A Review of Techniques Related to Automated Facial Expression Recognition

Austin Dase
Department of Computer and Information Sciences
Towson University
Towson, Maryland 21252
Email: adase1@students.towson.edu

CONTENTS

I. INTRODUCTION

The ability to recognize emotion displayed by the face is a critical part of human intelligence. This ability that is second-nature to humans is non-trivial for machines faced with the same task. Research has shown that humans can correctly identify expressions of emotion in facial images [1] with a consistently high degree of accuracy while computers have historically struggled when faced with similar tasks.

The goal of this paper is to survey the existing research that has been done on facial expression recognition in still images. Systems that have implemented facial expression recognition and that have implemented capabilities related to facial expression recognition are examined and reviewed. A robust facial expression recognition system at it's most basic level is an input-output system where inputs are images and outputs are the determined emotion expressed. In the following sections, three distinct modules that make up this type of system are examined: face detection, feature extraction, and classification. The first stage of a robust facial expression recognition system is determining if the image contains a face or not, this is the goal of the face detection module. Various methods for face detection are examined in II. If no face is detected then the system can cease processing. If a face is detected then the system can proceed with the second module, feature extraction. Feature extraction, examined in III, concerns itself with preparing the image data for the classification module. During the feature extraction stage, relevant data is extracted and transformed to help optimize the classification process. Because image files can have high volumes of extraneous data this phase is often concerned with reducing the volume of data that will be used to make a classification decision in order to improve performance. Finally, the classification module uses the output data from the feature extraction module to classify the emotion being expressed in the image. In IV, methods for classification are examined.

The result of this paper will be a consolidation of methods and techniques that can be used as part of a robust system for facial expression recognition in images. Based on the findings of this work, in future work a system will be proposed, implemented, and compared against prior work in the field.

II. FACE DETECTION

The first step in any system seeking to classify facial images is identifying the presence of a face in an image. The process of face detection involves two steps: determining if there is a face or not and then determining the boundaries of the face in the image. The primary challenge in the face detection stage is the trade-off between false positives/negatives and performance. In the following section, three techniques for detecting faces in images are presented: template models, geometric models and machine learning models.

A. Template Models

A method for detecting faces in images using template matching combined with support vector machines (SVM) is described in [2]. Templates are used for identifying face candidate areas and a SVM classifier is used to classify candidate areas as face or non-face areas.

The first stage in the model described in [2] is to utilize templates to filter out face candidate images for the classifier to analyze. Two templates are used for this stage: eyes-in-whole and face itself [2]. The two templates were generated through analysis of a set of 50 mugshots that were aligned, normalized for size, and then averaged to create one image called the average face [2]. The average face image is then sampled for two regions: the 20 x 20 whole face region and the 20 x 8 eyes-in-whole region [2]. These templates are applied to the testing images by finding correlation coefficients for intensity for the test image related to the templates. Images where both templates match over a threshold of .25 are retained for SVM analysis [2].

The subset of images that are identified as face candidate images are then classified by a SVM as either containing a face or not. The SVM was trained on 5125 face sample images. In order to improve training speed and performance the negative training examples were collected using samples identified incorrectly by the template matching method [2]. After training and testing once with incorrectly labeled images from template matching, all images incorrectly labeled by the SVM were subsequently used in the next iteration of training as negative examples [2]. This process was then repeated until an appropriate number of negative samples were available for testing [2].

In order to detect faces of varying scale in the original image, a pyramid of images, each a sub-sample of a ratio of 1.2 to the next, was generated and filtered using template matching and SVM [2].

For a test set of 230 images with a total of 545 faces, the algorithm described in [2] detected 516 faces with a positive detection rate of 94.7% and 815 false-positives. These results on the same test set compared with system 5 (single neural network) and system 11 (two neural networks) described by [3] show that systems described by [3] produce higher detection rates, 97.8% and 96.3% respectively. System 5 produces a much higher number of false-positives (1841) and System 11 produced a lower number of false positives (87).

The algorithm is also tested on the test set from [3] which consists of 130 images with 507 faces. The template matching and SVM system detected 415 faces with a detection rate of 81.9% and 465 false-positives. System 5 by [3] detected 459 faces with a detection rate of 90.5% and 570 false-positives. The System 11 by [3] detected 437 faces with a detection rate of 86.2% and 23 false-positives.

Overall the experimental results indicate that the template and SVM based system may be superior to System 5 proposed by [3] and inferior to System 11 proposed by [3].

B. Geometric Models

A method for detecting faces in images using geometric rules is proposed by [4]. In [4] know geometric characteristics of facial feature are used during analysis to detect the presence of a face in an image. The first step in the processes outlined by [4] is to read in the image and convert it to a binary image. The second step is to label all 4-connected components in the resulting images to form blocks of interest and then to determine the center of each of those blocks [4]. The third step identifies all groups of three blocks whose centers form an isosceles triangle. Finally, all blocks that are part of triangles found in step three are labeled as potential face regions [4].

The binary conversion first transforms any color images to grayscale by eliminating hue and saturation and retaining luminance [4]. All grayscale images are then thresholded (relying on the assumption that objects of interest are darker than the background) with a given threshold T. Morphological operations of erosion, then dilation (also called opening) are performed on the resulting binary images to remove noise followed by dilation and erosion (also called closing) to close holes.

Next, 4-connected components are identified in the image. Based on the idea that two eyes and a mouth will form an isosceles triangle [4], all sets of 4-connected blocks are then tested for conformance to rules, allowing for 25% deviation, that the centers of the three blocks create an isosceles triangle. After all isosceles triangles are identified as potential face regions, they are verified using a combination of size normalization, weighted mask and threshold to determine if the potential face region is a verified face.

All identified potential face regions are normalized in size (pixel height and width) using bi-cubic interpolation [4]. Based on the results of 10 binary training faces, the authors determined a target mask with which to compare candidate regions. Based on the similarity between the target mask and the mask obtained from the normalized face region, the region is classified as either a face region or a non-face region. The method for creating the target mask and for computing the weighted comparison is described in [4].

Experimental results from testing the system described in [4] showed that execution time for face detection varies based on size, resolution, and complexity of the images.

C. Machine Learning Models

1) AdaBoost: A method for face detection in images that attempts to balance performance with accuracy is described in [5]. By refining concepts from prior research and providing novel methods for calculation [5] attempt to achieve a more accurate system for face detection that improves performance over existing models.

