Adrian Marinovich
Springboard Data Science Career Track
Capstone Project #1 - In-depth analysis
December 10, 2018

Project: Detection of smiles in images of faces

Larger second dataset

To make further data available for the in-depth analysis, another freely-available smiles dataset was obtained (https://github.com/hromi/SMILEsmileD/tree/master/SMILEs), derived from the set of cropped LFW face images described above. Upon review, these additional images did not appear to have been as carefully classified for presence of smile as the first dataset, and also appeared to contain a wider variability of smile expressions with greater range of nuance than seen in the first dataset, possibly indicating greater 'real world' relevance. This author (an adult male) manually reclassified or excluded 581 images out of a total of 13,163 images in the second dataset (after removal of 2 non-smile duplicates present in the smile subset). Of the 3688 images initially labeled as smiles, 70 were reclassified into the non-smile subset, while 266 of the 9475 images initially labeled as non-smiles were reclassified into the smile subset, giving a total of 336 reclassifications. There were also 245 images (165 from the smile subset and 80 from the non-smile subset) excluded from the analysis due to: a) having ambiguous expressions (neither clearly smile not clearly non-smile); b) not having two eyes and a mouth visible in the image; or c) having facial characteristics consistent with a young child. This resulted in a revised large, second analysis dataset with a total of 12,918 images, with 3,719 images in the smile subset, and 9,199 images in the non-smile subset. The images, as with the small, first dataset, are 64 x 64 pixels, in grayscale. This gives a D/N ratio of 1.1 (4,096/3719) at the outset for this dataset.

Smile definition

The definition of smile used during the review of this second, larger dataset was adapted from the Merriam-Webster dictionary definition (https://www.merriam-webster.com/dictionary/smile). In their definition, 'smile' as a noun is '*a facial expression in which the eyes brighten and the corners of the mouth curve slightly upward and which expresses especially amusement, pleasure, approval, or sometimes scorn*.' This definition has been modified here to remove reference to scorn, such that the smile as defined above should be seen to express amusement, pleasure, approval, encouragement, joy or good will, without scorn.

Attempts were made to balance the need to exclude smile-like expressions such as smirking, grimacing, and 'posed' smiles that lack apparent emotional expression, with the need to include a range of subtle smiles. Images excluded from the smile-labelled subset were either reclassigied to the non-smile group or excluded from the analysis as ambiguous. Both 'mid-laugh' smiles and closed-mouth smiles are included in the smiles subset.

Data splitting and balancing

Data from the small, 'first' dataset were loaded and processed as entire datasets for training and validation sets, with 1000 training and 203 validation observations. The smile and non-smile datasets were already balanced (600 and 603 observations, respectively).

A generator function was created for the large, 'second' dataset, in which data augmentation by image processing (rotation, shear and zoom), standardization using training set mean and standard deviation (compatible with StandardScaler algorithm), and balancing of smile and non-smile subsets were performed on 'chunks' of images of a specified size. These chunks either served as the full dataset sample for non-generator-receptive models, or as batches to be fed into generator-receptive models. Note that for training and validation, random sampling with replacement was performed in this generator function for the large dataset. For the hold-out set prediction, the set was obtained as a whole. The training, validation and hold-out (test) sets comprised 80%, 15% and 5% of the large dataset, consisting of 10334, 1938 and 646 images respectively.

Modeling

The analysis calls for supervised learning, as it is focused on classification of images previously labeled as smile or non-smile. A range of supervised learning methods were applied, to examine the degree to which increasing the level of model complexity would increase the accuracy of the model on a validation sets, and allow comparison of model performance on the small, first dataset *vs.* the large, second dataset.

Modeling began with decision trees and random forests. A support vector machine model with a radial basis function kernel (SVM-RBF) was then applied. A fully-connected deep neural network (DNN) model was created, consisting of 2 layers of 1000 and 200 nodes, respectively, prior to the output layer. Finally, a number of convolutional neural network (CNN) models were applied, with adjustments to the architecture of this type of network based on validation set accuracies. The fit_generator method of the Keras CNN modeling framework was used to receive image batched from the above-described generator function. A range of batch sizes were evaluated, with a size of 10 selected as optimal for the small set and 16 for the large set. The best-performing CNN in the validation phase for the large set was then applied to the large set's hold-out test subset. Analyses were performed using both the small (first) and large (second) datasets, as summarized Table 1. The second dataset was also used for hold-out test set prediction after CNN model tuning. The hold-set was not used during model development. The hold-out set was delivered in its entirety by turning off random sampling in the generator.

For the small dataset, manipulation of the Keras ImageDataGenerator image preprocessing/ data augmentation method revealed shear, zoom and rescale to have overall detrimental effects on CNN model performance. Rotation range of 10 degrees was added with some improvement in performance. Width shifting and ZCA whitening were also attempted, without notable effects. The CNN validation accuracies shown in Table 1 for the small dataset are with a rotation of +/- 5 degrees on training data, and with no preprocessing on the validation data. The large set's

image processing showed more tolerance for use of shear and zoom, and accordingly rotation and shear of +/- 10 degrees, and zoom of 0.95 to 1.05 was applied to the large set's training images for data augmentation, but not to the validation or hold-out images.

