Machine Learning for Facial Expression Recognition: A review

Silas Curfman
Dept of Electrical and Computer Engineering
The University of New Mexico
Albuquerque, NM 87131-0001, USA
sgc98944@unm.edu

***Abstract*— The purpose of this work is to demonstrate to the reader an implementation of a machine learning workflow built to detect and categorize various emotional expressions on digital images containing human faces. We first briefly review the history of the problem of human facial expression recognition, both analog and digital methods. We then summarize the typical workflow necessary to successfully construct a machine learning method in general. From this workflow we select a suitable approach for Facial Expression Recognition. Finally, we construct a functioning implementation and test it on a selection of images with unknown facial expressions. We complete the implementation by measuring it's performance against the ground truth values. Machine learning methods prove to be effective and applicable to the field of human emotion recognition.**

Keywords—Special Affect Coding System (SPAFF), Facial Action Coding System (FACS), Facial Action Scoring Technique (FAST), Emotion Facial Action Coding System (EMFACS), Automatic Facial Coding (AFC), Facial Action Reciprocity (FAR), Action Units (AU), artificial neural net, machine learning, Facial Emotion Recognition (FER), support vector classifier (SVM)

I. Motivation

As the availability of both algorithms for and data sets for the automated recognition of facial emotions among humans has greatly increased, it is the intent of this paper to take a high level review of the various feature classification approaches to this process. By doing so we hope to identify a framework for selecting appropriate data sets according to the intended implementation of a given facial emotion recognition project.

II. Expected Contribution

We intend to contribute an historic review of how and why some of the feature coding methods have evolved and where they might agree or disagree with non-automated or manually coded emotion recognition research.

III. State of the ART REVIEW

A. History

According to Park(2018), "Nonverbal components contribute to over 90% of effective communication and help the appropriate delivery of feelings and attitude". Such a distinct imbalance between the presence of nonverbal vs verbal communicating in a given interaction gives increased import on correctly identifying nonverbal communication cues. Further, according to Buhler (2009) "Nearly 75% of communications that are received are interpreted incorrectly". This is a significant error rate for such a high value indicator. Missed or misinterpreted communication cues can be, and are, the source of countless problems in every field of human interaction.

The academic exploration of human nonverbal communication has a long history, overlapping both the pre- and post-digital revolution. For example, Charles Darwin's book "The Expression of the Emotions in Man and Animals" was first published in 1872.

Figure 1: Darwin (1872)

In the intervening years a number of different approaches have developed commensurate to the technological advances available at the time.

B. Analog

Prior to the use of Machine Learning methods in expression recognition, the most used approach has been video based facial coding analysis. Such methods, like Gottman's (1995) Specific Affect Coding System (SPAFF), require a trained technician to manually review and label the emotions expressed by a subject that has been video recorded during the course of some degree of interpersonal communication.

Figure 2: SPAFF, Gottman (1995)

There is a large body of knowledge on these manually reviewed approaches. While time consuming and requiring trained technicians, the results have been put to very good use in every arena of human interpersonal communication.

C. Digital

With the advent of Machine Learning methods, specifically computer vision methods, we can now experiment with the automation of such processes. As we continue to test automated machine learning methods, we can compare their performance to the existing body of knowledge built through the manually coded approaches.

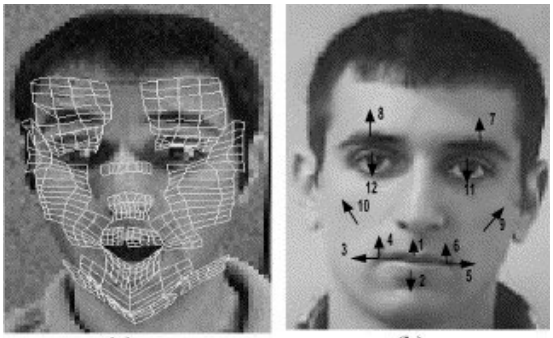

Figure 3:Cohen (2003)

IV. Methods

This paper will review an existing end to end approach of building a facial emotional recognition machine. In the process of building such a machine we will review the impact of the classification schema chosen on both the process of building the machine and the functional outputs of it.

A. Generalized Machine Learning Workflow

1) Statement of Problem

Machine Learning is best suited for problems that formatted as; Numeric, Categorical, or Ordinal. Combinations of the three can be solved through the use of hybrid or ensemble methods.

