

Computer vision application for improved product traceability in the granite manufacturing industry

 X. Rigueira^a,  J. Martínez^b ✉,  M. Araújo^a,  E. Giráldez^a,  A. Recamán^c

a. Department of Natural Resources and Environmental Engineering, University of Vigo, (Vigo, Spain)

b. Department of Applied Mathematics I, University of Vigo, (Vigo, Spain)

c. Pavestone S.L., (Madrid, Spain)

✉: javmartinez@uvigo.es

Received 14 October 2022

Accepted 8 February 2023

Available on line 10 August 2023

ABSTRACT: The traceability of granite blocks consists in identifying each block with a finite number of colour bands that represent a numerical code. This code has to be read several times throughout the manufacturing process, but its accuracy is subject to human errors, leading to cause faults in the traceability system. A computer vision system is presented to address this problem through colour detection and the decryption of the associated code. The system developed makes use of colour space transformations and various thresholds for the isolation of the colours. Computer vision methods are implemented, along with contour detection procedures for colour identification. Lastly, the analysis of geometrical features is used to decrypt the colour code captured. The proposed algorithm is trained on a set of 109 pictures taken in different environmental conditions and validated on a set of 21 images. The outcome shows promising results with an accuracy rate of 75.00% in the validation process. Therefore, the application presented can help employees reduce the number of mistakes in product tracking.

KEY WORDS: Computer vision; Granite; Traceability; Pattern detection; Colour detection.

Citation/Citar como: Rigueira, X.; Martínez, J.; Araújo, M.; Giráldez, E.; Recamán, A. (2023) Computer vision application for improved product traceability in the granite manufacturing industry. *Mater. Construcc.* 73 [351], e323. <https://doi.org/10.3989/mc.2023.308922>.

RESUMEN: *Visión artificial aplicada a la industria del granito para la mejora de la trazabilidad.* La trazabilidad de los bloques de granito consiste en identificar cada bloque con un número finito de bandas de color, las cuales representan un código numérico. Dicho código tiene que ser leído varias veces durante el proceso de producción, pero la precisión de esta lectura se encuentra afectada por el factor humano, lo cual lleva a fallos en el sistema. Se presenta un sistema de visión artificial basado en la detección de colores y la decodificación de dichas bandas. El sistema hace uso de transformaciones entre espacios de color y varios intervalos para la selección de los mismos. Se implementan métodos de visión artificial, incluyendo la detección de contornos para la identificación de la posición de los colores. En último lugar, se analiza la geometría del patrón de colores para su decodificación. El algoritmo propuesto es entrenado en un set de 109 imágenes tomadas en diferentes condiciones medioambientales y validado en un set de 21 imágenes. Los resultados son prometedores, demostrando una eficacia del 75% en el proceso de validación. Por lo tanto, el sistema propuesto se considera de utilidad a la hora de incrementar la eficacia de la trazabilidad en la industria del granito.

PALABRAS CLAVE: Visión artificial; Granito; Trazabilidad; Detección de patrones; Detección de colores.

Copyright: ©2023 CSIC. This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International (CC BY 4.0) License.

1. INTRODUCTION

The granite manufacturing industry holds a key role in the industrial network of the North-Western area of Spain. This region is the largest ornamental granite producer, accounting for 62% of the total national production (1). Traditionally the mining sector, including the granite industry, has been reluctant to adopt new technologies, but the competitive market leaves few options for the mining industry (2). In this context, an improvement in traceability is required. By definition, traceability involves any processes, procedures, or systems that support the generation of verifiable evidence about a product as it moves along its supply chain (3). Nowadays, blocks are initially identified in the quarry with marks indicating type and origin, once they reach the factory for further processing, they are colour-coded for identification purposes (4). Given this setting, a computer vision application could prove helpful in the automation of the traceability system in use.

The scope of computer vision is the development of theories and algorithms for automating the process of visual perception. The mathematical basis for image processing and feature analysis was defined by several authors, including Ervin E. Underwood, John C. Russ, and Jean Serra (5-8), in the last decades of the twentieth century and beginning of the twenty-first, while its applications reach a broad range of fields, and the potential benefit of its development can have a major impact in the upcoming years. Computer vision has found its applications in the mining sector, more specifically, in the slate industry for the detection of defects in the ceramic industry (9-11), for the same purpose in the marble industry (12-16), for pattern detection and classification (17, 18) as well as in the granite manufacturing industry for the characterization of granite varieties (4, 19), but there has not been found any previous research on the applications of computer vision for colour detection and analysis in order to improve product traceability in this sector.

The current traceability method based on colour-coding each granite block is affordable but entails a substantial number of drawbacks, which may lead to failures. The colours have to be identified and interpreted as a sequence of integers several times in the production process by an employee. This simple task can be subject to human error due to fatigue and the harsh work environment in the granite industry, plus its accuracy is affected by the weathering of the colours. The main goal of the program developed is to analyse pictures of granite slabs with the colour code drawn on their side and convert the colour bands into their corresponding numerical code. Consequently, the implementation of a program of this kind would dramatically reduce the time needed to decrypt the colour code, currently performed manually, and reduce human error. In addition, it decreases economic losses by avoiding the discard of those slates not cor-

rectly identified. The work presented in this research paper is a crucial step in the implementation of this system on mobile phone devices for in-factory use with the same purpose and even more features.

Section 2 of this paper introduces the materials needed for this research, including the granite slabs, the images with the colour bands, a middle-class computer, and Python 3.9. and explains the methodology of this research, focused on the algorithm developed. Section 3 presents the results obtained on the detection of colour bands on granite and the decryption process. Lastly, Section 4 concludes with the main findings and future applications.

2. MATERIALS AND METHODS

Granite is by definition a very hard natural igneous rock formation of visibly crystalline texture formed essentially of quartz and orthoclase or microcline and used especially for buildings and monuments. It typically contains 20-60% quartz, 10-65% feldspar, and 5-15% mica (biotite or muscovite), although a wide range of lithological materials is considered granite from a commercial perspective (4). Granite blocks mined in the quarry are wired cut when processed in the factory to produce the granite slabs. Their dimensions are not consistent and tend to range between 1.70 to 2.00 meters in width and 20 to 30 mm in thickness.

The database used in this research project consists of 130 pictures of granite slabs focused on one of the sides of the slab, where the colour code displayed is in the shape of bands manually drawn. The sole purpose of this particular type of code is to keep track of the granite slab from its entry into the factory until its final sale to the customer.

The colour code is made up of a variable number of colour bands drawn using spray paint on the granite block once it enters the factory. They are placed either close to the higher or lower edges of the block, with a distance of approximately 15 cm in between them. Each band can have any of the 8 different colours shown in Table 1, and they can be placed in any order. Each colour is associated with a number, which makes the block have a specific numerical code depending on the configuration of the colour bands it has drawn on its side. This method to numerate, count and keep track of every block has been chosen over simply drawing the corresponding number on the block because, to produce the final version of the product, the block has to be wired cut. This means that the colour bands drawn on the surface of the stone would be cut as well, and they would become illegible, losing their purpose.

