Exploring the Ability of Machine Learning Engineers to Predict the Best-Performing Machine Learning Model for the OLIVEs Data Set

1st Garrett C. Botkin
College of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, United States
gbotkin3@gatech.edu

2nd Jaime E Castro Cuevas
College of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, United States
jcastrocuevas3@gatech.edu

***Abstract*—This paper explores the ability for machine learning engineers to predict the best model for correctly classifying a particular complex data set by visualizing the data set. It discusses the results of applying various models and methods of visualization to the Ophthalmic Labels for Investigating Visual Eye Semantics (OLIVES) data set, a collection of medical images. The OLIVEs data set contains eight different labels corresponding to a Diabetic Retinopathy Severity Scale (DRSS) score. These eight labels are further reduced to three labels describing low (35, 43), medium (47, 53), and high severity (61, 65, 71, 85). To aid in visualizing the OLIVES data set in a 3D space, the number of features are reduced from 249,984 to 3 using principal component analysis (PCA).**

The PCA-reduced OLIVES data set is visualized through several methods of visualization including a 3D scatter plot, kernel density estimation (KDE) plot, and a pair plot. Based on these visualization methods, the team makes educated guesses about what model will perform the best on the OLIVEs data set. Models considered include K-Nearest Neighbors (KNN), Decision Tree, Gaussian Naive Bayes (GaussianNB) and Convolutional Neural Network (CNN). The true best model is determined by results in balanced accuracy and F1 scores among the four models. The paper concludes by evaluating the ability for the machine learning engineers to determine the true best model.

I. Introduction

A large part of any machine learning problem is selecting a good model for the problem. Selecting this model is so important that machine learning models are being developed to predict which models would perform the best for a specific data set. However, papers exploring the ability of machine leaning engineers' ability to predict a good model for a data set has been sparse so far.

This paper seeks to explore the ability of machine learning engineers to predict a good model for the OLIVEs data set, a complex data set made up of optical medical images. Visualization of the OLIVEs data set through a 3D scatter plot, a kernel distribution estimate (KDE), and a pair plot are provided to the machine learning engineers to aid in model selection. With the help of these visual aids, the machine learning engineers select the best model from a selection of four models: K-Nearest Neighbors (KNN), Decision Tree, Gaussian Naive Bayes (GaussianNB) and Convolutional Neural Network (CNN). After prediction, these models are trained, validated, and tested on the OLIVEs data set. Performance of the models are reported in terms of balanced accuracy score, precision, recall, and F1.

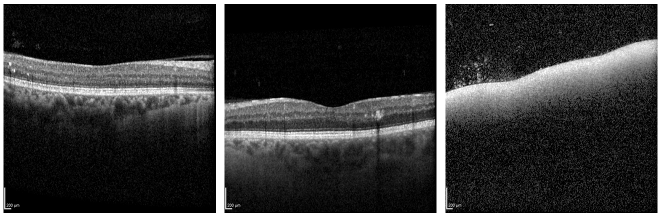

Fig. 1. From Left to Right: OCT scan images with from patients diagnosed with Diabetic Retinopathy; Severity 0 (Low), Severity 1 (Medium), Severity 2 (High)

II. The OLIVEs Data Set

The OLIVEs data set contains a total of 32,240 samples of gray scaled images with a size 504 pixels by 496 pixels. Each sample is labeled with eight different scores from the Diabetic Retinopathy Severity Scale, which represents the class of the image. As explained in [1], the Diabetic Retinopathy Severity Scale (DRSS) was determined by a trained image analyst based on color fundus photography during the Clinical Study. Theses images were created through the Optical Coherence Tomography (OCT) process, which is non-invasive imaging test that uses light waves to take pictures of the retina. Furthermore, metadata contained within the images also describes the patient in terms of age, gender, race, diabetes type, years as a diabetic, body mass index (BMI), best corrected visual acuity (BCVA), and central sub field thickness (CST). However, the models contained within this paper are trained using only the image data portion.

A. Preparing the Data Set

Samples within the OLIVEs data set first undergo several transformations before being split into train, validation, and test sets. Samples are center cropped to a size of 100 pixels

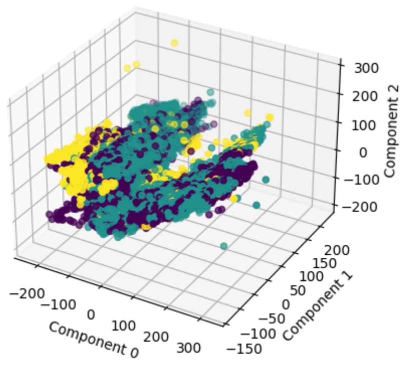

Fig. 2. PCA-reduced Train Data plotted using a 3D Scatter Plot

by 496 pixels. Individual pixels are scaled to a range of 0 to 1 from 0 to 255, and normalized with a mean of 0.1706 and standard deviation of 0.2112. The OLIVEs data set is then split into 7,987 test samples, 4,851 validation samples, and 19,403 training samples. The labels are reduced from eight different labels to the three, representing low (35, 43), medium (47, 53), and high severity (61, 65, 71, 85). Examples of each of the three labels of severity can be seen in Fig. 1.

