang, T.Z. Huang, X. Le Zhao, T.H. Ma, "Patch-Based Principal Component Analysis for Face Recognition," Computational Intelligence and Neuroscience, **2017**, 2017, doi:10.1155/2017/5317850.

[13] P. Shubha, M. Meenakshi, "Human face recognition using local binary pattern algorithm - Real time validation," in Advances in Intelligent Systems and Computing, Springer: 240–246, 2020, doi:10.1007/978-3-030-37218-7_28.

[14] F. Tabassum, M. Imdadul Islam, R. Tasin Khan, M.R. Amin, "Human face recognition with combination of DWT and machine learning," Journal of King Saud University - Computer and Information Sciences, 2020, doi:10.1016/j.jksuci.2020.02.002.

[15] Y. Shin, I. Balasingham, "Comparison of hand-craft feature based SVM and CNN based deep learning framework for automatic polyp classification," in Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, Institute of Electrical and Electronics Engineers Inc.: 3277–3280, 2017, doi:10.1109/EMBC.2017.8037556.

[16] S. Guo, S. Chen, Y. Li, "Face recognition based on convolutional neural network & support vector machine," in 2016 IEEE International Conference on Information and Automation, IEEE ICIA 2016, Institute of Electrical and Electronics Engineers Inc.: 1787–1792, 2017, doi:10.1109/ICInfA.2016.7832107.

[17] M. Coskun, A. Ucar, O. Yildirim, Y. Demir, "Face recognition based on convolutional neural network," in Proceedings of the International Conference on Modern Electrical and Energy Systems, MEES 2017, Institute of Electrical and Electronics Engineers Inc.: 376–379, 2017, doi:10.1109/MEES.2017.8248937.

[18] Z. Lu, X. Jiang, A. Kot, "Deep Coupled ResNet for Low-Resolution Face Recognition," IEEE Signal Processing Letters, **25**(4), 526–530, 2018, doi:10.1109/LSP.2018.2810121.

[19] S. Almabdy, L. Elrefaei, "Deep Convolutional Neural Network-Based Approaches for Face Recognition," Applied Sciences, **9**(20), 4397, 2019, doi:10.3390/app9204397.

[20] W. Passos, I. Quintanilha, G. Araujo, "Real-Time Deep-Learning-Based System for Facial Recognition," Sociedad Brasileira de Telecomunicacoes, 2018, doi:10.14209/sbrt.2018.321.

[21] S. Saypadith, S. Aramvith, "Real-Time Multiple Face Recognition using Deep Learning on Embedded GPU System," in 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2018 - Proceedings, Institute of Electrical and Electronics Engineers Inc.: 1318–1324, 2019, doi:10.23919/APSIPA.2018.8659751.

[22] G. Nam, H. Choi, J. Cho, I.-J. Kim, "PSI-CNN: A Pyramid-Based Scale-Invariant CNN Architecture for Face Recognition Robust to Various Image Resolutions," Applied Sciences, **8**(9), 1561, 2018, doi:10.3390/app8091561.

[23] Y. Wu, T. Hassner, K. Kim, G. Medioni, P. Natarajan, "Facial Landmark Detection with Tweaked Convolutional Neural Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, **40**(12), 3067–3074, 2018, doi:10.1109/TPAMI.2017.2787130.

[24] K.B. Pranav, J. Manikandan, "Design and Evaluation of a Real-Time Face Recognition System using Convolutional Neural Networks," in Procedia Computer Science, Elsevier B.V.: 1651–1659, 2020, doi:10.1016/j.procs.2020.04.177.

[25] G. Lou, H. Shi, "Face image recognition based on convolutional neural network," China Communications, **17**(2), 117–124, 2020, doi:10.23919/JCC.2020.02.010.

[26] L. Wen, X. Li, L. Gao, Y. Zhang, "A New Convolutional Neural Network-Based Data-Driven Fault Diagnosis Method," IEEE Transactions on Industrial Electronics, **65**(7), 5990–5998, 2018, doi:10.1109/TIE.2017.2774777.

[27] J. Ding, B. Chen, H. Liu, M. Huang, "Convolutional Neural Network with Data Augmentation for SAR Target Recognition," IEEE Geoscience and Remote Sensing Letters, **13**(3), 364–368, 2016, doi:10.1109/LGRS.2015.2513754.

[28] U.R. Acharya, S.L. Oh, Y. Hagiwara, J.H. Tan, M. Adam, A. Gertych, R.S. Tan, "A deep convolutional neural network model to classify heartbeats," Computers in Biology and Medicine, **89**, 389–396, 2017, doi:10.1016/j.compbiomed.2017.08.022.

[29] S. Tammina, "Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images," International Journal of Scientific and Research Publications (IJSRP), **9**(10), p9420, 2019, doi:10.29322/IJSRP.9.10.2019.p9420.

[30] Face Recognition on Olivetti Dataset | Kaggle, Apr. 2021.