The iPhone 12 Pro camera used to capture the side of the granite slabs in the RGB colour space has a resolution of 12MP, a $f/1.6$ aperture, and 26 mm of focal length. Development and testing of the program

TABLE 1. Code key for decoding the encrypted unique number that corresponds to every individual granite block processed in the factory. Each colour is associated with an integer between 0 and 7. The number of colour bands displayed grows depending on the number of blocks that the factory has processed. The images in the database display five bands, which corresponds with positive numbers of five digits.

Black	Brown	Red	Orange	Yellow	Green	Blue	Purple
0	1	2	3	4	5	6	7

have been completely performed with a 1.6 GHz Intel i5 10210U processor with 16 GB (2×8 GB) of RAM. The program to identify colour codes is written in Python 3.9 using OpenCV, which is an open-source computer vision library written in C++ and C.

2.1. Theory of colour detection

A correct choice of colour space is considered crucial in computer vision applications. There are several types of colours spaces available, but for the most part, they can be classified into two categories: device-dependent or device-independent. The first group includes those colour spaces, which their representation of colours lays further away when compared to how the human nervous system senses colours. According to (20), the colour spaces in this category, such as RGB and HSV, simply encode device-specific data at the device level. On the other hand, the colour spaces in the second group are directly related to the human visual system. They aim to define colour coordinates, which can be comprehended by the average observer. The basic colour space in this category is CIE XYZ and any other colour spaced that can be transformed directly into CIE XYZ is considered device-independent, such as CIE Lab or CIE Luv. Moreover, the concept of uniformity can be applied in this category: a colour space is said to be uniform when the Euclidean distance between colours in that space is proportional to colour differences as perceived by humans (21).

Digital images are usually captured in the RGB colour space, whereas hue-based colour models, such as HSV, are the most commonly implemented in colour detection in OpenCV due to their robustness against light changes, while CIE Lab can be more efficient in measuring colour differences in brightness. According to (22), HSV outperforms RGB because it is approximately uniform and divides the colour data into intensity (Value) and a chromatic part (Hue and Saturation).

2.1.1. The RGB colour space

Two of the main advantages of the RGB colour space are its simplicity and its additive property, which makes it very easy to store and display images (23). Nevertheless, this colour model is not the best for colour detection, since important colour properties such as brightness and purity are embedded

within the RGB channels, which makes it difficult to determine specific colours and their reliable working ranges (24). Furthermore, the original pictures were processed in the 24-bit RGB format, in which all components have a depth of 8 bits. This makes up a maximum number of colours of $28 * 28 * 28 = 16777216$. The components are internally presented as unsigned integers in range [0, 255], which is the exact value range of a single byte (25).

2.1.2. The HSV colour space

The HSV colour space is a transformation of the RGB colour space, and it can be represented in a different coordinate system. The idea of a representation in a hexagonal cone (hexacone) was first proposed by (26) and it can be observed in Figure 1, where points are defined by hue (H), saturation (S), and value (V). In this space, hue (H) contains the colour angle information, saturation (S) represents purity, which quantifies how a colour is diluted by white, and value (V), which stores the brightness of the colour measuring how far it is from black. The separation of crucial properties such as brightness and purity is what makes the HSV colour space a better fit for colour detection purposes.

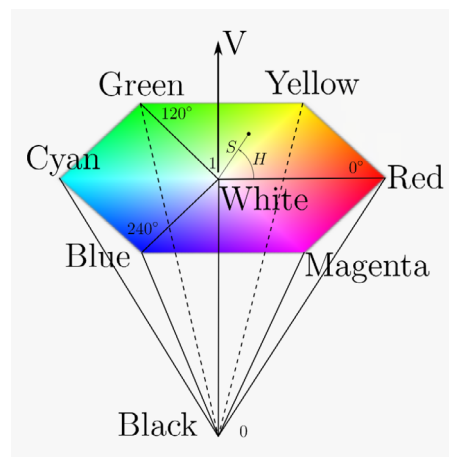


FIGURE 1. Hexagonal representation of the HSV colour space, where the central axis contains all the different shades of grey from black to white. All colours can be defined by their hue (H), saturation (S), and value (V). The horizontal cross-sections of the hexacone are hexagons of different sizes degrading to black, which is a single point. Since hue is an angular measure, the HSV colours space becomes highly effective in defining pure colours.

The transformation from RGB to HSV as described in (27) is given by (28) in the following equations:

$$V = \max(R, G, B) \quad [1]$$

$$S = \begin{cases} \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)}, & \text{if } \max(R, G, B) \neq 0 \\ 0, & \text{otherwise} \end{cases} \quad [2]$$

$$f(x) = \begin{cases} \text{undefined,} & \text{if } S = 0 \\ \frac{G - B}{\max(R, G, B) - \min(R, G, B)}, & \text{if } R = \max(R, G, B) \\ 2 + \frac{B - R}{\max(R, G, B) - \min(R, G, B)}, & \text{if } G = \max(R, G, B) \\ \frac{R - G}{\max(R, G, B) - \min(R, G, B)}, & \text{if } B = \max(R, G, B) \end{cases} \quad [3]$$

For our purposes, and due to its advantages, the HSV model was chosen to detect the different colours present on the granite slabs.

2.2. Workflow implemented

The presented program uses a computer-vision-based approach to identify different colour bands on granite slabs and output a numerical code defined by the colour of each band. The workflow diagram of the proposed system is presented in Figure 3, where the following stages are introduced: (1) data acquisition, (2) image pre-processing, and (3) image analysis. This workflow is represented graphically in Figure 2.

2.2.1. Data acquisition

The images were obtained *in situ*, under the effect of the changing conditions of the environment, and analysed afterward. As a simple requirement, the colour

bands must be completely visible. This means that the picture cannot be taken up too close to the point some colours are left out of the image or from a very distant location where the colour bands are unidentifiable. The RGB pictures that make up the database of the case studied were taken in three separate batches under non-identical conditions. Their features are shown in Table 2.

2.2.2. Image pre-processing

The pre-processing of the images includes two main operations. In the first place, pictures are scaled down to 15% of their initial dimension, decreasing the computational load without compromising the final outcome. Furthermore, the program allows the user to crop out the area where the colour bands are captured on the scaled picture, which entails an increase in the overall accuracy of the results obtained.

2.2.3. Image analysis

The different colour bands are detected by a total number of eight functions, one for every colour. They share a common algorithm (algorithm 1), but the input parameters change depending on each target colour. These inputs are not arbitrary; initially, they were set manually, but in order to increase the precision of the model, the parameters have to be optimized for every case. This is achieved in the training process, which aims to minimize the error in colour detection and the decryption of the code. In order to accomplish this, each colour detection algorithm iterates on the available images for every combination possible within the ranges set for each parameter. A success condition is defined based on the ground truth of colours contained in each image. This helps identify those combinations which succeed in their task, therefore, those parameters with a higher number of true positives become the ideal values in each function.

TABLE 2. Description of the different batches of pictures that make up the database. The first two columns show the name of the batch and the number of pictures. The next three columns include the distance between the camera and the granite slab, the angle of the camera and the slab, and the area where the picture is focused. The last three columns include the proportion of pictures taken in inside lighting conditions, outside and the resolution of the images, respectively.

