Uludağ University Journal of The Faculty of Engineering, Vol. 25, No. 1, 2020 RESEARCH

DOI: 10.17482/uumfd.660759

AN IMPLEMENTATION THAT DIAGNOSIS ANISOCORIA DISEASE USING IMAGE PROCESSING TECHNIQUES

Nerma Ćorović *

Emel ARSLAN *

Received: 18.12.2019; revised: 17.02.2020; accepted: 02.03.2020

Abstract: Human pupil abnormalities can be an indicator of many diseases. Anisocoria is a common condition that has been estimated at 20% of the average population. It is specified by inequality in the size of the pupils of the eyes. This paper proposes an anisocoria determining algorithm from a digital image by using the MATLAB computing environment that involves the usage of MATLAB Computer Vision and Image Processing Toolbox. The image used in this work as input data is an image that has been fetched from Siblings Image Database. An input image where anisocoria is present has been downloaded from the Internet. The paper gives an idea of understanding how pupil detection and measurement can be used in medical and psychology diagnostics by using a simple algorithm.

Keywords: Image processing, Anisocoria, Diagnosis, Pupil Detection, Pupil Measurement

Görüntü İşleme Teknikleri Kullanarak Anizokori Hastalığını Teşhis Eden Uygulama

Özet: Göz bebeğindeki anormallikler bir çok hastalığın belirtisi olabilir. Bu hastalıklardan biri olan Anizokori, iki göz bebeği büyüklüğünde bir eşitsizlik olma durumu ile karakterize edilmektedir. Anizokori normal popülasyonun %20'sinde görülen yaygın bir durumdur. Bu makalede, dijital görüntü üzerinde MATLAB hesaplama ortamı kullanılarak uygulanan görüntü işleme teknikleri ile her iki göz bebeğinin tespiti ve büyüklük karşılaştırması gerçekleştirilmekte ve elde edilen sonuçlar ışığında anizokor durumu olup olmadığının belirlenmesi için bir algoritma önerilmektedir. Söz konusu çalışmada giriş verisi ve test amaçlı kullanılmak üzere görüntüler Siblings Image Database ve internet ortamından temin edilmiştir. Bu makale, görüntü işleme algoritması kullanılarak, göz bebeğinin tespiti ve ölçümünün tıp ve psikoloji tanılarında nasıl kullanılabileceğini anlamaya yönelik fikir ortaya koymuş bir çalışmadır.

Anahtar Kelimeler: Görüntü İşleme, Anizokori, Teşhis, Gözbebeği Tespiti, Göz Bebeği Ölçümü

1. INTRODUCTION

As the old proverb says the face is the mirror of the soul, and the eyes are the gateway to it. Eyes can be a harbinger of body health. Many diseases can be diagnosed with symptoms that can appear in one eye or both eyes. Human pupil abnormalities can be an indicator of diabetes, multiple sclerosis, schizophrenia, autism, Alzheimer and Parkinson diseases, depression, alcohol and drug addiction (Iacoviello, 2006).

The pupil is a black part of the eye that should lie in the centre of the iris (coloured part). The standard shape of the pupil is circular, and small muscles of the iris regulate its size, the sphincter pupillae responsible for constriction and dilator pupillae as its name says for dilation. The pupil gets wider in dim light (5-8 mm), narrower in the bright light (5-8 mm) but under the ordinary conditions of lighting it should appear neither too large nor too small.

* Department of Computer Engineering, Istanbul University-Cerrahpasa, 34320, Istanbul, Turkey
Corresponding Author: Emel ARSLAN (earslan@istanbul.edu.tr)

The size of the pupil depends on a persons' age; larger pupils are found in small children. Usually, the size of the pupil is the same in each eye, with both eyes dilating or constricting together.

Anisocoria is a condition estimated at 20% of the average population. It is specified by inequality in the size of the pupils of the eyes (See Fig.1 b and c for example), vice versa is isochoric, pupils have the same size.

In the case of anisocoria, size dissimilarity (between left and right eye pupils) is in the range of 0.5 mm to 1mm (Friedman & Kaiser, 2009). When this size divergence is more significant than 1mm, it may be an indication of different diseases of eye, brain and nervous system; also, it may refer to various pathological conditions such as Horner syndrome.

