# Facial Emotion Recognition Using Full Face and Using Half Face (Eyes and Eyebrows)

Sara Sarrafan Chaharsoughi
Student, Masters in Applied Computing
Wilfrid Laurier University
Waterloo (ON), Canada
Sarr6790@mylaurier.ca

***Abstract*— The process of detecting human emotion is known as emotion detection. Human emotions are mental states of sentiments that arise spontaneously and are accompanied by physiological changes in facial muscles that result in facial expressions. Many applications of human-computer interaction involve nonverbal communication methods such as facial expressions, eye movement, and gestures; among these, facial emotion is commonly used because it expresses people's emotional states and feelings. Due to the lack of a clear pattern between the numerous emotions on the face, as well as the complexity and variety, emotion detection is challenging. The machine learning system will not achieve a high accuracy rate for emotion recognition since numerous crucial extracted elements used for modelling the face are hand-engineered and rely on prior information. In this study, convolutional neural networks (CNN) were used to recognize face emotion expressions. Facial expressions are important in nonverbal communication because they indicate a person's internal feelings, which are reflected on the face.**

***Keywords—Emotion recognition, convolutional neural network, machine learning, deep machine learning, image processing***

## I. Introduction

E Emotion is generated by specific events, and recognizing human emotion is a key topic in the research of human-computer interfaces (HCIs) that help people empathize [50][51]. Facial expressions, speech, physiological signs, and language are all examples of how emotions are expressed [52][53]. Facial emotion recognition and detection have very swiftly transitioned to a topic of continuous innovation and development, all due to the gradual removal of the limitations of computer vision with the help of Machine Learning [62]. Algorithms under Machine Learning and Deep Learning generally utilize a large level of computation power; hence the capabilities of algorithmic models match the magnitude of the real-world problems of Image Processing. However, using images to recognize and identify facial expressions and emotions is still a very challenging task as it is difficult to accurately extract emotional features in practice that are useful.

Every key facial element somewhat distinctively changes its values when emotions are expressed by humans. Keeping in contrast, the same features will generally have similar values across images with the same emotions. A general classification of human emotions could be as: happy, sad, anger, surprise, disgust, fear, and neutral. Specific facial expressions are linked to these emotions [54]. Facial expressions, along with other significant clues such as postures, gestures, spoken and vocal expressions, play a vital part in communicating feelings and attitudes [55]. All these emotions are dependent on several minimalistic facial muscle contortions which make our problem fulfilling to solve. Machine Learning and Neural Network algorithms have specifically been good solutions to extract and categorize facial features to further facilitate emotion recognition.

Automated emotion recognition systems, in particular, have applications in health care, such as detecting psychological discomfort [56], education, such as estimating student involvement [57], and gaming, such as improving the players' experience [58]. Emotion recognition has numerous real-world applications across various fields like autonomous vehicles, security, human-computer interaction, and healthcare. Graphical Processing Units have propelled the capabilities of Facial and Emotion Recognition being small pieces of hardware capable of completing millions of computations in a matter of seconds or minutes. Combining this hardware power with the statistically robust Machine Learning Algorithms help produce efficient solutions for our problem of Emotion Detection. Deep learning is based on the principle of learning a hierarchy of features in which higher-level features are made up of lower-level features. Edge detectors, for example, maybe the lowest-level characteristics, while groupings of edges could be detected at the second level, and filters, at the highest level, could approximate face features [59]. When trained on natural picture data, the lowest-level features in CNNs frequently seem Gabor-like. We focused more on the CNN algorithm as it is a very useful algorithm in this area for the recognition of these facial expressions. In this paper, we study facial emotion recognition by the CNN algorithm to determine one of the seven emotions from a facial image. The machine learning framework should, in theory, focus only on significant aspects of the face and be less sensitive to other areas of the face [63].

The rest of this paper is organized as follows. Section 2: Related Works, Section 3: Facial Emotion Recognition Using Machine Learning Algorithms, Section 4: Experimental Dataset, Section 5: Experiment Performed, Section 6: Experiment Result, and Section 7: Conclusion.

## II. Related Works

The elements of a gesture dynamic are proposed for an emotion recognition system and supervised learning methods are used to evaluate them [64]. The hidden units of a convolutional autoencoder are proposed as a framework for high-level parameters in body movements [65]. Using STIP features, a system for recognizing a person's affective state from face-and-body footage is proposed [66]. The survey on sentiment analysis and deep learning applications is discussed [67][68]. The backpropagation algorithm is used to construct a deep learning system for big data sets [69]. A self-organizing neural architecture was created to identify emotional states from full-body motion patterns [70]. For emotion recognition from video, a model combining CNNs and RNNs has been presented [71]. For the multimodal emoFBVP database, deep learning models with DCNN were created [72]. The results indicate a considerable improvement in accuracy when using a model with hierarchical feature representation for nonverbal emotion recognition [73]. Using promising neural network topologies, a unique design of an artificially intelligent system for emotion recognition is proposed [74]. Emotion Recognition in the Wild (EmotiW) is a system built utilizing a hybrid CNN-RNN architecture that outperforms existing methodologies [75]. A new framework for automatic emotional body gesture identification is being developed to differentiate culture and gender differences [76].

A quick comparison of the proposed approach with previous similar work is shown in Table 1.

| *Related Work* | *Algorithm* | *Dataset* | *Results* |
|---|---|---|---|
| Talegaonkar, K. Joshi, S. Valunj, R. Kohok, A. Kulkarni [77] | CNN + Batch Normalization | FER2013 | 60.12% |
| Amin, Chase, & Sinha [78] | CNN | FER2013 | 60.37% |
| A. Agrawal, N. Mittal [79] | CNN + Batch Normalization + Varying number of filters | FER2013 | 65% |
| O. Arriaga, M. Valdenegro-Toro, P.G. Plöger [80] | CNN + Batch Normalization + GAP | FER2013 | 66% |
| M. Quinn, G. Sivesind, G. Reis [81] | Custom CNN | FER2013 | 66.67% |
| Gede Putra Kusuma, Jonathan, Andreas, P Lim [82] | VGG-16 + GAP | FER2013 | 69.40% |
| Minaee, & Abdolrashidi [83] | Attentional CNN | FER2013 | 70.02% |
| H.-D. Nguyen, S. Yeom, I.-S. Oh, K.-M. Kim, S.-H. Kim [84] | Multi-Level CNN (MLCNN) | FER2013 | 73.03% |
| Amil Khanzada, Charles Bai and Ferhat Turker Celepcikav [85] | shallow CNNs and pre-trained networks based on SeNet50, ResNet50, and VGG16 | FER2013 | 75.8% |
| Isha Talegaonkar, Kalyani Joshi, Shreya Valunj, Rucha Kohok, Anagha Kulkarni [86] | CNN | FER2013 | 89.78% |
| Proposed | CNN | FER2013 | 91.64% |

Table 1: Comparison with related work.

## III. Facial Emotion Recognition Using Machine Learning Algorithms

Emotion recognition using machine learning algorithms is the basic method to perform this task. The first face is being detected and then facial expressions are being extracted using different techniques. Facial expressions extracted are then being used by machine learning classifiers to get the output of emotion. There are many different machine learning algorithms to perform this task that are being discussed.

### A. *Feature Extractors*

The face is being detected and then the face is being processed. There are many different techniques to extract important features from the face that are being discussed.

