Applied Machine Learning Systems ELEC0132 Assignment

Maryam Habibollahi
Department of Electronic and Electrical Engineering
University College London
zceemha@ucl.ac.uk

Abstract

Machine learning algorithms have been implemented for the detection and classification of human faces with binary and multiclass attributes using the Python programming language with several image manipulation and machine learning libraries. A comparative analysis of the performance of the different techniques for face detection and classification provided a higher obtainable accuracy from neural network architectures, particularly the single-shot-multibox detector for accurate face detection, the multi-layer perceptron model for binary classification tasks, with accuracies above 0.8, and the LeNet convolutional neural network architecture for multiclass classification, providing an accuracy of 0.81 on the hair colour classification task. The code and all results can be accessed via Google Drive or directly from the GitHub repository.

I. Introduction

The perception of visual information is a key element of human communication, particularly those from the face. The features and characteristics of an individual's face can provide information about their identity, emotion, and intent, with potential applications in access and security, law enforcement, marketing, and banking. Researchers in the fields of computer vision have been developing technological breakthroughs in the implementation of face recognition using machine learning tools and techniques over the past decades. The variations in real-world images from illumination, pose, expressions, and occlusions have required more complex learning tools to provide adequate predictions for a variety of images. A suitable balance between a model's accuracy and complexity is often required to meet the computational power limitations and performance requirements of a system.

This assignment aims to train machine learning models and perform binary and multiclass classification on a large dataset of 5000 Portable Network Graphic (PNG) image files consisting of pre-processed subsets from the "CelebFaces Attributes Dataset (CelebA)", a celebrity image dataset [1] , and the "Cartoon Set", an image dataset of random cartoons/avatars [2], as well as a number of noisy images (mainly of natural backgrounds) to be detected and removed from the training data, containing 80% of the entire dataset. All images include five labels; a multiclass hair colour label, and four binary labels classifying human/non-human subjects, whether the subject is young, wearing glasses, and smiling.

In order to train a suitable model for the required classification tasks, several preprocessing methods were taken into consideration to provide appropriate features for the process; for instance, facial landmarks were extracted upon face detection to train models using supervised learning algorithms such as Support Vector Machines (SVM) or Multi-Layer Perceptron (MLP) models based on the extracted facial landmarks. A comparative analysis of the performance of each method with respect to the labelled noisy images facilitated the selection of the most appropriate feature extraction method for the given dataset.

Prior to the extraction of features, various preprocessing techniques were carried out on the images to improve both the performance and processing power during the later stages of the extraction, training and classification procedures. Examples of such techniques include colour space transformation, capable of significantly reducing processing complexity, gamma correction (power-law equalisation), a non-linear function used to normalise illumination by raising the input value to the power γ, as shown in Equation 1, and mean normalisation.

$$V_{out} = AV_{in}^{\gamma} \text{ , where A = 1 in the common case} \quad (1)$$

The original dataset was otherwise rescaled and augmented to avoid overfitting for alternative models more specifically used for visual recognition tasks, such as Convolutional Neural Networks (CNN), where the noisy images of the training and validation data are removed using the results of the optimum face detector method with the maximum accuracy.

II. Proposed algorithms

A major step of the extraction of facial information for various classification tasks such as age, gender, emotion, and other attributes apparent on the face is to localise the fiducial facial key points [3]. The landmarks provide a set of x and y coordinates that either describe the specific points that describe a unique location of a particular component, or lay out the contours connecting those points, such as those shown in Figure 1. Several algorithms have been developed to achieve this purpose, namely the Haar cascade classifier, the first real-time face detector proposed in 2001, the Histogram of

Oriented Gradients feature with a linear classifier, and various Deep Learning-based detectors, which are significantly more accurate than the former two methods, though at a cost of higher complexity.

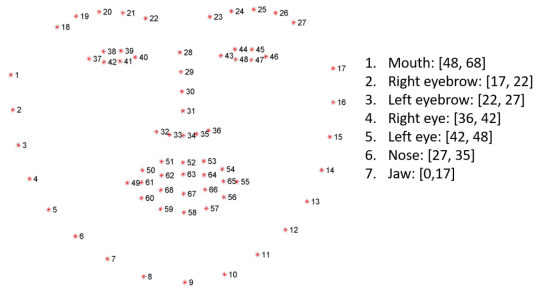

Fig. 1. Face shape defined by 68 landmarks

One such Deep Learning-based face detector based on Deep Neural Networks (DNN) is known as the Single-Shot-Multibox Detector (SSD), which uses the ResNet-10 architecture as a backbone. This method is ideal for a dataset with various orientations and considerable occlusion in the images; however, the complexity of the model requires a longer amount of time to carry out the detection. Another detector with Convolutional Neural Network (CNN) features is called the Maximum-Margin Object Detector (MMOD), which has a simple training process; though also more complex than the former two detectors. The relative balance between the expected accuracy and complexity of a HOG detector with respect to the Haar cascade and Deep Learning options makes it preferable for this dataset. Nonetheless, a comparative analysis was performed by recording the accuracy and training time of each detector as a measure of performance and complexity.