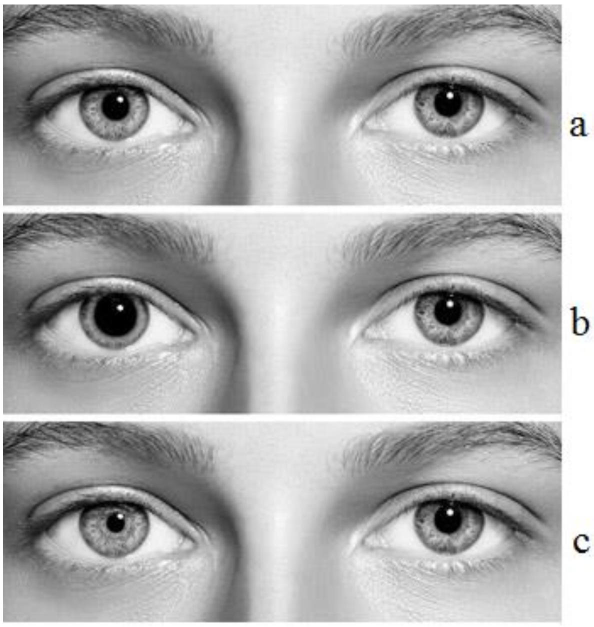


Figure 1:
a) *Normal size pupils;* ***b)*** *Anisocoria (size of one pupil is more significant than average size pupil);* ***c)*** *Anisocoria (size of one pupil is less than average size pupil)*

This paper tends to suggest an algorithm for human pupil detection from a face shown in a digital image and detect the possible presence of anisocoria by calculating the size difference between the left and right pupil. The proposed algorithm can serve as part of another medical diagnostic program for disease detection. The simplicity and reliability of the program can enable not only ophthalmologists but also non-professionals and laypeople to control their health using telemedicine.

2. PROPOSED METHOD

2.1. Face and Eye Region Detection

The detection of the face and eye in an image is a well-known research subject often studied in the field of image processing, computer vision, and pattern analysis and classification.

The algorithm that is based on Viola-Jones approach (Viola & Jones, 2004) has been used to detect face and eye region from an input image, which has previously been converted to a grayscale image.

The main reason why grayscale images are used to extract identifiers instead of colour images is that grayscale simplifies the algorithm and reduces computational requirements. Indeed, colour can provide a limited benefit in many applications, and presenting unnecessary information can increase the amount of training data needed to achieve excellent performance. The "rgb2gray" command is used to convert the RGB image to a grayscale image.

A realisation of Viola-Jones algorithm is created with the help of Computer Vision Toolbox build-in classes and functions from the Matlab computing environment.

The Matlab system type object vision.CascadeObjectDetector is used for creating the detector that detects the face and eye-pair region from an input image. By default, this detector is set to detect faces in an image, but it can also detect other parts of the face. The decision if the part of the image in the region of face or eye-pair or not is made by use of Haar-like features (rectangular areas) that are used for the core origin for Haar classifier. After the detector was created, the bounding box is defined by the step method bbox=step(detector, image).

Fig. 2 shows the face and eye-pair detection method consists of the following steps:

i. input an image (Fig. 2-a);
ii. convert the input image to grayscale image and create the detector object for the face (Fig. 2-b),
iii. determine the face area of an image (Fig. 2-c),
iv. identify eye-pair regions from the image (Fig. 2-ç),
v. divide the eye-pair region between the left and right eye regions (Fig. 2-d),
vi. show result images with the annotated face and eye region (eye-pair, left and right eye separately).

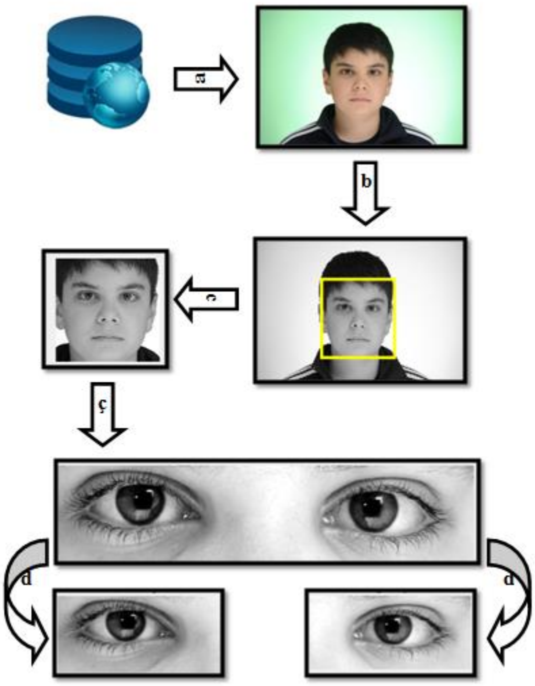


Figure 2:
The face and eye regions detection

Some factors such as

- face rotation (head pose not strictly frontal),
- face expression (an expression not always neutral and often with a smile),

- face feature (unusual facial expressions),
- occlusion (facial hair, glasses with dark lenses, usage of makeup),
- media quality parameters (non-uniform illumination, noise, low resolution)

can affect the performance of the detection algorithm. So, considering this situation, an input image used in work is a profile image that has been fetched from Siblings Image Database. Images (Fig. 3) from this database are in a resolution of 425x2832 pixels, with the uniformed background. Images have been taken under controlled lighting. Also, those images have been used for cases without anisocoria.

