Comparing Supervised Machine Learning Algorithms for Classification of Damaged Structures

Kerem ÖZÇELİK
MSc-Student Electronics and Communication
Istanbul Technical University
ozcelikk19@itu.edu.tr

Project Page Link: https://github.com/keremozcelik/Advanced-Topics-in-Computer-Vision-Project

Abstract -- There are a large number of Machine Learning (ML) algorithms available. Machine Learning Algorithms can be basically grouped into three categories. These are supervised, unsupervised and reinforcement. Supervised algorithms will be used in this project on the classification problem. Supervised classification is a type of machine learning that uses pre-labelled samples to train classification and then classify new data. In this project, SVM(Support Vector Machine), Naive Bayes, Decision Tree, Random Forrest supervised classification algorithms will be compared on the dataset with damaged structures. This comparison will compare the speed and accuracy of these algorithms.

Keywords – Machine Learning, Supervised Algorithms, Classification, SVM, Naive Bayes, Decision Tree, Random Forrest, ROC Curve, Confusion Matrix, Grid Search, HOG Features

1. INTRODUCTION

Machine learning can be defined as computers acting and learning like people by giving them human observations in information and data form. Today, machine learning has started to be used in the solution of problems in almost every field containing data and images. Machine Learning Algorithms can be basically grouped into three categories.These are supervised learning, unsupervised learning and reinforcement.

There are many machine learning algorithms. These algorithms have different advantages over different datasets. Accuracy and speed are important parameters especially in classification processes. Each algorithm has different values on these parameters. Therefore, choosing which algorithm is suitable for us can be a problem.

In this project, classification process will be carried out on the dataset containing damaged structures with supervised learning algorithms. The speed and accuracy rates of these algorithms in this process will be compared.

2. MACHINE LEARNING ALGORITHMS

2.1 Supervised Learning

It is a type of machine learning that uses labeled training data to make the learning process happen.

Supervised Learning problems are divided into two types. These are classification and regression.

2.1.1. Classification

It is the process of predict the result of a particular sample with output variables in categories. The input and output data in the classification process are categorical. As seen in Figure 1, the data is divided into classes categorized as class A and class B.

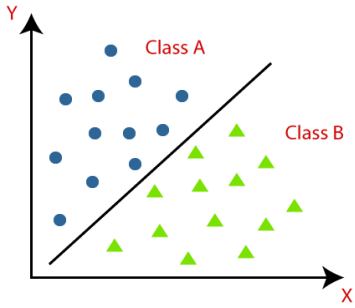


Figure (1) Two class classification process

2.1.2. Regression

It is the process of estimating numerically according to the relationship between two or more variables.In this process, input and estimated output values have continuous values.

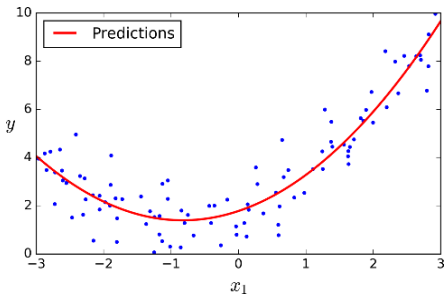


Figure (2) Regression process

2.2 Unsupervised Learning

In unsupervised learning problems, there are only input variables. The corresponding output values are not determined. Unlabeled training data is used to model the structure of the data.

Unsupervised learning problems can be examined in three types as clustering, dimension reduction and combining.

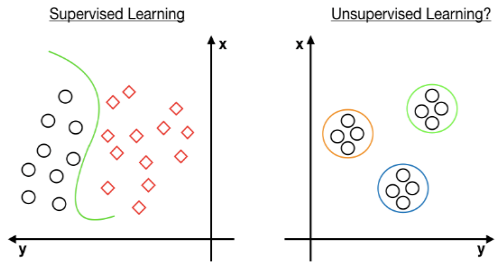


Figure (3) Supervised Learning vs Unsupervised Learning

2.3 Reinforcement Learning

It is a type of machine learning where learning is provided with an reward. It aims to ensure that the reward given as a result of learning is at the highest level. In reinforcement learning, software often learn the best actions through trial and error.