The VGG-like CNN model was adapted from the Keras documentation (https://keras.io/getting-started/sequential-model-guide/), and is an abbreviated version of the original model developed by the Oxford Visual Geometry Group (https://www.robots.ox.ac.uk/~vgg/). In addition to doubling the convolutional layers prior to pooling, this model has dropout layers not used in the other custom models here.

Results

Validation accuracies

As shown in Table 1 below, validation accuracies improved in a progression of models from decision tree, to random forest, to SVM-RBF, to fully connected DNN, and finally to CNN. Models produced higher validation accuracies for the small dataset compared to the large dataset, except for the final and best-performing VGG-like CNN model, which performed marginally better on the large dataset.

Loss and accuracy plotting for the VGG-like CNN model are shown in Figures 1 and 2, with training loss and accuracy shown in blue, and validation loss and accuracy shown in yellow. The plot show that the improvements in validation accuracy flatten out at approximately 10 epochs.

Hold-out (test) set accuracy

Using the model and weights of the 20-epoch VGG-like CNN model, prediction of smile in the hold-out (test) set resulted in an accuracy of 0.915.

As shown in Figure 3, the prediction contingency table shows a number of images in the 'false negative' and 'false positive' boxes that may justifiably fall into an 'ambiguous' category by another individual's labelling, or even be reclassified.

Conclusion

We demonstrate an abbreviated VGG-like CNN model that can produce approximately 91% accuracy on a hold-out set for smile detection in face images. This parsimonious model indicates the potential for the development of similar models in the rapid detection of human facial expressions that could be implemented in human-machine interfaces.

The superior performance of the VGG-like CNN model indicates the importance of doubled convolutional layers and drop-out layers in CNN models to allow greater generalizability, especially in the context of the large dataset, which appears to have greater nuance and variability of smile types as compared to the small dataset.

Future work may be focused on broader validation and further pruning of the dataset to reduce missclassification or ambiguous images. As an alternative, it may be worth creating a

three-class detector to allow for detection of ambiguous smile-like expressions, or a graded smile classification system by intensity and clarity of perceived intent.

Table 1. Validation set accuracy by model and dataset.

Model	Validation accuracy	
	Small set*	Large set**
Decision Tree Gini impurity criterion Max. depth = 4000	0.778	- - -
Decision Tree Entropy impurity criterion Max. depth = 4000	0.803	0.722 Train: 10,000 Valid: 2,000
Random forest Entropy impurity criterion Num. estimators = 500	0.906	0.855 Train: 10,000 Valid: 2,000
Support Vector Machine, RBF kernel	0.931	0.853 Train: 10,000 Valid: 2,000
Deep neural network Keras layers: dnn_clf = Sequential() dnn_clf.add(Dense(1000, input_dim=4096, activation='relu')) dnn_clf.add(Dense(200, activation='relu')) dnn_clf.add(Dense(1, activation='sigmoid')) Loss: binary cross-entropy Optimizer: 'sgd'	0.941 Epochs: 30 Batch size: 10 Train total: 30,000 Valid: 203	0.869 Epochs: 30 x 10,000 samples Batch size: 10 Train total: 300,000 Valid: 2,000
Convolutional neural network - Conv2D layers: 32, 320 Keras layers: cnn_clf2a = Sequential() cnn_clf2a.add(Conv2D(32, (3, 3), input_shape=(64, 64, 1), activation='relu')) cnn_clf2a.add(MaxPooling2D(pool_size=(2, 2))) cnn_clf2a.add(Conv2D(320, (3, 3), activation='relu')) cnn_clf2a.add(MaxPooling2D(pool_size=(2, 2))) cnn_clf2a.add(Flatten()) cnn_clf2a.add(Dense(units=128, activation='relu')) cnn_clf2a.add(Dense(units=1, activation='sigmoid')) Loss: binary cross-entropy Optimizer: 'adam'	0.956 Epochs: 30 Batch size: 10 Train total: 30,000 Valid: 203	- - -
Convolutional neural network - Conv2D layers: 64, 32 Keras layers: cnn_clf2f = Sequential() cnn_clf2f.add(Conv2D(64, (3, 3), input_shape=(64, 64, 1), activation='relu')) cnn_clf2f.add(MaxPooling2D(pool_size=(2, 2))) cnn_clf2f.add(Conv2D(32, (3, 3), activation='relu')) cnn_clf2f.add(MaxPooling2D(pool_size=(2, 2))) cnn_clf2f.add(Flatten()) cnn_clf2f.add(Dense(units=128, activation='relu')) cnn_clf2f.add(Dense(units=1, activation='sigmoid')) Loss: binary cross-entropy Optimizer: 'adam'	0.961 Epochs: 30 Batch size: 10 Train total: 30,000 Valid: 203	0.921 Epochs: 10 x 300 steps Batch size: 16 Train total: 48,000 Valid: 2,000