Name	# Pictures	Distance (m)	Angle (°)	Focus	Lighting		Resolution (pixels)
					Inside	Outside	
Batch1	14	1	Not consistent	Colour bands	100%	0.00%	3472x4640
Bacth2	40	1.75	Parallel	Colour bands	27.50%	72.50%	3024x4032
Batch3	84	2	Parallel	Granite slabs	25.90%	74.10%	3024x4032

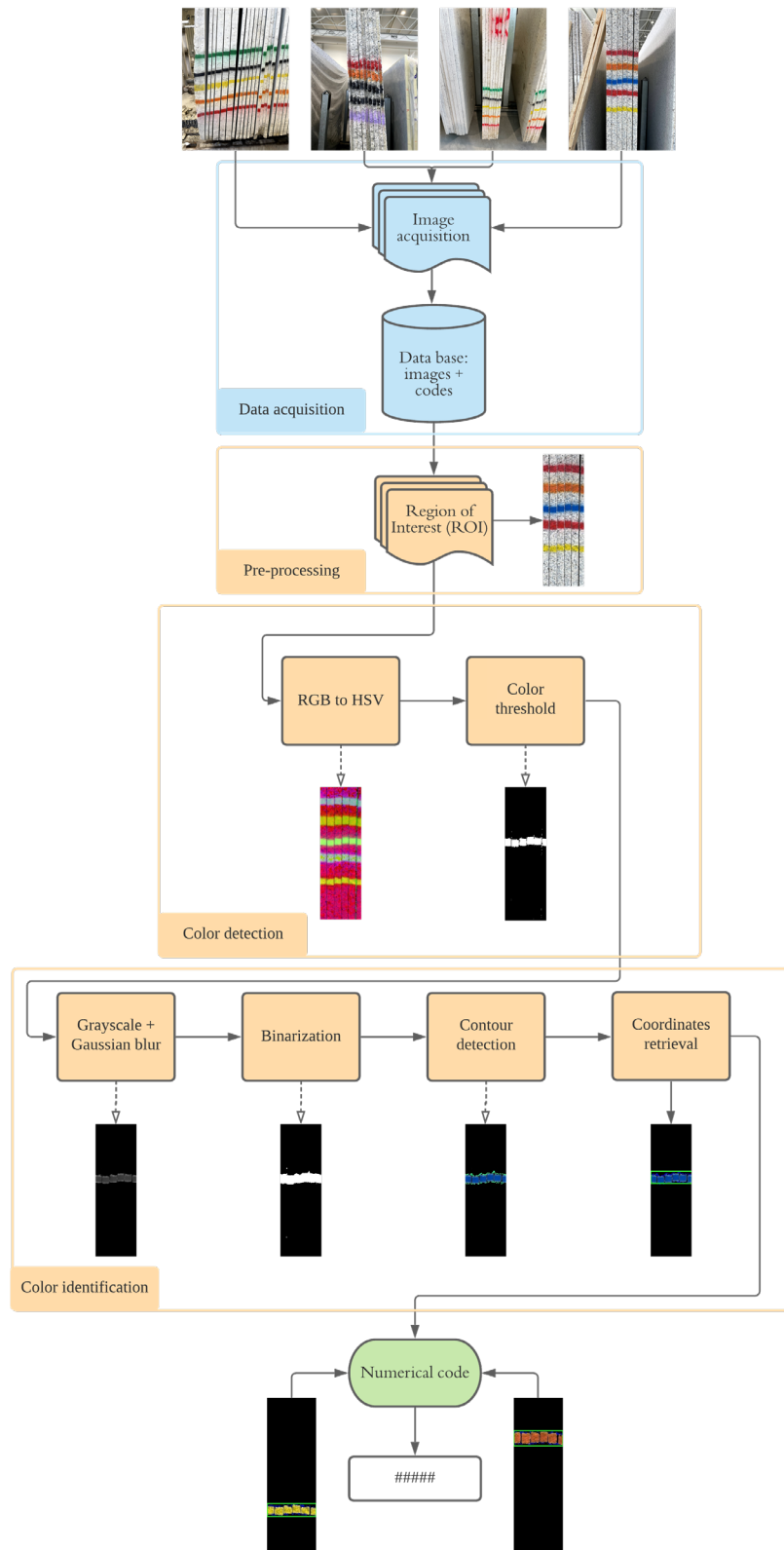


FIGURE 2. Workflow of the proposed computer vision system with the following steps: (1) data acquisition: the pictures used were taken under different lighting conditions -cloudy and sunny days- as well as varying distances and angles to the side of the granite slab, (2) image pre-processing: images are scaled to a 15% of their original size and the target area is cropped by the user. (3) Colour detection: conversion between the RGB and HSV colour spaces and thresholds are implemented to isolate the colour bands, (4) colour identification: blurring, binarization, and contour detection help retrieve the coordinates of each colour band which define the order of the resulting numerical code of the granite slab.

The input parameters for all functions are -1- coloured area (CA), -2- coloured ratio (CR), -3- width-to-height ratio (WHR), -4- maximum vertical distance (MVD), -5- minimum Hue (Hmin), -6- minimum Saturation (Smin), -7- minimum Value (Vmin), -8- maximum Hue (Hmax), -9- maximum Saturation (Smax), and -10- maximum Value (Vmax).

The coloured area -1- defines the minimum number of pixels that have to be turned on to be considered a true positive case of colour detection, while the colour ratio -2- is given by the relation between the number of pixels coloured with the target colour and the total amount of pixels; it has the same goal as the coloured area but implements a second layer of security to avoid false positives. The width/height ratio -3- analyses the relation between the width and the height of the colour areas detected. Given the fact that the colour bands have rectangular shapes, this parameter aims to discard all detected areas which do not have the minimum width-to-height ratios, such as shades or colour spots that do not belong to the colour code system. The minimum vertical distance -4- quantifies the required separation between different colour bands to be considered independent. These bands are drawn directly on the granite block, which is cut afterward to generate the granite slabs, in this process the colour bands are also divided into several sections. As a consequence of this division, gaps are generated throughout every colour band, with a direct effect on the colour detection process. When the picture is taken, these gaps are usually shown as dark spaces that break up the colour band, as a result, several coloured areas will be identified, but thanks to the implementation of this parameter, as long as the detected areas are within the height range set by the MVD, they will be considered as a single band. Lastly, parameters from 5 to 10 belong to the HSV colour space and define the thresholds for each colour.

Each colour function takes the RGB image, proceeds with the HSV transformation, and applies the colour thresholds on the resulting image. In this case, an adaptive multichannel threshold was implemented to identify colour in a more reliable way under the wide variety of lighting conditions presented in this problem. The first of the colour masks takes the lowest hue, saturation, and value possible for the target colour, delimiting the range for these three channels. Therefore, once the mask is applied to the HSV picture, the resulting image is binarized as the pixels that are defined within this range are turned into white [1] and the ones which fall out of this condition are turned off to display black [0]. The mathematical principle of this method is introduced in Equation [4]:

$$I = \begin{cases} 1, & \text{if } H \in [H_{min}, H_{max}] \wedge S \in [S_{min}, S_{max}] \wedge V \in [V_{min}, V_{max}] \\ 0, & \text{otherwise} \end{cases} \quad [4]$$

Where H is hue, S is saturation, and V is value. Similarly, the second mask applies the same principle

and methodology, but in this case, it takes the highest values of hue, saturation, and value for the target colour. The sum of both masks is applied to the original image, concluding the detection of the target colour in every case.

