A System for Automated Facial Expression Recognition

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

***Abstract*—Training a computer to understand images the way a human can is a long standing interest of academia and society. Face detection and facial recognition are two of the most widely studied techniques. Facial expression recognition, the next logical step in facial image processing, is less widely studied. The current paper, provides a threefold contribution to the current body of research around facial expression recognition. First, a detailed review of various techniques related to each stage of detecting a face, extracting key features and classifying the facial expression. Second, a demonstration that eye regions convey a significant amount of information related to the facial expression. Finally, that a more complex convolutional neural network, similar to ones used previously for face detection by [11], preforms better than a simpler neural network when classifying facial expressions. Multiple tests on a constrained (JAFFE) dataset are performed with varying system configurations to demonstrate the effectiveness of the system and the impact of extracted features.**

CONTENTS

I. INTRODUCTION

The ability to recognize emotion in a facial expression 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 expand prior work that reviewed techniques for facial expression recognition by applying those techniques in a novel way. 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 facial expression. The following paper will be organized into six sections: a review of prior related work, system design, experimental design, results reporting, a discussion of the results, and a conclusion.

In the review 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-A. 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 II-B, 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 facial expression in the image. In II-C, methods for classification are examined.

The result of this work will be a novel system for facial expression recognition and a comparison of various implementations of the system.

II. Prior Work

The following is a brief review of some relevant prior work in the field of automatic facial expression recognition, face detection and facial recognition. Three distinct modules that make up this type of system are examined: face detection, feature extraction, and classification.

A. 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.

1) Template Matching: 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].

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].

2) Geometric Rules: A method for detecting faces in images using geometric rules is proposed by [4]. In [4] known 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.

3) Haar Cascade: 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. [5] use a Haar feature-based cascade classifier to accurately and quickly identify faces in images.

In their paper [5] provide three main contributions: the concept of the integral image for faster calculation of Haar features, and the use of an AdaBoosted algorithm to select the subset of features most import to detection and, the concept of using a cascade of increasingly complex classifiers for more efficient detection. 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 Haar-like 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 Haar-like 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 reduce the number of features used for classification, and thereby the computation and time needed for classification, a variant of the AdaBoost algorithm used for feature selection is used [5]. The trained algorithm selects from the large number of Harr features, calculated using the integral image, the subset that have the most relevance to classification. By reducing the number of features, computation and classification time is significantly reduced [5].

The classification of the facial boundaries in the image is done in [5] by a cascade of weak classifiers. Each classifier feed into the next and is organized in increasing complexity because the more complex analysis is only needed for regions where a face is more likely to be present [5]. The result is a strong classifier that is the combination of all the weak classifiers [5]. The output of the above describes system is the boundaries in the image of the face region.

4) 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. Preprocessing 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].

B. Feature Extraction

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].

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].

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.

[6] use the JAFFE and Cohn-Kanade databases for experimental analysis. For each database, the subjects in the labeled images displayed one of six universal emotions: anger, disgust, fear, happy, sad or surprise. Classes were each of these emotions as well as a neutral class. Accuracy for each feature extraction method 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 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.

1) Facial Animation Parameters Extraction: Facial expression recognition is performed by [9] using 19 facial feature points 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 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. 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 I-A.

2) Convolutional Neural Networks: [10] use a multi-layered system for facial expression recognition. 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].

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] determine that different facial regions provide unequally weighted information about facial expression. For this reason in their SERD model, [10] 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 extracts features from images weighted according to their importance in expression recognition as determined by AU's as described by the FACS.

3) Facial Features: 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. [13] use techniques described in [5] to first detect face regions in an image and then segmented into regions of interest. To determine the precise eye center, [13] use color, luminescence and, anatomical properties to evaluate the regions of interest for the eye center.

[14] use a multi-step process for eye detection. First, candidate regions are identified by implementing a pattern matching algorithm that relied on the light to dark to light contrast of the pupils and eyelashes [14]. Candidate regions are then analyzed for conformance with known eye shape

and size, and the candidate regions most closely matching these characteristics are selected. Finally, the center point of each extracted eye is determined by finding local maxima on approximately the same horizontal level in the two eyes [14]. The eye regions are then finalized by selecting all pixels within a pre-determined distance from the detected eye centers [14].