2) Data Collection

Every Machine Learning solution needs a body of known data and labels from which to build possible solutions from. Generally the larger and more uniform the data set the better. If none are readily available, the subject will need manually construct a suitable data set for their project.

3) Data Pre-processing

Nearly all data will need to be manipulated in some manner before it can be used to train the model. In the case of images, the most common manipulations are to regularize the scale, resolution, color, and composition of the images.

4) Model Selection

Model selection will be dependent on the type of problem, the size of the dataset, and the type of data the subject will be working with. Generally, problems will follow either a Classification or a Regression approach. This could be thought of as categorical vs numeric in nature. Further, the degree to which the model can will be helped with a priori (pre determined) structure will matter. Supervised learning provides more structure to the model where Unsupervised provides a minimal amount of structure.

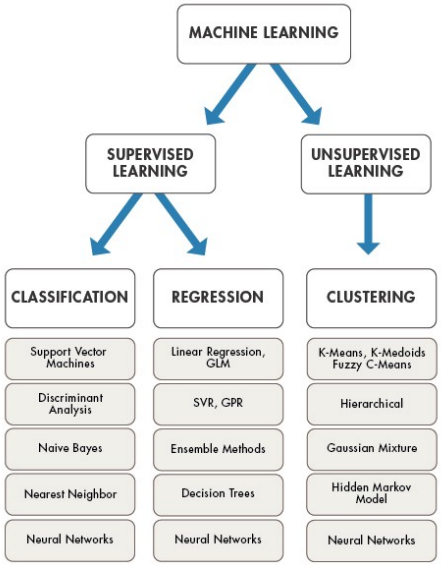

Figure 4: Model Selection

5) Model Training & Testing

Model training and testing follows a well defined approach regardless of algorithm. A general rule of thumb is to train a model on 80% of the data set, retaining the labels during the training process. Once trained or 'fitted' the model is run against the held out test data, without the labels. The testing process 'predicts' the label that should be associated with a given instance.

The training and testing process can be further extended by repeating the process of a series of iterations where the training set and testing set are reconstructed over a number folds. This fold, test, refold, retest approach is cross-validation and it can help prevent overfitting, or a too rigid match to the original data.

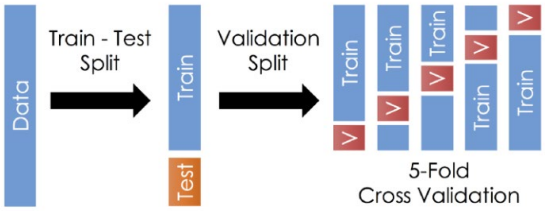

Figure 5: Cross Validation

6) Measure & Refine

The results of each iteration of training and testing can be evaluated by a number of performance measurements. Some of these include; F1-score, ROC curves, AUC scores, Precision, and Recall. It's also useful during this phase to visualize the performance through the use of Confusion Matrices.

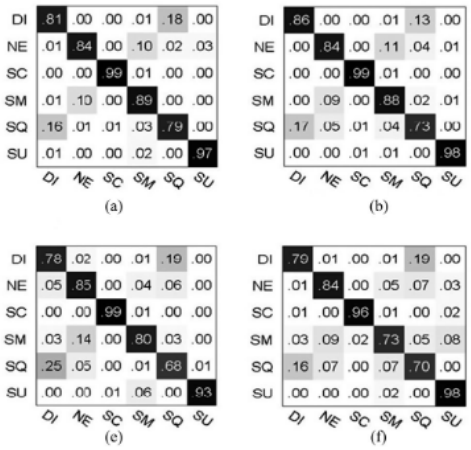

Figure 6: Confusion Matrix

From these performance measures the subject will make various changes or adjustments to parameters of the Model and run again. These parameters, that are user settable, are Hyperparameters.

An alternative to manually adjusting Hyperparameters is to run an automated test to look for the optimum Hyperparameters. This approach is called a Grid Search. The optimum parameters will always be a trade off. As performance increases, accuracy usually decreases. As speed goes down, cost often goes up. The user will stop the refinement process when a suitable balance between performance and accuracy has been found.

7) Deploy Final Model

Generally, a Machine Learning project will need to be deployed in some form of production environment in order to useful. This may be over a local network in a small business, hosted over the cloud, or some variation of either.

