PalArch's Journal of Archaeology
of Egypt / Egyptology

OCCLUDED FACIAL EXPRESSION RECOGNITION USING ALEXNET

Dr. K. Srinivas[1], T. Swathi[2], Dr. A. Hari Prasad Reddy[3]
[1,3]Professor, Department of CSE, Geethanjali college of Engineering and Technology, Hyderabad, Telangana, India.
[2]Department of CSE, Geethanjali college of Engineering and Technology, Hyderabad, Telangana, India.

Dr. K. Srinivas ,T. Swathi ,Dr.A.Hari Prasad Reddy ,Occluded Facial Expression Recognition Using Alexnet, Palarch's Journal Of Archaeology Of Egypt/Egyptology 17(12). ISSN 1567-214x.

Abstract:

In artificial intelligence and machine learning fields, facial emotion recognition (FER) is receiving an attention to research as it is an essential commercial and academic potential. By using facial images, this analysis is performed while multiple sensors can use for processing the FER. In interpersonal communication, one of the key channels is that the visual expressions. In the field of FER, a details analysis of research is performed over the past few decades in this paper. The conventional approaches of FER are listed out in addition to the describing representative categories of FER systems and their key algorithms. By using deep networks, the approaches of deep-learning-based FER are presented that allow the "end-to-end" learning. An approach of up-to-date hybrid deep-learning is also focused in this research and it integrates an individual frame's spatial features with the convolution neural network (CNN) for consecutive frames' temporal features. Consequently, the publicly available assessment metrics with a brief analysis and defining of comparing the benchmark results have been provided for a quantitative comparison of FER studies. The existing work carried out was using basic CNN variant which couldn't produce efficient results for larger datasets. This paper proposes a facial recognition system using ALEXNET which produced better results than the existing basic CNN and GoogleNet.

Introduction

In human communication, facial expression is a significant non-verbal way of expecting intentions. In the whole information sharing process, facial expression has tended to play a vital role. In different fields of pattern recognition, computer vision, and psychology, the study of Facial Expression Recognition (FER) has achieve progress attention. In multiple domains, FER has wide applications, including human-computer interaction, virtual reality, augmented reality, advanced systems of driver assistance, education and entertainment.

Humans commonly use various signs to convey their thoughts, such as facial expressions, movements of the hand and speech. Up to 55% of human communications are facial expressions, while a nominal 7% of emotional expressions are assigned to other means such as oral language. As the input of the emotion recognition system, various types of data can be used. Recognition of expression and recognition of emotions are related, but unique. The mainstream and promising input type is the human facial image, because it can provide ample information for research on expression recognition.

There are different algorithms for face recognition: template matching, facial recognition with geometric features, facial recognition with two-dimensional Fourier transformation, the face recognition with neural networks, the Hierarchical graph matching and facial recognition with facial faces.

There is a list of indicators used in various FER devices, such as the camera, eye-tracker, electrocardiogram (ECG), electromyography (EMG) and electroencephalography (EEG). However, due to its convenience and easy use the camera has to be the most common sensor.

Assessment of facial expressions involves both facial motion assessment and face recognition. There are three stages to the general approach to automated facial expression analysis (AFEA): face acquisition, extraction and presentation of facial data, and identification of facial expression.

Face acquisition is defined as a control mechanism for determining the face region for input images or sequences instantly. To detect the face in the first frame and control the face region for remaining of the video series, it can be used as a detector for each frame. To manage the broad head movements, the scene analysis, head tracking, and head finder can be added for an analysis of facial expression device.

The next step after the face is located is to remove the facial effects caused by facial expressions and reflect them. For expression analysis, two kinds of approaches are used primarily in extraction of facial features such as appearance-based approaches and geometric feature-based methods. In the geometric facial features including nose, eye-brows, mouth, eyes, etc., the positions and forms of facial components are presented. The extraction of facial feature points or facial components is done for shaping a feature vector that reflects the face geometry. To obtain a feature vector, image filters like Gabor wavelets are implemented to either particular regions or entire face in a face image by using the appearance-based methods. Before making the feature extraction or feature representation in prior to the phase of expression recognition, the removal of effects of in-plane head rotation and various scales of the faces can be made based on face normalization by relying on the different techniques of facial feature extraction.

The detection of facial emotion is the last phase of AFEA structures. The classification of facial changes is possible to happen as prototypic emotional expressions or facial action units. In this chapter, a recognition technique is classified as sequence-based or frame-based based on whether the temporal data is 19 Identification of Facial Expression 489 Fig. 19.2. By emotions like 1.Unhappiness; 2.Happiness; 3.Surprise; 4.Fear; 5.Disgust; and 6.Frustration.

Literature Survey

A multilevel haar wavelet-based method was proposed by Goyani, & Patel to extract presence characteristics at two different scales from protuberant face regions. First, via the Viola-Jones force object finder, the system segments the most useful geometric modules such as mouth, eye, eyebrows etc. Haar functions are derived from segmented modules. Of all the logistic regression methods, one was used for classification. The Haar features are also clearly computed and can effectively signify the signal in the low dimension, but the signal energy is reserved. The efficiency of the scheme presented was evaluated using available datasets such as CK, JAFFE and TFEID..

A Stationary Wavelet Transform (SWT) was proposed by Qayyum et al. to extract FER features in both the domains of spatial and spectral due to its decent localization features. In addition, as these subbands provide muscle measure details for mainstream FE, a mixture of vertical and horizontal sub-bands of SWT was practiced. Additionally, feature dimensionality was reduced by implementing the Discrete Cosine Transform (DCT) on these sub-bands. Then the selected features are transformed into the Back Propagation (BP) algorithm-trained Feed Forward Neural Network (FFNN). By using usable datasets such as JAFFE, CK+ and MS-Kinect datasets, the presented scheme output was evaluated.

Radlak and Smolka suggested a combination of two strategies for facial identification, such as tree models and gradient boosting. First, via the Dlib library, face detection was achieved. Then the technique of gradient boosting was used to observe facial landmarks. If Dlib fails, the recognized face normalization was completed by Affine Transformation (AT), which omitted face contour, using the tree model approach. To minimize its properties in facial classification, the removal of the contextual nearby detected face will be prevented. In order to generate function vectors for classification, multi-scale patches were extracted as the center point before perceived facial landmarks. For each area within this piece, the uniform local binary pattern (ULBP) histogram was determined. Finally, in order to construct a high-dimensional function vector, all histograms were merged. For feature extraction, Random Frog (RF) was used and fast feature selection was used. Finally, the one-on-one" technique of Support Vector Machine (SVM) for multiclass classifiers was used.

Kamarol et al. used the 3D approach and the proposed structure that requires less computational cost of contact between time and space. For feature extraction, Spatio-Temporal Texture Map (STTM) was applied to capture continuous and perfect movement of facial expressions that provide unique information in turn. It creates a textured map in 2D. By providing precise temporal and spatial variations of face expressions, it has very low computational costs. First the viola and jones face detector senses the face and then crops out the context in the proposed system. After that by using spatiotemporal data obtained from the 3-dimensional Harris corner function, STTM extracted and modeled facial features. In the form of histograms, features are extracted and represented using a block-based process. The classifier of help vector machines categorizes features into emotions.

A method implementing Histograms of Directed Gradients (HOG) in the FER system was proposed by Carcagnì et al. For one picture, HOG was a dense feature extraction process. It extracts all regions of interest via gradients from the image. This was a pretty fast technique. The paper explains how to set HOG perimeters so that the features of facial expression can be differentiated at their best. The system is split into 3 phases by an algorithmic pipeline pattern. In the 1st level, frontal face input in the device that then performs face registration after HOG has been applied to the face. The technique of Support Vector Machine (SVM) was applied for classification. HOG perimeters are then verified on datasets in Step 2; the sequence of input faces begins with a neutral face and ends with an expressive face.

Imran et al. suggested a novel method of recognition of speech by representing images in the form of high-order two-dimensional Gaussian Hermite orthogonal moments (GHMs). On the basis of instants with a high power of discrimination, a set of characteristics is chosen. In order to obtain differentially expressive elements of the instances, the unequal GHMs are cast on the current expression-invariants subspace using the relation between normal faces. To define an expression using the SVM classifier, features obtained from the differentially expressive elements of the instances and discriminatory instances are added. Experiments were performed on widely used databases, which resulted in overall expression recognition battering performance than related or current methods.

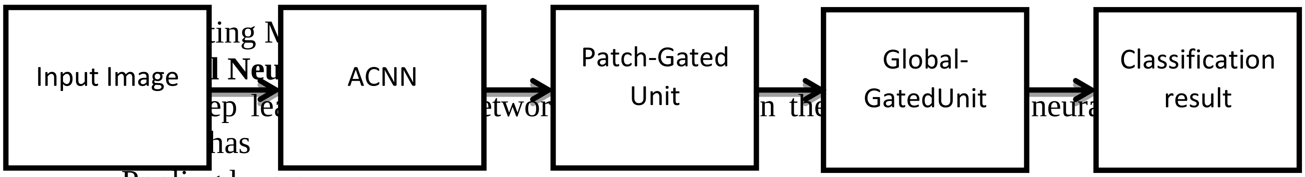
Saini et al., was applied the facial expression recognition techniques based on Principal Component Analysis (PCA) with Singular Value Decomposition (SVD). Experiments are conducted using images from a real database. For face classification, a Support Vector Machine (SVM) classifier was used. Emotion detection was conducted using the SURF (Speed Up Robust Feature) Regression Algorithm. The five main emotions to be identified are widely accepted: furious, happy, sad, disgust and surprise along with neutral.

K Kamakshaiah et al [21], proposed recognition of objects such as fishes has drawn more attentionwhile submerged pictures are showing some difficulty due to their poor picture qualitywhich also includes rough background surfaces when compared to general images.Medicines prepared from fishes help in curing different diseases to reduce the healthissues in the present world (for example, rheumatism problems, gel for wounds,bandages, etc). In our proposed method we are projecting a deep neural network thatsupports recognition of fishes to acquire their count, species and medical usage.

Existing system

Block diagram:

The existing technique's block diagram is shown in Figure 1. It contains Input image, Convolution Neural Network with Accumulation mechanism, Patch gated unit, Global gated unit blocks respectively.



- Pooling layers
- ReLU layers
- Convolutional layers
- A fully connected layer

An architecture of classic CNN is shown below:

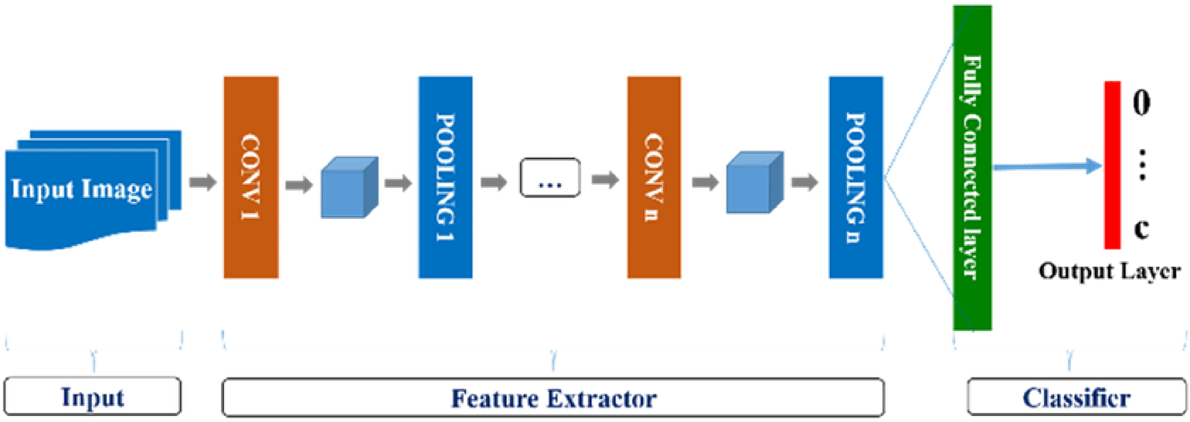


Figure 2: CNN Architecture

Input ->Convolution ->ReLU ->Convolution ->ReLU ->Pooling -> ReLU ->Convolution ->ReLU ->Pooling ->Fully Connected

Based on the image features extraction, a CNN is operated that replaces the requirement of manual extraction of features. The network trains on a set of images as there no preparation for features. The deep learning models are precise extremely for computer vision tasks. The functional detection can learn by CNNs through the tens or hundreds of hidden layers. By each layer, the learned characteristics' complexity is increased.

- Initiates with an input image
- For creating a feature map, many different filters are applied.
- A ReLU function implements for increasing non-linearlity.
- A pooling layer is applied to each feature map.
- The pooling images flatten into one long vector.
- The vector is applied as an input into a fully connected artificial neural network.

- Through the network, the features are processed. After that, the “voting” of the classes is provided by the fully connected layer.
- For many epochs, the network trains through the backpropagation and forward propagation. Until a well-defined neural network with feature detectors and trained weights is achieved, this is repeated.

ACNN:

The way people interpret the face expression is imitated by the Convolution Neural Network with Attention Mechanism (ACNN). The facial expressions can be understandable for humans intuitively based on certain patches of the forehead. By the face with symmetrical portion (ex, the lower right cheek) or other highly linked facial regions (ex, regions around the eyes or mouth), the expression may be judged when some parts of the face are blocked (ex, the lower left cheek). This intuition inspires the perceiving of blocked facial patches by ACNN immediately and attentive towards the insightful and unblocked patches. The approach with a main concept is demonstrated by Fig.1. By the significance of an adaptive weight or unhindered-ness, each Gate Unit in ACNN is learned.