The integral image concept is introduced by [5] as a method for pre-processing to improve performance. The integral image is created by scanning the image one time from left to right and generating an integral image where the value at each point (x, y) in the integral image is equal to the sum of all points above and to the left of the point (x, y) in the original image. This concept is expressed by [5] in the below formula where $ii(x, y)$ represents a point on the integral image and $i(x', y')$ represents a point on the original image:

$$ii(x,\ y)\ =\ \sum_{x' \leq x,\, y' \leq y} i(x',\ y') \tag{1}$$

After the integral image is created, it is used for computing values for rectangle features defined by [5]. The rectangle features are employed by [5] in an effort to improve performance over pixel-based classification methods. Three kinds of rectangle features are used: two-rectangle features, three-rectangle features, and four-rectangle features. Two-rectangle features represent the difference between the sum of two adjacent and equal-sized rectangular regions, three-rectangle features represent the sum of an inner rectangle subtracted from the sum of two adjacent and equal sized outer rectangles and four-rectangle features represent the difference in sums of diagonally adjacent and equal sized rectangles [5].

A principal benefit of the integral image is the computational advantage it provides over other methods. Most face detection methods scan the image in search of faces at multiple scales. In the example provided by [5], the scale of 24 x 24 pixels is used for generating the first set of rectangle features. Next, the size of the rectangle features is increased by a factor of 1.25 and the rectangle features are recalculated. This process is repeated until the rectangle feature is too large for the image, the number of times given by:

$$min(x', y') = max(x, y)(S)^n \tag{2}$$

$$n = \rceil \log_S max(x', y') \tag{3}$$

Where S is the scaling factor of the rectangle features, (x', y') are the dimensions in pixels of the image and n is the number of iterations of scaling features to produce. An image of size 384 x 288, starting with a base rectangle feature scale of 24 x 24 will produce 160,000 features for each image. Not only is this a large number of features, but the calculation required for each one is intensive and repetitive. [5] show that by using the integral image, the amount of computation necessary to compute each of these features is greatly reduced.

In order to perform classification, a variant of the AdaBoost algorithm used for feature selection is used [5]. For each feature, a weak learning algorithm first determines the optimal threshold of the classification function [5].From there multiple rounds of weak learning classification are performed and finally result in a strong classifier that is the combination of all the weak classifiers [5]. The result is a classification of the region as either a face or a non-face region.

D. Neural Network Model

A model using neural networks (NN) to detect the presence of a face in images is described in [3]. The models proposed by [3] utilizes multiple layers of neural networks to perform feature extraction and detection. In order to accommodate for the commonality of large variances in lighting, occlusion, pose, expression, and identity, [3] states that multiple classifiers should be used to help handle these variations and that the output of those classifiers can be used as input into a final classifier that determines the presence of a face in the image.

The systems for face detection proposed by [3] consists of 4 primary modules: localization and pose estimation, preprocessing, detection, and arbitration. Localization and pose estimation is performed with a neural network that analyzes the pixel values of the image as inputs. Localization determines the approximate boundary of the head in the image. Pose estimation determines the approximate angle of any tilt or roll from upright of the head in the image. Preporcessing is preformed using traditional image processing techniques for improving brightness and contrast as well as to reduce the impact of variations caused by lighting or camera quality [3]. The detection phase, performed with a neural network, makes an initial determination as to whether there is a face in the region or not. Finally, arbitration is performed with another neural network that analyzes the results of prior stages to make the final determination if the face detected in other stages is valid or not.

The first system presented by [3] is concerned with upright face detection. First, the image is segmented into regions of 20 x 20, 10 x 10, 5 x 5 and 20 x 5 pixels and those regions are fed to a neural network classifier that determines locations that might contain a face. This same process is repeated using a sub-sample of the original image and applying the detection networks again. The outputs of these networks are connected to a final arbitrator network that makes the determination as to whether or not the original 20 x 20 region contains a face [3].

The second system proposed by [3] is concerned with tilted face detection. The system for upright detection is prepended with another neural network that detects the possible tilt or rotation of the face in the window and performs the necessary rotation to make the face upright for the face detector [3]. Even if there is no face in the window, the rotation is still applied as the non-face window will still result in a non-face detection [3].

The systems proposed by [3] both performed well on the testing data compared to prior techniques. The upright face detection system was evaluated with a testing data set of images compiled by the author as well as images from the FERET database that contain frontal faces. The tilted face detection system was evaluated using the same data set collected for the upright detection algorithm as well as images from the FERET database, classified into three groups based on how far the face in the image is aligned from the frontal position [3].

III. Feature Extraction and Selection

In any facial expression recognition system there are at least two stages: feature extraction and classification [6]. In the feature extraction stage, features (also called attributes) are extracted from the raw image data and are subsequently used as input to the classification stage of the process. The quality of the feature extraction methodology therefore has a major impact on the ability of the classification stage to accurately identify the expression displayed in the original image [6]. Each of these two stages can be further decomposed into submodules and can be performed in a variety of ways.

In any automatic classification system, a major subset of the system will be concerned with extracting features from the raw data in a format that the classification algorithm can process successfully. The feature extraction process can be decomposed into two steps: feature construction and feature selection [7]. Feature construction concerns itself mostly with transforming the raw data into a format that the system can process; features used as input to classification algorithms can be binary, continuous or categorical [7]. Common feature construction methods include: standardization, normalization, signal enhancement, feature discretization, and non-linear expansion [7]. Feature selection then analyzes the output of the feature construction process and selects the subset of information that will be used as features for classification. While feature selection's primary goal is to filter out noise by selecting the most relevant and informative features, it can also reduce the size of data that must be processed thereby improving algorithm performance and reducing storage requirements [7].

Construction of features from facial images generally takes one of two approaches: geometric feature-based or appearance-based [8]. Geometric feature-based approaches include information about the shape and location of facial components like the mouth, eyes, nose, and eyebrows. These components are represented as features that encode the geometric information each of the components for use in classification [8]. Appearance-based methods apply image filters in order to extract relevant data for use in classification and focus on either the entire face or select regions of the face image without regard to the geometric location of the features [8].

After the feature set has been extracted, it is often necessary to select a subset of features for use in classification. There are two general techniques for feature selection: filter-based methods and wrapper or embedded based methods [7]. Filter-based methods rank features by their relevance or importance to the classification model. Rankings are commonly produced by statistical analysis and do not provide any optimization for prediction performance [7]. Wrapper, or embedded methods utilize a learning model for selection of an optimal feature subset. Features which have the strongest predictive power for the model are selected for use in classification. Embedded feature selection refers to analysis that is done during the training of a model [7].

