A Face-Orientation Estimation System

ACP23SD

University of Sheffield

Abstract

This investigation deploys and analyzes the performance of the Multi-layer Perceptron (MLP) in the application of face orientation estimation. The classifier leverages deep neural network techniques to accurately identify 4 orientations - upright, rotated left, upside down, and rotated right. The proposed solution delivers an efficient and robust approach to the classification challenge with sub-images of varied sizes.

1. Introduction

The field of computer vision has undergone dynamic evolution since the emergence of processing images with specialized algorithms focusing on multifaceted functionalities. One of the notable functionalities in the recent innovation of specialized functionalities is the face orientation estimation.[1] This functionality can be further extended to traditional face detection algorithms to enable diverse applications in sectors including surveillance and virtual reality.

Advancements in application-oriented classification algorithms observed across different industries ensure the effectiveness of applying machine learning algorithms for image processing.[2] Harnessing the precision of intricate neural networks, 4 orientations of the face comprising upright, rotated left, rotated right, and upside-down images are classified.[3]

The report provides a comprehensive description of the proposed model and its hyperparameters in Section 2. A series of experiments are conducted on different models to tune the hyperparameters and architecture that are developed in Section 3. Section 4 discusses and analyzes the performances obtained on different pixel sizes from the model. Finally, Section 5 encapsulates the final remarks on the model's efficacy and the future scope of the project.

2. System Description

A pipeline is utilized to organize the set of transformations that are performed on the set of images to classify the orientation. The pipeline initiated in the analysis consists of Principal Component Analysis followed by a Multi-layer Perceptron classifier.

Principal Component Analysis (PCA) is a decomposition technique that reduces the number of features to a target amount. The operation is performed by evaluating the variances of the features using Single Value Decomposition (SVD) or Eigen Value Decomposition (EVD). The features with the highest variances are selected to decrease the dimensional space of the class of images. For each pixel size of the images, the resulting number of components from PCA is tuned.

The classification of face-oriented images is performed with a Multi-Layer Perceptron (MLP) classifier. The classifier consists of a set of weights that are applied to each feature. The array of weights is updated in iterations using a loss function and an optimizing function. The loss function computes a function of the error in classification and the optimizer aids in updating the weights based on the loss function. Apart from the function of error used for the loss function, a function of the weights called the regularization function is used to add constraints on the weights.

The MLP classifier from the Sci-kit learn neural networks module comprises multiple parameters that include hidden layer sizes, activation functions, solver functions, alpha (regularization), and maximum iterations. Hidden layer size is a tuple parameter indicating the number of layers and neurons in each layer. The function applied in each neuron in the layer is termed the activation which is chosen between Identity, Tanh, Relu, and Logistic.[4] The solver parameter is the optimizer function with default as Adam due to its efficient processing in large datasets.[5] The solver parameter is fixed as Adam throughout the analysis with consideration of its relative performance. The alpha parameter quantifies the L2 regularization applied to the weights. Increasing alpha creates a higher weightage on the weights compared to the loss function.

3. Experiments

The training data is developed from the 'Labeled Faces in the Wild' dataset. Sub-images of sizes 30,50 and 90 are extracted utilizing the patch extractor class. Considering the size of the collection of images, the maximum number of patches is initialized in the extractor object.

The extracted sub-images are rotated by 90 degrees utilizing the rot90 method from the NumPy package. The k parameter indicates the number of times the 90-degree rotation transformation is randomized between the range 0 and 4. The random number generated for every transformation is stored as the labels of the training data.

The presented classification problem with different pixel sizes is approached with 5 classification algorithms from the Sci-kit learn packages - K Nearest Neighbours (KNN), Random Forest Classifier, Support Vector Machine (SVM), and Multi-layer Perceptron (MLP).

Table 1: *Comparison of MLP with different classifiers for different pixel values.*

Classifier	% Accuracy-30	% Accuracy-50	% Accuracy-90
SVM	57.4	78.5	98.1
KNN	48.9	73.7	95.7
RFC	71.8	94.4	93.9
MLP	57.7	78.2	97.3

The superior performance of SVM and MLP across all pixel sizes is evident. The MLP is selected over SVM owing to the high tunability factor and expeditious training time.

The hidden layers attribute indicates the number of layers and neurons in each layer. The performance for different layer combinations with varying numbers of neurons is empirically computed based on a validation set (train.small) and presented below.