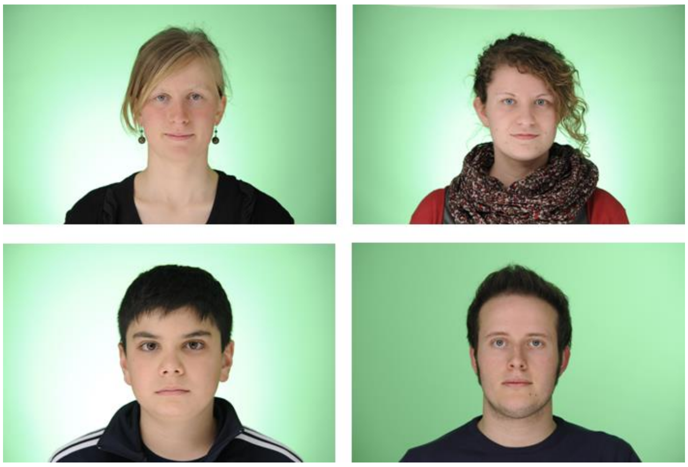


Figure 3:
Input data (Samples from the database for the case without anisocoria presence)

Because no database has profile pictures of people for cases where anisocoria is present, a database has been created with images downloaded from the Internet. That is the reason why the algorithm is tested by applying different quality and size images. Some samples of the input image for this case are given in Fig. 4.

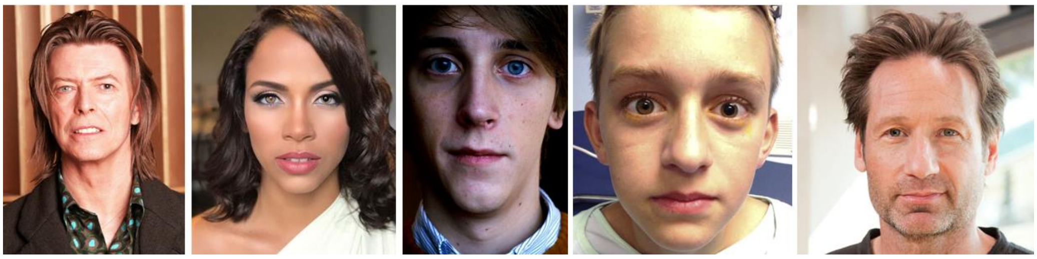


Figure 4:
Input data (Samples from the Internet for the case with anisocoria presence)

2.2. Pupil Detection and Measurement

In literature, different techniques have been proposed to detect the pupil of the human eye. Review of the literature demonstrates that the method which uses high contrast between the dark intensity of the pupil region and the bright sclera region is commonly used way for pupil detection.

Many algorithms localize pupil region or iris region in some studies with combinations of various image processing methods such as thresholding, morphological operations, linked component analysis, and others with different additional steps. Some algorithms are given as an example in Figure 5.

Algorithms	**Thresholding**	**Morphological Operations**	**Edge Detection**	**Fitting of Circle or Ellipse**	**Additional Step**
(Pasarica, Cehan, Bozomitu, & Lupu, 2015)	✓			Hough transform	Gaussian smoothing filter
(Nowak, Zarowska, Szul-Pietrzak, & Misiuk-Hojło, 2014)	✓	✓	✓	Hough transform	Median filter
(Mahmood, Uyop, Mansor, & Jumadi, 2011)	✓	✓	✓		
(Lin, Lin, LiFang, & Lun, 2010)	✓	✓	✓		Centre of mass
(Basit, Javed, & Masood, 2008)	✓		✓		DivisionPoints
(Dey & Samanta, 2007)			✓	Circle	BLOB