Different facial regions of ACNN’s interest are considered such as: (1) pACNN crops interest patches from the last convolution feature maps based on the corresponding feature point locations. By the unhindered-ness determined from the patch itself for each patch, a Patch-Gated Unit (PG-Unit) is learned to weigh the patch’s local representation. (2)gACNN integrates the local and global representations simultaneously. To learn and weigh global representation, a Global-Gated Unit (GG-Unit) is adopted by gACNN in addition to the local weighted functionality.

It is considered to be a primary version of this work. In the presence of actual occlusions, a dataset of facial expression released and expanded outcomes are presented in the technical descriptions of facial area decomposition with more datasets and further comparisons. The summary of this work with achievements can be demonstrated below:

1) The facial expressions are extracted from partly occluded faces by using a convolutional neural network with attention mechanism (ACNN). The occluded areas of the face will be perceived by ACNN automatically and the most insightful and un-blocked regions have been focused.

2) The Gate-Unit or the critical component of ACNN is successful based on the visualized results which show that in perceiving the occluded facial patches. For an insightful and unblocked one, a low weight and a high weight is learned by PG-Unit for pACNN for a blocked area. The efficiency of FER under occlusion is improved by gACNN further by embedding the PG-Unit and GG-Unit.

Patch Based ACNN (pACNN)

The capturing of regional facial muscle distortions is required for classifying the facial expressions into different categories. To focus on local patches that are representative and discriminatory, pACNN is designed. In pACNN, two essential systems are included such as perception of occlusion and region decomposition. These are presented in detail below:

1) Region Decomposition: As expressions are facial activities which are invoked by sets of muscle motions, facial expression is derived in particular facial regions. For identifying the facial expression, the encoding and standardization of parts relevant to expression is advantageous. By categorizing the face into multiple local patches, the position of occlusions is determined using the region decomposition.

Global-Local Based ACNN (gACNN)

Owing to the incorporation of prior knowledge of facial expression, pACNN is effective in learning the representations local facial characteristics with an attention mechanism. Those facial patches in pACNN may ignore some complementary information displayed in facial images. To improve the performance of FER, the integration with global representation in the occlusions presence is expected.

1) Full Face Region Integration: gACNN is considered the global face region while focusing on local facial patches. From images, global context signals and local details have inferred simultaneously in the Global-Local Attention technique [34]. The diversity among the learned features is promoted by gACNN which can view as a type of ensemble learning. In VGG16 net, the whole face feature maps are encoded from conv4-2 to conv5-2. The encoded region with the size of 512×14×14 is obtained based on the feature maps of 512×28×28.

2) Geo-Gated Unit (GG-Unit): The GG-Unit is integrated into gACNN to weight the global facial representation automatically. The GG-Unit with a detailed structure is shown in fig 2 which shows in the lower-most blue dashed rectangle. As the vector-shaped global representation of the two branches in GG-Unit, the input function maps are encoded by the first branch. An Attention Net is contained in the second branch which acquires a scalar weight for denoting the contribution of global facial representation. The global representation weights by the computed weight.

Proposed System

GoogLeNet

GoogLeNet is a 22 layer deep CNN, which 2014 at ILSVRC2014 (ImageNet Challenge) first on the so-called inception architecture based network was presented [20]. To make more efficient use of the possibility of parallelization of modern computer architectures (both in the CPU and in the GPU area)

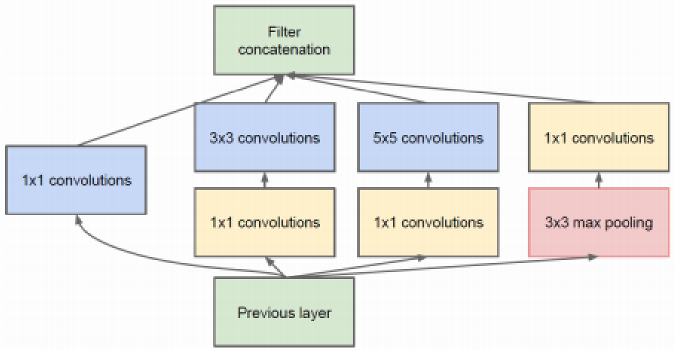


Figure 3: Schematic structure of an inception module

Folding masts of different sizes (1 ×1, 3× 3 and 5× 5), as well as max pooling within one shift. So the network will not expanded only in depth, but also in width. As in previous CNN architectures, the max pooling unit is said to be implement the Hebbian learning rule. At the exit of one, in Figure 3 will be shown, inception layer the expenses from the individual units are concatenated.

About the computational effort caused by the rather large 5 ×5 filters to keep it low, the authors decided to insert 1 ×1 filter units. The depth of previous feature maps before processing reduce with the 3 ×3 and 5 ×5 filters. At the ImageNet Challenge 2014, the team set up the figure 4, consisting of Inception blocks. All in Figure 6 are additional on the right side of the picture Recognize output layers. Just like the exit of the entire network two fully connected layers can be seen here, which are in a Softmax Classifiers.

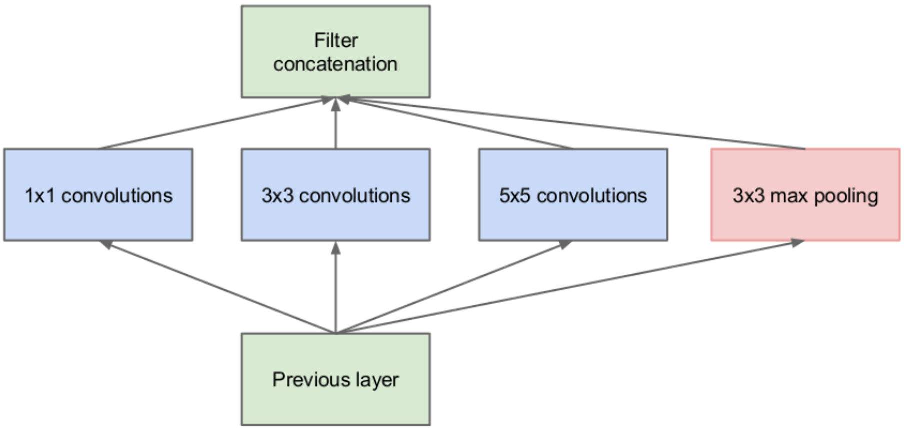


Figure 4: Naive version of an inception block

The authors introduced these additional layers, because despite the extensive use of ReLU units the gradients of the output layer in the back propagation step not through the already very deep network can propagate back to the input layer.

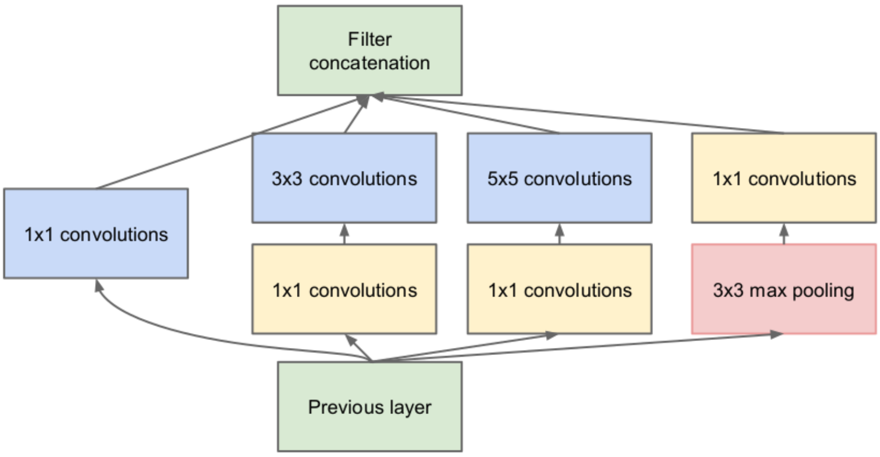


Figure 5: Inception block with dimension reduction

Augmentation in which random image sections from the training images the size 28× 28 px were cut out. The additional costs regarding the computational effort due to this preprocessing are in the column Cost shown in the table.

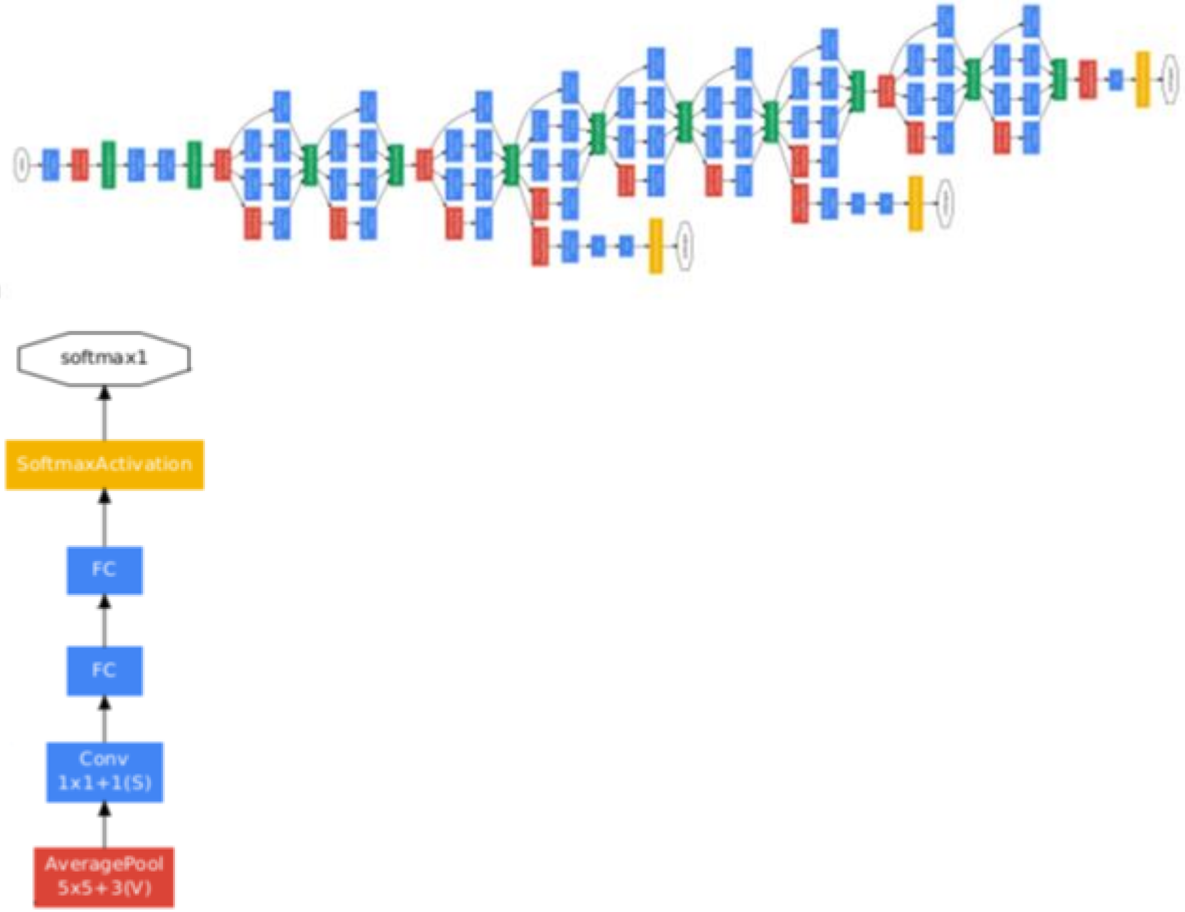


Figure 6: Structure of GoogLeNet

AlexNet:

In 2012 (ILSVRC), AlexNet was created and designed by Krizhevsky for the ImageNet Large Scale Visual Recognition Competion. To make the classification of 1.2 million images into 1000 classes, AlexNet or a deep convolutionary neural network (CNN) was utilized in the ImageNet challenge [30]. AlexNet generates best results substantially than the previous models like LeNet. The differentiations between AlexNet and previous versions are showed by numeric layers and parameters. Five CONV layers are contained in AlexNet, some of which are accompanied by max-pooling layers, three linked layers, and a final 1000-way softmax.

AlexNet has the total trainable parameters number is around 60 million. In contrary to this, LeNet is contained two convolutionary layers, followed by three fully connect layers and two pooling layers. Approximately, the total trainable parameters of LeNet is 60,000. The non-linear (ReLU) function uses in AlexNet whereas LeNet is used the sigmoid logistic function. To reduce the overfitting in the fully linked layers, a “Dropout” form of regularization is used by AlexNet and it is the last distinction. While designing LeNet, this principle is not used. Before describing the topology of AlexNet, some more terms used in the model are require to explain. In the following subsections, the features of AlexNet are listed in detail.

Local Response Normalization

After implementing the ReLU in CONV layers, local normalization is used by the topology of AlexNet while normalization of input is not require for the activation function of ReLU to prevent the saturating region. By using the below equation, the normalized activity of response $b^{i}_{x,y}$is determined:

$$b^{i}_{x,y} = \frac{a^{i}_{x,y}}{(k + \alpha \sum_{j=max(0,i-n/2)}^{min(N-1,i+n/2)} (a^{j}_{x,y})^{2})^{\beta}}$$

Where, N is the total number of features in the layer and $a^{i}_{(x,y)}$ is the neurons activity which computes by using the filter i at (x, y). By using a validation test, the constant hyper parameters n, k, ∝, and β values are computed. In the experiments, n=5, k=2, β = 0.75, and ∝ ∝ = 10^{-4}. In the same position, the summation runs over n adjacent feature maps in the above equation.