Convolutional neural network - Conv2D layers: 32, 32, 32 Keras layers: cnn_clf2a = Sequential() cnn_clf2a.add(Conv2D(32, (3, 3), input_shape=(64, 64, 1), activation='relu')) cnn_clf2a.add(MaxPooling2D(pool_size=(2, 2))) cnn_clf2a.add(Conv2D(32, (3, 3), activation='relu')) cnn_clf2a.add(MaxPooling2D(pool_size=(2, 2))) cnn_clf2a.add(Conv2D(32, (3, 3), activation='relu')) cnn_clf2a.add(MaxPooling2D(pool_size=(2, 2))) cnn_clf2a.add(Flatten()) cnn_clf2a.add(Dense(units=128, activation='relu')) cnn_clf2a.add(Dense(units=1, activation='sigmoid')) Loss: binary cross-entropy Optimizer: 'adam'	0.951 Epochs: 30 Batch size: 10 Train total: 30,000 Valid: 203	- - -
Convolutional neural network - Conv2D layers: 128, 64, 32 Keras layers: cnn_clf2e = Sequential() cnn_clf2e.add(Conv2D(128, (3, 3), input_shape=(64, 64, 1), activation='relu')) cnn_clf2e.add(MaxPooling2D(pool_size=(2, 2))) cnn_clf2e.add(Conv2D(64, (3, 3), activation='relu')) cnn_clf2e.add(MaxPooling2D(pool_size=(2, 2))) cnn_clf2e.add(Conv2D(32, (3, 3), activation='relu')) cnn_clf2e.add(MaxPooling2D(pool_size=(2, 2))) cnn_clf2e.add(Flatten()) cnn_clf2e.add(Dense(units=128, activation='relu')) cnn_clf2e.add(Dense(units=1, activation='sigmoid')) Loss: binary cross-entropy Optimizer: 'adam'	0.965 0.956 Epochs: 30 Batch size: 10 Train total: 30,000 Valid: 203 0.965 Epochs: 4 Batch size: 10 Train total: 4,000 Valid: 203	0.927 Epochs: 10 x 600 steps Batch size: 16 Train total: 96,000 Valid: 2,000
Convolutional neural network - VGG-like model Keras layers: cnn_clfvgg = Sequential() cnn_clfvgg.add(Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 1))) cnn_clfvgg.add(Conv2D(32, (3, 3), activation='relu')) cnn_clfvgg.add(MaxPooling2D(pool_size=(2, 2))) cnn_clfvgg.add(Dropout(0.25)) cnn_clfvgg.add(Conv2D(64, (3, 3), activation='relu')) cnn_clfvgg.add(Conv2D(64, (3, 3), activation='relu')) cnn_clfvgg.add(MaxPooling2D(pool_size=(2, 2))) cnn_clfvgg.add(Dropout(0.25)) cnn_clfvgg.add(Flatten()) cnn_clfvgg.add(Dense(256, activation='relu')) cnn_clfvgg.add(Dropout(0.5)) cnn_clfvgg.add(Dense(1, activation='sigmoid')) Loss: binary cross-entropy Optimizer: 'adam'	0.921 Epochs: 10 Batch size: 10 Train total: 10,000 Valid: 203	0.926 Epochs: 10 x 600 steps Batch size: 16 Train total: 96,000 Valid: 2,000 0.943 Epochs: 20 x 600 steps Batch size: 16 Train total: 192,000 Valid: 2,000

*Small, or first set: 1000 training and 203 validation images

**Large, or second set: see generator settings by model for training and validation sampling sizes

Figure 1. Loss and accuracy plots for VGG-like CNN model, first 10 epochs, large dataset.

Figure 2. Loss and accuracy plots for VGG-like CNN model, additional 10 epochs, large dataset. (Note change in scales, compared to Figure 1 above.)

Figure 3. Contingency table showing images from the hold-out set by prediction outcome using the 20-epoch VGG-like CNN model.

Project code
https://github.com/adriatic13/springboard/blob/master/dsct_capstone1/Adrian_Marinovich_Cap1_smiles_data_wrangling_large_set.ipynb

https://github.com/adriatic13/springboard/blob/master/dsct_capstone1/Adrian_Marinovich_Cap1_smiles_indepth.ipynb

References

Arigbabu, Olasimbo Ayodeji, et al. "Smile detection using hybrid face representation." Journal of Ambient Intelligence and Humanized Computing (2016): 1-12.

Huang GB, Mattar M, Berg T, Learned-Miller E (2007) Labeled faces in the wild: a database for studying face recognition in unconstrained environments. University of Massachusetts, Amherst, Technical Report.

https://github.com/hromi/SMILEsmileD/tree/master/SMILEs

https://keras.io/getting-started/sequential-model-guide/

https://www.robots.ox.ac.uk/~vgg/

https://www.merriam-webster.com/dictionary/smile