Research Article
Journal of Intelligent Systems: Theory and Applications 4(2) (2021) 94-102
DOI: 10.38016/jista.772145

# Automatic Ship Detection and Classification using Machine Learning from Remote Sensing Images on Apache Spark

Caner Özcan[1]*, Betül Dolapcı[2]

[1] Karabük University, Department Of Computer Engineering, Karabük, Turkey

[2]Kastamonu University, Department Of Computer Engineering, Kastamonu, Turkey

canerozcan@karabuk.edu.tr, bdolapci@kastamonu.edu.tr

**Abstract**
Ship detection and classification is very important for port and coastal security. Due to maritime safety and traffic control, high-resolution images of ships should be obtained. High resolution color remote sensing ship images taken from short distances provide advantages in ship detection applications. But the analysis of these high-dimensional images is complicated and requires long time. Dividing the image data into smaller blocks and representing them with a vector with distinctive and independent features facilitates the analysis process. For this reason, a block division method is applied first, dividing the image data into small pixel blocks. These obtained image blocks are also represented by the hybrid feature vectors. These feature vectors are created by adding the sub-features extracted from the color and texture properties of the images one after another. Using the obtained hybrid vectors, the images are classified using machine learning methods on Apache Spark. Classification studies were realized using Naive Bayes, Decision Trees and Random Forest methods in the MLlib. The analysis of the images was realized much faster with the clustering architecture created on Apache Spark platform. According to the obtained classification results, 99.62% classification success was achieved by using Random Forest method. In addition, an average of 3.4 times acceleration was achieved by running each method on 1 master + 4 workers clustering architecture on Spark.

**Keywords:** Apache Spark, Classification, Clustering, Machine Learning, Remote Sensing, Ship Detection

# Apache Spark Makine Öğrenimi Kullanılarak Uzaktan Algılama Görüntülerinden Otomatik Gemi Tespiti ve Sınıflandırma

**Öz**
Gemi tespiti ve sınıflandırması, liman ve kıyı güvenliği açısından çok önemlidir. Deniz güvenliği ve trafik kontrolü nedeniyle, gemilerin yüksek çözünürlüklü görüntülerinin elde edilmesi gerekmektedir. Kısa mesafeden çekilmiş yüksek çözünürlüklü renkli uzaktan algılama gemi görüntüleri, gemi tespiti uygulamalarında avantaj sağlamaktadır. Fakat yüksek boyutlu bu görüntülerin analiz edilmesi süreci karmaşık ve uzun süreler gerektirmektedir. Görüntü verilerinin daha küçük parçalara bölünmesi ve bu parçalardan elde edilen ayırt edici ve bağımsız özelliklere sahip bir vektörle temsil edilmesi analiz işlemini kolaylaştırmaktadır. Bu nedenle, öncelikle görüntü verilerini küçük piksel bloklarına bölen bir blok bölümü yöntemi uygulanır. Elde edilen bu görüntü bloklarının da hibrit bir öznitelik vektörleri ile temsil edilmesi gerçekleştirilir. Bu öznitelik vektörleri, görüntülerin renk ve doku özelliklerinden çıkarılan alt özelliklerin birbiri ardına eklenmesi ile oluşturulur. Elde edilen hibrit vektörler Apache Spark'daki makine öğrenmesi yöntemleri ile kullanılarak görüntülerin sınıflandırılması sağlanmıştır. MLlib kütüphanesinde bulunan Naif Bayes, Karar Ağaçları ve Rastgele Orman yöntemleri kullanılarak sınıflandırma çalışmaları gerçekleştirilmiştir. Görüntülerin Apache Spark ortamında analiz edilmesi oluşturulan kümeleme mimarisi ile çok daha hızlı bir şekilde gerçekleştirilmiştir. Ayrıca her bir yöntemin Spark 1 master + 4 worker kümeleme mimarisi üzerinde çalıştırılması sonucu ortalama 3.4 kata yakın hızlanma sağlanmıştır.

**Anahtar Kelimeler:** Apache Spark, Sınıflandırma, Kümeleme, Makine Öğrenmesi, Uzaktan Algılama, Gemi Tespiti

* Corresponding Author.
E-mail: canerozcan@karabuk.edu.tr

Recieved : 21 July 2020
Revision : 04 Feb. 2021
Accepted : 23 Feb. 2021

## 1. Introduction

Today ship detection is very important in terms of maritime safety and maritime traffic. It is advantageous to work with ship images obtained by remote sensing due to coastal and port security (Yang et al., 2018). Also, ship detection and classification with remote sensing images is a crucial for military and civilian fields. In the literature, ship detection studies with remote sensing images are common in these fields (Liu et al., 2017). Another study in literature proposes a ship detection method from optical remote sensing images based on the network with visual attention (Bi et al., 2019). The type of image is crucial for correct feature extraction. Whether the image is a synthetic aperture radar (SAR) or optical remote sensing image requires different feature extraction techniques (Cavallaro et al., 2015). Less complex structure of color image data obtained from short distance provides an advantage in image detection and classification (Morillas et al., 2015). Local feature-based algorithms are used for object recognition in large-scale data obtained from satellite images (Ergul and Alatan, 2013).