Classification of the binary tasks was performed using the landmark features to train models with supervised learning algorithms, namely Support-Vector Machines (SVM), which are capable of linearly separating classes in a high-dimensional space through the implementation of different kernel functions. The hyperplane which isolates one class from another can be refined via gradient descent, an iterative optimisation algorithm, such as that shown in Equation 2, which represents the gradient in linear regression for a model of n data points and m features. This technique is used to minimise a parameter called the cost function, which represents an attribute of the error in the response, such as the squared sum of residuals.

$$\theta_{j+1} = \theta_j - \frac{\alpha}{n}\sum_{i=1}^{n}\left[\sum_{k=1}^{m}\theta_k x_k^{(i)} - y^{(i)}\right]x_j^{(i)} \tag{2}$$

Higher-complexity models based on artificial neural networks that are capable of providing higher accuracies were also implemented using the features extracted. A Multi-Layer Perceptron (MLP) model, which carries out the training using Backpropagation, an efficient method that computes the partial derivates in gradient descent, was therefore selected for the binary tasks. The derivatives are calculated from each layer's error term, δ_i^l, which is computed using Equation 3, for $a^{(l)}$ representing the activation vector of layer l, resulting in the output $z^{(l)} = \theta^{(l)}a^{(l)}$ for that layer.

$$\delta^{(l)} = \left(\theta^{(l)}\right)^T \delta^{(l+1)}.g'(z^{(l)}) \tag{3}$$

Despite the minimised processing requirement when the landmark features are used, they pose limitations to classification tasks less reliant on the key component locations, and requiring important information omitted from the images such as colour. Thus, for the final task of detecting hair colour, a more commonly-used classifier for image processing with multilayer neural networks called Convolutional Neural Networks (CNN) was implemented on a LeNet architecture. The popularity of CNNs in image classification is primarily due to the 3D volumes of neurons, resulting in connectivities of small regions between layers (known as the receptive field), which can result in a lower complexity than the traditional neural networks, while taking advantage of 3-dimensional images. The LeNet architecture is a small yet powerful tool for image classification using CNN. Primarily used for Optical Character Recognition (OCR), LeNet implements a 7-level convolutional network composed of convolutional layers, (ReLU) activation and pooling layers, as illustrated in Figure 2.

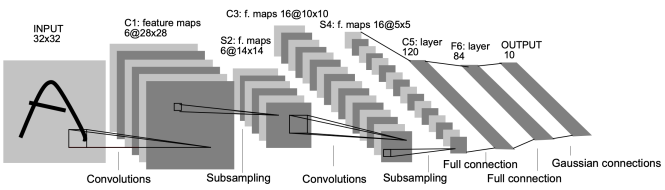

Fig. 2. Architecture of LeNet-5 by LeCun et al. [4]

Increasing the number of layers in a neural network is often believed to provide a better model, given the higher complexity. However, a model can be easily overfit to the training set if the parameters follow the data too closely. In order to obtain a more generalised model of the data, cross-validation was selected to perform out-of-sample testing on the dataset.

III. Implementation

The primary image manipulation and pre-processing tasks were performed with the aid of the Open Source Computer Vision (OpenCV) and dlib libraries, which are widely used in image processing. The implemented functions range from colour space transformation to face detection and landmark prediction for the binary classification tasks. Mathematical manipulation and analysis as well as file handling were mainly carried out using the NumPy and Pandas software libraries.

As for the implementation of machine learning algorithms, such as SVM, MLP, and the corresponding tools to carry out cross-validation and obtain the confusion matrix, the Scikit-learn library was employed to mainly carry out binary

classification tasks. Likewise, Keras was implemented for running neural network algorithms, specifically the Convolutional Neural Networks, enabling fast experimentation on a large dataset. Finally, the Matplotlib plotting library was included to provide visual outputs on the data, such as learning curves, which outline the validation and test accuracies over a range of training samples, such as that shown in Figure 3.