After theses transformations, the test, validation, and train data sets are then reduced using principal component analysis (PCA) to three components to enable visualization of the train data set through a pair plot, a 3D scatter plot, and a kernel density estimator plot.

III. Visualization Predictions

Visualization of the OLIVEs data set is performed after separation into train and tests sets, but before separation into train and validation sets. The machine learning engineers examine each plot to determine if any hints within them existed to aid in predicting the best model. The 3D scatter plot and pair plot are used to determine if the data set is linearly separable. The KDE plot is used to determine if any separability exists in the Gaussian distributions of components between the three labels.

A. The 3D Scatter Map

The plotting of the PCA-reduced train and validation data set upon the 3D scatter plot, as seen in Fig. 2, reveal what seems to be a linear separation between label 0 and labels 1 and 2. The machine learning engineers believe that this would give a small edge to the decision tree and CNN models, which both make use of linearity within the models to make predictions.

B. Kernel Density Estimator Plot

The plotting of the individual component distributions for each label of the train and validation data sets, as seen in Fig. 3, show similar component distributions among the three labels. As a result of this observation, The machine researchers believe that this would result in the GaussianNB model performing worse than the other three models.

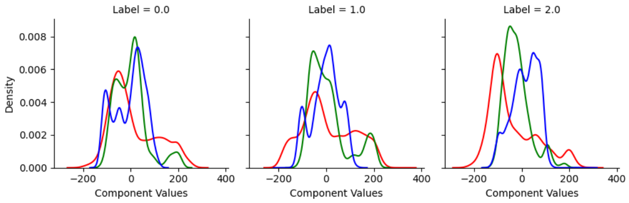

Fig. 3. PCA-reduced Train Data Components Density Visualized Using a Kernel Density Estimator Plot

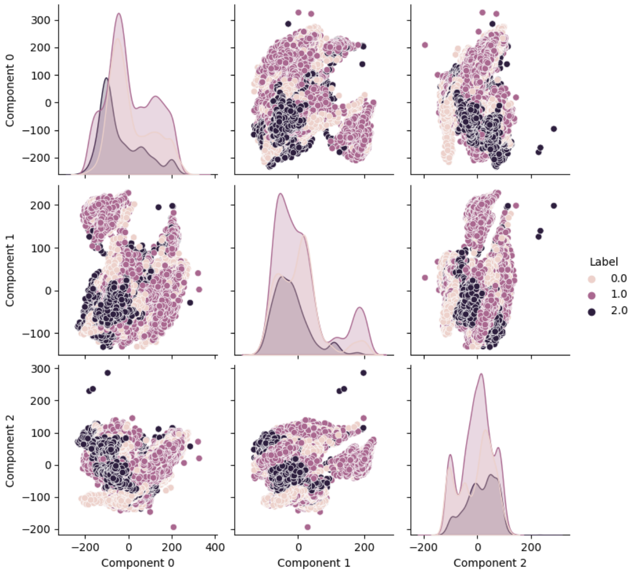

Fig. 4. PCA-reduced Train Data Components visualized using a Pair Plot

C. The Pair Plot

The plotting of the PCA-reduced train and validation data set upon the Pair Plot, as seen in Fig. 4, does little to help the machine learning engineers in narrowing down which model would perform the best. Examining the plot does not seem to reveal any linearity or patterns that would assist the machine learners in their prediction.

D. Visualization Conclusions

After examining the visualized OLIVEs data set, the machine learning engineers quickly eliminate GaussianNB as being the best performing model. The seemingly linear separation of label 1 and labels 1 and 2 seen in the 3D scatter plot initially led the machine learning engineers to believe that decision trees would have the best performance, with an obvious risk of over fitting. However, after examining the pair plots, the complexity of the data led them to believe that a CNN would lead to the best performance, as the numerous parameters and combination of linear and non-linear layers would hopefully allow it to learn the data better than the decision tree. The machine learning engineers also considered KNN closely behind CNN, believing that a appropriately chosen k-value would enable it to perform only slightly worse.

IV. The Models

After visualization, machine learning engineers train and test the four models: K-Nearest Neighbors (KNN), Decision Tree, Gaussian Naive Bayes (GaussianNB) and Convolutional Neural Network (CNN). Performance metrics of each model are collected in terms of balanced accuracy, F1, recall, and precision.