Overall Architecture

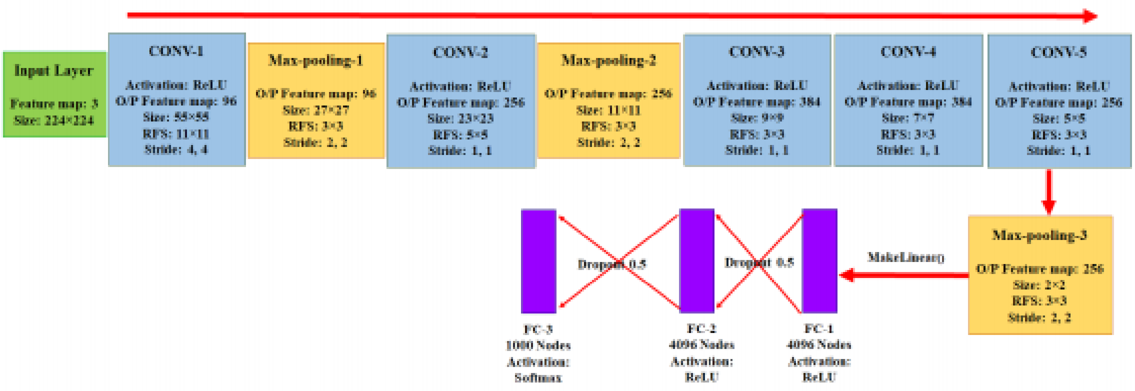


Figure 7: An illustration of the AlexNet architecture.

In Figure 7, the AlexNet layers with overall architecture and configuration are shown. A subset of generated activation maps or feature maps is sent as input to the next layer and it can be viewed in each layer. By using the linear unit in ReLU, fully connected layer is rectified and activation function utilizes after each CONV layer. The response normalization layers are applied on the first and second CONV layers. The max pooling layers are followed after the fifth CONV layer and each response normalization layer. The minimization of overfitting is the most essential function in this model by applying the dropout regularization technique after completely linked layers. From 224×224×3 to 2×2×256, the number of input features is reduced by this configuration after the third max pooling layer.

An input image of 224×224×3 is considered in the first CONV layer and the input is filtered out with 96 Receptive Field Size (RFS) feature maps of 11×11×3 with a 4 pixel stride size (distance between the receptive fields of neurons' centers in a feature map). The CONV (second layer input) layer's first layer output is computed as 224/4×224/4×96 = 55×55×96. After the response normalization layer and max pooling layer, the first layer's output is sent to the second CONV layer. The second CONV layer is filtered the data with 256 feature maps of size 5×5×48. The last three CONV layers are connected to each other without any normalization or pooling layers. After the fifth CONV layer, the output is transferred to the new max pooling layer. The maximum pooling layer's outcome is transferred to the completely connected layers that have 4096 neurons at each layer.

AlexNet– Classification with Deep Convolutional Neural Networks

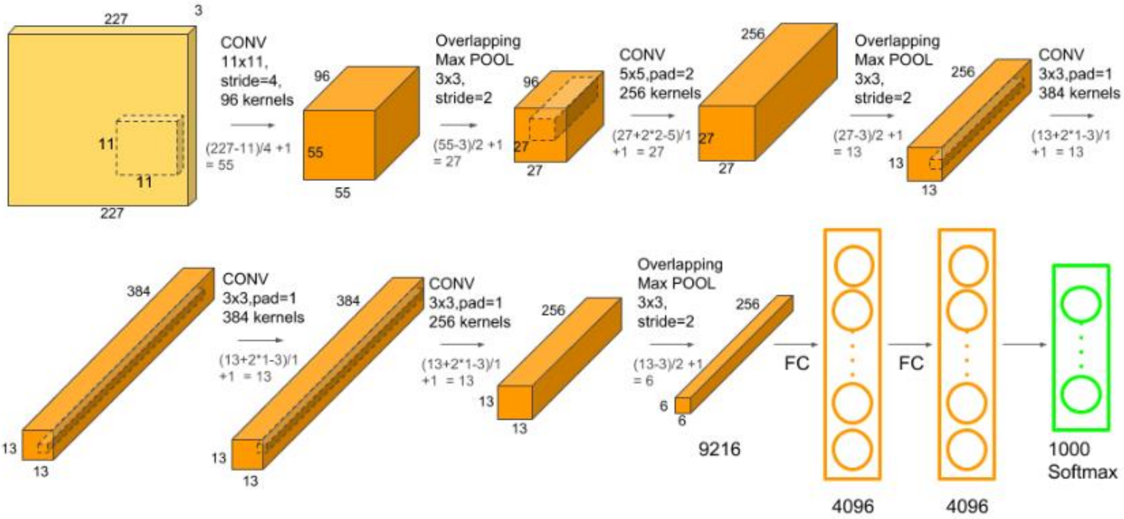


Figure 8:AlexNet consists of 5 Convolutional Layers and 3 Fully Connected Layers.

On the machine learning field, a large impact had involved by the AlexNet which is the name of a convolutionary neural network specifically in the deep learning application to machine vision. The competition of ImageNet LSVRC-2012 was won famously by a large margin (15.3 percent Vs 26.2 percent (second place) error rates). An architecture in a network is similar to the LeNet of YannLeCun. But, it was deeper with stacked convolutionary layers and more filters per layer. This network includes momentum SGD, ReLU activation, data increase, dropout, max pooling, convolutions of 11×11, 5×5, and 3×3. The activations of ReLU are attached after each fully-connected and convolutional layers.

Results and Discussion

Convolution Neural Networks:

Train 1 (Minimal Occlusion):

The training process requires set of layers and options. Reach the computational architecture of the neural network. When regression is not an easy thing to do, add a regression layer at the end of the network. Constructing a collection of layers requires the input image unique size.

Build a variety of training solutions for a network with a stochastic pitch. By a factor of 0.2, analysis reduces per 5 epochs. The maximum number of training epochs is set to 50 and a mini-batch of 64 observations is added every time. Switch on the course of training.

Set the default parameters with momentum for the stochastic descent. Set the maximum epoch number to 20 and start training at 0,0001 initial level.

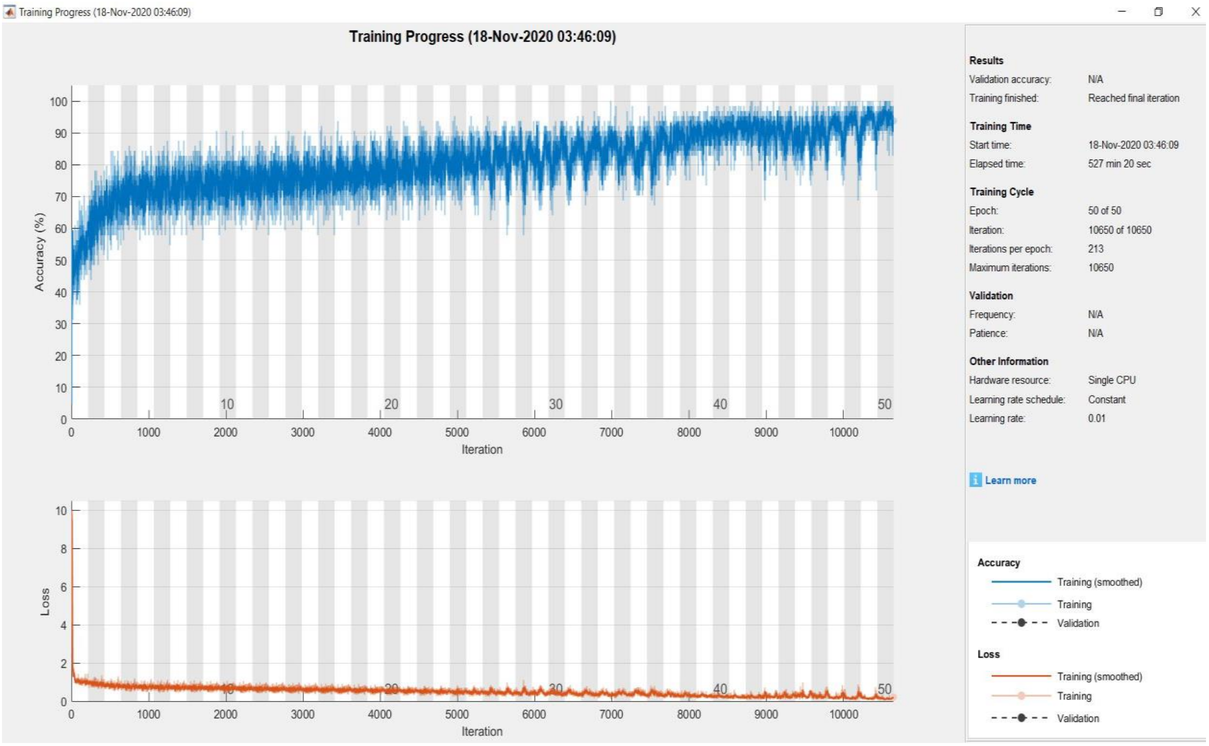
Using increased image data to train a convolutionary neural network. Data increase helps prevent the network from overrunning and storing exact training picture information.

Reach the computational architecture of the neural network. Include a regression layer at the end of the network for problems with regression.

The training process includes generally two graphs which are accuracy and loss. If the training process reaches to maximum epoch the accuracy graph reach to 100% and the loss graph reaches to 0%.

In the both accuracy and loss graphs consists 3 lines. The thick line represents the training(smoothed). Thin line with dots represents the training and the dash line represents the validation.

The figure 9 shows thetraining process forminimalocclusion face images using convolution neural network.



Results

Validation accuracy:	N/A
Training finished:	Reached final iteration

Training Time

Start time:	18-Nov-2020 03:46:09
Elapsed time:	527 min 20 sec

Training Cycle

Epoch:	50 of 50
Iteration:	10650 of 10650
Iterations per epoch:	213
Maximum iterations:	10650

Validation

Frequency:	N/A
Patience:	N/A

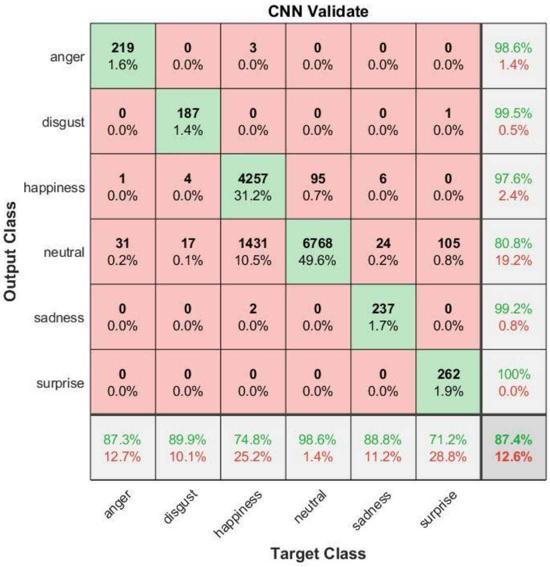
Other Information

Hardware resource:	Single CPU
Learning rate schedule:	Constant
Learning rate:	0.01

Figure 9: Training process1 for CNN.

Validate 1 (Minimal Occlusion):

The CNN uncertainty matrix is shown in figure 10. Overall validation1 accuracy of CNN is 87.4%.

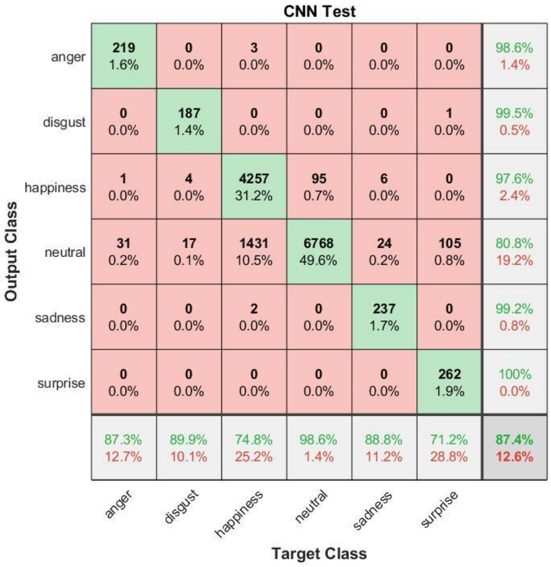


	anger	disgust	happiness	neutral	sadness	surprise	
anger	219 1.6%	0 0.0%	3 0.0%	0 0.0%	0 0.0%	0 0.0%	98.6% 1.4%
disgust	0 0.0%	187 1.4%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	99.5% 0.5%
happiness	1 0.0%	4 0.0%	4257 31.2%	95 0.7%	6 0.0%	0 0.0%	97.6% 2.4%
neutral	31 0.2%	17 0.1%	1431 10.5%	6768 49.6%	24 0.2%	105 0.8%	80.8% 19.2%
sadness	0 0.0%	0 0.0%	2 0.0%	0 0.0%	237 1.7%	0 0.0%	99.2% 0.8%
surprise	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	262 1.9%	100% 0.0%
	87.3% 12.7%	89.9% 10.1%	74.8% 25.2%	98.6% 1.4%	88.8% 11.2%	71.2% 28.8%	87.4% 12.6%

Figure 10: Validation1 Confusion matrix of CNN.

Test 1 (Minimal Occlusion):

The CNN confound matrix appears in Figure 11. Overall CNN test1 accuracy is 87.4%.