A new detector called CenterNet++, working with SAR images, has been proposed (Guo et al., 2020). In this study, CenterNet++ method was developed to reduce complex background and increase detection capability. In another study developed using SAR images, ship detection was carried out according to the extraction of areas connected to water (Shi et al., 2019). Besides these, a new ship detection and classification method for complex sea surface is presented. (Wang et al., 2019).

A different approach presents a method for ship detection using satellite videos (Li et al., 2019). Another paper about vessel detection algorithms presents summarize of studies from optical spaceborne sensor images (Kanjir et al., 2018). Deep learning is used for autonomous ship detection in another paper. In this study, a novel hybrid deep learning method that combines a modified Generative Adversarial Network and a Convolutional Neural Network based detection approach is proposed for small ship detection (Chen et al., 2020). For some ship detection studies, images with land areas on the sea were used. The island filter is used for ship detection in the sea area with a land area in these studies (Wang et al., 2020).

Remote sensing is the technique of recording and examining the earth and ground resources without physical connection with them. In other words, remote sensing aiming to capture the earth images without any physical contact by means of aircraft and satellites and to obtain information through these images. The energy source used for remote sensing is either the sun or an artificial power source. Remote sensing technology has allowed the monitoring local and global environment for object detection (Yuan et al., 2020).

In this study, a hybrid feature vector has been developed for high performance classification and detection operations. The aim is to combine all the distinctive features of the image in a vector space and create a meaningful feature vector that will produce the correct result. Detection studies were carried out on remote sensing ship images in the marine environment. Before the feature vector was extracted on the images, the pre-processing was carried out. Noise removal and complex background cleaning were performed on the image. Pixel-based approaches make analysis difficult by considering unnecessary and non-distinctive variables (Morillas et al., 2015).

In this study, the block section is proposed to overcome the specified problem. Thanks to this block section, images are divided into small pixel blocks labeled as ship blocks or non-ship blocks. The classification of the blocks was carried out with the MLlib module of Apache Spark, which is used to classify large amounts of data. Naive Bayes, Decision Trees and Random Forest methods under this module have been applied.

In this approach, color and texture analyzes of the image are made and different features from both contents are combined in a hybrid vector. With the new hybrid vector formed, features are extracted from each block and then used for training and classification. In terms of efficiency of classification results, image blocks were analyzed in three different sizes and compared.

The rest of the article is organized as follows. Apache Spark Technology and architecture of cluster system are explained in Chapter 2. The detection process and the process of the formation of the feature vector, is detailed in Chapter 3. Machine learning classification algorithms of Spark used for fast data classification are explained in Chapter 4. The results of the analysis are presented in Chapter 5 with the tables and an evaluation is made by comparing the results of three different methods. Future work is presented in Chapter 6.

## 2. Apache Spark Clustering System

Apache Spark is an open-source library developed with Scala, which enables parallel processing on large data sets formed by high volume data. Spark has been developed as an alternative to the MapReduce method. Spark can be developed with Java, Scala, Python and R programming languages and supports SQL, data flow, machine learning and graphics processing.

The ability of the Hadoop Distributed File System (HDFS) and MapReduce method offered by the Hadoop environment to store data on multiple machines and to achieve parallel processing is faster and easier to achieve. It is due to the architecture that Apache Spark processes data faster and easier. Data is analyzed much faster using more than one machine. An abstraction method, defined as flexible distributed datasets (Resilient Distributed Datasets, RDD), is a collection of divided objects among a series of machines that allow lost data to be reproduced. With this method Spark

performs 10 times better than Hadoop in iterative machine learning operations and can analyze 39 GB of data interactively in under 2 seconds.

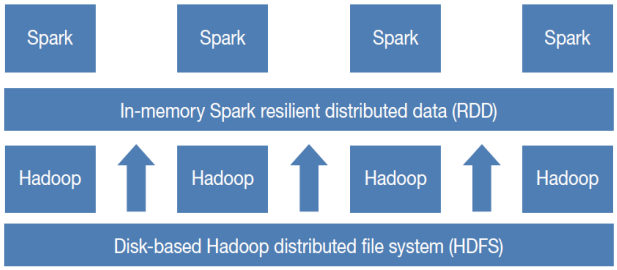


**Figure 1.** Spark and Hadoop in the cluster

Using Apache Spark clustering architecture, tasks can be distributed to computers in parallel. A Spark standalone cluster provides own web UI (User Interface) by monitoring cluster processes and running applications. It has a simple and efficient architecture. Standalone cluster consists of master and workers. Master is cluster manager that configures worker's processes and running applications. Workers start application's executors for tasks. Worker nodes communicate after completing their tasks in parallel and give the result of application to the master node.