Feature extraction and selection methods are examined by [6] in order to compare accuracy of different models. Models examined by [6] include Gabor filters, log Gabor filters, local binary pattern (LBP) operators, higher-order local autocorrelation (HLAC) and HLAC-like features (HLACLF). A system is proposed that includes pre-processing, face detection, facial feature detection, facial feature extraction, feature selection, training, and classification modules.

The pre-processing module generates an image that is uniform in shape and size, has normalized intensity, and depicts a face expressing an emotion [6]. In order to detect the face area, the Viola-Jones method [5] is used. The result of the pre-processing is a new image of pixels that the face area detection determined are within the face area.

The methods for feature extraction use by [6] are appearance-based extraction methods. For feature selection, [6] employ both filter and wrapper based methods. Mutual information selection seeks to select the features that have the most mutual information between the feature and the classes [6]. Minimum redundancy seeks to select features that have the least amount of redundant information between each other [6]. A wrapper based optimization approach where classification error is iteratively reduced by removing features at each iteration that have the least predictive power. This wrapper based approach identifies the subset of features that produces the smallest error when classifying the training set of data [6].

A Naive Bayesian classifier is used for final classification for images from the Cohn-Kanaade database and a K-NN classifier was used for images from the JAFFE database.

The JAFFE and Cohn-Kanade database were used for experimental analysis. For each database, the subjects in the labeled images displayed one of six basic emotions: anger, disgust, fear, happy, sad, and surprise. Classes were each of these emotions as well as a neutral class. Accuracy for each feature extraction methods was compared based on the feature selection method used. For each combination of feature extraction and feature selection method, training and testing was performed three training/testing times and the average result was reported. There was no overlap between subjects in the training set and the testing set in order to ensure person-independent classification.

A. Facial Animation Parameters Extraction

A robust feature detection method for facial expression recognition in video sequences is presented in [9]. 19 facial feature points are extracted based on facial animation parameters (FAP) related to the facial action coding system (FACS). In addition to feature extraction, a confidence factor for each extracted point was produced allowing the expression classification stage to take into consideration the quality of each of the extracted features [9].

The initial phase of face detection identified face and non-face areas in the image through nonparametric discriminant analysis performed with a support vector machine [9]. A rectangle boundary of the face is produced and then segmented using anthropometric rules into three regions of interest that are considered candidate regions for the left eye/eyebrow, right eye/eyebrow and the mouth [9]. Feature extraction is performed on these candidate regions instead of the entire image for improved accuracy and performance.

The head in the image is rotated to an upright position after estimating the head roll rotation based on the position of the eyes relative to one another [9]. In order to calculate the the angle between the horizontal plane and the eye centers, the eyes and eye centers are identified using a multilayer perceptron network applied to both eye candidate regions independently to produce a template for the shape of the eyes. In subsequent frames, the template is matched to the image to identify the eye using sum of absolute differences [9]. After rotating the head to an upright position, the segmentation of the face into candidate regions for feature extraction is performed again.

The feature extraction stage from [9] utilized a series of masks to extract boundaries of the eyes, eyebrows, and mouth. The nose was also detected but used mainly for geometric reference of other features (the eyes should be above the nose and mouth below). In addition to boundary extraction, center point and corner features were determined.

Eye boundary detection was performed with a combination of masks based on luminance and color information and an edge-based mask [9]. The first mask described by [9] is the luminance and color information mask. This mask attempts to identify the eye boundaries based on the iris center and relies on the assumption that the eyelids usually appear darker than the skin and are typically adjacent to the iris [9]. Each of the eye-candidate regions identified by the neural network are examined independently to create one of these masks for each eye-candidate region. The region is first dilated and then a luminance threshold t_b^e is found by:

$$t_b^e = \frac{1}{3}(2\langle f_c(L^e, M_{nn}^e)\rangle) + min(L^e)), \tag{4}$$

$$f_c(A,B) = \{c_{ij}\},\ c_{ij} = \begin{cases} a_{ij}, & b_{ij} \neq 0, \\ 0, & b_{ij} = 0, \end{cases} \tag{5}$$

where l^e is the luminance channel of the eye candidate area, $\langle \bullet \rangle$ is the average over the image area, and $min(X)$ is the minimum value of area X [9]. After applying t_b^e to L^e a mask identifying the eyelashes and iris is derived. The iris point is found based on the thickness of the object detected in the mask and is determined as the point where the distance obtained from a distance transform for the object is maximized [9].

After all intended features are extracted, they can be used for classification. A rule based classification system is described by [9] which is discussed in IV-A.

B. Deep Attentive Multi-path Convolutional Neural Network

A novel system for facial expression recognition, Deep Attentive Multi-path Convolutional Neural Network (DAM-CNN), is proposed by [10]. The system, consists of three modules: feature extraction, salient region determination and classification. Feature extraction is done with the VGG-Face network as proposed by [11]. Feature refinement and salient region determination is performed with the proposed attention-based Salient Expressional Region Descriptor (SERD) [10]. Classification is performed with the proposed Multi-Path Variation-Suppressing Network (MPVS-Net) [10]. Within the context of feature extraction the VGG-Face and the SERD models are the most relevant, the MVS-Net will be discussed further in IV-B.

Initial feature extraction in [10] is performed with the VGG-Face model described in [11]. The VGG-Face model is a Convolutional Neural Network (CNN) and a pre-trained VGG-Face network that was fine-tuned for feature extraction by mapping the last pooling layer of the network to features of size 7 x 7 x 512 [10] for use in the SERD stage.

Based on concepts of Facial Action Coding System (FACS) around facial action units (AU's) described by [12], [10] determined that different facial regions provide unequally weighted information about facial expression. For this reason, the SERD model is proposed to modify the VGG-Face model by identifying features produced by the VGG-Face model that provide the greatest amount of information about the facial expression. The SERD consists of a network that produces an attention mask that quantifies the importance of each position in the feature maps and weights the features accordingly for input into the classification module [10]. By joining together the VGG-Face model and the SERD the researchers produce a model they call VGG-SERD which produces extracts features from images weighted according to their importance in expression recognition as determined by AU's as described by the FACS.

C. Precise Eye Center Detection

Feature extraction has also been done based on facial features and their known positions relative to one another. When this approach is taken, often the eyes are the first feature detected and other features (nose, mouth, eyebrows... etc.) are determined based on their known relative position to the eyes. A method for identifying the precise location of eye centers using color information is described in [13].