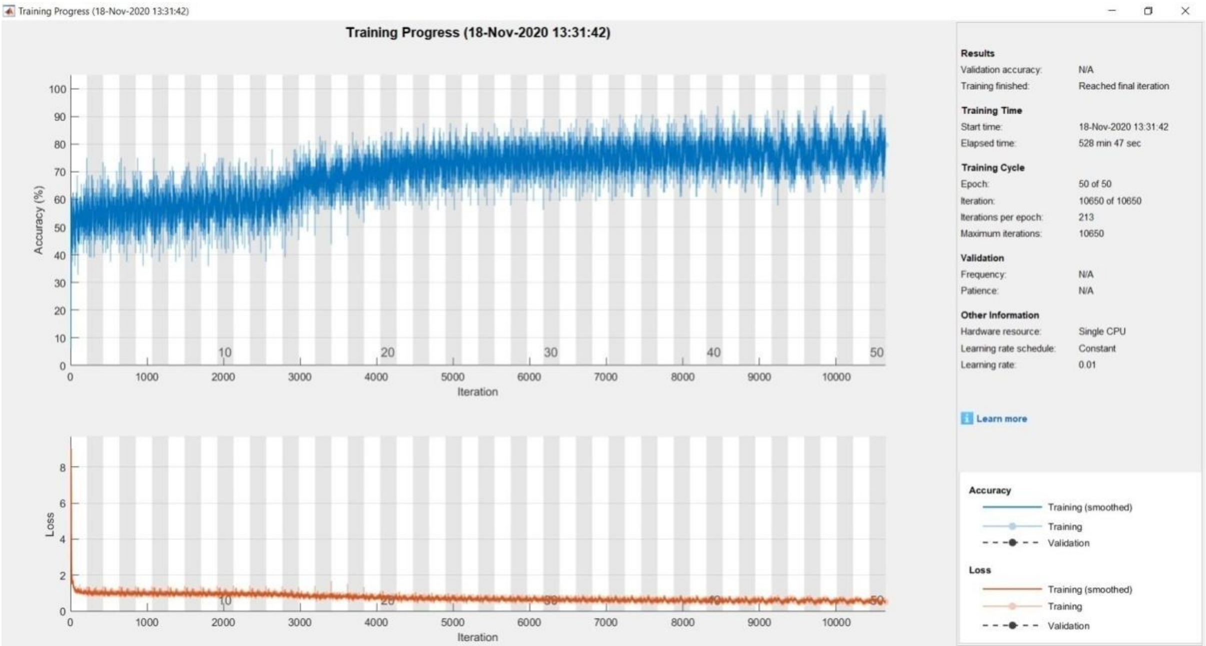

CNN Test

Output Class \ Target Class	anger	disgust	happiness	neutral	sadness	surprise	
anger	219 1.6%	0 0.0%	3 0.0%	0 0.0%	0 0.0%	0 0.0%	98.6% 1.4%
disgust	0 0.0%	187 1.4%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	99.5% 0.5%
happiness	1 0.0%	4 0.0%	4257 31.2%	95 0.7%	6 0.0%	0 0.0%	97.6% 2.4%
neutral	31 0.2%	17 0.1%	1431 10.5%	6768 49.6%	24 0.2%	105 0.8%	80.8% 19.2%
sadness	0 0.0%	0 0.0%	2 0.0%	0 0.0%	237 1.7%	0 0.0%	99.2% 0.8%
surprise	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	262 1.9%	100% 0.0%
	87.3% 12.7%	89.9% 10.1%	74.8% 25.2%	98.6% 1.4%	88.8% 11.2%	71.2% 28.8%	87.4% 12.6%

Figure 11: Test1 confusion matrix for CNN.

Train 2 (Medium Occlusion):

The figure 12 shows the training process for medium occlusion face images using convolution neural network.



Results
Validation accuracy: N/A
Training finished: Reached final iteration

Training Time
Start time: 18-Nov-2020 13:31:42
Elapsed time: 528 min 47 sec

Training Cycle
Epoch: 50 of 50
Iteration: 10650 of 10650
Iterations per epoch: 213
Maximum iterations: 10650

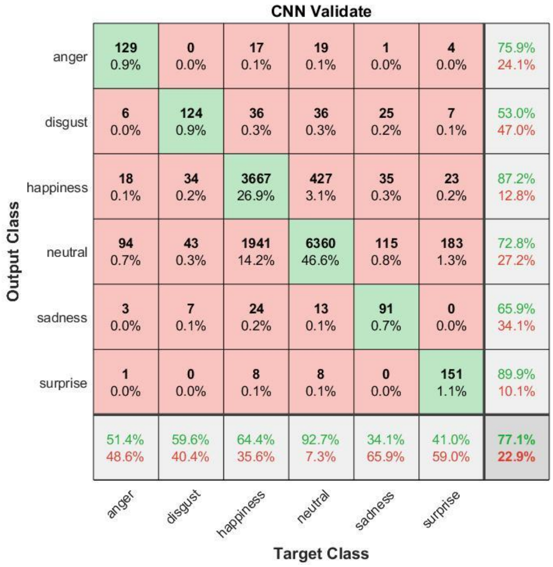
Validation
Frequency: N/A
Patience: N/A

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

Figure 12: Training process for CNN.

Validate 2 (Medium Occlusion):

The CNN uncertainty matrix is shown in figure 13. Overall validation2 accuracy of CNN is 77.1%.



CNN Validate

	anger	disgust	happiness	neutral	sadness	surprise	
anger	129 0.9%	0 0.0%	17 0.1%	19 0.1%	1 0.0%	4 0.0%	75.9% 24.1%
disgust	6 0.0%	124 0.9%	36 0.3%	36 0.3%	25 0.2%	7 0.1%	53.0% 47.0%
happiness	18 0.1%	34 0.2%	3667 26.9%	427 3.1%	35 0.3%	23 0.2%	87.2% 12.8%
neutral	94 0.7%	43 0.3%	1941 14.2%	6360 46.6%	115 0.8%	183 1.3%	72.8% 27.2%
sadness	3 0.0%	7 0.1%	24 0.2%	13 0.1%	91 0.7%	0 0.0%	65.9% 34.1%
surprise	1 0.0%	0 0.0%	8 0.1%	8 0.1%	0 0.0%	151 1.1%	89.9% 10.1%
	51.4% 48.6%	59.6% 40.4%	64.4% 35.6%	92.7% 7.3%	34.1% 65.9%	41.0% 59.0%	77.1% 22.9%

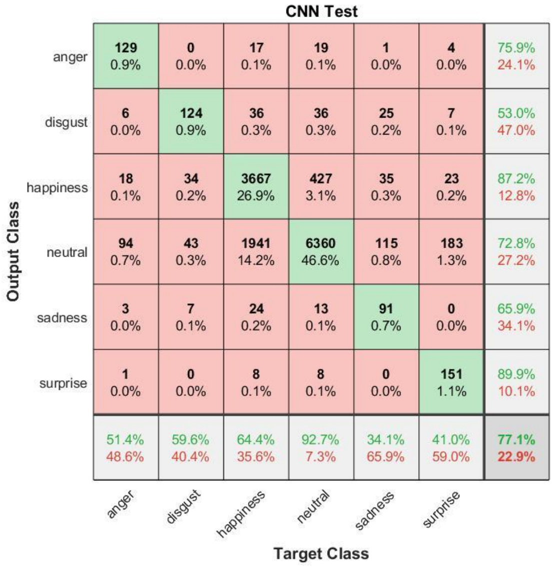
Output Class

Target Class

Figure 13: Validation Confusion matrix of CNN.

Test 2(Medium Occlusion):

The CNN confound matrix appears in Figure 14. Overall CNN test2 accuracy is 77.1%.



CNN Test

	anger	disgust	happiness	neutral	sadness	surprise	
anger	129 0.9%	0 0.0%	17 0.1%	19 0.1%	1 0.0%	4 0.0%	75.9% 24.1%
disgust	6 0.0%	124 0.9%	36 0.3%	36 0.3%	25 0.2%	7 0.1%	53.0% 47.0%
happiness	18 0.1%	34 0.2%	3667 26.9%	427 3.1%	35 0.3%	23 0.2%	87.2% 12.8%
neutral	94 0.7%	43 0.3%	1941 14.2%	6360 46.6%	115 0.8%	183 1.3%	72.8% 27.2%
sadness	3 0.0%	7 0.1%	24 0.2%	13 0.1%	91 0.7%	0 0.0%	65.9% 34.1%
surprise	1 0.0%	0 0.0%	8 0.1%	8 0.1%	0 0.0%	151 1.1%	89.9% 10.1%
	51.4% 48.6%	59.6% 40.4%	64.4% 35.6%	92.7% 7.3%	34.1% 65.9%	41.0% 59.0%	77.1% 22.9%

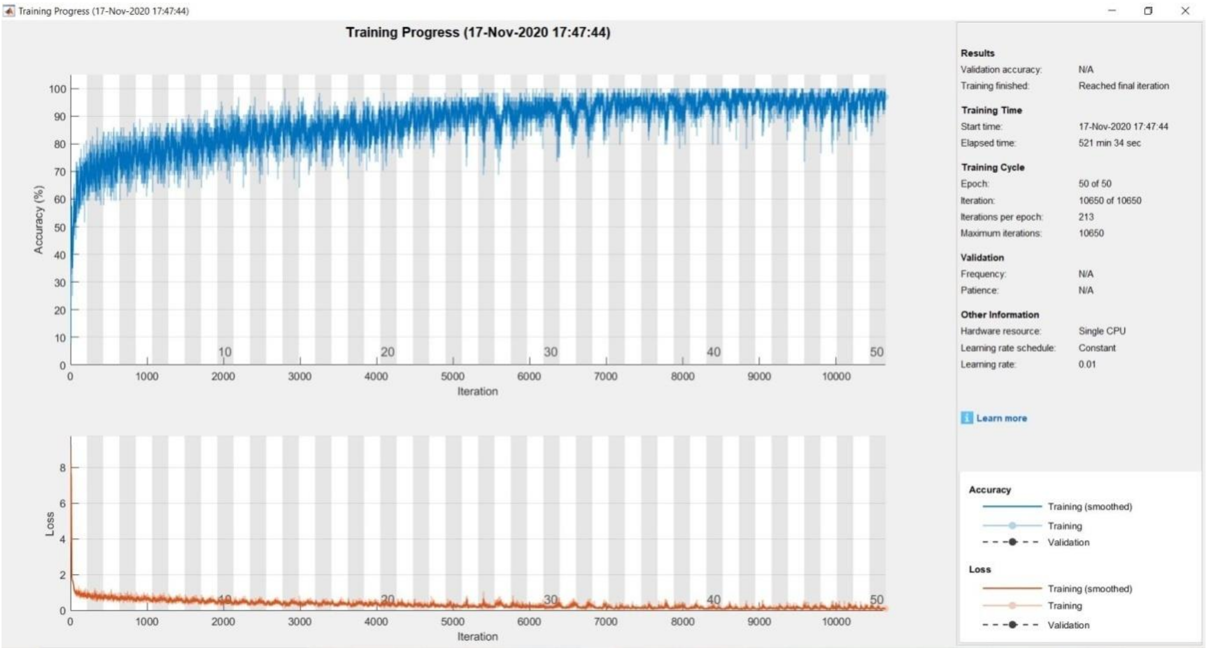
Output Class

Target Class

Figure 14: test confusion matrix for CNN.

Train 3 (Maximum Occlusion):

The figure 15 shows the training process for Maximum occlusion face images using convolution neural network.



Results
Validation accuracy: N/A
Training finished: Reached final iteration

Training Time
Start time: 17-Nov-2020 17:47:44
Elapsed time: 521 min 34 sec

Training Cycle
Epoch: 50 of 50
Iteration: 10650 of 10650
Iterations per epoch: 213
Maximum iterations: 10650

Validation
Frequency: N/A
Patience: N/A

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

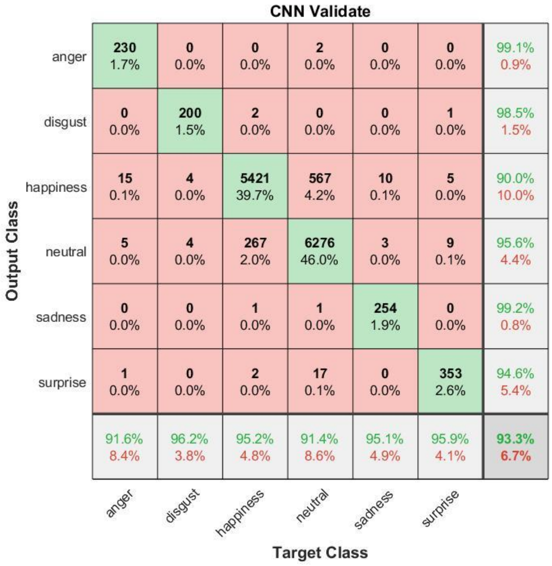
Accuracy
Training (smoothed)
Training
Validation

Loss
Training (smoothed)
Training
Validation

Figure 15: Training process for CNN.

Validate 3 (Maximum Occlusion):

The CNN uncertainty matrix is shown in figure 16. Overall validation3accuracy of CNN is 93.3%.