A. K-Nearest Neighbors

The K-Nearest Neighbor classifier model is trained upon PCA-reduced train data set with the ”K” number of nearest neighbors set to 50 neighbors. The value for the k-nearest neighbors is determined experimentally through maximizing balanced accuracy with different k-values. The implemented model uses the Minkowski distance metric with a power parameter equal to two, which makes the distance metric equivalent to the Euclidean distance.

The performance of the K-Nearest Neighbor model can be seen in Table. II.

As explained in section II, the images in the set were center cropped to a size of 100 pixels by 496 pixels. Most of the useful image information was contained within the resultant image, and the post dimensionality reduction conducted through principal component analysis kept some of this information. However, the OCT scan images contain high variation from label to label and even for sample to sample within the same class. This variation in the OCT scans arises due to different manifestations of Diabetic Retinopathy disease, explained in [1]. This imposes a significant challenge for classification based on spatial features with a K-Nearest Neighbor classifier.

B. Decision Tree

The second deployed model was a Decision Tree Classifier that is trained upon the PCA-reduced train data set as well. This Decision Tree was setup using the Gini Impurity Index for node splitting criteria, and equally weighted samples so child nodes with zero or negative weight cannot be created.

The performance of the Decision Tree can be seen in Table. II.

The 3D Scatter Plot of Fig. 2 displays a seemingly linear separation between the components of the PCA-reduced data set and this along with the dense and somewhat-homogeneous regions in which the data is distributed were important factors for achieving better result from the Decision Tree Classifier. Furthermore, the principal component analysis helps to avoid over fitting on the training data.

C. Gaussian Naive Bayes

The Gaussian Naive Bayes is the third model implemented that is trained on the PCA-reduced train data set. The Gaussian Naive Bayes Classifier was setup without weights for individual samples.

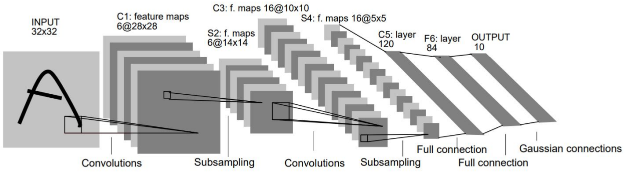

Fig. 5. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition, as illustrated in Fig 2 of [2]

The performance of the Gaussian Naive Bayes can be seen in Table. II.

In addition to similar component distributions displayed by the Kernel Density Estimator Plot in Fig. 3, a Gaussian Naive Bayes Classifier assumes feature independence which is not the case for the OCT scan images from the data set. These images contain features that violate the conditional independence assumption of the Gaussian Naive Bayes model. Furthermore, the dimensionality reduction provided by the application of principal component analysis results in features that are not independent from each other. These are some of the reasons that the machine learning engineers attribute to for the low performance of the model.

D. Convolutional Neural Networks

The fourth classification model that the machine learning engineers deploy is a Convolutional Neural Network. The Convolutional Neural Network is based on the LeNet-5 [2] architecture developed in 1998.

As described in [2], this model is designed to classify handwritten digits from the Modified NIST datasets or MNIST dataset. This model comprises of seven layers in addition the input layer. The inputs are 32 x 32 pixel images and the first layer (C1) is a convolutional layer with six feature maps of 28 x 28 pixels each. Layer 2 (S2) is a sub-sampling layer that outputs six features maps of 14 x 14 pixels, each. The third layer (C3) is a convolutional layer with 16 feature maps of 10 x 10 pixels, which are subsequently sub-sampled in Layer 4 (S4) to 16 feature maps of size 5x5 pixels. Layer 5 (C5) is a convolutional layer with 120 feature maps. Layer 6 (F6) is a fully connected network 84 neurons. The output layer is a fully connected layer with 10 outputs. Layers 1 (C1) to 6 (F6) use a scaled hyperbolic tangent as the activation function, while a Softmax function is used for the final layer's activation. Figure 5 displays the complete Architecture of LeNet-5 [2]

The machine learning engineers adapt the LeNet5 architecture for use with the OCT scan images from the data set. As explained in section II, the images are center cropped to a size of 100 x 496 pixels and these are used as the input to the proposed Convolutional Neural Network. The first layer (C1) is a convolutional layer with kernel size 3, stride 1 and no padding. The activation function for Layer 1 is ReLU. Layer 1 contains eight feature maps with a size of 98 x 494 pixels each. The second layer (S2) is a sub-sampling layer that uses a 2D Max Pooling operation with kernel size 4 x 4, which results in eight feature maps with size 24 x 123 pixels. The