Figure 5:
Pupil detection algorithms classification

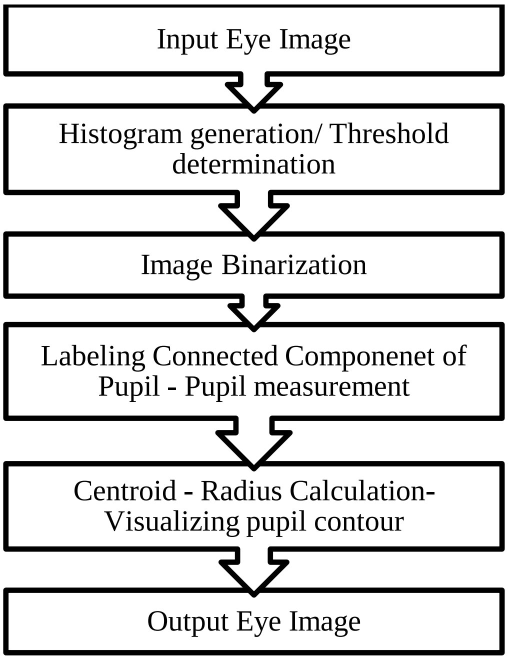
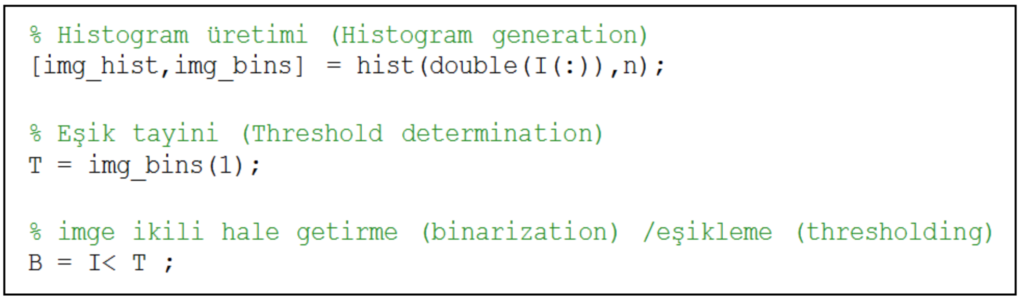

Input Eye Image
Histogram generation/ Threshold determination
Image Binarization
Labeling Connected Componenet of Pupil - Pupil measurement
Centroid - Radius Calculation- Visualizing pupil contour
Output Eye Image

Figure 6:
The steps of the proposed method for pupil detection and measurement

Figure 6 shows the steps of the proposed method for pupil detection and measurement. Pupil detection is implemented using these image processing techniques: Histogram generation, Image binarization, Connected components labelling, and Centroid-radius calculation. This method is used for the automatic detection of the pupil.

Histogram Generation and Image Binarization

The pupil localization involves the image thresholding and binarization. Converting a grayscale image of an eye to the binary image is based on a threshold. In the detection of the pupil, the purpose of binarization and image thresholding is to divide the eye image into two parts, the pupil and the background. Finding the appropriate threshold value as a general rule has a significant impact on the final result. The darkness of the pupil colour is more or less the same for every human eye, and that is the reason why the same threshold value can be used in this case. The Matlab code used for histogram generation and threshold value determination is shown in Figure 7.



```
% Histogram üretimi (Histogram generation)
[img_hist,img_bins] = hist(double(I(:)),n);

% Eşik tayini (Threshold determination)
T = img_bins(1);

% imge ikili hale getirme (binarization) /eşikleme (thresholding)
B = I< T ;
```

Figure 7:
Matlab code used for histogram generation and threshold value determination

[*img_hist*, *img_bins*] = hist(*I*,*n*) calculates the histogram for the grayscale image **I**. The ***hist()*** function returns the histogram counts in ***img_hist*** and the bin locations in ***img_bins***. **n** specifies the number of columns used to calculate the histogram. The input image is grayscale, so the default value for n is 256 (Figure 8). In case that input image is a binary image, n is two columns.

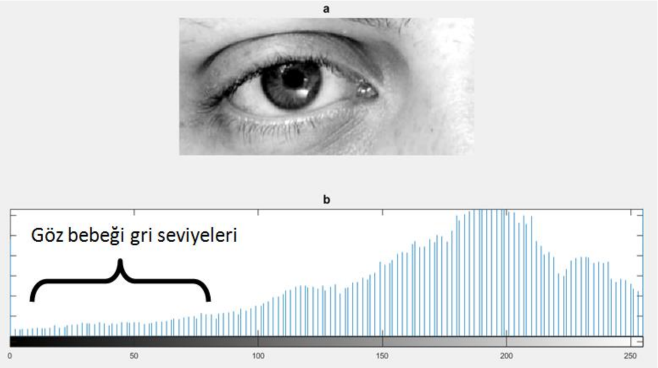


Figure 8:
Grayscale eye image (a) and its histogram(b)

T (threshold value) is selected to remove the object (pupil) from the background. I is a grayscale image. If the value of the point I (x, y) is smaller than the threshold value T, that point is an object point (pupil). Otherwise, the point is the background point:

$$B(x,y) = \begin{cases} 1, \ I(x,y) < T & Object(Pupil)\ Point \\ 0\ ,\ I(x,y) \geq T & Background\ Point \end{cases}$$

An example of the Grayscale eye image and its histogram is shown in Figure 8.