3. SUPERVISED LEARNING ALGORITHMS

In this project, SVM (Support Vector Machine), Naive Bayes, Decision Tree, Random Forest supervised learning algorithms are used for classification on Satellite Images of Hurricane Damage dataset [1].

4.1 Naive Bayes Classification

Bayesian classifiers are statistical classifiers and estimate class membership probabilities with conditional probability. Naive Bayesian classifiers are based on the assumption that the effect of the attribute value of a particular class is independent of other attribute values [2].

The way the Naive Bayes algorithm works simply calculates the probability of each state for an element and classifies it based on the highest probability value. It can produce good results with little training data.

Naive Bayes classifier formula is ;

p(class | data) = p(data | class) * p(class) / p(data)

p (class | data) = The probablity of class being true given that data.
p (class) = The probablity of class being true.
p (data | class) = The probablity of data being true given that class.
p (data) = The probablity of data being true.

4.2 Decision Tree Classification

Decision tree is a tree structure made up of leaves containing condition expressions (node) and class tags. In the decision tree model, control at each node function is applied and one of the branches is selected according to the test result.

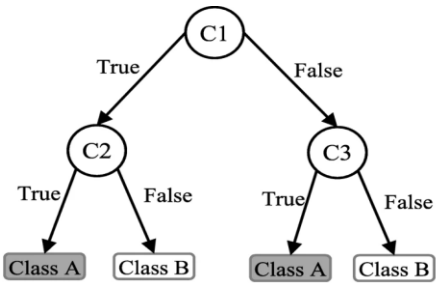


Figure (4) Decision Tree Model Example [3]

4.3 Random Forest Classification

Random Forest is an ensemble classifier consisting of many decision trees [4]. In the Random Forest model, the data is randomly divided into subset. These subsets are trained in decision trees that form the random forest. Each decision tree produces individual results. As a final result, the ensemble average of the results produced in decision trees is taken.

One of the biggest problems of decision trees is over-learning. To solve this problem, the random forest model randomly selects and trains many different sub-sets from both the data set and the attribute set.

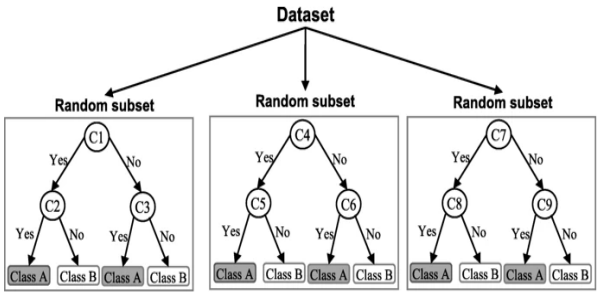


Figure (5) Random Forest Model Example [3]

4.4 Support Vector Machine(SVM) Classification

The support vector machine (SVM) is basically the method used to optimally separate the data of two classes. It identifies the hyperplane that separates the data items into two classes while maximising the marginal distance for both classes and minimising the classification errors [5].

Support vector machine (SVM) algorithm can classify both linear and nonlinear data. SVM, which draws hyperplanes when classifying linear datasets, cannot draw a linear hyper-plane in a nonlinear

dataset. For this reason, kernel tricks called kernel numbers are used. The kernel method greatly increases machine learning in nonlinear data. The most used kernel methods are; Polynomial Kernel, Gaussian RBF (Radial Basis Function) Kernel.

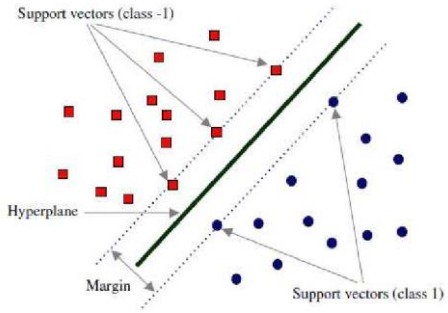


Figure (6) SVM Model Example [6]

5. EXPERIMENTAL EVALUATION

Fit and test operations of algorithms are written in python language on the spyder platform. Sklearn libraries have been used to implement algorithms.