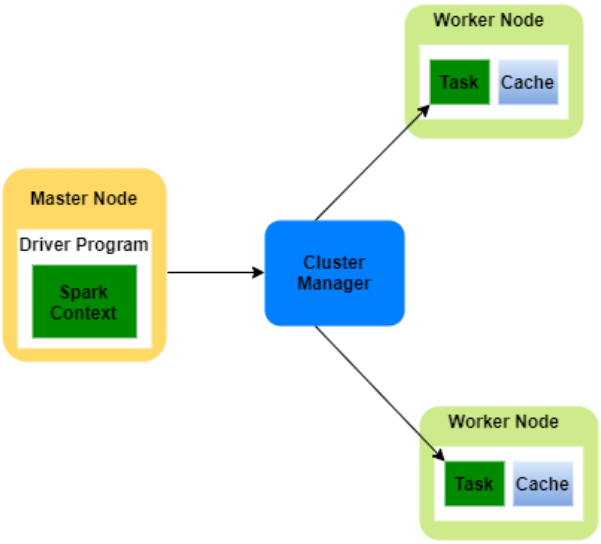


**Figure 2.** Architecture of Apache Spark cluster

In the literature, several methods have been developed with Spark to analyze various types of data. It was observed that text data was used for fast classification (Ogul et al., 2017) as well as image data for fast detection and classification (Ozcan et al., 2018). In the detection of objects in large-scale image data, Spark produces efficient results and can provide high performance in classification processes (Wang et al., 2020).

## 3. Feature Extraction Method

In this study firstly, a block section is applied to the images. After this step, features are extracted from the image blocks to be used as training data by extracting the color and texture features. These features are combined to create a hybrid feature vector. Then, the Naive Bayes, Decision Tree and Random Forest classifiers are trained based on the previously extracted feature vectors. As the last step after the classifiers have been trained, the classification between ship blocks and non-ship blocks has been carried out on the blocks of test images.

### *3.1. Block Division*

Block-based approach provides more meaningful and holistic detection as opposed to pixel-based approach. Thanks to this method, which provides more homogeneous information depending on the color and texture content of the image, the rapid creation of the vector is also provided. In this study, features were obtained by applying the block section. The color images passed through the preprocessing stage are divided into 16x16, then 32x32 and then 64x64 pixel blocks and recorded in a folder. In Figure 3, some parts of an image divided into 64x64 blocks are shown.

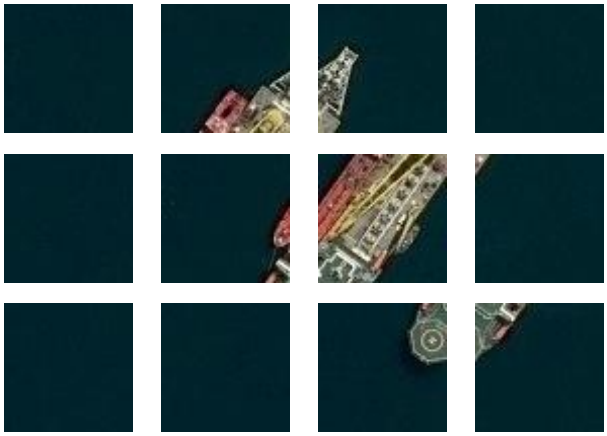

**Figure 3.** Example of an image divided into 64x64 pixel blocks

After the block division section, a binary mask application was applied for each block for labeling purposes. The purpose of this approach is to label images as ship or no ship in advance. The accuracy of the classification to be carried out in the next stages will be made by comparing with the labeled data. For this reason, it is very important to tag the data correctly. Ship blocks represent the pixel regions within the ship. Non-ship blocks consist of pixel areas outside the ship boundaries, such as water and sky areas. The reason for creating a binary mask is the labeling operations of the available image data. In image data, black regions are labeled 0 and white regions 1 (Figure 4.).

(a) (b)

**Figure 4.** (a) Original image. (b) Binary mask applied to the image

Compared to a pixel approach, this block division approach significantly reduces the complexity of the classification process because the number of elements to be classified is significantly reduced. Decreasing the number of elements is important in terms of dimension reduction of big data. In this study a detection was made on image data with image dimensions of 16x16 pixels, 32x32 pixels and 64x64 pixels, and comparisons were presented in tables in the Experimental Studies section.

### *3.2. Feature Extraction*

In classification algorithms such as Naive Bayes, Decision Trees and Random Forests, it is an important decision to select the appropriate features to achieve the desired classification. This selection depends on the split blocks of the image and the type of images available (without pixel-based distortion). In case of ship detection in images obtained from short distance the marine environment provides useful visual features that can be used as a feature. This study proposes extracting color and texture features from each block of images. After analyzing whether the extracted features are meaningful in terms of data, they are added one after another and the feature vector is created. These features are then used during the training and classification stages.

#### *3.2.1. Color Feature*

Colors define the visual perception of pixels, tone distribution according to light and give information about their chromatic densities. In this approach, three different color spaces are evaluated: RGB (Red-Green-Blue), HSV (Hue-Saturation-Value) and L*a*b* defined by CIE (International Commission on Illumination). RGB is a color image area based on the color model commonly used in computer graphics because it works similar to the human visual system. In this model, primary colors are defined by red, green, and blue colors represented by the value of each of the RGB components (Morillas et al., 2015). HSV is a color space used in computer vision and image analysis with applications such as object recognition and image segmentation. One of the main advantages of HSV is the distinction between density and color information similar to that performed by the human brain. Hue (H) describes the shade of the color and its location in the color spectrum. Saturation (S) represents the purity of the tint according to a white reference. Value (V) is the measure of the brightness of the color, that is, the ratio of white in it (Morillas et al., 2015). The CIE LAB color space is based on the human perception of different wavelengths and can identify any color perceived by the average human observer. CIE LAB is a device-independent color compared to RGB and HSV which are device-dependent colors. At the CIE Lab, three parameters are represented by a sphere. The vertical axis L* represents lightness. The horizontal axis a* measures the difference between the red and green components, and the horizontal axis b* measures the difference between the blue and yellow components (Morillas et al., 2015).

