Facial Emotion Recognition for Automated Interview feedback

Maël Fabien, Research Project with Pôle Emploi

***Abstract*— Affective computing is a field of Machine Learning and Computer Science that studies the recognition and the processing of human affects. The aim of this project is to provide job-seeking candidates a platform that analyses their their facial emotions when answering pre-defined questions using their webcam. I built deep-learning computer vision pipelines using XCeption architectures and deployed the trained algorithm on a Flask web app.**

I. SELECTED APPROACH

The french employment agency aims to leverage real time facial emotion recognition as a way for job-seeking candidates to perceive automated feedback on their interview performance.

A. Training Data

Training deep learning pipelines for real-time facial emotion recognition requires annotated training data. I have used the FER2013 Kaggle Challenge data set. The data consists of 48x48 pixel grayscale images of faces. The labels of the images are: Happy, Sad, Angry, Fearful, Surprise, Neutral and Disgust.

B. Convolution Neural Network

In the field of computer vision, the new standard is the convolution neural network (CNN). CNNs are special types of neural networks for processing data with grid-like topology.

In typical SVM approaches, a great part of the work was to select the filters (e.g Gabor filters) and the architecture of the filters in order to extract as much information from the image as possible. With the rise of deep learning and greater computation capacities, this work can now be automated. The name of the CNNs comes from the fact that we convolve the initial image input with a set of filters. Mathematically, the convolution is expressed as : $(f * g)(t) = \int_{-\infty}^{+\infty} f(\tau)g(t-\tau)\partial\tau$

C. XCeption Architecture

One of the main challenges in this task is to limit overfitting implied by the large class imbalance and the number of parameters that need to be learned. Xception is a deep convolutional neural network architecture that involves Depthwise Separable Convolutions, outperforming VGG-16, ResNet and Inception V3 in most classical classification challenges, while reducing the number of parameters, the training time and the risk to overfit.

The input image has a certain number of channels C, a dimension A and we apply a convolution filter of size d*d. For N Kernels, with a classical CNN, we must perform $K^2 \times d^2 \times C \times N$ operations where K is the resulting dimension after convolution. To overcome the cost of such operations, depthwise separable convolutions have been divided into 2 main steps: depthwise and pointwise convolutions. Through Depthwise an Pointwise convolution, the total number of operations compared to convolutions by a factor proportional to $\frac{1}{N}$.

The training takes 4 hours on Google Colab's GPUs and the accuracy I reached was 64.5%, slightly below state-of-the-art papers at 68%, but with much less parameters (around 1 million) and a faster prediction time.

II. IMPLEMENTATION

I have implemented a pipeline using Python, Tensorflow and OpenCV which analyses the video image by image, applies a grayscale filter to work with fewer inputs, identifies the faces using pre-trained model (Histogram of Oriented Gradients), transforms the input image to a model readable input and predicts the emotion of the input.

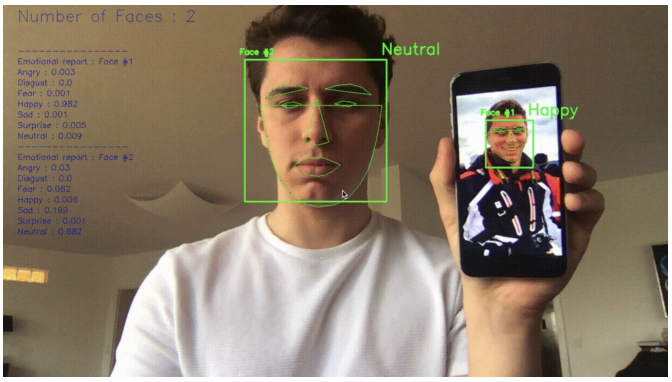

Fig. 1. Live facial emotion recognition

The trained algorithm is then deployed on a web application built using Flask. The user has 45 seconds to answer a single question. Using a D3.js interactive chart, the user is finally provided at the end of the interview of report of the perceived emotions as well as a benchmark of other candidates. I was able to present my work at the Nantes headquarters of Pôle Emploi in front of their data science team.

III. CONCLUSION

The scope of the work presented is limited by the availability of suitable training data for interviewing processes. The accuracy reaches close to 65% while maintaining a reasonable training time and an acceptable close to real-time performance. This work is now being implemented by the data science teams of Pole Emploi.