A Computer Vision Pipeline for Facial Emotion Recognition

Behzad Javaheri

Abstract

In this study, we performed facial emotion recognition using the "Real-world Affective Faces (RAF)" dataset. We compared the performance of standard vector machine (SVM), multilayer perceptron (MLP) and convolutional neural network (CNN). For the SVM and MLP, we extracted 4 features of ORB, SIFT, HOG and utilised flattened pixels. To identify optimal hyperparameters, a search for both SVM and MLP was conducted. In addition, to overcome the initial low performance of CNN architecture, data augmentation was performed. Our data show that MLP is the best-performing algorithm for emotion classification in this study. The lowest-performing was CNN, modification to parameters and implementation of different architecture did not enhance performance. Exhaustive parameter tuning may result in greater CNN performance on this particular dataset. We find that HOG and flattened pixels allow better classification and ORB and SIFT to lead to low facial emotion recognition accuracy.

Introduction and motivation

Emotion recognition is an area of research receiving increasing attention due to its application in several areas including clinical settings, lie detection, and entertainment. Although the task of emotion recognition is a trivial task for humans, however, this is a significant challenge in machine learning (Badshah et al., 2017).

Emotion recognition aims to interpret a person's perspective and state expressed through speech, facial expression and body language. For humans building, this ability is essential for the healthy development of interpersonal relationships (Niedenthal and Brauer, 2012). It is therefore an area of interest for psychologists as well as developmental researchers. The ability to successfully recognise one's emotion can indeed be used as a predictor for empathy (Gery et al., 2009), social behaviour (Marsh et al., 2007), and in building emotional intelligence (Salovey and Mayer, 1990).

Thus, any impairment in emotion recognition leads to significant lifelong consequences in building relationships, forming and describing emotions and impairment in social functioning (Batty and Taylor, 2006). Additionally, impairment in this area may also be associated with mental disorders (Phillips et al., 2003, Blair, 2003). Taken together, these studies highlight the significant potential of research in this area and in using recent development in machine learning to address unmet needs.

Whilst the vast majority of humans naturally build emotion recognition capabilities for immediate emotion recognition without any issue, such skill needs to be taught to machines often with significant challenges (Brave and Nass, 2009).

This project particularly focuses on facial emotion recognition and aims to build a pipeline implemented using Python programming language utilising "Real-world Affective Faces (RAF)" Database for accurate emotion classification (Li et al., 2017).

Description of the pipeline

The emotion recognition pipeline typically involves providing images as input for initial face detection, followed by feature extraction that will subsequently be used by the desired machine learning algorithm for classification. For traditional machine learning algorithms, these features need to be extracted before model construction. In more advanced algorithms including convolutional neural network (CNN), the algorithm acts as both a feature extractor and a classifier.

For this project, we, therefore, need to select desired features for extraction as well as to briefly define the type of machine learning algorithms.

Feature extraction

There are several features reported that can potentially be extracted from input images. For example, Lowe et al., (2004) developed Scale Invariant Feature Transform (SIFT) aiming to solve intensity, viewpoint changes and image rotation in feature matching. SIFT allows estimation of scale-space extrema followed by keypoint localisation, orientation and subsequently computation of local image descriptor for each key point (Lowe, 2004). Moreover, SIFT offers efficient recognition of objects in a given image, however, its implementation is computationally expensive and time-consuming.

SIFT has inspired the development of other variants in particular to overcome the complexity and computational demand. A variant is Speed up Robust Feature (SURF), which reportedly outperforms SIFT without negative impact on the robustness of detected points and overall model accuracy (Bay et al., 2008).

Another alternative to SIFT is Robust Independent Elementary Features (BRIEF) offering similar performance with significantly less complexity (Calonder et al., 2010). Furthermore, Rublee et al., (2011) reported that Oriented FAST and Rotated BRIEF (ORB) provides more efficient performance (Rublee et al., 2011).