Being able to use color spaces as features depends on the mean and standard deviation from each block for each color component. Mean:

$$\mu = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N} I(x,y)}{M \times N} \quad (1)$$

Standard Deviation:

$$\sigma = \sqrt{\frac{\sum_{x=1}^{M}\sum_{y=1}^{N} (I(x,y)-\mu)^2}{M \times N}} \quad (2)$$

where I(x, y) is the color component of the pixel in (x, y), M is the width of each block in pixels and N is the height of each block in pixels.

#### *3.2.2. Texture Feature*

Texture is a feature that represents the structure and spatial properties of pixels in a region. The texture can be characterized by the density properties of pixels and the spatial relationship between them on a gray level. Unlike color properties, texture properties describe region-based information instead of individual pixel. To extract texture features, images are first converted to grayscale, eliminating the hue and saturation information while preserving the brightness component. After this transformation, two types of texture features are extracted from each block: first-order-statistics (FS) and Gray Level Co-occurrence Matrices (GLCM).

In GLCM-based texture analysis, some statistical data provided by this algorithm are based on. Statistical data are explained in the table below. The FS-based statistical data is just like the other table (Gonzalez et al., 2003):

**Table 1**. Features of GLCM

| Feature | Definition |
|---|---|
| Contrast | Density and gray level variations |
| Correlation | Gray level values linear dependence |
| Energy | Pixel homogeneity criterion |
| Homogeneity | Similarity criterion in different regions |

**Table 2**. Features of FS

| Feature | Definition |
|---|---|
| Mean | Pixel values average |
| Standard Deviation | Square root of variance information |
| Variance | Squared deviation from the mean |
| Distortion | Criterion of the asymmetry of its distribution |
| Entropy | Gray level spatial irregularity |
| Energy | Pixel homogeneity criterion |

The reason why the four features above are preferred for feature extraction with GLCM algorithm is that these features combined with FS features best represent the gray level intensities. FS features are given in Table 2.

**Table 3**. Abbreviation for features used in feature extraction.

| Feature | Definitions |
|---|---|
| RGB | Mean of RGB component |
| HSV | Mean of HSV component |
| LAB | Mean of LAB component |
| SD | Standard deviation of color components |
| FS | First Order Statistics |
| GLCM | Gray Level Co-occurrence Matrices |

## 4. Classification using Spark MLlib

Machine learning algorithms under Spark used in image classification are computationally intensive. Spark contributes well to machine learning, as it supports fast in-memory computing and recursive querying of data. MLlib is Spark's scalable machine learning library (Lagerstrom et al., 2016). It consists of common learning algorithms and utilities such as classification, regression, clustering, collaborative filtering and dimensionality reduction. It also includes opportunities to model and train deep neural networks. Spark MLlib provides the use of an application programming interface in Java, Scala and Python, which facilitates integration with an existing Java application that uses OpenIMAJ for image extraction and classification (Han et al., 2006).

### *4.1. Naive Bayes*

Naive Bayes (NB) algorithm is a controlled machine learning algorithm. It is a simple probability model for multiple classifications with the assumption of independence between features. NB assumes that each feature contributes independently to the possibilities assigned to a class. The NB classifier performs the analysis operations according to the formula below:

$$P(c|F) = (P(F|c)P(c))/(P(F)) \quad (3)$$

where P(c) and P(F) are the preliminary probabilities of events c and F, P(c|F) indicates the probability of event c occurring in the event of event F, P(F|c) indicates the probability of occurrence of event F when c event occurs. If P(F) probabilities are the same in all classes, it is aimed to maximize the dividend only. If P(c) probabilities are unknown, classes are assumed to be equal, and then we just maximize P(F|c). When many sets of data are given, computing P(F|c) will be computationally expensive. Reduction of the computational complexity in the evaluation of P(F|c)P(c) is only done with the naive assumption of class conditional independence using formula below:

$$P(c|F) \sim \prod_{k=1}^{n} P(c_k|F)) \quad (4)$$

It is partially more difficult to train the dataset with the NB algorithm, but it is a classification algorithm that works quite fast after training. It acts according to the condition of being the highest probability of a situation. The disadvantage is that the data is constantly changing. Because every new data will extend the training process (Kaya and Yıldız, 2014). Laplace smoothing was used in the Naive Bayes algorithm in this study and a parameter called lambda was used during the training as equaled to 1.

### *4.2. Decision Trees*

The second method that provides the most effective results among machine learning algorithms is the Decision Tree (DT) algorithms. It can be used for classification and regression.

