Applied Machine Learning Systems ELEC0132 Assignment

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

Abstract

Brief overview of the methodology/results presented.

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 of 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 image variations of real-world scenarios such as illumination, pose, expressions, and occlusions have required more complex methods in need of preprocessing techniques to prepare images for training and classification.

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 , and the **Cartoon Set**, an image dataset of random cartoons/avatarts , as well as a number of noisy images (mainly of natural backgrounds) to be detected and removed from the training data, which generally constitues 80% of the entire dataset. All images are labelled with hair colour, whether the subject is wearing glasses, is smiliing, and is classified as human.

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 Perceptor (MLP) models based on the extracted facial landmarks. A comparative analysis of the performance of each method with respect to the labaled noisy images facilitated the selection of the most appropriate feature extraction method for the given dataset.

Prior to feature extraction, various preprocessing techniques were carried out on the images to improve both the performance and the processing power during the later stages of the extraction, training and classification procedures. Some of the 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, and mean normalisation.

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 [1]. 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.

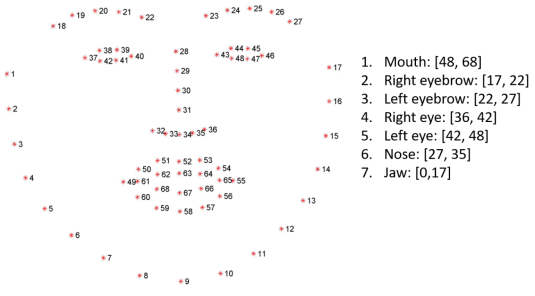

Fig. 1. Face shape defined by 68 landmarks

The relative balance between the expected accuracy and complexity of a HoG detector with respect to the Haar

casccade and Deep Learning options makes it preferable for this dataset. Nonetheless, a comparative analysis was perfomed 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 1, 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.

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

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 2, for $a^{(l)}$ representing the activation vector of layer l, resulting in the output $z^{(l)} = \theta^{(l)} a^{(l)}$ for that layer.

$$\delta^{(l)} = \left(\theta^{(l)}\right)^T \delta^{(l+1)} . g'(z^{(l)}) \tag{2}$$

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 ommited 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, resuling 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.

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.

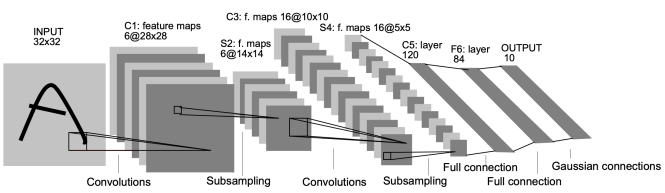

Fig. 2. Architecture of LeNet-5 by LeCun et al. [2]

III. Implementation

The primary image manipulation and pre-processing tasks were performed with the aid of the Open Source Computer Vision (OpenCV) and dlib librarires, 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 manpulation 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 provides a verification of a model's performance.

Each classifier 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 schotastic gradient descent (`sgd`).

IV. Experimental result

V. Conclusion

Summaries all findings and suggest direction for future improvement.

VI. Related Work

Summarise latest reserach on the topic, discussing merits/disadvantages of diff approaches.

REFERENCES

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

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