#### 1) *Face Registration and Representation*

Input facial image is being aligned with the similar data provided before. Landmark points are being used to point out important facial features such as eyebrows, eyes, nose, and mouth. This is done on the whole training dataset. After that, it is being processed by three mostly used algorithms i.e., Gabor Feature, Local Binary Pattern, and Histogram of Oriented Gradient. These three are the best-used feature extraction algorithms that provide accurate results. [1]

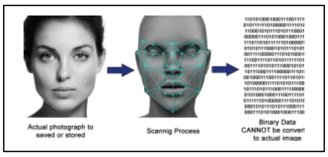

Actual photograph to saved or stored
Scannig Process
Binary Data CANNOT be convert to actual image

Figure 3.1: Face registration and representation.

#### 2) *Gabor Feature*

Gabor features can be compared to human visuals because of their orientation representations. By rotating and filtering one primary wavelet, these filters can be created. Among the various important image attributes, such as edge orientation histograms and box filters, they are the best.

For Gabor analysis, the eye centers must first be located before the images can be aligned properly. Transform, rotate, and scale are used to achieve this alignment. This is how 2D images are usually registered. Manual landmark determination is used to perform normalization. This is to avoid any registration scheme misalignment impacts. [2]

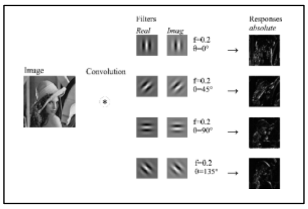


Figure 3.2: Gabor feature.

#### 3) *Local Binary Pattern (LBP)*

The texture and geometry of a digital picture are described by Local Binary Pattern. This is accomplished by segmenting a picture into tiny sections from which the characteristics are retrieved. These characteristics are binary patterns that characterize the pixels' surroundings in the areas. The acquired features from the areas are combined into a single feature histogram, which serves as an image representation. The similarity of their histograms may then be used to compare images. Face recognition utilizing the Local Binary Pattern approach, according to various studies, produces excellent results in terms of speed and discrimination. The approach appears to be highly resistant to face photos with various facial expressions, lighting circumstances, image rotation, and human aging. [4]

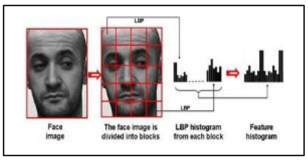

LBP
LBP
Face image
The face image is divided into blocks
LBP histogram from each block
Feature histogram

Figure 3.3: Local binary pattern.

The original Local Binary Pattern operator is a useful tool for describing the texture. The image's pixels are labeled by thresholding each pixel's 3x3-neighborhood with the center value and treating the result as a binary integer. After that, the labels' histogram may be utilized as a texture description. A diagram of the basic LBP operator may be seen in Figure 3.4. The operator was then expanded to include neighborhoods of various sizes. Any radius and number of pixels in the neighborhood may be achieved by using a circular neighborhood and bilinearly interpolating the pixel values.

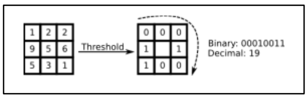


Figure 3.4: Original local binary pattern operator.

### 4) *Histogram of Oriented Gradients (HOG)*

Histogram of Oriented Gradient (HOG) is a feature extraction descriptor that is utilized in a variety of applications. Geometric and photometric modifications have no effect on the HOG features. HOG is implemented by splitting the picture into small linked sections called cells and producing a histogram of gradient directions or edge orientations for the pixels within each cell. The HOG descriptor is the result of combining these histograms.

There are two major parameters that define the HOG descriptor. The first argument is the cell size per row and column. The size of the cell corresponds to the size of the patch used to compute the histogram. The second argument is the number of bins orientation, which is mostly used to construct the gradient's angle intervals. [7]

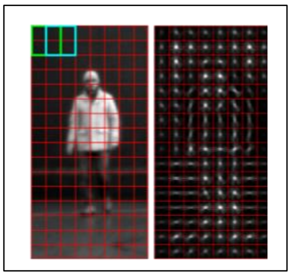


Figure 3.5: Histogram of oriented gradients.

### 5) *Dimensionality Reduction*

After the features are being extracted it is high-dimensional data and it contains many extra features that are of no use. These redundant features should be removed and high-dimensional data should be converted into low-dimensional data.

Laplacian eigenmaps algorithm can be used to convert high-dimensional data into low-dimensional data. The nearest point in high-dimensional space is converted into a close point in low-dimensional space using this procedure. The extended eigenvector problem is used to solve problems. The first n eigenvectors correspond to the first n eigenvalues in n-dimensional Euclidean space. A spectral regression algorithm is used to map high-dimensional data to find a projection function. High-dimensional data we get from feature extraction algorithm, i.e., Gabor Feature, Local Binary Pattern, and Histogram of Oriented Gradient. [1]

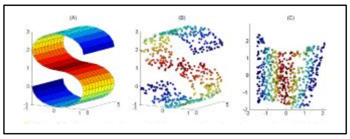


Figure 3.6: Dimensionality reduction.

## B. *Machine Learning Classifiers*

After the face is being processed and important features are being extracted from the face. Machine learning classifiers will be used to predict emotion. There are a variety of classifiers available. Some of them are discussed.

### 1) *Support Vector Machine (SVM)*

The Support Vector Machine (SVM) is a supervised learning technique for regression and classification [8]. SVMs belong to the generalized linear classifiers family. Support Vector Machine (SVM) is a classification and regression prediction tool that uses machine learning theory to enhance predicted accuracy while minimizing overfitting automatically.

Support Vector machines are high-dimensional feature space systems that use the hypothesis space of linear functions and are taught using an optimization theory learning algorithm with a statistical learning theory learning bias. The support vector machine rose to prominence in the NIPS community and is now an important part of machine learning research throughout the world. In a handwriting identification task, SVM achieves performance comparable to sophisticated neural networks with elaborated features when using pixel maps as input [9].

It's also used for things like handwriting analysis and facial analysis, with a focus on pattern recognition and regression applications.

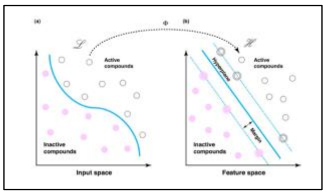


Figure 3.7: Support vector machine.

### 2) *Random Forest*

The random forest classifier consists of many tree classifiers, each of which is built using a random vector sampled independently from the input vector, and each tree casts a unit vote for the most popular class to categorize an input vector. [10]. The random forest classifier used in this study employs randomly selected attributes or a blend of features at each node to form a tree. Bagging, a method of producing a training dataset by randomly picking replacement N samples, where N is the size of the original training set, is used for each feature/feature combination chosen [10], was employed.

Any occurrences (pixels) are classified by picking the class with the most votes from the forest's tree predictors. The selection of an attribute selection measure as well as a pruning mechanism was required for the building of a decision tree. The bulk of the strategies for picking features for decision tree induction supply a quality measure directly to the attribute. In decision tree induction, the Information Gain Ratio (Quinlan) and the Gini Index are the most widely used attribute selection metrics. [10].

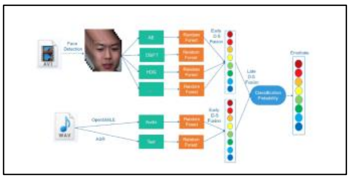


Figure 3.8: Random forest.