A decision tree; consists of knot, branch and leaf. The top part is the root, the path from the root to the other nodes is the branch and the last result through these branches is the leaf (Kavzoglu and Colkesen, 2010). With this algorithm, a series of questions are asked to the data to be trained, and the results are reached in line with the answers obtained. While forming a decision tree, it is calculated with the information gain and information gain rate approaches according to which criterion or attribute value of the branch in the tree (Ozcan et al., 2020). DT is a variant of a greedy algorithm that progresses in the form of dividing and conquering in a top-down repetition, applying a set of decision rules (Man et al., 2018). In this algorithm, a tree structure is created, and class tags are expressed in the leaves of the tree. The last tree predicts the same tag for all samples that reach the leaf node. Each section is determined by choosing the best separation from the set of possible divisions to maximize knowledge gain in a tree node. When the split selected in each tree node is applied to the T dataset of a split v, the arguments necessary to maximize knowledge gain are obtained by calculating IG (T, v). Here, two different measures (Gini impurity and entropy impurity) are proposed for classifying the dataset (Man et al., 2018). Gini impurity:

$$\sum_{a=1}^{C} f_a(1 - f_a) \quad (5)$$

is calculated as. Here, C is the number of unique tags, and $f_a$ frekans is the frequency of tag a in a node. The impurity measure defined for entropy is as follows:

$$\sum_{a=1}^{C} -f_a \log(f_a) \quad (6)$$

Information gain is based on subtracting the main node impurity from the weighted sum of the two sub-node impurities. Information gain is defined as follows:

$$IG(T,v)=Gini(T)-\frac{N_{left}}{N}Gini(T_{left})-\frac{N_{right}}{N}Gini(T_{right}) \quad (7)$$

Here, the data set T with size N is obtained by dividing the sections and the terms $T_{left}$ and $T_{right}$ in sizes $N_{left}$ and $N_{right}$. In this study, the maximum number of divisions for decision trees was chosen as 10. Maximum number of bins used for splitting features was chosen as 32.

### *4.3. Random Forest*

The Random Forest (RF) method is one of the most successful machine learning models. RF is a community learning algorithm that comes together by decision trees to solve supervised learning tasks such as classification, has a good tolerance to noise and does not tend to over-sleep. Compared to the NB and DT approach, it provides much higher performance classification results. It combines multiple decision trees by producing stronger models to get a more accurate and stable estimate. The algorithm creates a model of multiple decision trees based on different data subsets using a random data sample during the training phase. This randomness constitutes an advantageous feature of the random forest model, which makes it more robust than a single decision tree and overcomes the problem of traditional data being overly compatible and similar (Man et al., 2018).

Overcompliance is defined as the model's over-learning and memorizing data while training on data. The RF approach generates and trains random subtrees from the dataset and feature vectors to overcome the problem of over-adaptation, a disadvantage of the DT approach. In this structure, each of which consists of different decision trees, the classification process is realized through the estimates with the highest votes.

The information gain (BK) and Gini index obtained using the attributes b to divide the sample set T are shown by the node division formula given below (Cortes and Vladimir, 1995):

$$BK(T,b)=Ent(T)-\sum_{n=1}^{V}-\frac{|T^{(n)}|}{|T|}Ent(T^{n}) \quad (8)$$

$$Gini(T,b)=\sum_{n=1}^{V}-\frac{|T^{(n)}|}{|T|}Gini(T^{n}) \quad (9)$$

Here ($T^{n}$) shows that in n branch node it contains all instances in T with the value of $b^{n}$ in the b attribute. The number of trees in the random forest used for training was determined as 10. The classification number in the algorithm is determined as 6.

### *4.4. Training and Classification*

In general, a supervised learning process consists of two stages: training and classification. The images to be classified are divided into two, as a training and test data at a predetermined rate. The training set consists of images used to train the machine learning classifier. In this approach, features are extracted from the blocks of these training images and combined in a hybrid feature vector. Before starting training, the created feature vector goes through normalization processes. The correct classification of each block during training is also provided through binary masks (Figure 4. (b)), in which the blocks are correctly labeled in advance.

After the image data is trained with machine learning algorithms, the classification processes are performed in the test set created by the images used for evaluation. The classifiers created during the training estimate the correct classification of the blocks in these test images and classify them as ship or non-ship blocks. Ship blocks and non-ship blocks are represented by white pixels and black pixels, respectively. The results of the classification analysis on the test images were examined by three different machine learning algorithms and evaluations were made. High performance in classifications made by these algorithms depends on the block size selected for analysis. The smallest blocks allow much more detail to be considered than the images. Compared to small block sizes, classifications using larger block sizes show less performance. Besides, factors such as brightness of the images, whether it is bright due to weather conditions, it is considered as a disadvantage in the classification stage that camouflage of the sea ships using shades similar to the colors of the sea to prevent them to be watched mostly by enemy forces. This disadvantage is solved by using texture features in addition to the features obtained from color.