In the case that the resulting image delivered by the colour detection algorithm fulfils the requirement established by the CR parameter, the image is proposed for further processing, where the coordinates of the colour bands identified are extracted. This process begins with the detection of the contours of all isolated coloured areas. To achieve this, the picture is converted to grayscale and blurred out with the implementation of Gaussian blur. This helps reduce the sharpness of the structures contained in the picture and increases the efficiency of the contour detection method.

Gaussian blur is classified as a low-pass filter because it reduces the high-frequency components of the image. It uses Equation [5] explained in (29, 30):

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad [5]$$

Where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. Applying Equation 5 produces a surface on which contours are concentric circles with a Gaussian distribution from the centre point. The convolution matrix, built from the values of this distribution, is applied to the original image, where the new value of each pixel is set to a weighted average of the pixels in its surrounding area. The pixels further away from the centre (original pixel) receive smaller weights as the distance increases, while the original holds the highest value. This helps preserve boundaries and edges in a better way compared to simpler filters. Additionally, the blurred image is binarized before applying contour detection, in this step all pixels that are black remain in this state, while the rest are converted into white according to Equation [6]:

$$dst(x, y) = \begin{cases} 1, & \text{if } src(x, y) \in (0, 1) \\ 0, & \text{otherwise} \end{cases} \quad [6]$$

Where src is the source image, and dst is the destination image of the same size and type as the source. The contour detection method, as defined Suzuki and Abe (31, 32), is a border following algorithm, which works by starting at a given point on the image and following the contour of the object until it returns to the starting point. The algorithm keeps track of the points visited along the contour, as well as the junctions (i.e., points where the contour branches or reconnects) encountered along the way. In this process, those areas of the picture with an intensity gradient

strong enough to be noticed by the algorithm are detected and marked up with individual points. The result is a point cloud surrounding the different structures featured in the image. By leveraging this result, the area that encloses the colour band is defined with the aid of Green's theorem (33). Such theorem relates the circulation of a vector field around a closed curve to the flux through the surface bounded by that curve. It states that the circulation of a vector field F around a closed curve C is equal to the flux of the curl of f through the surface S bounded by C , as expressed mathematically in Equation [7]:

$$\oint_C F \cdot dr = \iint_S \text{curl}(f) \cdot dS \quad [7]$$

Here, dr is an infinitesimal line element along the curve C , and dS is an infinitesimal surface element on S . The circulation is the line integral of F along C , and the flux is the double integral of the curl of F over the surface C . The curl of a vector field is defined as a measure of the local rotation or twisting of the field. It is defined as the vector cross-product of the *del* operator and F :

$$\text{curl}(F) = \nabla \times F \quad [8]$$

In the case that the obtained coloured area is bigger than CA, the contours are drawn, and their length is calculated. Next, the irregular and complex shape of the contours is simplified to a rectangle, which contains each valid colour area detected. This is achieved with the Ramer–Douglas–Peucker algorithm (34, 35), which reduces the number of points in a curve through approximation, resulting in a smaller series of points. The algorithm works by recursively eliminating points that are within a given tolerance of the curve. This process is repeated until the desired tolerance is achieved. The resulting curve is typically a simplified version of the original curve, with fewer points but still closely approximating the original curve. Given that the colour bands tend to have fairly rectangular shapes, a rectangle was selected for this purpose, which wraps all around every area defined in this step. Figure 3 shows a simplified configuration of these rectangles.

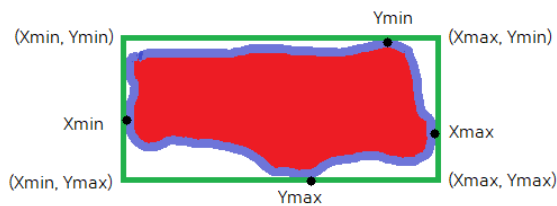


FIGURE 3. Illustrative representation of how the coordinates of the wrapping rectangle are selected. These are defined by the maximum and minimum values of x and y contained within the perimeter of the coloured area.

Up until this last step, there can be coloured areas of the picture that fulfil all the requirements to be considered as colour bands, but in reality, they are not. Therefore, the parameter WHR is introduced, which sets a limit to the numeric relation between width and height for every rectangle. It allows filtering out the vast majority of areas with rectangular shapes, eliminating those in which the vertical sides are the longest.

Following up comes the implementation of the last parameter, MVD. As explained before in this section, its goal is to group under the same band those areas with equal colour, but which are broken down into smaller rectangles as seen in Figure 4. This step avoids the anomalous detection of a not realistic number of bands. The relative coordinates of each band are retrieved from the information contained in the rectangles. For this particular application, the y coordinates are sorted from least to greatest and analysed in this order. In the case the result of the subtraction between $i + 1$ and i is smaller than the MVD, those two rectangles would be considered as the same colour band, otherwise, they would be set to belong to separate colour bands. Lastly, the average of all the coordinates belonging to each colour band is calculated to get a single value per band, which allows the algorithm to read the colours and write the code in the correct order. This process is represented in Figure 5.

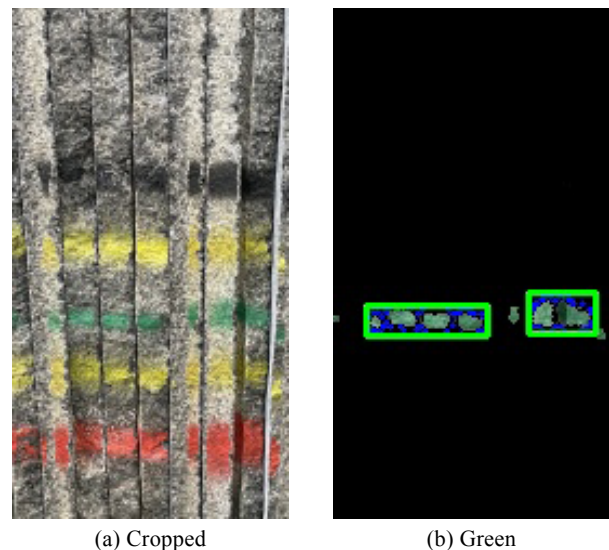


FIGURE 4. Example of a faded green colour band in which the algorithm detects two different sections. This scenario could account for a wrong number of bands, but the implementation of the maximum vertical distance (MVD) parameter limits the amount of space between bands to be considered as such and joins under a single band those which have similar y coordinates.

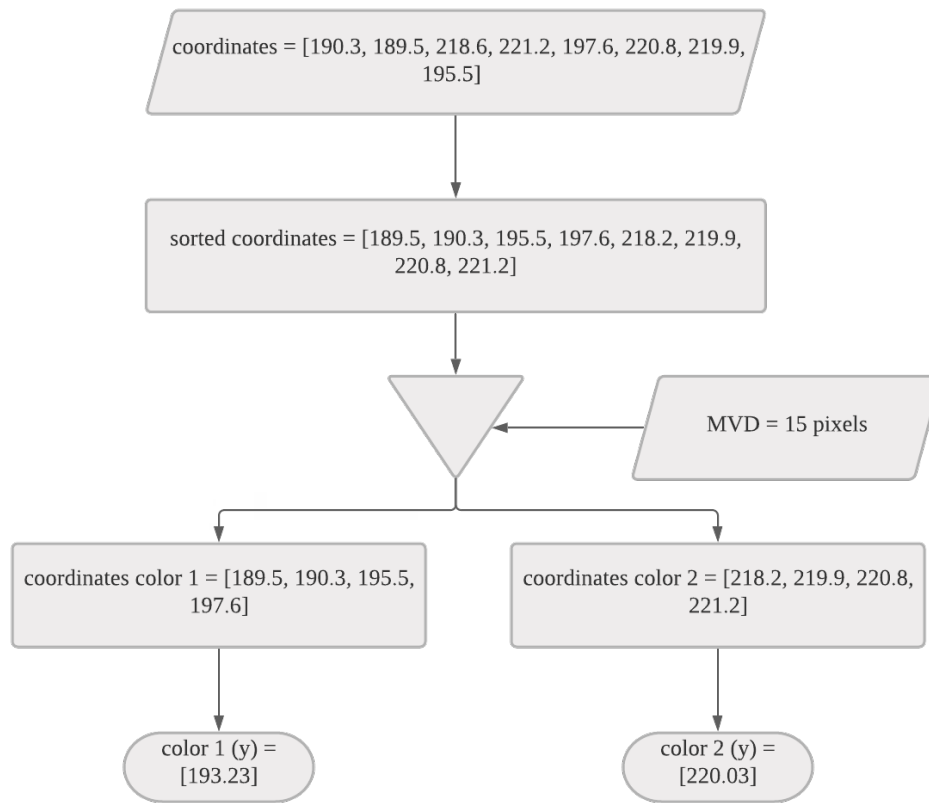


FIGURE 5. Simplified representation of the process followed by the algorithm to group together detected areas or assign them to separate colour bands. Firstly, the coordinates are sorted from least to greatest and put together if the result of their subtraction is smaller than the number defined by the maximum vertical distance parameter, otherwise, they are divided into different colour bands.

Algorithm 1. [colour_detector(image, CA, CR, WHR, MVD, Hmin, Smin, Vmin, Hmax, Smax, Vmax)]

Input: An image and all the parameters defined.

Output: A list containing the (x, y) coordinates of each colour band and the colour name.

1. image \leftarrow RGB2HSV(image)
2. masked_image \leftarrow thresholds(image)
3. binarized_image \leftarrow bitwise_and(image, masked_image)
4. coloured_ratio \leftarrow (# active pixels in binarized_image)/(# total pixels in binarized_image)
5. **if** coloured_ratio \geq CR:
 6. gray_image \leftarrow BRG2GRAY(masked_image)
 7. blurred_image \leftarrow GaussianBlur(gray_image)
 8. binary_thresholded_image \leftarrow bin_threshold(blurred_image). Pixels turned on are set to white, otherwise to black.
 9. Initialize a list that will store the vertical coordinate of the upper-right corner of each bounding box (yColor)
 10. Initialize a list that will store the horizontal coordinates of the bounding boxes (ys)
 11. Initialize a list that will store the vertical coordinates of the bounding boxes (xs)
 12. contours \leftarrow findContours(binary_thresholded_image)
 13. **for** i in contours:
 14. area \leftarrow contourArea(i)
 15. **if** area > CA:
 16. drawContour(image, i)
 17. perimeter \leftarrow arcLength(i)
 18. corners \leftarrow approxPolyDP(i, 0.015*perimeter)
 19. x, y, width, height \leftarrow boundinRect(corners)
 20. **if** width/height \geq WHR:
 21. yColor \leftarrow store(y)
 22. xs \leftarrow store(x, x+width)
 23. ys \leftarrow store(y, y+height)
 24. drawRectangle(img, (x, y), (x+width, y+height))
 25. **if** yColor NOT empty:
 26. coordinates \leftarrow group if yColor if < MVD value.
 27. Return dictionary with coordinates and colour name.

The final algorithm (algorithm 2) takes an image as input and return the corresponding numerical code encrypted in the colour bands. All colour detection functions are integrated in its architecture. Addition-

ally, this algorithm sorts the obtained coordinates of each colour to be able to output the numerical code in the correct order. Regarding the order in which the coordinates are sorted, this is defined by the position of the different colour bands, meaning that if the bands are placed closer to the top of the slab, the reading direction is downwards, while the reading direction changes to upwards if the bands are placed closer to the bottom of the slab. Essentially, the colour code has to be decrypted starting with the bands which are close to one of the horizontal edges, and the algorithm knows this by checking which colour band, the higher or the lower one, is close to a horizontal edge. By knowing which colour corresponds to which number, retrieving the coordinates of each colour band, and defining the appropriate reading direction, the algorithm can output the encrypted code correctly.

Algorithm 2. [code_decryptor(image, $P_{10 \times 10}$)]

Input: An image and a matrix P with all parameters for each colour detection function.

Output: The numerical code encrypted in the colour bands draw on the granite slates.

1. dict(coordinates, color_name) \leftarrow colour_detector(image,)
2. dict(sorted_coordinates, sorted_color_names) \leftarrow sort(coordinates, color_name)
3. numerical_code \leftarrow convert the sorted_color_names into numerical code according to Table 1.
4. Return the numeric code contained in the image.

2.3. Training and validation

For this task, the database, which contains a total of 130 pictures, is divided into two different sets: the training set which has 109 images, and the validation set with 21 images, which contain all colours studied and are representative of the characteristics of the database as well as the main challenges for the computer vision system. All functions are trained and validated respectively on those images that have the corresponding target colour. The same database structure was used to implement the final algorithm, which is capable of detecting all colours and decrypting the code associated. In the training process, each function analyses those images which contain their respective colour. If the detection result agrees with the ground truth provided for every image, the combination of parameters that achieved a satisfactory output is rewarded. Finally, the combination of parameters that has the better performance is considered the final input values for each corresponding colour function.

Since the colour functions need initial parameters to start off the training process, they are given approximate values calculated manually based on the HSV definition of each colour for the hue, value, and saturation parameters, while CA, CR, WHR, and MVD are iterated from zero to their highest value, being the

last one determined from direct analysis of the image features.

3. RESULTS AND DISCUSSION

The training process for each function defined the ideal values for the parameters that the colour detection functions depend on. These results are presented in Table 3, where the first column contains the name of the different colours present on the slabs. The next four columns contain information regarding the geometrical characteristics of the colour bands, and the rest of the columns present the hue, saturation and value limits for each colour threshold.

It is noticeable how, in the training process, the colour ratio (CR) between the number of active pixels after applying the thresholds and the total number of pixels in the image tends to zero. This is due to the presence of some faint and weathered colour bands that would not be detected otherwise. If CR took higher values, some of these images, which actually contain areas of interest, would get discarded and count as a classification error. Additionally, the results presented in Table 3 show how the hue measure is the most effective in isolating different colours compared to the saturation and value. The difference between the maximum and minimum hue values is the smallest in all colours except black, which need to be isolated with the aid of the value variable. This reduces the importance of the two other parameters (saturation and value), which leads to them having more wide ranges explaining the reasoning behind the higher limit of the saturation and value threshold having the maximum number possible (255) in almost all cases. An exception to this rule is the colour black, which has to be isolated with the aid of the value parameter.