### 3) K-Nearest Neighbor (KNN)

The KNN (K-Nearest Neighbors) algorithm is a non-parametric, instance-based, or lazy approach that has been recognized as one of the most straightforward methods in data mining and machine learning [11][12][13]. The KNN method works on the idea that the most comparable samples in the same class have a high probability. In general, the KNN method determines the k nearest neighbors of a query in the training dataset and then predicts the query based on the main class in the k nearest neighbors. As a result, it was recently chosen as one of the top ten algorithms in data mining [14].

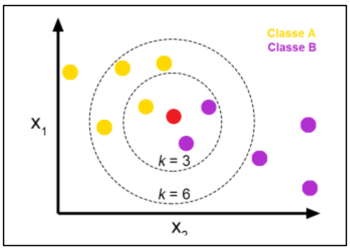


Figure 3.9: k-nearest neighbor.

### 4) Artificial Neural Networks (ANN)

An Artificial Neuron is simply a technology for designing biological neurons. It's a device with several inputs but just one output. An artificial neural network (ANN) is made up of many tiny processing units that are connected and layered together. [15][16]

Artificial neural networks, like actual neurons, have artificial neurons that receive inputs from other components or artificial neurons, and the result is translated into the output via a transfer function after the inputs are weighted and mixed. The transfer function might be a sigmoid, hyperbolic tangent function, or a step. [15]

Basically, computers are adept at calculations in that they accept inputs, analyze them, and then return a result based on calculations performed at specific Algorithms that are coded in software, but ANN enhance their own rules; the more judgments they make, the better the decisions may become [15].

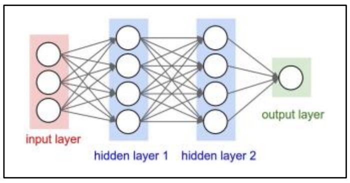


Figure 3.10: Artificial neural network.

### 5) Decision Tree

A graphical depiction of such mappings is a decision tree. A tree is a structure made up of a test node linked to two or more subtrees or a leaf node with a class label. A test node computes various outcomes based on the attribute values of an instance, with each possible result linked to one of the subtrees. An instance is classified starting at the root node of the tree. If this node is a test, the outcome of the instance is determined, and the process is restarted using the appropriate subtree. When a leaf is detected, its label provides the instance's expected class.

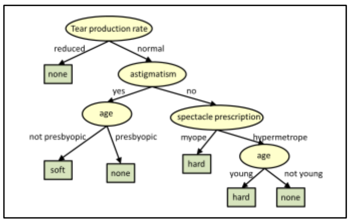


Figure 3.11: Decision tree.

## C. Deep Machine Learning Algorithm

Deep learning (also known as deep machine learning, deep structured learning, hierarchical learning, or DL) is a type of machine learning that uses a series of algorithms to learn. that tries to model high-level abstractions in data using Considering complicated structures or using model architectures. Otherwise, it's made up of a lot of non-linear transformations [17]. Deep neural networks, convolutional deep neural networks, deep belief networks, and recurrent neural networks have all been used to produce state-of-the-art results on various tasks in fields such as computer vision, automatic speech recognition, natural language processing, audio recognition, and bioinformatics. Deep learning has also been referred to as a buzzword or a rebranding of neural networks [18]. Deep learning is a class of machine learning algorithms that use for:

1. A cascade of numerous layers of nonlinear processing units is used for feature extraction and transformation. Each successive layer takes its input from the output of the previous layer. Pattern analysis (unsupervised) and classification are two applications for the supervised and unsupervised approaches (supervised) [19].

2. They are based on the (unsupervised) learning of several layers of data characteristics or representations in order to construct a hierarchical representation, with higher-level features derived from lower-level features.

3. They are a subset of the larger machine learning area of data representation learning.

4. They acquire a hierarchy of concepts by learning several levels of representations that correspond to various levels of abstraction [20].

### 1) *Convolutional Neural Network (CNN)*

CNN is an important area of deep learning, a neural feedforward network based on biological receptive field mechanisms [21], [22]. CNN does not need to manually extract features. CNN's design is influenced by visual perception. [23] CNN has been widely used in image analysis and speech recognition in recent years. However, using CNNs for card classification remains difficult [24]. Several CNN architectures have been introduced in the last decade. From 1989 to the present, various changes have been made to the CNN architecture. Such changes include structural reformulation, regularization, and parameter optimization. On the contrary, note that the significant improvement in CNN performance is mainly due to the reorganization of processing units and the development of new blocks. In particular, the latest developments in the CNN architecture for the use of network depth have been carried out. CNN enables the learning of data-driven, highly representative, layered hierarchical image features from sufficient training data [25]. Convolutional neural networks have recently demonstrated excellent image classification performance in large-scale visual recognition challenges. One of the functions of a CNN is to reduce images into a format that is easier to handle while preserving elements that are important for accurate prediction. This is crucial for creating a building that is not just beautiful but also functional.

#### *a) Usage of CNN*

Oquab M [26] showed how to use a limited amount of training data to effectively transfer image representations learned from CNNs to other visual recognition tasks in large annotated datasets. Gatys LA [27] used an image representation derived from a neural convolutional network optimized for object recognition. This makes the expanded image information explicit. The results provide new insights into depth imaging for learning neural convolutional networks and show their potential in advanced image synthesis and manipulation. Milletari F [28] uses convolutional neural networks (CNNs) to solve problems in the areas of computer vision and medical image analysis and proposes volume, full convolution, and 3D image segmentation techniques based on neural networks. bottom. Zbontar J [29] shows how to extract depth information from a modified image pair. It approached the problem by learning a similarity measure for small image fields using a convolutional neural network. Ma L [30] suggested using CNN for image question answering (QA) tasks. The CNNs proposed to provide an end-to-end framework with a convolutional architecture for learning not only the representation of images and questions but also the intermodal interactions to generate answers. Pathak D [31] suggested a way to learn high-density pixel labels from tags at the image level. Each image-level label applies restrictions to the output markup of the CNN classifier.

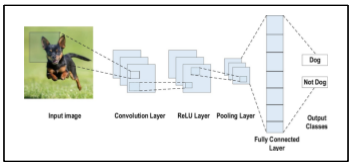


Figure 3.12: Structure of a CNN.

#### *b) CNN Architecture*

As illustrated in Figure 4.13, a CNN's overall structure includes an input layer, a convolution layer, a pooling layer, a fully connected layer, and an output layer.

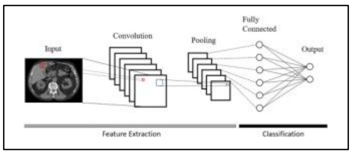


Figure 3.13: Structure of a typical CNN model.

Conv is used to extract features from the input picture, as demonstrated in figure 3.14. The Dot Product operation extracts the input picture using a convolution kernel made up of weight matrices to generate a labelled image.

A CNN is a deep learning system that takes an image as input, assigns relevance (learnable weights and biases) to various aspects/objects in the image, and can distinguish between them. A CNN requires substantially less preparation than conventional classification techniques. The architecture of a CNN is inspired by the arrangement of the visual cortex [32] and is akin to the connectivity network of neurons in the human brain.