Labeling Connected Component of Pupil and Pupil Size Measurement

To eliminate noisy pixels and accurately identify the pupil feature from the threshold eye image, the connected component labelling algorithm is used. A region in an image is a set of pixels with similar intensity values adjacent to each other. Areas are tagged with different colours after subtraction. Some results of this step are shown in Figure 8c.

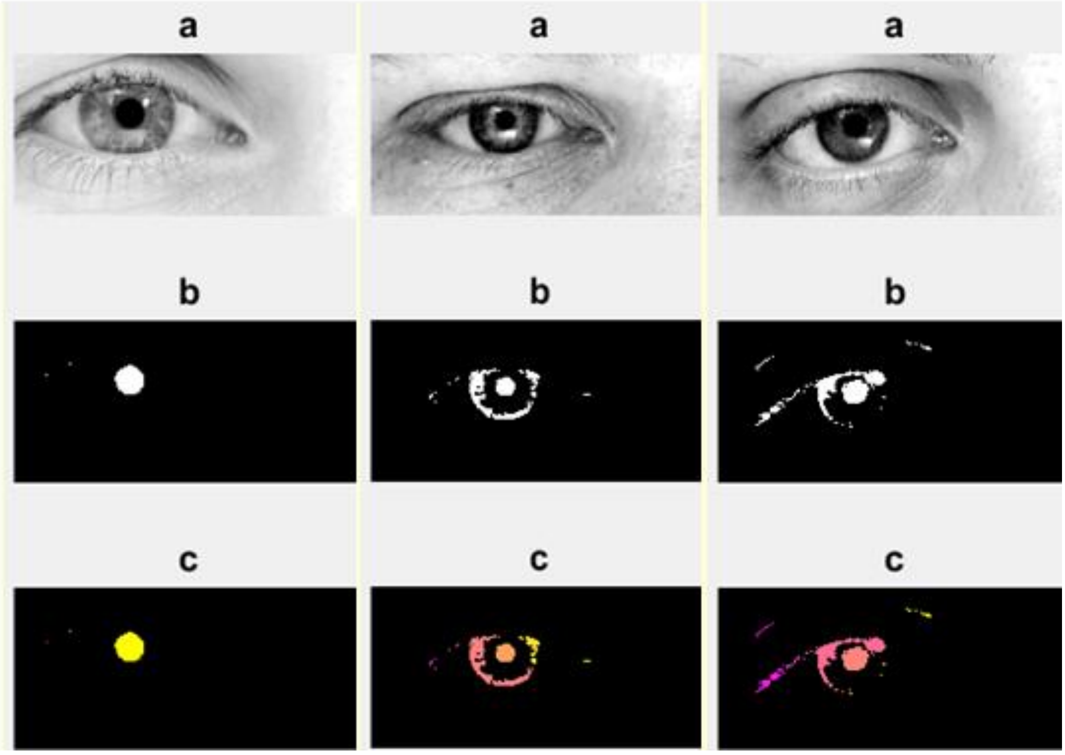


Figure 9:
Grayscale eye image (a), binary image(b), and labeled image (c)

Matlab code used for connected component labeling and pupil size measurement is shown in Figure 10.

In Matlab, the bwlabel function is used for labeling a binary image . EI is the input eye image. Pixel connectivity is 8. Pixels are connected when they touch the edges or corners. Two adjacent pixels are part of the same object, if both are connected in horizontal, portrait, or diagonal direction.