The conditions and the success rate of every colour in the training phase along with the computation time are included in Table 4. During the development of the program, it was noticed that the colours red and brown display two different varieties. The changes in the shade of brown have their origin in the weathering of the colour, resulting in two categories: dark brown and light brown. In the case of the red colour, we hypothesize that its variance is heavily dependent on the brightness conditions under which the picture was taken. The presence of clouds in the sky or shadows projected on the granite slabs seems to be a likely cause for a high hue value, while if the light rays hit the granite slab and the shadow is projected backward the value of hue for the red colour tends to be much lower as it is shown in Table 3. In order to improve the accuracy, a double mask method was implemented in the algorithm, which works by defining two distinct thresholds for each colour variety but still identifying them as the same colour.

The results of the validation process for each colour function are presented in Table 5. It can be seen in the

TABLE 3. Final input parameters for each colour function obtained by the algorithm after the training process on the 109 images available in the database. The first four parameters are directly related to the geometrical features of the colour bands, while the six last columns present the different ranges in the hue, saturation, and value for the detection of each colour.

Colour	CA	CR	WHR	MVD	Hmin	Smin	Vmin	Hmax	Smax	Vmax
Black	50	0	0.8	16	0	0	3	179	255	51
Dark Brown	300	0	1.2	72	5	49	64	15	255	128
Light Brown	300	0	1.2	72	14	74	132	18	255	165
High-hue Red	50	0	0.4	22	171	100	103	179	255	255
Low-hue red	50	0	0.4	30	1	93	97	5	255	255
Orange	250	0	1.1	29	6	112	116	15	255	255
Yellow	150	0	0.2	16	23	69	42	43	255	255
Green	150	0	0.8	18	40	60	0	80	255	255
Blue	100	0	0.6	32	81	113	0	109	255	255
Purple	200	0	0.7	30	120	21	81	155	255	255

TABLE 4. Results of the training process for each colour detection function. The average success rate stands at 84.28%. The colour light brown shows the lowest accuracy due to its similarity with orange.

Colour	Images	Success rate	Total time (sec.)
Black	32	90.32%	0.37
Dark Brown	9	100.00%	0.13
Light Brown	32	57.69%	0.32
High-hue Red	36	83.33%	0.29
Low-hue red	65	79.69%	0.54
Orange	57	78.95%	0.47
Yellow	60	88.33%	0.50
Green	40	95.00%	0.32
Blue	32	83.97%	0.28
Purple	38	85.53%	0.44

fourth column that the times required for each calculation decrease in direct proportion to the number of images analysed. The images in this set are completely new for the algorithm, meaning that it has not been

trained on them. Therefore, the high success rates in this process confirm that the parameters defined in the training stage are accurate, and the model can read the correct codes of new images which makes it valid.

TABLE 5. Results for the validation process for each colour detection function. All colours average a success rate of 86.36%.

Colour	Images	Success rate	Total time (sec.)
Black	8	88.89%	0.14
Dark Brown	4	75.00%	0.04
Light Brown	4	75.00%	0.04
High-hue Red	6	100.00%	0.09
Low-hue red	15	81.25%	0.13
Orange	10	80.00%	0.09
Yellow	13	85.71%	0.16
Green	9	100.00%	0.07
Blue	8	88.89%	0.09
Purple	9	88.89%	0.09

After the training and validation of all colour detection functions, these are included in the main architecture of the final algorithm, which is tasked with the detection of each colour band and its conversion into the correctly ordered numerical code. In this case, a result is classified as true when the output of the algorithm matches the exact numerical code an employee would read off the granite slab. Those codes obtained from the algorithm whose numbers are not the same and/or are not placed in the right orders would be considered faults. The colours which present a lower success rate tend to have an important negative effect on the final result. Particularly, light brown can be mistaken for orange, and dark brown can be confused with the low-hue red due to their proximity on the HSV colour space. These conditions make it difficult for the algorithm to achieve a very precise detection on all colours, which hijacks the results leading to a faulty detection, as can be seen in Figure 6.

The replacement of the colour brown is proposed in order to tackle this issue. The spray paint colours cyan and light green are encouraged due to their strong contrast with the current colours in use, as

shown in the HSV hexacone contained in Figure 1. Moreover, the election of the colour pink was motivated by its great endurance to weathering, as it is demonstrated by Alonso-Villar *et al.* (36). According to these authors (36), the use of silicate-based paint can be considered as a future line of research to solve the detection problem associated with the erosion of the spray paints currently employed.

Additionally, most colour detection systems have to operate within steady light conditions (3, 5, 37, 38) and require stable camera positioning (39-42), while the system proposed, although not perfect, achieves results in very different conditions. In this case, the algorithm shows a success rate of 74.42% on the training set, while the validation shows an efficiency of 75.00%. Several examples of correct detection are displayed in Figure 7.

Several results of the validation process for the final algorithm are shown in Table 6. The first column contains the ground truth for each image, which is the numerical code defined by the colour bands in each case. The last column shows the numerical code decrypted by the algorithm from the information represented in the corresponding image of the granite slabs. The rest of the columns contain each of the sorted numbers outputted by the system. According to the success condition previously defined, a result is correct when all numbers match the ones present in the ground truth in the correct order. Two errors can be noted in red in Table 6. The result of the fourth code presented (24036) is affected by the detection of a black area out of place, and the same happens in the case of the fifth code of the table (24066). Consequently, the rest of the numbers in the code decrypted are displaced by these anomalous detections of black areas leading to a faulty result. Considering the fact that granite has important black areas in its structure due to the presence of mica, this becomes the main

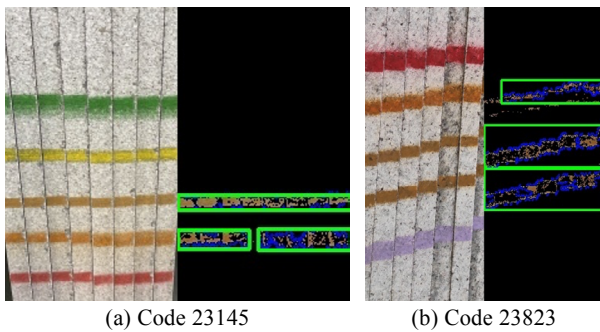


FIGURE 6. Faults in the detection of the colour brown in the training set: (a) and (b) failure in the identification of the colour brown due to its proximity to orange in the HSV colour space.

