Technical University "Gheorghe Asachi" Iasi

Faculty of Electronics, Telecommunications and Information Technology

Specialization: Technologies and Telecommunications Systems

DEGREE THESIS

Tutoring teacher:

Prof. Dr. Eng. Nicolae Cleju

Graduate:

Vieriu-Avădăni Ștefania

IASI 2021-2022

Technical University "Gheorghe Asachi" Iasi

Faculty of Electronics, Telecommunications and Information Technology

Specialization: Technologies and Telecommunications Systems

Iris detection from face images

Tutoring teacher:

Prof. Dr. Eng. Nicolae Cleju

Graduate:

Vieriu-Avădăni Ștefania

content

Supporting memorandum

The perception of visual stimuli is the most important human sense, being an essential component for information processing. Much of the information taken in and processed by the brain gives us significant details about the size, shape and color of an object.

Continuous development is required in the case of image acquisition and analysis applications to enable devices to evaluate and extract information based on visual stimuli. Nowadays, thanks to advanced technology, there are types of applications capable of helping or completely replacing the human observer in many areas of activity such as: driving on automatic pilot, locating and tracking objects, identifying people or locations, etc.

Iris recognition or iris detection is a process that uses visible light or near-infrared light to take a high-contrast photo of a person's iris. This is an automated identification method that uses mathematical techniques to recognize patterns in images or video files of one or both irises of a person's eyes, whose complex patterns are unique, stable and can be seen from a certain distance.

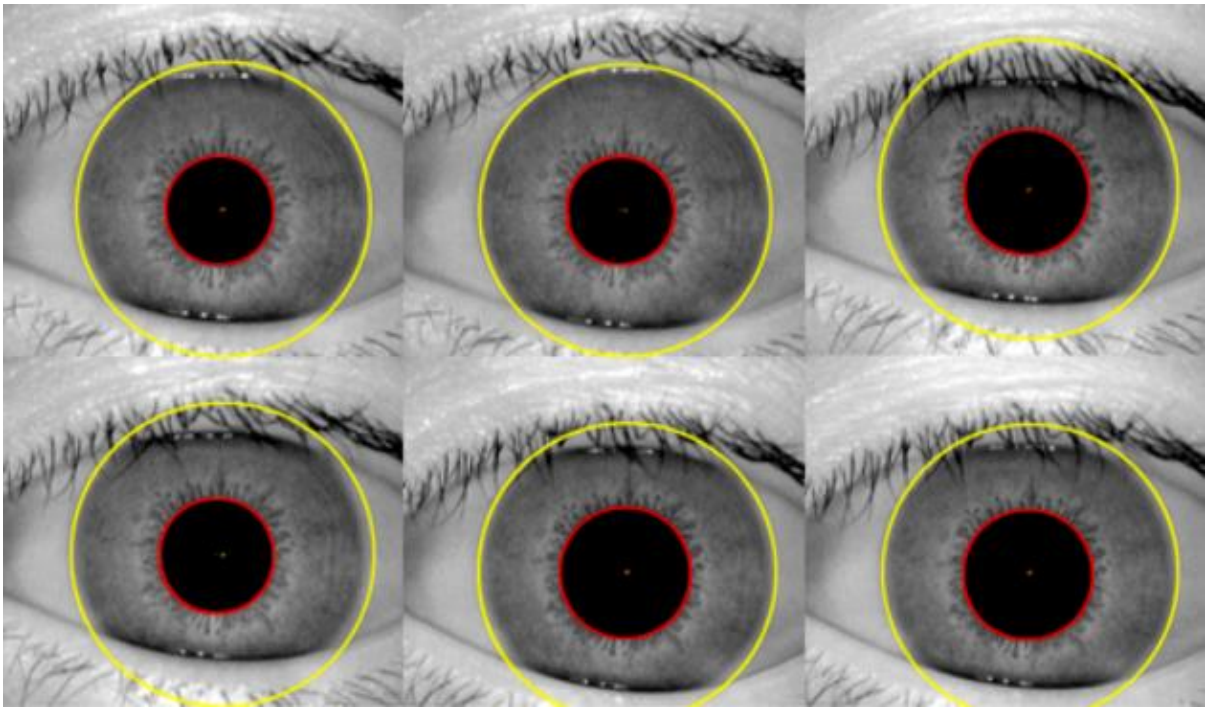


Figure 0.1: Iris and pupil recognition[1]

Iris scan cameras can be mounted on a wall or other fixed location, or they can be portable as well as hand-held. Researchers at Carnegie Mellon University are developing long-range scanners that can be used to stealthily capture images from up to 40 meters away.

This system first recognizes the face, then recognizes the eyes, and then determines whether the eyes are open or closed. The system process uses the information obtained from the binary version of the image to find the edges of faces, narrowing down the areas where eyes can exist.

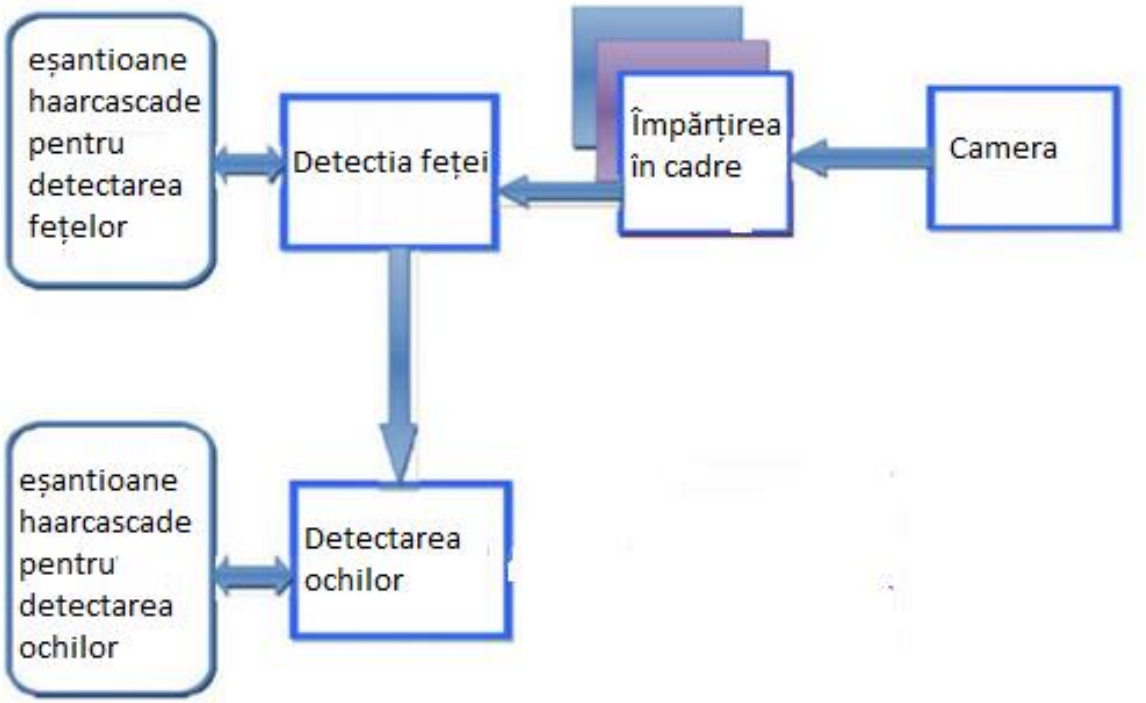


Figure 0.2: Schematic of the application

Starting with these first words, this theme is the result of a curiosity to help improve the fatigue detection system through image processing. The paper "Iris detection from face images" addresses the theoretical and practical part of an image processing application in the MATLAB program, which offers image retrieval, analysis and manipulation functions through access to images stored in the computer, or those received from webcam.

1. CHAPTER I: Introduction

Image processing is a method in which some operations are performed on an image to get an improved image or to extract some useful information from it. It is a type of signal processing where the input is an image and the output can be an image or features associated with that image. This type of detection is among the technologies preferred by the majority of users, forming a basic field also within the disciplines of engineering and computer science.

Image processing basically includes three simple steps:

- Image import via image acquisition tools (photos, web camera); Image
- analysis and manipulation;
- The output where the output can be the altered image or the report which is based on the image analysis;

There are two types of methods used for image processing, namely, analog or digital image processing.

Analog image processing can be used for hard copies such as prints and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help to manipulate digital images through the use of computers. The three general phases that all types of data must go through while using the digital technique are preprocessing, enhancement and display, information extraction.

Depending on the representation of the image (in analog or digital format) different advantages and disadvantages can be distinguished. Digital image processing has several advantages over analog processing. This method allows a wide range of algorithms to be applied to the input data through circuits that are much easier to make, and on the other hand allow the images to be modified without the risk of creating additional noise or inducing unwanted distortions.

Therefore, digital implementation can provide much better performance in simple processes than analog signal processing methods, which would be impossible and unreasonably expensive most of the time.

1.1 Brief history

Image manipulation dates back to the time of the first photographs taken in the 19th century. The history of photography began in antiquity with the discovery of two critical principles: the projection of the image in the camera obscura and the observation that some substances are visibly altered by prolonged exposure to sunlight. There are no artifacts or descriptions that can prove any attempt to capture images with light-sensitive materials before the 18th century.

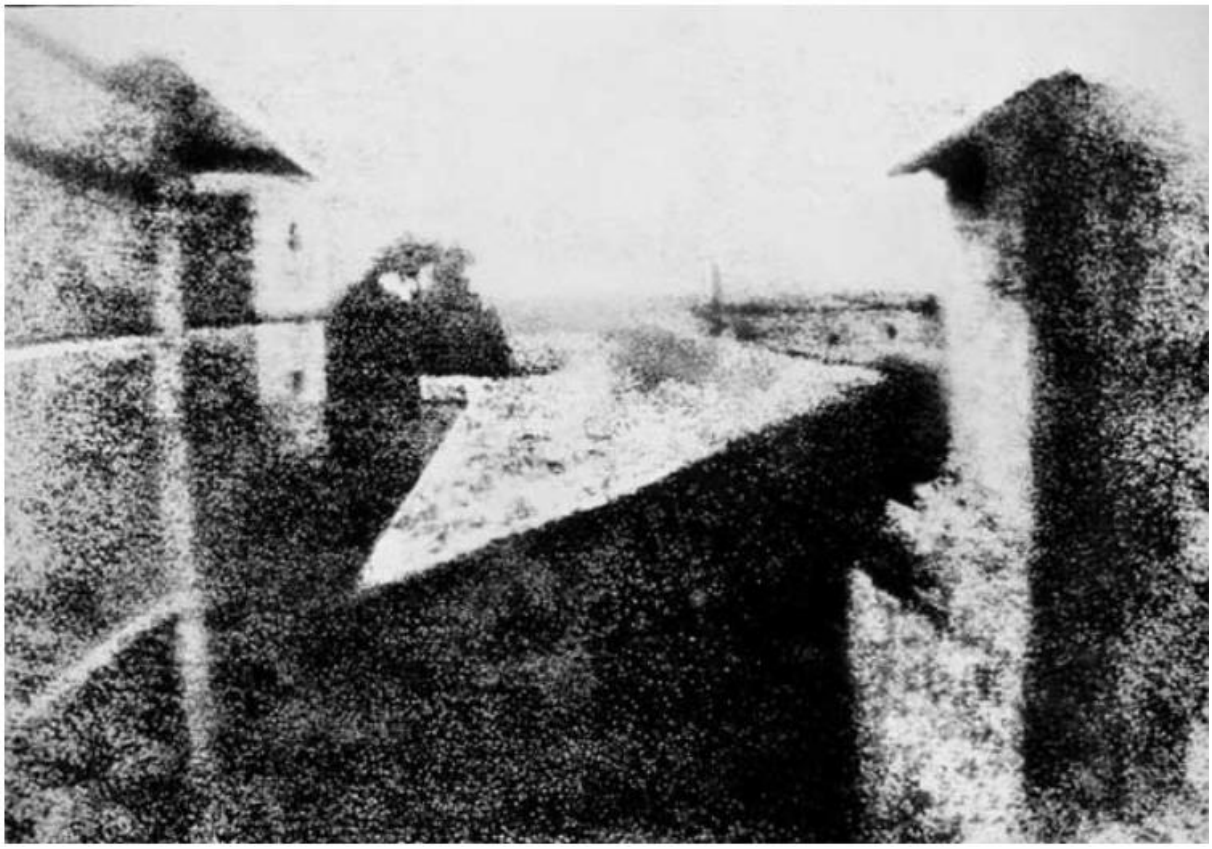


Figure 1.1: "View from the window of Le Gras" - the first photograph in the world which presents a pose from reality [2]

The developed images were modified by means of ink touch-ups or by removing the ink by scratching. Undeveloped photographs (negatives) were only modified inside darkrooms due to the materials used which were photosensitive.

