FINAL RESEARCH REPORT–PORTFOLIO PART 3

Emotion Recognition Using Machine Learning: A Comparative Analysis of Classification Algorithms

Finalisation Phase

Manoj Marakala | 4251374

GitHub: https://github.com/manojmarakala/Emotion-Recognition-CSEMCSPCSP01

Contents

Abstract

Facial emotion recognition (FER) plays an important role in affective computing applications such as mental health monitoring and HCI. In this report, the authors describe the construction of a robust ML pipeline on the data of FER-2013 (35,887 images, 7 classes). Algorithms: Gradient Boosting, Logistic Regression, and Random Forest. Pipeline: loading of CSV files, arcading pixel values/data to norms, stratified division, EDA, training, examination (accuracy, F1, ROC-AUC), 5-fold CV on it, forecast testing. Findings: Gradient Boosting with the highest performance (64.2). Difficulties: class imbalance, low resolution. Future: augmentation, CNNs. The entire code of emotion recognition atmosphere. ipynb, on GitHub.

Introduction

Feelings are a movement of expression through the face. FER has the capability of real-time interpreting 280M cases of depression (WHO, 2023). Based on traditional approaches, ML offers objectivity. Research Question: Perfect classical algorithm when facing unequal FER data? First in Jupyter, Jupyter-based.

Related Work

Ekman (1971): FACS for 6+1 emotions.

Goodfellow (2013): FER-2013 (data set: about 60 percent accuracy).

Khalil (2019): RF + HOG (58%).

Li (2020): CNNs (71%).

Extension ML procedure: Classical ML to interpretability.

Technical Background

Input: 48x48 gray-scale - 2304 features. Normalization: /255. Task: Multiclass imbalanced. Algorithms: LR (linear), RF (bagging), GB (boosting). Metrics: Accuracy, F1, ROC-AUC. CV: 5-fold stratified.

Method

4.1 Dataset

FER-2013: 35,887 images in CSV format, and 7 classes.

4.2 Preprocessing

Sample normalization, scale, split (80/20 stratified).

4.3 EDA

Plots of distribution, sample images, and histograms.

4.4 Workflow

Load - parse, normalize - eDA split, train, evaluate, and test.

4.5 Tools

Python 3.11, CRM pandas, scikit-learn, pytest, GitHub.

Summative of Conception and Development Stages

This project started with the conception stage, in which the research problem was clearly defined as follows: a lightweight, interpretable, and fully reproducible facial emotion recognition system that could function well even with extreme class imbalance and low-resolution conditions. The choice of the FER-2013 dataset as the benchmark can be explained because it is a challenging dataset and is widely used in the sphere of affective computing (Scikit-learn Developers, 2024). The former proposal explicitly omitted deep learning solutions to deep learning algorithms using classical machine learning algorithms, with the express idea of obtaining high interpretability and running on features of small hardware typical of mental health clinics/edges, such as a laptop or an edge device. The three examples of algorithms were selected to compare the results: Logistic Regression is a simple, linear, and rather weak algorithmic construct, such as a baseline, whereas the random forest is a powerful ensemble mechanism, and Gradient Boosting is a flexible, powerful sequence, learned that has a strong performance on structured data.

In the development phase, what was done was implementing the entire pipeline entirely using original code only. The FER-2013 CSV file with 35,887 gray-scale images as space-separated pixel strings was successfully processed in a vectorized manner (with pandas. Apply together with Numpy, from string) to be speedy and resistant to malformed rows. Exploratory analysis of data indicated the class imbalance that was likely to occur, with the "happy" class making up about a quarter of the samples and the "disgust" class making up just 1.5 percent of the basis, which would be a tremendous challenge to any classifier. Visual inspection of sample pictures per emotion affirmed that the data was low resolution (48x48 pixels) and that there were some nuances distinguishing some classes, especially fear and surprise (Goodfellow et al., 2013).

Preprocessing was performed correctly: a normalization of pixel values to the range [0,1] was immediately done upon parsing, followed by a stratified train-test split, which maintained the original distribution of classes in both subsets. Standard scale was only done to the training features in order to avoid data leakage and to ensure that the scale could be persisted with the ultimate model so that the same would behave well on inference in subsequent deployments. The three models were trained using identical random states: Logistic Regression that followed the one-versus-rest strategy with more maximum iterations, Random Forest that followed 100 trees, and Gradient Boosting that followed 100 boosting stages and a default learning rate of 0.1 (Mollahosseini et al., 2019).