Output Class
Target Class

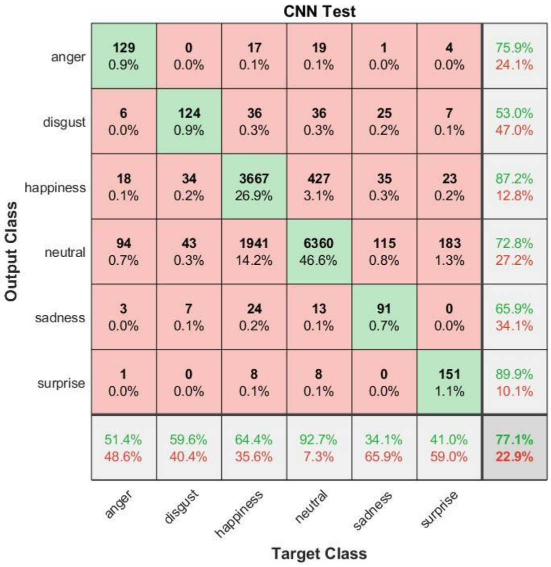
CNN Validate

	anger	disgust	happiness	neutral	sadness	surprise	
anger	230 1.7%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	99.1% 0.9%
disgust	0 0.0%	200 1.5%	2 0.0%	0 0.0%	0 0.0%	1 0.0%	98.5% 1.5%
happiness	15 0.1%	4 0.0%	5421 39.7%	567 4.2%	10 0.1%	5 0.0%	90.0% 10.0%
neutral	5 0.0%	4 0.0%	267 2.0%	6276 46.0%	3 0.0%	9 0.1%	95.6% 4.4%
sadness	0 0.0%	0 0.0%	1 0.0%	1 0.0%	254 1.9%	0 0.0%	99.2% 0.8%
surprise	1 0.0%	0 0.0%	2 0.0%	17 0.1%	0 0.0%	353 2.6%	94.6% 5.4%
	91.6% 8.4%	96.2% 3.8%	95.2% 4.8%	91.4% 8.6%	95.1% 4.9%	95.9% 4.1%	93.3% 6.7%

Figure 16: Validation Confusion matrix of CNN.

Test 3 (Maximum Occlusion):

The CNN confound matrix appears in Figure 17. Overall CNN test3 accuracy is 77.1%.



	anger	disgust	happiness	neutral	sadness	surprise	
anger	129 0.9%	0 0.0%	17 0.1%	19 0.1%	1 0.0%	4 0.0%	75.9% 24.1%
disgust	6 0.0%	124 0.9%	36 0.3%	36 0.3%	25 0.2%	7 0.1%	53.0% 47.0%
happiness	18 0.1%	34 0.2%	3667 26.9%	427 3.1%	35 0.3%	23 0.2%	87.2% 12.8%
neutral	94 0.7%	43 0.3%	1941 14.2%	6360 46.6%	115 0.8%	183 1.3%	72.8% 27.2%
sadness	3 0.0%	7 0.1%	24 0.2%	13 0.1%	91 0.7%	0 0.0%	65.9% 34.1%
surprise	1 0.0%	0 0.0%	8 0.1%	8 0.1%	0 0.0%	151 1.1%	89.9% 10.1%
	51.4% 48.6%	59.6% 40.4%	64.4% 35.6%	92.7% 7.3%	34.1% 65.9%	41.0% 59.0%	77.1% 22.9%

Figure 17: test confusion matrix for CNN.

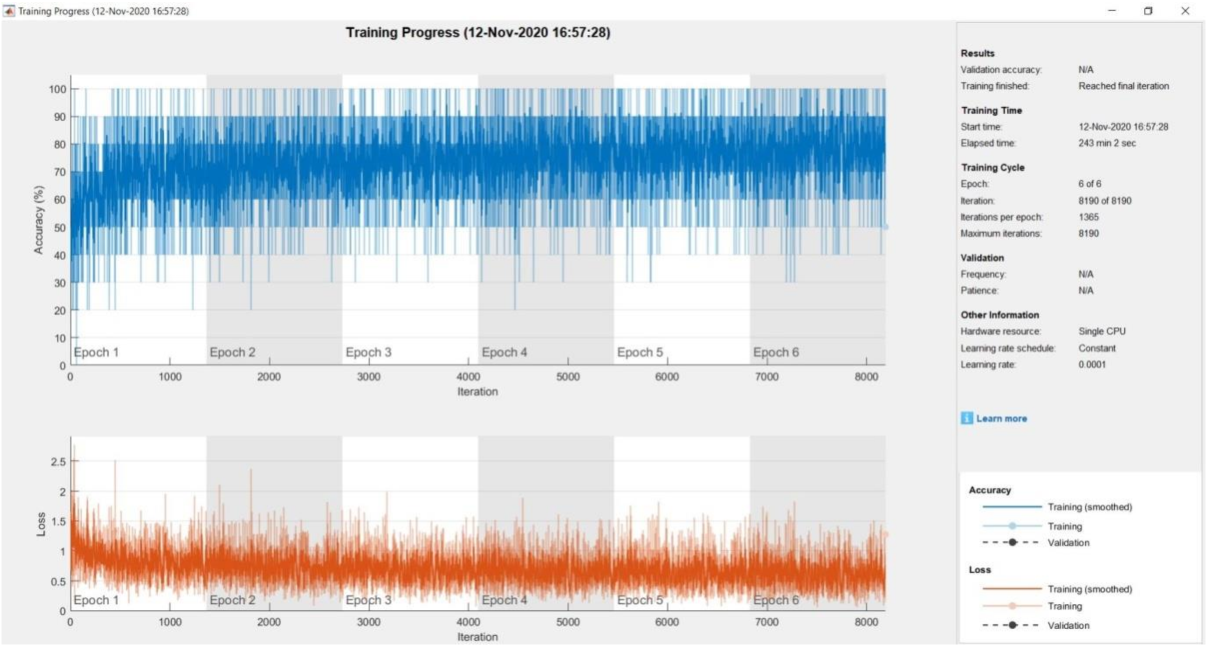
GoogleNet:

Training (Minimal Occlusion):

The train network needs 224-by-224-by-3 input images, but the images are of various sizes in the data stores. To automatically resize the training images, using an expanded image datastore.

To resize images, an expanded image datastore is used without any further data extension and specifying additional operations of pre-processing.

Specify translation learning options, retain the features from the early layer (transferred weights) of the pre-trained network. Set the initial learning rate to a small value to minimize learning in transmitted layers. In the previous stage, you raised learn levels to accelerate learning in new final layers for the completely linked layer. This variation of learning speeds ensures that only new layers learn quickly and the other layers are less lengthy. You will not have to prepare for as many epochs as you know how to move. An era is a complete course on the whole collection of data. Specify the size and validation data of the mini array. The program validates all iterations during training for the network ValidationFrequency.
Join the number of training epochs. You will not have to prepare for as many epochs as you know how to move. An era is a complete course on the whole collection of data. Specify the size and validation data of the mini array. Compute the precision of validity once a time.
The figure 4 explain guidance method1 for GoogleNet.



Results

Validation accuracy:	N/A
Training finished:	Reached final iteration

Training Time

Start time:	12-Nov-2020 16:57:28
Elapsed time:	243 min 2 sec

Training Cycle

Epoch:	6 of 6
Iteration:	8190 of 8190
Iterations per epoch:	1365
Maximum iterations:	8190

Validation

Frequency:	N/A
Patience:	N/A

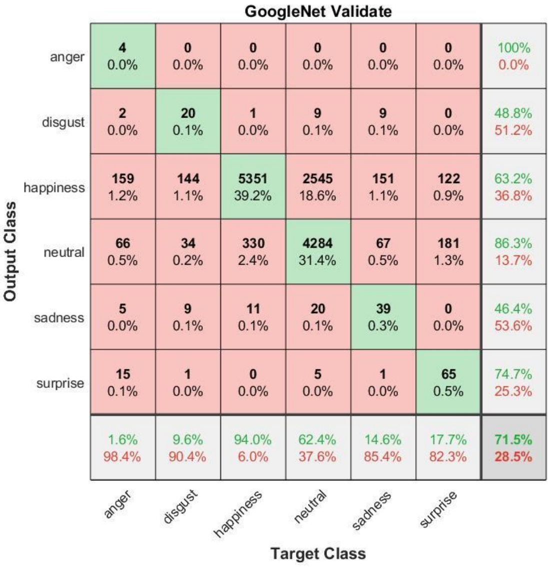
Other Information

Hardware resource:	Single CPU
Learning rate schedule:	Constant
Learning rate:	0.0001

Figure 18:Trainingprocedure1 for GoogleNet

Validation1 (Minimal Occlusion):

The figure 19 shows the confusion matrix for GoogleNet. The overall validation1 accuracy is 71.5%.

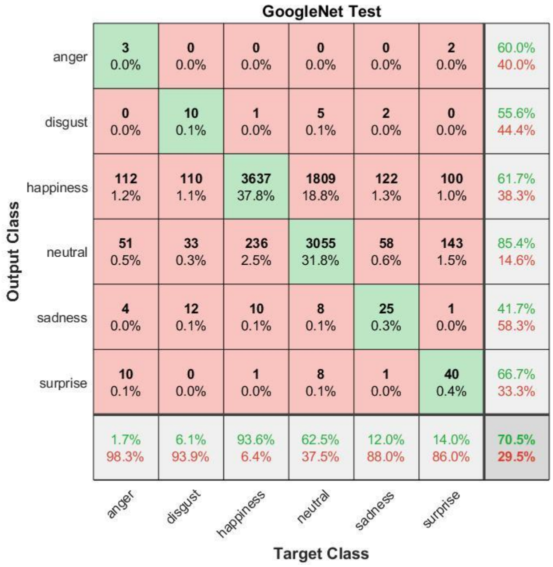


	anger	disgust	happiness	neutral	sadness	surprise	
anger	**4** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
disgust	**2** 0.0%	**20** 0.1%	**1** 0.0%	**9** 0.1%	**9** 0.1%	**0** 0.0%	48.8% 51.2%
happiness	**159** 1.2%	**144** 1.1%	**5351** 39.2%	**2545** 18.6%	**151** 1.1%	**122** 0.9%	63.2% 36.8%
neutral	**66** 0.5%	**34** 0.2%	**330** 2.4%	**4284** 31.4%	**67** 0.5%	**181** 1.3%	86.3% 13.7%
sadness	**5** 0.0%	**9** 0.1%	**11** 0.1%	**20** 0.1%	**39** 0.3%	**0** 0.0%	46.4% 53.6%
surprise	**15** 0.1%	**1** 0.0%	**0** 0.0%	**5** 0.0%	**1** 0.0%	**65** 0.5%	74.7% 25.3%
	1.6% 98.4%	9.6% 90.4%	94.0% 6.0%	62.4% 37.6%	14.6% 85.4%	17.7% 82.3%	**71.5%** **28.5%**

Figure 19:GoogleNet validation1 confusion matrix.

Test1 (Minimal Occlusion):

The test confusion matrix for GoogleNet is shown in figure 20. The overall test1 Accuracy is 70.5%.



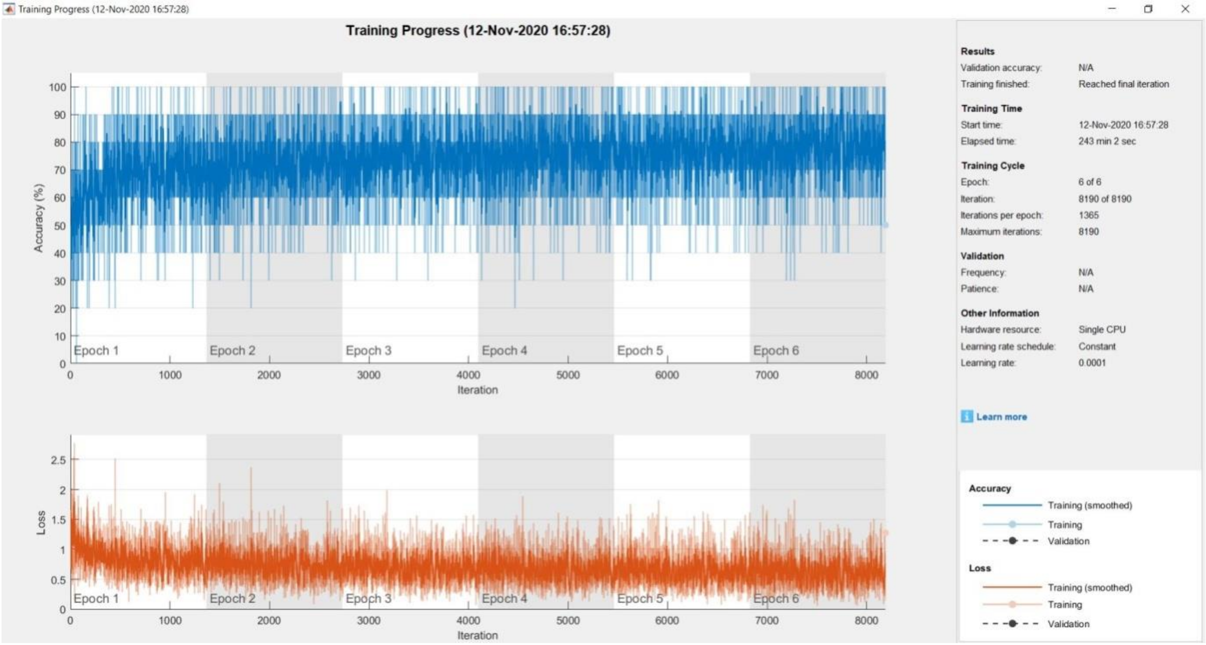
GoogleNet Test

Output Class \ Target Class	anger	disgust	happiness	neutral	sadness	surprise	
anger	3 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.0%	60.0% 40.0%
disgust	0 0.0%	10 0.1%	1 0.0%	5 0.1%	2 0.0%	0 0.0%	55.6% 44.4%
happiness	112 1.2%	110 1.1%	3637 37.8%	1809 18.8%	122 1.3%	100 1.0%	61.7% 38.3%
neutral	51 0.5%	33 0.3%	236 2.5%	3055 31.8%	58 0.6%	143 1.5%	85.4% 14.6%
sadness	4 0.0%	12 0.1%	10 0.1%	8 0.1%	25 0.3%	1 0.0%	41.7% 58.3%
surprise	10 0.1%	0 0.0%	1 0.0%	8 0.1%	1 0.0%	40 0.4%	66.7% 33.3%
	1.7% 98.3%	6.1% 93.9%	93.6% 6.4%	62.5% 37.5%	12.0% 88.0%	14.0% 86.0%	70.5% 29.5%

Figure 20: Test1 confusion matrix for Google Net.