A simple CNN is a series of layers and each layer that uses a differentiable function to transform one volume of activations to another. CNN architectures are built with three types of layers: Convolutional Layer, Pooling Layer, and Fully Connected Layer (exactly like in regular Neural Networks). These layers will be stacked to form the full CNN architecture.

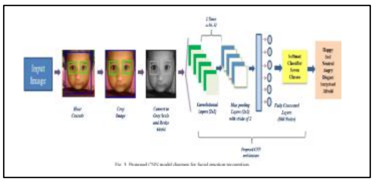


Figure 3.14: CNN architecture.

### I. *Convolution Layer*

The convolution operation's goal is to extract high-level characteristics from an input image, such as edges. The following are the functions of the convolution layer. Edges, color, gradient orientation, and rudimentary textures are all learned by the first convolutional layers.

The following convolutional layer (or layers) learns increasingly complex textures and patterns. Objects or portions of objects are learned in the final convolutional layers. The kernel is the component responsible for performing the convolution operation. A kernel filters out of the data that isn't relevant to the feature map, leaving only the relevant data. With a specific strid, the filter shifts to the right until it parses the entire width. Then, with the same stride length, it returns to the left of the image and repeats the procedure until the entire image has been traversed. Figure 6.3 shows a picture with dimensions of 5 X5 (shown in green) and a kernel filter of 3x3. The stride length is set to one, which causes the kernel to move nine times, each time doing a matrix multiplication of the kernel and the image beneath it. The input or kernel can have the same dimensions as the convolved feature. This is accomplished using same or valid padding. When the convolved feature has the dimensions of the input picture, it is the same padding, and when this feature has the dimensions of the kernel, it is valid padding. [33]

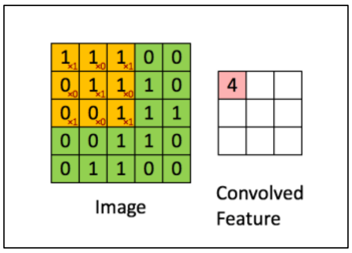


Figure 3.15: Convolving a 5×5 image with a 3×3 kernel to get a 3×3 convolved feature.

*II. Pooling Layer*

A convolved feature's spatial size is reduced by the pooling layer. This is done to reduce the number of computations needed to analyze the data and extract the most important properties, such as rotation and position invariance. Pooling can be divided into two types: maximum pooling and average pooling. The largest value from the portion of the picture covered by the kernel is returned by max pooling, while the average value is returned by average pooling. The outputs of max and average pooling on a picture are shown in Figure 3.16.

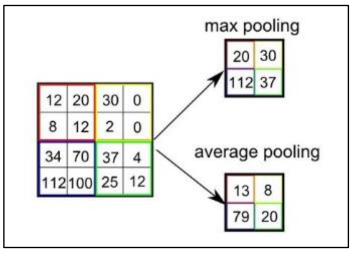


Figure 3.16: Max and average pooling outputs for an image [33].

In Pooling Operation, a completely connected layer's neurons are linked to all neurons in the previous layer. This layer appears at the end of a CNN. The preceding layer's input is flattened into a one-dimensional vector in this layer, and an activation function is used to generate the output [34].

*III. Fully Connected Layer*

All neurons in the previous layer are connected to neurons in a completely connected layer. A CNN's last layer is this layer. The output of this layer is obtained by flattening the input from the previous layer into a one-dimensional vector and applying an activation function. Because the last convolutional layer's receptive field does not cover the complete spatial dimension of the picture, the features generated by it correspond to a fraction of the input image in a shallow CNN model. As a result, only a few FC layers are required in this case. Despite their widespread use, hyperparameters such as the number of FC layers and neurons necessary in FC layers for a given CNN architecture to achieve improved performance are rarely investigated. [35]

*IV. Drop Out*

Dropout is a technique for avoiding overfitting. When the training accuracy is significantly higher than the testing accuracy, an ML model is said to be overfit. Dropout refers to the practice of neglecting neurons during training so that they are not taken into account during a certain forward or backward pass, resulting in a smaller network. Figure 3.17 shows an example of how these neurons are picked at random. The dropout rate is the likelihood of training a given node in a layer, with 1.0 indicating no dropout and 0.0 indicating that all layer outputs are disregarded [36].

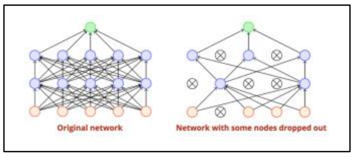


Figure 3.17: How ReLU works in CNNs

*V. Activation Function*

The activation function is one of the most crucial elements in the CNN model. They're utilized to learn and approximate any form of network variable-to-variable association that's both continuous and complex. In simple terms, it determines which model information should fire in the forward direction and which should not at the network's end. It gives the network non-linearity [37]. The ReLU, Softmax, tanH, and Sigmoid functions are some of the most often utilised activation functions. [38] Each of these functions has a distinct use. In a CNN model for binary classification, sigmoid and softmax functions are preferred, while in a multi-class classification, softmax is usually used. ReLU activation function, which has solid biological and mathematical foundations. It was proved in 2011 that it could improve deep neural network training. It works by setting the threshold=0, i.e., f (x) = max (0, x). Simply said, when x < 0 it outputs 0 and when x ≥ 0 it outputs a linear function (see Figure 6.7 for a graphic illustration). [39] Using ReLU can be not only as an activation function in each hidden layer of a neural network, but also as a classification function in the network's last layer.

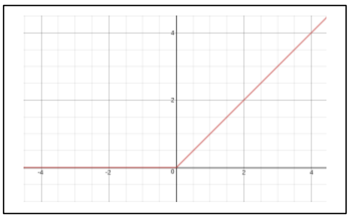


Figure 6.7: The Rectified Linear Unit (ReLU) activation function produces 0 as an output when x < 0, and then produces a linear with slope of 1 when x > 0.

In artificial neural networks, activation functions are employed to convert an input signal into an output signal, which is then sent as input to the next layer in the stack. We calculate the sum of products of inputs and their corresponding weights in an artificial neural network, then apply an activation function to it to acquire the output of that layer and feed it as the input to the next layer. The number of layers and, more crucially, the type of activation function used in a Neural Network determine its prediction accuracy. There is no manual that specifies the minimum or maximum number of layers that should be used to improve the accuracy and outcomes of neural networks, but a thumb rule suggests that at least two layers should be employed. [48]

### *c) Emotion Detection*

Emotion is also a significant part of intelligence. The challenge is that we want to develop a system that can discriminate between emotions, which forces scholars to use their own emotional cognitive system. Bionics and biology have been utilized to detect emotion in the voice [40]. They employ the physical anatomy of the human ear to design models to improve recognition performance, such as the MFCC and Lyon cochlear model [41], or to produce relevant characteristics for human ear perceived attributes.

### *d) Face detection using CNN*

L.Tan et al. used MTCNN to detect photos [42]. MTCNN is a face detection approach based on CNN. It employs three cascaded CNNs for rapid and accurate face detection as well as face detection learning (five facial landmarks detection, i.e., two eyes, two mouth corners, and nose). It creates a picture pyramid based on the input photos, which then feeds into a three-stage cascaded architecture. In the first stage, candidate regions are created, then refined in the second and third stages. The third stage produces the final detection findings as well as the associated facial landmark position [43].

