Student Number: 270862

1. Introduction

Machine learning can be broadly defined as the usage of algorithms to make (artificially) intelligent predictions [1]. In the case of supervised learning, the algorithm is first trained on a labelled training dataset before its performance is evaluated on a test dataset. There exists a myriad of machine learning algorithms that are designed to perform a wide range of prediction tasks, from classification (binary or multi-class) to regression.

1.1. Approach

In this assignment, machine learning was used to perform a supervised binary classification task, to identify images as either 'happy' or 'sad'. The training dataset comprises 2,500 samples and 2,304 features, of which 2,048 were extracted from the fully-connected activation layer (fc7) of the CaffeNet deep learning framework [2]. The remaining 256 features are GIST features, which are high-level representations of the overall scene, providing general yet meaningful information about the key features of the image [3].

To determine the optimal machine learning algorithm for this task, a two-pronged approach was adopted. Firstly, the overall feature space was divided into two subspaces – the CaffeNet features and the GIST features. Each feature subspace was used with a separate algorithm, and the validation accuracy for each algorithm was compared in order to select the final algorithm to be evaluated on the test dataset. The CaffeNet features were used to train a multi-layer perceptron classifier, whereas the GIST features were used to train the following algorithms – K-nearest neighbours (kNN), support vector machine (SVM), random forest (RF), and logistic regression (LR).

Confidence labels provided in the training dataset were used as sample weights for model training. Default sample weights are equal in most machine learning algorithms, but the *sample_weight* parameter allows for more weight to be given to specific samples. In this case, samples which were classified with full agreement were deemed to be more important and thus given more weight (1.0) than samples which were classified with partial agreement (0.66).

The key metric evaluated for this binary classification task is model accuracy, which is formally defined as the fraction of correct classifications over the total number of classifications.

2. Methods

This section outlines the methods and techniques used in this task. The Python modules/ libraries used include: NumPy, Pandas, Matplotlib, Scikitlearn, TensorFlow, and IPython.display.

2.1. Data Preprocessing

Firstly, the training and test datasets contain randomly distributed missing values that have to be imputed prior to model selection and training. Due to the widespread distribution of missing values, whereby a sample has more than one missing feature value, missing values were imputed using a K-nearest neighbours approach (Scikitlearn's *KNNImputer*). The missing values were imputed with the mean value of the five nearest neighbours within the dataset.

After imputation of missing values, the input data was normalized using Scikitlearn's StandardScaler. Values were standardized by centering values around mean 0 and variance 1.

2.2. CaffeNet Features

For the training of the MLP classifier, all 2,048 features extracted from the fc7 layer of the CaffeNet convolutional neural network (CNN) were used. The training dataset was split into training and validation datasets in an 80:20 ratio.

The model was built using the Keras Sequential API, with the input layer having 2,048 neurons, and one hidden dense layer with 128 neurons, activated by the rectified linear activation (ReLU) function. A batch normalization layer was added to normalize the output of the dense layer, which has been shown to stabilize the learning process and reduce the number of epochs required to train the model, and a dropout layer was added to counter the possibility of overfitting. A L2 activity regularizer parameter was added to the dense layer as an attempt to reduce the overall complexity of the model, to counter the problem of overfitting. To further combat potential overfitting, an EarlyStopping callback was implemented, which stops the training of the model when validation loss stops decreasing for 3 epochs. Finally, an output layer with one output neuron was used, with the sigmoid activation function for binary output. A brief schematic of the model structure can be seen in figure 1 below.

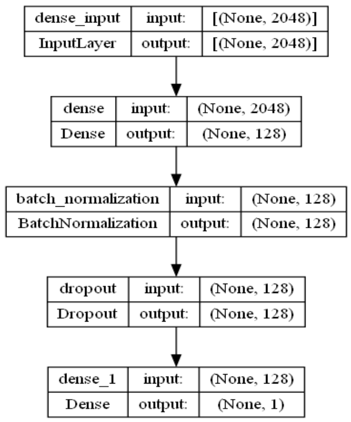

Figure 1: Schematic of MLP classifier.

2.3. GIST Features

All 256 GIST features were used to train the following classifiers: kNN, SVM, RF, and LR. This also doubles as an algorithm selection step that allows us to choose the classifier that performs best on the GIST features. The classifiers were trained in a 5x2-fold nested cross-validation algorithm [4]. Each of the 5 folds is first set aside, followed by a 2-fold cross-validation for hyperparameter selection in each of the remaining 4 folds. The best hyperparameter combination is used to estimate the model's validation score on the initial hold-out fold. This process is repeated for each classifier.