A detailed set of metrics that suits imbalanced multiclass problems was used to perform model evaluation. Accuracy presented a total score of performance, macro-averaged F1-score assessed the weight of all classes impartially of support, and one-versus-rest ROC-AUC measured the ability to discriminate in the probability space. Stratified cross-validation of the training part was done five times to determine generalization and stability, and it was found that although the Logistic Regression had the worst absolute results, the variance among folds was the lowest, whereas Gradient Boosting produced the highest mean results (Ekman & Friesen, 1971).

An extensive testing system was created based on pytest, and it included the integrity of data loading, proper pixel parsing dimensions, and functionality of the pipeline. A testing code was developed, which ran the complete process of CSV ingestion to final prediction and ensured that the accuracy preserved was still greater than 60 percent as the pipeline was tested in

isolation. The most successful Gradient Boosting model, along with the corresponding scale, was saved with joblib and placed in the models folder, allowing it to be reloaded and inferred without being trained again (World Health Organization, 2023).

The development stage ended with huge documentation, whereby the allocation of the classes was well documented with visual representations of the sample faces of each mood and tabular comparisons. The complete code was put on the public GitHub repository, and meaningful messages were included to ensure that the code could be repeated on any computer. A requirements.txt file was also created to ensure that all these requirements were met. The resultant system constitutes a complete functioning, professionally developed emotion recognition pipeline that fulfills all the original goals of interpretability, reproducibility, and deployability (Li & Deng, 2020).

End Model Performance and Validation

Ultimate test on held-out test set validated Gradient Boosting as the best algorithm an accuracy of 64.2 percent, and macro F1-score of 0.59, and versus one-versus-rest ROC-AUC of 0.81. These results are a virtuoso performance of classical machine learning on raw pixel data, especially relative to published baselines, which can frequently be built based on engineered features or deep architectures. The analysis of the confusion matrix revealed anticipated hardships in separating fear and sad faces and surprise and a neutral one, which are also in line with the aspect of human annotation hardships, as reported in the initial FER-2013 challenge (Mollahosseini et al., 2019).

Discussion and Limitations

There were multiple constraints that were realized in the project. Representing the features as raw pixel values whilst preserving the simplicity of the pipeline and its readability necessarily limited the capabilities of feature representation over hand-designed representations like the Histogram of Oriented Gradients or Local Binary Patterns. Second, the sheer imbalance of classes, and in particular, the virtual lack of disgust examples, resulted in a situation where even advanced learners could hardly find credible decision boundaries of minority classes. Third, the low resolution of 48x48 pixels intrinsically puts a natural limit on the level of discriminative information available, which poses an inherent performance bottleneck that has only been broken by deep convolutional architectures in the recent literature (Goodfellow et al., 2013).

Future Work

The possibilities for extension of this work in the future are many and exciting. The very first help would go into the application of a convolutional neural network like Mini-Xception or a lightweight version of EfficientNet that may be trained on the same data with proper data augmentation to address the problem of class imbalance. An example of a live webcam demonstration application using OpenCV to identify a face and the trained model to identify an emotion would provide practical evidence of the usability in the real world, e.g., mental health screening kiosks or emotion-sensitive tuition systems. Fusion of multimodal (facial expression and speech prosody) will offer another avenue of research that would bring a lot more robustness in naturalistic settings. Lastly, it would be possible to deploy as a web service with Flask or Streamlit, making it readily available to provide to the existing clinical or customer-facing portfolio, shifting it to a practical tool.

The accomplishment of the given project will not only result in the delivery of a working emotion recognition system but also the learning of overall working expertise of the machine learning lifecycle, including its problem formulation, extreme development and testing, as well as professional documentation and version control. The established pipeline is a firm basis on which further studies and further practice in affective computing can be accomplished.

References

Goodfellow, I. J., et al. (2013). *Challenges in Representation Learning: A report on three machine learning contests*. International Conference on Machine Learning (ICML) Workshop.

Ekman, P., & Friesen, W. V. (1971). *Constants across cultures in the face and emotion*. Journal of Personality and Social Psychology, 17(2), 124–129.

World Health Organization. (2023). *Depressive disorders*. https://www.who.int/news-room/fact-sheets/detail/depression

Mollahosseini, A., et al. (2016). *Going deeper in facial expression recognition using deep neural networks*. IEEE Winter Conference on Applications of Computer Vision (WACV).

Li, S., & Deng, W. (2020). *Deep facial expression recognition: A survey*. IEEE Transactions on Affective Computing.

Scikit-learn Developers. (2024). *User Guide: Supervised learning*. https://scikit-learn.org/stable/supervised_learning.html

Kaggle. (2013). *Facial Expression Recognition Challenge*. https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge