INTERNATIONAL
HELLENIC
UNIVERSITY

Rock Image Classification
Ore/Waste – Cut Off Identification
For Decision Making

Aikaterini Mantela

SID: 3308170011

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

DECEMBER 2019

THESSALONIKI – GREECE

Rock Image Classification
Ore/Waste – Cut Off Identification
For Decision Making

Aikaterini Mantela

SID: 33081700011

Supervisor: Prof. Rigas Kotsakis

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

DECEMBER 2019

THESSALONIKI – GREECE

Abstract

For the identification complexity of rock image classification, based on a certain cut-off grade, an automatic classification recognition of rock images method is proposed in this dissertation, which is a part of the MSc in Data Science at the International Hellenic University. The main topic of this research was the identification of Rock image Classification through images in order to use the results for further decision making. Basis of specific approach in face mapping images taken by mine geologists for traditional rock-mass characterization. Digital grey image processing of rock face mapping images is used for features extraction. Then features contains the process of knowledge extraction from images in Matlab and furthermore as the classification procedure in Weka, in which different classifiers have been trained and tested in order to classify Ore or Waste. Finally, the model output is the rock image classification. Hellas Gold Company, subsidiary of Eldorado Gold provided a sample of 600 images for the case study. Specifically face mapping images are from Olympias mine located in North-East Chalcidice Prefecture, Greece. For the experiment, the dataset is divided into 300 images as a training dataset in order the algorithm to classify the >15% Ore and 300 images for classify the <15% of Waste. As an outcome the optimum classifier reached 98.5% accuracy for automatic identification of rock face mapping image. Therefore, the proposed method for improving geological pattern is effective and can result in accepted identification performance for rock image classification quickly and accurately. Nevertheless, the target is to reach an autonomous level such that no human intervention will be necessary.

Acknowledgement

I would like to acknowledge my thesis supervisor, Prof. Riga Kotsaki, for his constructive comments and discussion which significantly support me to execute this experiment. My colleague, Paraskeui Christodoulou, for inspire me to choose this project with Rock images and her offered expert advice on Image Analysis. My colleague Emmanouil Tzimtzimis for helping me with Matlab Neural Network, in related work. Hellas Gold - Eldorado Gold Company and specially Ms. Eleutheria Vagli, Mr. Giorgos Papakonstantinou, Mr. Ertan Uludag and my manager Konstantinos Bastis for his confidence and encouragement to execute this project.

Aikaterini Mantela, December 2019

CONTENTS

1. INTRODUCTION

The scope of this thesis is to deal with general classification problems using images. A case study in an operating underground mine, named Olympias Mine and is located in North-Earth Khalkidhiki Prefecture, Greece. Classification is the grouping objects in some orderly and logical manner into compartments (X.S Wei,2014).

Rock classification is applied in order to identify the type of an ore deposit. It is one of the primary steps accomplished, identifying the original ore environment of a mine. Rock images are usually explored by visual sense of a geologist based on experience, but human factors result in inaccuracies. For example, the time of observation is too short, there is a certain error and the different standard in the human identification (L.Lepisto,2006).

The procedure the geologists follow, is Ore Types/Complexity using a coding scheme defined to better assess Au and base metal distribution patterns. Eight standard rock classification types need to be chosen for preliminary classification of rock. by D. Rhys et al, 2013 (C.R.Shiron,2018). Based on the computer techniques such as artificial neural network, it can undertake the automatic classification recognition of rock images and help the geologists quickly analyze a large number of rocks images and gives the cut-off grade percentage.

Image 1: The Case Study of Olympias Mine located in North-East Chalcidice Prefecture, Greece.

The case study aims to clarify the use of the rock classification autonomous tool, to apply the proposed approach to underground mine optimization and grade risk assessment to one of Eldorado Gold's mining projects, which was the company that provided the dataset (Hellas Gold, Eldorado Gold).

Mineralization at the Olympias deposit occurs in lenses of replacement styles, manto-like, sulphidedominat mineralization and siliceous sulphide replacement and breccia mineralization within, and along the contacts of the Olympias marble unit, which is in turn hosted by amphibolite grade biotite-quartz-feldspar gneiss. Mineralization is structurally late in timing and is superimposed on the metamorphic fabrics in the area and in association with an extensional, brittle to semi-brittle fault network that was likely active coevally with the ore-hosting Stratoni Fault to the south. Further description of the deposit geology, its structural controls, and illustrations of different ore textures and types are in Rhys, 2013 (C.R.Shiron,2018).

Cut-Off grade is the minimum grade required in order for a mineral or metal to be economically mined (M. Bootsma). Material found to be above this grade is considered to be ore, while material below this grade is considered to be waste. In our case study the Cut-off grade is 20%.

However, the experiments were implemented in Matlab (The MathWorks, 2017), where feature extraction was made from images that have been manually captured and then the results were used for classification using the Weka platform in order to calculate the accuracy of different models (Hall et al., 2009).

Specifically, this research contains the following. Section 1 is a brief introduction in rock image classification and the related topic of this paper, which is the identification of a grade cut - off . Ore of Waste, through rock images. All the related work from other researchers, who have done a similar research in rock image - identification can be 9 found in section 2. Moving forward to section 3, the used dataset is presented and in section 4 there is a description of the whole process from the conception of the idea to its realization through the process and all the experiments that have been carried out. In addition, section 5 and 6 include the feature extraction in Matlab and the classification process in Weka respectively, while section 7 summarizes the entire research. Finally, section 8 is a discussion regarding suggestions for future analysis and the implementation of an autonomous rock image tool.

2. RELATED WORK

Rock type classification is of great importance in many stages of mine operations. In the literature review, several approaches to the rock image analysis were found and a large number of different kinds of proposed algorithms. In the case of mining, rock classification plays a crucial role at different stages of the extraction process ranging from the design of the mine to mineral grading and floatation plant (Chatterjee et al., 2010b). Characterization of the constituent rocks of an ore deposit including gangue material could be useful in the selection of the required equipment for excavation, and specially the strategies for blasting, among others. From a geological point of view, rock classification is useful for understanding the local properties of the ore deposit that determine the mine design.

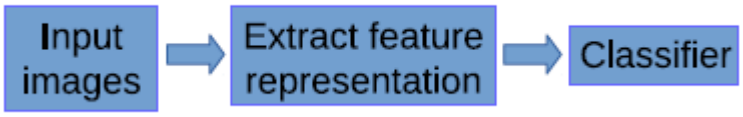


Figure 1: The typical framework of image classification , (Lei Shu, 2017)

In the case of image classification, as a typical framework showed in Figure 1, includes feature extraction representation for input images and feeding the feature that representation into a classifier. In general, authors present the performance of image classifiers if is heavily dependent on the selection of a feature representation. Unfortunately, rock textures are seldom homogeneous. As a result, the design of a feature representation is difficult, which makes rock image classification extremely challenging. There have been a few attempts at developing feature representation for rock image classification to date. From previous works use either hand engineered features manually selected for the specific application, or automatically selected features chosen using time-consuming methods. (Lei Shu, 2017)

Moreover, prior works mostly involve manually selected features. In order to reduce the time - consuming process of manual identification of rock samples, Ślipek and Młynarczuk (2013) and Młynarczuk and Górszczyk (2013) conducted autonomous classification of microscopic images of rocks by four pattern recognition methods - nearest neighbour, knearest neighbours (k-NN), nearest mode, and optimal spherical neighbourhoods. Sharif et al. (2015) built a small library of grayscale images from a total of 30 hand samples, and used Bayesian analysis to classify them with selected Haralick textural features (Haralick et al., 1973).

According to the same paper, in order to distinguish adjacent outcrops, Francis et al. (2014a) started with some fundamental visual "channels" such as colour and difference between colour channels, then utilized multi-class linear discriminant analysis (Multimedia Data Analysis) to identify the principal visual components. Harinie et al. (2012) utilized Tamura features (Tamura et al., 1978) to classify hand samples of rocks into the three major categories, namely, igneous, sedimentary and metamorphic. Dunlop (2006) studied features such as shape, albedo, colour and textures, then conducted rock classification with different feature combinations. Singh et al. (2004) compared seven (7) well-established image texture analysis algorithms for rocks classification and the results suggested that Law's masks (Laws, 1980) and co-occurrence matrices (Haralick et al., 1973) were best. Lepistö et al. (2003) classified rock images by methods based on textural and spectral features. The spectral features are some colour parameters and the textural features are calculated from the co-occurrence matrix.

Then, in order to improve the classification accuracy, Lepistö et al. (2005) combined colour information in Gabor space (Tou et al., 2007) to the texture description. Given that various visual descriptors extracted from images are often high dimensional and nonhomogenous, Lepistö et al. (2006b) conducted rock images classification based on k-nearest neighbour voting, which combined k-NN base classifiers for different descriptors by voting.

After that, a similar idea of combining base classifiers came to Lepistö et al. (2006a). Each feature descriptor had a corresponding separate base classifier, and better classification accuracy can be achieved by combining opinions provided by each base classifier.

As mentioned before, authors concentrated on feature selection. Chatterjee (2013) used the genetic algorithm to select features, and try to classified limestone with multi-class SVM (Support Vector Machine). Shang and Barnes (2012) utilized a reliability-based method and mutual information to select features, then classified rocks images in a more general dataset. However, both works showed that their own feature selection methods worked well in their dataset, but feature selection itself is time consuming. (Lei Shu, 2017)

In addition, when the dataset becomes complicated, one might have to think of what kind of feature pool to select from, or even devising a brand new feature representation. All the previous representations used for rock images consist either of an entirely manually crafted feature set or a set of features automatically selected from a set of manually crafted features. These manual features are not good enough to represent inhomogeneous rock images and are time-consuming to get. (Lei Shu, 2017)

Another paper also describes rock classification or characterization and is performed visually by mineralogists or geologists. However, a more sophisticated method for mineral identification for ore grading is done by collecting and chemically analyzing rock samples in a laboratory (Chatterjee et al., 2010b). Because of the time needed for the chemical analysis, it is not possible to perform it online. Therefore, a faster sensing system is desirable to achieve online estimation of rock composition. This could be possible with a machine vision system since visual classification of rocks is carried out by humans. Machine vision in the mineral industry has been applied in several mining operations such as online inspection of crushed aggregates (Al-Batah et al., 2009), online ore sorting and classification (Casali et al., 2001; Chatterjee et al., 2010a, 2010b; Guyot et al., 2004; Perez et al., 1999; Singh and Rao, 2006; Tessier et al., 2007), particle and blast fragment size estimation and/or distribution (Al-Thyabat et al., 2007; Hunter et al., 1990; Koh et al., 2009; Petersen et al., 1998; Salinas et al., 2005; Thurley and Ng, 2008), and froth monitoring (for a complete review see Aldrich et al. (2010), and others).

However, authors like Tessier et al. (2007) presented a very complete study of an online automatic ore composition estimator mounted on a pilot plant. They studied five ore types from Raglan's mine in Canada and used principal component analysis (PCA) and

wavelet texture analysis (WTA) for color and texture feature extraction, respectively. They obtained promising results in both dry and wet rock images. The studies presented by Chatterjee et al. (2010a, 2010b) analyzed minerals coming from two different deposits of limestone and iron. Although very similar approaches were used in both cases. They selected a segmentation algorithm from several tests and then applied morphological, textural and color feature extraction. PCA was used to reduce the feature vector and a neural network was used for classification. Figure 2 below illustrated the block diagram for rock composition estimation (Claudio A. Perez, August 2011).

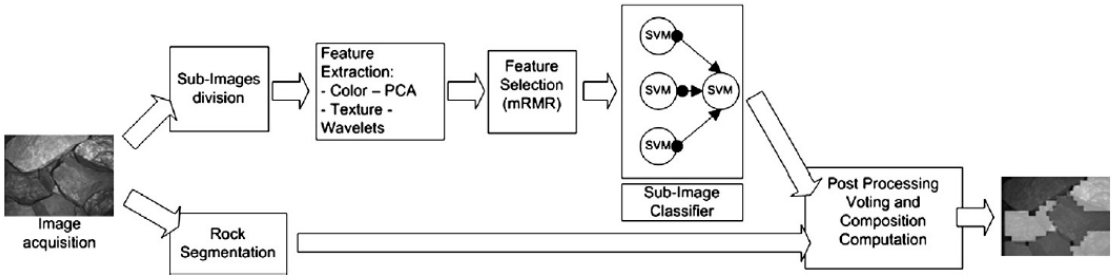


Fig. 2. Block diagram for rock composition estimation(Claudio A. Perez, August 2011).

Other rock classification systems were reported by Paclik et al. (2005) who used local texture information with co-occurrence likelihoods to build an industrial rock classification system; Lepisto et al. (2005) applied Gabor filtering to different color spaces for the classification of natural rock images. (Claudio A. Perez, 2011)

According to the implementations above, some other authors, Linek et al. (2007) combined Haralick features and wavelet analysis for classification of rocks in electrical borehole wall images which are used in the exploration of ocean basins and the ocean crust by drilling, Kachanubal and Udomhunsakul (2008) utilized a neural network combined with PCA and applied spatial frequency measurement to separate 26 stone classes, Donskoi et al. (2008) modeled and optimized a hydrocyclone using optical imaging and texture classification, Murtagh and Starck (2008) used up to fourth order moments of wavelet and curvelet transforms to classify images of mixture aggregate , Goncalves et al. (2009) presented a study for classification of macroscopic rock texture based on a hierarchical neuro-fuzzy model and Singh et al. (2010) developed an application of image processing on basalt rock samples where parameters are input to a neural network for classification. (Claudio A. Perez, 2011)

3. DATASET

3.1 Rock Image Characteristics

The domain of this coursework was to proceed into a general classification problem using images. A case study in a real mining industry in Greece, Olympias mine – Chalkidiki.

- Classification is the grouping objects in some orderly and logical manner into compartments.

- Object Recognition: is a process for identifying a specific object in a digital image or video. Object recognition algorithms rely on matching learning, or pattern recognition algorithms using appearance-based or feature-based techniques.

What a geologist does before mining?

- Capturing images from mining faces
- Mapping the Geological face
- Extracting the % Grade Cut off – Ore/Waste of the face
- Ore is a natural occurrence of rock that contains sufficient minerals with economically important elements
- Waste is unwanted material left over from a production process, or output which has no marketable value.
- Cut-Off grade is the minimum grade required in order for a mineral or metal to be economically mined. Material found to be above this grade is considered to be ore, while material below this grade is considered to be waste.
- In our case study the Cut off grade is 15%

(a)  (b) 

Image 2: (a): Initial Photo, Ore >15% (b): Initial Photo, Ore <15%

3.2. Dataset description

The Initial dataset consisted of 1000 images of geological face mapping of Hellas Gold Mining Company. The used camera was a DSLR Nikon 5100. The photo shoot was held in August 2019 without filters. However, the final used dataset included only 600 pictures. The rest of them were excluded due to machineries equipment, bad lighting conditions, image date stamp within them, images marked with highlighted spray regarding the mining operational procedure cycle. The challenge was to train an algorithm to classify an image as Ore or as Waste, without human interaction.

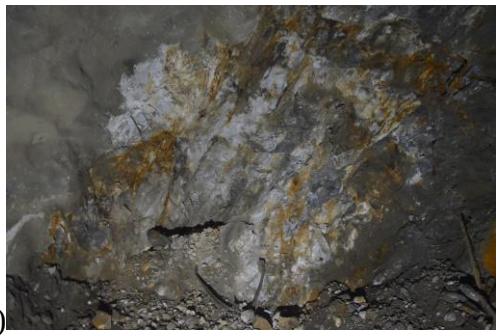
(a)  (b) 

Image 3: (a): Initial Photo, Ore >15% (b): Initial Photo, Ore <15%

However, from the sample of 600 images were used, which 300 of them have Ore >15% and 300 images do not, which automatically classifies the dataset as unbalanced since the two classes does not have the same amount of images.

Table 1: The Total Sample Images

Classify as Ore [1]	**Classify as Waste [0]**
300	300

Ore is a natural occurrence of rock that contains sufficient minerals with economically important elements. The mineralization of Mixed Sulfides in this case study is composed of:

- Au: it is part of the internal structure of the Arsenopyrite and Arsenian Pyrite, free gold is very rare
- Pb, Ag: are associated with the mineral Galena
- Zn: is associated with Sphalerite

Learn to Classify as Ore – Class [1] all underground mine images above > 15%

Waste is unwanted material left over from process or output which has no marketable value. In specific case study waste composed by:

- Marble: Grey to grey white, thin-banded, coarse crystalline, massive, silicified, medium to slightly fractured.
- Gneisses: Biotite Gneisses (Bt+Qtz+Kfs+Pl), ortho/para-gneisses. Amphibolite phases peak metamorphism/ retrogressed to green-schist phases. Primary gneissic foliation (S1) is folded by at least 2 major incidents.
- Pegmatites: Leucocratic (rarely including biotite mainly removed from the host Gneisses). Multiple stages of intrusions, sin-kinematic to post kinematic. Intruding both marble and gneisses.

Learn to Classify as Waste – Class [0] all underground mine images below the <15%.

Moving Forward, the classification Ore types that geologists used presented in the table below:

Table 2: Eight (8) Standard Rock Classification Ore Types

A case study of Olympias Mine Geologists Pattern Recognition	
Ore types	**Mineralization**
1	**Galena > Pyrite mineralization**
3	**Mixed Pyrite – Sphalerite – Galena - Arsenopyrite**
7	**Arsenopyrite - rich mineralization - major Au source**
2	**Pyrite - dominant mineralization**
6	**Grey siliceous, often breccia mineralization, arsenopyrite-bearing**
4,5,8	**Disseminated or veinlet mineralization in wall rock**

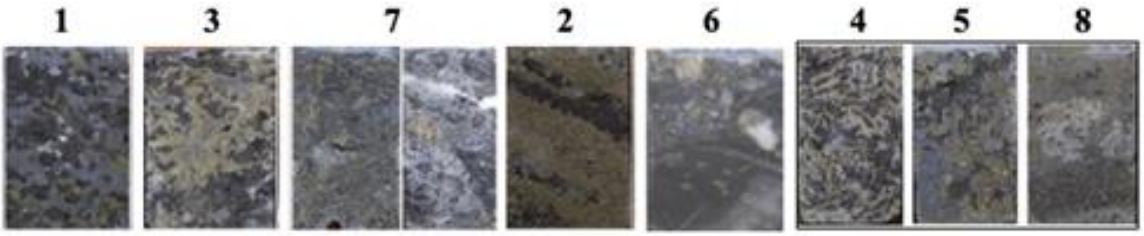


Image 4: Olympias Mine Geologists Pattern Recognition (Hellas Gold – Eldorado Gold)

4. TOPIC DESCRIPTION

In this specific topic, we presented a method for classification of real rock images. Due to complex and non-homogenous nature, the classification of them is a difficult task and in Rock Classification it is very important to be able to identify the rock from some meters distance from the face.

However, this research focuses on the first step, where the identification of the rocks takes part. In order to implement that, a dataset containing images from different mining faces from Olympias mine underground as mentioned in section 3. After creating the dataset, which has a primary material, it was time for the annotation part, where from geology face mapping by hand and based to their Grade cut-off percentage has to define which images contain ore and and which of them do not. With the help of Hellas Gold Mine Geologists and two other colleagues of mine, the annotation was done manually. Each picture was examined separately and two groups were created regarding the presence of the Rock, in which all the images were distributed.

From all the performed steps accomplished and based on the generated results, the following observations and suggestions are provided:

- The classification part should be further researched including a larger dataset. This will result in further model training and better model accuracy.

- In addition, images are proposed to be captured from better position in order to avoid picture distortion. Picture distortion results in lower accuracy.

- The same implies for objects in front of the rock face. For instance, machineries equipment, bad lighting conditions, image date stamp within them, images marked with highlighted spray regarding the mining operational procedure cycle.

- Moreover, different kind of materials found at the surroundings boundaries of the face e.g. shotcrete gunite or construction materials. Construction materials

must be manually or automatically excluded from the picture in order to achieve better accuracy.

The images below illustrates some of the difficulties that face mapping images have had from objects and from mining operation cycle.

(a)

(b)

(c)

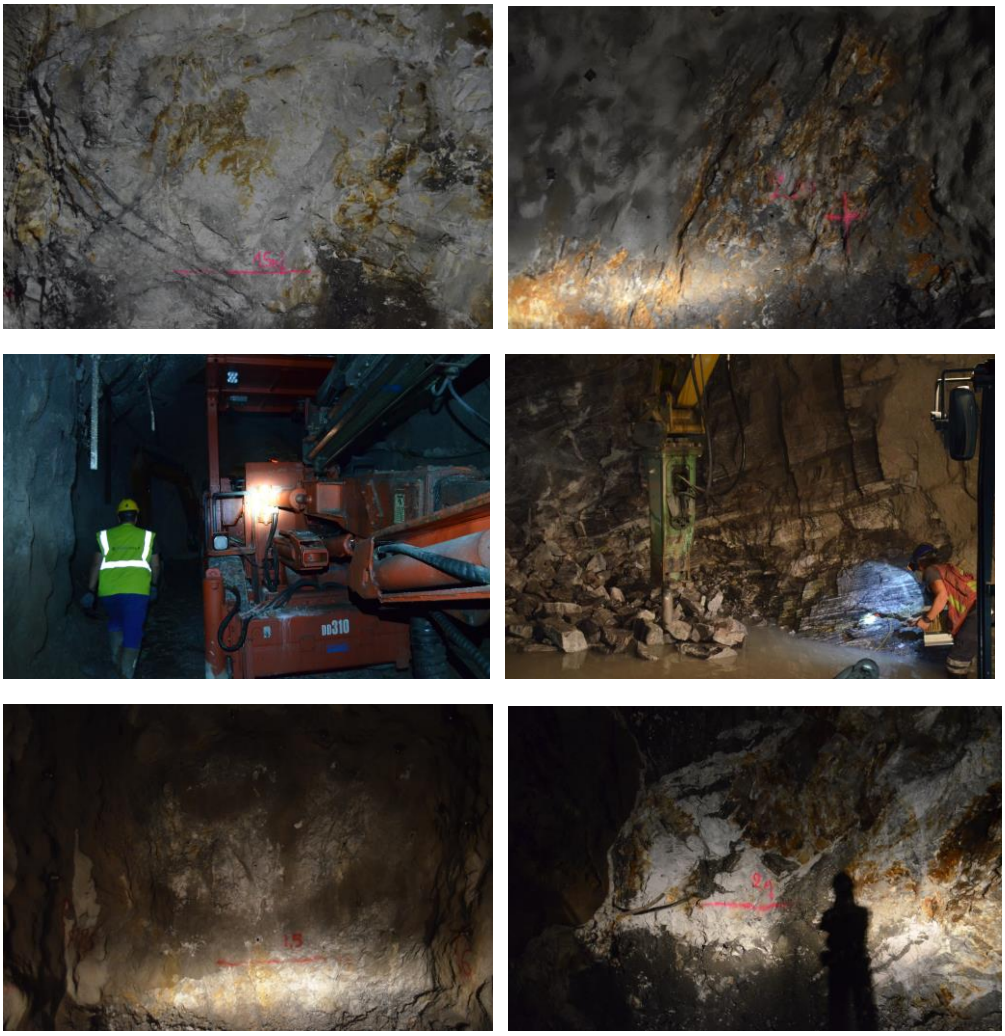


Image 5: (a): Face map with highlighted spray, (b): Parts of mobile machineries equipment and (c) bad lighting and shadows.

The next step was to extract some knowledge from the images. All images were transferred in Matlab, where the feature extraction was made. At first, only a part of the initial dataset was used, in order to navigate in a sample of them and try to understand which features would be more helpful and efficient. After all the sample experiments, the features that took place in the final process were the following:

- image contrast, homogeneity, correlation, energy, and entropy
- statistical measures, more specific mean and standard deviation

- the number of edges, as well as their average length
- the average length of lines
- the number of rectangles
- the number of white pixels in the images
- images histogram.
-

A more detailed explanation of all the above features will be presented in the next section.

Moving forward, all the above measures were used in the total sample and a final .xls file was extracted. The file contained all the features from both classes, Ore and Waste, which were distributed into columns and each column represented one feature. Also, each row represented one image and the last column contained 0s or 1s, whether the picture was Waste or Ore respectively. This file was the one that included all the data needed for the further research.

After that, it was time for the classification part, which was implemented in a classification tool, WEKA. All classifiers were tested for their ability to correctly identify whether a picture contains Ore or Waste. This ability was measured in terms of accuracy, precision, recall and F-score. At last, the classifiers with the best accuracy were further tested, in order to optimize accuracy levels.

5. FEATURE EXTRACTION PROCESS

Having prepare the dataset with all the useful and important images, it was important to apply some preprocessing commands or filters to them, in order to be able to extract features and then proceed to the machine learning techniques. There exist a lot of techniques on how to manipulate images before proceed with the machine learning algorithms. Therefore, various filters were tested. The whole image preprocessing phase was made in Matlab Software. Matlab stands for Matrix Laboratory, and is a numerical computing environment specialized for engineers and scientists. It uses matrix-based language that can be used for analyzing data, developing algorithms and creating models for machine learning applications (Mathworks, Matlab). Each of the following processes has generated different features that are described in the next chapter.

5.1 Reduce Haze

One of the most important image preprocessing phases was to make the images more visible and without any atmospheric haze within them. The reduce haze command in Matlab Environment was used and as Image 6(b) illustrates the images became more "clear" and the useful details were highlighted better than previously.

(a)  (b) 

Image 6: (a): Initial photo with Ore , (b): Same photo after Reduce haze

(a)  (b) 

Image 7: (a): Initial photo with Waste , (b): Same photo after Reduce haze

5.2 Image Resize

The next step was the resizing of all the images. The resize process was very important for two main reasons. Firstly, not all the images had the same dimensions, regarding the pixels, so the amount of data in every image was different. Moreover, the computational time needed to be reduced, since the size of the files was very large. For this reason, all the pictures resized to 400x600 pixels. These dimensions were carefully chosen, so to keep the same aspect ratio (width/depth) which was around 0.66 (mean value of initial pictures).

5.3 Grayscale Image

A grayscale image is an image that is described in a grayscale palette. Grayscale contains only the information of how bright or not the gray color is in every pixel (Technopedia, Grayscale). This means that the RGB colors are totally removed from the picture, and converted to a grayscale representation. The intensity values are between 0-255. In the image processing application, this conversion gives a lot of extra information, since there are a lot of features that can be extracted for further data exploration.

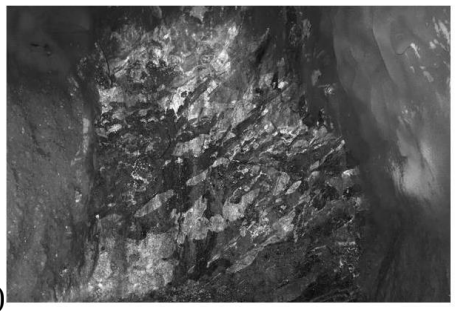
(a)  (b) 

Image 8: Grayscsle Image in (a) Ore and (b) in Waste

5.4 Binary Image

Comparing to the grayscale image, the binary one is, as the name implies, a digitized image consists of zeros (0) and ones (1). The zero numbers correspond to black pixels, while the "one" numbers are equal to white pixels (R.Kotsakis, 2015). The binary image is more efficient due to the less computational time it requires (M.H.Al. Amiri).

(a)  (b) 

Image 9: Binary Image in (a) Ore and (b) in Waste

5.5 Increase the RGB Colors

Every image is described by three components (Red, Green, Blue). Each color can be composed by these components, and it is a grayscale image with values from 0-255 for the brightness of the red, green and blue color respectively. By increasing the intensity of one color at a time, different image representation is succeeded and new features became available. For this experiment the intensity of its color was doubled. (twice the portion of red etc.).

5.6 Increase Contrast

The increase contrast process is alike the reduce haze command, but this time for the grayscale image, and not the initial full-colored one. If an image has a lot of values around specific intensity numbers (0-255), the contrast adjustment is distributing those numbers in larger field (more corresponding numbers). As a result, the highlights of the image become brighter (Mathworks, Matlab). The contrast adjustment can also be filtered with values from 0 (brighter) to 1(darker). After several alterations it was found out that the limit number for this experimentation were 0.2 and 0.5.

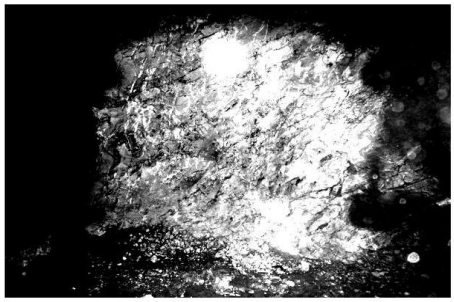
(a)  (b) 

Image 10: Increase contrast in (a) Ore and (b) in Waste

5.7 Histogram Equalization

The final preprocessing of the images was another method of increasing the contrast. The histogram equalization uses the histogram of each image and regenerates a new image by equalizing it around a probability distribution. The histogram itself defines the frequencies for all the intensity values in an image. In other words, it counts how many pixels have the same intensity value from 0 to 255 (A. Karnewar).

(a)  (b) 

Image 11: Histogram Equalization in (a) Ore and (b) in Waste

5.8 Features extraction

Following the image preprocessing phase, the features extraction was mandatory, since the machine learning algorithms cannot be applied without them. Some of those features extract morphological knowledge, while others extract statistical measures.

As described previously, the images have been regenerated in many ways so each one of them can be treated as different.

Therefore, some features originate from the grayscale and the binary images, while others from the increase contrast and the histogram equalization.

5.9 Feature Extraction from Grayscale Image

A specific set of visual modality properties can be produced by using the Gray Level Co-Occurrence Matrix (GLCM). The GLCM is a pair of numbers that corresponds to different relationships between a pixels and its neighbors within the image (M.Partio,). Each set of numbers describes different statistical features and can be stored as individual for further analysis. The four most important relationships where computed and each one turned into a unique feature.

5.10 Constrast

The contract property measures the intensity contrast between a pixel and its neighbor across the image (R. Kotsakis, 2015), and it is computed by using the following formula. Also, if the image is constant, the contrast number becomes 1.

$$C_F = \sum_{i=1}^{K}\sum_{j=1}^{K} |i-j|^2 \times glcm(i,j) \text{ where } K = 256 \text{ levels} \quad (1)$$

5.11 Homogeneity

Homogeneity measures the distances among pixels. It is calculated as shown below and it can take values from 0 to 1. In fact, it becomes 1 when the GLCM is diagonal, which means that image's texture is coarse enough. At last, it is calculated as follows.

$$H_F = \sum_{i=1}^{K}\sum_{j=1}^{K} \frac{glcm(i,j)}{1+|i-j|} \text{ where } K = 256 \text{ levels} \quad (1)$$

5.12 Correlation

Correlation, as its name implies, measures the correlation between a pixel and its neighbor and it takes values between -1 to 1. If the value is 1 means that the pixels are positively correlated and on contrary to that when the value is -1, it implies a negative correlation. However, the correlation value will not be available if the image is constant. The used formula is given below.

$$AC_F = \sum_{i=1}^{K}\sum_{j=1}^{K} \frac{(i-\mu i)\times(j-\mu j)\times glcm(i,j)}{\sigma_i \times \sigma_j} \quad \text{where } K = 256 \quad (1)$$

5.13 Energy

The energy feature measures the image energy and its result is the sum of all the squared elements in the GLCM. It takes values in the range [0,1] and specifically, it becomes 1 if the image is constant. Below is the calculated formula.

$$E_F = \sum_{i=1}^{K}\sum_{j=1}^{K} \text{glsm}(i-j)^2 \quad \text{where } K = 256 \text{ levels} \quad (1)$$

5.14 Entropy

Entropy is a statistical measure, which computes the randomness and can be used in order to describe the texture of an image. It is computed as follows.

$$\text{entropy}(X) = -\sum_{i=1}^{N}\sum_{j=1}^{M} X(i,j) \times \log_2 X(i,j) \quad (1)$$

The entropy can be also calculated in the colorful image for every element (RGB colors), but it was not considered important for the purposes of this research.

5.15 Mean Value and Standard Deviation

Finally, the statistic measures of the mean value and the standard deviation were computed and stored as two individual features. The mean value returns the mean intensity number of all the pixels, while the standard deviation computes the distance between each pixel and the mean value. (R.Kotsakis,2015) is related to the mean value, and to the standard deviation.

$$\text{mean}(X) = \frac{1}{N \times M}\sum_{i=1}^{N}\sum_{j=1}^{M} X(i.j) \quad (1)$$

$$\text{std}(X) = \sqrt{\frac{1}{N \times M}\sum_{i=1}^{N}\sum_{j=1}^{M} [X(i,j) - \text{mean}]^2} \quad (1)$$

5.16 Features Extrasion from Binary Image

After proceeding with all the features that are correlated with the grayscale image, the next step was to use the binary version of the images to extract useful features from them as well. There are four different features that are described as follow:

5.17 Number of Edges and Average

The first feature was the calculation of all the edges that exist in every image. This process has a lot of different alterations since there many algorithms that can be used in order to find the aforementioned edges. In this research, only the "Sobel" method was used. The "Sobel" method is an operator that performs a spatial gradient measurement and it is emphasizing in regions with high spatial frequency. These regions form the final edges (HIPR2). Additionally, the average length of the edges was listed as another feature. **Error! Reference source not found.** illustrates the detected edges of an image either it is ore or waste.

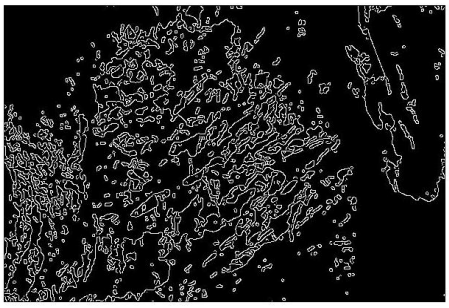
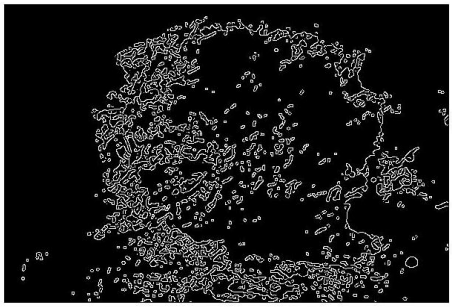
(a)  (b) 

Image 12: Left side (a) Edges in Ore Image, Right side (b) Edges in Waste Image

5.18 Lines

In the same way with edges recognition, another useful characteristic of the image is the total length of the lines. For the detection of the lines the Hough transform was necessary. The Hough transform uses the parametric representation of each line in order to detect it (Mathworks, Hough Transform) The formula for the Hough transform is the following

$$\mathrm{rho} = x \times \cos(\mathrm{theta}) + y \times \sin(\mathrm{theta})$$

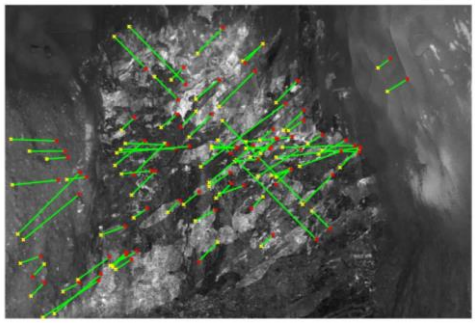
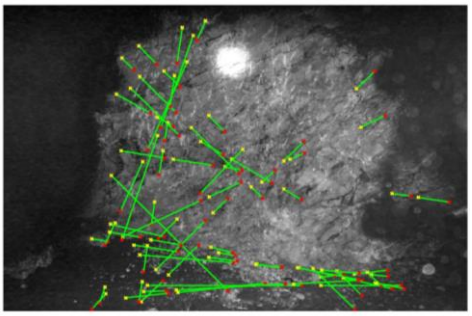
(a)  (b) 

Image 13: Left (a) Line Detection in Ore , Right (b) Line Detection in Waste

Where rho is the distance the line has from the origin and theta is the angle of this vector. Moreover, it was decided that only the 20 strongest Hough-peaks with distance between the detected lines up to 10 and with minimum line length of 20.

5.19 White Pixels

Due to the fact that the main goal was to identify if there is ore on the image, the total number of white pixels in the binary image was considered really important, and classified as a mandatory feature. The amount of black pixels could also be calculated, but it was redundant because it is supplementary to the summation of the white pixels.

5.20 Extra Statistical Features

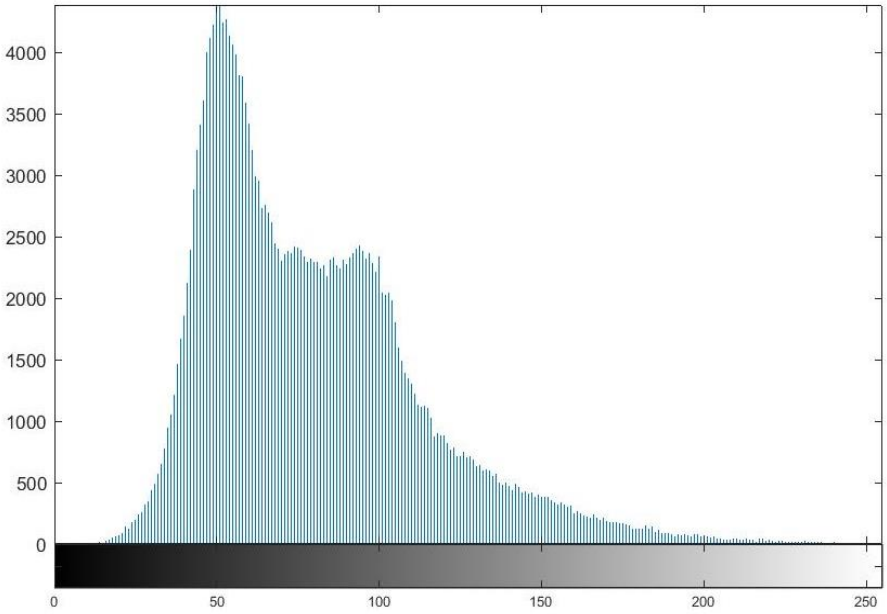
Besides the grayscale and the binary image, the rest of the image preprocessing phases were used only for some extra statistical measurements.

At the images where the intensity values of red, green and blue was doubled, the mean value and the standard deviation were computed. Moreover, the same statistical properties were also calculated for the image with the increased contrast and the image with the histogram equalization.

5.21 Histogram Mean Value

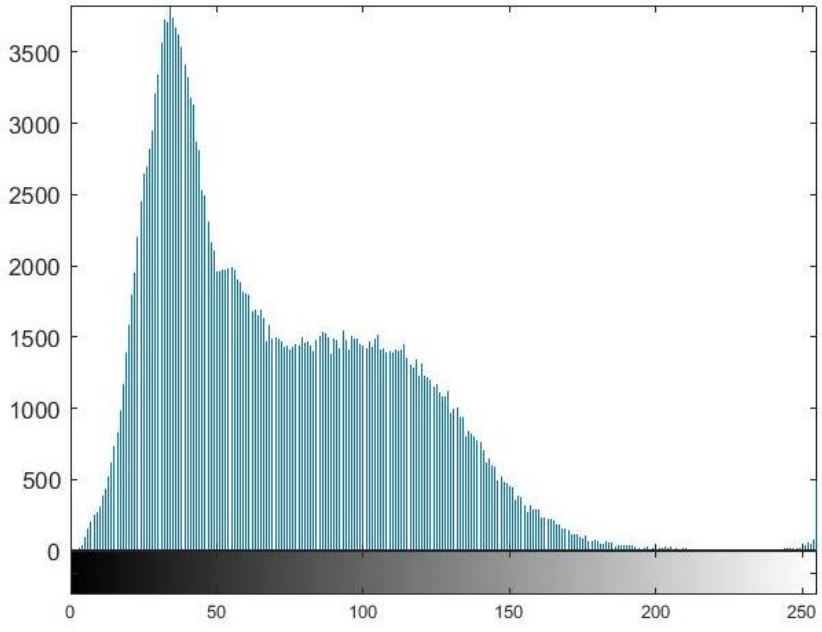
The final feature was related with the histogram representation of the greyscale image. As below graphs shows the histogram of the “ore” image, differs to the histogram of the

“waste” image only at the values between 25 and 125. Thus, the average amount of pixels in those intensity values stored as the final feature.



(a)

Graph 1 : (a) Histogram of Ore Image



(b)

Graph 2 : (b) Histogram of Waste image

6. CLASSIFICATION & EXPERIMENTAL RESULTS

The class label is the feature that describes if the image is classified as “ore” or “waste”. It is in binary format and takes the value one (1) if the image is “ore” and zero (0) if the image is “waste”.

Table 2: Features of the Dataset is a more comprehensive way to describe all the features that were extensively described on this chapter.

Table 2: Features of the Dataset

FN	Feature	Corresponding Image
1	Contrast	Grayscale Image
2	Homogeneity	Grayscale Image
3	Correlation	Grayscale Image
4	Energy	Grayscale Image
5	Entropy	Grayscale Image
6	Mean Value	Grayscale Image
7	Standard Deviation	Grayscale Image
8	Number of Edges	Binary Image
9	Average Length of Edges	Binary Image
10	Length of Lines	Binary Image
11	White Pixels	Binary Image
12	Mean Value	Double-Red Image
13	Standard Deviation	Double-Red Image
14	Mean Value	Double-Green Image
15	Standard Deviation	Double-Green Image
16	Mean Value	Double-Blue Image
17	Standard Deviation	Double-Blue Image
18	Mean Value	Increase Contrast Image
19	Standard Deviation	Increase Contrast Image
20	Mean Value	Hist. Equalization Image
21	Standard Deviation	Hist. Equalization Image
22	Histogram Average	Grayscale Image
23	Class	Initial Image

After the feature extraction process, it is time for the classification part. As mentioned before, all the features were listed in columns, where the last column was filled with the number 1 if the specific image has a grade cut off above 15% and 0 if not. Thus, it is a binary classification problem from which we want to predict if there is a grade cut off >15% or not on the images.

Moreover, the collected data have been modified in a way (.arff file) that they could be used in the WEKA platform, where the classification process has been implemented, while testing several algorithms. The WEKA tool (named by a bird from New Zealand) is a machine learning library, which is mainly used for data mining tasks (Waikato, Weka Classifiers). For the purposes of this thesis all the available algorithms were tested, in order to end up with the best algorithms that scored the highest accuracies using the given dataset (Appendix C).

As mentioned before, the main target was to predict the presence of Ore/Waste – Grade cutoff in an image. Thus, it was a binary classification problem, in which all classifiers, after the training part, should be able to predict if an unseen before image has Ore or Waste. All experiments involved either the k-fold cross validation process or the percentage split method. In both cases, one random section of the dataset is used in the training part and the rest as a test set. Specifically, all classifiers are instructed from the train set and produce results regarding the test set.

6.1. Attributes' Evaluation

Before any implementation in the classification process, the correlation attribute evaluation was extracted from WEKA in order to identify which features are more significant than others into the classification phase (Bouckaert et al., 2016). The results are presented in the image below and the extracted numbers were between 0.17 and 0.5807. Thus, at the beginning all attributes have been considered as significant ones and in case that the results would not be the expected ones, attributes would be evaluated again, according to the image below. However, according to the extracted results, this evaluation has never been processed.

```
Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 23 class):
        Correlation Ranking Filter
Ranked attributes:
 0.5807    5 ENTROPY
 0.5418   11 RECTANGLES
 0.5366   15 MEAN2_GREEN
 0.5167    6 MEAN
 0.5163   17 MEAN2_BLUE
 0.5144   16 STD2_GREEN
 0.4978   14 STD2_RED
 0.4884   18 STD2_BLUE
 0.4559    7 STD
 0.4422   12 WHITE_PIXELS
 0.4336    8 EDGES_NUMBER
 0.3661   19 MEAN2_GRAY
 0.3655    2 HOMOGENEITY
 0.3198    4 ENERGY
 0.2518   22 STD2_HISTO
 0.1601    3 CORRELATION
 0.0778   10 LINES
 0.0664    1 CONTRAST
 0.0379   13 MEAN2_RED
 0.0275    9 EDGES_AVGLEN
 0.019    20 STD2_GRAY
 0.017    21 MEAN2_HISTO

Selected attributes: 5,11,15,6,17,16,14,18,7,12,8,19,2,4,22,3,10,1,13,9,20,21 : 22
```

Figure 1: Correlation Attribute Evaluation

6.2. Classifiers' Description

The two most used machine learning tasks for predicting values are the supervised and the unsupervised learning. In supervised learning, models should be able to predict in which class new unseen before instances belong. In contrast to that, in unsupervised learning the main goal is to identify specific patterns (clusters) or extreme cases (outliers) from the given dataset and be able to cluster a new unknown instance in one of the already generated clusters. Nevertheless, in this research, the supervised learning method has been chosen, since the main goal was to correctly identify if an image has Ore or Waste. So, it was a binary classification problem and below, a brief description of some classifiers can be found.

6.2.1 Decision Trees

Decision trees are connected graphs, which includes nodes in different layers (Song & Lu, 2015). In order for a decision tree to be generated, a train part with known data from the dataset is mandatory, since they are used for the gradually configuration of the tree.

The rest of the dataset remains as a test set and evaluates the accuracy level of the tree. As it is obvious, in order for the tree to be more effective, the majority of the dataset should be used as the train set and fewer as a test set, Nevertheless, decision trees are presenting a high complexity in terms of their structure when the dataset includes many attributes and targets. In addition, trees are completely dependent on the given dataset, which could lead to a completely different structure of the tree and accordingly, completely different results, even with small changes in the initial dataset.

6.2.2 Statistical Regression

In statistical regression models, the predicted variable from the given dataset is a linear equation of one or more from the independent variables. Specifically, this method could be divided into different models which are presented below.

- **Linear Regression**

Linear Regression is calculated by the following function:

$$y = ax + b$$

where y is the dependent variable and x the independent variable, while a and b are two constants that are determined after the training part. For this determination, the model uses the least square method, in which a and b are chosen in a way that they minimize the sum of the squared differences between real and unseen output values. Since the above formula includes only one output variable, the simple linear regression is not the best choice in data mining tasks, since there are more than one dependent or independent values. A simple linear regression model is presented in the picture below with a simple linear function that minimizes the distance error from each instance.

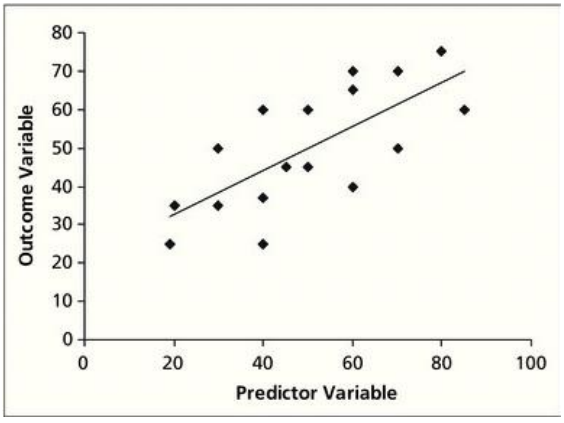


Figure 2: Simple Linear Regression (Worster, Fan, & Ismaila, 2007)

- **Multiple Linear Regression** The model uses the following formula:

$$y=a_3x_3+a_1x_1+a_2x_2+\cdots+a_nx_n+c$$

where y is the dependent variable and x_1, x_2, x_n are the independent ones. Also, a_1, a_2, a_n, c are the constants that are calculated in the training phase. In this case, a surface is produced instead of a line, which is also minimizes the distance error between instances and itself. In contrast to the simple linear regression, this model is suitable for data mining problems, since multiple independent variables could be implemented (Kotsakis, 2015).

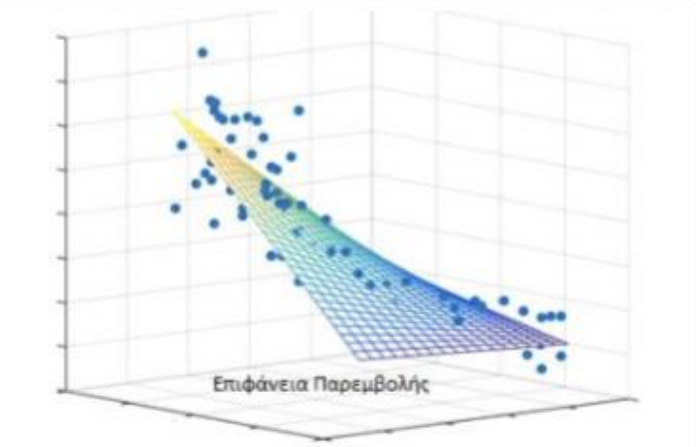


Figure 3: Multiple Linear Regression (Kotsakis, 2015)

- **Regression Trees**

Regression trees are just like decision trees, but their nodes consist of numerical instead of categorical values. Also, the value of each node could be calculated as the mean value from all the tree nodes until the presence node.

- **Logistic- Logarithmic Regression**

The logarithmic regression uses logarithms in order to generate results between values 0 and 1. The output variable is a conditional probability and it is calculated by the following formula:

$$y = \frac{1}{1 + e^{(a+bx)}}$$

where y is the dependent variable, x is the independent one and a and b are two constants that are determined during the training phase.

6.2.3 Bayessian classifiers

The calculated mathematical formula of Bayesian classifiers is the following Bayes theorem:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

where A is the independent variable and B the dependent one, while the probability P(B|A) is the probability of B given A. Also, probability P(A) and P(B) are the a priori probabilities of A and B respectively (Bouckaert et al., 2016).

6.3 Evaluation Metrics

Commonly Machine Learning experiments have been evaluated results of using Recall, Precision and F-measure. This measures for their origin were named Inforamtion Retrieval and present specific biases, namely that they ignore performance in correctly handling negative examples, the propagate the underlying marginal prevalences and biases, and they fail to take account the chance level performance.(Powers, D.M.W.,2011) In this experiment the results that have been evaluated were the Accuracy, the Precision, the Recall, the F- score and finally the time taken to build the model for each classifier.

Firstly the Accuracy, is the number of instances that have been correctly classified, which means that if an instance in Class Ore (or in Waste), it is also classified in class Ore (or Waste). Accuracy described with the following formula:

$$Accuracy = \frac{correctly\ predicted\ instances}{total\ instances}$$

According to the above, Precision is the number of instances that correctly identidied as Ore. thus classified as Positives divided by the total number of Ore images that the classifier has classified as Ore even if some of them are Negative. Precision, is given by the following formula:

$$Precision = \frac{correctly\ identified\ as\ ore}{total\ number\ of\ predicted\ as\ ore}$$

In addition, Recall is the number of instances that correctly identified as Ore divided of total number of Ore images. Recall, is calculated with the formula below:

$$Recall = \frac{correctly\ identified\ as\ ore}{total\ number\ of\ \text{ore}\ images}$$

F-Score, finally, is the geometric combination of Precision and Recall and is given by the following formula:

$$F - Score = \frac{2 \times precision \times recall}{precision + recall}$$

Moreover . for the binary case it is common to preserve the varius measures in the context of a dichotomous binary classification problem, where the labels are by convention + and the predictions of a classifier are summarized in a four-cell contingency table. The contingency table expressed using raw counts of the number of each predicted label is associated with real class, or may be expressed in relative terms.

Cell and margin labels may be formal probability expressions, may derive cell expressions from margin labels or vice-versa, may use alphabetic constant labels a, b, c, d or A, B, C, D, or may use acronyms for the generic terms for True and False, Real and Predicted Positives and Negatives. (Powers, D.M.W.,2011)

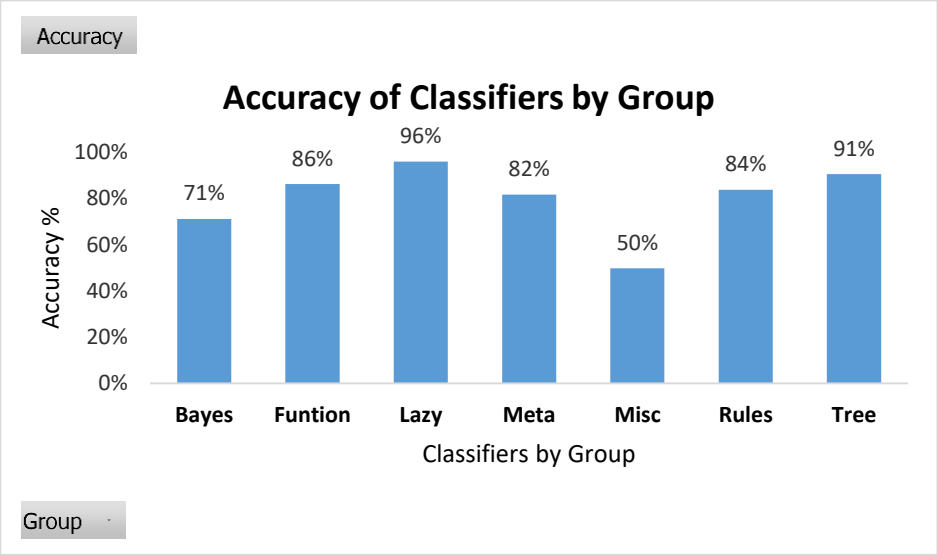
Table 3: Confusion Matrix - Systematic and traditional notations in a binary contingency table. Shading indicates correct (light=green) and incorrect (dark=red) rates or counts in the contingency table

	+R	−R	
+P	tp	fp	pp
−P	fn	tn	pn
	rp	rn	1

	+R	−R	
+P	A	B	A+B
−P	C	D	C+D
	A+C	B+D	N

6.4. Experimental results

The first part of the experiments was to evaluate all the given classifiers in Weka with the initial dataset and their default parameters. The process was implemented with the 10-fold cross validation and all the results are presented below in a descending order from the classifiers with the best accuracy to the ones with the worst one.



Graph 3: Accuracy of Classifiers by group

In this part, it is observed that 61.36% of all classifiers has an accuracy over 90% and only 18.18% of them between 60% and 89%, while 20,45%, nine (9) classifiers, presents a constant accuracy of 49.83%. The "winner" classifier in this round of experiments was the Random Committee, from the Meta group classifiers, with an accuracy level of 98,83%. In the second place, four classifiers, Multilayer Perceptron, Ibk, Kstar and Random Forest, from different groups, are shared the same percentage of 98.50%, but with significant differences on the other metrics, like precision, recall and F-score. After those classifiers, there are 22 more that has an accuracy over 90%. Also, in all the above procedures, the measure of time taken is less than or equal to 0.5 seconds.

Table 4: Classifier Results – Default Parameters – 10 fold Cross Validation

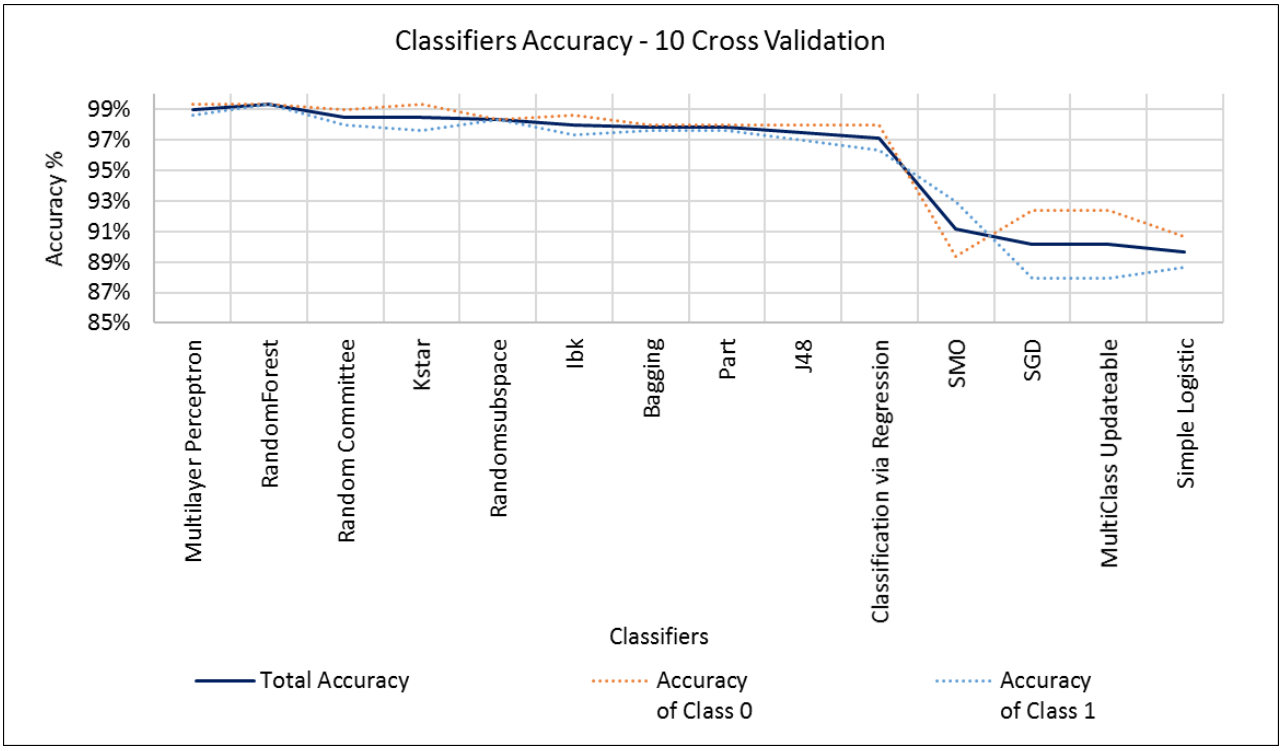
Group	Classifiers	Accuracy	Precission	Recall	F-Score	Time Taken
Meta	**Random Committee**	**98.8%**	**98.3%**	**98.3%**	**98.3%**	**0.0**
Funtion	**Multilayer Perceptron**	**98.5%**	**83.1%**	**79.8%**	**62.9%**	**0.2**
Lazy	**Ibk**	**98.5%**	**65.8%**	**63.2%**	**61.5%**	**0.5**
Lazy	**Kstar**	**98.5%**	**49.2%**	**49.8%**	**40.3%**	**0.5**
Tree	**RandomForest**	**98.5%**	**83.1%**	**79.8%**	**79.3%**	**0.2**
Meta	**Randomsubspace**	**98.3%**	**96.5%**	**96.5%**	**96.5%**	**0.0**
Funtion	**SGD**	**97.7%**	**98.5%**	**98.5%**	**98.5%**	**0.0**
Meta	**MultiClass Updateable**	**97.7%**	**97.7%**	**97.7%**	**97.7%**	**0.0**
Meta	**Classification via Regression**	**97.3%**	**49.2%**	**49.8%**	**40.3%**	**0.5**
Tree	**J48**	**97.2%**	**96.7%**	**96.7%**	**96.7%**	**0.1**
Meta	Bagging	97.0%	96.9%	96.8%	96.8%	0.0
Rules	Part	97.0%	71.6%	97.7%	66.1%	0.3
Funtion	SMO	96.8%	98.5%	98.5%	98.5%	0.0
Funtion	Simple Logistic	96.7%	98.5%	98.5%	98.5%	0.0
Tree	LMT	96.7%	90.8%	90.8%	90.8%	0.2
Funtion	Logistic	96.5%	93.9%	93.8%	93.8%	0.1
Meta	MultiClass	96.5%	96.0%	96.0%	96.0%	0.0
Meta	AttributeSelected Classifier	96.0%	97.0%	97.0%	97.0%	0.1
Tree	RandomTree	95.7%	97.3%	97.3%	97.3%	0.1
Meta	Filterclassifier	94.7%	49.2%	49.8%	40.3%	0.5
Meta	Itterativeclassifieroptimizer	94.3%	64.7%	94.7%	94.7%	0.1
Meta	Logit boost	94.3%	94.4%	94.3%	94.3%	0.1
Tree	RepTree	94.3%	94.4%	94.3%	94.3%	0.1
Meta	AdaBoostM1	93.8%	96.5%	96.5%	96.5%	0.0
Rules	Jrip	93.8%	97.7%	97.7%	97.7%	0.0
Rules	Decision Table	93.0%	49.2%	49.8%	40.3%	0.5
Lazy	LWL	90.8%	98.8%	98.8%	98.8%	0.0
Bayes	BayesNet	89.2%	89.6%	89.2%	89.1%	0.1
Rules	OneR	85.0%	98.3%	98.3%	98.3%	0.1
Bayes	Naïve Bayes	79.8%	49.2%	49.8%	40.3%	0.5
Tree	Decision stump	77.5%	49.2%	49.8%	40.3%	0.5
Bayes	Naïve BayesUpdateable	73.8%	49.2%	49.8%	40.3%	0.5
Tree	Hoeffding Tree	73.7%	49.2%	49.8%	40.3%	0.5
Funtion	Voted Perceptron	67.7%	93.1%	93.0%	93.0%	0.1
Bayes	Naïve Bayes Multinomial	63.2%	93.8%	93.8%	93.8%	0.1
Bayes	Naïve Bayes Multinomial Text	49.8%	85.2%	85.0%	85.0%	0.2
Funtion	SGDText	49.8%	97.0%	97.0%	97.0%	0.0
Meta	CVParameter Selection	49.8%	49.2%	49.8%	40.3%	0.5
Meta	MultiScheme	49.8%	81.8%	77.5%	76.7%	0.3
Meta	Stacking	49.8%	76.2%	73.2%	72.3%	0.3
Meta	Vote	49.8%	97.2%	97.2%	97.2%	0.0
Meta	Weighted instances	49.8%	96.7%	96.7%	96.7%	0.0
Misc	Input Mapped	49.8%	98.5%	98.5%	98.5%	0.1
Rules	ZeroR	49.8%	95.7%	95.7%	95.7%	0.0

The next step in the experimentation process was to keep only the classifiers with a total accuracy over 85% and test them again under different circumstances, in order to record their behavior more specialized in terms of accuracy of class 0 (Waste) and class 1

(Ore). The experiments that took place included three different k-fold cross validation processes and one percentage split process.

6.4.1. Classifiers 10-fold Cross Validation

In case of 10-fold cross validation process, the total accuracy results of each classifier are the same as in the first experimentation. However, in this part a further analysis has been made and the individual's accuracy of each class has computed as shown in the table below.



Graph 4: Accuracy using 10-fold cross validation.

Table 5: Accuracy results with 10-fold cross validation

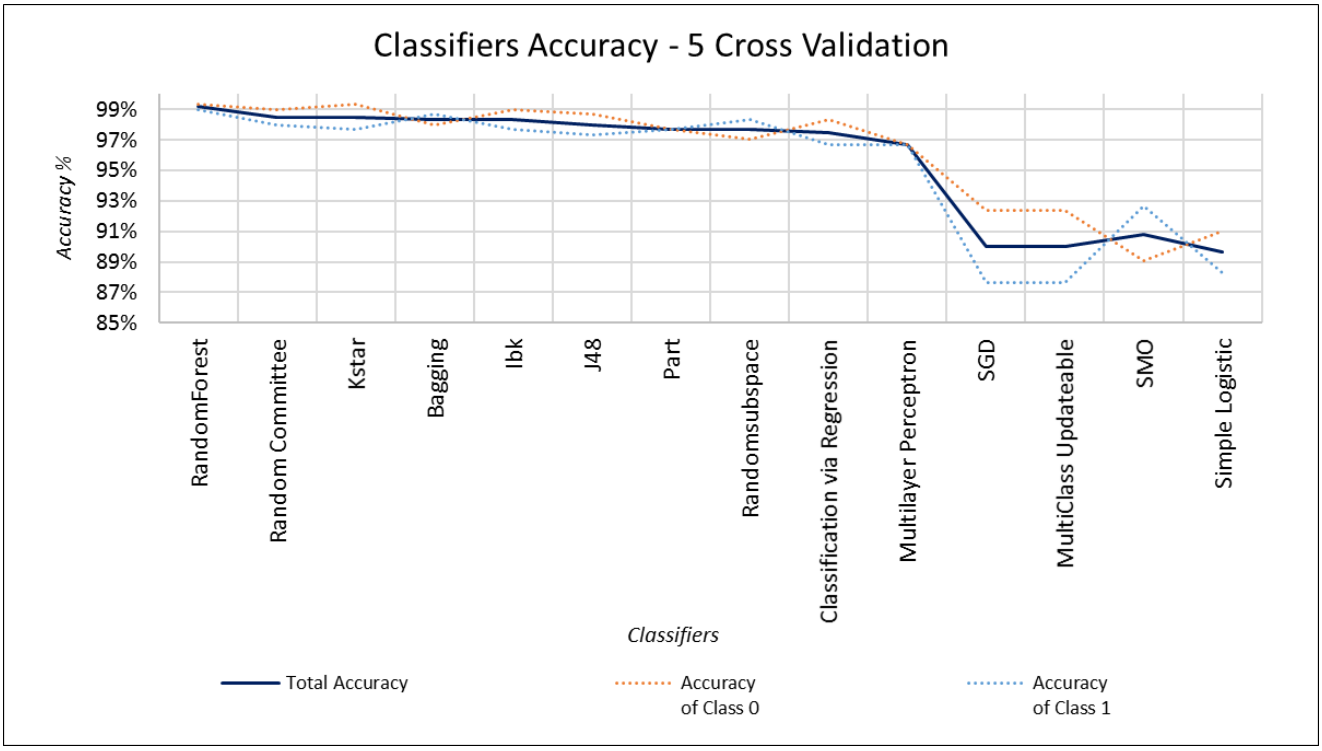
Group	Classifiers	Total Accuracy	Accuracy of Class 0	Accuracy of Class 1
Functions	**Multilayer Perceptron**	99.00%	99.34%	98.66%
Tree	**RandomForest**	99.33%	99.34%	99.33%
Meta	**Random Committee**	98.50%	99.00%	97.99%
Lazy	**Kstar**	98.50%	99.34%	97.66%
Meta	**Randomsubspace**	98.33%	98.34%	98.33%
Lazy	**Ibk**	98.00%	98.67%	97.32%
Meta	**Bagging**	97.83%	98.01%	97.66%
Rules	**Part**	97.83%	98.01%	97.66%
Tree	**J48**	97.50%	98.01%	96.99%
Meta	**Classification via Regression**	97.16%	98.01%	96.32%
Functions	**SMO**	91.16%	89.37%	92.98%
Functions	**SGD**	90.16%	92.36%	87.96%
Meta	**MultiClass Updateable**	90.16%	92.36%	87.96%
Functions	**Simple Logistic**	89.67%	90.70%	88.63%

From the above results, it is clear that the classifier with the highest accuracy is the Random Committee, as mentioned earlier, but in terms of classes' accuracies, things should be redefined, since accuracy of class 0 is 61.13%, which means that only this percentage of the total images in class 0 have been correctly identified in their class. On the other hand, accuracy of class 1 is slightly better with an accuracy of 83.95%. In addition, a similar situation is observed in the next four classifiers, Multilayer Perceptron, Ibk, Kstar and Random Forest. Nevertheless, Multilayer Perceptron classifier, from the fuction group, seems to be the best choice in this part of the experiment, since its total accuracy is 98.50% and also its individual accuracy of class 0 is high as well, 95.68%. However, the best accuracy of class 0 is 99.34% in SMO and Simple Logistic classifier, both from Function group, which means that those classifiers perform better with images from class 0. Nevertheless, their accuracy of class 1 is also remarkable (97.66%), while their total accuracy is high enough, too. In contrast to that, the best accuracy of

class 1, 98.33% is observed in Logistic classifier, which also has a really high but not the best total accuracy.

6.4.2. Classifiers 5-fold Cross Validation

Moving forward, the same classifiers have been tested using 5-fold cross validation process. In this case, seventeen (17) of them presented an increase with an average of 0.39% up and two (2) of them had no increment at all. In addition, nine classifiers had a reduced accuracy with an average of 1.43% down and only one classifier, OneR, presented a dramatically drop with a reduction of 35.17% in his total accuracy.



Graph 5: Accuracy using 5-fold cross validation

Table 6: Accuracy results with 5- fold cross validation

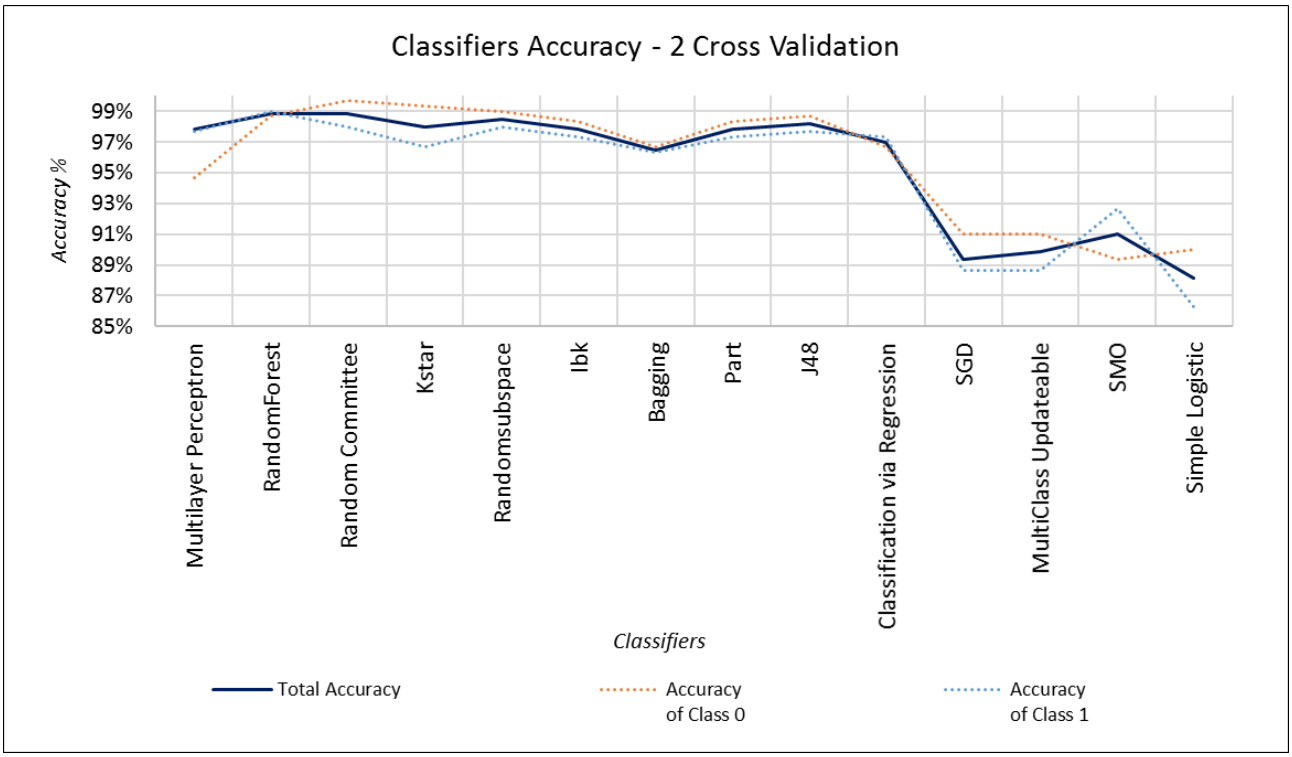
Group	Classifiers	Total Accuracy	Accuracy of Class 0	Accuracy of Class 1
Tree	**RandomForest**	99.17%	99.34%	99.00%
Meta	**Random Committee**	98.50%	99.00%	97.99%
Lazy	**Kstar**	98.50%	99.34%	97.66%
Meta	**Bagging**	98.33%	98.01%	98.66%
Lazy	**Ibk**	98.30%	99.00%	97.66%
Tree	**J48**	98.00%	98.67%	97.32%
Rules	**Part**	97.67%	97.67%	97.66%
Meta	**Randomsubspace**	97.67%	97.01%	98.33%
Meta	**Classification via Regression**	97.50%	98.34%	96.66%
Funtions	**Multilayer Perceptron**	96.67%	96.68%	96.66%
Funtions	**SGD**	90.00%	92.36%	87.63%
Meta	**MultiClass Updateable**	90.00%	92.36%	87.63%
Funtions	**SMO**	90.83%	89.04%	92.64%
Funtion	**Simple Logistic**	89.67%	91.03%	88.29%

In this experiment, it seems that Random Forest had the highest accuracy, 99.17%, but it presented an abnormality in individual accuracies of class 0 and 1. Instead, Bagging and Ibk classifiers had the best scores in terms of class 0, while Ibk had also the best accuracy in class 1. Furthermore, it is observed that Multilayer Perceptron classifier had a drop in its total accuracy, but the individual ones had a slightly better performance, especially the accuracy of class 1. Due to those differences between the 10-fold and 5-fold cross validation process, a further analysis is decided to be implemented. A new experiment was made using 10-fold cross validation by using less instances, in total 400 images and more specific, 200 with Ore and 200 with waste. In this case, the total accuracy of Multilayer Perceptron was 67%, significantly lower than the accuracy of the initial dataset. Then, the same classifier, after the training phase, was tested by using 200 unshown before images, 100 with Ore and 100 with Waste. The extracted accuracy from this process was almost 95%, which indicates the avoidance of overfitting effect during the initial classification data.

In addition, all the other classifiers maintain an almost constant total accuracy and sufficient individual accuracies. In contrast to that, it is observed that only one classifier, OneR, had a significant drop from 85% in 10-fold to 49.83% in 5-fold cross validation process. However, in both cases, this classifier has an extremely low accuracy of class 1, which means it cannot correctly classify images with Ore, while accuracy of class 0 is also low in 5-folds.

6.4.3 2-fold Cross Validation

Furthermore, the same classifiers were tested under 2-fold cross validation circumstances. In this case, it is also observed a fluctuation among classifiers, since only 12 of them presented an increase in their total accuracies, while 16 of them had a fall of about 4.66%. However, there is only one classifier, Filter classifier, which maintain the same total accuracy, while it slightly pushed up the accuracy of class 1.



Graph 6: Accuracy using 2-fold cross validation

Table 7: Accuracy results with 2-fold cross validation

Group	Classifiers	Total Accuracy	Accuracy of Class 0	Accuracy of Class 1
Funtions	Multilayer Perceptron	96.17%	94.68%	97.66%
Tree	RandomForest	98.83%	98.67%	99.00%
Meta	Random Committee	98.83%	99.67%	97.99%
Lazy	Kstar	98.00%	99.34%	96.66%
Meta	Randomsubspace	98.50%	99.00%	97.99%
Lazy	Ibk	97.83%	98.34%	97.32%
Meta	Bagging	96.50%	96.68%	96.32%
Rules	Part	97.83%	98.34%	97.32%
Tree	J48	98.17%	98.67%	97.66%
Meta	Classification via Regression	97.00%	96.68%	97.32%
Funtions	SGD	89.33%	91.03%	88.63%
Meta	MultiClass Updateable	89.83%	91.03%	88.63%
Funtions	SMO	91.00%	89.37%	92.64%
Funtion	Simple Logistic	88.17%	90.03%	86.29%

In this experiment, it is observed that Multilayer Perceptron had a total accuracy of 96.33%, slightly less than in the 5-folds. According to these results, it was decided to test SVM classifier in Matlab classification learner toolbox, since it does not be included in WEKA. Specifically, SVM quadratic method generated a total accuracy of 99% and KNN fine a 97.50%, while their individual accuracies are presented in the following confusion matrices and they are significant high.

Table 9: SVM Quadratic Confusion matrix

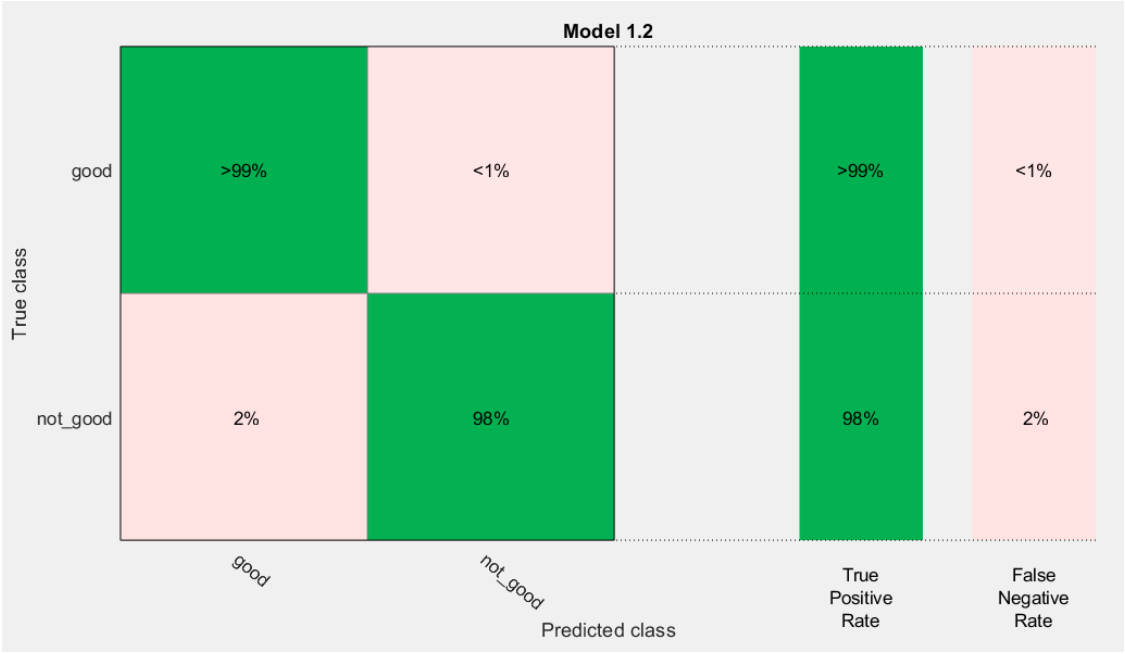
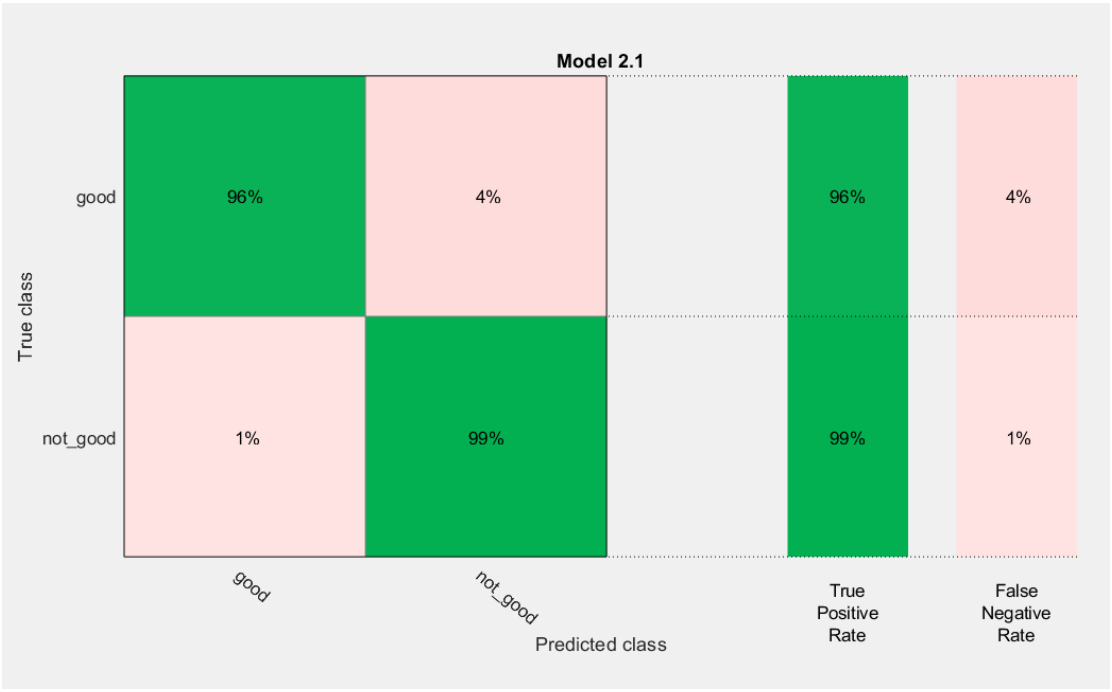


Table 10: KNN fine Confusion matrix



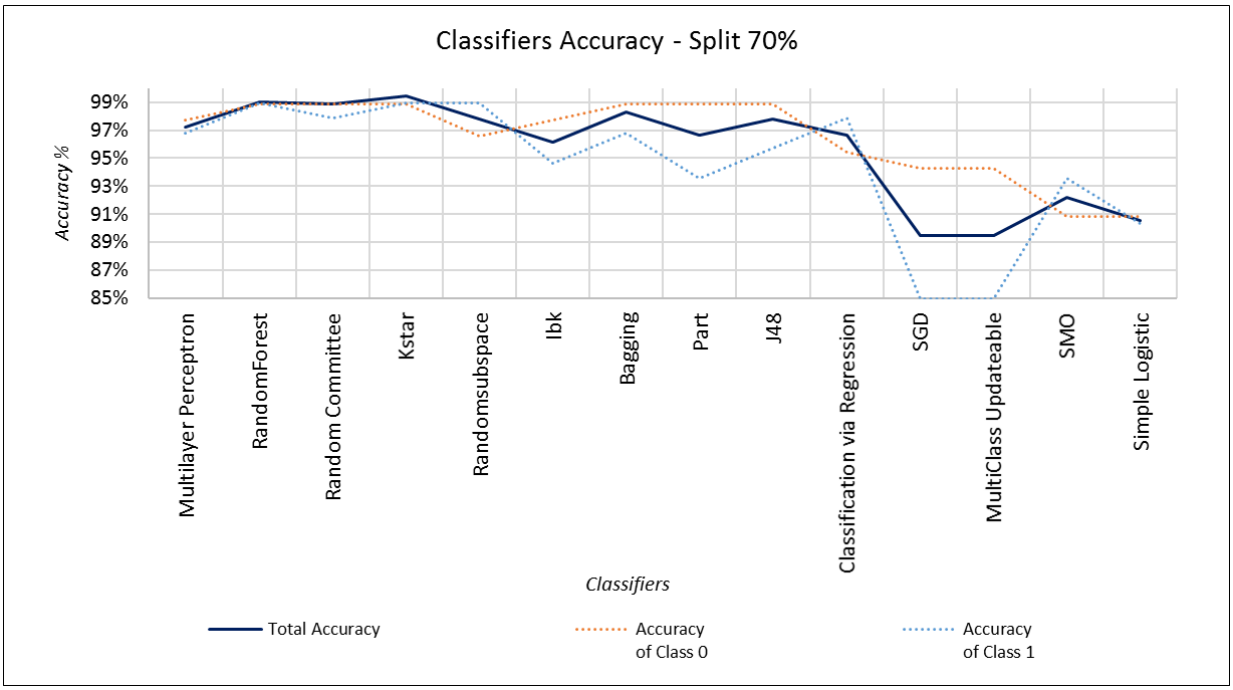
6.4.4. Percentage split

The last experiment was to implement the percentage split method in Weka, in which the dataset is separated into train and test set with a manually chosen separation. In our case, the percentage split has been chosen to be done in 70%, which means that 70% of the dataset was used as a train set and the rest as the test set. Specifically, 420 out of 600 images were used as train set and the rest 180 as test. The extracted results can be found below.

Table 11: Accuracy results with percentage split

Group	Classifiers	Total Accuracy	Accuracy of Class 0	Accuracy of Class 1
Funtions	Multilayer Perceptron	97.24%	97.70%	96.77%
Tree	RandomForest	98.89%	98.85%	98.92%
Meta	Random Committee	98.35%	98.85%	97.85%
Lazy	Kstar	99.89%	98.85%	98.92%
Meta	Randomsubspace	97.74%	96.55%	98.92%
Lazy	Ibk	96.16%	97.70%	94.62%
Meta	Bagging	97.81%	98.85%	96.77%
Rules	Part	96.20%	98.85%	93.55%
Tree	J48	97.27%	98.85%	95.70%
Meta	Classification via Regression	96.63%	95.40%	97.85%
Funtions	SGD	89.60%	94.25%	84.95%
Meta	MultiClass Updateable	89.60%	94.25%	84.95%
Funtions	SMO	92.18%	90.80%	93.55%
Funtion	Simple Logistic	90.56%	90.80%	90.32%

Inside class 0 and 1, it looks like an overfitting effect is presented in the majority of the classifiers, except Multilayer Perceptron, which preserve almost the same accuracy as in the previous experiments. However, this is due to the fact that the given dataset was an accurate simulation of real conditions and the overfitting effect was eliminated when all classifiers have been tested in the smaller dataset of 400 images.



Graph 7: Accuracy using percentage split

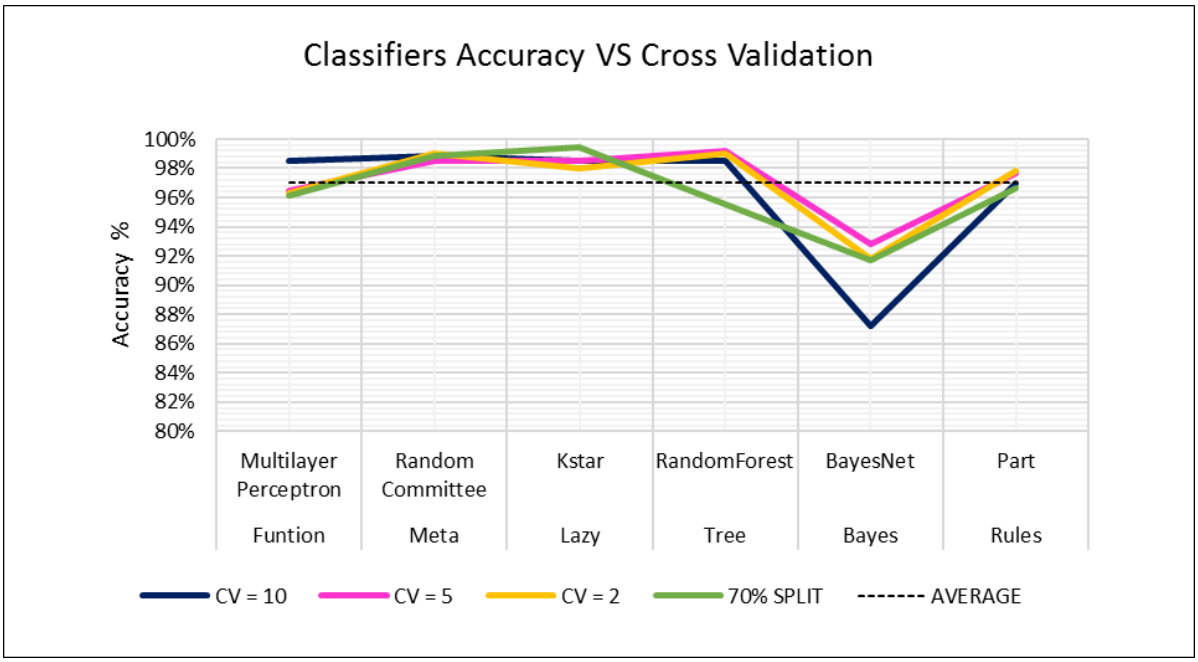
6.4.5. Comparison

In the following table, a summary comparison between some classifiers is presented regarding all the different experiments that have been implemented earlier. As mentioned before, Multilayer Perceptron is a classifier that in case of 10-fold cross validation process had a significantly high accuracy level of 98.50%. Despite that, in all the other experimental processes, its imputation was dropping gradually, with a minimum accuracy of 96.11% in the percentage split method. In contrast to that, Bayes Net classifier, which at first had a total accuracy of 87.17%, seems to improve this percentage in case of less than 10-folds.

Specifically, the turning point was appeared to be number 5, since in 5-fold cross validation method, it reached 92.83% accuracy. All the other listed classifiers present a fluctuation regarding accuracy among the specific experiments. For example, Random Committee had at first a percentage of 98.83%, which had been decreased in 5-folds case scenario and then it becomes better in 2-folds and in the percentage split experiment. These ups and downs are also observed in the majority of the listed classifiers.

Table 12: Accuracies comparison by Group Classifier

Group	Classifiers	CV = 10	CV = 5	CV = 2	70% SPLIT
Funtion	Multilayer Perceptron	98.50%	96.50%	96.33%	96.11%
Meta	Random Committee	98.83%	98.50%	99.00%	98.89%
Lazy	Kstar	98.50%	98.50%	98.00%	99.44%
Tree	RandomForest	98.50%	99.17%	99.00%	95.56%
Bayes	BayesNet	87.17%	92.83%	91.83%	91.72%
Rules	Part	97.00%	97.67%	97.83%	96.67%



Graph 8: Accuracies comparison by Group Classifier

7. CONCLUSION–DISCUSSION

This dissertation was an effort of correctly identifying Ore or Waste in an image in order to use this learning process for further analysis and to be related to an autonomous classification system.

The initial dataset was a primary material of pictures from a mining industry in Greece, Olympias mine – Chalkidiki. All the images have been captured under real case scenarios and under ideal condition, which made all the experimentation process more efficient in contrast to previous related work. In real work situations, specialists after studying and editing those photos, decide whether they include Ore or Waste and how much of them. In the same aspect, all the initial dataset has been annotated as class 0 if the image was Waste and 1 if the image was Ore.

After the annotation part, Matlab has been chosen to be the next used tool, in order to extract the needed knowledge from each image. The final feature extraction has been made after a lot of trials in order to identify the ones that represent best the given dataset. Furthermore, the extracted results were further analyzed and processed in Weka, where a supervised learning process took place, given that the dependent value was binary in the present classification problem. At first, all the classifiers have been tested with their default parameters and they have been evaluated regarding accuracy, precision, recall and F1-score. The generated results from 10-folds cross validation method showed that the classifier with the best accuracy was Random Committee with 98.83%, while Multilayer Perceptron was in the second place with 98.50% of accuracy.

Moreover, all classifiers with an accuracy over 85% were further analyzed using different perspectives. The first experiment was implemented with the 10-fold cross validation process, the second one with the 5-fold, the third one with the 2-fold and the last one was a percentage split of the dataset into 70% train and 30% test set. In all cases, Multilayer Perceptron was in the first place regarding accuracy levels and it never fell below 90%. However, its individual accuracies of class 0 and 1 presented really big diversities and only in the percentage split experiment the results were balanced. On the other hand, Multilayer Perceptron had lower total accuracy than Random Commitee, but in all experiments except the last one, it had small differences between individual accuracies, something which means that these balanced results are more accurate than the

produced results of Random Committee. In addition, all experiments have shown that all classifiers work better with instances from class 1 rather than class 0.

Specifically, in case of 5-folds, Multilayer Perceptron has been chosen as the best classifier, despite the fact that it did not had the best accuracy in this part. However, due to the experiment with the training from a smaller dataset than the initial one, it came out that the specific classifier works very efficiently with out-of-sample images, which is the final scope of the primary idea. In addition, in 2-folds, it is observed an overfitting effect in the majority of the classifiers, which is again overpassed with an accompanied experiment using SVM quadratic and KNN fine methods in Matlab toolbox. In the last part, a percentage split classification process has been implemented, in which the extracted results were really high and this can be explained by the given dataset, which marked the ideal conditions of a mine, like specific lights, distance from the ore and high image quality.

Hence, all models could identify the presence of Ore/Waste Grade cut-off, Rock classification in the majority of the images.

8. FUTUREWORK

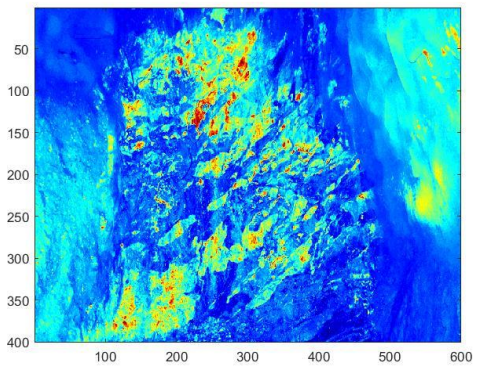
After all the generated results from the above experiments, some thoughts and proposals for future work have been raised.

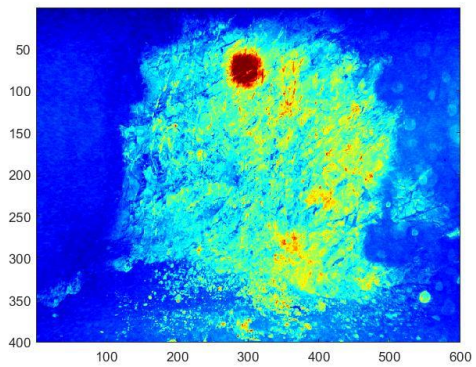
First of all, the main idea behind this thesis was to reach an autonomous level, in which no human intervention will be necessary at all. In order to implement that, a tool should be designed in order for the user to insert the captured images and extract the final results regarding whether an image has Ore or Waste. This tool could also be a sensor or a camera in front of the machines that generates results in real time. This autonomous rock image classification tool can also be applied in more generic geological labelling tasks in order to enhance the planetary exploration.

In addition to this idea, the above tool could also be able to work with an input video as well, since video consists of multiple frames in a row. Thus, the tool should be able to "read" a lot of static and sequential images and produce results according to the given dataset.

Moreover, one more idea for further research is to specify the percent existence of Ore in an image. Specifically, after the classification process of whether an image is Ore or Waste, there should be an extra training phase in case of Ore images. Thus, if an image will be classified as Ore, there should be an extra step that generate results regarding how much is that Ore. This training could be implemented by using the Ore level patterns that specialists use nowadays.

Generally, image processing analysis with colormaps also may lead to extract better rock classification pattern recognition for the mineralization. Comparative future research is proposed as well.





APPENDICES

A. Granting Rights

With this note , I grant all the rights to use, publish and edit the dataset images in Hellas Gold -Eldorado Gold Company which is located in Olympias mine, in NA Chalcidice Peninsula.

Olympias, Chalcidice

2/12/2019

Aikaterini Mantela

B. Matlab code sample

```
dirName=pwd;
   [img,map]=imread(fullfilename);
   img_haze=imreducehaze(img);
   img_size=imresize(img_haze, [400,600]);
   img_gray=rgb2gray(img_size);
   %Co-Occurrence Matrix
   glcm=graycomatrix(img_gray,'NumLevels',256);
   %Contrast
   contr=graycoprops(glcm,'Contrast');
   contrast=getfield(contr,'Contrast');
   %Homogeneity
   homog=graycoprops(glcm,'Homogeneity');
   homogeneity=getfield(homog,'Homogeneity');
   %Correlation
```

```
    correl=graycoprops(glcm,'Correlation');
    correlation=getfield(correl,'Correlation');
    %Energy
    nrg=graycoprops(glcm,'Energy');
    energy=getfield(nrg,'Energy');
    %Entropy
    entropy(img_gray);
    %Statistic properties
    mean2(img_gray);
    std2(img_gray);
    threshold=graythresh(img_size);
    img_bw=im2bw(img_size,threshold); %convert image into a binary one
    %number of edges
    edgimg=edge(img_bw,'Canny');
    [label_num, num_edges]=bwlabel(edgimg);
num_edges;
    %average length of all edges
    len_edges=sum(sum(edgimg));
    avg_length=len_edges/num_edges;
    %Lines
    [H T R] = hough(edgimg);
    P = houghpeaks(H,20);
    lines = houghlines(edgimg,T,R,P,'FillGap',10,'MinLength',20);
   length(lines);
    %Rectangles
```

```
stats1 = regionprops(img_bw,'Area','BoundingBox');
stats2 = regionprops(not(img_bw));
x1=sum(cellfun(@(x)prod(x(3:4)),{stats1.BoundingBox})==[stats1.Area]);
x2=sum(cellfun(@(x)prod(x(3:4)),{stats2.BoundingBox})==[stats2.Area]);
x1+x2;
%count the white pixels
w=sum(img_bw(:)==1);
%Statistics to colors
red_img(:,:,1)=2*img_size(:,:,1);
green_img(:,:,2)=2*img_size(:,:,2);
blue_img(:,:,3)=2*img_size(:,:,3);
mean2(red_img);
std2(red_img);
mean2(green_img);
std2(green_img);
mean2(blue_img);
std2(blue_img);
%Increase contrast and statistics
img_adj=imadjust(img_gray, [0.2, 0.5], []);
mean2(img_adj);
std2(img_adj);
%Histogram Equalization
img_cor=histeq(img_gray);
mean2(img_cor);
std2(img_cor);
```

C. ".arff" file sample

A sample of the used ".arff" file, which was loaded in Weka, is presented below.

```
%% Title: ROCK IMAGE CLASSIFICATION
@relation GradeCutoff_identification
@attribute CONTRAST numeric
@attribute HOMOGENEITY numeric
@attribute CORRELATION numeric
@attribute ENERGY numeric
@attribute ENTROPY numeric
@attribute MEAN numeric
@attribute STD numeric
@attribute EDGES_NUMBER numeric
@attribute EDGES_AVGLEN numeric
@attribute LINES numeric
@attribute RECTANGLES numeric
@attribute WHITE_PIXELS numeric
@attribute MEAN2_RED numeric
@attribute STD2_RED numeric
@attribute MEAN2_GREEN numeric
@attribute STD2_GREEN numeric
@attribute MEAN2_BLUE numeric
@attribute STD2_BLUE numeric
@attribute MEAN2_GRAY numeric
@attribute STD2_GRAY numeric
@attribute MEAN2_HISTO numeric
@attribute STD2_HISTO numeric
@attribute class {0,1}
@data
```

208.3,0.2,0.9,0.000276,7.13,77.22,36.85,747,35.25,88,89626,144.42,64.95,77.84,90.75,
42.1,71.7, 92.8,92.22,127.4,74.76,2032.96,1
264.9,0.2,0.9,0.000192,7.28,90.09,38.67,629,49.24,103,120382,190.03,62.72,86.15,96.87,
37.6,68.2,124.4,91.11,127.49,74.84,1883.5,1
182.8,0.2,0.9,0.000247,7.2,108.12,36.33,835,35.2,101,107768,202.77,51.94,100.67,107.33,
66.5,100.3,168.4,81.93,127.45,74.85,1619.95,1
243.8,0.2,0.9,0.00023,7.17,105.12,35.62,838,37.37,103,104975,197.89,52.27,98.77,105.78,
66,99.6,162.1,82.79,127.44,74.73,1699.52,1
191.2,0.2,0.9,0.000241,7.19,109.22,36.13,791,37.61,98,110334,203.66,51.88,102.08,108.4
,65.7,99.3,171.2,80.74,127.45,74.93,1598,1
283.2,0.2,0.9,0.000184,7.22,105.09,37.24,742,44.67,110,92472,194.58,53.93,98.15,105.16,
64.2,98,158.7,83.51,127.5,74.79,1718.18,1
263.8,0.2,0.9,0.000184,7.25,104.13,37.83,757,43.57,100,96514,192.83,55.35,97.64,105.03,
62.8,96.5,156.4,85.4,127.42,74.88,1728.29,1
222.1,0.2,0.9,0.000195,7.27,105.8,38.34,731,43.61,99,95444,194.75,54.86,98.58,105.81,
63.7,97.6,159.6,84.82,127.47,74.79,1691.8,1
191.5,0.2,0.9,0.000207,7.27,106.07,38.46,729,42.53,106,96153,195.39,54.85,98.67,105.86,
64,98,160.2,84.85,127.5,74.8,1683.83,1
286.1,0.2,0.9,0.000182,7.24,105.71,37.51,790,42.55,111,94499,195.33,54,98.69,105.72,
64.2,97.9,160.2,83.67,127.51,74.82,1700.91,1
277.9,0.2,0.9,0.000184,7.24,103.06,37.62,766,43.96,109,96335,192.61,54.93,96.44,104.08,
63.4,97.2,154.2,85.76,127.48,74.87,1753.14,1
243,0.2,0.9,0.000191,7.25,103.92,37.75,775,42.19,106,96149,191.94,55.78,97.33,104.82,
64,97.7,156.1,85.58,127.47,74.88,1733.02,1
293.3,0.2,0.9,0.000186,7.25,87.71,37.97,675,48.67,109,117060,186.03,63.06,84.04,95.07,
38.4,69.1,118.9,90.74,127.51,74.78,1919.77,1
294.6,0.2,0.9,0.000165,7.29,107.16,39.13,709,46.29,115,94652,196.51,55.31,98.8,106.04,
65.5,99.8,161.6,85.16,127.4,74.88,1645.68,1

REFERENCES

[1] Bouckaert, & Scuse, D. (2016). *WEKA Manual for Version 3-8-1*. Hamilton, New Zealand.

[2] Bootsma, M., (2018) *Thesis Cut-Off Grade Based Sublevel Stope Mine Optimization*, Delt University of Technology.

[3] Christodoulou, P. (2018). *Crosswalk identification for Decision Making*.

[4] Sokolova, M., & Lapalme, G. (2009). *A systematic analysis of performance measures for classification tasks. Information Processing & Management*, 45(4), 427–437. https://doi.org/10.1016/J.IPM.2009.03.002.

[5] Hall, M., & Witten, I. H. (2009). *The WEKA Data Mining Software: An Update. SIGKDD Explorations*, Vol. 11, no. 1, pp. 10-18.

[6] Hellas. Gold, (2015), *Geology Department of Olympias Mine Hellas Gold - Eldorado Gold*,. Available: https://www.eldoradogold.com/home/default.aspx , https://www.hellasgold.gr.

[7] HIPR2, (2003), *Sobel Edge Detector*, Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm.

[8] Kotsakis, R. (2015). *Application of machine learning algorithms for extracting and classifying content information*, Degree Grantor: Aristotle University of Thessaloniki (AUTH)

[9] Karnewar, A., (2018) *Back-to-basics Part 1: Histogram Equalization in Image Processing*, Medium, Available: https://medium.com/@animeshsk3/back-to-basics-part-1-histogram-equalization-in-image-processing-f607f33c5d55

[10] Kotsakis, R.,(2018), *Image Processing," in Multimedia Data Analysis*, International Hellenic University.

[11] Lepisto, (2006), *Thesis of Colour and Texture Based Information of Rock Images Using Classifier Combinations*, Tampere University of Technology, April 2006.

[12] MathWorks, (2019), *MATLAB*, Available: https://ch.mathworks.com/products/matlab.html

[13] MathWorks, (2019), *Contrast Adjustment*, Available: https://ch.mathworks.com/help/images/contrast-adjustment.html

[14] MathWorks, (2019), *Hough Transform*, Available: https://ch.mathworks.com/help/images/hough-transform.html?s_tid=srchtitle

[15] M. H. Al Amiri, (2013), *The Binary Image in Digital Image Processing*, p. 11.

[16] Partio, M. & A. Visa., (2006), *Rock Texture Retrieval Using Gray Level*, Tampere Univercity of technology, April, 2006

[17] Shiron, C. R., D. Rhys, (2018) & T. Baker, T. Veligrakis & L. Dalampiras, *Structural Controls on Porphyry Au-Cu and Au-Rich Polymetallic Carbonate-Hosted Replacement Deposits of the Kassandra Mining District, Northern Greece*, March 2018., Available: https://pubs.geoscienceworld.org/segweb/economicgeology/article-abstract/113/2/309/529212/structural-controls-on-porphyry-au-cu-and-au-rich?redirectedFrom=fulltext

[18] Waikato, Weka 3, (2019), *Machine Learning Software in Java*, Available: https://www.cs.waikato.ac.nz/ml/weka/
[19] Waikato, Weka 3, (2019), *Weka.Classifiers.Rules*, Available: http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/package-summary.html
[20] Waikato, Weka 3, (2019), *Weka.Classifiers.Trees*, Available: http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/package-summary.html
[21] Wei, X. S. & Y. Liu, (2014), *Rock Classification Based on Image Processing and Neural Networks*, June 2014. Available: https://www.scientific.net/AMM.568-570.685
[22] Lei Shua,b, (2017), *Unsupervised feature learning for autonomous rock image classification*, Department of Electrical and Computer Engineering, University of Western Ontario
[23] Claudio A. & E. Medina, (2011), *Ore grade estimation by feature selection and voting using boundary detection in digital image analysis*, International Journal of Mineral Processing