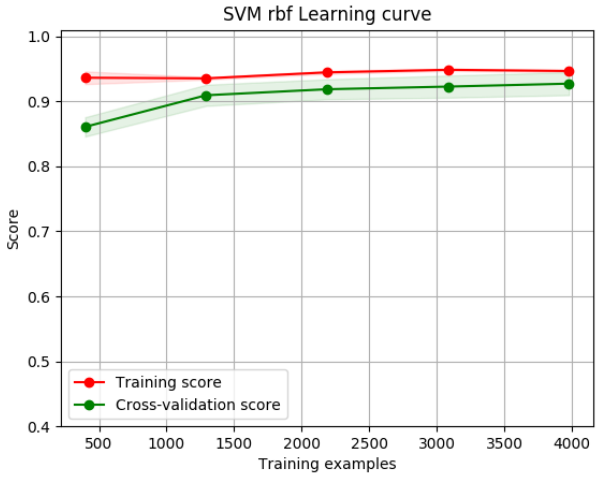

Fig. 3. Learning curve of emotion recognition

Each classification task was performed using various functions and parameters. For instance, the SVM was implemented with different Kernels to find the optimum performance of binary classification, such as linear, polynomial (`poly`), radial basis function (`rbf`), and sigmoid. A sweep of the penalty parameter C was carried out to observe the effects of parameter variations. In a similar manner, variations in the number of hidden layers, regularisation parameter alpha, and the solver were applied to achieve the optimum response. The log-loss function of the MLP is optimised using the lbfgs or stochastic gradient descent (`sgd`).

IV. Experimental Result

As previously explained, the HOG face detector is expected to provide an accuracy higher than that of the Haar Cascade and lower than that of Deep Learning methods. To observe the trade-offs between the performance and complexity of these models, a comparison of the four was carried out with the same 4,000 training data and 1,000 test data for all detectors. Table I lays out a comparative view of each method based on the overall accuracy, the true positive rate (TPR), true negative rate (TNR), false positive rate (FPR), false negative rate (FNR), and the time taken in minutes to train and carry out the predictions.

The performance of the binary model classifiers were analysed and compared via the calculation of the model accuracy, recall (true positive rate), specificity (true negative rate), and precision based on its respective confusion matrix. As shown in table II, the MLP classifier with lbfgs solver was often found to be the optimum solution for the binary tasks, as the number

TABLE I
Comparative Measures of Face Detector Scores

Method	Accuracy	TPR	TNR	FPR	FNR	Time
Haar Cascade	0.78	0.77	0.40	0.03	0.72	1.88
HOG with SVM	0.97	0.97	1.00	0.00	0.33	2.79
SSD (DNN)	0.99	0.98	1.00	0.00	0.14	4.55
MMOD (CNN)	0.98	0.98	1.00	0.00	0.18	10.85

of hidden layers were maintained low to avoid overfitting to the training data. Carrying out rotation estimation on the data showed an instance of overfitting for the MLP classifier, as the overall score obtained from a 10-fold cross-validation on task 2 was measured to be 0.79, whereas the SVM classifier with the RBF kernel provided an accuracy as high as 0.84.

TABLE II
Performance Measures of Binary Classifiers

Task	Method	Accuracy	Recall	Specificity	Precision
1	MLP (lbfgs)	0.93	0.95	0.86	0.96
2	MLP (lbfgs)	0.79	0.81	0.60	0.96
3	MLP (lbfgs)	0.86	0.81	0.88	0.66
4	SVM (poly)	0.98	0.98	0.97	0.96

The SVM and MLP classifiers were initially trained on the raw image dataset (with lowered image sizes converted to arrays) to carry out multiclass classification on hair colour for the fifth and final task. The linearised formatted of the feature set resulted in a higher complexity with inadequate test scores. Hence, a CNN with the LeNet architecture was implemented, providing a more acceptable test accuracy, as illustrated in Table III. The training data of this task was divided in two stages for this task in order to provide sufficient validation data for backpropagation.

TABLE III
Performance Measures of Multiclass Classifiers

Method	Accuracy	Time
SVM with Sigmoid Kernel	0.14	1.70
SVM with RBF Kernel	0.30	2.37
MLP with SGD solver	0.49	0.38
MLP with LBFGS solver	0.60	0.20
CNN with LeNet architecture	0.81	7.85

As shown in the plot of the training and validation accuracies of the LeNet model throughout the backpropagation process in Figure 4, the model achieved a maximum score of 0.81 after 25 epochs. This is considerably higher than the achieved accuracies of the SVM and MLP model classifiers.