### *e) CNN Model Parameters*

There are many choices for CNN architecture. How do we decide which is the best? We must first define what "best" implies. The best could be the simplest, or it could be the most effective at achieving accuracy while reducing processing complexity. The most effective method for determining an appropriate network structure is to check the accuracy. There is no such thing as a one-size-fits-all network, and only you are aware of the best model for your data. Cross validation is the most efficient approach to execute the required number of tests for reach the best accuracy in our algorithm.

#### *I. SGD Optimizer*

In the machine learning community, Stochastic Gradient Descent (SGD) is the most extensively used optimization approach. SGD's runtime performance has been optimized, and a theoretical basis for its empirical success has been developed, by researchers in both academia and industry. Recent advances in deep neural networks, for example, have largely been obtained because, surprisingly, SGD has been shown to be suitable for training them. SGD has problems navigating ravines, which are widespread around local optima and are regions where the surface curves significantly more sharply in one dimension than in another [46].

#### *II. Adam Optimizer*

We introduce Adam, a first-order gradient-based stochastic objective function optimization technique. The method is simple to use and is based on adaptive estimates of the gradients' lower-order moments. The method is computationally efficient, requires little memory, and is ideally suited to situations with a lot of data and/or parameters. The method can also be used to solve problems with non-stationary targets and/or very noisy and/or sparse gradients. By adapting to the geometry of the objective function, the approach is invariant to diagonal rescaling of the gradients. The hyper-parameters have straightforward interpretations and require little adjustment in most cases. There are some connections to related algorithms that Adam was inspired by. We also look at the algorithm's theoretical convergence features and offer a regret bound on the convergence rate that matches the best-known findings in the online convex optimization framework. We show that Adam performs well in practice and that it outperforms other stochastic optimization approaches [45].

#### *III. Epoch*

Among all the parameters, epoch has a considerable impact on the function's performance. The presentation of a set of training (input and/or target) vectors to a network, as well as the generation of new weights and biases, is referred to as an epoch. The training vectors can be supplied one by one or in a batch [49]. Epoch, more than any other parameter, has a significant impact on the function's performance. An epoch is defined as the presentation of a set of training (input and/or target) vectors to a network, as well as the development of new weights and biases. The training vectors can be delivered individually or in batches.

#### *IV. Learning Rate*

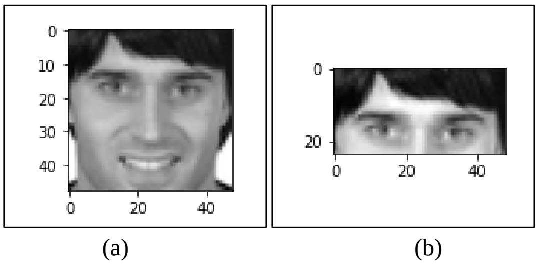
The "learning rate," or the amount of change to the model during each stage of the search process, is known as the "step size," and it is possibly the most critical hyperparameter to optimize for your neural network in order to get optimal performance on your challenge. The learning rate is an adjustable hyperparameter that has a modest positive value, usually between 0.0 and 1.0, and is used in the training of neural networks. The learning rate of a neural network model determines how quickly or slowly it learns a problem. And how to set a sensible default learning rate, diagnose behavior, and do a sensitivity analysis. Moreover, how to use learning rate schedules, momentum, and adaptive learning rates to boost performance even further. Backpropagation of error measures the amount of error that a node in the network's weights is accountable for during training. Instead of updating the weight with the full amount, the learning rate is used to scale it. This means that a learning rate of 0.1, which is a popular default setting, means that each time the weights in the network are updated, they are modified by 0.1 * (estimated weight error), or 10% of the estimated weight error.

## IV. Experimental Dataset

Dataset used in this project is Fer2013 from Kaggle. The dataset contains 3 columns and 35,887 rows. Columns are Emotion, Pixels, and Usage. The Emotion column contains integer values from 0 to 6, where 0 represents Angry, 1 is Disgust, 2 is Fear, 3 is Happy, 4 is Sad, 5 is Surprise, and 6 is Neutral. Pixel column contains images in form of a 48*48 matrix of pixels. Usage column has three values Training, represent training data which is used for training the model, Public Test used as a validation set, and Private Test used as testing set. Dataset is divided into 80% training set, 10% validation set, and 10% testing set based on the Usage column. Figure 4 is showing the structure of the dataset.

In the dataset, total of 35,887 images are being divided into different emotions. There are 4,953 images of Angry emotion, 547 images of Disgust emotion, 5,121 images of Fear emotion, 8,989 images of Happy emotion, 6,077 images of Sad emotion, 4,002 images of Surprise emotion, and 6,198 images of Neutral emotion. Figure 5.2 is illustrating the count of each emotion in the dataset.

In this project we first started with emotion recognition using Full Face. After that in according to the current Covid-19 situation where everyone is wearing a mask, we did the emotion recognition using Half Face (using eyebrows and eyes). For emotion recognition using full face, we had the dataset fer2013 but for emotion recognition using half face, we did not have the dataset available so we altered the original fer2013 dataset and crop all the images into half and then used this half face dataset to recognize emotion. In figure 4.1, (a) is showing the image of the full face which is from the original dataset and (b) is showing the image of half face with is from the altered dataset.



(a) (b)

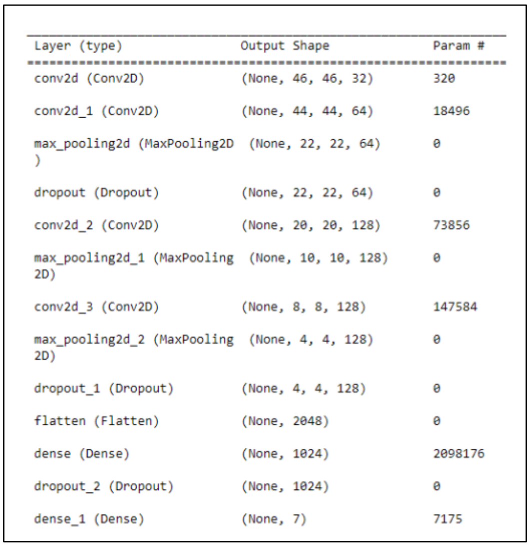
Figure 4.1: Image in the dataset before and after alteration.

## V. Experiment Performed

We did emotion recognition using full face and half face and changed different parameters such as the number of Epoch, the learning rate, SGD and Adam optimizer, to get better accuracy.