5.1 Dataset

Satellite images of hurricane damage dataset [1] was preferred in this project. There are 10000 train data (damaged and no damage), 2000 validation data, 11000 test data (unblanced and balanced) in this dataset. The data are divided into two groups as damage and no_damage.

The algorithms used were trained with 10000 train data and tested with 2000 validation data and 2000 test data.

Figure 7 shows examples of damage and no damage images from the train of the dataset.


Damage image(train set) No damage image(train set)
Figure (7) Damage and No damage example images

Images labeled as Damage are categorized by 1 in the code, and images labeled as no_damage are categorized by 0 in the code.

5.2 Feature Extraction

Feature extraction is the process of representing the inputs in the system not as whole, but with its sub-features representing that input. Also feature extraction is a kind of reduction dimension method. Features are the information or list of numbers that are extracted from an image.

Feature descriptors use these features to distinguish inputs from each other. A feature descriptor is a simplified representation of the image that contains only the most important information about the image. There are a wider range of feature descriptor algorithms in Computer Vision. In this project, HoG(Histogram of Oriented Gradients) algorithm was preferred from feature descriptors.

5.2.1 Histogram of Oriented Gradients(HOG)

The HOG algorithm can be called characteristic of the orientation (θ) and magnitude values of the pixels in the image.The main purpose in the HOG method is to define the image as a group of local histograms. These groups are histograms in the orientations of the gradients in a local region of the image, in which the magnitudes of the gradients are collected.[7]. The HOG descriptor focuses on the structure or the shape of an object.

In the HOG method, the image is divided into cells in desired sizes. By combining the desired number of these cells, blocks are obtained. Sliding is performed on the image in the size of the blocks obtained. The gradient angle and magnitude of each pixel are calculated with each sliding step. The collected of the gradient angles and sizes obtained are shown as histograms in the desired oriantation number.

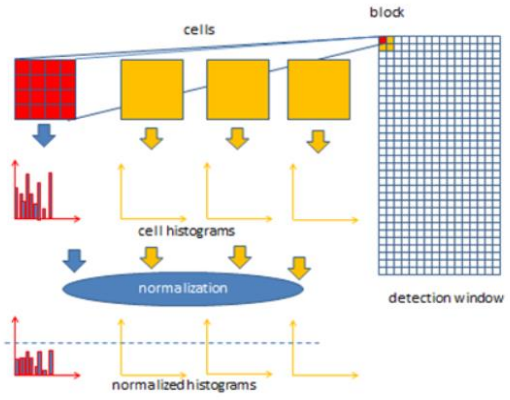


Figure (8) HOG Feature Descriptor

Figure 8 shows the working steps of the HOG feature descriptor.

5.2.1.1 Implementation of HOG on Images

RGB images have been converted to grayscale. The dimensions of the images obtained were 128 * 128. HOG parameter pixels_per_cell 16 * 16 is selected. As a result of this process, 64 cells, 8 horizontally and 8 vertical, were obtained. With the cells_per_block parameter selected as 4 * 4, a block

consists of 16 cells. As a result of this process, the dimensions of the blocks were formed as 64 * 64. In other words, an image has 2 blocks horizontally, 2 vertically and 4 blocks in total.

The blocks can sliding 5 times horizontally, 5 times vertically, cells 4 times horizontally and 4 times vertically. As a result of the 5 * 5 * 4 * 4 * orientation number, finally feature is obtained. Since the orientation number is selected as 8, a total of 3200 features are obtained for one image. In summary, instead of representing an image with 128 * 128, we represented it with 3200 features.

```
orientations=8,
pixels_per_cell=(16,16),
cells_per_block=(4, 4),
```

The HOG parameters used are shown above.

5.3 Model Hyperparameters Tuning

In learning models, the values that must be decided by the designer when designing the models are called hyperparameters. The appropriate selection of hyperparameters is one of the most important factors affecting the success of the model. But some of the hyper parameters are in a position to take an infinite number of values.

There are different methods for determining hyperparameters. In this project, grid search technique was used to select hyperparameters.

5.3.1 Grid Search

In hyper parameter selection with grid search; For combinations of all values in the specified range, the network is trained and the best combination is selected as the hyper parameter group, depending on the situation observed [8].