## 5. Experimental Studies

In this study, a hybrid vector was obtained by extracting color and texture features from image contents. The length of the hybrid feature vector is 28x1. The results and classification success are evaluated through the different block sizes with the following tables and graphs. By creating the vector in different sizes, only the color spaces were first evaluated, then only the texture properties were evaluated, and classification procedures were performed. The classification results using different feature sizes were also evaluated.

When ship images with dimensions of 768x768 are divided into 16x16 block sizes, a total of 652032 block images are obtained for 283 images while 2304 block images are obtained from one image. When the same image data is divided into 32x32 block sizes, 163008 block images are obtained, and when they are divided into 64x64 block sizes, 40752 block images are obtained. In total, analysis operations were performed with 855792 block images obtained from 283 images.

The dataset consists of 283 images shared by the Airbus Company publicly. The dimension of the images was 768x768 pixels and stored as a jpg format. The training percentage of the data was determined as 70%, and the test percentage was 30%. The results of this study are presented in Table 4 and Table 5. The success of classification operations using the feature vector and label vector were calculated using the MLlib library of Apache Spark in the Eclipse Oxygen version environment. The results are taken from the GNU/Linux operating system distributions in Ubuntu 16.04 environment. Figure 5 shows example images from used in this study.

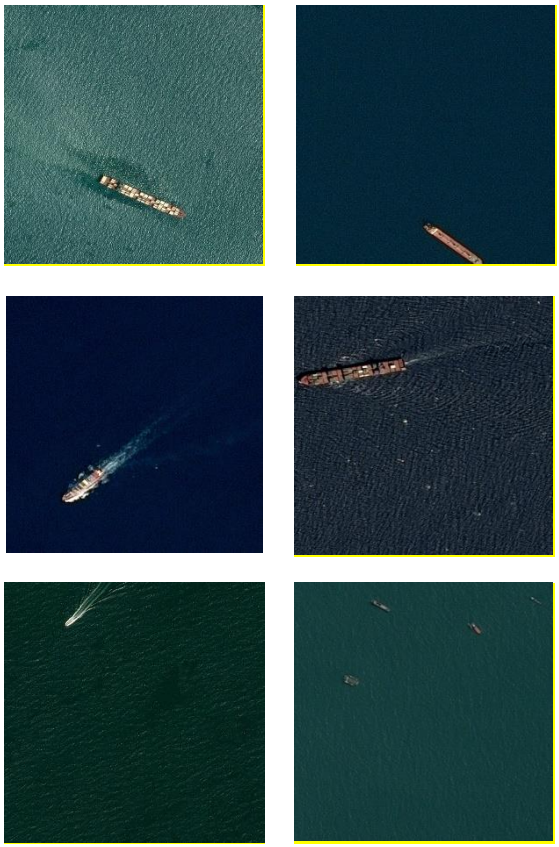

**Figure 5.** Example of ship images

As seen in the Figure 6, the best classification success has been achieved with RF algorithm. As the block sizes decrease, it gives much more successful classification criteria in DT and RF algorithms. When the graphic is analyzed, it is seen that the Random Forest approach gives the most successful accuracy. Again, according to the graph, Decision Trees are more successful than Naive Bayes approach.

In order to evaluate the classification performance criteria used in this study from a different perspective, the classification success was tested by dividing the hybrid vector created into several feature bases. For this, firstly, classification results were obtained with the vectors created in each color space. Then, SD and FS features were added to each color space vector and results were obtained. Lastly, GLCM features were added to measure which vector was obtained with a better classification result. The results obtained in these experiments are as follows:

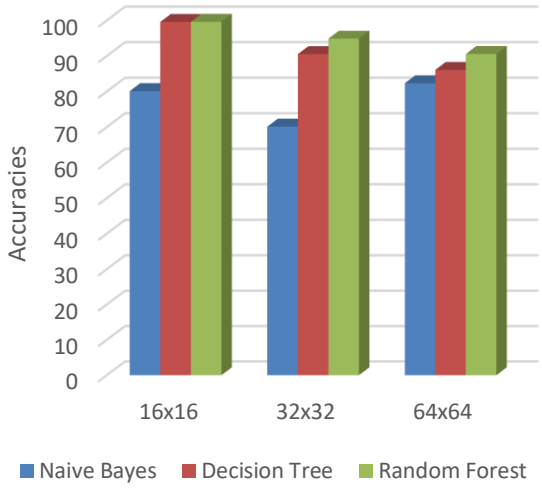


**Figure 6.** Accuracy results of ML algorithms in three different block sizes with the hybrid feature vector (%).

According to Table 4, the 28x1 dimension hybrid feature vector is divided into parts on ten different feature bases, which features contribute to the dominant degree of classification. According to the table, the vector with the highest performance in 3 different classification algorithms is our hybrid vector. In addition, although the classification performance decreased when trained with Naive Bayes algorithm, RGB+SD+FS, HSV+SD+FS, LAB+SD+FS vectors, it is seen that the classification success does not decrease below 94% when the RF algorithm is trained with all ten vectors.