Additionally, Histogram of Oriented Gradients (HOG) HOG is utilised in various tasks including object detection allowing the occurrence of edge orientations to be measured. In principle, this descriptor is representative of a local statistic of the orientations for the image gradients for a key point. Simply, each descriptor can be considered as a collection of histograms that are composed of pixel orientations assigned by their gradients (Monzo et al., 2011).

For this project, SIFT, ORB and HOG descriptor will be extracted to use for emotion recognition. In addition to these, flattened pixels will also be utilised.

Emotion recognition classifiers

Herein, we aim to implement standard vector machine (SVM), multilayer perceptron (MLP) and convolutional neural network (CNN) and compare their performance.

SVM is a supervised learning algorithm that allows regression and classification to produce good generalisation on unseen data. SVM builds a hyperplane leading to maximising the margin between classes (Auria and Moro, 2008).

SVM is considered to have a shallow architecture with a two-layer neural network utilising a hidden layer of radial units and one neuron output. Its inner workings do not allow significant manipulation of its parameters and the main adjustment is achieved by changing kernel functions (Osowski et al., 2004). This lack of flexibility and significant reliance on kernel function is one of the main disadvantages of SVM. Additionally, SVMs particularly for hyperparameter tuning require significant computation (Feng et al., 2020, Hesami et al., 2020).

MLP is also used for a variety of tasks including classification and regression (Pal and Mitra, 1992, Rumelhart et al., 1986). In this architecture, neurons are arranged in layers, starting with the input layer, to function through hidden layers and to produce output in the output layer. MLP is a feedforward algorithm that allows the processing signal to propagate from the input layer to the output layer. One of the main disadvantages of MLP is that it generally overfitting particularly to small datasets (Geetha, 2020, Ghorbani et al., 2016).

Convolutional Neural Network (CNN) is a learning algorithm primarily used in pattern recognition in images. It's a deep learning algorithm taking a given image as input and through learnable weight/biases will assign importance to aspects of the image that can identify. The main advantage is that it requires significantly less pre-processing compared to other algorithms and produces higher performance (O'Shea and Nash, 2015).

Description of the dataset

A subset of the Real-world Affective Faces (RAF) Database was utilised in this study. The original database contains ~30,000 images that were obtained from the Internet. Crowdsourcing was used for annotations leading to each image receiving 40 independent annotations. The images cover a diverse set of participants, age, gender, ethnicity as well as variations in image properties including lighting. The subset provided for this study contains training (12271) and a testing dataset (3068). The images are in jpg format and the emotions expressed include surprise, fear, disgust, happiness, sadness, anger, neutral.

Hypothesis

That the CNN will result in higher performance than SVM and MLP in emotion recognition.

Methodology

Initially, the label of each image in both training and testing datasets were extracted with the path to the corresponding image. Subsequently, a for loop was defined to extract pixels, flattened pixels, SIFT, HOG and ORB. For SIFT and ORB features, we observed that 7 and 251 images in the training dataset failed to produce these descriptors, respectively. In addition, 69 images did not produce ORB in the testing datasets. Attempts were made to explore the reason for this partial failure; however, this did not fall within the scope of this study and in the interest of time it was not pursued. The following snippet demonstrates how this was implemented:

```
if des_orb is None:
        pass
else:
        train_orb.append(des_orb[0].tolist())
        y_train_orb.append(y_train[indiv_img])
```

Additionally, to avoid a mismatch with the labels the corresponding labels were extracted for each feature. Moreover, the emotion target is multiclass and as such encoded using LabelEncoder and the shape of the dataset checked to ensure appropriate shape for training. Analysis of data distribution suggested for the extracted features suggested that scaling was required. Sklearn's StandardScaler was used to scale the features before hyperparameter tuning.

To identify optimal hyperparameters and exhaustive search using Sklearn's GridSearchCV for both SVM and MLP were implemented. This approach, however, did not succeed after running for multiple days both on a local machine and Google Colab pro. Alternatively, Sklearn's RndomizedSearchCV was utilised to identify optimal parameters.

This fine-tuning for SVM and MLP was performed using SIFT, HOG, ORB and flattened pixels. Therefore, a total of 8 successful searches was performed. The performance of the search was visualised using a scatter 3D plot.