### A. Emotion Recognition using Full Face

Figure 5.1, is visualizing the CNN model we used to recognize emotion using full face images. In this CNN model, the input is of 48*48 pixels images. We compiled the CNN model using different parameter tuning. Below are the five experiments we performed using different parameter tuning.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 46, 46, 32)        320

 conv2d_1 (Conv2D)           (None, 44, 44, 64)        18496

 max_pooling2d (MaxPooling2D  (None, 22, 22, 64)       0
 )

 dropout (Dropout)           (None, 22, 22, 64)        0

 conv2d_2 (Conv2D)           (None, 20, 20, 128)       73856

 max_pooling2d_1 (MaxPooling  (None, 10, 10, 128)      0
 2D)

 conv2d_3 (Conv2D)           (None, 8, 8, 128)         147584

 max_pooling2d_2 (MaxPooling  (None, 4, 4, 128)        0
 2D)

 dropout_1 (Dropout)         (None, 4, 4, 128)         0

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 1024)              2098176

 dropout_2 (Dropout)         (None, 1024)              0

 dense_1 (Dense)             (None, 7)                 7175
```

Figure 5.1: CNN Model for Emotion Recognition using Full Face Images.

1) *Experiment 1*

In experiment 1, we used SGD optimizer, Epoch value 60, and Learning Rate of 0.0001.

2) *Experiment 2*

In experiment 2, we used the same SGD optimizer and Epoch value 60 but changed the Learning Rate to 0.001.

3) *Experiment 3*

In experiment 3, we used Adam optimizer, Epoch value 30, and Learning Rate of 0.0001.

4) *Experiment 4*

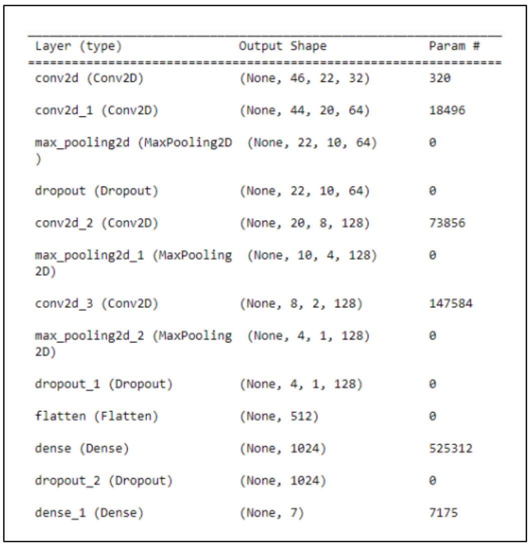
In experiment 4, we used the same Adam optimizer but changed the Epoch value to 60 and Learning Rate to 0.001.

5) *Experiment 5*

In experiment 5, we used the same Adam optimizer and Epoch value 60 but changed the Learning Rate to 0.0001.

### B. Emotion Recognition using Half Face

Figure 5.2, is visualizing the CNN model we used to recognize emotion using full face images. In this CNN model, the input is of 48*24 pixels images. We compiled the CNN model using different parameter tuning. Below are the four experiments we performed using different parameter tuning.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 46, 22, 32)        320

 conv2d_1 (Conv2D)           (None, 44, 20, 64)        18496

 max_pooling2d (MaxPooling2D  (None, 22, 10, 64)       0
 )

 dropout (Dropout)           (None, 22, 10, 64)        0

 conv2d_2 (Conv2D)           (None, 20, 8, 128)        73856

 max_pooling2d_1 (MaxPooling  (None, 10, 4, 128)       0
 2D)

 conv2d_3 (Conv2D)           (None, 8, 2, 128)         147584

 max_pooling2d_2 (MaxPooling  (None, 4, 1, 128)        0
 2D)

 dropout_1 (Dropout)         (None, 4, 1, 128)         0

 flatten (Flatten)           (None, 512)               0

 dense (Dense)               (None, 1024)              525312

 dropout_2 (Dropout)         (None, 1024)              0

 dense_1 (Dense)             (None, 7)                 7175
```

Figure 5.2: CNN Model for Emotion Recognition using Half Face Images.

#### 1) *Experiment 1*

In experiment 1, we used SGD optimizer, Epoch value 60, and Learning Rate of 0.0001.

#### 2) *Experiment 2*

In experiment 2, we used the same SGD optimizer and Epoch value 60 but changed the Learning Rate to 0.001.

#### 3) *Experiment 3*

In experiment 3, we used Adam optimizer, Epoch value 60, and Learning Rate of 0.0001.

#### 4) *Experiment 4*

In experiment 4, we used the same Adam optimizer and Epoch value 60 but changed the Learning Rate to 0.001.

## VI. Experiment Result

As we changed different parameters such as number of Epoch = 60 and 30, Optimizer= SGD and Adam, and the learning rate = 0.0001 and 0.001, we got different accuracy because of changing the parameters.

### A. *Emotion Recognition using Full Face*

After implementing all 5 experiments of the parameter change on the CNN model on full face images, below are the accuracies we got on the training set, on the validation set, and on the testing set.

| *Experiment* | *Training Accuracy* | *Validation Accuracy* | *Testing Accuracy* |
|---|---|---|---|
| *1* | 0.2464 | 0.2494 | 0.2449 |
| *2* | 0.2955 | 0.3137 | 0.3059 |
| *3* | 0.7245 | 0.6155 | 0.6133 |
| *4* | 0.8788 | 0.6225 | 0.6272 |
| *5* | 0.9164 | 0.6252 | 0.6715 |

### B. *Emotion Recognition using Half Face*

After implementing all 4 experiments of the parameter change on the CNN model on half face images, below are the accuracies we got on the training set, on the validation set, and on the testing set.

| *Experiment* | *Training Accuracy* | *Validation Accuracy* | *Testing Accuracy* |
|---|---|---|---|
| *1* | 0.2484 | 0.2494 | 0.2449 |
| *2* | 0.2508 | 0.2499 | 0.2449 |
| *3* | 0.5321 | 0.4249 | 0.4271 |
| *4* | 0.6339 | 0.4182 | 0.4319 |

## VII. Conclusion

This paper outlines the experiments done for emotion recognition using full face images and using half face images using Convolutional Neural Network (CNN).

After doing all the five experiments of parameter tuning on the CNN model using full face images and all the four experiments of parameter tuning on the CNN model using half face images, we got different accuracies for each.

Using full face images on CNN model, using Adam optimizer, Epoch value 60, and Learning Rate of 0.0001, we got the highest accuracy using the full face images of 91% (0.9164) on the training set, 62% (0.6252) accuracy on the validation set, and 67% (0.6715) accuracy on the testing set.

Using half face images on CNN model, using Adam optimizer, Epoch value 60, and Learning Rate of 0.001, we got the highest accuracy using half face image of 63% (0.6339) on the training set, 41% (0.4182) accuracy on the validation set, and 43% (0.4319) accuracy on the testing set.

In this project, we altered the original dataset of full face images to half face images. If there will be a dataset available containing images of people’s faces wearing a mask, we can get better accuracy.

## Acknowledgment

We would like to appreciate Dr. Safaa Bedawi whose guidance was very helpful in researching existing Emotion Recognition Techniques. We are also grateful to people pioneering the research in Emotion Recognition to give us a chance to explore the latest technology and best methods.

## References

[1] Mehta, Dhwani, Mohammad F.H. Siddiqui, and Ahmad Y. Javaid, "Recognition of Emotion Intensities Using Machine Learning Algorithms: A Comparative Study", Sensors, 2019.

[2] Hassan, Masoud & Hussein, Haval & Eesa, Adel & Mustafa, and Ramadhan, "Face Recognition Based on Gabor Feature Extraction Followed by FastICA and LDA", Computers, Materials, and Continua, 2021.

[3] Ahonen, Timo & Hadid, Abdenour & Pietikäinen, Matti, "Face Recognition with Local Binary Patterns", 8th European Conference on Computer Vision, 2004.

[4] O. S. Kulkarni, S. M. Deokar, A. K. Chaudhari, S. S. Patankar and J. V. Kulkarni, "Real Time Face Recognition Using LBP Features", 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), 2017.

[5] Md. Abdur Rahim, Md. Najmul Hossain, Tanzillah Wahid, & Md. Shafiul Azam, "Face Recognition using Local Binary Patterns (LBP)", Global Journal of Computer Science and Technology Graphics & Vision, Volume 13, Issue 4, 2013.