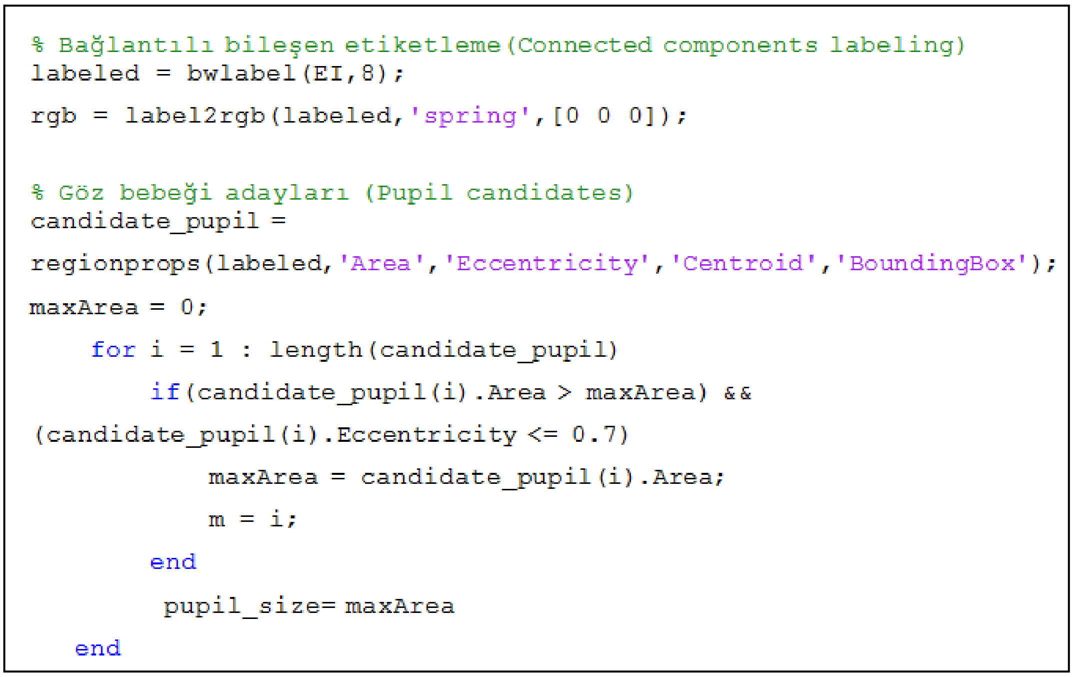
rgb is a new lebeled image.The ***label2rgb*** function converts the label matrix (rgb image) to RGB image. ***RGB = label2rgb (L, cmap, zerocolor)*** indicates the RGB color of the background elements (the labeled pixels are 0).

L (labeled) - The label matrix of adjacent regions specified as the m-by-n matrix of non-negative integers. Background pixels are labeled with 0. Pixels labeled with 1 create an object; pixels labeled with 2 form a second object; and such.

cmap (spring) - adjusts the background pixels to cyan color and randomizes how the color is assigned to the label.

zerocolor ([0 0 0] - black) -The label matrix of adjacent regions is specified as a 3-element vector.

This algorithm also determines pupil characteristics. ***Regionprops*** returns measurements (Area, Eccentricity, Centroid (Mass center), BoundingBox for the set of properties specified by the properties of each 8-linked component in the binary image (***labeled***).



```
% Bağlantılı bileşen etiketleme(Connected components labeling)
labeled = bwlabel(EI,8);
rgb = label2rgb(labeled,'spring',[0 0 0]);

% Göz bebeği adayları (Pupil candidates)
candidate_pupil =
regionprops(labeled,'Area','Eccentricity','Centroid','BoundingBox');
maxArea = 0;
    for i = 1 : length(candidate_pupil)
        if(candidate_pupil(i).Area > maxArea) &&
(candidate_pupil(i).Eccentricity <= 0.7)
            maxArea = candidate_pupil(i).Area;
            m = i;
        end
         pupil_size= maxArea
   end
```

Figure 10:
Matlab code used for connected component labeling and pupil size measurement

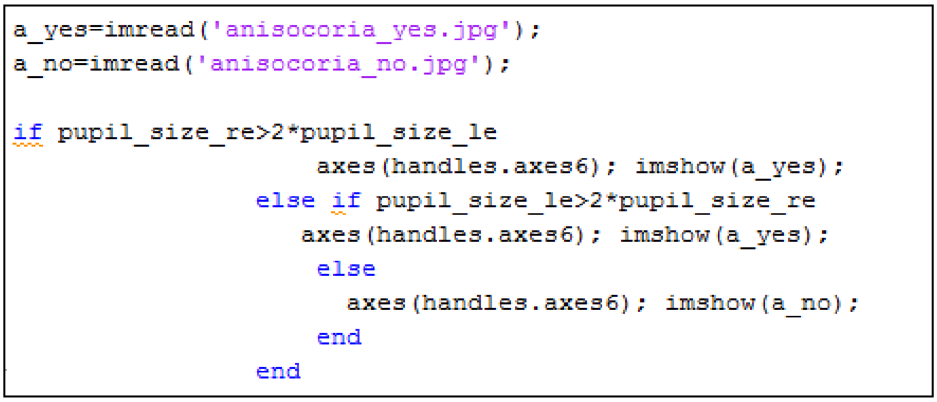
Image name	Resolution	Left Eye	Right Eye
Image 1	640x464	232	51
Image 2	186x271	12	41
Image 3	640x778	276	48
Image 4	960x1207	736	291
Image 5	892x1020	55	19

Figure 11:
Size of the left and right pupil area (the case with anisocoria presence)

Image name	Resoltion	Left Eye	Right Eye
Image 1	4256x2832	440	442
Image 2	4256x2832	472	461
Image 3	4256x2832	597	579
Image 4	4256x2832	1116	1256
Image 5	4256x2832	821	833
Image 6	4256x2832	554	431
Image 7	4256x2832	650	640
Image 8	4256x2832	641	767
Image 9	4256x2832	329	359
Image 10	4256x2832	1120	1120
Image 11	4256x2832	404	413
Image 12	4256x2832	667	627
Image 13	4256x2832	684	633
Image 14	4256x2832	434	464
Image 15	4256x2832	685	568

Figure 12:
Size of the left and right pupil area (the case without anisocoria presence)

Size of left and right eye pupil are measured at this step of pupil detection. The results are shown in Figure 11 and Figure 12. These data were used for anisocoria detection. In the case with anisocoria, it was concluded that one pupil size was at least two times larger than the other.



```
a_yes=imread('anisocoria_yes.jpg');
a_no=imread('anisocoria_no.jpg');

if pupil_size_re>2*pupil_size_le
                    axes(handles.axes6); imshow(a_yes);
                else if pupil_size_le>2*pupil_size_re
                   axes(handles.axes6); imshow(a_yes);
                    else
                      axes(handles.axes6); imshow(a_no);
                    end
                end
```

Figure 13:
Matlab code used to determine the presence of anisocoria in the given image

Matlab code used to determine the presence of anisocoria in the given image is shown in Figure 13. ***Pupil_size_re*** is the size of the right pupil. ***Pupil_size_le*** is the size of the left pupil.

Centroid - Radius Calculation and Visualizing Pupil Contour

Matlab code used for centroid - radius calculation and visualizing pupil contour is shown in Figure 14.

The main idea is to draw a circle with R radius centered in coordinates (Cx, Cy). The circle was created in polar coordinates and then turned into Cartesian coordinates using the ***pol2cart*** function.

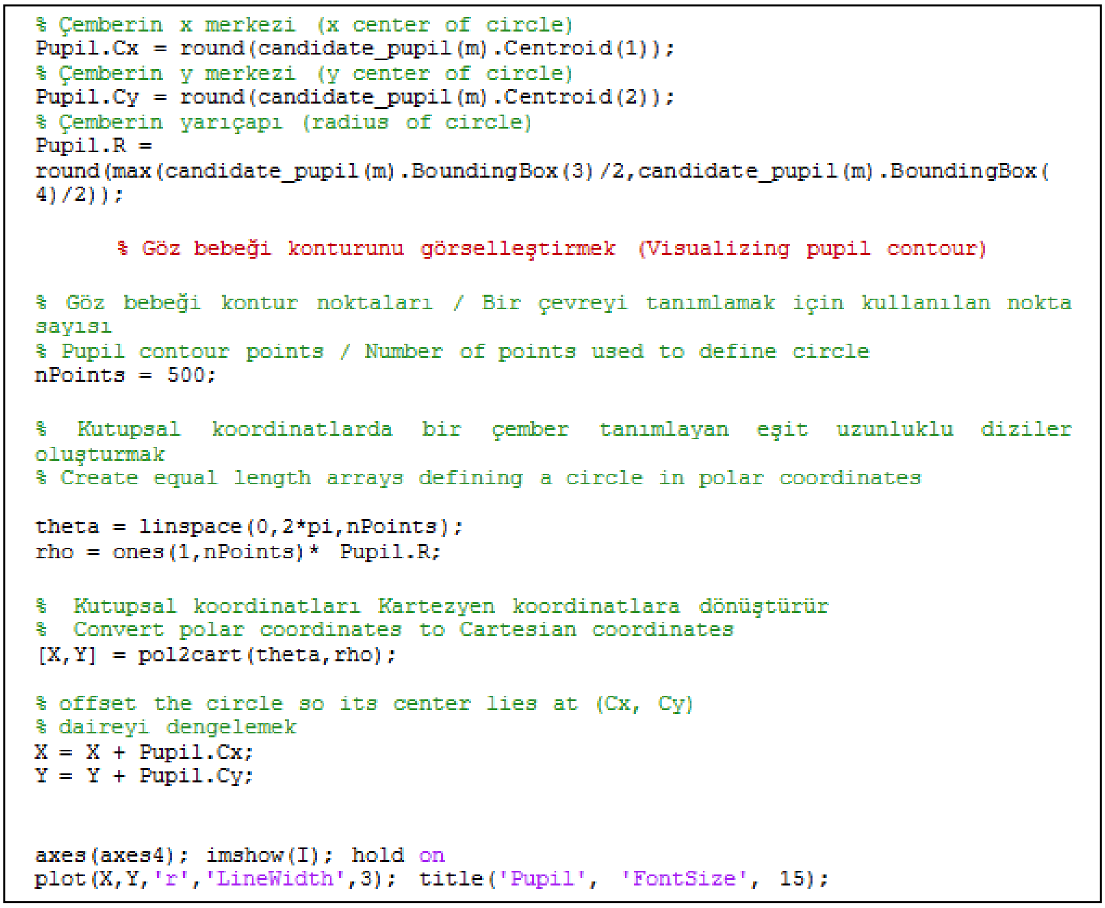
theta = linspace(0,2*pi,nPoints); This line and the next define a circle in polar coordinates. This line of code creates a theta array with nPoints (pupil contour points) ranging from 0 to 2 *π.