Choice of parameters

For SVM, RandomizedSearchCV was performed for kernel (rbf, poly and linear), C (uniform distribution 2, 10), gamma (uniform distribution 0.01, 1) and 5 StratifiedKfold CV. For MLP the grid included search for hidden_layer_size [(8,), (180,), (300,), (100,50,), (10,10,10)], activation ['tanh', 'relu', 'logistic'], solver ['sgd', 'adam'], alpha [0.0001, 0.001, 0.01], epsilon [1e-08, 0.1], and learning rate ['adaptive', 'constant']. Accuracy was used to identify the optimal parameters and the model was tested on the entire training dataset. Subsequently performance on the testing dataset was evaluated.

Attempts were made to visualise and compare the performance of these models, cross-validation score, learning curve, classification report, class prediction error, precision-recall, ROC/AUC graph and confusion

matrix were plotted using Yellowbrick library. These, however, did not complete due to computation. The code is present within the notebook.

Results, analysis and critical evaluation:

The computation time for the SVM hyperparameter search using ORB was 111.9 seconds with 1.1 for C, 0.46 gamma and linear as the kernel. The SVM performance was 38.7 and 38.5% on the entire training and testing datasets.

In addition, the computation time was 3398.9 for SVM SIFT hyperparameter search with 1.52 for C, 0.66 gammas and rbf as the kernel. The computation time to fit the model was 228.3 seconds with 100 and 38% accuracy on the entire training and testing dataset. This suggests overfitting to training dataset and poor generalisation to unseen test subset.

Furthermore, we observed a computation time of 2875 seconds for HOG SVM parameter search with 3.89 for C, 0.34 gamma and linear as the kernel. The computation time to fit the model was 896.3 seconds with 100 and 66.3% accuracy on the entire training and testing dataset. This suggests overfitting to training dataset and better generalisation than those observed for ORB and SIFT.

In addition, it took 6326.9 seconds for parameter search for SVM flattened pixels to complete. The suggested parameters were 8.3 for C, 0.58 for gamma and linear kernel. Subsequent fitting to the testing dataset took 5183.2 seconds with 100 and 58.3% accuracy for training and testing dataset.

Taken together, these data suggest that HOG produced the highest performance using an SVM classifier with ORB as the least performing descriptor.

We next performed hyperparameter tuning for MLP using our 4 descriptors. Our data suggest that for MLP using ORB to utilise sgd as the solver, adaptive learning rate and (100, 50) hidden layer size, epsilon of 1e-8, alpha of 0.01 and activation function of logistic with a computation time of 312.3s. The performance of optimal MLP using ORB showed an accuracy of 38.72 and 38.51% for the training and testing dataset respectively with a computation time of 21.6 seconds.

A similar search was performed for SIFT suggested utilising sgd as the solver, constant learning rate and (100, 50) hidden layer size, epsilon of 1e-8, alpha of 0.001 and activation function of logistic with a computation time of 800.6s. The performance of optimal MLP using SIFT showed an accuracy of 38.9 and 38.62% for the training and testing dataset respectively with a computation time of 85.9 seconds.

Furthermore, our parameter search for HOG indicates to utilise adam as the solver, adaptive learning rate and (300) hidden layer size, epsilon of 0.1, alpha of 0.0001 and activation function of logistic with a computation time of 6587.4s. The performance of optimal MLP using HOG showed an accuracy of 79.1 and 72.9% for the training and testing dataset respectively with a computation time of 1208.7 seconds.

In addition, fine-tuning of parameters for flattened pixels suggested using sgd as the solver, constant learning rate and (180) hidden layer size, epsilon of 0.1, alpha of 0.01 and activation function of logistic with a computation time of 17178.8s. The performance of optimal MLP using flattened pixels showed an accuracy of 72.1 and 66.1% for the training and testing dataset respectively with a computation time of 1236.8 seconds.