[6] Muhammet Fatih Aslan, Akif Durdu, Kadir Sabanci, Meryem Afife Mutluer, "CNN and HOG based comparison study for complete occlusion handling in human tracking", Measurement, Volume 158, 2020.

[7] Deniz, Oscar & Bueno, Gloria & Salido, Jesús & De la Torre, Fernando, "Face recognition using Histograms of Oriented Gradients", Pattern Recognition Letters, 2011.

[8] Vapnik, V., Estimation of Dependencies Based on Empirical Data. Empirical Inference Science: Afterword of 2006, Springer, 2006.

[9] Nello Cristianini and John Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods", Cambridge University Press, 2010.

[10] Breiman L Random forests—random features. Technical Report 567, Statistics Department, University of California, Berkeley, 1999.

[11] Zhang, S. Jin, Z. Zhu, X. Zhang, J., "Missing data analysis: A kernel-based multi-imputation approach", Gavrilova, M.L., Tan, C.J.K. (eds.) Transactions on Computational Science III. LNCS, vol. 5300, pp. 122–142. Springer, Heidelberg 2009.

[12] Zhu, X., Zhang, L., Huang, Z.: A sparse embedding and least variance encoding approach to hashing. IEEE Transactions on Image Processing 23(9), 3737–3750, 2014.

[13] Zhu, X., Zhang, S., Jin, Z., Zhang, Z., Xu, Z.: Missing value estimation for mixed-attribute data sets. IEEE Transactions on Knowledge and Data Engineering 23(1), 110–121, 2011.

[14] Zhu, X., Huang, Z., Shen, H.T., Zhao, X.: Linear cross-modal hashing for efficient multimedia search. In: ACM Multimedia, pp. 143–152, 2013.

[15] Ajith Abraham, "Artificial Neural Networks", Stillwater, OK, USA, 2005.

[16] Prof. Leslie Smith, " An Introduction to Neural Networks", University of Stirling., 1996,98,2001,2003.

[17] R. Collobert, "Deep Learning for Efficient Discriminative Parsing". videolectures.net. Ca. 7:45., May 6, 2011.

[18] G. Lee. "Machine-Learning Maestro Michael Jordan on the Delusions of Big Data and Other Huge Engineering Efforts". IEEE Spectrum, 20 October 2014.

[19] P. Glauner, "Comparison of Training Methods for Deep Neural Networks", arXiv:1504.06825,2015.

[20] S. Hyun Ah, and S.Y, Lee. "Hierarchical Representation Using NMF." Neural Information Processing. Springer Berlin Heidelberg, 2013.

[21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, pp. 436-444, May 2015.

[22] F. B. Zhou, L. H. Zou, X. J. Liu, and F. Y. Meng, "Micro landform classification method of grid DEM based on convolutional neural network", *Geomatics Inf. Sci. Wuhan Univ.*, 2018.

[23] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[24] Luo, L. Liu, J. Yin, et al., "Deep learning of graphs with Ngram convolutional neural networks". IEEE Trans. Knowl. Data Eng. 29(10), 1–1, 2017.

[25] S. H.-Chang, H.R. Roth, M. Gao, et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset. M. " 2016.

[26] M Oquab, L. Bottou, I. Laptev, et al., "Learning and transferring mid-level image representations using convolutional neural networks ", 2014.

[27] L.A. Gatys, A.S. Ecker, M. Bethge, "Image Style Transfer Using Convolutional Neural Networks", IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[28] F. Milletari, N. Navab, S.A. Ahmadi, V-Net: "Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation", 2016.

[29] J. Zbontar, Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches".J. Mach. Learn. Res. 17(1–32), 2,2016.

[30] L. Ma, Z. Lu, H. Li, "Learning to answer questions from image using convolutional neural network", AAAI 3(7), 16 ,2016.

[31] D. Pathak, P. Krahenbuhl, T. Darrell, Constrained convolutional neural networks for weakly supervised Segmentation, IEEE International Conference on Computer Vision. IEEE Computer Society (ICCV2015, Santiago, pp. 1796–1804, 2015.

[32] A. Amini, A. Soleimany, "MIT deep learning open access course", 2020.

[33] T. Lindeberg, "Scale invariant feature transform," Scholarpedia, volume 7, May 2012.

[34] Yu, D., et al. Mixed pooling for convolutional neural networks. in International conference on rough sets and knowledge technology. 2014. Springer.

[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015.

[36] D. M. Hawkins, "The problem of overfitting," J. Chem. Inf. Comput. Sci., vol. 44, no. 1, pp. 1–12, Jan. 2004.

[37] Zuo, Z.; Li, J.; Wei, B.; Yang, L.; Fei, C.; Naik, N. Adaptive Activation Function Generation Through Fuzzy Inference for Grooming Text Categorisation. In Proceedings of the 2019 IEEE International Conference on Fuzzy Systems, New Orleans, LA, USA, 23–26 June 2019.

[38] Lohani, H.K.; Dhanalakshmi, S.; Hemalatha, V. Performance Analysis of Extreme Learning Machine Variants with Varying Intermediate Nodes and Different Activation Functions. In Cognitive Informatics and Soft Computing; Springer: Singapore, 2019.

[39] Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature 405, 6789 947, 2000.

[40] M. Drolet, Ricarda I. Schubotz, Julia Fischer: “Explicit authenticity and stimulus features interact to modulate BOLD response induced by emotional speech”, Cogn Affect Behav Neurosci 13, 2013.

[41] L. Caponetti, C. Alessandro Buscicchio, and G. Castellano: “Biologically inspired emotion recognition from speech”, EURASIP Journal on Advances in Signal Processing, 2011.

[42] L. Tan, K. Zhang, K. Wang, X. Zeng, X. Peng, and Y. Qiao, “Group emotion recognition with individual facial emotion CNNs and global image based CNNs,” in Proc. 19th ACM Int. Conf. Multimodal Interact., Nov. 2017.

[43] K.Zhang, Z. Zhang, Z. Li, and Y. Qiao. “Joint face detection and alignment using multitask cascaded convolutional networks” IEEE Signal Processing Letters 23, 10, 2016.

[44] D. P. Kingma and J. L. Ba, “ADAM: a method for stochastic optimization,” in Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), 2015.

[45] Richard S. Sutton. Two problems with backpropagation and other steepest-descent learning procedures for networks, 1986.

[46] Face detection in colour images Rein-Lein Hsu, Student member IEEE, Mohamed Abdel Mottaleb, Member, IEEE and AnilK.Jain, Fellow, IEEE, 2020.

[47] B. Yang, J. Yan, Z. Lei, and S. Z. Li, “Convolutional channel features,” in IEEE International Conference on Computer Vision, 2015.

[48] AGOSTINELLI, F., HOFFMAN, M., SADOWSKI, P., & BALDI, P. (2014). LEARNING ACTIVATION FUNCTIONS TO IMPROVE DEEP NEURAL NETWORKS. ARXIV PREPRINT ARXIV:1412.6830.

[49] Bowden, G.J., Maier, H.R., Dandy, G.C.: Optimal division of data for neural network models in water resources applications. Water Resour. Res., 2002.