**Table 4**. Accuracy results (%) of 3 different classification algorithms with piecewise feature vector.

| Features | NB | DT | RF |
|---|---|---|---|
| RGB | 95.03 | 96.60 | 97.59 |
| HSV | 98.72 | 98.75 | 98.84 |
| LAB | 93.36 | 94.03 | 94.38 |
| RGB+SD+FS | 95.62 | 98.64 | 98.72 |
| HSV+SD+FS | 93.72 | 98.63 | 98.70 |
| LAB+SD+FS | 94.39 | 98.52 | 98.81 |
| RGB+SD+FS+GLCM | 95.45 | 97.33 | 98.20 |
| HSV+SD+FS+GLCM | 97.12 | 97.83 | 98.93 |
| LAB+SD+FS+GLCM | 95.29 | 98.01 | 98.65 |
| Our Hybrid Vector | 80.12 | 99.58 | 99.62 |

After that, classification was made on image test data using Apache Spark clustering architecture. In this classification using the master worker architecture, the analysis of different image data sizes with different master-worker architecture was evaluated. The variation

of the Spark speed performance by data size and number of workers was observed. Time results developed using three different methods were evaluated. From left to right, as the number of workers decreased, the test processing time increased. This shows that parallel architecture is important in terms of time in big data processing.

In Table 5, Spark clustering architecture consist of master and workers has seen using three machine learning algorithms. It represents the vector number in million (M). The variation of the Spark speed performance by data size and number of workers was observed. The obtained results are presented in Table 5. Time results developed using three different methods were evaluated. From left to right, as the number of workers decreases it was observed that the test process time increased. With the NB algorithm, it is seen that there is 3.5 times increase in speed between 1 master+4 workers and 1 master+1 worker on 18 M data. With the DT algorithm, it is seen that there is a 3.3 times speed increase between 1 master+4 workers and 1 master+1 worker on 18 M data. Also, it is seen that the RF algorithm has a speed increase of 3.3 times between 1 master+4 workers and 1 master+1 worker on 18 M data. Although the time difference between the data size decreases and decreases proportionally with the data, it is seen that multiple workers defined processes are always very fast. As the data size passes to each lower row, the speed increase is observed when the classification times are reduced to a certain extent for evaluation. When it comes to big data, it is much more reasonable to prefer cluster architecture.

**Table 5**. Classification times (milliseconds) with NB-DT-RF methods using Master (Ms) - Worker (Wr) clustering architecture

| Number of Vectors | Naive Bayes | | Decision Tree | | Random Forest | |
|---|---|---|---|---|---|---|
| | 1 Ms+1 Wr | 1 Ms+4 Wr | 1 Ms+1 Wr | 1 Ms+4 Wr | 1 Ms+1 Wr | 1 Ms+4 Wr |
| 18 M | 4939 | 1384 | 2549 | 763 | 25710 | 7680 |
| 15 M | 3999 | 1179 | 2519 | 692 | 21075 | 7054 |
| 12 M | 2972 | 1056 | 1925 | 617 | 17059 | 4400 |
| 9 M | 2345 | 663 | 1566 | 383 | 12754 | 4410 |
| 6 M | 1563 | 541 | 1134 | 315 | 8610 | 2454 |
| 3 M | 882 | 271 | 596 | 199 | 4419 | 1398 |

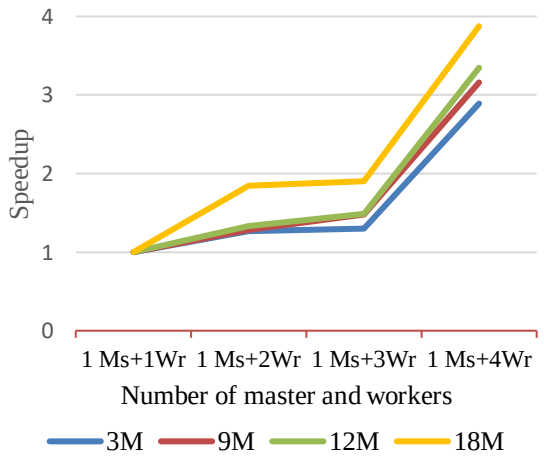


**Figure 7.** Classification testing phase speedups as a function of number of worker nodes

## 6. Conclusions

The aim of this study is to classify short distance images between ship and non-ship blocks using machine learning methods. The highest achievement was the Random Forest method with a rate of 99.62 %. In the comparative study between three color areas evaluated, the HSV+SD+FS+GLCM feature vector achieved the highest performance rate. It has been observed that when color and texture features are used together, higher success is achieved. Higher success can be achieved by adding the shape feature to the color and texture features. Thanks to the vector formed by adding the shape feature, ships can be classified according to their shapes and sizes. Evaluating more complex features using different machine learning methods and extracting features with deep neural networks can affect the performance of this study and provide much more efficient results.


## Acknowledgment

This work was supported by the Scientific Research Projects Unit of Karabuk University under project number FYL-2019-2044.


## References


Bi, F., Hou, J., Chen, L., Yang, Z., Wang, Y., 2019. "Ship Detection for Optical Remote Sensing Images Based on Visual Attention Enhanced Network". Sensors, 8, 4634-4646, 2015.

Cavallaro G, Riedel M, Richerzhagen M, Benediktsson JA, Plaza A. "On Understanding Big Data Impacts in Remotely Sensed Image Classification Using Support Vector Machine Methods". *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8, 4634-4646, 2015.