We next constructed our CNN model using Keras. The following output represents the model summary:

Layer (type)	Output Shape	Param #
conv2d_366 (Conv2D)	(None, 96, 96, 16)	416
batch_normalization_346 (Bat	(None, 96, 96, 16)	64
max_pooling2d_212 (MaxPoolin	(None, 48, 48, 16)	0
dropout_332 (Dropout)	(None, 48, 48, 16)	0
conv2d_367 (Conv2D)	(None, 44, 44, 32)	12832
batch_normalization_347 (Bat	(None, 44, 44, 32)	128
max_pooling2d_213 (MaxPoolin	(None, 22, 22, 32)	0
dropout_333 (Dropout)	(None, 22, 22, 32)	0
conv2d_368 (Conv2D)	(None, 18, 18, 64)	51264
batch_normalization_348 (Bat	(None, 18, 18, 64)	256
max_pooling2d_214 (MaxPoolin	(None, 9, 9, 64)	0
dropout_334 (Dropout)	(None, 9, 9, 64)	0
conv2d_369 (Conv2D)	(None, 5, 5, 128)	204928
batch_normalization_349 (Bat	(None, 5, 5, 128)	512
max_pooling2d_215 (MaxPoolin	(None, 2, 2, 128)	0
dropout_335 (Dropout)	(None, 2, 2, 128)	0
flatten_63 (Flatten)	(None, 512)	0
dense_170 (Dense)	(None, 256)	131328
dropout_336 (Dropout)	(None, 256)	0
dense_171 (Dense)	(None, 7)	1799

Total params: 403,527
Trainable params: 403,047
Non-trainable params: 480

This is the summary of our CNN model None

The implementation of this model on the image pixels over 25 episodes resulted in ~38% accuracy for training and validation subsets. Whilst an improvement of accuracy compared to initial episodes were observed, nevertheless, this accuracy indicate the low performance of our CNN architecture.

The performance on the unseen test data was 38%. Variation to parameters was tested and the model run close to 100 times however this did not improve performance. In addition, pixel and label structure and shape were thoroughly checked to ensure correct data shape as input.

To enhance CNN performance, we conducted data augmentation using Keras ImagedataGenerator. This approach did not improve the final performance, although the accuracy for both training and validation was higher in initial episodes compared to prior augmentation.

Conclusions

Our data show that MLP is the best-performing algorithm for emotion classification in this study. The lowest-performing algorithm was CNN, variation to parameters and implementation of different architecture did not enhance performance. It is therefore possible that hyperparameter tuning may enhance performance. Keras tuner is such a tool, however, its implementation was hampered in this study due to low computation power.

Furthermore, both MLP and SVM performed better using HOG and flattened pixels. The low performance of ORB and SIFT in this task may be specific to this particular dataset and if repeated using other datasets, it will indicate that these two descriptors are poor predictors of facial emotion classification.

Lessons learned

The algorithms implemented in this study are powerful classifiers producing significantly different performance. For SVM and MLP exhaustive grid search may allow better hyperparameter tuning. This, however, requires sufficient computational resources and as such exploiting cluster computing may help with this shortcoming. In addition, CNN can be tuned, for example using Keras tuner. Whilst this was explored, its implementation was not reported as various attempt did not successfully result in completion.

Future work

Building a multimodal architecture reported previously using a combined implementation of CNN and recurrent neural network (Vinyals et al., 2015) will allow the production of the description of the images, i.e., the emotion to be automated and streamlined and as such may result in higher performance.

References

AURIA, L. & MORO, R. A. 2008. Support vector machines (SVM) as a technique for solvency analysis.

BADSHAH, A. M., AHMAD, J., RAHIM, N. & BAIK, S. W. Speech emotion recognition from spectrograms with deep convolutional neural network. 2017 international conference on platform technology and service (PlatCon), 2017. IEEE, 1-5.

BATTY, M. & TAYLOR, M. J. 2006. The development of emotional face processing during childhood. *Developmental science,* 9**,** 207-220.