5.3.1.1 SVM with Grid Search

C (Regularization parameter), kernel and gamma (Kernel coefficient), max_iter are the most important parameters in SVM model. Kernel = rbf was chosen from these parameters. To accelerate, max_iter parameter was chosen as 1000. This value is the recommended value in the linear model. Grid search technique was applied for C and gamma values. Combinations of different C and gamma values were tried on train data with grid search and the best parameters were obtained.

The best SVM model parameters determined as a result of the grid search can be seen in figure 9. Other parameters of SVM are used as default values.

Key	Type	Size	Value
C	int	1	10
gamma	str	1	scale
kernel	str	1	rbf
max_iter	int	1	1000

Figure (9) Best SVM Model Hyperparameters

5.3.1.2 Decision Tree with Grid Search

Grid search was used for the value of criterion, max_features,min_samples_leaf, min_samples_split parameters of the Decision Tree model. Criterion is the function to measure the quality of a split. Splitter is the strategy used to choose the split at each node. Splitter was chosen “best”.

The best Decision Tree model parameters determined as a result of the grid search can be seen in figure 10. Other parameters of Decision Tree are used as default values.

Key	Type	Size	Value
criterion	str	1	gini
max_features	int	1	10
min_samples_leaf	int	1	10
min_samples_split	int	1	2
splitter	str	1	best

Figure (10) Best Decision Tree Model Hyperparameters

Grid search result min_samples_leaf value was 10, but 1 value was better on test data.

5.3.1.3 Random Forest with Grid Search

Grid search was used for the value of criterion , n_estimators, max_features, min_samples_leaf, min_samples_split parameters of the Random Forest model. Value of n_estimators shows the number of trees to be used.

The best Random Forest model parameters determined as a result of the grid search can be seen in figure 11. Other parameters of Random Forest are used as default values.

Key	Type	Size	Value
criterion	str	1	entropy
max_features	int	1	10
min_samples_leaf	int	1	1
min_samples_split	int	1	3
n_estimators	int	1	150

Figure (11) Best Random Forest Model Hyperparameters

5.3.1.4 Naive Bayes with Grid Search

The Naive Bayes model has two parameters on the sklearn library. Grid search was used for the value of var_smoothig.

The best var_smoothing value of the Naive Bayes Model found in the grid search result is shown in figure 12.

Key	Type	Size	Value
var_smoothing	float	1	1e-12

Figure (12) Best Naive Bayes Hyperparameter

5.4 Evaluation Metrics

Confusion matrix and ROC curve analysis techniques were used as evaluation metrics.

5.4.1. Confusion Matrix

Confusion matrix, error matrix where estimates and real values are compared is often used to evaluate the performance of classification models used in machine learning [9]. There are TP (True Positive), TN (True Negative), FN (False Negative), FP (False Positive) values on the confusion matrix. The diagonals of the matrix give the number true.

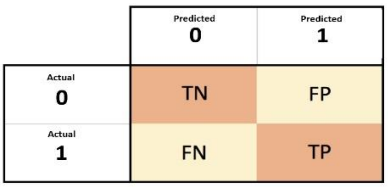


Figure (13) Confusion Matrix Example

5.4.2. ROC (Receiver Operating Characteristic) Curve Analysis

ROC curve is the curve that shows the performance of the model. A typical ROC curve has False Positive Rate (FPR) on the X axis and True Positive Ratio (TPR) on the Y axis. The larger the area under the ROC curve, the better the model's performance.

6. EXPERIMENTAL RESULTS

Four algorithms were trained with the same train set. Then four algorithms were tested on the same validation data and test data. The results of the test performed on validation data and test data are given in figure 14 and figure 15, respectively.

According to these results, while SVM and Random Forest algorithms gives the best accuracy value, the best run time belongs to Decision Tree algorithm. Run time includes the duration of the train and test operations. Of course, the run time values of the algorithms will vary depending on the performance of the computer being run.