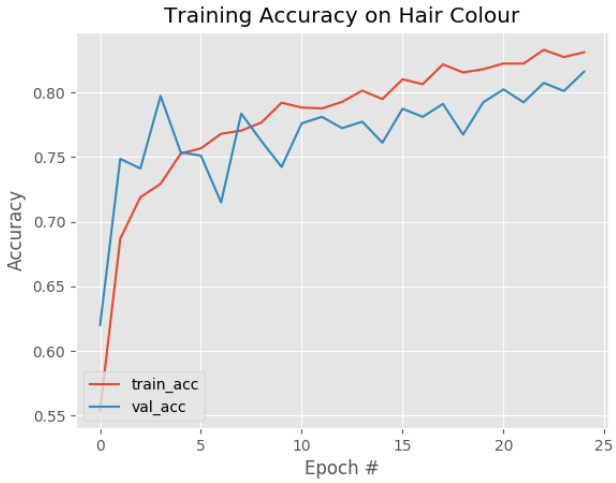

Fig. 4. LeNet training and validation accuracy variation

V. Conclusion

A comparative analysis of several methods of face detection and image classification was carried out on a labelled dataset of 5,000 images using binary and multiclass classifiers with support vector machines and neural networks via OpenCV, dlib, Keras, and Scikit-learn Python libraries for image processing and machine learning. An analysis of the performance and complexity of the models was carried out using cross-validation for a non-biased response and confusion matrices to obtain the statistics of true predictions.

The multi-layer perceptron neural network was generally found to provide more accurate binary classification models, except for particular test data on glasses detection (task 3), where the model was overfit. It can therefore be concluded that the higher complexity of neural networks does not necessarily ensure higher prediction scores.

Implementation of the LeNet CNN architecture provided considerably more accurate predictions for the multiclass classification task of hair colour classification. However, higher accuracies can be obtained via the implementation of more advanced convolutional neural networks such as AlexNet, a larger model which includes max pooling, ReLU nonlinearity, and dropout regularisation.

VI. Related Work

A number of CNN frameworks have been implemented for image classification since the introduction of the LeNet architecture in 1998. Some of the more recent frameworks include GoogLeNet with 19 layers, capable of achieving 6.67% error with the Inception Module, and ResNet with 152 layers and an error as low as 3.57% on the ImageNet dataset [5]. The Inception Module in both architectures implements parallel paths of various receptive field dimensions and operations in order to capture sparse correlation patterns, as illustrated in Figure 5.

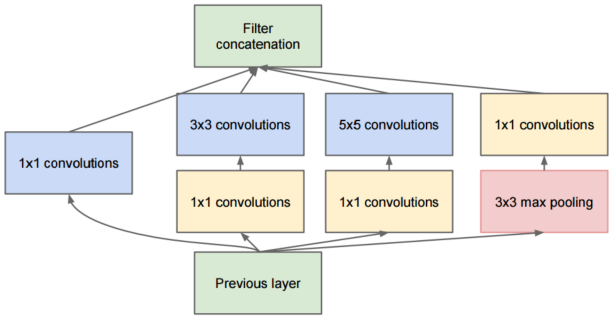

Fig. 5. Implementation of the Inception Module [6]

Improved variants of the Inception reduces the effects of auxiliary classifiers via the implementation of batch normalisation in order to provide regularised training [7]. The factorised convolutions and aggressive regularisation of these variants result in more efficient computation in the network.

References

[1] S. Yang, P. Luo, C. C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 International Conference on Computer Vision, ICCV 2015, no. 3, pp. 3676–3684, 2015.

[2] F. Cole, I. Mosseri, D. Krishnan, A. Sarna, A. Maschinot, B. Freeman, and S. Fuman, "CartoonSet." [Online]. Available: https://google.github.io/cartoonset/people.html

[3] Y. Wu and Q. Ji, "Facial Landmark Detection: A Literature Survey," *International Journal of Computer Vision*, pp. 1–28, 2018.

[4] Y. LeCun, L. Bottu, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *IEEE*, 1998.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 45, no. 8, pp. 1951–1954, 2006.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 1–9, 2015.

[7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2015. [Online]. Available: http://arxiv.org/abs/1512.00567