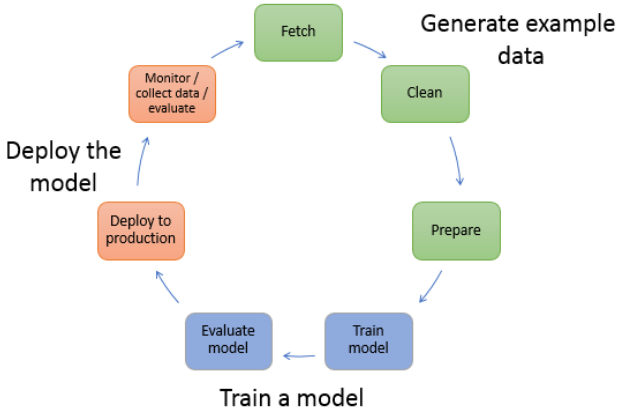

Figure 7: Amazon AWS (2022)

B. Project Model

1) Statement of Problem

Accurately predict the emotional expression on human faces present in a given digitized still image.

2) Dataset

This project will use the JAFFE(Japanese Female Facial Expression) dataset. This is a relatively small dataset consisting of 213 8-bit greyscale images. Each image is a photograph of a Japanese female subject expressing one of seven emotional expressions. The photographs are highly regularized, subject looking directly at the camera, no background objects, and all lit in a consistent manner. Each digital image is exactly 256 x 256 pixels. This is dataset is distinct from others in that it is has a low number of total samples and a low degree of entropy.

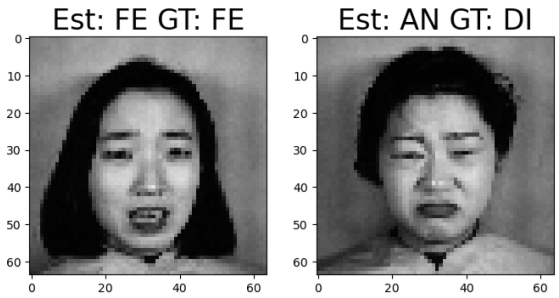

Figure 8: JAFFE dataset

Alternative datasets suitable for use in FER include the Facial Expression Recognition 2013 Dataset (FER2013) and the Extended Cohn-Kanade dataset (CK+). FER2013 is comparably both a high sample, high entropy dataset. It contains 30,000 facial images recorded in RGB color. It too uses a list of 7 emotions as target labels. As seen in Figure 9 the subjects photographed in FER2013 are framed in a wide variety of look angles, lighting, background objects, and ethnicities.

Figure 9: FER2013 samples

Likewise, the CK+ dataset is much more generalized than JAFFE, having a total of 5,876 labeled images from photographs of 123 subjects, also in a variety of different poses and background lighting.

3) Data Pre-processing

Processing will require image pixel transformation and normalization. Categorical labels will be converted to numeric. A color decomposition step will be added that will reduce an RGB color images to a greyscale representation. In the case of the JAFFE images, which are already greyscale, this process will run but be redundant. By having it in place though, the workflow will be suited for later importing of RGB images.

4) Model Selection

The initial implementation will compare results from a Support Vector Classifier and a Multi Layer Perceptron. Also included here is a suggested implantation of a Convolutional Neural Network for FER classification.

The SVM model (Figure 8) is the most simple, as well as the most likely to be overfit of the models. It's useful to include though as a prototype to quickly debug data framing and pipeling of the machine learning algorithm. With an established workflow that successfully imports the data, preprocesses the images, splits the data sets, trains, fits, and finally predicts; such a workflow can readily be repurposed for a more robust algorithm.

The MLP algorithm shown in Figure 9 consists of a single input and output layers, as well as 1 or more hidden layers. The nodes in the hidden layers aggregate the inputs fed into them and then apply an activation function on those inputs. The MLP is a feedforward network that utilizes back propagation to update weights based on the loss or error(s) calculated when comparing the predicted labels to the ground truth labels.

The proposed CNN architecture differs from the MLP architecture in the additional preprocessing of the data before it is fed into the neural network. Each initial image is progressively decomposed via a scanning 'window' or kernel that iterates across the entirety of the image. This sliding window of some multiple of n-pixels by n-pixes tall and wide records the pixel values inside the window at a given progression across the initial image and further reduces that n x n array of data into a single scalar value. That scalar value is then appended to a new empty array and a new representation of the initial image is produces. This process then repeats over and over until the image is no longer decomposable by a sliding window larger than 1 pixel by 1 pixel. At this point the represented data is formatted into an input array to be fed into a neural network. The number of hidden layer and hidden nodes are hyperparameters to which the user will need to make some initial guesses to and constraints on. The CNN architecture differs from the SVM and MLP approaches in that it is much more processing intensive and produces better results with larger data sets.