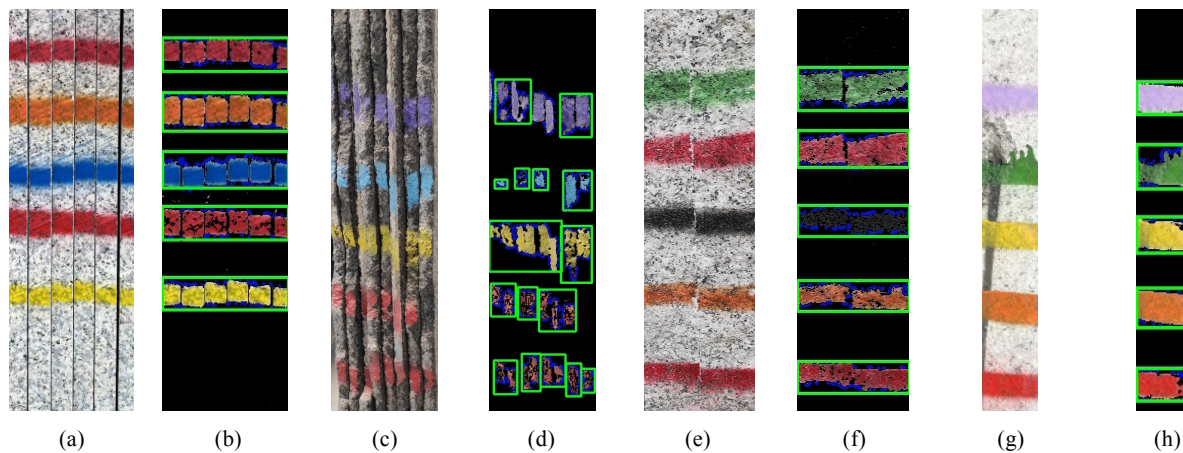


FIGURE 7. Several examples of correct detection of the different colour bands and an accurate output of the code represented by those. Images (b) and (e) display the results yielded by the algorithm in the training set, while image (h) belongs to the validation set. Images (c) and (d) show how the algorithm is able to detect colour bands that are disrupted in the horizontal axis.

TABLE 6. Detailed results obtained by the algorithm on some examples of the validation set. The success condition turns all incorrect and/or out-of-place digits into wrong detections marked in red.

Code	Number 1	Number 2	Number 3	Number 4	Number 5	Result
22475	2	2	4	7	5	22475
23405	2	3	4	0	5	23405
23457	2	3	4	5	7	23457
24036	4	0	0	3	6	40036
24066	2	4	0	6	0	24060
24227	2	4	2	2	7	24227
24465	2	4	4	6	5	24465
24526	2	4	5	2	6	24526

source of errors, despite the final algorithm having an accuracy of 75.00% in the testing phase.

4. CONCLUSIONS

The traceability problem in the granite industry has been studied in the present research paper. The current method to keep track of the granite blocks until they reach the final product stage is fairly simple and cheap, consisting in the usage of graffiti spray to assign a color-coded number to each granite block. Weathering of the paint and human error due to different causes, such as fatigue, lead to a faulty reading of the numerical code represented by colour bands and ultimately to economic losses for the industry as well as pollution originated by the granite that has not been traced. In this paper, a computer vision algorithm is proposed to automate the traceability process in this industry. The computational method developed operates in the HSV colour space and makes use of a double threshold for the correction isolation of colours on the granite slabs. This is followed by the conversion of the thresholded image to grayscale and the application of Gaussian blurring to soften the edges. After the binarization of the blurred image, contours are detected and the coordinates of those shapes that approximate to rectangular colour bands are extracted. Lastly, those coordinates are sorted and converted into the decrypted numerical code.

The proposed system performs an accurate detection of a combination of seven out of the eight colours used in the granite industry. It shows success rates on the reading of colour codes and the output of their corresponding numerical codes of 74.42% in the training set and 75.00% in the validation set. Moreover, these results are achieved on images that display different lighting conditions and were not taken with a fixed target distance, which increases the difficulty in the detection of different colours and adds up to the value of the results presented. In conclusion, the computer vision system presented is considered to be

helpful in the granite manufacturing industry owing to its accuracy, and it is part of the first approach towards a full implementation on mobile phone devices for the corresponding users in each factory.

ACKNOWLEDGEMENTS

This work was supported by Project PID2020-116013RB-I00 financed by MCIN/AEI/10.13039/501100011033

AUTHOR CONTRIBUTIONS:

Conceptualization: J. Martínez, M. Araújo. Data curation: X. Rigueira, M. Araújo. Formal analysis: J. Martínez. Funding acquisition: E. Giráldez. Investigation: X. Rigueira, M. Araújo, J. Martínez. Methodology: X. Rigueira, J. Martínez. Project administration: E. Giráldez. Resources: A. Recamán. Software: X. Rigueira. Validation: X. Rigueira, M. Araújo. Visualization: X. Rigueira, J. Martínez. Writing, original draft: X. Rigueira. Writing, review & editing: M. Araújo, J. Martínez.

REFERENCES

- Dirección General de Política Energética y Minas. (2019) Estadística minera de España 2019. Retrieved from https://energia.gob.es/mineria/Estadistica/DatosBibliotecaConsumer/2019/estadistica_minera_anual-2019.pdf.
- Qi, C. (2020) Big data management in the mining industry. *Int. J. Miner., Metall. Mater.* 27 [2], 131-139. <https://doi.org/10.1007/s12613-019-1937-z>.
- Anh Vo, S.; Scanlan, J.; Mirowski, L.; Turner, P. (2018) Image processing for traceability: A system prototype for the Southern Rock Lobster (SRL) supply chain. *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 1-8. Retrieved from <https://eprints.utas.edu.au/29370>.
- Araújo, M.; Martínez, J.; Ordóñez, C.; Vilán, J.A. (2010) Identification of granite varieties from colour spectrum data. *Sensors (Basel)*. 10 [9], 8572-8584. <https://doi.org/10.3390/s100908572>.
- Underwood, E.E. (1973) Quantitative stereology for microstructural analysis. microstructural analysis. Springer, Boston, M.A., (1973). https://doi.org/10.1007/978-1-4615-8693-7_3.
- Underwood, E.E. (1986) Quantitative fractography. Applied metallography. Springer, Boston, M.A., (1986). https://doi.org/10.1007/978-1-4684-9084-8_8.