These techniques were largely influenced by the limitations of technology. An example would be negatives that could just be cut and pasted or bleached to brighten the frame or some objects altogether.

The influence on the evolution of image processing was had by 3 factors: the development of computing systems capable of processing a large amount of information, the development of the mathematical methods behind the processing and compression algorithms (the creation and development of the theory of discrete mathematics), but not finally, the need in several industries for applications that allow computers to extract important information from the images received from a digital camera (especially in the military, medical, astronomy and industry fields).

Due to the extremely accelerated evolution of technology, representation in digital format has become more and more popular, and technology has migrated to this new format of data representation being easily stored, processed and redistributed to other devices on an extremely large variety of transmission paths, with a much better error detection rate than analog signals.

1.1.1 Digital processing techniques

The vast majority of digital image processing techniques were developed after 1960 in order to make people's lives easier and were especially used in the following applications: satellite imagery, medical imaging, video teleconferencing and finally in enhancing photographs and character recognition. The purpose of these processing techniques was to improve the quality of images, but also to encode or compress them.

After the rediscovery of the "Fast Fourier Transform (FFT)" it became possible to manipulate the frequency content of signals on a computer. Finally, a remarkable application made in 1964 would revolutionize the rapidly developing process of image processing and information transmission. It applies geometric correction and noise removal procedures to the many pictures taken by a space probe, generating the first detailed map of the moon.

As the processing power of computers increased considerably, it became possible for images to be processed in real time, and thus the first TV standards for converting and compressing audio-video signals appeared.

1.1.2 Image acquisition sensors

The human visual system and the digital imaging system have the same functions and purposes. Both analyze visual, spatial data with the goal of extracting information and recognizing specific objects or aspects in images. Given that the human visual system is too complex and the neural system has not been fully understood, even the best computing and image acquisition systems fail to match it. Rather than shaping or replacing the way people perceive information, digital image acquisition and processing techniques have been directed at enhancing it.

Humans perceive images based on visual stimuli, in the form of photons, which reach the nearly 130 million photoreceptor cells on the retina. The photoreceptor cells responsible for color perception are of 3 types, one type for each part of the light spectrum: blue, green and red.

The basis of modern imaging sensors is metal oxide semiconductor (MOS) technology, a technology that triggered the appearance of the first MOS field-effect transistor (MOSFET = metal oxide semiconductor field-effect transistor). Designed by Mohamed M. Atalla and Dawon Kahng in 1959, it led to the development of semiconductor (CCD and later CMOS) image sensors.

CCD (charge-coupled device) technology was invented by Willard S. Boyle and George E. Smith in 1969. While studying MOS technology, they discovered that an electric charge can be compared to a magnetic bubble and that it can be stored in a small parasitic capacitance inside a MOS device. After putting several such capabilities together and applying a voltage source to them, the two built a semiconductor circuit, which was later used in the first digital video cameras that transmitted images to the general public.

The CMOS sensor (CMOS active-pixel sensor) was developed by a team led by Eric Fossum, within NASA, the result being the outselling of these sensors compared to the old technologies (CCD).

1.2 Theoretical foundation

Digital image processing is based in practice on:

- Classification

- Feature extraction

- Analysis of a signal on several levels

- Segmentation

- Transformations

- Shape recognition

Some of the techniques used in image processing are: anisotropic diffusion, editing, restoration, independent component analysis, median filtering, neural networks, partial differential equations, pixelation, feature identification, etc.

Digital photos are characterized by a series of properties influenced by the environmental conditions in which they are taken (contrast, brightness, saturation, exposure, etc.), the device that took the photo (resolution, file extension - implicitly the type of compression used, resolution, space of color, etc.).

1.2.1 Extension of an image

A first step in the digital processing of an image is to identify the format and compression under which it was stored. These extensions are divided into 2 categories:

1) The raster image is represented in computer systems in the form of a two-dimensional matrix, where each pixel corresponds to the intersection between the line and column of the matrix. They contain most of the formats used for the web:

- GIF (Graphics Interchange Format): used especially in WEB pages because it allows transparent pixels, but contains at most 256 colors;
- JPEG(Joint Photographic Experts Groups): a type of compressed image with minimal loss of quality used in consumer applications;
- PNG(Portable Network Graphics) appeared to replace the GIF format and allows up to 16 million colors.

2) The vector image uses a system of lines and curves in a Cartesian plane sized according to the entire area of the image. It allows an enlargement of the image without distorting or losing quality. The most common formats among vector images are: SVG (Scalable Vector Graphics), DXF (AutoCAD drawing exchange format), EPS (Encapsulated PostScript).

1.2.2 Color spaces

In order to be processed, the graphic images are digitized in the first phase, i.e. divided so that each element has only one color, or at least one clearly dominant color. This element, called a pixel, has three attributes that can be expressed digitally (numerically): color, opacity (transparency) and position in the matrix into which the image is divided.

If the pixels are very small and numerous, the complete presentation of the image on a screen or by printing can have a very high quality or optical fidelity that can be compared to the quality of ordinary (analog) printed images.

A very simple analysis can be done, for example, on a black and white image (binary image). In this way, the numbers in the two-dimensional matrix provide information related to the cell content (white/black) for the pixel with the same coordinates, so only black and white pixels appear in the image. A binary image is obtained from a grayscale image by a simple operation called thresholding.

Fig 1.2 Binarization of an image [3]

The next type of image is grayscale. In this case, each pixel is a value from 0 to 2^n-1, where n stands for the number of bits of the numerical representation. Most of the time, grayscale images have 256 shades, a large enough number that the difference between areas of a different shade is difficult for the human eye to see.

Increasing in complexity, the last type of image is represented by the color image. In this type, each pixel is a combination of the 3 primary colors: red, green and blue (RGB), to make up all the other colors of the spectrum. In addition to the RGB color coding system, there are several color spaces.

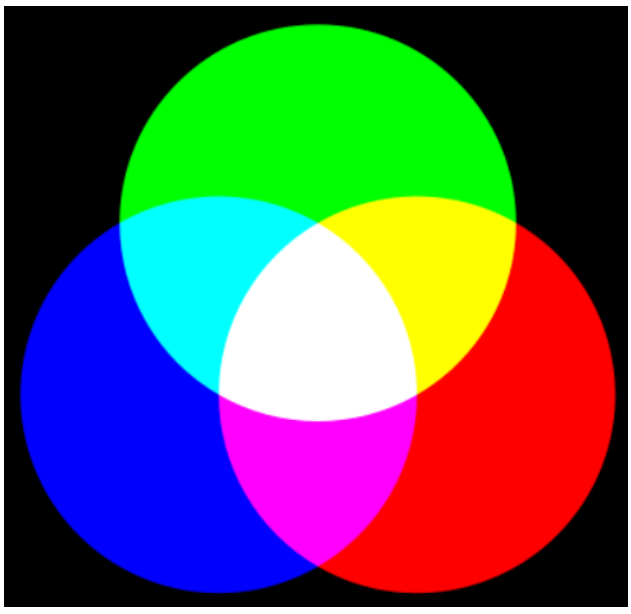


Fig 1.3: The RGB color model [4]

Other color spaces based on RGB are the following: RGBA (based on RGB, to which a new channel, alpha, is added, which indicates the absolute transparency of the pixel), Adobe RGB (created to compensate for colors that can be obtained by printing),s RGB (used in monitors, printers and web), scRGB and ProPhoto RGB.

CMYK is short for "**C**yan**m**agent,**Y**yellow, Blac**k".**This uses a subtractive method of combining colors (unlike RGB) and is used in digital color printing devices (printers, plotters), but also in large machines (offset printing, rotogravure). It consists of the mixture of substances that, reflecting only part of the light spectrum, appear in a specific color desired by the user. This color space stores color values for cyan, magenta, yellow, black, and may also be called four-color printing.

HSB, short for "Hue Saturation Brightness", is used because its approach to interpreting colors in terms of saturation and hue is much more natural than other approaches.

The HSL (hue, saturation, luminance) model is similar to the HSV space, but brightness is replaced by lightness.

1.2.3 Resolution

Digital resolution is a measure of the clarity or detail of a digital image. Images are sampled in a finite number of cells from an image sensor, and with this method, the obtained edge is degraded with respect to the object. In order for the image to be as close to reality as possible, the number of cells in the sensor must be as large as possible.

As the number of pixels present in the image is greater, the level of detail increases proportionally to the stage where it becomes as faithful a copy of reality as possible. Below we have an example where one can clearly see an image in several resolutions:

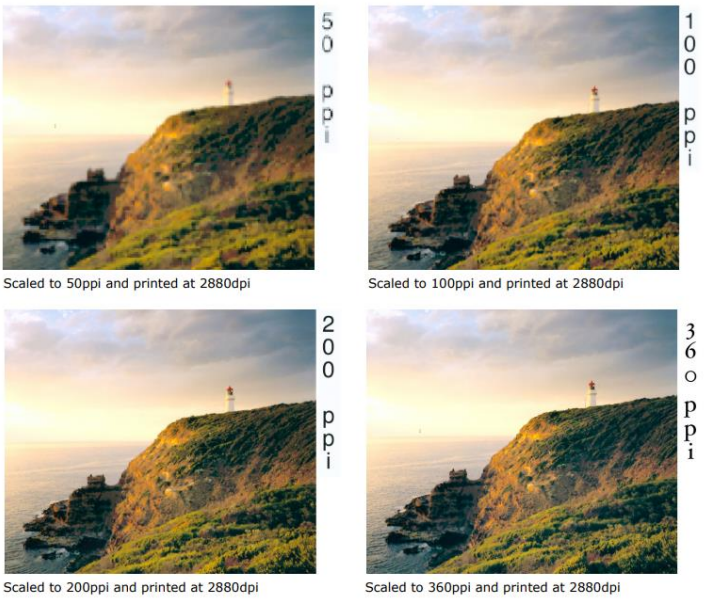


Figure 1.4: Resolution differences [5]

The spatial resolution of an image depends on the spatial density and optical resolution of the device used to capture the image. If the optical resolution of the capture device is higher than the spatial density, the spatial resolution of the resulting image is limited only by the spatial density.

According to the Nyquist criterion, a sampling interval at least 2 times larger than the highest high spatial frequency we want to capture is needed to obtain the highest possible fidelity of the resulting image.

Another alternative to Nyquist's criterion is Shannon's sampling theorem which states that the capture device must use a sampling interval less than half of the smallest representable feature of the optical image.

1.2.4 Luminance

Luminance is the amount of light detected by a typical human eye emitted by a flat surface. Luminance is represented by the ratio between the luminous intensity of a light-generating source and the projection of the source's area on a plane. In industry, this concept is used to characterize the brightness of a device's display.

$$L = \frac{d^2\Phi_v}{d\Sigma d\Omega_\Sigma cos\theta_\Sigma}$$

= luminance (brightness)

^2Φ =the luminous flux leaving the area unit Σ,in any continuous direction within the solid angle ΩΣ

=the infinitesimal unit of area

=the infinitesimal unit of the solid angle

Σ =is the angle between the normal to the surface Σand the specified direction

1.2.5 Contrast

The term contrast refers to the amount of color differentiation or grayscale that exists between different image features (in both analog and digital images). Images with a higher contrast level generally display a greater degree of color or grayscale variation than those with lower contrast.

Luminance contrast (Weber Contrast) is defined as the difference between the maximum brightness of an area in an image and the brightness of a dark area, all divided by the minimum brightness of the picture.

= (−)/

Peak-to-peak contrast (Michelson Contrast) measures the relationship between the difference and the sum of two luminances. This definition is often used in signal processing to determine the quality of a signal relative to noise.

h = (−)/(+)

1.2.6 Histogram

The histogram is a graph in which the relative frequencies of each pixel value within an image are represented, for each channel, separately. This helps to analyze and, even more, to correct the contrast of an image. Technically speaking, the histogram maps the luminance, which is defined starting from the way the human eye perceives the brightness of certain colors. For example, human eyes are much more sensitive to the color green, which means that the color will be seen much brighter than blue.

Each pixel in the color or grayscale image has a luminance value between 0 and 255. The histogram will graphically map each luminance value for each pixel. The length of each vertical line in the histogram shows how many pixels have a luminance value of 0, how many 1, 2, etc., up to a maximum value of 255.

The graph composed of vertical bars from which the histogram is made up shows the image of the relative distribution of tones over the entire scale of values. The image will not cover the entire possible black to white scale, which may suggest that the image could be improved in terms of contrast.

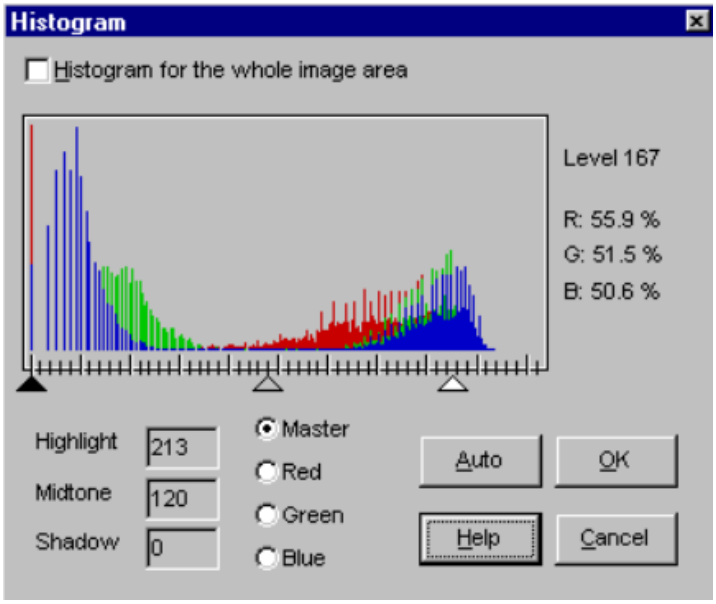


Figure 1.5: Example of histogram[6]

1.3 Image processing algorithms

To perform transformations on images, all processing techniques are based on mathematical calculation methods. In this way, all types of changes are reduced, from a mathematical point of view, to simple operations that can be easily performed by the computer.

1.3.1 Convolution

Convolution represents an operation that describes the evolution of the overlap between the areas of 2 functions, when one of them is moved on the time axis. This is defined according to the formula:

$$f(t) * g(t) := \int_{-\infty}^{+\infty} f(\tau)g(t-\tau)d\tau = \int_{-\infty}^{+\infty} f(t-\tau)g(\tau)d\tau;$$

In the case of digital processing of discrete signals and implicitly of images, the convolution has the expression:

$$f[n] * g[n] := \sum_{m=-\infty}^{+\infty} f[n]g[n-m] = \sum_{m=-\infty}^{+\infty} f[n-m]g[m];$$

In the formulas above, it can be seen that the convolution operation is commutative. The addition operation is much easier to perform by a computer than calculating the integral itself.

1.3.2 Gaussian Blur

Gaussian Blur is often called a bell curve because its probability density plot looks like a bell and it is assumed that during any measurement the values will follow a normal distribution with an equal number of measurements above and below the mean value. This is obtained by convolving some function with a Gaussian function of the form:

$$g(x) = a * \exp\left(-\frac{(x-b)}{2*c^2}\right)$$

- a= the height of the top of the graph b= the
- displacement of the center from the origin
- c= standard deviation and influences the width of the graph

For the coefficients a, b, c are constant real values and c different from 0, this function is named after the famous mathematician Carl Friedrich Gauss and is also called "Gauss's Bell" because of the shape it takes. But convolution between a Gaussian function and the original image limits rapid changes in pixel intensities and smoothes edges.

1.4 Image processing techniques

1.4.1 Classification

Segmentation

Segmenting an image in digital image processing is the process of dividing the image into meaningful units. This technique is used to obtain a more compact representation, to extract objects or as a tool for image analysis. It allows the partitioning of digital images into sets of pixels. The purpose of this process is to simplify or change the image representation, so that it can be much easier to analyze. In this way, pixels with similar characteristics belong to the same object.

The simplest segmentation method is binarization. The simplest case, for example, is an image containing a single object or several individuals that are on a background of different intensity compared to that of the objects. So in this case, the separation between the objects and the background is achieved by binarization.

Fig. 1.6: Separation of background objects by binarization [7]

Thresholding (Threshold Method)

The threshold method, also called "Thresholding", is another segmentation method. In this method, portions of the image are classified according to a range of values. So, it replaces each pixel in an image with a black pixel if the image intensity is less than a fixed value called a threshold, or a white pixel if the pixel intensity is greater than that threshold. For example, in the image below you can see that after this process the dark tree becomes completely black and the shiny snow becomes completely white.

Applying to a color image has the same result. A high threshold applied to the red color channel and a low threshold applied to the other channels will cause all red objects in the resulting image to fade out.

Local segmentation

On the other hand. another method used in numerous image processing algorithms is local segmentation. Compared to general segmentation, this method processes separate groups of pixels, eventually being associated to be able to recognize the outline and features of an object.

An important step in local segmentation is noise reduction. After this process, it is desired that the image preserves the structure of the objects, but to reduce as much as possible the level of disturbances introduced by the limitations of the sensor or the resolution of the quantization circuits.

The resulting local regions after the fragmentation process contain a small number of pixels, and the probability that a pixel belongs to 2 distinct groups is very small. This method of considering local clusters of pixels, then denoising them, is an important step that supports edge detection, pixel classification, and image compression operations.

1.4.2 Transformation

Image transformation involves converting the image from one domain to another. Visualizing an image in domains such as Hough space or frequency, allows the identification of features that might not be as easy to detect in the spatial domain.

The most common transformations used include:

The Hough transformis a feature extraction technique used in image analysis, computer vision, and digital image processing. The purpose of this technique is to find imperfect instances of objects from a certain shape class through a voting procedure.

Because it requires the desired features to be specified in a parametric form, the classical Hough transform is frequently used for detecting regular curves,

such as lines, circles, ellipses, etc. A generalized Hough transform can be used in applications where a simple analytical description of a feature is not possible. Due to the computational complexity of the generalized Hough algorithm, we limit the main focus of this discussion to the classical Hough transform.

Despite its domain restrictions, the classical transformation retains many applications because most fabricated parts (and many anatomical parts examined in medical images) contain feature boundaries that can be described by regular curves. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise.

Fig 1.7 Detection of circles using the Hough transform method [8]

in math,**Radon transform**is the integral transformation that takes a function f defined in the plane to a function Rf defined on the (two-dimensional) space of lines in the plane, whose value at a given line is equal to the line integral of the function over that line. The Radon transform, whose name is due to the famous mathematician Johann Radon, was introduced in 1917 and provided, on the other hand, a formula for the inverse transform. The complex analogue of the Radon transform is also known as the Penrose transform.

In the medical field, such as computed tomography, the purpose of this transformation is to obtain the image of a section of the object from projections obtained with the help of radiation

penetrated. So, for each line a one-dimensional projection of a section of the object is obtained.

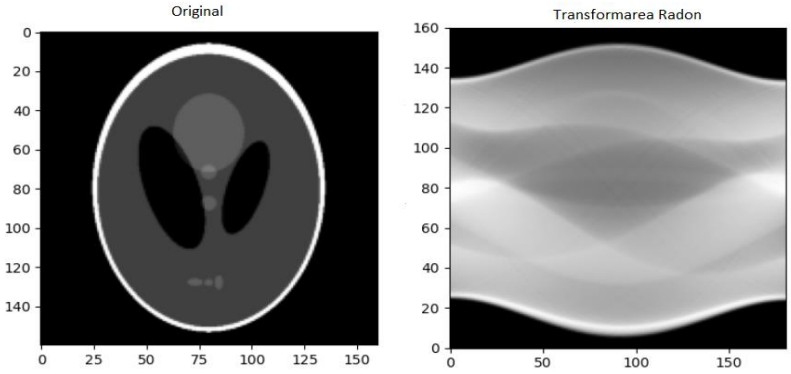


Fig 1.8: Applied Radon transform [9]

1.4.3 Median filter

The class of enhancement operations includes image filtering, the main purpose of which is to remove noise superimposed on an image. By filtering it is understood that the values of the neighboring pixels will also contribute to the calculation of the new value of a pixel. This set of pixels (neighborhood) can have various shapes and sizes, but the most common are odd-sized square-shaped neighborhoods. Median filtering is a filtering technique used to remove noise from images and signals.

The median filter is very important in the field of image processing because it is known to preserve edges while removing noise.

The median filter is suitable for removing salt and pepper noise. After ordering the pixel values, the noise values (ie 0 or 255) will be in the first or last positions, and at the filter output we will have a different value than the noise values.

There are situations where after filtering, after filtering there are pixels affected by noise. In this case, the filter was "pierced" by the noise. It is only possible if when more than half of the pixels selected by the filter window are affected in the same noise mode (salt or pepper, 255 or 0).

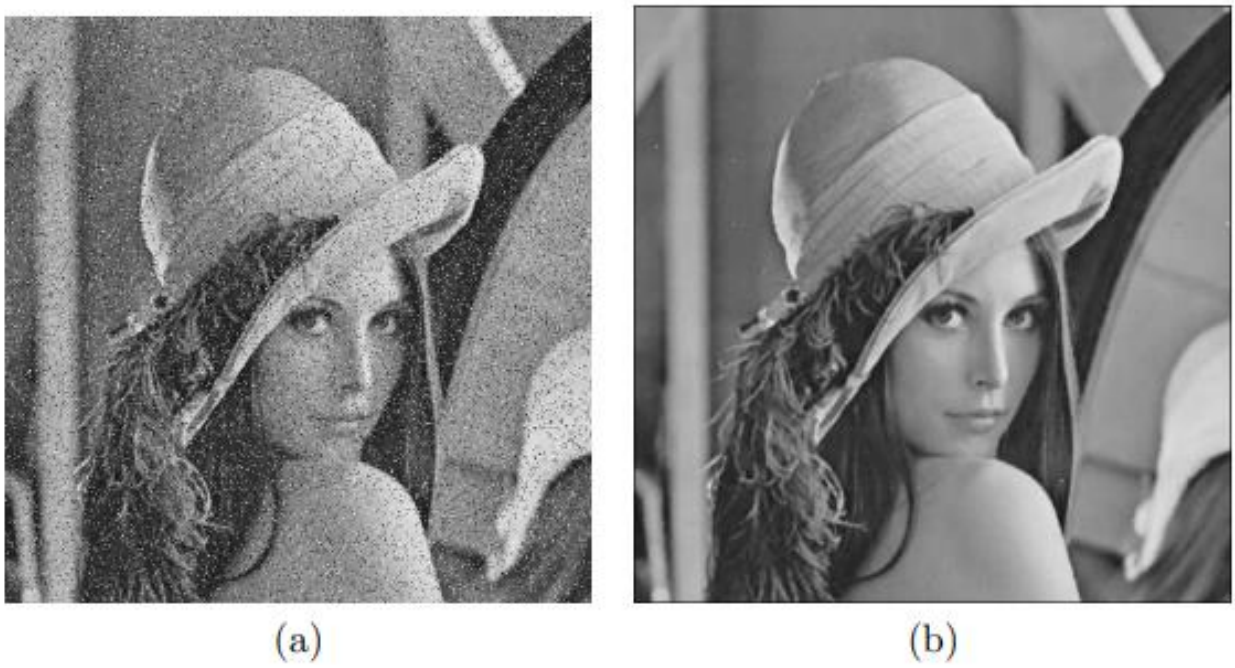


Fig 1.9: Representation of the median filter: a. The original image affected by noise "salt and pepper"; [10]

b. the filtered image

1.5 Identification of traits

Pattern recognitionin image processing it is based on the extraction of the characteristics based on which an initial set of measured data and the construction of some values resulting from these features to be informative, to support the subsequent steps of identification.

Feature extractionis based on substantially reducing the dimensionality of information. If the input data to a computational algorithm is too large and the processing time is remarkable and is assumed to be redundant information, it can be transformed into a reduced set of features (feature vector).

Constructing a subset of the initial features is called feature extraction. The selected features capture the main information from the input data so that the desired task can be performed using this reduced representation in exchange for the original data.

Analyzing a considerable amount of information usually requires a large amount of memory and computing power. Feature extraction is a term used for methods of constructing feature vectors to solve these problems while characterizing the data with sufficient accuracy.

1.5.1 Object detection

Due to the close relationship between object detection and video analysis and image understanding, it has attracted much research attention in recent years. Object detection is a computer vision technique for locating objects in images or videos. Object detection algorithms usually use machine learning or deep learning to produce meaningful results.

This applies distinct object classification and uses bounding boxes as shown below.

Object detection is fully interconnected with other similar computer vision techniques, such as image segmentation and recognition, which help us understand and analyze scenes from videos and images.

1.5.2 Haar characteristics

Face recognition means identifying the person, and face verification means verifying the person they claim to be (by comparing them to a certain database). Its applications are found in various fields such as schools, colleges, organizations, factories, public places, surveillance, etc. These techniques are gaining ground around the world and, extended with the recognition of human emotions, its applications are huge.

The key aspect in face recognition is the detection of relevant features of the human face, such as eyes, eyebrows, nose, lips.

In 2001, Paul Viola and Michael Jones supported the idea of using Haar wavelets and developed the so-called Haar features. A Haar feature considers adjacent rectangular regions at a given location within a detection window, sums the pixel intensities in each region, and calculates the difference between these sums. This difference is then used to classify the subsections of an image.

For example, in the case of a human face, it can be observed that, among all faces, the eye region is darker than the cheek region. So, a common Haar feature for face detection is a set of two adjacent rectangles that lie above the eye and cheek region. The position of these rectangles is defined relative to a detection window that acts as a bounding box for the target object (in this case, the face).

Image 1): the image that to be processed

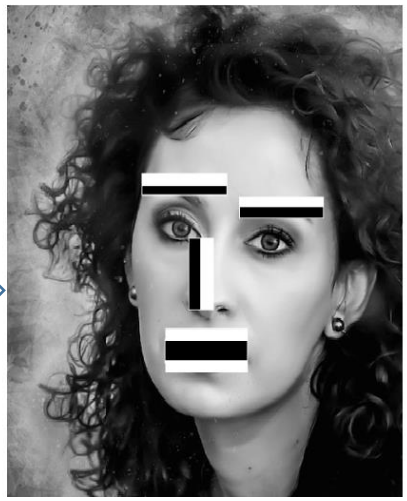


Image 2) Haar features applied to image 1)

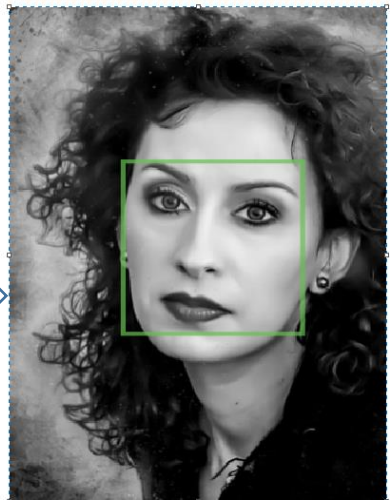


Image 3): face detected from image 1)

To detect the eyebrows, we use the Haar feature (image (1)), because the forehead and eyebrows form lighter pixels - darker pixels as in the image. Similarly, to detect lips, we will use a Haar-like feature (image (3)) with lighter - darker - brighter pixels. To detect the nose, we could use the similar darker-lighter Harr feature in (image (1)).

1.6 Morphological operations on binary images

Morphological operations on images affect the shape or structure of objects. They only apply to binary images (black and white images). These morphological operations are used as pre-processing or post-processing steps of images (filtering or removing protrusions) or to obtain a description of the shape of objects or regions (contours). The main morphological operations are dilation and erosion. Operations of this type can be tailored by selecting the element used, which determines how the objects will be dilated or eroded. Dilation enlarges objects and allows small gaps to be filled, while erosion shrinks objects by eroding the edges of objects.

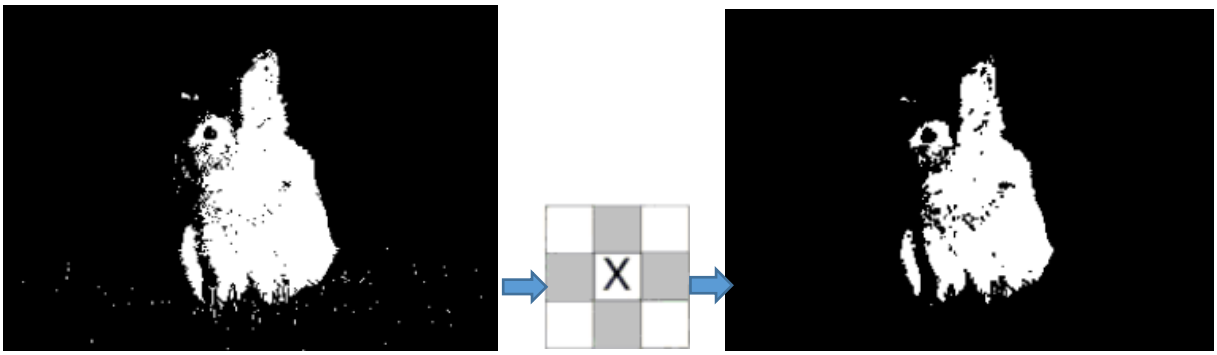
The "object" pixels are the set of pixels on which the morphological operations are applied, while the background pixels are the complement of the set of "object" pixels.

1.6.1 Erosion

The erosion operation is similar to dilation, but certain pixels will be turned into "background" pixels, the opposite of dilation. The image being binary, each pixel is replaced either by 0 for black or by 1 for white.

If the structural element placed over a given pixel touches the background, i.e. a point in the intersection is black, the current pixel is sent to the background. The structural element is moved over the image, so the following steps will be used:

1. If the origin of the structural element overlaps a "background" pixel in the image, then no change is made and it moves to the next pixel.
2. If the origin of the structural element overlaps an "object" pixel in the image, and there is at least one "object" pixel of the structural element that overlaps a "background" pixel in the image, then the current pixel in the image will be transformed to " background".



a. b. c.

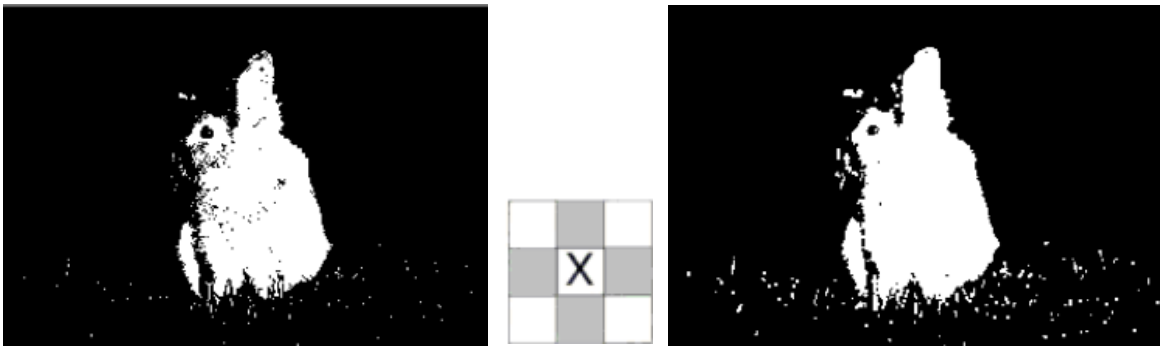
Example of erosion: a. Original image; b. Structural element; c. The resulting image

1.6.2 Expansion

Dilation is the complement of erosion, that is, it replaces the current pixel with the maximum value in the set of pixels intersected by the structural object. If the structural object touches the foreground of the image, then the current pixel becomes white. In this way, the foreground is larger than the original image. Dilation is defined by the following sequence of steps:

1. If the origin of the structural element overlaps with a "background" pixel in the image, no change is made and it moves on to the next pixel.

2. If the origin of the structural element coincides with an "object" pixel in the image, then all pixels covered by the structural element become "object" pixels.



a. b. c.

Example of erosion: a. Original image; b. Structural element; c. The resulting image

1.6.3 Opening (O-opening) and closing (C-closing)

With the help of dilation and erosion, operations of a higher order can be built: opening (Oopening) and closing (C-closing). These techniques can also be used to find specific shapes.

O-opening removes small objects from the foreground of an image (eg bright pixels), placing them in the background. Opening is a process in which first an erosion operation and then a dilation operation are performed. In general, this process smooths the outline of an object better and removes thin protrusions.

C-closing removes small holes in the foreground, in other words it removes the background pixels that match the structuring element. This tends to smooth out sections of contours, but compared to opening, it merges narrow breaks and long thin gaps, removes small holes and fills contour gaps. Closure is a process where first a dilation operation is performed, then an erosion operation.

2. Chapter II: Development environment

To develop the practical part of this bachelor thesis, the development environment used is MATLAB®. Through it, the application was developed in the programming language MATLAB.

MATLAB®, short for "Matrix Laboratory," is a programming platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world. The core of MATLAB is the MATLAB language, a matrix-based language that enables the most natural expression of computational mathematics.

This program includes a code editor with support for intelligent code completion.

What we can do with the MATLAB application:

- Data analysis
- Algorithm development Creating
- models and applications

MATLAB enables ideas to be taken from research to production through implementation in enterprise applications and embedded devices, as well as integration with Simulink® and Model-Based Design.

MATLAB has evolved over a period of years with input from many users. In university settings, this is the standard instructional tool for courses

introductory and advanced math, engineering and science. In industry, MATLAB is the tool of choice for high-throughput research, development, and analysis.

2.1 The MATLAB programming language

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Mathematics and calculation
- Algorithm development Modeling,
- simulation and prototyping
- Analysis, exploration and visualization of data scientific and engineering graphics
- Application development, including creation of graphical user interfaces

MATLAB is an interactive system whose basic data element is a dimensionless array. This allows you to solve many technical computational problems, especially those with matrix and vector formulations, in a fraction of the time it takes to write a program in a non-interactive scalar language such as C or Fortran.

The name MATLAB means matrix laboratory. MATLAB was originally written to provide easy access to the matrix software developed by the LINPACK and EISPACK projects, which together represent the current state of matrix computing software.

MATLAB has a family of application-specific solutions called toolboxes. Very important to most MATLAB users, toolboxes allow you to learn and apply specialized technologies. Toolboxes are complete collections of MATLAB functions (M files) that extend the MATLAB environment to solve specific classes of problems. Areas where toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and more.

2.2 Programming languages similar to MATLAB

Matlab is a programming language that supports multiple programming paradigms. This is a numerical computing environment developed by MathWorks. It features a high-performance, general-purpose language, but this language is tightly integrated with Java - the Matlab interpreter is written in Java.

Simulations in C++ show superior performance, 100 times better in terms of time complexity than an equivalent MATLAB implementation. Most of the time, code in C++ is mostly serial, and no high-fidelity optimization is done explicitly.

C++ has an average processing speed of over 500 times faster than Matlab code. This is not only true for this code, but can also be applied to any other code comparison between Matlab and C++ MEX files. By comparison, the speed advantages of C++ far outweigh the simplicity of MATLAB.

Python is a high level language, it is easier to use, easier to read and more portable. MATLAB has the array function and Python can use NumPy and the library can achieve similar results. MATLAB is faster than Python, but Python is better at running multiple tasks in parallel.

MATLAB may run slowly because it has a limited amount of RAM (below 128 MB). The RAM used by MATLAB during execution is between 40 MB-60 MB. The HELP browser can take up another 12MB. If memory is limited (RAM), the processor may start using virtual memory (from the hard disk).

3. Chapter III: Practical implementation

The practical part of this work consists of an application developed in the MATLAB programming language, in the development environment provided by the application of the same name, i.e. MATLAB. The application offers several selection menus through a text interface that facilitates navigation among the provided options and to streamline the user experience, functions running under different threads for various resource-consuming functions.

The application directly uses functions from the MATLAB library to facilitate accessing, transferring and editing pictures, and to keep the user always in touch with the application, he receives real-time feedback on the changes made, whether under message form when an error has occurred, whether in the form of a window displaying the resulting image.

3.1 Application Description

The image processing application offers iris detection functions along with various effects and manipulations of a user-supplied image.

All these functions are implemented in threads, so that the user interaction with the application is lag-free, and the navigation menus are simple and easy to use.

To detect the iris we will use an image stored on the device and apply different filters to transform or remove parts of it such as: framing the eye area and cropping it, transforming the original image into a grayscale image, binarizing the image. All these processes will be applied to the original image until the final result is reached, where the color image will have the iris framed and a suggestive message.

3.2 Application Structure/Organizational Chart

The application has the following functional structure:

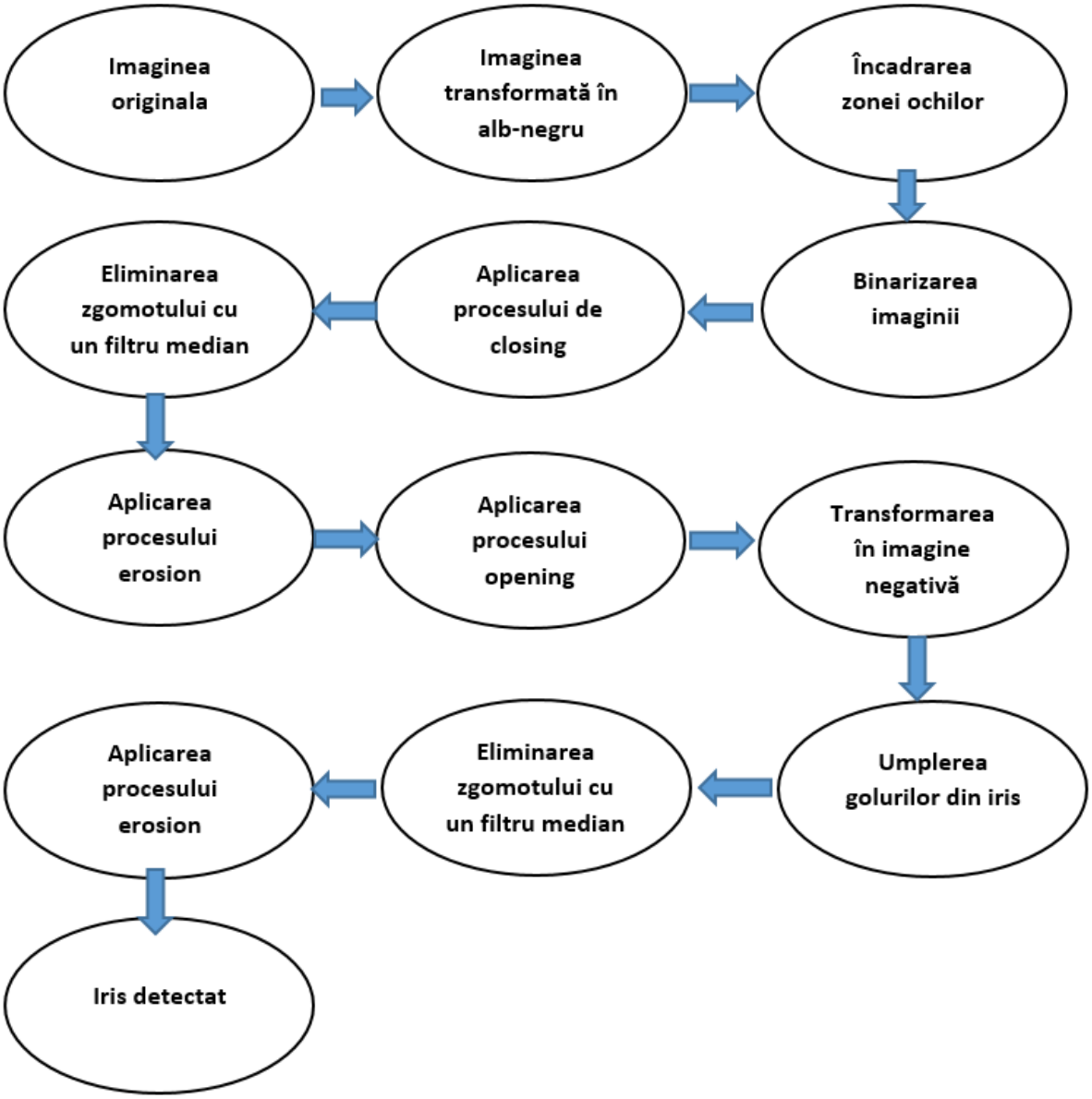


Figure 3.01: Block diagram of the MATLAB application

3.3 Implementation of the program

During the iris detection stage, we will perform the following steps:

Reading the image from the device

We will read the image stored on the device using the imread() function.

```
clc
clear all
close all

% EyeDetect este o variabila care va avea valoarea dată de return-ul
% funcției apelate cu parametrul 'EyePairBig'
EyeDetect=vision.CascadeObjectDetector('EyePairBig');
%citirea imaginii stocate pe dispozitiv
I=imread('D:\licenta\eye3.jpg');
```

The imread() function reads the image from the local file specified by the given path, inferring the file format from its contents. If filename is a multi-image file, then imread reads the first image in the file.

Convert image to grayscale (black and white)

The original image on the device is converted to black and white using the rgb2gray() function, and the eye area is framed. The rgb2gray function converts RGB images to grayscale by removing hue and saturation information while preserving luminance.

```
%In variabila igray se stocheaza imaginea data de utilizator in format
%alb-negru
igray=rgb2gray(I);
%In variabila BB se stocheaza zona ochilor
BB=step(EyeDetect,igray);
```

Image cropping

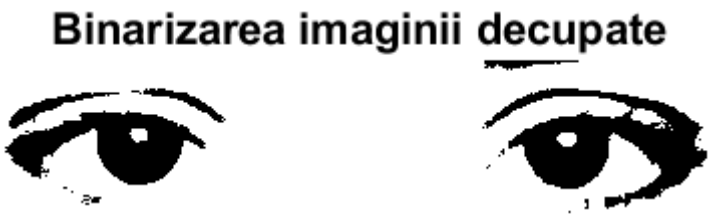
After the eye area has been detected and framed in a window, it is cropped using the imcrop() function. This function displays the image indexed in the specified parameters and creates a crop tool associated with that image.

```
%In variabila Eyes se stocheaza imaginea decupata
Eyes=imcrop(igray,BB);
```

Image binarization

The cropped image goes through the binarization process with the binarize() function. This function creates a binary image from the 2-D or 3-DI grayscale image by replacing all values that exceed a globally determined threshold with values of one second and by setting all other values to 0.

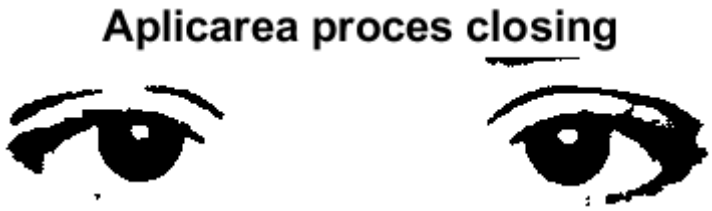
```
% Binarizarea imaginii
imshow(Eyes);
title("Ochii");
BW = imbinarize(Eyes,0.4);
figure
imshow(BW);
title("Binarizarea imaginii (alb-negru)");
imwrite(BW, "BW.png");
```



Application of the closing process

On the binarized image, the closing process is applied with the function imclose(). This performs the morphological closing of the grayscale or binary BW image, using a disk-type structuring element of size 2. The morphological closing operation (closing) is a dilation followed of an erosion, using the same structuring element for both operations.

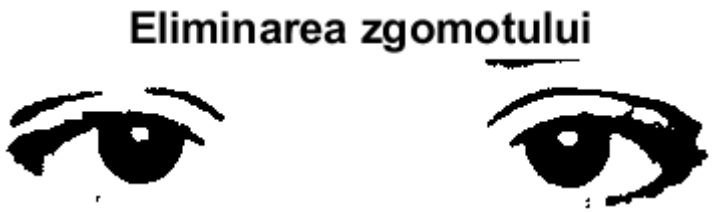
```
% Functia strel cu parametru disk va crea o figura in forma de disk cu raza
% egala cu doi
se = strel('disk',2);
closeBW = imclose(BW,se);
figure
imshow(closeBW);
title("Aplicare proces de closing");
imwrite(closeBW,"closeBW.png");
```



Noise elimination

A median filter is applied over the image to reduce noise with the medfilt2() function. It does median filtering of the closeBW image in two dimensions. Each output pixel contains the median value in a 3x3 neighborhood around the corresponding pixel in the input image.

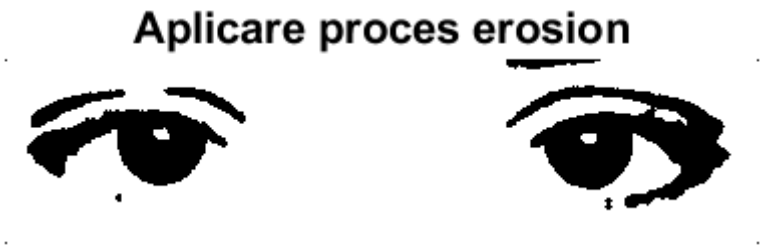
```
% Se aplica un filtru median pentru imagine pentru a reduce zgomotul
filter1 = medfilt2(closeBW);
figure
imshow(filter1);
title("Eliminarea zgomotului");
imwrite(filter1,"filter1.png");
```



Application of the erosion process

After the noise has been removed, the erosion process is applied over the resulting image with the imerode() function. The function erodes the filter1 image in grayscale or binary, using the SE structuring element.

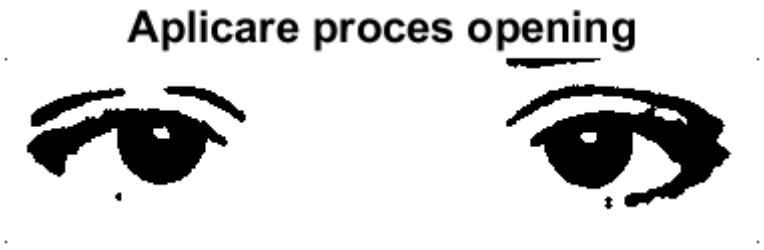
```
% creează un element structurant în formă de disc, unde 2 specifică raza
se3 = strel('disk',2);
%functia imerode erodeaza imaginea stocata dupa filtrarea acesteia
%elimina pixelii de la limitele obiectului
er1=imerode(filter1,se3);
figure
imshow(er1);
title("Aplicare proces erosion");
imwrite(er1,"er1.png");
```



Application of the opening process

The bwareaopen() function removes all connected components (objects) that are less than 30 pixels from the binary image er1, producing another binary image. This operation is known as area opening ("area opening").

```
%functia bwareaopen elimina parti din imagine care au mai putini de 30 de
%pixeli
open = bwareaopen(er1,30);
figure
imshow(open);
title("Aplicare proces opening");
imwrite(open,"Aplicare proces de opening.png");
```



The transformation into a negative image

The image becomes a negative image using the imcomplement() function. For a grayscale image, dark areas become lighter and light areas become darker.

```
%functia imcomplement transforma in imagine negativa variabila open
neg = imcomplement(open);
figure
imshow(neg);
title("Imagine negativa");
imwrite(neg,"Imagine_negativa.png");
```

Filling the gaps

To fully form the iris, gaps projected due to light are removed with the imfill() function.

```
%umple golurile proiectate din cauza luminii din iris
fill = imfill(neg,'holes');
figure
imshow(fill);
title("Umplerea golurilor");
imwrite(fill,"Umplerea golurilor.png");
```

Noise elimination

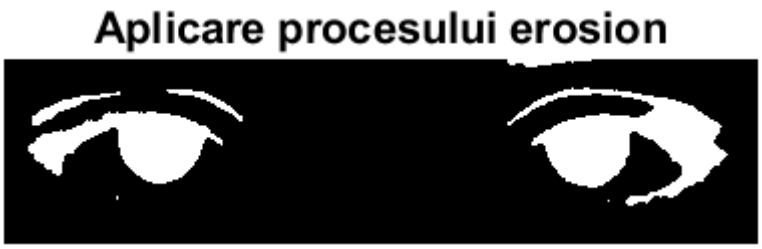
Denoise the negative image again with the medfilt2() function.

```
%Se aplica un filtru median pentru imagina stocata in variabila fill ce ia
%ca valoarea imaginea cu culorile inversate din variabila neg
filter2 = medfilt2(fill);
figure
imshow(filter2);
title("Eliminarea zgomotului");
imwrite(filter2,"Eliminarea zgomotului.png");
```

Application of the erosion process

After the noise has been removed, the erosion process with the imerode() function is applied over the resulting image.

```
%Procesul erosion
se4 = strel('disk',4);
er2=imerode(filter2,se4);
figure
imshow(er2);
title("Proces de eroziune");
imwrite(er2,"Erosion.png");
```



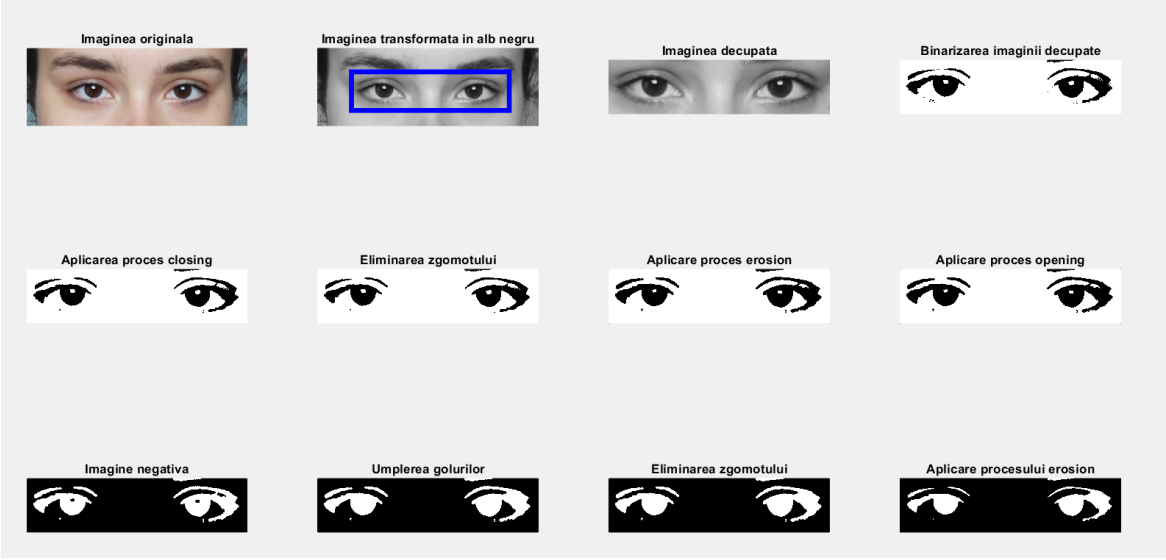
Creating circles to frame the iris

The imfindcircles function finds circles with radii in the range specified by er2. The additional output argument, radii, contains the estimated radii corresponding to each circle center in centers.Finally, the viscircles function draws circles with the specified centers and radii on the current axes.

```
%Cu functia imfindcircle se cauta in imagine irisul care a fost incadrata
%mai sus
[centers,radii] = imfindcircles(er2,[30 120],'ObjectPolarity','bright', ...
    'Sensitivity',0.92)

%Detectarea pupilelor
figure,imshow(Eyes1)
title('Iris detectat')
%functia viscrircles creeaza un cerc cu centru si raza stocate mai sus cu
%functia imfindcircles
h = viscircles(centers,radii);
```

In conclusion, the iris is detected.



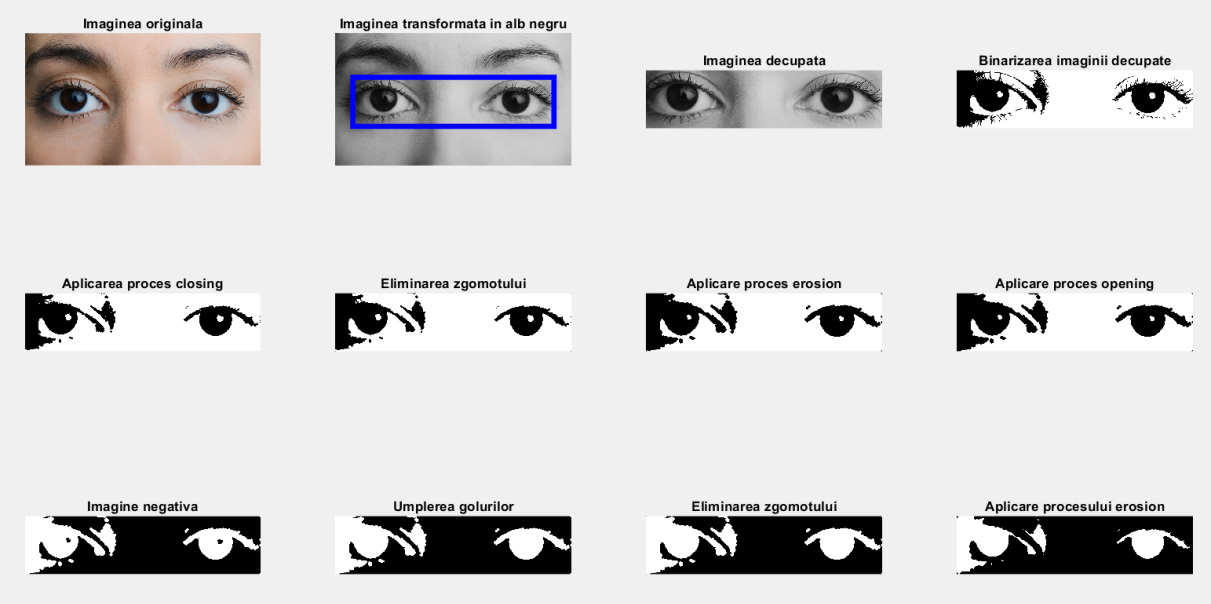
3.4 Simulation results

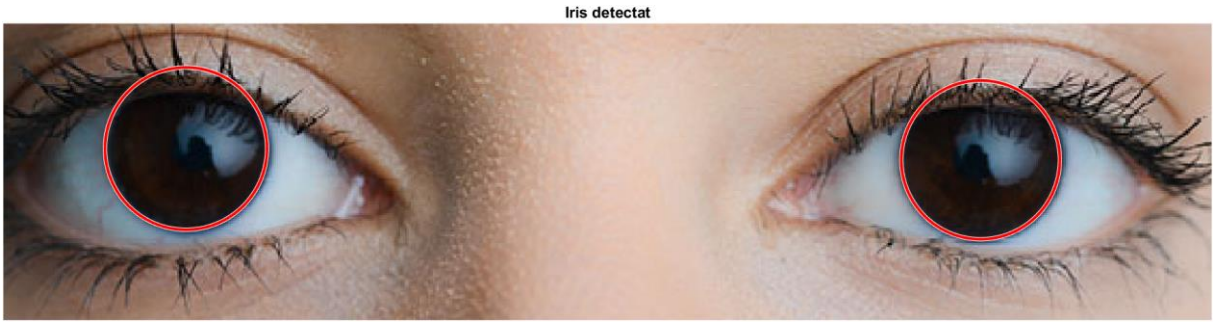
To observe the results obtained from the simulation, I chose three different images and exemplified each chosen frame with one image to see what the result is when the eyes are in different positions or have different shapes.

In the case of the iris detection algorithm, the detection rate is extremely high as long as the subjects' eyes are facing the camera and not covered by anything.

First case:

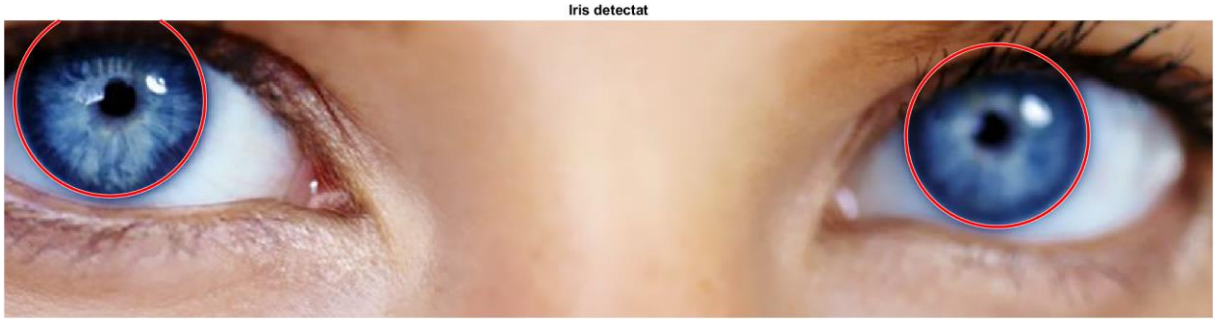
In the case of large and round eyes, it is expected that the algorithm will easily detect the iris. The image is clear, the contour of the eyes is fully visible and the brightness is suitable for all stages of the program.

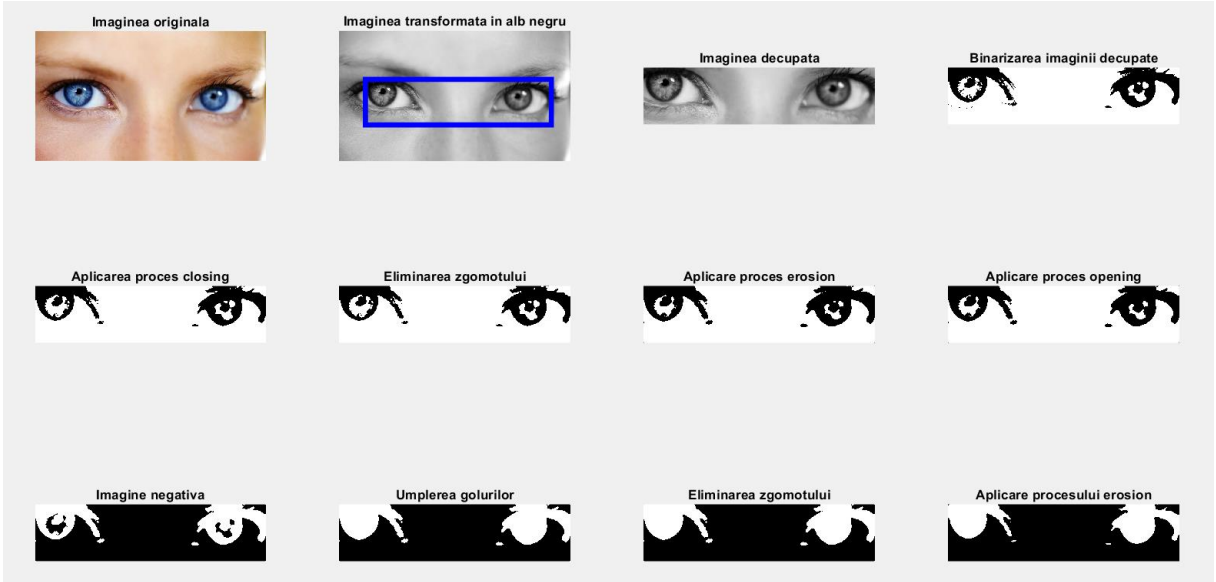




The second case:

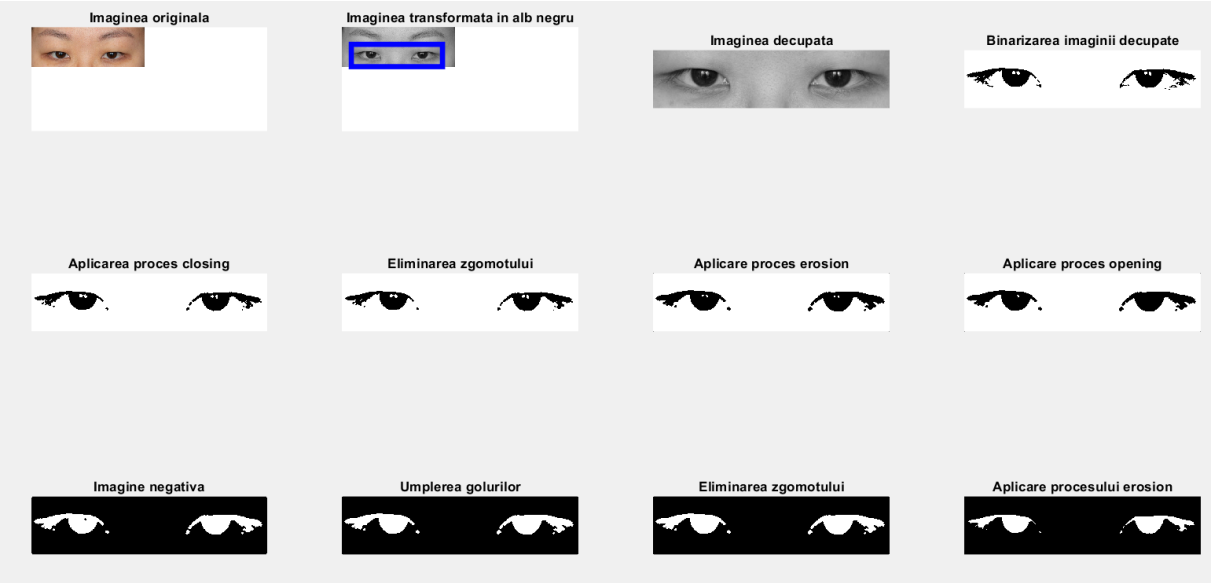
I chose this image because the iris is much lighter compared to the first case, but also the position of the eyes is slightly different which would mean that it could make detection difficult.





The third case:

In the case of elongated eyes and drooping eyelids, it is much more difficult to achieve iris detection, but the application manages to complete each step.



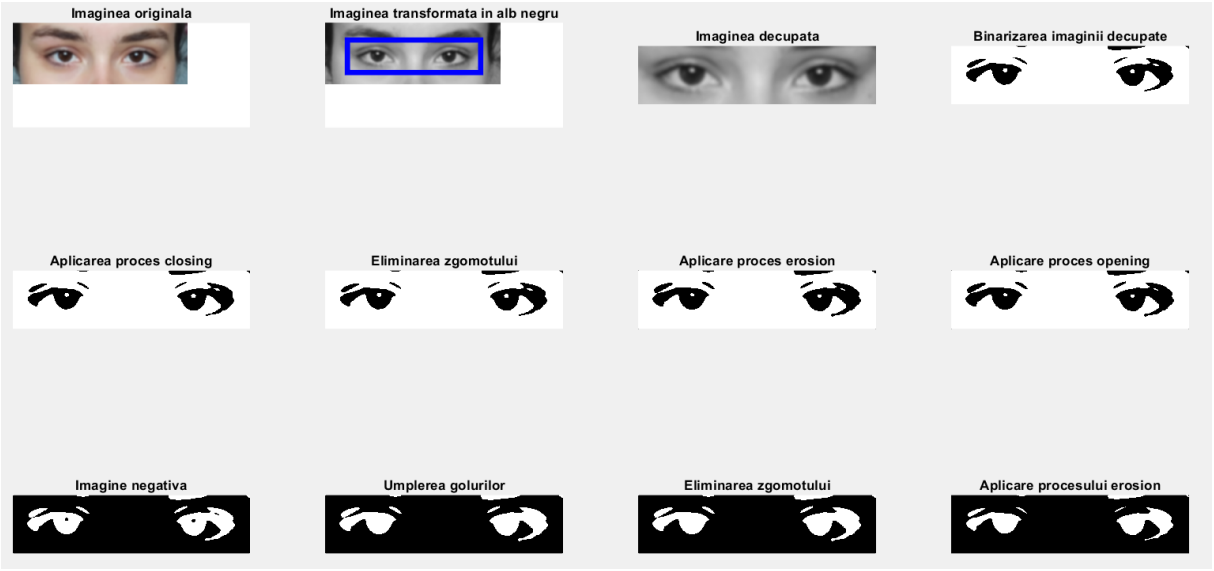
3.5 Analysis of the algorithm. Limitations

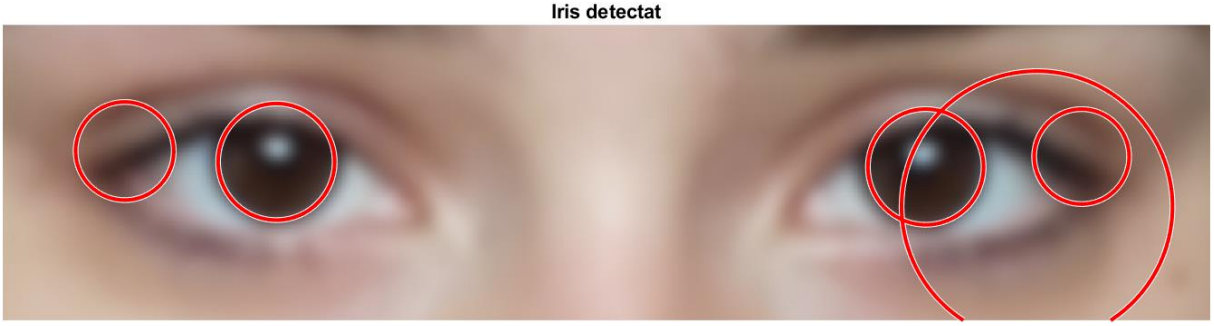
An iris detection algorithm has some limitations that could make it difficult to properly detect the corners of an object. The most important problems in the iris are: the influence of resolution, the influence of noise, blurring, brightness and other detected objects. In the following paragraphs we will exemplify each of them in order to be able to observe the effects.

3.5.1 Blurring

The iris detection algorithm has results dependent on the sharpness of the pictures and their level of detail. To be able to recognize shapes with the highest possible accuracy, the subject must be in focus and its outline must not be covered, as this leads to misclassification.

If, in some cases, the outline of the iris is not completely visible, the algorithm is more likely to detect extra circles in the entire document, as can be seen in the image below:



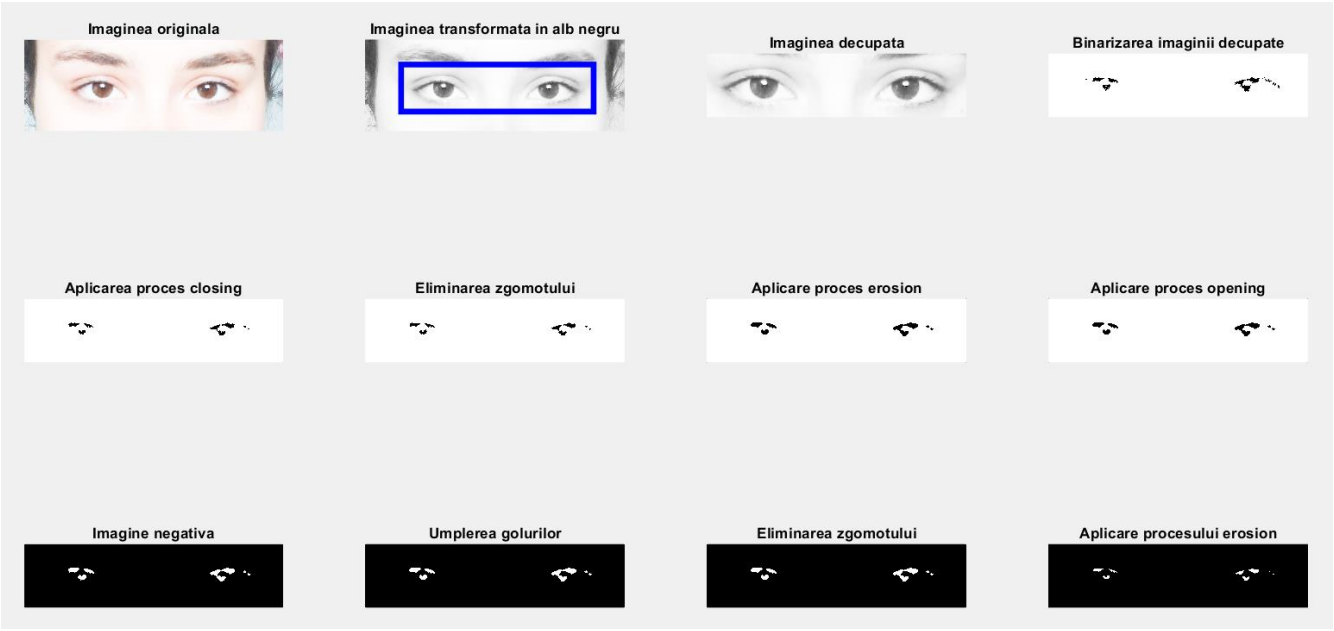


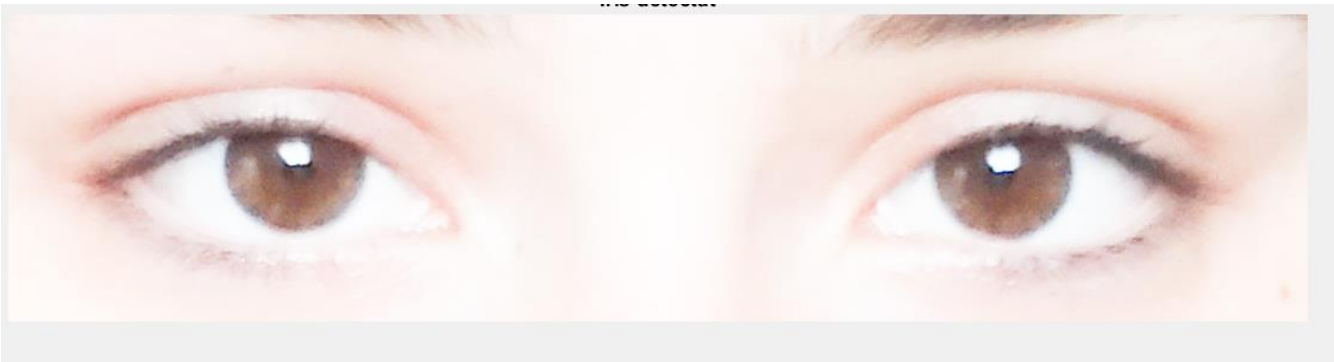
3.5.2 Increased brightness

In our application brightness has an important role and we will demonstrate it in the following example.

We will add a white level of 0.9 to our frames and observe how it influences the framing of the iris in the circle. As you can see, the program executes each transformation of the image, but in the end the framing of the iris in the circles is not achieved.

The eye area is framed, then it is cropped, but in the binarization process the image undergoes great changes. Its purpose is to segment the image into regions of interest and remove regions that are not considered essential. In our case, the wall being white and the intensity of the pixels being increased, the algorithm cannot fully detect the outline of the eyes.

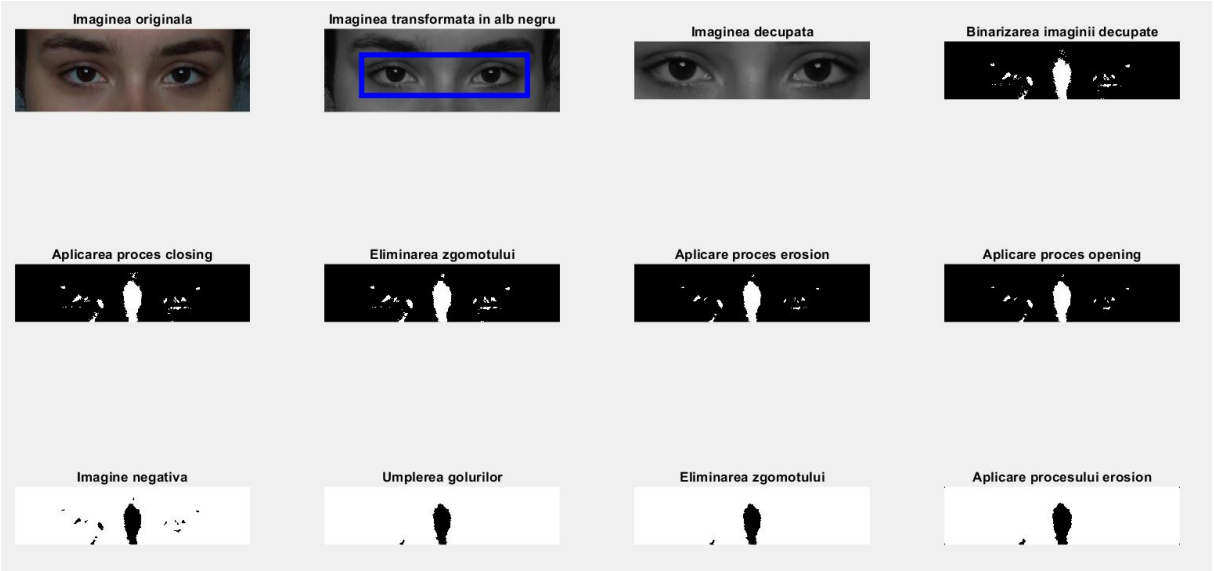


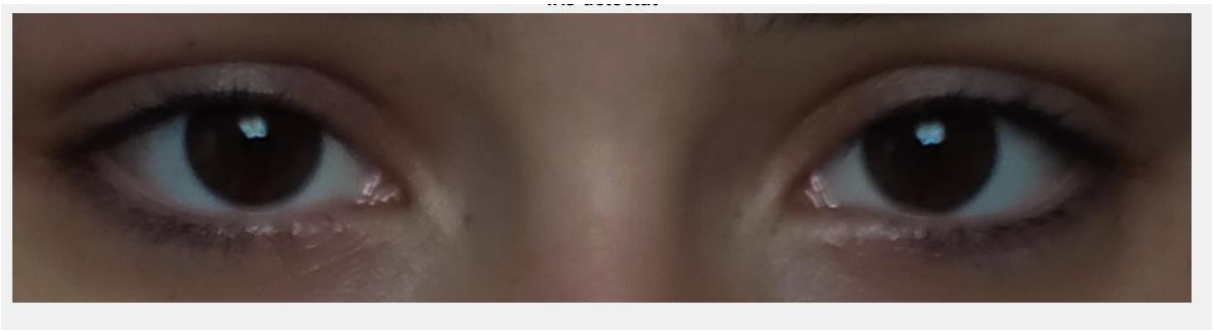


So, brightness has an important role decisively influencing iris detection when a value close to the maximum white level is applied to the image.

