Assignment 2: Detecting emotions with ML algorithms

Expressing emotions is one of the things that make us humans. To recognize emotions from facial expressions is a task which in computer vision is often referred to as "facial expression recognition" or "emotion recognition". In this project you will be performing emotion recognition by training machine learning algorithms that operate on images.

The goal of this assignment is to design system which is able to facial expressions within some reasonable levels from the camera of your laptop in (approximately) real time. To achieve this goal you need to make a small pipeline that identifies the face, extracts the face picture from the camera stream, extracts features, and uses a machine learning algorithm to classify the expression in the image. You need to critically reflect on the design choices that you perform for each of the steps. Moreover, you need to report the performance of your model in terms of accuracy, the confusion matrix for the private and public test sets.

Dataset

For this assignment we use as starting point the FER 2013 dataset [R1] which has the following characteristics

- The images are 48x48 grayscale patches which contain faces expressing the emotions labeled as "angry", "disgust", "fear", "happy", "sad", "Surprise", "neutral".
    - As we saw, the FER 2013 challenge implies to recognize 7 emotions. If this is too difficult because class imbalance, you can simplify the assignment by excluding class labels. However, this will have a penalization in the assignement.
- There are different versions of this dataset in kaggle. Use the version where all the dataset is stored as a singles .csv file, in which the numerical values of the images are stored, as well as the class labels (https://www.kaggle.com/datasets/deadskull7/fer2013). This version is easier to use, as you can read it out of the box with pandas's read_csv routine. You may need to reshape the image data to 48x48 images for feature extraction.
- The dataset from the challenge already contains specified train and private and public test splits. The dataset originally belong to a challenge and the test set was not known to the participants. Please also fine tune your models on the private test set and report the performance both, the public and private test sets.

- Beware that the dataset is imbalanced (the number of samples per class is not the same). You will have to deal with this to achieve better performance 😉.

[R1] Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., ... & Bengio, Y. (2013, November). Challenges in representation learning: A report on three machine learning contests. In *International conference on neural information processing* (pp. 117-124). Berlin, Heidelberg: Springer Berlin Heidelberg.

Technical aspects to be evaluated

Feature extraction for training machine learning classifiers

You are free to choose which feature extraction to use. Examples of features are facial landmarks, histogram of oriented gradients, eigenfaces/eigenspaces, Haar-like features, pixel, local binary patterns, pixel intensities etc. Note that the choice of features can have severe impacts in the computation time for both, training and testing. For reporting, we are interested in knowing why you choose for a particular feature, what it is modeling and this features help to solve this task.

In addition to feature extraction, further ways to improve the performance of your method could be to perform data augmentation (e.g. color jitter, horizontal flips). This augmentations should make sense for this application. (e.g. flipping the image up and down does not make any sense , nobody recognizes images like that).

Machine learning model

The algorithms to be used are the following

- SVM
- MLP
- Random forest
- Fuzzy system

In your report discuss the performance of each model, which one is better for this application and why. Note that different classifiers may perform better with specific features. Also in the case of the fuzzy classifier it should be observed that this model is suited to relatively low-dimensional problems. Therefore you need to use either lower-dimensional features, to use a dimensionality reduction, or to use feature selection.

Real time demo of your algorithms

Part of this assignment will consist of implementing a simple pipeline that uses the best model that you train and classifies images that are extracted form the camera of your laptop and shows the classification of your face expressions. The face extraction for the demo can be performed with Open CV for python, For this you can follow (this tutorial can be helpful https://www.geeksforgeeks.org/python/face-detection-using-cascade-classifier-using-opencv-python/, also this one https://www.datacamp.com/tutorial/face-detection-python-opencv). Moreover Open CV also has the routines to get the camera feed (VideoCapture). Also it can be performed with Dlib. This last library also has models for landmark detection.

Reporting

For reporting we expect a report of maximum 6 pages. The report should clearly describe

- Handling of data imbalance.
- Preprocessing, feature extraction, data standardization (if applicable).
- Choices of hyperparameters in your machine learning models.
- Obtained performance for each of the algorithms.
- Discussion on the performance and the process to develop the delivered solution.
- Issues limiting performance of the algorithm (data, models, etc).

What to deliver?

You have to deliver a .zip file with the following files:

1. The report in PDF.
2. Jupyter notebook where you trained your models and do the performance measurements.
3. A set of .pkl files which contains your trained models.
4. And a demo python file which we can call from the command line to execute the code and which will prompt the camera feed where the teaching team can test your best performing algorithm.
5. The report in PDF.

If the feature extraction file is too big you optionally can put the feature extraction in a separate python file. This is the only exception, the rest of the code should be implemented in the notebook.

What are we grading?

Think along the following points

1. Understanding of the datasets. How does the data looks like? What are its characteristics? How does this characteristics influence the design decisions you make? (e.g. feature extraction or other hyperparameters).
2. Understanding the modeling. How do you reflect your understanding on the design choices you made?
3. Evaluating the results. How do you evaluate the effectiveness of your model?
4. Reflecting on the modeling results. Discuss which model (fussy model, SVM, CNN) is the most fit for this application and elaborate why.
5. How does the implementation looks like? Is the code readable?

Code

The report and code should be packed as a single zip file. The packages for this algorithm should be the standard ones (numpy, scikit learn, scipy, open CV. etc.). For the neural networks, please refrain from using large models (e.g. CNN, vision transformer, etc.), Scipy has a built in neural network module, which is preferred. The objective of this assignment is not to build a state-of-the-art model but to develop basic machine learning models and to get hands-on experience in their development process.

Python libraries

For this assignment we will use the same libraries as usual. New libraries that can be used are Dlib and/or Open CV for handling the camera and landmark detection (if applicable). For the fuzzy classifier PyFUME (https://github.com/CaroFuchs/pyFUME), which needs Numpy (Numpy 1.24.4). HIGHLY recommended new environment for this. Otherwise you may break the environment you use for the other models. An additional class thagt can be used is imbalanced learn.

Besides the specified environments you will use the usual suspects (numpy, scipy, skimage, sklearn, scikit-learn). For the MLP also use scikit-learn. We are not aiming for state-of-the-art neural networks here, but to test your understanding of the topics in this course. Alternative libraries will not be considered unless there is a very good reason for using them.

Use of AI

As discussed in the first lecture, it is not forbidden the use of AI as long as you are honest and report how and for which parts of your work have been used. Despite of this, we want to evaluate your knowledge and learnings from the assignment, so avoid using the AI model to generate everything as this will be considered when grading.

Challenges

There are a few challenges that you should consider for implementing your algorithm. For example, there is data imbalance (some classes are better represented than others). Moreover, the dataset is relatively large with respect to what you have used so far. Therefore, depending on your computer and the size of your features, you may need to train with minibatches, rather than with the entire dataset. In addition, the choice for features is not trivial and it requires some tests to get a decent performance. This is particularly important for the fuzzy classifier, as these models are design to work with relatively low-dimensional features.