[15] use convolutional neural networks (CNN) and support vector machines (SVM) for eye detection. 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 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 second layer, feature extraction, 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. The CNN is trained in a traditional method where the final layer of the CNN represents the classification and training done until classification results converge for the network [15]. Once this traditional training is complete the output layer of the network is replaced by a SVM [15]. The final layer, classification, is the SVM. 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.

4) Gabor Filters: [6] performed facial feature extraction using Gabor filters and log Gabor filters. 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, which are the independent variables that can be modified to tune the Gabor filter [16]. The features extracted using 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]. [17] compare facial expression using geometric features and features extracted with Gabor filters. Using the JAFFE database, [17] found that Gabor filters tended to add useful information to the feature set and thereby produce a greater rate of accurate classification.

C. 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.

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 [18] 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].

2) Naive Bayesian Classifier: A Naive Bayesian (NB) classification algorithm is used by [6] to classify images based on the output of each of the aforementioned feature extraction methods. Naive Bayesian classifiers are Bayesian networks where all attributes are assumed to be independent [19]. 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 [19].

Bayesian classifiers make a classification determination based on the results of an evaluation function that examines the probability of each class based on given values for each feature. The selected class is 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

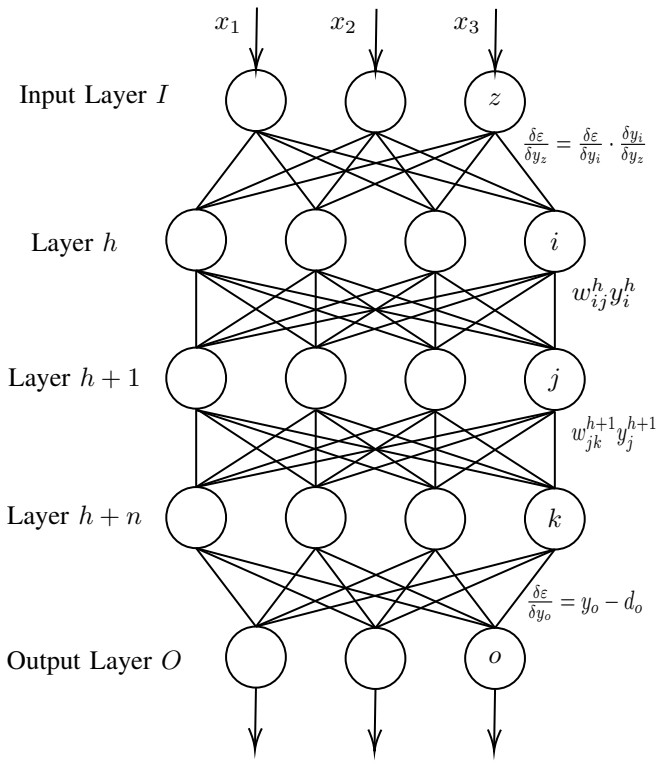


Fig. 1: A Neural Network with $n+1$ hidden layers, neuron outputs y and connection weights w.

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 [20]. 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 [19] [20]. However, as shown in [19] that is not the case, rather Naive Bayesian classifiers have been shown to produce results on par with more sophisticated classification algorithms [19].

3) Neural Network Classifier: Neural networks are a popular machine learning algorithm used for multi-class classification problems. Neural networks have 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 [21] 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 [22]. 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, and the final layer is the output layer which encodes the classification [22].

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:

$$x_j^{h+1} = y_i^h w_{ij}^h \tag{4}$$

or

$$x_j = y_i w_{ij} \tag{5}$$

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:

$$y_j^I = x_j^I \tag{6}$$

and for nodes in all other layers h as:

$$y_j^h = f(x_j^h) \tag{7}$$

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 [23]. Currently the most popular of these activation functions is the ReLU function [23].

The logistic sigmoid function is given as:

$$f(x) = \frac{1}{1+e^{-x}} \tag{8}$$

the hyberbolic tangent is given as:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{9}$$

and the ReLu function is given as:

$$f(x) = max(0, x) \tag{10}$$

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

$$x_j^{h+1} = \sum_{k=1}^{n} y_k^h w_{kj}^h - \theta_j^{h+1} \tag{11}$$

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 it's 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 [23]. 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 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 [23].

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 [24]. Where error ε at one node o is given by:

$$\varepsilon_o = \frac{1}{2}(y_o - d_o)^2 \tag{12}$$

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

$$\varepsilon = \frac{1}{2}\sum_{h=1}^{m}\sum_{j=1}^{N_L}(y_j - d_j)^2 \tag{13}$$

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 12 [24], this gives:

$$\frac{\delta\varepsilon}{\delta y_o} = y_o - d_o \tag{14}$$

Then based on the chain rule [25], 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:

$$\frac{\delta\varepsilon}{\delta y_k} = \frac{\delta\varepsilon}{\delta y_o}\cdot\frac{\delta y_o}{\delta y_k} \tag{15}$$

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 4, 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 15 expressed in terms of the weight of the connection between nodes k and o:

$$\frac{\delta\varepsilon}{\delta w_{ko}} = \frac{\delta\sigma}{\delta y_o}\cdot\frac{\delta y_o}{\delta w_{ko}} \tag{16}$$

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

$$\Delta w_{ij}^h = -\varepsilon\frac{\delta\varepsilon}{\delta w_{ij}} + \alpha\Delta w_{jk}^{h+1} \tag{17}$$

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 [24].

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 [26].

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 [26]:

$$E(y_j^h|c) = \beta_0 + \beta_1 c \tag{18}$$

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., $0 \leq E(y_j^h|c) \leq 1$] [26]. Based on this, a regression based on the logistic distribution can be represented as:

$$E(y_j^h|c) = \frac{e^{\beta_0+\beta_1 c}}{1 + e^{\beta_0+\beta_1 c}} \tag{19}$$

(a) Original (b) Face Image (c) Eye Image

Fig. 2: Images feature extraction progression

this model can be transformed using the logit transformation to:

$$g(x) = \log\frac{E(y_j^h|c)}{1 - E(y_j^h|c)} = \beta_0 + \beta_1 x \tag{20}$$

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:

$$y_j^h = E(y_j^h|c) + \varepsilon \tag{21}$$

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.

III. System Design

Based on the information gathered in II a system for automated facial expression was designed. The system was designed in three distinct modules: face detection, feature extraction and classification. In the face detection stage, similar to the work done by [5], a cascade classifier using Harr like features was used to identify a face in the image and the corresponding facial region boundaries. The classifier used was a pre-trained face detection model from the OpenCV project [27]. As [5] describe in their paper, the model is trained using the integral image to calculate Haar rectangle features for a cascade of classifiers that determine the face region boundaries in the image. This process is described in detail in II-A3. The boundaries determined by this face detection method are a rectangular region that contains the face detected in the image. As mentioned in I, one of the major challenges with image processing systems is removing as much extraneous data as possible in order to reduce time and resources needed for computation. By detecting and extracting just the face region in the image, a significant amount of extraneous information is removed before passing the image to the classification portion of the system.

The quality of any classifier is significantly dependent on the quality of the data used to train it and by limiting the inputs to just the most relevant information, the quality of the classifier is improved. Gabor filters have been shown by [6] to provide feature information that produces more accurate facial expression classification rates. Gabor filters can be used for texture discrimination and to extract texture information from images [28]. As discussed in II-B4, prior research has utilized Gabor filters for expression recognition and face detection with promising results. The Gabor filter is applied as a convolutional mask to the image and is a linear filter that extracts both frequency and orientation information from the image. The 2D-Gabor filter can be represented as the product of a 2D-Gaussian function and a exponential function. [28] designates the expression for the filter given $g(\lambda, \theta, \varphi, \sigma, \gamma)$ as:

$$exp\left(\frac{c(\theta)^2 + \gamma^2 r(\theta)^2}{2\sigma^2}\right) exp\left(j\left(2\pi\frac{c(\theta)}{\lambda} + \varphi\right)\right) \qquad (22)$$

Features extracted using Gabor filters were examined as inputs to the system. For each image, eye and non-eye, Gabor filters were applied and the result was used as input to the classifier. For simplicity, only magnitude information, and the maximum value of each of the 16 filter results was used for input to the classifier.

In the current work, 16 Gabor kernels were applied to each image and the maximum frequency information across all results was used as the input to the classification stage. The Gabor filters were applied by convolving a 31 x 31 kernel of Gabor coefficients with the original image. The values of the inputs to 22 were then determined for each of the filters: The value for σ, which is the standard deviation of the Gaussian function used in the Gabor filter was 4. The value for θ, which is the orientation, was modified for each of the 16 filters as one of 16 evenly spaced values between 0 and π. λ, which is the wavelength of the sinusoidal factor was set to 10. γ, which is the spatial aspect ratio, is set to .5. φ, which is the phase offset, is set to 0. After applying the resulting filters to the image, the maximum value at each point x, y across each filter output was used as the input to the classifier.

In systems III, IV, VII and $VIII$ further feature extraction was done in an attempt to improve the quality of the data fed into the classifier and thereby the overall accuracy of the classifier. As shown in [9] the eyes can be a region of particular interest when attempting to determine facial expression. Additionally, it is commonly thought that eyes play a significant role in human perception of facial expression and emotion. The common phrase "the eyes are the window to the soul" is emblematic of this belief and there has been academic research into this concept as well. [29] found that 16 male subjects spent approximately 40% of their looking time focused upon the eye region of facial photographs when attempting to form impressions about subjects in the photographs. [30] found, in children, that a deficit in fear recognition not only could be partly attributed to a visual neglect of the eye region of others, but that the deficit could be reversed by directing focus to the eye region of others. [31] found that when assessing facial expressions, the subjects gaze almost always initially fixates on the eyes and confirmed the relevance of the eyes and mouth in emotional decoding.

Layer	Layer Type	Shape	Parameters
1	Input	(128, 128, 1)	0
2	Padding	(130, 130, 1)	0
3	Conv2D	(128, 128, 16)	160
4	Conv2D	(128, 128, 16)	2320
5	MaxPooling2D	(63, 63, 16)	0
6	Padding	(65, 65, 16)	0
7	Conv2D	(63, 63, 32)	4640
8	Conv2D	(61, 61, 32)	9248
9	MaxPooling2D	(30, 30, 32)	0
10	Padding	(32, 32, 32)	0
11	Conv2D	(30, 30, 64)	18496
12	Conv2D	(28, 28, 64)	36928
13	MaxPooling2D	(14, 14, 64)	0
14	Flatten	12544	0
15	Dense	4096	51384320
16	Dropout	4096	0
17	Dense	4069	16781312
18	Dropout	4096	0
19	Dense	512	2097664
20	Dropout	512	0
21	Dense (Output)	256	131328

TABLE I: CNN model structure for 128 x 128 input image size. All activation functions are ReLu with the exception of layer 21 which uses softmax.

In order to examine the impact of the eye region on automated facial expression recognition, systems III, IV, VII and $VIII$ were designed to emphasize the eye regions during classification. To do this, eye features were extracted and appended to the face image as shown in Fig.2c. Similar to the face detection stage, a pre-trained Haar Cascade classifier was used for detecting eye candidate regions. From the eye candidate regions identified by the Haar cascade classifier, the eyes regions were selected by logic that identified the two candidate regions higher up in the face region and on a similar horizontal plane as one another.

In prior work, [3] used neural networks for face detection, [10], [9], and [15] used neural networks for feature extraction and [21] and [11] used neural networks for face recognition. Of the research that attempts facial expression recognition or emotion detection; [6] use a Naive Bayesian classifier, [17] and [10] use neural networks and [9] use a rule based classification model.

In the present work, two variations of a neural network were used for facial expression classification: a simple neural network model (NN) and a more complex very deep convolutional neural network model (CNN). The simple model, similar to the one shown in Fig.1 consisted of 6 layer of fully connected nodes with ReLu activation functions followed by an output of n fully connected nodes, where n is the number of classes (in this case 7), and a softmax activation function for the final layer. Overall, the simple NN model had $3,849,347$ trainable parameters. The complex model (CNN), based on the model

System	Face Detection	Eyes	Gabor	Classifier
I	Haar Cascade	N/A	N/A	NN
II	Haar Cascade	N/A	Gabor	NN
III	Haar Cascade	Haar Cascade	N/A	NN
IV	Haar Cascade	Haar Cascade	Gabor	NN
V	Haar Cascade	N/A	N/A	CNN
VI	Haar Cascade	N/A	Gabor	CNN
VII	Haar Cascade	Haar Cascade	N/A	CNN
$VIII$	Haar Cascade	Haar Cascade	Gabor	CNN

TABLE II: System Configurations

Expression	Count
Happy	31
Surprise	30
Neutral	30
Fear	32
Sadness	31
Disappointment	29
Anger	30

TABLE III: Facial Expressions In Dataset

proposed and tested by [11] consists of four sections, the first three sections are three layers each; two layers of 2D Convolution followed by a max pooling layer. The final section is composed of 3 fully connected layers with 4096 nodes at the first two and 512 at the third The output layer consists of 256 fully connected nodes and is analyzed with a softmax activation function. After each iteration of training a dropout of .5 is applied to each of the layers to avoid overfitting the training data and to force the system to find a more generalized solution during training. Overall, the complex CNN model had $70,466,416$ trainable parameters. The structure of the CNN is shown in Table I.

Similar to prior work, the six universal emotions as well as a neutral expression identified by [18] are used as the classes. The images in the data set are pre-labeled labeled with these facial expressions based on the emotion that subjects in the dataset were asked to display. The classifier attempts to predict the class label of the image exactly.

Using the components described above and based on the information gathered in II, 8 systems were designed for automatic facial expression recognition. The systems were designed to provide the current work the ability to compare, within the context of facial expression recognition, the impact various feature extraction and selection techniques as well as two classification techniques. System I consisted of one Haar Cascade classifier and one simple NN. System II consisted of one Haar Cascade classifier, Gabor filters and, one simple NN. System III consisted of two Haar Cascade classifiers and one simple NN. System IV consisted of two Haar Cascade classifiers, Gabor filters and, one simple NN. System V consisted of one Haar Cascade classifier and one very deep CNN. System VI consisted of one Haar Cascade classifier, Gabor filters and, one very deep CNN. System VII consisted of two Haar Cascade classifiers and one very deep CNN. System $VIII$ consisted of two Haar Cascade classifiers, Gabor filters and, one very deep CNN. Based on the differences between the accuracy of the systems when tested, conclusions are drawn about feature extraction, feature selection and classification techniques.

All test were done on a machine running Ubuntu 64-bit distribution of Linux, with an Intel® Core™i5-8600K 6 core CPU @ 3.60GHz, a GeForce GTX 1060 6GB GPU and 16GB DDR4 RAM. Image processing was done with the OpenCV library [27], NN and CNN models were created, trained and tested using TensorFlow r2.0 [32].

IV. Experimental Design

Based on prior work three hypothesis were developed: $H1$ that a system with a deep CNN will outperform a similar system using a simpler NN as the classifier. $H2$ that systems with additional expressive facial features extracted will outperform systems that do not extract additional features from the face image. $H3$ using Gabor filters for feature extraction will positively impact the classification rate.

Similar to [6] and [17], the Japanese Female Facial Expression (JAFFE) database [33] was used for training and testing the system. The database consists of 213 images of 7 facial expressions posed by 10 Japanese female models. The distribution of expressions is shown in III. In each image, the subject is posing the image that they are instructed to express. The author of the dataset acknowledges the possibility for some error and that the emotion being expressed may not be a pure emotion but rather a mix of multiple emotions [33].

For each model, training was done using a k-fold cross-validation technique. Systems using eyes as an additional feature used 174 training images, and a validation set size of 32 for $k = 7$ groups of 29 images. Systems without eyes as an additional feature used 180 training images and a validation set of size 33 for $k = 7$ groups of 30 images. The difference between the two was that the eye detector was unable to detect eyes for seven of the images and they were then left out of training and testing for systems using eye features. In each case, 3 images were withheld initially from all training only to be used for testing (and thereby making the entire dataset evenly divisible over k groups). The result is validation groups of sizes $\%14$ and $\%15$, respectively, of the entire dataset. In each iteration of training and testing, the current group of 29 or 30 images are set aside, along with the 3 additional images, for testing while the remaining images are used to train the model. After each round of training for a designated number of epochs, the trained model was tested on the test set of 32 or 33 images. This training and testing process was repeated seven times at the same number of epochs and the results of all seven tests were averaged to produce the final reported accuracy at that number of epochs. In this way, ther reported accuracy for a system trained at n epochs takes into account the model trained on 7 sets of 174 or 180 images and tested on 7 distinct sets of 32 or 33 images. Overall, this gives a training size of 1,218 or 1,260 and a testing size of 224 or 231.

As evaluation metrics, the exact match accuracy as well as the accuracy allowing for the expected class to be in the

System	Accuracy	Top-2 Accuracy
I	73.7%	90.3%
II	70.2%	86.1%
III	78.8%	93.0%
IV	74.8%	89.6%
V	82.1%	92.9%
VI	82.6%	93.6%
VII	83.6%	94.3%
$VIII$	**85.1**%	**95.3**%

TABLE IV: Average accuracy across each testing iteration. Highest accuracy in bold.

top 2 of the classes the network determined most probable, is collected for every set of training and testing data. Both of these metrics were collected on the testing data and used to calculate two test accuracy scores for every iteration of training and testing. Results of the system trained and tested between 50 and 90 epochs are reported in intervals of 5, averages of all the average scores for each system are also reported.