Chen, Z., Chen D., Zhang Y., Cheng X., Zhang M., 2020. "Deep learning for autonomous ship-oriented small ship detection". Safety Science, 130.

Cortes C, Vapnik Vladimir. "Support-Vector Networks", Machine Learning, 20, 273-297 (1995).

Ergul M, Alatan AA. “Geospatial Object Recognition From High Resolution Satellite Imagery”. 2013 21st Signal Processing and Communications Applications Conference (SIU), Haspolat, Turkey, 24-26 April 2013.

Gonzalez RC, Woods RE, Eddins SL. *Digital Image Processing using Matlab*. New Jersey, Prentice Hall, 2003.

Guo H, Yang X, Wang N, Gao X. “A CenterNet++ model for ship detection in SAR images”, Pattern Recognition, 112, 2020.

Han J, Kamber M, Pei J. Data Mining: Concepts and Techniques. Waltham, MA, USA: Elsevier, Second Edition, 2006.

Kanjir, U., Greidanus, H., Ostir, K., 2018. “Vessel detection and classification from spaceborne optical images: A literature survey”. Remote Sensing of Environment, 207 (1-26).

Kavzaoglu T, Colkesen İ. “Karar Ağaçları ile Uydu Görüntülerinin Sınıflandırılması: Kocaeli Örneği”, Electronic Journal of Map Technologies, 2, 2010.

Kaya C., Yıldız O. “Makine Öğrenmesi Teknikleriyle Saldırı Tespiti: Karşılaştırmalı Analiz”. Marmara Fen Bilimleri Dergisi

Li, H. Chen, L., Li, F., Huang M., 2019. “Ship detection and tracking method for satellite video based on multiscale saliency and surrounding contrast analysis”. Applied Remote Sensing 13 (2).

Li Y, Zhang H, Guo Q, Li X. “Machine Learning Methods for Ship Detection in Satellite Images”.

Liu, Y., Cuı, H.Y., Kuang, Z., Lı, G.Q., 2017. ITM Web of Conferences, 12.

Man W, Ji Y, Zhang Z. “Image Classification Based on Improved Random Forest Algorithm”, 2018 the 3rd IEEE International Conference on Cloud Computing and Big Data Analysis, 20-22 April 2018, Chengdu, China.

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, L., et al. (2015). MLlib: Machine Learning in Apache Spark.

Morillas JRA, Garsia IC, Zolzer U. “Ship Detection Based on SVM Using Color and Texture Features”. 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 3-5 September 2015.

Nie T, He B, Bi G, Zhang Y, Wang W. “A Method of Ship Detection under Complex Background”. International Journal of Geo-Information, 2017.

Ogul IU, Ozcan C, Hakdaglı O. “Fast Text Classification with Naive Bayes Method on Apache Spark”. 2017 25th Signal Processing and Communications Applications Conference (SIU), Antalya, Turkey.

Ogul IU, Ozcan C, Hakdaglı O. “Text Classification with Spark Support Vector Machine”. 1. Ulusal Bulut Bilişim ve Büyük Veri Sempozyumu, Antalya, 2017.

Ozcan C, Ersoy O, Ogul IU. “Classification of SAR Image Patches with Apache Spark Using GLCM Texture Features”. International Conference on Advanced Technologies, 3rd World Conference on Big Data, İzmir, 28 - 30 Nisan 2018.

Ozcan C, Ersoy O, Ogul IU. “Fast texture classification of denoised SAR image patches using GLCM on Spark”, Turkish Journal of Electrical Engineering & Computer Sciences,28, 2020.

Shi H, He G, Feng P, Wang J. “An On-Orbit Ship Detection And Classification Algorithm for SAR Satellite”, IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July-2 August. 2019.

Wang C, Pei J, Wang R, Huang Y, Yang J. “A new ship detection and classification method of spaceborne SAR images under complex scene”, 6th Asia-Pacific Conf. on Synthetic Aperture Radar (APSAR), Xiamen, China, 26-29 Nov. 2019.

Wang N, Chen F, Yu B, Qin Y. “Segmentation of large-scale remotely sensed images on a Spark platform: A strategy for handling massive image tiles with the MapReduce model”. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162, 137-147, 2020.

Wang. Z, Yang T., Zhang H. 2020. “Land contained sea area ship detection using spaceborne image”. Pattern Recognition Letters, 130 (125-131).

Yang, X., Sun, H., Fun, K., Yang, J., Sun, X., Yan, M., Guo, Z., 2018. “Automatic Ship Detection of Remote Sensing Images from Google Earth in Complex Scenes Based on Multi-Scale Rotation Dense Feature Pyramid Networks”. Remote Sens, 132, 10.

Yuan, Q., Shen, H., Li, T., Li, Z., Li, S., Jiang, Y., Xu, H., Tan, W., Yang Q., Wang, J., Gao, J., Zhang, L., 2020. “Deep learning in environmental remote sensing: Achievements and challenges”. Remote Sensing of Environment, 241.

Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Medjahed I. “Spark: Cluster Computing with Working Sets”.