After face regions are detected and segmented into regions of interest using methods described by [5], the regions of interest for each of the eyes are evaluated by the system proposed in [13] to determine the precise center of the eye. The first stage of analysis uses color information to build the eye map. The eye map is built off of evaluation of the $YCbCr$ color space of the image. A map $EyeMapc$ of the eye region is built using the Cb and Cr values of the $YCbCr$ color space where:

$$EyeMapC = \frac{1}{3}\{(Cb)^2 + (\overline{Cr})^2 + (\frac{Cb}{Cr})\} \tag{6}$$

with Cb and Cr normalized over the range [0, 1] and $\overline{Cr} = 1 - Cr$ [13]. Large values in the resulting map indicate a large difference in the color of pixels which is indicative of a boundary between skin and eye. After obtaining the $EyeMapC$ information, the map is divided by the Y information from the $YCbCr$ space to emphasize the iris area which is typically the brightest area in $EyeMapC$ [13]. To further emphasize the iris area, morphological operations with a circular structuring element are applied to derive $EyeMapI$ [13]. As [13] state that iris size does not significantly vary across humans, the expected proportion of the iris to the face is used to estimate the size of the iris in the image:

$$EyeMapI = \frac{EyeMapC \oplus B1}{(Y \ominus B2) + \delta} \tag{7}$$

where $\oplus$ detonates gray-scale dilation and $\ominus$ detonates gray-scale erosion and:

$$B1 = \frac{IrisRad}{2}, \tag{8}$$

$$B2 = \frac{B1}{2}, \tag{9}$$

$$IrisRad = \frac{EyeRegionWidth}{10}, \tag{10}$$

$$\delta = mean(Y \ominus B2) \tag{11}$$

[13]

Based on the fact that the eye exhibits properties of symmetry, a radial symmetry transform is used to enhance the eye center location in both the luminance image as well as the *eye map* by maximizing values at the eye center [13]. The maximum pixel value (x_C, y_C) of the sum of the radial transformation of the luminance image as well as the radial transformation of the *eye map* (C) designates the estimated eye center location.

Performance of the system proposed by [13] was evaluated using four publicly available database: the GTAV face database, the MUCT face database, the FERET face database and the BUHP database. The normalized error between estimated eye center and manually labeled eye center is used to evaluate the performance of the system on test images. Algorithm accuracy was expressed as the number of normalized errors that fell below an assigned threshold (.25, .1, and .05), divided by the total number of images in the database [13]. Results showed accurate estimation of eye center in cases when the eyes were not completely closed and when there was not extreme uneven illumination of the subject [13]. For comparison, the algorithm proposed by [13] was evaluated against results from other methods that had either been reported with the same dataset or where an implementation of the algorithm was publicly available. The proposed algorithm outperformed all algorithms compared to in almost every case showing that the proposed method provides a significant improvement over existing techniques [13].

D. Face Recognition Using Spatiotemporal ICA and Euclidean Features

A method combining Spatiotemporal Independent Component Analysis (ICA) and Euclidean Features for face recognition is proposed by [14]. In their work, [14] extract the distance between the centers of the eyes as well as the distance from the center of the left eye to the center of the mouth in order to create a ratio to scale the image. In order to determine these points, the eyes and mouth had to be identified in the image. For identification of the eyes a multistep process was used. First candidate regions were identified by implementing a pattern matching algorithm that relied on the light to dark to light contrast of the pupils and eyelashes [14]. After identification, candidate regions were analyzed for conformance with known eye shape and size and the candidate regions most closely matching this were selected. Finally the center point of each extracted eye was determined by finding local maxima on approximately the same horizontal level in the two eyes [14]. The eye regions were then finalized by selecting all pixels within a pre-determined distance from the detected eye centers [14].

The mouth was detected using edge detection algorithms and known characteristics of mouths. A kirsch edge detection filter was used to identify edges in the image and because mouths are one of the most contrasting features of the face, the mouth was one of the prominent objects highlighted by the edge detector [14]. Again, after initial edge detection, the candidate regions were analyzed for conformance to known size and shape characteristics of mouths and the best match was selected as the detected mouth [14].

After identifying the eyes and mouth, the measurements discussed above were calculated and used to scale the image in preparation for Spatial and Spatiotemporal ICA. Spatial ICA breaks down the input image into components by multiplying the original image pixel values as a matrix with varying equal sized weighting matrices and treating the results as features for analysis. In addition to Spatial ICA, Spatiotemporal ICA was also used by [14], however because Spatiotemporal ICA involves analysis of a sequence of images it is less relevant to the current work analyzing single images. Finally, after using ICA for feature extraction, face recognition was performed using K-NN and SVM classifiers.

E. Eye Detection Using CNN and SVM

An eye detection method based on convolutional neural networks (CNN) and support vector machines (SVM) is presented by [15]. Classifiers are applied to the image to detect and refine the eye positions in the image. The classification techniques described by [15] start with more coarse filtration methods and progressively utilize finer classification to determine the eye location [15]. The methods proposed by [15] were shown perform better than previously proposed methods, especially in situations where other detection methods have shown to have shortcomings [15].

The eye detection technique proposed by [15] is a three layer system. The first layer is a eye variance filter (EVF). The EVF relies on the observation that the grey level intensity in the eye region is more drastic than in other areas of the face [15]. The EVF is constructed based on the intensity of 30 known eye images in the database. By averaging the variation in grey intensity of the 30 training images, a variance image is constructed that represents the average variance values for the 30 known eye images. After the variance image is calculated another 30 known eye images as well as 30 non-eye images are used from the database to calculate the EVF threshold. Each of the 60 images are analyzed for grey intensity variance and their individual variances are correlated with the variance image. A variance threshold is selected so that non eye-images would be discarded and eye images would not be discarded. After this training phase, the EVF is ready to scan new images and filter out non-eye images before moving to the feature extraction phase.

The feature extraction phase of the eye detection system proposed by [15] is performed by a CNN. Eye images identified by the EVF are used as the input to the CNN, because the images are 32 x 32 the input layer of the CNN has 1024 inputs. Each of these inputs then map to a smaller set of nodes as they move through the networks hidden layers. What is typically considered the output layer of a CNN is replaced in [15] by a SVM, however for training the CNN is treated independently from the SVM. The CNN is trained in a traditional method where the final layer of the CNN represents the classification and training is done until classification results converge for

the network [15]. Once this traditional training is complete the output layer of the network, which consisted of just two neurons, is replaced by a SVM [15]. The inputs to the SVM are the outputs of the hidden layer of the CNN just before the output layer, in the case of [15] layer 5. At this point the SVM is trained using the output features from the CNN. The intent of this approach is to utilize each classifier for processes with which it handles best. The CNN is typically used for multi-class classification problems and the SVM is used mainly for solving two-class problems. By introducing the SVM as the last layer of the CNN, the multi-class information derived by the CNN can be used by the SVM to make the final two-class determination.

