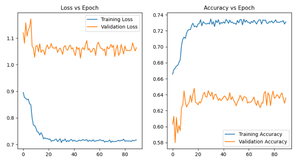
About the project

GOAL

This is a capstone project achieved in the context of CONCORDIA UNIVERSITY bootcamp in DataSciences .
The goal was to use a Convolutional Neural Network to detect and label emotions from the image of a human face.
7 types of emotion can be evaluated by the CNN model: [angry],[disgust],[fear],[happy],[sad],[neutral],[surprise]

The project was implemented by architecturing a Convolutional Neural Network and training it to recognize human Facial Emotions. The CNN Architecture is based on a smaller version of VGG. The training dataset was taken from the FER2013 dataset which is free to access and has over 28000 pictures. Upon filtering the most relevant pictures, the CNN was trained on 19774 pictures from this dataset.

PERFORMANCES



	precision	recall	f1-score	support
angry	0.42	0.51	0.46	67
disgust	1.00	0.47	0.64	80
fear	0.43	0.40	0.41	53
happy	0.80	0.88	0.84	84
neutral	0.56	0.71	0.62	78
sad	0.41	0.52	0.46	56
surprise	0.79	0.67	0.73	73
accuracy			0.61	491
macro avg	0.63	0.59	0.59	491
weighted avg	0.66	0.61	0.62	491

The total f1-score for this CNN on the FER-2013 dataset is at 61%. A good score would be over 90%. The score obtained here is not high, but still very close from the score obtained by the best competing teams on this dataset:

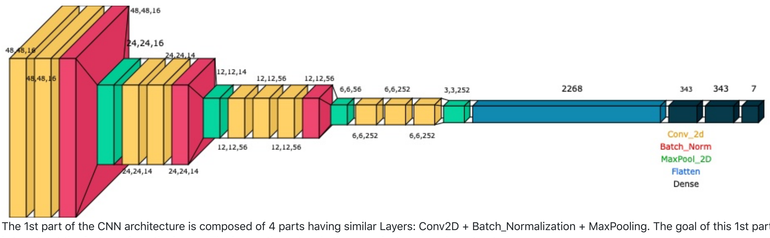
"It is one of the more challenging datasets with human-level accuracy only at 65±5% and the highest performing published works achieving 75.2% test accuracy." *http://cs230.stanford.edu*

Indeed, this dataset is rather challenging due to the way the images can be mislabelled and due to the various non-uniform poses taken by the people into pictures.

We can notice that the training accuracy is 73% while the validation accuracy is at 61%. This means that while the training accuracy could be improved by having a better dataset (persons pose, labelling correctness, image resolution, etc..) or a deeper CNN to get more features extraction, the validation accuracy could also be improved by further reducing the overfitting .

IMPLEMENTATION

- CNN Architecture: 10 Layers



The 1st part of the CNN architecture is composed of 4 parts having similar Layers: Conv2D + Batch_Normalization + MaxPooling. The goal of this 1st part is to extract the features from the image. The longer the network, the more features can be extracted.
The 2nd part of the CNN architecture is dedicated to the classifier. It is composed of a flattening layer which converts its input into a single dimension. Itself followed by the 2 dense fully connected layers.
The very last layer has 7 classes only to match the numbers of emotions to be identified.
In order to improve the accuracy, the CNN was tuned using Keras_tuner to better adapt its layers input/outputs hyper-parameters.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 16)	416
conv2d_1 (Conv2D)	(None, 48, 48, 16)	2320
batch_normalization (BatchNormalization)	(None, 48, 48, 16)	64
max_pooling2d (MaxPooling2D)	(None, 24, 24, 16)	0
conv2d_2 (Conv2D)	(None, 24, 24, 14)	2030
conv2d_3 (Conv2D)	(None, 24, 24, 14)	1778
batch_normalization_1 (BatchNormalization)	(None, 24, 24, 14)	56
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 14)	0
conv2d_4 (Conv2D)	(None, 12, 12, 56)	7112
conv2d_5 (Conv2D)	(None, 12, 12, 56)	28280
conv2d_6 (Conv2D)	(None, 12, 12, 56)	28280
batch_normalization_2 (BatchNormalization)	(None, 12, 12, 56)	224
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 56)	0
conv2d_7 (Conv2D)	(None, 6, 6, 252)	127260
conv2d_8 (Conv2D)	(None, 6, 6, 252)	571788
conv2d_9 (Conv2D)	(None, 6, 6, 252)	571788
max_pooling2d_3 (MaxPooling2D)	(None, 3, 3, 252)	0
flatten (Flatten)	(None, 2268)	0
dense (Dense)	(None, 343)	778267
dense_1 (Dense)	(None, 343)	117992
dense_2 (Dense)	(None, 7)	2408