V. Results

Accuracy results at each number of training epochs can be found in Fig.3. Fig.3a reports the exact match average accuracy at each epoch for systems V through $VIII$. Fig.3b reports the exact match average accuracy at each epoch for systems I through IV. Fig.3c reports the top-2 average accuracy at each epoch for systems V through $VIII$. Fig.3b reports the top-2 average accuracy at each epoch for systems I through IV.

Table IV shows the average accuracy and average top-2 accuracy for each system calculated as the average of all calculated testing accuracies of the system trained from 50 to 90 epochs. As mentioned in IV, each system was trained and tested on n number of epochs in intervals of 5 from 50 to 90. At each step, the system was initialized with random weights and completely retrained in the method described in IV. The average of seven instances of training and testing the system at the desired number of epochs was recorded as the accuracy for that number of epochs and is reported in Fig.3. The accuracies in IV are the average of all reported average accuracies from epochs 50 to 90 (the accuracies reported in Fig.3).

VI. Discussion

The highest exact match accuracy was achieved by system $VIII$ at 86.3% and the highest top-2 accuracy was achieved by system $VIII$ at 96.0%. This system utilized the CNN, Gabor filters and eye image to produce the results.

Comparisons can be made between systems by looking at corresponding systems with only one variable change. To understand the impact of the classifier used: system I can be compared to system V, system II to system VI, system III to system VII, and system IV to system $VIII$. In each of these cases, the system utilizing the CNN achieved both a higher accuracy as well as a higher top-2 accuracy. The accuracy was improved by 11.3%, 17.6%, 6.0%, and 13.7% for the respective comparisons for an average improvement of 12.1% when a CNN was used in place of a NN. The top-2 accuracy was improved by 2.8%, 8.7%, 1.3%, and 6.3% for the respective comparisons for an average improvement of 4.7% when a CNN was used in place of a NN. These results indicate that the CNN is an overall better classifier for facial expression recognition than the NN. While the CNN does have more trainable parameters than the NN, both had a relatively large number of parameters compared to prior work in [19], as well as in [3]. While [10] uses a similarly large number of parameters, the VGG like network used in [10] is used for face detection and not for classification as done in the current work.

To understand the impact of the Gabor filter features: system I can be compared to system II, III to IV, V to VI and VII to $VIII$. In these cases, the Gabor filter features had a negative impact on the systems using NN as the classifier and a positive impact on the systems using CNN as the classifier. Gabor features brought accuracy down by 4.7% and 5.0% in the respective comparisons using the NN and up by 0.6% and 1.7% in respective comparisons using CNN. Gabor features brought top-2 accuracy down by 4.6% and 3.6% in the respective comparisons using the NN and up by 0.7% and 1.0% in respective comparisons using CNN. The Gabor filter comparison results show that a more complex, convolutional neural network used for facial expression recognition can be positively impacted by the use of Gabor filters as feature extractors while a simpler neural network is negatively impacted by the use of Gabor filters as feature extractors.

To understand the impact of eye features: system I can be compared to system III, system II to system IV, system V to VII and, system VI to system $VIII$. In each of these cases, the addition of the extracted eye features to the input improved the accuracy and top-2 accuracy. Eye features improved the accuracy of each of the respective comparisons by 6.9%, 6.5%, 1.8% and, 3.0% for an average improvement of 4.5%. Eye features improved the top-2 accuracy of each of the respective comparisons by 2.9%, 4.0%, 1.5% and, 1.8% for an average improvement of 2.5%.

This supports the conclusion that the eyes are of particular importance when detecting emotion in an image and further suggests that highlighting facial features that humans use to determine emotion improves an artificial emotion detection system as well. Additionally, the CNN consistently outperforms the NN. This supports the findings of [23] and the conclusion that deep convolutional neural networks can be used to more accurately classify facial images. Unlike [23] however, this experiment focused on emotion instead of facial detection or recognition.