7. Russ, J.C.; Neal, F.B. (2016) The image processing handbook (7th ed.). CRC Press, Boca Raton, F.L., (2016). <https://doi.org/10.1201/b18983>.
8. Serra, J. (1982) Image analysis and mathematical morphology. Academic Press. Cambridge, M.A., (1982).
9. Iglesias, C.; Martínez, J.; Taboada, J. (2018) Automated vision system for quality inspection of slate slabs. *Comput. Ind.* 99, 119-129. <https://doi.org/10.1016/j.compind.2018.03.030>.
10. Martínez, J.; López, M.; Matías, J.M.; Taboada, J. (2013) Classifying slate tile quality using automated learning techniques. *Math. Comp. Model.* 57 [7-8], 1716-1721. <https://doi.org/10.1016/j.mcm.2011.11.016>.
11. López, M.; Martínez, J.; Matías, J.M.; Vilán, J.A.; Taboada, J. (2010) Application of a hybrid 3D-2D laser scanning system to the characterization of slate slabs. *Sensors (Basel)* 10 [6], 5949-5961. <https://doi.org/10.3390/s100605949>.
12. Ozkan, F.; Ulutas, B. (2016) Use of an eye-tracker to assess workers in ceramic tile surface defect detection. Proceedings of the International Conference on Control, Decision and Information Technologies (coDIT). <https://doi.org/10.1109/CoDIT.2016.7593540>.
13. Hanzaei, S. H.; Afshar, A.; Barzandeh, F. (2017) Automatic detection and classification of the ceramic tiles' Surface defects. *Pattern. Recogni.* 66, 174-189 <https://doi.org/10.1016/j.PATCOG.2016.11.021>.
14. Sioma, A. (2020) Automated control of surface defects on ceramic tiles using 3D image analysis. *Materials (Basel)* 13 [5], 1250. <https://doi.org/10.3390/ma13051250>.
15. Hocenski, Z.; Matic, T.; Vidovic, I. (2016) Technology transfer of computer vision defect detection to ceramic tiles industry. Proceedings of the International Conference on Smart Systems and Technologies (SST). 301-305. <https://doi.org/10.1109/SST.2016.7765678>.
16. Samarawickrama, Y.C.; Wickramasinghe, C.D. (2017) Matlab based automated surface defect detection system for ceramic tiles using image processing. Proceedings of the National Conference on Technology and Management (NCTM). 34-39. <https://doi.org/10.1109/NCTM.2017.7872824>.
17. Avci D.; Sert, E. (2021) An effective Turkey marble classification system: Convolutional neural network with genetic algorithm-wavelet kernel-extreme learning machine. *Colloq. Traitement. Signal. Imag.* 38 [4], 1229-1235. <https://doi.org/10.18280/ts.380434>.
18. Panda, G.; Satapathy, S.C.; Biswal, B.; Ramesh, B. (2028) Microelectronics, electromagnetics and telecommunications. Proceedings of the International Conference on Micro-Electronics, Electromagnetics and Telecommunications (ICMEET). Retrieved from <https://www.springerprofessional.de/en/microelectronics-electromagnetics-and-telecommunications>.
19. López, M.; Martínez, J.; Matías, J.M.; Taboada, J.; Vilán, J.A. (2010) Functional classification of ornamental stone using machine learning techniques. *J. Comput. App. Math.* 234 [4], 1338-1345. <https://doi.org/10.1016/J.CAM.2010.01.054>.
20. Kang, H. (2006) Computational Color Technology (1st ed.). Spie Press, Bellingham, WA.
21. Bianconi, F.; Fernández, A.; González, E.; Saetta, S.A. (2013) Performance analysis of the colour descriptors for parquet sorting. *Expert. Syst. Appl.* 40 [5], 1636-1644. <https://doi.org/10.1016/j.eswa.2012.09.007>.
22. Paschos, G. (2000) Fast colour texture recognition using chromaticity moments. *Pattern Recognit. Lett.* 21 [9], 837-841. [https://doi.org/10.1016/S0167-8655\(00\)00043-X](https://doi.org/10.1016/S0167-8655(00)00043-X).
23. Xiong, N.N.; Shen, Y.; Yang, K.; Lee, C.; Wu, C. (2018) Color sensors and their applications based on real-time color image segmentation for cyber physical systems. *EURASIP J Image Video Process.* 2018, 23. <https://doi.org/10.1186/s13640-018-0258-x>.
24. Ibraheem, N.A.; Hasan, N.M.; Khan, R.Z.; Mishra, P.K. (2012) Understanding color models: a review. *ARPN J. Eng. Appl. Sci.* 2 [3], 365-275. Retrieved from https://haralick.org/DV/understanding_color_models.pdf.
25. Sebastian, P.; Voon, Y.V.; Comley, R. (2010) Colour space effect on tracking in video surveillance. *Int. J. Electr. Eng. Inform.* 2 [4], 298-312. <https://doi.org/10.15676/ijeei.2010.2.4.5>.
26. Smith, A.R. (1978) Color gamut transform pairs. Proceedings of the Conference on Computer Graphics and Interactive Techniques ACM SIGGRAPH Computer Graphics. 12 [3], 12-19. <https://doi.org/10.1145/800248.807361>.
27. Roger, D.F. (2016) Procedural elements of computer graphics (1st ed.). McGraw-Hill, New York City, New York, (2016).
28. Bhatia, P.K. (2013) Computer graphics (3rd ed.), I.K. International, Daryaganj, New Delhi, Delhi, (2013).
29. Shapiro, L.; Stockman, G. (2001) Computer vision (1st ed.), 137-150. Prentice Hall., New York City, New York, (2001). Retrieved from <https://theswissbay.ch/pdf/>.
30. Nixon, M.; Aguado, A. (2019) Feature extraction and image processing for computer vision (1st ed.), 650. Academic Press, Cambridge, MA, (2019).
31. OpenCV: The OpenCV reference manual. 2.4.13.7 edn. OpenCV, (2014). OpenCV
32. Suzuki, S.; Abe, K. (1985) Topological structural analysis of digitalized binary images by border following. *Comput. Vis Image Underst.* 30 [1], 32-46. [https://doi.org/10.1016/0734-189X\(85\)90016-7](https://doi.org/10.1016/0734-189X(85)90016-7).
33. Edwards, C.; Penney, D. (1982) Calculus and analytical geometry (1st ed.), 859-866. Prentice Hall, Upper Saddle River, NJ, (1982).
34. Ramer, U. (1972) An iterative procedure for the polygonal approximation of plane curves. *Comput. Graph. Image Process.* 1 [3], 244-256. [https://doi.org/10.1016/S0146-664X\(72\)80017-0](https://doi.org/10.1016/S0146-664X(72)80017-0).
35. Douglas, D.H.; Peucker, T.H. (1973) Algorithms for the reduction of the number of points required to represent a digitalized line or its caricature. *Cartographica.* 10 [2], 112-122. <https://doi.org/10.3138/FM57-6770-U75U-7727>.
36. Alonso-Villar, E.M.; Rivas, T.; Pozo-Antonio, J.S. (2021) Resistance to artificial daylight of paints used in urban artworks. Influence of paint composition and substrate. *Prog. Org. Coat.* 154, 106180. <https://doi.org/10.1016/J.PORGCOAT.2021.106180>.
37. Kondo, N. (2009) Robotization in fruit grading system. *Sens. Instrum. Food Qual. Saf.* 3 [1], 81-87. <https://doi.org/10.1007/s11694-008-9065-x>.
38. Burgus-Artizzu, X.P.; Ribeiro, A.; Guijarro, M.; Pajares, G. (2011) Real-time image processing for crop/weed discrimination in maize fields. *Comput. Electron. Agric.* 75 [2], 337-346. <https://doi.org/10.1016/j.compag.2010.12.011>.
39. Carew, T.; Ghita, O.; Whelan, P.F. (2003) Exploring the effects of a factory-type test-bed on a painted slate defect detection system. Proceeding of the International Conference on Mechatronics (ICOM). 365-370. Retrieved from https://doras.dcu.ie/18806/1/whelan_2003_126.pdf.
40. Andrew, W.; Hannuna, S.; Campbell, N.; Burghardt, T. (2016) Automatic individual Holstein Friesian cattle identification via selective local coat pattern matching in RGB-D imagery. Proceedings of the International Conference on Image Processing (ICIP) vol. August 2016. 484-488. <https://doi.org/10.1109/ICIP.2016.7532404>.
41. Ghita, O.; Whelan, P.F.; Carew, T.; Padmapriya, N. (2005) Quality grading of painted slates using texture analysis. *Comput. Ind.* 56 [8-9], 802-815. <https://doi.org/10.1016/j.compind.2005.05.008>.
42. Ghita, O.; Carew, T.; Whelan, P. (2006) A vision-based system for inspecting painted slated. *Sens. Rev.* 26 [2], 108-115. <https://doi.org/10.1108/02602280610652695>.