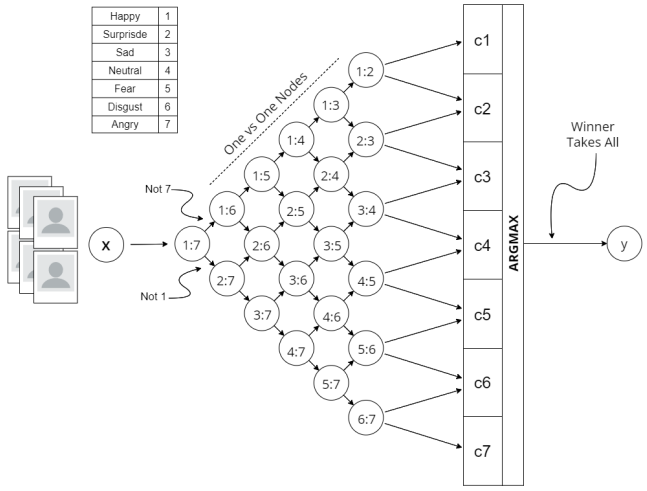

Figure 10: SVM FER Classifer

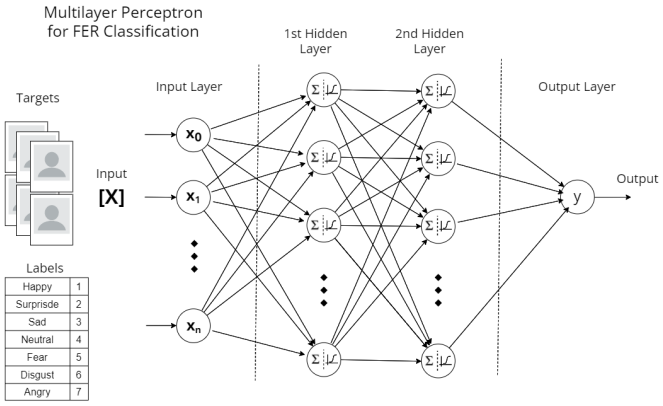

Figure 11: MLP FER Classifier

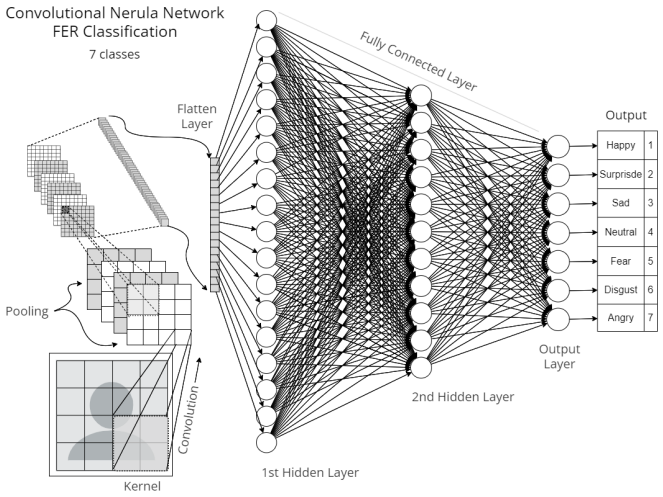

Figure 12: CNN FER Classifier

5) Initial Model Training & Testing

Beginning with the SVM model, each model was built and tested with some initial guess for starting parameters. During the initial runs checks were performed along the way to inspect the size and shape of the training arrays and test arrays. After the initial runs were performed, the results were measured for accuracy and plotted as confusion matrices. By reviewing these initial confusion matrices, suggested search parameters and constraints were determined for use in a 'gridsearchcv' optimization search. The results of the 'gridsearchcv' optimization was compared to the confusion matrix of the initial run and the process was either repeated or completed based on the level of improvement or minimization of errors.