Systems using Gabor filters performed overall better for both of the CNN configurations, however they performed worse than their counterpart models for the NN system. This may be because the NN systems lacked the number of parameters to accurately encode the information provided by the Gabor filters. All systems performed better when eye information was emphasized. In each case, the system with the best performing configuration was one that extracted the eye regions for emphasis. The systems using a CNN instead of a NN for classification generally performed better than their

counterpart systems. Systems using a CNN for classification also tended to converge faster to a higher accuracy rate as can be seen in Fig.3. In all cases, emphasizing the importance of the eye regions by appending them to the face image improved accuracy. This confirms $H2$ and aligns with the intuition that eyes provide a significant amount of information that can be used to determine facial expression. The variable that had the largest positive or negative impact on classification accuracy was the type of classifier used. The 12.1% improvement in exact match accuracy and the 4.7% improvement in top-2 accuracy indicate that the most significant factor in developing an accurate facial expression recognition system is the classifier. While feature extraction and selection both can have a significant impact on the classifier, they serve mainly to enhance or fine tune an already relatively accurate system. A high quality feature extraction and selection method may be of greater significance in a situation where computing power to train a classifier is limited or more precise results are required.

The top-2 accuracy results reported by this system are very accurate and provide an interesting insight into the fact that when the system's first choice is wrong, its second choice is usually correct. It has been pointed out by other research like [1] and [33] that this may be in line with actual human perception. The classification of the dataset used in the present work was done by asking the subject to pose with a specified emotion. This leaves room for error in the labeling of the classes where the subject might not have accurately portrayed the emotion. Further, as the author of the dataset indicates, in each of the images, there is not pure representation of just one emotion but rather a mix of different emotions. Therefore, identification of the most likely set of emotions displayed in an image may be a more accurate representation of reality than attempting to choose one particular emotion.

VII. Conclusion

Overall, the results of the experiment validate each of the three hypothesis. $H1$ is supported as the CNN based systems consistently outperformed the NN based systems. $H2$ is supported by almost all cases where eye image inputs showed higher accuracy than counterpart systems with no eyes detected. $H3$ is only partially supported by the results as Gabor filters positively impacted systems with CNN and negatively impacted systems with NN classifiers.

Based on the observed results a few conclusions can be drawn that can serve as a guide for further research. First, that a complex, deep convolutional neural network outperforms a simpler network was clearly demonstrated. This confirms results from [11]. Second, the significance of the eye in classifying emotion is clearly demonstrated. Both systems where the eyes were extracted and concatenated with the original image show significant improvements in accuracy. Third, it can be reasonably concluded that texture information plays a role in detecting facial expressions as shown by the improved accuracy of systems using Gabor filters along with CNN.

The current work provides contributions to the current body of research in four major areas. First, a robust system for facial expression recognition is described in detail. Second, the conclusion that a complex CNN is more accurate for facial expression recognition than a simple NN. Third, that the eye region encodes a significant amount of information related to the facial expression. Fourth, that texture information is also significant to the facial expression recognition process. Each of these findings can be built upon and provide a starting point for future research.

While this work has shown the significance of the eyes in facial expression recognition, further work extracting the mouth, or eyebrows could show how each major facial feature decodes facial expression information. This work could be expanded to analyze how other features of the face impact expression recognition accuracy. As prior work has used eyebrows, eyes, the nose, and the mouth to detect faces in images, examining the relative impact of each of these features on expression recognition seems like a logical next step. This in turn could show a way to reduce input to the classifier to only facial features that have a great impact on the expression, thereby reducing noise in the classifier.

There are a few limitations that are apparent in the current work. Primarily, the dataset size limits the potential any classifier trained from the dataset to be used more broadly. While the current work has been done using only images of Japanese females, the findings of [18] indicate that there are facial expressions that are both cultural as well as universal. As the work presented here was done using the expressions that [18] demonstrated are universally recognizable, the work may be more generalizable to other datasets than one might initially expect. However, it is likely that the model system used in the current work could be trained on a more robust dataset and provide more generalizable results. Studies examining the current systems' performance on other common datasets would be able to demonstrate how this technique generalizes.

Further, the current system is dependent on face images that are in an upright position in images. The classifier can only classify the faces that the face detection module detects. Therefore, a more robust system for facial detection could be used to improve the current system. For example, a system that accounts for head tilt and role like the ones proposed by [3] could produce a system that is able to identify facial expressions in a wider variety of cases.

While there has been other work done examining the ability of automated systems to recognize facial expressions in images, there is not a consistent benchmark by which to make direct comparisons. In order to compare this work with prior work on the same dataset, a similarity in training and testing configuration and potentially image scaling, would be needed. However, the numerous opportunities for variation in training and validation set configuration make that difficult to do. A standardized way to calculate an appropriate validation set size would be of use in the current work and to the research community at large. Additionally, a standard calculation of appropriate size of k based on the sample size when using k-fold cross validation would be useful to allow direct comparison of research work.