[50] Cowie, R.; Douglas-Cowie, E.; Tsapatsoulis, N.; Votsis, G.; Kollias, S.; Fellenz, W.; Taylor, J.G. Emotion recognition in human-computer interaction. IEEE Signal Process. Mag., 2001.

[51] Busso, C.; Deng, Z.; Yildirim, S.; Bulut, M.; Lee, C.M.; Kazemzadeh, A.; Lee, S.; Neumann, U.; Narayanan, S. Analysis of emotion recognition using facial expressions, speech, and multimodal information. In Proceedings of the 6th International Conference on Multimodal Interfaces, State College, PA, USA, 14–15; ACM: New York, NY, USA, 2004; pp. 205–211, October 2004.

[52] Ayadi, M.; Kamel, M.S.; Karray, F. Survey on speech emotion recognition: Features, classification schemes, and databases. Pattern Recognit., 2011.

[53] Wu, C.H.; Chuang, Z.J.; Lin, Y.C. Emotion recognition from text using semantic labels and separable mixture models. ACM Trans. Asian Lang. Inf. Process. TALIP 5, 2006.

[54] Ekman, Paul. Are there basic emotions? Psychological Review, 1992.

[55] Mehrabian, Albert. Silent messages. Wadsworth, 1971.

[56] Scherer, Stefan and Stratou, Giota and Mahmoud, Mohamed and Boberg, Jill and Gratch, Jonathan and Rizzo, Alessandro and Morency, Louis-Philippe. Automatic behavior descriptors for psychological disorder analysis. Automatic Face and Gesture Recognition (FG), 10th IEEE International Conference and Workshops on. IEEE, 2013.

[57] Shan, Caifeng and Gong, Shaogang and McOwan, Peter W. Facial expression recognition based on local binary patterns: A comprehensive study. Image Vision Comput., 2009.

[58] [58] Shaker, Noor, and Asteriadis, Stylianos and Yannakakis, Georgios N and Karpouzis, Kostas. A game-based corpus for analyzing the interplay between game context and player experience. Affective Computing and Intelligent Interaction, Springer, 2011.

[59] Khorrami, Pooya, and Paine, Tom Le and Brady, Kevin and Dagli, Charlie and Huang, Thomas S. How deep neural networks can improve emotion recognition on video data. arXiv preprint arXiv:1602.07377, 2016.

[60] Mehta, Dhwani, Mohammad F.H. Siddiqui, and Ahmad Y. Javaid, "Recognition of Emotion Intensities Using Machine Learning Algorithms: A Comparative Study", Sensors, 2019.

[61] Hassan, Masoud & Hussein, Haval & Eesa, Adel & Mustafa, and Ramadhan, "Face Recognition Based on Gabor Feature Extraction Followed by FastICA and LDA", Computers, Materials, and Continua, 2021.

[62] Shan, L.; Deng, W. Deep facial expression recognition: A survey. IEEE Trans. Affect. Comput, 2020.

[63] Shervin Minaee, Mehdi Minaei, and Amirali Abdolrashidi. Deep-emotion: Facial expression recognition using attentional convolutional network. Sensors, 2021.

[64] J. Arunnehru, M. Kalaiselvi Geetha. “Automatic Human Emotion Recognition in Surveillance Video”, Intelligent Techniques in Signal Processing for Multimedia Security, Springer-Verlag, 2017.

[65] D.Holden, J.Saito, T.Komura. “A Deep Learning Framework for Character Motion Synthesis and Editing” SIGGRAPH ’16 Technical Paper, July 24 - 28, Anaheim, CA, 2016.

[66] H. Gunes, C.Shan, Sh.Chen, Y.Tian. “Bodily Expression for Automatic Affect Recognition. Emotion Recognition: A Pattern Analysis Approach” Published by John Wiley & Sons, Inc., 2015.

[67] L.Zhang, Sh.Wang, B.Liu. “Deep Learning for Sentiment Analysis”, 2018.

[68] H.Brock, “Deep learning – Accelerating Next Generation Performance Analysis Systems”, 12th Conference of the International Sports Engineering Association, Brisbane, Queensland, Australia, 2018.

[69] Y. Le Cun, Y. Bengio, Geoffrey Hinton. "Deep learning", Nature, Volume 521, 2015.

[70] N. Elfaramawy, Pablo Barros, German I. Parisi, Stefan Wermter. "Emotion Recognition from Body Expressions with a Neural Network Architecture", Session 6: Algorithms and Learning, Bielefeld, Germany, 2017.

[71] P. Khorrami, Tom Le Paine, Kevin Brady, Charlie Dagli, Thomas S. Huang. "How Deep Neural Networks Can Improve Emotion Recognition on Video Data", 2017.

[72] H. Ranganathan, Sh. Chakraborty, S. Panchanathan, "Multimodal Emotion Recognition using Deep Learning Architectures", 2017.

[73] P. Barros, D. Jirak, C. Weber, S. Wermter, "Multimodal emotional state recognition using sequence dependent deep hierarchical features", Neural Networks, 2015.

[74] E. Correa, A. Jonker, M. Ozo, R. Stolk, "Emotion Recognition using Deep Convolutional Neural Networks", 2016.

[75] S. Ebrahimi, V. Michalski, Kishore Konda, Roland Memisevic, Christopher Pal, "Recurrent Neural Networks for Emotion Recognition in Video", ICMI 2015, Seattle, WA, USA., 2016.

[76] F. Noroozi, C. Adrian Corneanu, D. Kami´nska, T. Sapi´ nski, S. Escalera, and G. Anbarjafari, "Survey on Emotional Body Gesture Recognition", Journal of IEEE Transactions on Affective Computing, 2015.

[77] I. Talegaonkar, K. Joshi, S. Valunj, R. Kohok, A. Kulkarni, "Real Time Facial Expression Recognition using Deep Learning," SSRN Electronic Journal, 2019.

[78] Amin, D., Chase, P., & Sinha, K., "Touchy Feely: An Emotion Recognition Challenge", Palo alto: Stanford, 2017.

[79] A. Agrawal, N. Mittal, "Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy," Visual Computer, 2020.

[80] O. Arriaga, M. Valdenegro-Toro, P.G. Plöger, "Real-time convolutional neural networks for emotion and gender classification," ESANN 2019 - Proceedings, 27th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2019.

[81] M. Quinn, G. Sivesind, G. Reis, "Real-time Emotion Recognition From Facial Expressions", 2017.

[82] Gede Putra Kusuma, Jonathan, Andreas Pangestu Lim, "Emotion Recognition on FER-2013 Face Images Using Fine-Tuned VGG-16", Advances in Science, Technology and Engineering Systems Journal, Volume 5, 2020.

[83] S. Minaee, A. Abdolrashidi, "Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network", 2019.

[84] H.-D. Nguyen, S. Yeom, I.-S. Oh, K.-M. Kim, S.-H. Kim, "Facial expression recognition using a multi-level convolutional neural network", Proceedings from the International Conference on Pattern Recognition and Artificial Intelligence, 2018.

[85] Amil Khanzada, Charles Bai, and Ferhat Turker Celepcikav, "Facial Expression Recognition with Deep Learning Improving on the State of the Art and Applying to the Real World", 2019.

[86] Isha Talegaonkar, Kalyani Joshi, Shreya Valunj, Rucha Kohok, Anagha Kulkarni, "Real Time Facial Expression Recognition using Deep Learning", International Conference on Communication and Information Processing (ICCIP-2019), 2019.