MODEL	SVM-OVR	MLP
Hyper Parameters	C: 1.0 Kernel: rbf gamma: 0.001	Solver: lbfgs Alpha: 0.00001 Hidden Layers / Nodes H1 : 1000N H2: 200N
Train : Test Split	80 : 20	80: 20
PERFORMANCE SCORES		
Precision	0.24	0.86
Recall	0.28	0.81
F1	0.24	0.80

Figure 13: Initial run parameters

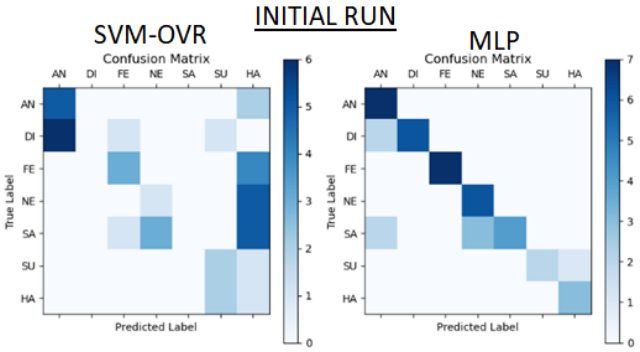

Figure 14: Initial run results

6) Measure & Refine

Shown here in Figure 15 and in Figure 16, it is clear that while the SVM model benefited greatly from the gridsearch optimization, the MLP model was already approaching the upper end of any accuracy improvements gridsearch would be able to produce. This is to be expected, that an initial run of the SVM model would likely show more error than the same run of the MLP model.

Also, while at first glance it looks like the SVM model does surprisingly well at such a complex task as Facial Expression Recognition, it should be recalled that this initial run is on the JAFFE dataset, which is both very small and very regularized. It would be reasonable to expect the performance of the SVM model tuned here to be much worse when provided much more generalized inputs, such as those used in FER2013 or what might be provided in a real world production environment.

MODEL	SVM-OVR	MLP
Hyper Parameters	C: 1.0 e^7 Kernel: rbf gamma: 1.0 e^-6	Solver: sgd Alpha: 0.0001 Hidden Layers / Nodes H1 : 200N H2: 200N H3: 200N H4: 200N H5: 200N H6: 200N
Train : Test Split	80 : 20	80: 20
PERFORMANCE SCORES		
Precision	0.94	0.85
Recall	0.91	0.81
F1	0.91	0.81

Figure 15: Best parameters via gridsearch

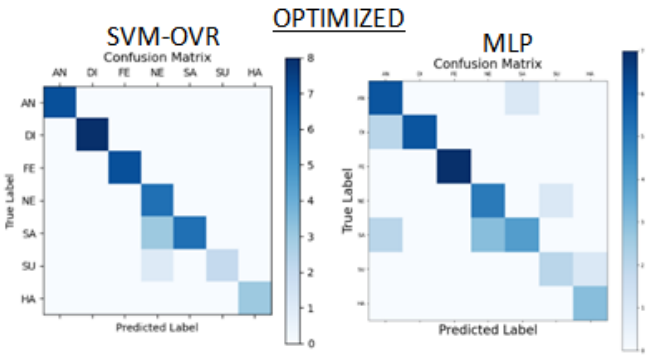

Figure 16: post optimization metrics

7) Deployment

While the scope of this review does not call for deploying these methods in a production environment, the next most logical step would be to make them accessible for further testing on real world images. This could be done through hosting on a cloud computing platform and incorporating a means for a remote user to upload either still images or real time images via a webcam for testing classification.

V. Conclusion

The intent of this review was to familiarize ourselves with the history of, the current state of the art of, and the expected changes that may soon happen in the use of machine learning methods for the classification of emotional expressions presented on human faces.

From an economic motivation, it's clear through this review that none of the methods demonstrated here required costly resources and were all within the grasp of the early professional and even the amateur data scientist. There remains great potential for high value returns on low cost investment in this field. If the maxims that "90% of human communication is non-verbal" and that "75% of non-verbal communication is misinterpreted" are even only partially accurate statements; it would still be a wise investment to further explore the benefits of augmenting human interpretation with computation methods such as ML FER.

From an academic perspective, facial expression recognition applications of machine learning represent a great opportunity for cross discipline influence between the sciences and the arts. Such cross pollination of ideas and influence often precedes large advancements human understanding.

Finally, from the point of view of the user / programmer, an exercise in building and demonstrating such algorithms offer a great opportunity for an all-in-one training approach as well as producing a final product that can be used to generate interest from individuals not otherwise engaged in computer programming.

Bibliography

[DRAFT COPY ONLY]

[REFERENCES IN PROGRESS]