3.5.3 Contrast

Increasing contrast increases the difference between light and dark areas of the image. After the binarization process, the separation of regions in the image could not be clearly achieved due to the binarization threshold, therefore where the light was stronger the pixels became white, and the region where the light was low the pixels became black. When the image became negative, the eye region could no longer be distinguished, affecting the iris detection process.



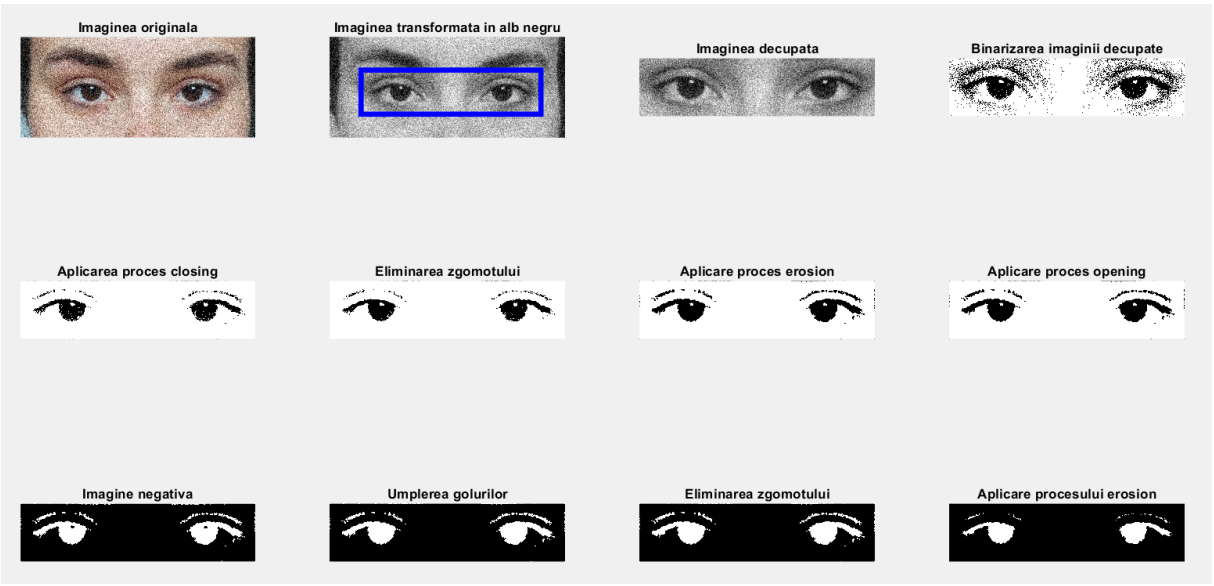


3.5.4 Influence of noise

Noise caused by lost or altered bits can make it difficult to detect the iris, and implicitly frame it in a circle.

In the salt & pepper noise model, there are only two possible values, a and b. Because of this, it is also called speckle noise. The probability of occurrence of each is less than 0.1. At values higher than these, the noise will dominate the image.

This type of noise will affect pixel values in two ways: "jump" meaning the new pixel value is 255 (the pixel is white) or "pepper" where the new pixel value is 0 (the pixel is black).



4. Chapter IV: Conclusions

Digital image processing is a vast field, but one that is developing at a fast pace. Image processing is often used in many fields and makes our lives easier or even protects us from certain dangers such as possible accidents. The main use of this application is to optimize the information contained in images for human analysis.

In this work, the objective was to successfully achieve iris detection, through different processes such as: image binarization, noise removal, image erosion or opening/closing processes. This application has the advantage that regardless of the shape of the eyes, the detection is not influenced , being able to easily detect the iris.

Like any other detection application, it also has some disadvantages, and in this paper we highlighted some negative influences on the correct and precise detection of the iris, implicitly framing with the help of circles. The main negative influences are: increased brightness, increased blurring of the image or noise picked up.

These objectives were generally met, with the program flowing smoothly and quickly. At the application level there is a waiting time, caused by the ability to update the image in real time, suffering the passage through the filtering or transformation processes.

In a much more complex form, the application aims to make the work of those working in the medical system or banking easier, replacing cumbersome and time-consuming systems based on PIN codes and passwords. The work is in prototype form, with future improvements to be made to its operation. For example, capturing images will be done automatically, in real time, by using the web camera. A more refined continuity of video sequence segmentation and their processing or filtering is also pursued. The fastest possible update of processes, images and a more user-friendly interface will be pursued.

Designed to work as a prototype, the device proposes several functions to be successfully performed, adding new ones and updating existing ones.

5. Appendix

```
cl

clearall

closedall


% EyeDetect is a variable that will have the value given by the return

% function called with parameter 'EyePairBig'

EyeDetect=vision.CascadeObjectDetector('EyePairBig');

%reading the image stored on the device

I=imread('D:\licenta\eye3.jpg');


%The image given by the user in the format is stored in the igray variable

% black and white

igray=rgb2gray(I);

%The eye area is stored in the BB variable

BB=step(EyeDetect,igray);


%The cropped image is stored in the Eyes variable

Eyes=imcrop(igray,BB);


% Image binarization

imshow(Eyes);

title("The eyes");

BW = binarize(Eyes,0.4);

figures

imshow(BW);

title("Image binarization (black and white)");
```

```
imwrite(BW,"BW.png");


% Framing the eye area
% The strel function with the disk parameter will create a disk-shaped figure with a radius
% equals two
se = strel('disc',2);
closeBW = imclose(BW,se);
figures
imshow(closeBW);
title("Application of closing process");
imwrite(closeBW,"closeBW.png");



% A median filter is applied to the image to reduce noise
filter1 = medfilt2(closeBW);
figures
imshow(filter1);
title("Noise Removal");
imwrite(filter1,"filter1.png");



% creates a disc-shaped structuring element, where 2 specifies the radius
se3 = strel('disc',2);
%imerode function erodes the stored image after filtering it
%remove pixels from object boundaries
er1=imerode(filter1,se3);
figures
```

```
imshow(er1);

title("Erosion process application");

imwrite(er1,"er1.png");


%the bwareaopen function removes parts of the image that are less than 30 de

%pixels

open = bwareopen(er1,30);

figures

imshow(open);

title("Opening process application");

imwrite(open,"Opening process application.png");

The function %imcomplement transforms into a negative image in the open variable

neg = incomplete(open);

figures

imshow(neg);

title("Negative Image");

imwrite(neg,"Negative_image.png");


%fills the gaps projected due to the light from the iris

fill = imfill(neg,'holes');

figures

imshow(fill);

title("Filling in the Gaps");

imwrite(fill,"Filling the gaps.png");

%A median filter is applied to the image stored in the fill ce ia variable

% as the image value with inverted colors from the neg variable

filter2 = medfilt2(fill);
```

```
figures

imshow(filter2);

title("Noise Removal");

imwrite(filter2,"Noise Removal.png");

%

se4 = strel('disc',4);

er2=imerode(filter2,se4);

figures

imshow(er2);

title("Noise Removal");

imwrite(er2,"Noise Removal.png");



subplot(3,4,1),imshow(I)

title('Original Image')

subplot(3,4,2),imshow(igray)

title('Image converted to black and white')

rectangle ('Position'BB'LineWidth',4,'Line Style','-','EdgeColor','b');

subplot(3,4,3),imshow(Eyes)

title('cropped image')

subplot(3,4,4),imshow(BW)

title('Crop image binarization')

subplot(3,4,5),imshow(closeBW)

title("Application of closing process")

subplot(3,4,6),imshow(filter1)

title('Noise Removal')

subplot(3,4,7),imshow(er1)
```

```
title('Erosion process application')
subplot(3,4,8),imshow(open)
title('Opening process application')
subplot(3,4,9),imshow(neg)
title('Negative Image')
subplot(3,4,10),imshow(fill)
title('Filling in the gaps')
subplot(3,4,11),imshow(filter2)
title('Noise Removal')
subplot(3,4,12),imshow(er2)
title('Application of the erosion process')

% In BB1, the eyes will be stored from the initial image
BB1=step(EyeDetect,I);
Eyes1=imcrop(I,BB1);
[rows, columns, numberOfColorChannels] = size(Eyes1);
%Eye framing
righteye = Eyes1(1:end, 1:round(columns/2), :);
lefteye = Eyes1(1:end, round(columns/2):end, :);

%With the imfindcircle function, the pupil that was framed is searched for in the image
%above
[centers,radii] = imfindcircles(er2,[30 120],'ObjectPolarity','bright',...
    'Sensitivity',0.92)

%Pupil detection
figure,imshow(Eyes1)
```

```
title('Iris detected')

%the viscrircles function creates a circle with center and radius stored above with

%imfindcircles function

h = viscircles(centers, radii);
```

6. References

Figures:

[1] https://cnx.org/contents/ WeBHosox@1.1 : ZSSjvvgF@2 /Iris-Recognition-Results-and-Conclusions

[2] https://www.nationalgeographic.com/photography/article/milestones-photography

[3] https://bionescu.aimultimedialab.ro/index_files/tapai/TAPAI_BIonescu_M4.pdf

[4] https://www.lcipaper.com/kb/what-are-the-differences-between-pantone-cmyk-rgb.html

[5] https://www.researchgate.net/figure/An-example-comparison-of-the-low-resolution-inputof-the-625-of-k-space-with-the_fig4_349044795

[6] http://easy-learning.neuro.pub.ro:8888/Laboratoare/L4- Echipamente%20de%20scanare/Echipamente%20de%20scanare/Chap5/Histograma/Histogram a.htm

[7] http://users.utcluj.ro/~rdanescu/pi_c03.pdf

[8]https://www.mathworks.com/matlabcentral/fileexchange/26978-hough-transform-forcircles

[9] https://koaha.org/wiki/Trasformata_di_Radon

[10] https://www.rasfoiesc.com/educatie/informatica/Metode-de-binarizare-a-imagini32.php

- https://cnx.org/contents/ WeBHosox@1.1 : ZSSjvvgF@2 /Iris-Recognition-Results-and-Conclusions
- https://medium.com/analytics-vidhya/what-is-haar-features-used-in-face-detectiona7e531c8332b
- https://www.mathworks.com/help/images/ref/imcomplement.html
- https://www.geeksforgeeks.org/matlab-erosion-of-an-image/
- https://www.mathworks.com/matlabcentral/answers/334096-why-does-imfindcirclesnot-find-circles-in-my-image

- https://cimss.ssec.wisc.edu/wxwise/class/aos340/spr00/whatismatlab.htm
- https://cmmi.tuiasi.ro/wpcontent/uploads/cursuri/ Limbaje%20de%20programare%20structurata.%20Aplicatii% 20MATLAB.pdf

- https://en.wikipedia.org/wiki/Thresholding_(image_processing)
- https://koaha.org/wiki/Trasformata_di_Radon
- http://www.master-taid.ro/Cursuri/IVOM_curs/ivom_trasataturi%20faciale_2019.pdf https://
- www.mathworks.com/help/images/ref/bwareaopen.html?s_tid=srchtitle_bware aopen_1

- https://anale-informatica.tibiscus.ro/download/lucrari/1-2-18-Mark.pdf
- https://www.miv.ro/books/MIvanovici_PI.pdf
- https://dokumen.tips/documents/operatii-morfologice.html?page=3
- http://alpha.imag.pub.ro/ro/cursuri/archive/color.pdf https://
- www.rasfoiesc.com/educatie/informatica/Metode-de-binarizare-aimagini32.php
- https://webspace.ulbsibiu.ro/catalina.neghina/Resurse/PI/pdf_PI/L1.pdf https://
- www.mathworks.com/matlabcentral/answers/334096-why-does-imfindcirclesnot-find-circles-in -my-image