BAY, H., ESS, A., TUYTELAARS, T. & VAN GOOL, L. 2008. Speeded-up robust features (SURF). *Computer vision and image understanding,* 110**,** 346-359.

BLAIR, R. 2003. Facial expressions, their communicatory functions and neuro–cognitive substrates. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences,* 358**,** 561-572.

BRAVE, S. & NASS, C. 2009. Emotion in human–computer interaction. *Human-computer interaction fundamentals.* CRC Press Boca Raton, FL, USA.

CALONDER, M., LEPETIT, V., STRECHA, C. & FUA, P. Brief: Binary robust independent elementary features. European conference on computer vision, 2010. Springer, 778-792.

FENG, L., ZHANG, Z., MA, Y., DU, Q., WILLIAMS, P., DREWRY, J. & LUCK, B. 2020. Alfalfa Yield Prediction Using UAV-Based Hyperspectral Imagery and Ensemble Learning. *Remote Sensing,* 12**,** 2028.

GEETHA, M. 2020. Forecasting the Crop Yield Production in Trichy District Using Fuzzy C-Means Algorithm and Multilayer Perceptron (MLP). *International Journal of Knowledge and Systems Science (IJKSS),* 11**,** 83-98.

GERY, I., MILJKOVITCH, R., BERTHOZ, S. & SOUSSIGNAN, R. 2009. Empathy and recognition of facial expressions of emotion in sex offenders, non-sex offenders and normal controls. *Psychiatry research,* 165**,** 252-262.

GHORBANI, M. A., ZADEH, H. A., ISAZADEH, M. & TERZI, O. 2016. A comparative study of artificial neural network (MLP, RBF) and support vector machine models for river flow prediction. *Environmental Earth Sciences,* 75**,** 476.

HESAMI, M., NADERI, R., TOHIDFAR, M. & YOOSEFZADEH-NAJAFABADI, M. 2020. Development of support vector machine-based model and comparative analysis with artificial neural network for modeling the plant tissue culture procedures: effect of plant growth regulators on somatic embryogenesis of chrysanthemum, as a case study. *Plant Methods,* 16**,** 1-15.

LI, S., DENG, W. & DU, J. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017. 2852-2861.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision,* 60**,** 91-110.

MARSH, A. A., KOZAK, M. N. & AMBADY, N. 2007. Accurate identification of fear facial expressions predicts prosocial behavior. *Emotion,* 7**,** 239.

MONZO, D., ALBIOL, A., SASTRE, J. & ALBIOL, A. 2011. Precise eye localization using HOG descriptors. *Machine Vision and Applications,* 22**,** 471-480.

NIEDENTHAL, P. M. & BRAUER, M. 2012. Social functionality of human emotion. *Annual review of psychology,* 63**,** 259-285.

O'SHEA, K. & NASH, R. 2015. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

OSOWSKI, S., SIWEK, K. & MARKIEWICZ, T. Mlp and svm networks-a comparative study. Proceedings of the 6th Nordic Signal Processing Symposium, 2004. NORSIG 2004., 2004. IEEE, 37-40.

PAL, S. K. & MITRA, S. 1992. Multilayer perceptron, fuzzy sets, classifiaction.

PHILLIPS, M. L., DREVETS, W. C., RAUCH, S. L. & LANE, R. 2003. Neurobiology of emotion perception I: the neural basis of normal emotion perception. *Biological psychiatry,* 54**,** 504-514.

RUBLEE, E., RABAUD, V., KONOLIGE, K. & BRADSKI, G. ORB: An efficient alternative to SIFT or SURF. 2011 International conference on computer vision, 2011. Ieee, 2564-2571.

RUMELHART, D. E., HINTON, G. E. & WILLIAMS, R. J. 1986. Learning representations by back-propagating errors. *nature,* 323**,** 533-536.

SALOVEY, P. & MAYER, J. D. 1990. Emotional intelligence. *Imagination, cognition and personality,* 9**,** 185-211.

VINYALS, O., TOSHEV, A., BENGIO, S. & ERHAN, D. Show and tell: A neural image caption generator. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015. 3156-3164.