The system proposed by [15] was trained using images from multiple datasets including: the extended M2VTS dabatbase (XM2VTS), Psychological Image Collection at Stirling (PICS) face database, Japanese Female Facial Expression (JAFFE) face database, Milborrow/University of Cape Town (MUCT) face database, California Institute of Technology (Caltech) face database, Self-face database, and face images from Internet websites [15]. The system was then tested on images from separate datasets including the BioID, IMM, FERET, and ORL face databases [15].

F. Feature Extraction Using Linear Regression

A method for feature extraction using gradient and Laplacian operators along with linear regression is proposed and analyzed in [16]. A fundamental operation to the feature extraction process outlined by [16] is edge detection. For this reason [16] examined the performance of various edge detection operators: Sobel filter (SF), Prewitt filter (PF), Roberts filter (RF), zero cross filter (ZF), Laplacian of Gaussian filter (LG), and Canny filter(CF) [16]. The various edge detection filters were combined with the slope of the linear regression (SLP) analysis and artificial neural networks for face recognition [16]. While face recognition is a separate field from expression detection, there are many overlapping concepts between the fields. For this reason, the feature extraction methodology described in [16] is of greatest relevance to the current work and will be discussed.

The first step in the face recognition algorithm proposed by [16] is pre-processing and edge detection. The second step is image segmentation and feature extraction, and the third step is classification. During pre-processing, the face image is first scanned by the selected edge detection algorithm and then segmented into equal sized blocks and treated as a feature vector. Two groups of edge detectors are used: gradient based operators are SF, PF, and RF and Laplacian based operators are LF, ZF, and CF [16]. After feature vectors have been created, the proposed SLP extraction method from [16] is used to select features that are impacted least by outliers or noise. The SLP method is a unique method proposed by [16], for refining features found by edge detection. First in SLP, images are converted to binary using an edge detection filter then each image is divided into segments. Then for each segment, the pixels with values of 1 are considered points on the (x, y) plane and regression is used to determine the equation for the line of best fit for the points. The slope of the line is then added to a matrix of the slopes of the lines from each of the other segments and the resulting matrix is used as the feature input to the classification algorithm [16].

Combination edge detector and feature extraction methods were tested by [16] on the BIO OS database from [17]. The experimental results of [16] show that the number of segments is one of the most important parameters and that 100 total segments (10 horizontal and 10 vertical) is the optimal number of segments [16]. Additionally, the optimal number of neurons in the hidden layers of the neural network is 30 [16]. [16] also found that the thresholds for the edge detecting algorithms were important to the classification process, however the optimal thresholds for each of the edge detection filters were determined in a separate experiment. Finally, the best performing system is the LoG, SLP and ANN combination with a weighted mean score of over 95% [16].

G. Gabor Filters

Feature extraction using features from the frequency domain as well as the spatial domain is examined in [6].

2D Gabor filters can provide information about an image in both the spatial and frequency domains depending on the filter used. Gabor filters have eight degrees of freedom including $(x,\ y)$ coordinates specifying the location of the filter in the spatial domain and (u, v) coordinates in the frequency domain, these are the independent variables that can be modified to tune the Gabor filter [18]. [6] performed facial feature extraction using Gabor filters focused on the spatial domain and log Gabor filters in the frequency domain. Log Gabor filters are used to overcome the limitations of Gabor filters in obtaining broad spectral information with maximal spatial localization [6]. Based on the results reported by [6], log Gabor filters were shown to have better accuracy detecting facial expressions than Gabor filters. The features extracted using log Gabor filters in [6] are intended to capture information about shape, motion, color, texture and spatial configuration of the face aligned at particular orientations. For feature extraction, [6] use a bank of Gabor filters with five frequencies and eight orientations. Filters are expressed in [6] and are convoluted with the original image to create Gabor features. The results of facial expression classification using, Gabor filters, log Gabor filters, local binary pattern, HLAC features, and HLACLF features are compared in [6].

IV. Facial Expression Classification

The classification portion of a facial expression recognition system at it's most basic level will take features as inputs and based on those inputs will make a determination of the expression displayed in the image. The input features will be the output of the feature extraction and selection stage and the determined class will be one of a pre-provided set of possibilities. In general, this type of classification has been performed either with a rule based scheme or using supervised machine learning classification algorithms. In the following section, rule-based and machine learning based classification methods will be examined.

A. Rule Based Classification

1) Probabilistic Model: As part of their system for robust facial expression recognition [9] implement a rule based classification model. Instead of using the six archetypal expressions described by [19] a quadrants of emotions wheel is used to classify emotions. A rule based probabilistic measure is used to determine which emotion is described in the image. The rules used by [9] consist of ranges corresponding to high, medium, and low activation of the facial animation parameters detected by the extraction phase. Because it is possible that not all possible defined features could be extracted from an image, [9] designed rules to ensure that only features that are known with a high degree of confidence have a significant impact on classification and those features that are not extracted or are extracted with lower degrees of confidence have less of an impact on the classification decision [9]. Based on the calculations of the rules outlined in [9], the emotion is determined based on the (x, y) point on the plot of activation emotion space as described in [9]. The x and y coordinates are determined based on the rules outlined in [9].

B. Machine Learning Based Classification

1) Naive Bayesian Classifier: A Naive Bayesian (NB) classification algorithm is used by [6] to classify images based on the output of each of the afore mentioned feature extraction methods. Naive Bayesian classifiers are Bayesian networks where all attributes are assumed to be independent [20]. Despite the fact that this assumption, called conditional independence, is rarely true in real world situations (there is typically some interdependence or correlation among features), Naive Bayesian classifiers have been shown to perform well compared to other classification methods [20].

Bayesian classifiers make a classification determination based on the results of an evaluation function that examines the probability of each class given values fore each feature and selects the class with the highest probability. The algorithm stores the conditional probability, sometimes called the weight, of each feature given the class and for each iteration of training the conditional probabilities of each class are updated based on the result. It is an important characteristic of the Bayesian classifiers that the algorithm assumes that the features are statistically independent [21]. Because of the fact that in most real world scenarios features are not statistically independent, initial work assumed that this classification method might produce poor results [20] [21]. However, as shown in [20] that is not the case, rather Naive Bayesian classifiers have been shown to produce results on par with more sophisticated classification algorithms [20].