Index	Naive Bayes	Decision Tree	Random Forest	SVM
accuracy score(%)	83.85	71.45	86.15	88.55
run time(second)	1.62581	0.407372	13.8964	79.735

Figure (14) Validation Data Results

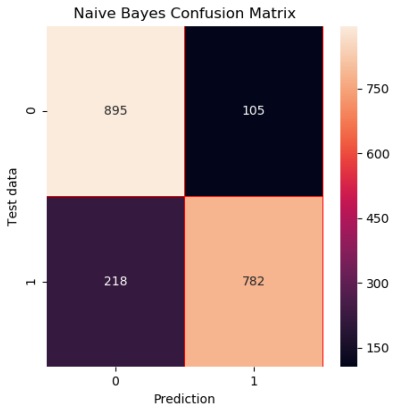
Index	Naive Bayes	Decision Tree	Random Forest	SVM
accuracy score(%)	82.1	86.5	92.3	94.2
run time(second)	1.02503	0.233178	13.9433	79.6687

Figure (15) Test Data Results

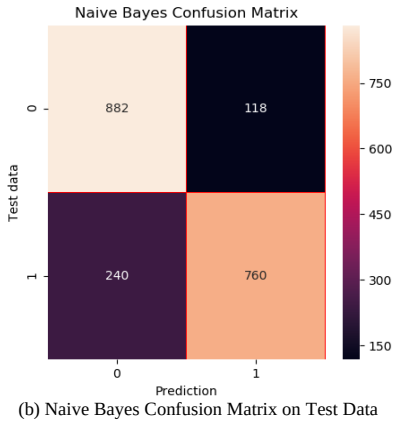
The results on the test data usually are better than the validation data results. This indicates that the validation data used is more difficult data.

6.1. Naive Bayes Classification Results

In Naive Bayes model, the accuracy results of the test and validation data are very close to each other and approximately 82-83%. Although it lags behind SVM and Random Forest algorithms as its accuracy value, it has a run time of about 1 second.



(a) Naive Bayes Confusion Matrix on Validation Data



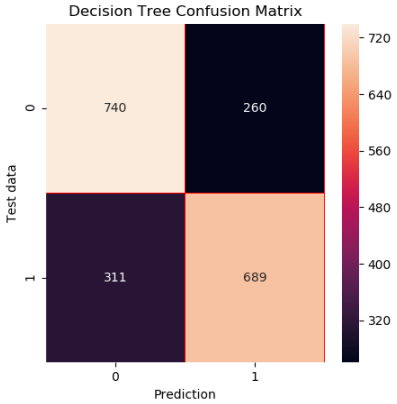
(b) Naive Bayes Confusion Matrix on Test Data

Figure (16) Naive Bayes Confusion Matrices

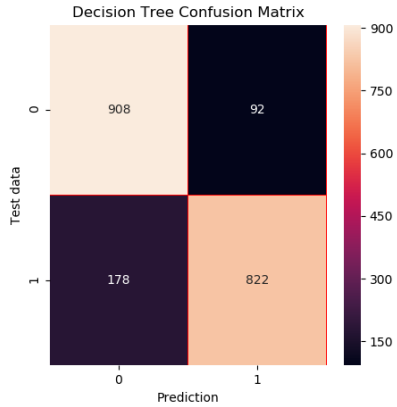
6.2. Decision Tree Classification Results

Although the accuracy value of the decision tree algorithm was 71.45% on the validation data, it achieved a high success of about 86.5% on the test data. This accuracy value is a level that can be accepted well.

The Decision Tree Model, which took about 0.5 second in run time, was the fastest model. This run time is a very good degree.



(a) Decision Tree Confusion Matrix on Validation Data



(b) Decision Tree Confusion Matrix on Test Data

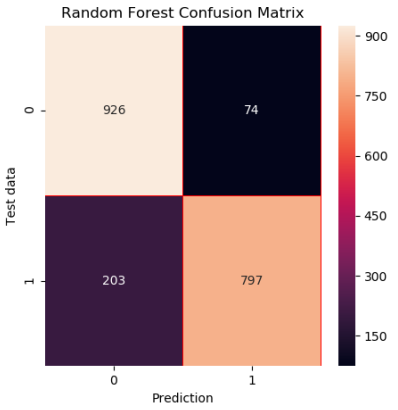
Figure (17) Decison Tree Confusion Matrices

Figure 17 shows the confusion matrix results of the decision tree algorithm. Figure 17(b), according to test data result, 92 of 0 (no_damage) image and 178 of 1 (damage) image were misclassified on 2000 test images. Diagonal values on the Confusion matrix show the correct classification values.

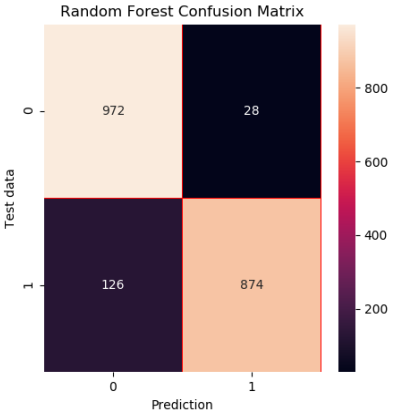
6.3. Random Forest Classification Results

The Random Forest classifier provided the second highest accuracy on both validation and test data. Since it is a model consisting of many decision trees(in this projet 150 trees), it is slower than the decision trees as run time.

The Random Forest model, which is accuracy less than the SVM model, is much better than the SVM in run time. This indicates that it can produce better results with higher feature sizes.



(a) Random Forest Confusion Matrix on Validation Data

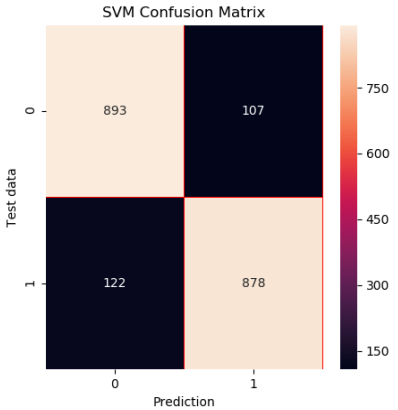


(b) Random Forest Confusion Matrix on Test Data

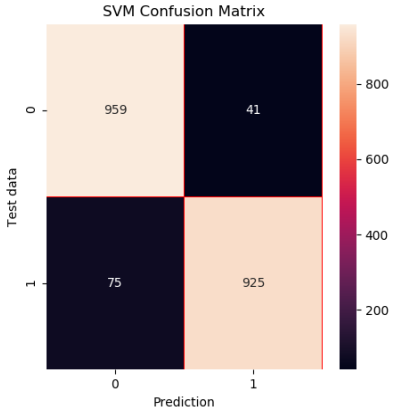
Figure (18) Random Forest Confusion Matrices

6.4. SVM Classification Results

The SVM classifier provided the highest accuracy on both validation and test data. But it has much slower run time than other models. Increasing the number of features will make the SVM model run slower. However, this has been the model that gives the best accuracy value in line with the selected parameters.



(a) SVM Confusion Matrix on Validation Data



(b) SVM Confusion Matrix on Test Data

Figure (19) SVM Confusion Matrices

In addition, SVM model on 1 (damage) and 0(no_damage) values made a more balanced estimate data than other models. The accuracy values on the two labels are almost the closest to each other.