GoogleNet Train 2 (Medium Occlusion):

The figure 21 explainstraining2 process for GoogleNet.



Training Progress (12-Nov-2020 16:57:28)

Results
Validation accuracy: N/A
Training finished: Reached final iteration

Training Time
Start time: 12-Nov-2020 16:57:28
Elapsed time: 243 min 2 sec

Training Cycle
Epoch: 6 of 6
Iteration: 8190 of 8190
Iterations per epoch: 1365
Maximum iterations: 8190

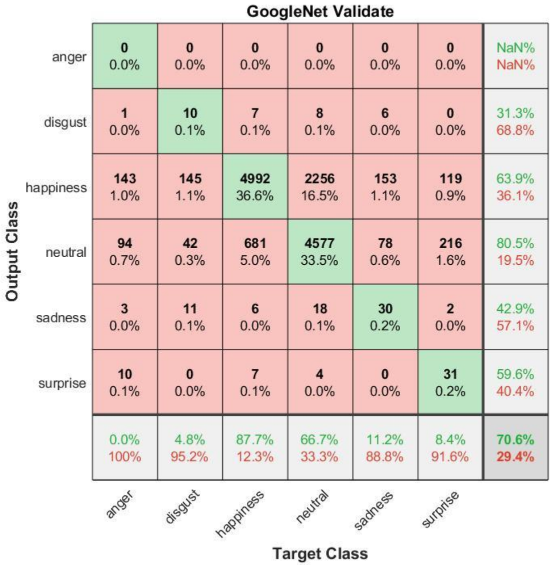
Validation
Frequency: N/A
Patience: N/A

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.0001

Figure 21: Training procedure2 for GoogleNet

Validation2 (Medium Occlusion):

The figure 22 shows the confusion matrix for GoogleNet. The overall validation2 accuracy is 70.6%.



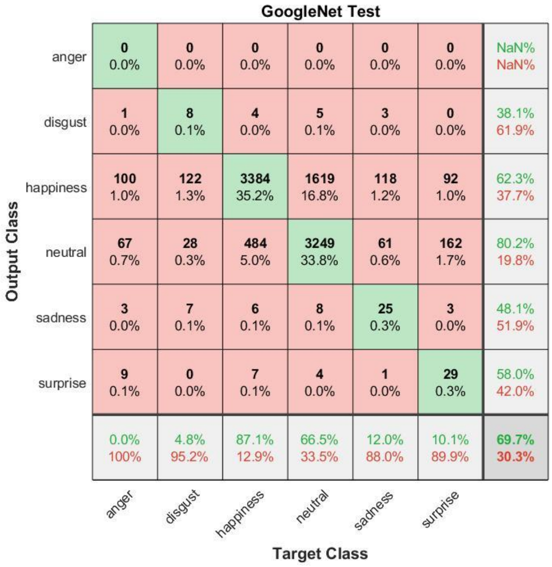
GoogleNet Validate

Output Class \ Target Class	anger	disgust	happiness	neutral	sadness	surprise	
anger	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
disgust	1 0.0%	10 0.1%	7 0.1%	8 0.1%	6 0.0%	0 0.0%	31.3% 68.8%
happiness	143 1.0%	145 1.1%	4992 36.6%	2256 16.5%	153 1.1%	119 0.9%	63.9% 36.1%
neutral	94 0.7%	42 0.3%	681 5.0%	4577 33.5%	78 0.6%	216 1.6%	80.5% 19.5%
sadness	3 0.0%	11 0.1%	6 0.0%	18 0.1%	30 0.2%	2 0.0%	42.9% 57.1%
surprise	10 0.1%	0 0.0%	7 0.1%	4 0.0%	0 0.0%	31 0.2%	59.6% 40.4%
	0.0% 100%	4.8% 95.2%	87.7% 12.3%	66.7% 33.3%	11.2% 88.8%	8.4% 91.6%	70.6% 29.4%

Figure 22: GoogleNet validation2 confusion matrix.

Test2(Medium Occlusion):

The test confusion matrix for GoogleNet is shown in figure 23. The overall test2 Accuracy is 69.7%.



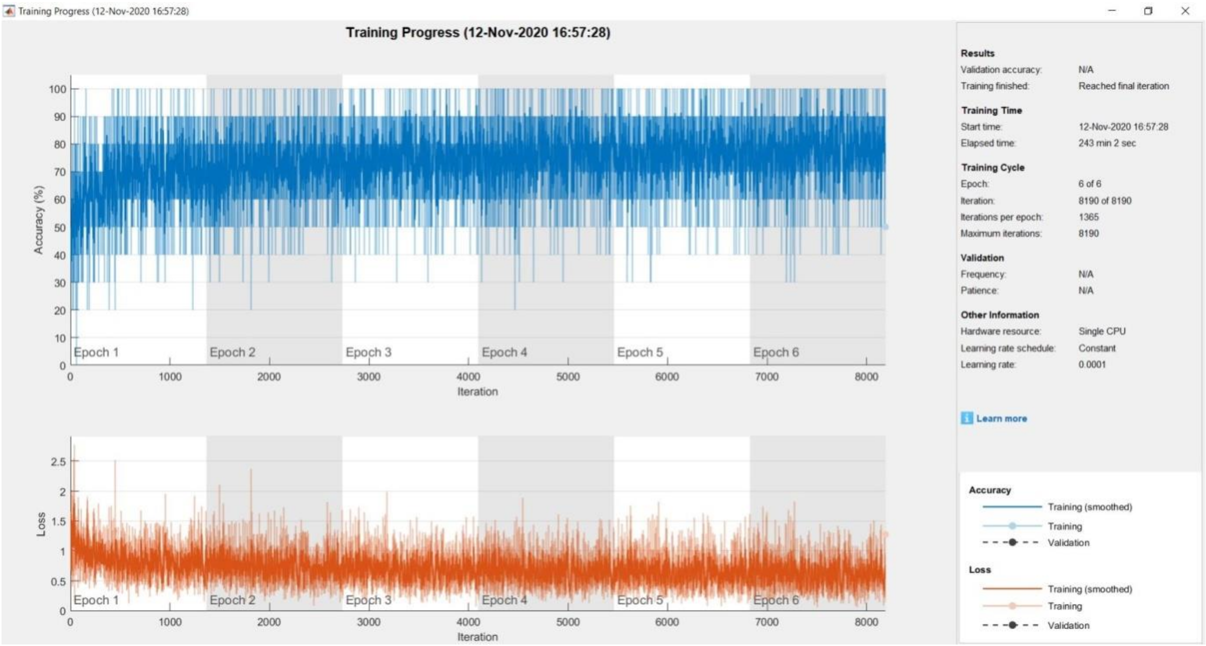
GoogleNet Test

Output Class \ Target Class	anger	disgust	happiness	neutral	sadness	surprise	
anger	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
disgust	1 0.0%	8 0.1%	4 0.0%	5 0.1%	3 0.0%	0 0.0%	38.1% 61.9%
happiness	100 1.0%	122 1.3%	3384 35.2%	1619 16.8%	118 1.2%	92 1.0%	62.3% 37.7%
neutral	67 0.7%	28 0.3%	484 5.0%	3249 33.8%	61 0.6%	162 1.7%	80.2% 19.8%
sadness	3 0.0%	7 0.1%	6 0.1%	8 0.1%	25 0.3%	3 0.0%	48.1% 51.9%
surprise	9 0.1%	0 0.0%	7 0.1%	4 0.0%	1 0.0%	29 0.3%	58.0% 42.0%
	0.0% 100%	4.8% 95.2%	87.1% 12.9%	66.5% 33.5%	12.0% 88.0%	10.1% 89.9%	69.7% 30.3%

Figure 23: Test2 confusion matrix for Google Net.

GoogleNet Train 3 (Maximum Occlusion):

The figure 24 explains training3 process for GoogleNet.



Training Progress (12-Nov-2020 16:57:28)

Results

Validation accuracy: N/A
Training finished: Reached final iteration

Training Time

Start time: 12-Nov-2020 16:57:28
Elapsed time: 243 min 2 sec

Training Cycle

Epoch: 6 of 6
Iteration: 8190 of 8190
Iterations per epoch: 1365
Maximum iterations: 8190

Validation

Frequency: N/A
Patience: N/A

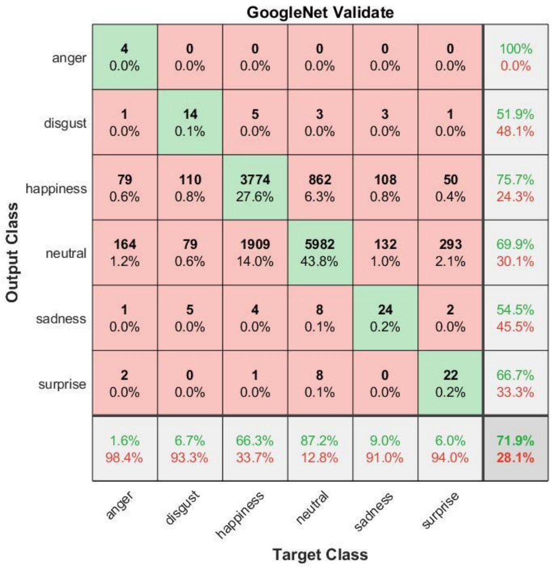
Other Information

Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.0001

Figure 24: Training procedure3 for GoogleNet

Validation3 (Maximum Occlusion):

The figure 25 shows the confusion matrix for GoogleNet. The overall validation3 accuracy is 71.9%.



Output Class
Target Class

GoogleNet Validate

	anger	disgust	happiness	neutral	sadness	surprise	
anger	4 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
disgust	1 0.0%	14 0.1%	5 0.0%	3 0.0%	3 0.0%	1 0.0%	51.9% 48.1%
happiness	79 0.6%	110 0.8%	3774 27.6%	862 6.3%	108 0.8%	50 0.4%	75.7% 24.3%
neutral	164 1.2%	79 0.6%	1909 14.0%	5982 43.8%	132 1.0%	293 2.1%	69.9% 30.1%
sadness	1 0.0%	5 0.0%	4 0.0%	8 0.1%	24 0.2%	2 0.0%	54.5% 45.5%
surprise	2 0.0%	0 0.0%	1 0.0%	8 0.1%	0 0.0%	22 0.2%	66.7% 33.3%
	1.6% 98.4%	6.7% 93.3%	66.3% 33.7%	87.2% 12.8%	9.0% 91.0%	6.0% 94.0%	71.9% 28.1%

Figure 25: GoogleNet validation3 confusion matrix.

Test3 (Maximum Occlusion):

The test confusion matrix for GoogleNet is shown in figure 26. The overall test3 Accuracy is 71.9%.

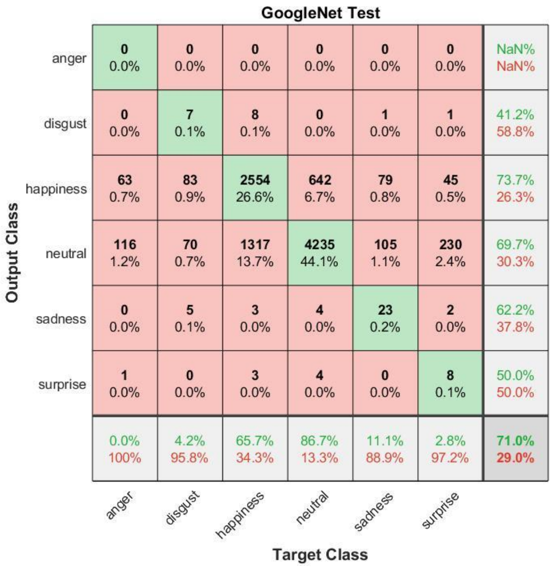


Figure 26: Test3 confusion matrix for Google Net.