The hyperparameters tested are included in table 1 below:

Table 1: Hyperparameter tuning for each classifier.

Algorithm	Hyperparameters Tested
kNN	• n_neighbors: [1, 2, 5]
SVM	• kernel: ['rbf', 'poly', 'sigmoid'] • C: [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000] • gamma: [0.0001, 0.001, 0.01, 0.1]
RF	• criterion: ['gini', 'log_loss'] • n_estimators: [10, 100, 200, 500, 1000, 5000, 10000]
LR	• solver: ['lbfgs', 'saga', 'liblinear'] • C: [0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]

3. Results and Discussion

3.1. Algorithm Selection Results

From the validation accuracy attained by each classifier, it can be concluded that the MLP classifier, using the extracted CaffeNet CNN features, is the most accurate in classifying the samples in the validation dataset (table 2).

Table 2: Validation accuracy of various tested classifiers, along with the feature set that was used to train the classifiers.

Algorithm	Features	Validation Accuracy (%)
MLP	CaffeNet	72.72
kNN	GIST	62.32 ± 1.40
SVM	GIST	64.80 ± 2.30
RF	GIST	67.40 ± 2.32
LR	GIST	65.32 ± 3.79

Thus, the MLP classifier was selected to predict the test data classifications, using the 2,048 features extracted from the CaffeNet CNN.

3.2. Discussion

Interestingly, although GIST features have been shown on multiple occasions to perform better than CNNs alone [5, 6], in this case, the MLP classifier using the features extracted from the CaffeNet CNN actually performed better than the GIST features across a range of classifiers. However, one possible limitation of using the MLP classifier is the risk of overfitting. During training, training accuracy from about 67% to 88% in 10 epochs, while validation accuracy plateaued at about 70%. Despite implementing measures to avoid overfitting (see section 2.2), there was still clear evidence of overfitting which will likely affect the model's accuracy on the test data. A test accuracy of about 70% is expected.

Although the models tested on the GIST features did not perform as well as the MLP did on the CaffeNet CNN features, it should be noted that all 256 GIST features were included in the models. Univariate feature selection was preliminarily carried out using F-distributions (Scikitlearn's *SelectKBest* function) to identify relative feature importance values, and to observe if model performance could be improved by using features with high importance. The best 80 features (with the highest F-scores) were selected but did not yield any significant improvements in model performance across the four tested classifiers (kNN, SVM, RF, LR). Future work could involve the use of other feature selection methods such as recursive feature elimination (RFE) or principal component analysis (PCA), and to evaluate these methods on overall model performance.

Throughout the course of this project, many important lessons have been learned regarding the lifecycle of a typical machine learning project. From data preprocessing, feature engineering (extraction and/or selection), algorithm selection, model training, and model evaluation, the author has gained invaluable skills that will undoubtedly be useful in his foray into machine learning.

References

[1] Nichols, J. A., Herbert Chan, H. W., & Baker, M. A. B. (2019). Machine learning: applications of artificial intelligence to imaging and diagnosis. Biophysical reviews, 11(1), 111–118. https://doi.org/10.1007/s12551-018-0449-9

[2] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... & Darrell, T. (2014, November). Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 675-678).

[3] Yajamanam, S., Selvin, V. R. S., Di Troia, F., & Stamp, M. (2018). 4th International Conference on Information Systems Security and Privacy. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018) (pp. 553–561). SCITEPRESS. Retrieved May 14, 2023, from https://www.scitepress.org/papers/2018/66858/66858.pdf.

[4] Schönleber, D. (2018, December 10). A “short” introduction to model selection. Towards Data Science. https://towardsdatascience.com/a-short-introduction-to-model-selection-bb1bb9c73376

[5] Tran, K., Di Troia, F., Stamp, M. (2022). Robustness of Image-Based Malware Analysis. In: Bathen, L., Saldamli, G., Sun, X., Austin, T.H., Nelson, A.J. (eds) Silicon Valley Cybersecurity Conference. SVCC 2022. Communications in Computer and Information Science, vol 1683. Springer, Cham. https://doi.org/10.1007/978-3-031-24049-2_1

[6] Mostajer Kheirkhah, F., & Asghari, H. (2019). Plant Leaf classification using gist texture features. IET Computer Vision, 13(4), 369–375. https://doi.org/10.1049/iet-cvi.2018.5028