The Naive Bayesian classifier is a probabilistic based model. First the probability of each class c is calculated based on the given feature set $X = (x_1, x_2, ..., x_n)$ as:

$$p(c|X) = \frac{p(X|c)\, p(c)}{p(X)} \tag{12}$$

where

$$p(c) = \frac{Number\ of\ Samples\ in\ Class\ c}{Total\ Number\ of\ Samples} \tag{13}$$

[6] and $p(X)$ is equal for all classes and can therefore be ignored [22].

The evaluation function $p(X|c)$ accounts for the probability of each feature set X given each class c. This is often called the conditional probability. The function for the conditional probability is given as:

$$p(X|c) = \prod_{i=1}^{n} p(x_i|c) \tag{14}$$

and the class C is determined by the maximum value of $p(c|X)$ after evaluating all classes c where the total number of classes k is shown as:

$$C = max\{p(c_1|X), ..., p(c_k|X)\} \tag{15}$$

2) Neural Network Classifier: Neural networks are a popular machine learning algorithm used for multi-class classification problem. Neural networks have been used been used in Face Detection, Feature Extraction and Classification stages of other facial expression recognition systems with success. [3] used neural networks for face detection, [10], [9], and [15] used neural networks for feature extraction and [16] and [11] used neural networks for face recognition.

Neural networks, modeled after the function of neurons in the human brain, are networks of connected nodes called perceptrons [23]. There is an input layer of nodes where each node corresponds to one feature of input, followed by one or more "hidden" layers of nodes that connect prior layers to the next with weighted connections, the final layer is the output layer which encodes the classification [23].

Given node i in layer h connected to node j in layer $h+1$, the equation for the input (x) to j from i is given as the product of the output of i and the weight of the connection from i to j:

$$x_j^{h+1} \ = \ y_i^h w_{ij}^h \tag{16}$$

or

$$x_j \ = \ y_i w_{ij} \tag{17}$$

where y_i^h is the output of neuron i and w_{ij}^h is the weight of the connection between i and j. y_i^h is defined for nodes in the input layer I as:

$$y_j^I = x_j^I \tag{18}$$

and for nodes in all other layers h as:

$$y_j^h = f(x_j^h) \tag{19}$$

where $f(x_j^h)$ is the activation function. The most popular activation functions are the logistic sigmoid function, the hyberbolic tangent function (tanh) and the rectified linear unit (ReLU) function [24]. Currently the most popular of these activation functions is the ReLU function [24].

The logistic sigmoid function is given as:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{20}$$

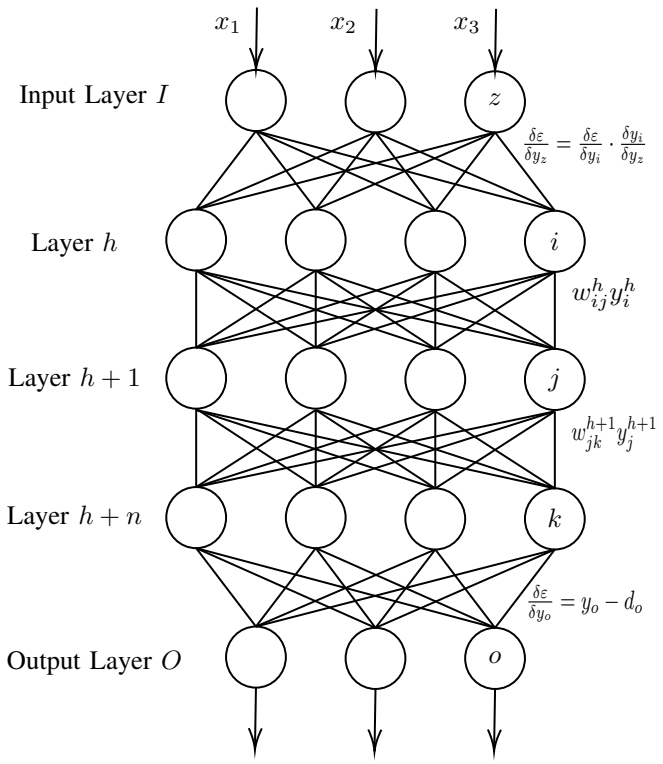

Fig. 1. A Neural Network with $n+1$ hidden layers, neuron outputs y and connection weights w.

the hyberbolic tangent is given as:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{21}$$

and the ReLu function is given as:

$$f(x) = max(0, x) \tag{22}$$

Subsequentially the input to node j in layer $h+1$ from n nodes in layer h that connect to j is given by:

$$x_j^{h+1} = \sum_{k=1}^{n} y_k^h w_{kj}^h - \theta_j^{h+1} \tag{23}$$

where θ_j^{h+1} is the threshold of neuron j in layer $h+1$.

While data flows from the input layer of the network to the output layer, learning flows from the output layer to the input. During training, the weights of each connection are adjusted automatically to minimize the error between the expected output of any node and its actual output. This can be thought of as a relationship that indicates the magnitude of the impact of a change in weight on the error for that node [24]. The weights for each node are adjusted based on the derivative of the cumulative error at each of the nodes in layers closer to the the output side of the network. Therefore the flow of the weight adjustments is in the opposite direction as the flow of data through the network, for this reason, this flow is called backpropogation [24].

Backpropogation begins at the output layer of the network where error for each node is calculated based on a selected cost function. Two commonly used cost functions are logistic regression and quadratic.

The quadratic cost function is the more simple of the two and has been used in some of the earliest neural networks [25]. Where error ε at one node o is given by:

$$\varepsilon_o = \frac{1}{2}(y_o - d_o)^2 \tag{24}$$

And where error of the entire network can be expressed as:

$$\varepsilon = \frac{1}{2} \sum_{h=1}^{m} \sum_{j=1}^{N_L} (y_j - d_j)^2 \tag{25}$$

where m is the total number of layers in the network and N_L is equal to the number of nodes in the current layer h.

Error at a node can be minimized using gradient decent by differentiating equation 24 [25], this gives:

$$\frac{\delta\varepsilon}{\delta y_o} = y_o - d_o \tag{26}$$

Then based on the chain rule [26], the value of $\frac{\delta\varepsilon}{\delta y_k}$ for any node k connected to node o, where o is one layer closer to the output layer than k, can be given as:

$$\frac{\delta\varepsilon}{\delta y_k} = \frac{\delta\varepsilon}{\delta y_o} \cdot \frac{\delta y_o}{\delta y_k} \tag{27}$$

Because we cannot directly modify the output of a node k based on error for the output of node o which is in the layer above it, based on the relationship from equation 16, the input to node o can be indirectly modified by adjusting the weight of the connection between k and o.