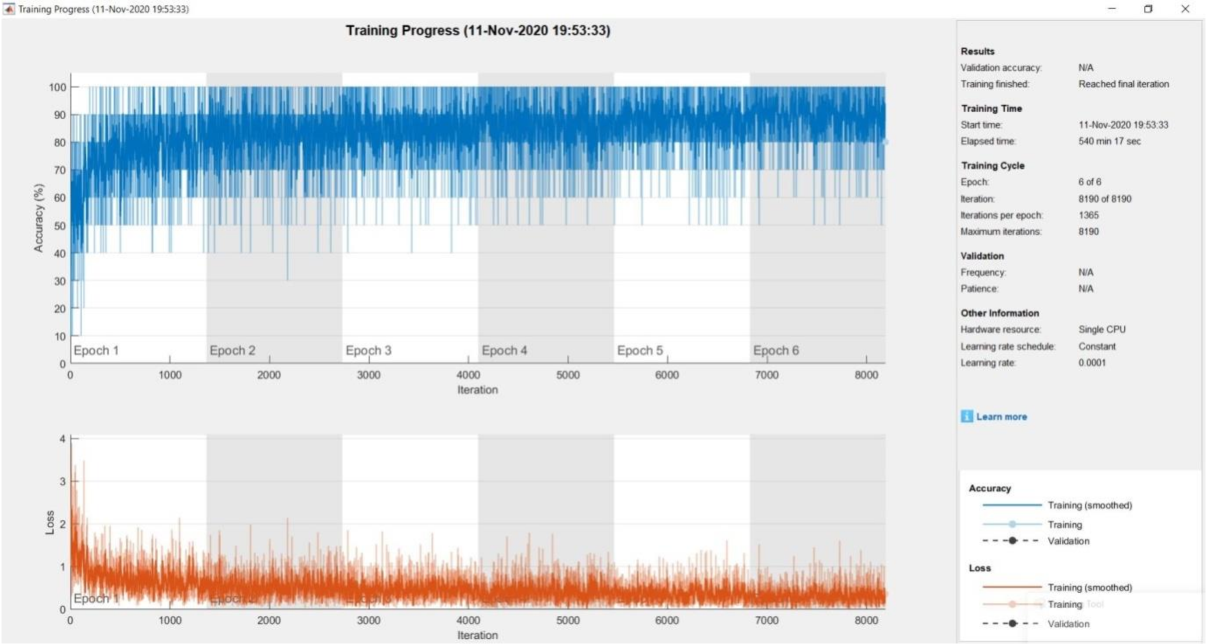
Alexnet:

Train1(Minimum Occlusion):

The train network needs 227-by-227-by-3 input pictures, but the data controllers have different sizes of image files. To automatically resize the training images, using an expanded image datastore. Indicate more amplification operations in the training pictures: pull the training pictures around the vertical axis randomly and those are converted vertically and horizontally to 30 pixels. The data increase prevents the network overload and stores the exact information of the training images.

With the replacement of three layers with a grading output layer, a Softmax layer, and a completely linked layer, the layers are defined and transferred to new classification. Based on the new data, the new fully connected layer options are specified. The completely connected layer is set to the size of new data groups. The learning on new layers faster through the fully connected layer's BiasLearnRateFactor and WeightLearnRateFactor than on the transferred layers.

The training process1 for AlexNet is shown in the figure 27.



Results
Validation accuracy: N/A
Training finished: Reached final iteration

Training Time
Start time: 11-Nov-2020 19:53:33
Elapsed time: 540 min 17 sec

Training Cycle
Epoch: 6 of 6
Iteration: 8190 of 8190
Iterations per epoch: 1365
Maximum iterations: 8190

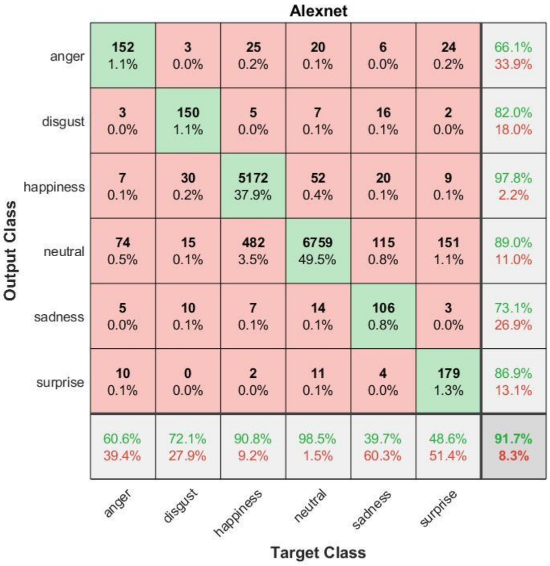
Validation
Frequency: N/A
Patience: N/A

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.0001

Figure 27: Training process1 for AlexNet

Validation1 (Minimum Occlusion):

The figure 28 shows the confusion matrix for AlexNet. The overall validation1 accuracy is 91.7%.

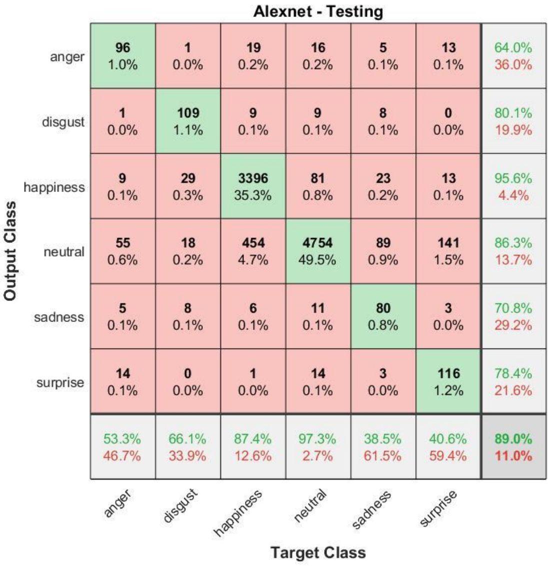

Alexnet
Output Class
Target Class

	anger	disgust	happiness	neutral	sadness	surprise	
anger	152 1.1%	3 0.0%	25 0.2%	20 0.1%	6 0.0%	24 0.2%	66.1% 33.9%
disgust	3 0.0%	150 1.1%	5 0.0%	7 0.1%	16 0.1%	2 0.0%	82.0% 18.0%
happiness	7 0.1%	30 0.2%	5172 37.9%	52 0.4%	20 0.1%	9 0.1%	97.8% 2.2%
neutral	74 0.5%	15 0.1%	482 3.5%	6759 49.5%	115 0.8%	151 1.1%	89.0% 11.0%
sadness	5 0.0%	10 0.1%	7 0.1%	14 0.1%	106 0.8%	3 0.0%	73.1% 26.9%
surprise	10 0.1%	0 0.0%	2 0.0%	11 0.1%	4 0.0%	179 1.3%	86.9% 13.1%
	60.6% 39.4%	72.1% 27.9%	90.8% 9.2%	98.5% 1.5%	39.7% 60.3%	48.6% 51.4%	91.7% 8.3%

Figure 28:AlexNet Validation1 Confusion matrix.

Test1 (Minimum Occlusion):

The test confusion matrix for AlexNet is shown in figure 29.The overall test1 Accuracy is 89.0%.

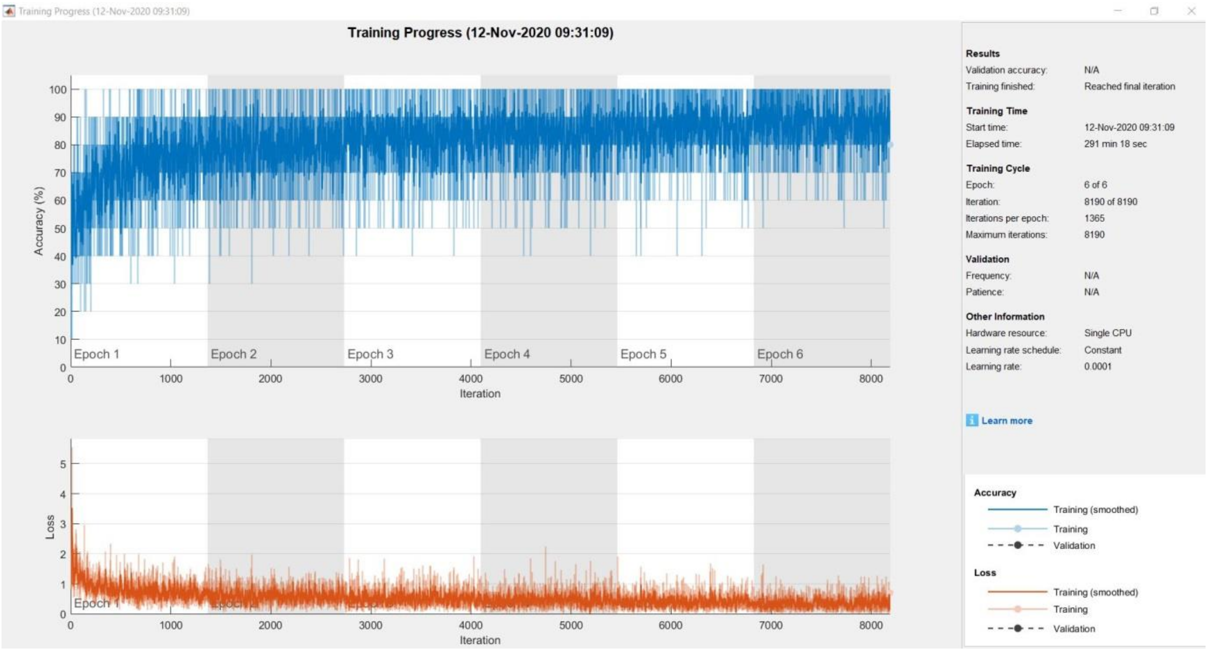

Alexnet - Testing
Output Class
Target Class

	anger	disgust	happiness	neutral	sadness	surprise	
anger	96 1.0%	1 0.0%	19 0.2%	16 0.2%	5 0.1%	13 0.1%	64.0% 36.0%
disgust	1 0.0%	109 1.1%	9 0.1%	9 0.1%	8 0.1%	0 0.0%	80.1% 19.9%
happiness	9 0.1%	29 0.3%	3396 35.3%	81 0.8%	23 0.2%	13 0.1%	95.6% 4.4%
neutral	55 0.6%	18 0.2%	454 4.7%	4754 49.5%	89 0.9%	141 1.5%	86.3% 13.7%
sadness	5 0.1%	8 0.1%	6 0.1%	11 0.1%	80 0.8%	3 0.0%	70.8% 29.2%
surprise	14 0.1%	0 0.0%	1 0.0%	14 0.1%	3 0.0%	116 1.2%	78.4% 21.6%
	53.3% 46.7%	66.1% 33.9%	87.4% 12.6%	97.3% 2.7%	38.5% 61.5%	40.6% 59.4%	89.0% 11.0%

Figure 29: Test1 confusion matrix for AlexNet.

Train 2(Medium Occlusion):

The figure 30 shows the training process2 for AlexNet.



Results

Validation accuracy: N/A
Training finished: Reached final iteration

Training Time

Start time: 12-Nov-2020 09:31:09
Elapsed time: 291 min 18 sec

Training Cycle

Epoch: 6 of 6
Iteration: 8190 of 8190
Iterations per epoch: 1365
Maximum iterations: 8190

Validation

Frequency: N/A
Patience: N/A

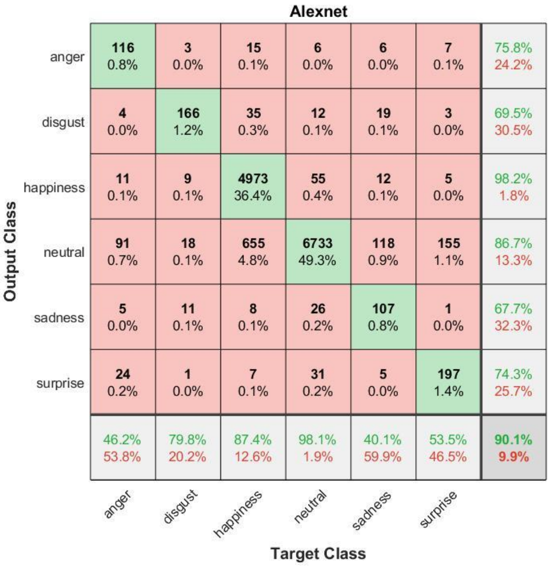
Other Information

Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.0001

Figure 30: Training process2 for AlexNet

Validation2 (Medium Occlusion):

The figure 31 shows the confusion matrix for AlexNet. The overall validation2 accuracy is 90.1%.



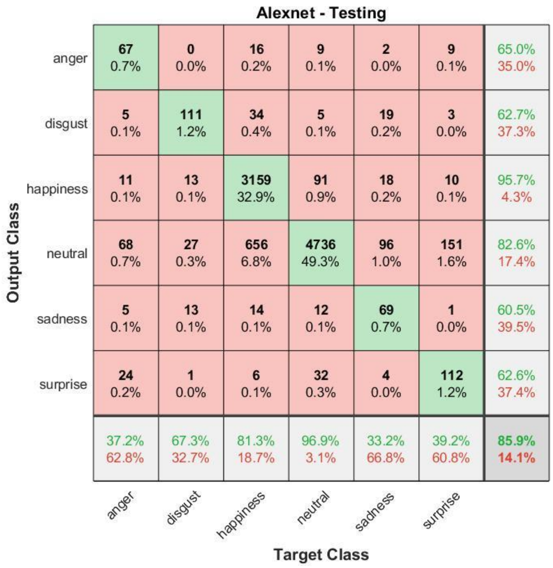
Alexnet

Output Class \ Target Class	anger	disgust	happiness	neutral	sadness	surprise	
anger	116 0.8%	3 0.0%	15 0.1%	6 0.0%	6 0.0%	7 0.1%	75.8% 24.2%
disgust	4 0.0%	166 1.2%	35 0.3%	12 0.1%	19 0.1%	3 0.0%	69.5% 30.5%
happiness	11 0.1%	9 0.1%	4973 36.4%	55 0.4%	12 0.1%	5 0.0%	98.2% 1.8%
neutral	91 0.7%	18 0.1%	655 4.8%	6733 49.3%	118 0.9%	155 1.1%	86.7% 13.3%
sadness	5 0.0%	11 0.1%	8 0.1%	26 0.2%	107 0.8%	1 0.0%	67.7% 32.3%
surprise	24 0.2%	1 0.0%	7 0.1%	31 0.2%	5 0.0%	197 1.4%	74.3% 25.7%
	46.2% 53.8%	79.8% 20.2%	87.4% 12.6%	98.1% 1.9%	40.1% 59.9%	53.5% 46.5%	90.1% 9.9%

Figure 31: AlexNet Validation2 Confusion matrix.