Table 2: *The system performance for different hidden layer combinations across pixel values*

Hidden layers	% Accuracy-30	% Accuracy-50	% Accuracy-90
100	45.7	63	96.7
100,100	48	66.6	95.8
100,100,100	46.8	64.4	95.8
50,50	46.9	63.5	93.9
200,200	50.1	71.6	94.8
50	45.7	64.3	95.9
200	44.2	70.8	96.5

Observations reveal that the maximum accuracy in 90 pixels is observed with a single hidden layer while 50 and 30 pixels perform better with 2 hidden layers. This phenomenon is attributed to the increase in the complexity of features as the resolution decreases. Across all the pixel sizes, a trend emerges of augmenting the number of neurons in each hidden layer enhancing the accuracy.

The activation function applied in the neuron is empirically varied on the validation set and the performance is tabulated.

Table 3: *The system performance for different activation functions across different pixel values.*

Activation	% Accuracy-30	% Accuracy-50	% Accuracy-90
ReLu	48.7	69.4	96.7
Tanh	50.3	70.4	96.3
Logistic	34.4	68	95.6
Identity	35	45.8	85.7

ReLu and Tanh demonstrate enhanced performance compared to Identity and Logistic functions. The discernable conclusion is that the inherent linearity of the functions plays a vital role in the performance.

Finally, the regularization factor of alpha is tuned based on the accuracy with the best-performing hidden layers and activation function parameters.

Table 4: *The system performance for different alpha values across different pixel values.*

Alpha	% Accuracy-30	% Accuracy-50	% Accuracy-90
0.1	50.3	71.4	96.7
0.01	47.8	71	96.0
0.001	48.3	70	96.0
0.0001	48	71.5	95.6

The regularization table indicates a pattern of decreasing accuracy as the alpha parameter shrinks. This implies that increasing alpha prevents the model from overfitting and enhances the capability of generalizing on unseen data.

After empirically tuning the hyperparameters of the MLP classifier considering computational resources, the number of components for PCA is tuned using GridSearchCV. Considering the high demand for computation resources for the grid search class, the parameters are carefully chosen.

4. Results and Analysis

The performance of the tuned MLP Classifier for each pixel resolution is presented below.

Table 5: *The system performance for different pixel values.*

Pixels	% Accuracy	% Precision	% Recall
30	58.6	58.5	58.1
50	80.5	80.5	80.5
90	98.1	98.0	98.0

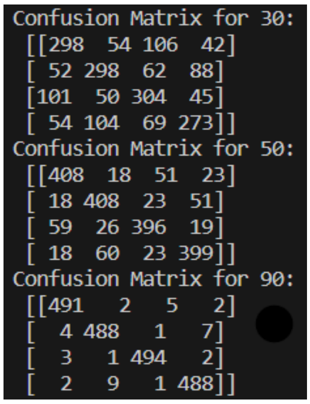

Figure 1: *Confusion matrix - Pixels - 30,50,90*

Based on the confusion matrix, it can be inferred that off-diagonal errors increase as the resolution of the pixels decreases. This is further supported by precision and recall metrics.

5. Conclusions

In conclusion, the MLP classifier proved to be an adaptable robust solution for face orientation classification across all pixel resolutions. The model performs with an accuracy score of 98%, 80%, and 58% for 90,50, and 30 pixels respectively, indicating a decrease in pixel resolution, and the accuracy of prediction reduces. This is due to the unavailability of sufficient features to evaluate the orientation of the image. The tuning of hyperparameters provided a significant boost in the performance of the classifier with valuable insights into the algorithmic complexity of MLP. In the future, consideration of more hyperparameters of the MLP classifier and tuning them would prove vital to improving the performance.

6. References

[1] H. N. Viet, L. N. Viet, T. N. Dinh, D. T. Minh, and L. T. Quac, “Simultaneous face detection and 360 degree head pose estimation,” in *2021 13th International Conference on Knowledge and Systems Engineering (KSE)*, 2021, pp. 1–7.

[2] X. Zhang, D. Zhang, J. Ge, K. Hu, L. Yang, and P. Chen, “Multi-stage real-time human head pose estimation,” in *2019 6th International Conference on Systems and Informatics (ICSAI)*, 2019, pp. 563–567.

[3] X. Glorot and Y. Bengio, “13th international conference on artificial intelligence and statistics (aistats),” *IEEE Transactions on Acoustics, Speech and Signal Processing*, 2010.

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, 2015.

[5] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *3rd International Conference for Learning Representations, San Diego*, 2014.