Total params: 2,240,063
Trainable params: 2,239,891
Non-trainable params: 172

- Pre-Processing: CROP + BW

TRAINING/TESTING: The original pictures are 48,48 in black and white. Using the Dlib library and OpenCV, and in order to improve the training set, the dataset was filtered on whether the images had a detected face on it. If no face was detected, or the person's pose in the picture did not allow for a face detection, the picture was rejected from the training set. If a face was detected, the image was also cropped to best fit the face of the person, and to make sure the background would not be a source of noise during the training. Grayscale with only 1 channel was used as image input to the CNN.

USER INPUT: The same pre-processing approach is then used in this application. While the user would pass an RGB unfocused image to the CNN, the pre-processing function turns it to black and white then crops the picture to fit the face.

NOTE: Any picture with no face or more than 1 face is rejected from evaluation in this current version.

CONCLUSION AND FUTURE IMPROVEMENTS

This project was not straighforward as several options of CNN architecture were tested prior selecting the current model (around 20). Among them, a simple KNN (K-NearestNeighbors) approach was considered, along with Pre-trained/ Pre-architectured models such as VGG16, ResNet50V2, MobileNetV2. But the drawback on reusing pre-trained models is that the weights (e.g imagenet) are not fit for the specific purpose of extracting the features of a human face and its related emotion traits. So, one of the main challenge faced while re-training an entire a pre-architectured CNN, was the limitation of computing power, and the significant training time necessary to update a CNN weights. For example, a large CNN such as VGG16 with 138,4 Millions trainable parameters, would take up a couple of hours to train on 28000 images. To reduce the training time, the dataset was then reduced to 4626 images. During this 1st phase, the best performing pre-architectured CNN was MobileNetV2 which only has 3.5Millions trainable parameters. But it only yielded a 47% accuracy on a reduced dataset of 4636 images. In the 2nd phase, a better trade-off between accuracy vs training time had to be made: in order to improve the accuracy, the entire dataset was necessary for the training but since training on this amount of images would take too long, a smaller CNN was required. The CNN in question is the one presented in this page. It only has over 2.23 Millions trainable parameters. Its tunning took 05h 37m 47sec while the training on 19774 images only took 1H45min with the computing resources available at that time.

To achieved this project a small and tuned CNN was required, along with a pre-processing step consisting grayscaling, detecting and cropping the faces. The main hyper-parameters which had an influence on the accuracy were: - the quality of the dataset (labelling, person pose, picture resolution). - The amount of the training images - The number of epoch - The batch size - The tuned CNN perceptrons.

Further improvement on this project are possible. A first option would be to try to improve the training accuracy by adding more layer to extract the finest features. To improve the validation accuracy, a technic would be to reduce the overfitting by adding dropout layers/ regulizers, batch_normalization layers (although already assessed on the current CNN during its tunning phase). One of the option is to re-train the MobileNetV2 on the entire dataset this time with more computing power. It seems also interesting to note that mixing datasets such as FER2013, JAFFE, CK+, etc... could lead to accuracy improvement.