Therefore we can adjust the input to o from k by adjusting the value of w_{ko} instead of y_k. This gives us equation 27 expressed in terms of the weight of the connection between nodes k and o:

$$\frac{\delta\varepsilon}{\delta w_{ko}} = \frac{\delta\sigma}{\delta y_o} \cdot \frac{\delta y_o}{\delta w_{ko}} \tag{28}$$

The simplest modification of weight w expressed as Δw is by an amount proportional to

$$\Delta w_{ij}^h = -\varepsilon \frac{\delta\varepsilon}{\delta w_{ij}} + \alpha \Delta w_{jk}^{h+1} \tag{29}$$

where α is an exponential decay factor between 0 and 1 that designates the impact of Δw_{jk}^{h+1} on the Δw_{ij}^h [25].

A second option for the cost function is logistic regression. Logistic regression is used to examine the relationship between a outcome and a set of dependent variables [27].

While the entire network solves multi-class problems, each individual node has a binary output (1 or 0). The mean expected value of a node's output y_j^h based on a given class c can be expressed as a linear function $E(y_j^h|c)$, also called the conditional mean [27]:

$$E(y_j^h|c) = \beta_0 + \beta_1 c \tag{30}$$

Further, due to the dichotomous nature of the node output (the values of y_j^h are either 0 or 1) the values for $E(y_j^h|c)$ all fall between 0 and 1 [i.e., $0 \leq E(y_j^h|c) \leq 1$] [27]. Based

on this, a regression based on the logistic distribution can be represented as:

$$E(y_j^h|c) = \frac{e^{\beta_0+\beta_1 c}}{1+e^{\beta_0+\beta_1 c}} \tag{31}$$

this model can be transformed using the logit transformation to:

$$g(x) = \log \frac{E(y_j^h|c)}{1-E(y_j^h|c)} = \beta_0 + \beta_1 x \tag{32}$$

the value of y_j^h can now be expressed based on the value of $E(y_j^h|c)$ plus some error ε, thus:

$$y_j^h = E(y_j^h|c) + \varepsilon \tag{33}$$

as this function is a linear equation it can be differentiated to understand the sensitivity of the change in the output value (y_j^h in this case) to change in the input value (c).

As mentioned above, many prior works have utilized neural network classification models during one of the stages of the facial expression recognition process. Neural networks have been used for a variety of classification tasks and have performed well when trained properly. The quality of the classifier is strongly tied to the quality of the training data. Often, the ability to train a model to high degrees of accuracy requires a significant amount of time and data. Based on the above information, a neural network model, with appropriate parameters can be used for any classification task within a facial expression recognition system.

V. Analysis

The findings in [2] show that the binary nature of the decision, face or non-face, lends itself well to the SVM classifier for face detection. Furthermore, the technique of training the SVM on negatively identified images from the initial filtering phase as well as on initially incorrectly classified images from subsequent testing phases could prove useful in any system implementing a SVM for face detection. Finally, a comparison in terms of performance between [3] and the proposed system in [2] could mitigate the superiority of System 11 proposed by [3].

The geometric models for face detection described in [4] make logical sense using known characteristics of the face to find faces in images. However, this methods seems susceptible to differences in light and camera quality and also might struggle with subjects wearing hats, glasses, or other clothing that obstructs the face. The system proposed by [4] is simple and straightforward. It can be utilized for both color and grayscale images and has no limitation for image size, resolution or complexity. Accuracy information has not been provided but could be tested. The speed reported may be prohibitively slow but advances in software and hardware since the time of publication (2000) may improve performance. Implementing an automatic thresholding algorithm could improve the performance of the system.

The methodology proposed by [5] is used by many other papers and studies for face detection. The system provides improvements over prior work in terms of speed and accuracy. The methods proposed could be used effectively for face detection and feature extraction in a facial expression recognition system.

The techniques proposed by [3] demonstrate the ability of neural networks to be used for a variety of tasks with a face detection system. This concept can be applied to facial expression where neural networks could be used for multiple parts of the system. The added handling of face position in [3] provides a technique that can be used to make a system for identifying faces in an image more robust. Further, the multiple layers of neural networks with one arbitrator network at the end provides a way to reduce problems associated with over fitting during model training as well as variance in characteristics of input.

The techniques proposed by [9] for feature extraction using the eyes, eye brows, and mouth to understand the expression are clearly related to how a human would describe the process of interpreting a facial expression. This provides the benefit of clarity to the system. The methods for eye detection and iris detection outlined in [9] could be used in other systems for more accurate detection of facial features.

Another system attempting to find eye center locations proposed by [13] seems to be effective and efficient at determining eye center locations. The identification of eye center location can be leveraged by multiple other systems that attempt to extract facial features from images as other features can be located relative to the position of the eyes. However, the method proposed in [13] is reliant on color information with is a major limiting factor and may also cause more inefficient processing when analyzing three times as much data as the same image in greyscale would produce.

The methods described in [14] again mimic how a human might process facial expression recognition. Based on the results of [14], the SVM was found to perform slightly better than K-NN for classification. While the extracted Euclidean features were used primarily for scaling in their initial work, [14] propose that a combination of ICA features and Euclidean distance features could improve accuracy. This method of scaling could be used in combination with other systems and might provide performance improvements over other more complex scaling methods.

A major limitation of the study done by [15] was the fact that the face detection methods proposed was not able to achieve 100% accuracy and therefore the researchers manually extracted face regions on those images incorrectly labeled by the face detection phase. Further, the lack of established criteria for evaluating the accuracy of eye region detection posed a challenged so a correct identification was defined as an upper and lower eyelid and two corners fall within the eye region [15]. This measure may not provide proper comparison to past research but will suffice for the comparison between CNN and the CNN+SVM system proposed by [15]. The results of [15] showed that the CNN+SVM method outperformed the CNN for each of the four datasets [15]. The CNN+SVM methods were also shown to have greater accuracy than methods proposed in prior work [15]. The CNN+SVM method showed the greatest improvement over

the SVM methods on images that included subjects wearing glasses. In all, [15] showed that a CNN+SVM system is more effective than a standalone CNN at eye region detection in facial images.

The linear regression analysis technique proposed by [16] is efficient and shown to achieve accurate results. However, with any linear regression, only two features can be assessed for a relationship, this increases the possibility that a relationship between features that were not involved in the regression is missed. Despite this, [16] demonstrated reasonably accurate results using their SLP method.

The multiple methods for feature extraction were examined and compared by [6]. Gabor filters and log Gabor filters were applied to the entire image for collection of localized frequency information. LBP operators were applied to sub-regions of the image for extracting texture information. Mainly frequency information was extracted with Gabor filters, Gabor log filters, HLAC and HLACLS. [6] found log Gabor filters to produce more accurate results than Gabor filters and additionally that Gabor filters are time consuming to apply to images. Overall [6] found the optimal feature extraction method utilized HLACLF for extraction.

While a rule based system like the one demonstrated in [9] is simple and efficient, a system built around a machine learning classifier can be built to more easily process a larger set of data points. Further, machine learning based classification has shown to perform well at the type of multi-class classification problem that is determining facial expression in images. The activation-emotion space is a method of classification demonstrated in [9] more accurately reflects the complexity of emotions that humans express. For these reasons, a machine learning model that determines the (x, y) points on the activation-emotion space plot could more accurately reflect the true emotion being expressed in the image.

VI. Conclusion

The prior sections have compiled many of the most common techniques used for facial expression recognition, face detection and facial recognition. Each of these three tasks are similar in that they involve the processing of image data, extracting features related to faces and making a determination based on the data. A new system that implements the most effective modules of these systems could be reasonably assumed to achieve more accurate results that prior systems. Based on the information provided in this paper, a new system will be proposed, implemented and tested. Based on the techniques examined in this paper, a system using a support vector machine for face detection, edge and object detection for feature extraction and a neural network for classification will be proposed. Support vector machines have been shown to be optimal in the case of a binary decision therefore they may have an advantage when making the determination if there is a face in the image or not. Feature extraction methods that involve multiple layers of extraction and selection have been shown to perform best, therefore a multi-layered extraction model will be proposed. Layers of the model will include face boundary detection detection as proposed by [5] and gradient operators similar to those used in [14] and [16]. Neural networks have been used in the past with positive results for multi-class supervised classification and based on the prior analysis are preferable to a rule based classification scheme. Therefore, a Neural Network will be proposed for the classification module. In all, this work has analyzed methods for face detection, face recognition and facial expression recognition. A system for facial expression recognition is decomposed into three sub-modules: face detection, feature extraction and classification and techniques for each from prior work are examined. The foundation has been provided for subsequent work implementing and comparing combinations of the analyzed techniques.

REFERENCES

[1] J. L. Cecilione, L. M. Rappaport, B. Verhulst, D. M. Carney, R. Blair, M. A. Brotman, E. Leibenluft, D. S. Pine, R. Roberson-Nay, and J. M. Hettema, "Test–retest reliability of the facial expression labeling task.," *Psychological assessment*, vol. 29, no. 12, p. 1537, 2017.

[2] H. Ai, L. Liang, and G. Xu, "Face detection based on template matching and support vector machines," in *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, vol. 1, pp. 1006–1009, IEEE, 2001.

[3] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 23–38, Jan 1998.

[4] C. Lin and K.-C. Fan, "Human face detection using geometric triangle relationship," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 2, pp. 941–944, IEEE, 2000.

[5] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[6] S. M. Lajevardi and Z. M. Hussain, "Automatic facial expression recognition: feature extraction and selection," *Signal, Image and video processing*, vol. 6, no. 1, pp. 159–169, 2012.

[7] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature extraction: foundations and applications*, vol. 207. Springer, 2008.

[8] Y. Tian, T. Kanade, and J. F. Cohn, "Facial expression recognition," in *Handbook of face recognition*, pp. 487–519, Springer, 2011.

[9] S. Ioannou, G. Caridakis, K. Karpouzis, and S. Kollias, "Robust feature detection for facial expression recognition," *Journal on image and video processing*, vol. 2007, no. 2, pp. 5–5, 2007.

[10] S. Xie, H. Hu, and Y. Wu, "Deep multi-path convolutional neural network joint with salient region attention for facial expression recognition," *Pattern Recognition*, 2019.

[11] O. M. Parkhi, A. Vedaldi, A. Zisserman, *et al.*, "Deep face recognition.," in *bmvc*, vol. 1, p. 6, 2015.

[12] E. Friesen and P. Ekman, "Facial action coding system: a technique for the measurement of facial movement," *Palo Alto*, vol. 3, 1978.

[13] E. Skodras and N. Fakotakis, "Precise localization of eye centers in low resolution color images," *Image and Vision Computing*, vol. 36, pp. 51–60, 2015.

[14] J. Lei, T. Lay, C. Weiland, and C. Lu, "A combination of spatiotemporal ica and euclidean features for face recognition," in *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pp. 395–403, Springer, 2006.

[15] M. Yu, X. Tang, Y. Lin, D. Schmidt, X. Wang, Y. Guo, and B. Liang, "An eye detection method based on convolutional neural networks and support vector machines," *Intelligent Data Analysis*, vol. 22, no. 2, pp. 345–362, 2018.

[16] A. Alazzawi, O. N. Ucan, and O. Bayat, "Performance of face recognition system using gradient laplacian operators and new features extraction method based on linear regression slope," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[17] "Bioid-face-database,"

[18] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *JOSA A*, vol. 2, no. 7, pp. 1160–1169, 1985.

[19] P. Ekman, W. V. Friesen, M. O'sullivan, A. Chan, I. Diacoyanni-Tarlatzis, K. Heider, R. Krause, W. A. LeCompte, T. Pitcairn, P. E. Ricci-Bitti, *et al.*, "Universals and cultural differences in the judgments of facial expressions of emotion.," *Journal of personality and social psychology*, vol. 53, no. 4, p. 712, 1987.

[20] H. Zhang, "The optimality of naive bayes," *AA*, vol. 1, no. 2, p. 3, 2004.

[21] P. Langley, W. Iba, K. Thompson, *et al.*, "An analysis of bayesian classifiers," in *Aaai*, vol. 90, pp. 223–228, 1992.

[22] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, pp. 41–46, 2001.

[23] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Transactions on neural networks*, vol. 3, no. 5, pp. 683–697, 1992.

[24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[25] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *et al.*, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

[26] T. Abdeljawad, "On conformable fractional calculus," *Journal of computational and Applied Mathematics*, vol. 279, pp. 57–66, 2015.

[27] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*, vol. 398. John Wiley & Sons, 2013.