Test2 (Medium Occlusion):

The test confusion matrix for AlexNet is shown in figure 32. The overall test2 Accuracy is 85.9%.



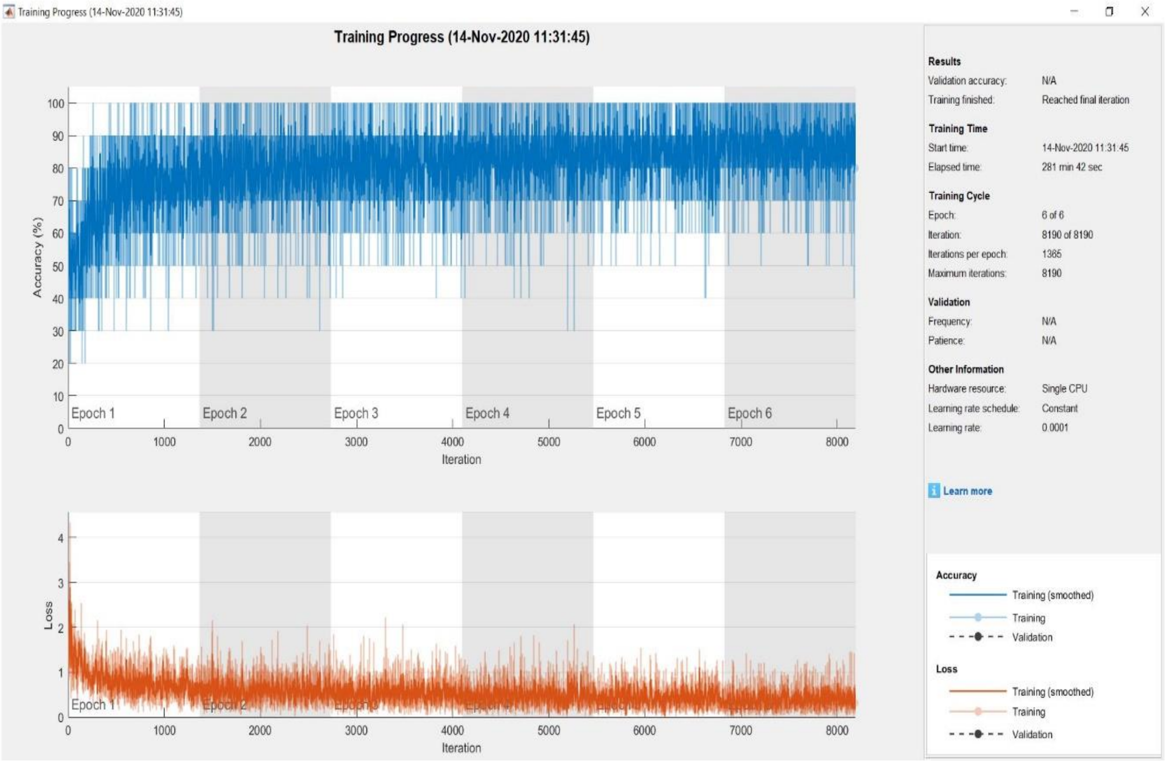
Alexnet - Testing

Output Class \ Target Class	anger	disgust	happiness	neutral	sadness	surprise	
anger	67 0.7%	0 0.0%	16 0.2%	9 0.1%	2 0.0%	9 0.1%	65.0% 35.0%
disgust	5 0.1%	111 1.2%	34 0.4%	5 0.1%	19 0.2%	3 0.0%	62.7% 37.3%
happiness	11 0.1%	13 0.1%	3159 32.9%	91 0.9%	18 0.2%	10 0.1%	95.7% 4.3%
neutral	68 0.7%	27 0.3%	656 6.8%	4736 49.3%	96 1.0%	151 1.6%	82.6% 17.4%
sadness	5 0.1%	13 0.1%	14 0.1%	12 0.1%	69 0.7%	1 0.0%	60.5% 39.5%
surprise	24 0.2%	1 0.0%	6 0.1%	32 0.3%	4 0.0%	112 1.2%	62.6% 37.4%
	37.2% 62.8%	67.3% 32.7%	81.3% 18.7%	96.9% 3.1%	33.2% 66.8%	39.2% 60.8%	85.9% 14.1%

Figure 32: Test2 confusion matrix for AlexNet.

Train 3(Maximum Occlusion):

The figure 33 shows the training process3 for AlexNet.



Training Progress (14-Nov-2020 11:31:45)

Results
Validation accuracy: N/A
Training finished: Reached final iteration

Training Time
Start time: 14-Nov-2020 11:31:45
Elapsed time: 281 min 42 sec

Training Cycle
Epoch: 6 of 6
Iteration: 8190 of 8190
Iterations per epoch: 1365
Maximum iterations: 8190

Validation
Frequency: N/A
Patience: N/A

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.0001

Figure 33: Training process3 for AlexNet

Validation3 (Maximum Occlusion):

The figure 34 shows the confusion matrix for AlexNet. The overall validation3 accuracy is 90.5%.

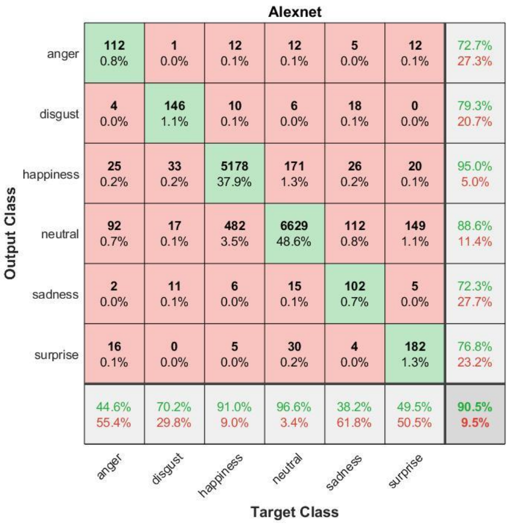


Figure 34: AlexNet Validation3 Confusion matrix.

Test3 (Maximum Occlusion):

The test confusion matrix for AlexNet is shown in figure 35. The overall test3 Accuracy is 93.2%.

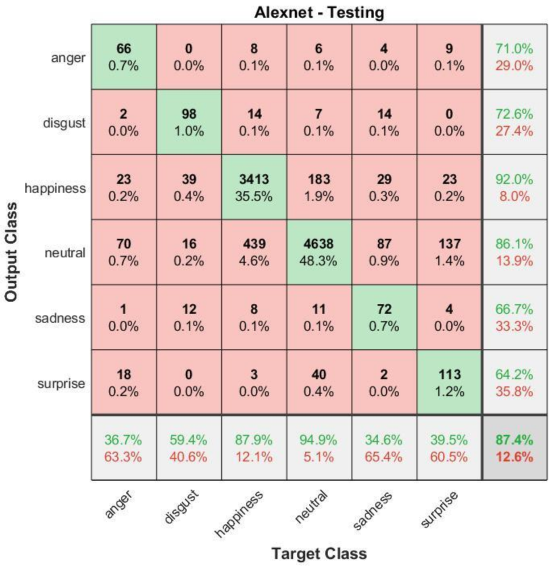


Figure 35: Test3 confusion matrix for AlexNet.

Comparison Results:

In the below table 1 shows the comparison of different classification algorithms.

Table 1 Accuracy comparison for different algorithms

Algorithm	Validation Accuracy	Testing Accuracy
CNN (Minimum Occlusion)	87.4%	87.4%
CNN (Medium Occlusion)	77.1 %	77.1%
CNN (Maximum Occlusion)	93.3%	93.3%
GoogleNet(Minimum Occlusion)	71.5%	70.5%
GoogleNet(Medium Occlusion)	70.6%	69.7%
GoogleNet(MaximumOcclusion)	71.9%	71.0%
AlexNet(Minimum Occlusion)	91.7%	89.0%
AlexNet(Medium Occlusion)	90.1%	85.9%
AlexNet(Maximum Occlusion)	90.5%	87.4%

Conclusion:

A short review of FER approaches was presented in this paper. Two main streams can divide by such approaches as described: traditional approaches of FER contain three steps such as face and facial components' identification, features extraction, and expression classification. In conventional FER and deep-learning based FER approaches, Adaboost, random forest is used as the classification algorithm. By allowing end-to-end learning from the input images directly in the pipeline, the dependence on face-physics-based models and other pre-processing methods greatly reduces. To understand the model learned through different datasets of FER, the input images visualize by a CNN as a particular type of deep learning. In both various FER-related tasks and datasets, the ability of emotion detection-trained networks is demonstrated. An analysis of a CNN architecture has provided in a few recent studies for facial expressions. The temporal averaging can utilize for aggregation to improve the performance of previously utilized CNN techniques. A number of limitations has included in the deep-learning-based FER approaches such as large amounts of memory, massive computing power, and the need for large-scale datasets. Both the phases of testing and training are time consuming. More accuracy and performance have achieved with the AlexNet than the Google and CNN networks.

In addition, to include standard metrics for comparison, assessment metrics of FER-based approaches were added. In the field of identification, evaluation metrics have been commonly evaluated. Primarily, recall and precision are utilized. A modern assessment tool for identifying consecutive facial expressions, however is beneficial that separate studies are being performed on its future use.

References:

[1] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," in Proc. CVPRW, Jun. 2010, pp. 94–101.

[2] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," in Proc. ICME, Jul. 2005, p. 5.

[3] G. Zhao, X. Huang, M. Taini, and S. Z. Li, "Facial expression recognition from near-infrared videos," Image Vis. Comput., vol. 29, no. 9, pp. 607–619, 2011.

[4] S. Li, W. Deng, and J. Du, "Reliable crowdsourcing and deep localitypreserving learning for expression recognition in the wild," in Proc. CVPR, Jul. 2017, pp. 2584–2593.

[5] A. Mollahosseini, B. Hasani, and M. H. Mahoor. (2017). "AffectNet: A database for facial expression, valence, and arousal computing in the wild." [Online]. Available: https://arxiv.org/abs/1708.03985.

[6] Y. Li, J. Zeng, S. Shan, and X. Chen, "Patch-gated CNN for occlusionaware facial expression recognition," in Proc. Int. Conf. Pattern Recognit. (ICPR), Aug. 2018, pp. 2209–2214.

[7] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 2, pp. 210–227, Feb. 2009.

[8] E. Osherov and M. Lindenbaum, "Increasing CNN robustness to occlusions by reducing filter support," in Proc. CVPR, Oct. 2017, pp. 550–561.

[9] M. Ranzato, J. Susskind, V. Mnih, and G. Hinton, "On deep generative models with applications to recognition," in Proc. CVPR, Jun. 2011, pp. 2857–2864.

[10] I. Kotsia, I. Buciu, and I. Pitas, "An analysis of facial expression recognition under partial facial image occlusion," Image Vis. Comput., vol. 26, no. 7, pp. 1052–1067, 2008.

[11] A. M. Martinez, "Recognizing imprecisely localized, partially occluded, and expression

variant faces from a single sample per class," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 6, pp. 748–763, Jun. 2002.

[12] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. N. Metaxas, "Learning active facial patches for expression analysis," in Proc. CVPR, Jun. 2012, pp. 2562–2569.

[13] X. Huang, G. Zhao, W. Zheng, and M. Pietikäinen, "Towards a dynamic expression recognition system under facial occlusion," Pattern Recognit.Lett., vol. 33, no. 16, pp. 2181–2191, 2012.

[14] A. Dapogny, K. Bailly, and S. Dubuisson, "Confidence-weighted local expression predictions for occlusion handling in expression recognition and action unit detection," Int. J. Comput.Vis., vol. 126, nos. 2–4, pp. 255–271, 2017.

[15] W. Li, F. Abtahi, and Z. Zhu, "Action unit detection with region adaptation, multi-labeling learning and optimal temporal fusing," in Proc. CVPR, Jul. 2017, pp. 6766–6775.

[16] L. Zhang, D. Tjondronegoro, and V. Chandran, "Random Gabor based templates for facial expression recognition in images with facial occlusion," Neurocomputing, vol. 145, pp. 451–464, Dec. 2014.

[17] R. Min, A. Hadid, and J.-L. Dugelay, "Improving the recognition of faces occluded by facial accessories," in Proc. Autom. Face Gesture Recognit. Workshops, Mar. 2011, pp. 442–447.

[18] J.-C. Lin, C.-H.Wu, and W.-L. Wei, "Facial action unit prediction under partial occlusion based on error weighted cross-correlation model," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., May 2013, pp. 3482–3486.

[19] H. Towner and M. Slater, "Reconstruction and recognition of occluded facial expressions using PCA," in Proc. Int. Conf. Affect.Comput.Intell. Interact. Springer, 2007, pp. 36–47.

[20] Y. Deng, D. Li, X. Xie, K.-M.Lam, and Q. Dai, "Partially occluded face completion and recognition," in Proc. ICIP, Nov. 2009, pp. 4145–4148.

[21] SushmaPulidindi, K. Kamakshaiah and SagarYeruva, A Deep Neural Network on Object Recognition Framework for Submerged Fish Images, ISBN :978-981-15-0978-0, LNDECT, volume 37(2020), pp 443-450.