6.5 ROC Curve Results

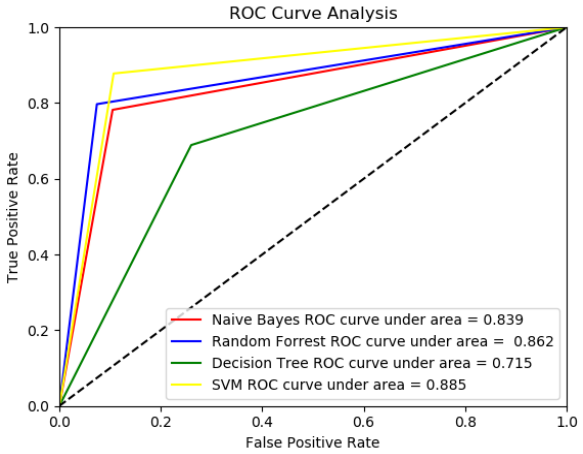


Figure (20) ROC Curve on Validation Data

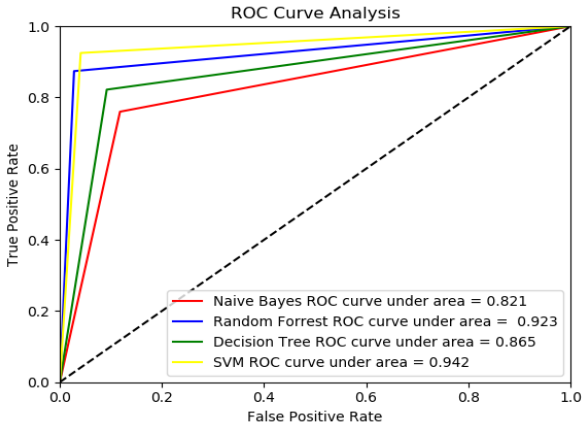


Figure (21) ROC Curve on Test Data

When the ROC Curve graphics are examined, the classifier with the largest area under the curves is the SVM classifier. Also the accuracy value of the Random Forest classifier is very close to the SVM classifier. Naive Bayes classifier better value on validation data than Decision tree, while Decision Tree has shown better results on test data.

6. CONCLUSION

Naive Bayes, Decision Tree, Random Forest, SVM, which are the most preferred machine learning supervised classification algorithms in this project, were compared on the same dataset as accuracy and run time. It is known that these algorithms will give different results on different datasets. SVM algorithm had the best result on Satellite images of hurricane damage dataset [1], which includes the subject of classification of damaged data. As run time, Decision Tree algorithm gave the best result. However, the Random Forest model produced close accuracy values to the SVM model. Random Forest model, which has a better rank in run time, can be preferred instead of SVM model in structures that contain more data and features.

REFERENCES

[1]https://www.kaggle.com/kmader/satellite-images-of-hurricane-damage

[2] Eli Cortez, Mauro Rojas Herrera, Altigran S. da Silva, and Edleno S. de Moura, “Lightweight Methods for LargeScale Product Categorization”, Department of Computer Science, Federal University of Amazonas

[3] Shahadat Uddin, Arif Khan, Md Ekramul Hossain and Mohammad Ali Moni, “Comparing different supervised machine learning algorithms for disease prediction”,
BMC Medical Informatics and Decision Making volume 19, Article number: 281 (2019)

[4] Breiman L. Random forests. Mach Learn. 2001;45(1):5–32.

[5] Joachims T. Making large-scale SVM learning practical. SFB 475: Komplexitätsreduktion Multivariaten Datenstrukturen, Univ. Dortmund, Dortmund, Tech. Rep. 1998. p. 28.

[6] Hamid Reza Baghaee, Dragan Mlaki´c, Srete Nikolovski and Tomislav Dragiˇcevi´c, Support Vector Machine-based Islanding and GridFault Detection in Active Distribution Networks, Ieee Journal Of Emergıng And Selected Topıcs In Power Electronıcs, Vol. Xx, No. Xx, May 2019

[7] Murat Peker, Halis Altun , Fuat Karakaya, “HOG Temelli Bir Yöntem ile Ölçek ve Yönden Bağımsız Gerçek Zamanlı Nesne Tanıma”

[8] James Bergstra, Yoshua Bengio, 2012, Random Search for Hyper-Parameter Optimization, Journal of Machine Learning Research 13 (2012) 281–305

[9] Fawcett T. An introduction to ROC analysis. Pattern Recogn Lett. 2006;27(8):861–74.

The project is located on my Github page. The link is below. The project page includes project presentation, project codes, project report and project video presentation.

Project Page Link:
https://github.com/keremozcelik/Advanced-Topics-in-Computer-Vision-Project

Project Codes Link:
https://github.com/keremozcelik/Advanced-Topics-in-Computer-Vision-Project/blob/master/project_codes.py

Project Video Link:
https://github.com/keremozcelik/Advanced-Topics-in-Computer-Vision-Project/blob/master/project_video.mp4

Project Presentation Link:
https://github.com/keremozcelik/Advanced-Topics-in-Computer-Vision-Project/blob/master/Project%20Presentation_KeremOzcelik.pdf