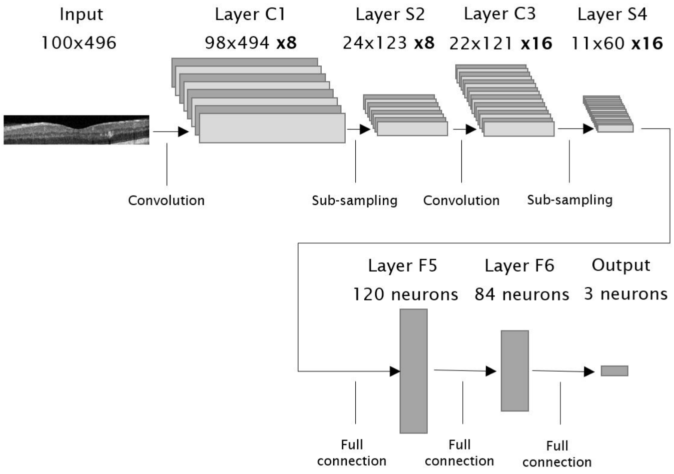

Fig. 6. Proposed Convolutional Neural Network Architecture

TABLE I
DETAILS OF PROPOSED CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE

Layer	Layer Type	Operations	Filter Size	Size of Feat. Map	Activ. Func.
Input	Input	-	-	100x496x1	-
C1	Convolution	-	3x3	98x494x8	ReLU
S2	Sub-Sample	2D MaxPool	4x4	24x123x8	-
C3	Convolution	-	3x3	22x121x16	ReLU
S4	Sub-Sample	2D MaxPool	2x2	11x60x16	-
F5	Fully Connected	Flattening +Dropout	-	120	ReLU
F6	Fully Connected	-	-	84	ReLU
Output	Fully Connected	-	-	3	ReLU

TABLE II
TESTING RESULTS FOR PROPOSED MACHINE LEARNING MODELS

Model	Balanced Accuracy	Precision	Recall	F1
CNN	0.4530	0.6686	0.3492	0.2356
GaussianNB	0.3359	0.3194	0.4674	0.3549
KNN	0.3081	0.3663	0.3857	0.3585
Decision Tree	0.3042	0.3115	0.4317	0.3330

third layer (C3) is a convolutional layer with kernel size 3, stride 1 and no padding. The activation function for Layer 3 is ReLU. Layer 3 has 16 feature maps of size 22 x 121 pixels. Layer 4 (S4) is a sub-sampling layer that uses a 2D Max Pooling operation with kernel size 2 x 2, which results in 16 feature maps with size 11 x 60 pixels. Layer 5 (F5) is a fully connected neural network with 120 neurons. Dropout Regularization at 25 percent is applied after flattening the 16 feature maps input to this layer, which improves the accuracy of the model. The activation function for Layer 5 is ReLU. Layer 6 (F6) is a fully connected neural network with 84 neurons; the activation function for this layer is ReLU. The output layer is a fully connected neural network with 3 neurons and uses a ReLU activation function. It is also important to note that the activation function used for all the convolutional layers is ReLU due to its advantage in preventing vanishing gradients.

The complete architecture of the proposed Convolutional Neural Network model is displayed in Fig. 6. A summary of the methods and techniques in the model is presented in Table I. The performance of the Convolutional Neural Network model can be seen in Table. II. The proposed model has a good performance based on the metrics presented. One of the reasons for this is the known ability of Convolutional Neural Networks for exploiting the spatial information and structures present in images.

V. VALIDITY OF HUMAN PREDICTION

Table II contains the testing results of the four proposed models. The best performing model is the Convolutional Neural Network followed by the Gaussian Naive Bayes model and the K-Nearest Neighbor model. The model that has the lowest Balanced Accuracy is the Decision Tree.

Traditionally, for a classifier used in a medical diagnosis application, the Recall is a metric that has more importance than Precision because the goal is generally to identify the largest amount of patients that would receive a positive diagnosis. However, since the data set used from OLIVEs comes from patients already diagnosed with Diabetic Retinopathy, the Precision metric becomes more important in this case. The listing of the models based on the Precision metric, from greatest to lowest performance, only results in KNN and GaussianNB exchanging places.

The prediction for models' performance order made by the machine learning engineers based on the data visualization tools detailed in Section II is quite different from the actual performance of the models considering both the Balanced Accuracy and Precision metrics. The initial prediction of models performance, from greatest to lowest, are CNN, KNN, Decision Tree, GaussianNB, This prediction only correctly guesses the best model: CNN. Therefore, it can be concluded that while it is possible for machine learning engineers to predict the best performing model amongst a given group of machine learning models after a careful evaluation of the data set using visualization techniques, it is difficult to predict which models of similar complexity will perform best.

Code and results can be found at the public GitHub repository: https://github.com/gbotkin3/ECE8803FinalProject

REFERENCES

[1] M. Prabhushankar, K. Kokilepersaud, Y. Logan, S. Corona, G. AlRegib, and C. Wykoff "OLIVES Dataset: Ophthalmic Labels for Investigating Visual Eye Semantics" Advances in Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks, Nov 29 - Dec 1, 2022.

[2] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.