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] 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.

[17] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu, "Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron," in *Proceedings Third IEEE International Conference on Automatic face and gesture recognition*, pp. 454–459, IEEE, 1998.

[18] 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.

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

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

[21] 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.

[22] 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.

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

[24] 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.

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

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

[27] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[28] S. E. Umbaugh, *Digital Image Processing and Analysis: Applications with MATLAB and CVIPtools*. CRC Press, 2017.

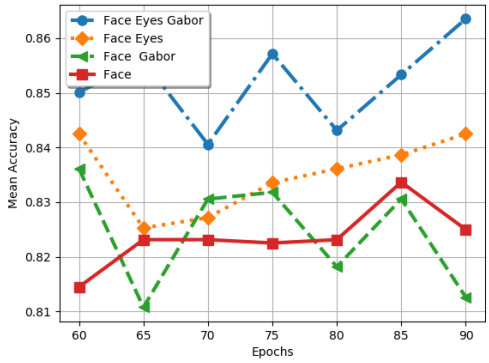
[29] S. W. Janik, A. R. Wellens, M. L. Goldberg, and L. F. Dell'Osso, "Eyes as the center of focus in the visual examination of human faces," *Perceptual and Motor Skills*, vol. 47, no. 3, pp. 857–858, 1978.

[30] M. R. Dadds, Y. Perry, D. J. Hawes, S. Merz, A. C. Riddell, D. J. Haines, E. Solak, and A. I. Abeygunawardane, "Attention to the eyes and fear-recognition deficits in child psychopathy," *The British Journal of Psychiatry*, vol. 189, no. 3, pp. 280–281, 2006.

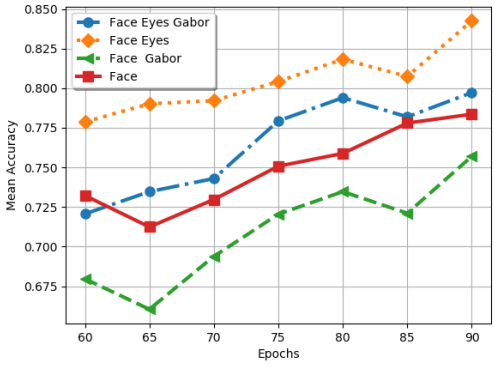
[31] S. J. Bayless, M. Glover, M. J. Taylor, and R. J. Itier, "Is it in the eyes? dissociating the role of emotion and perceptual features of emotionally expressive faces in modulating orienting to eye gaze," *Visual cognition*, vol. 19, no. 4, pp. 483–510, 2011.

[32] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

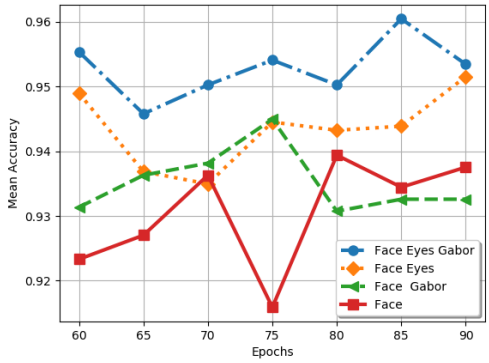
[33] M. Lyons, M. Kamachi, and J. Gyoba, "The japanese female facial expression JAFFE database," Apr 1998. n.b. Publication or exhibition of the images in any form requires explicit permission. Redistribution of the JAFFE database is not permitted.



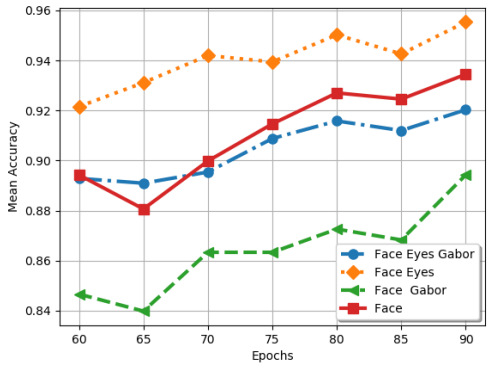
(a) Exact Match Deep Model (CNN)



(b) Exact Match Simple Model (NN)



(c) Top 2 Deep Model (CNN)



(d) Top 2 Simple Model (NN)

Fig. 3: Epochs vs. Average Accuracy