rho = ones(1,nPoints) * Pupil.R; The rho array stores the radius for each defined theta value.

The ***ones(1,nPoints)*** command creates a 1xnPoints line matrix filled with ones.

[X,Y] = pol2cart(theta,rho); The ***pol2cart*** function converts polar coordinates ***(theta, rho)*** to Cartesian coordinates ***(x, y)***. This function is equivalent to the following:

$$x = rcos\theta$$
$$y = rsin\theta$$



```
% Çemberin x merkezi (x center of circle)
Pupil.Cx = round(candidate_pupil(m).Centroid(1));
% Çemberin y merkezi (y center of circle)
Pupil.Cy = round(candidate_pupil(m).Centroid(2));
% Çemberin yarıçapı (radius of circle)
Pupil.R =
round(max(candidate_pupil(m).BoundingBox(3)/2,candidate_pupil(m).BoundingBox(
4)/2));

      % Göz bebeği konturunu görselleştirmek (Visualizing pupil contour)

% Göz bebeği kontur noktaları / Bir çevreyi tanımlamak için kullanılan nokta
sayısı
% Pupil contour points / Number of points used to define circle
nPoints = 500;

% Kutupsal koordinatlarda bir çember tanımlayan eşit uzunluklu diziler
oluşturmak
% Create equal length arrays defining a circle in polar coordinates

theta = linspace(0,2*pi,nPoints);
rho = ones(1,nPoints)* Pupil.R;

% Kutupsal koordinatları Kartezyen koordinatlara dönüştürür
% Convert polar coordinates to Cartesian coordinates
[X,Y] = pol2cart(theta,rho);

% offset the circle so its center lies at (Cx, Cy)
% daireyi dengelemek
X = X + Pupil.Cx;
Y = Y + Pupil.Cy;


axes(axes4); imshow(I); hold on
plot(X,Y,'r','LineWidth',3); title('Pupil', 'FontSize', 15);
```

Figure 14:
Matlab code used for centroid - radius calculation and visualizing pupil contour

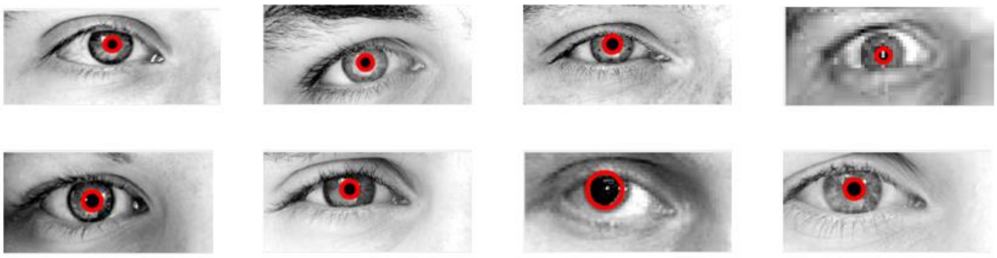


Figure 15:
Some examples of correct pupil detection

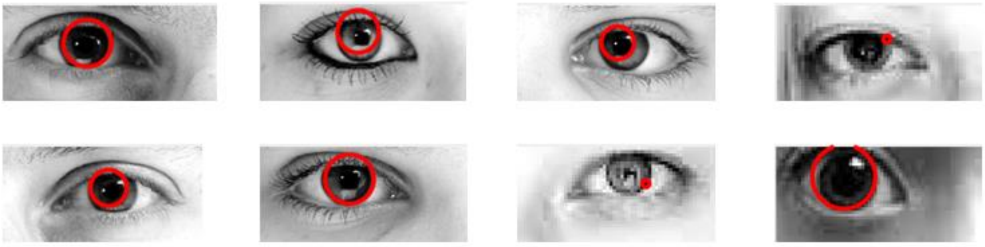


Figure 16:
Some examples of incorrect pupil detection

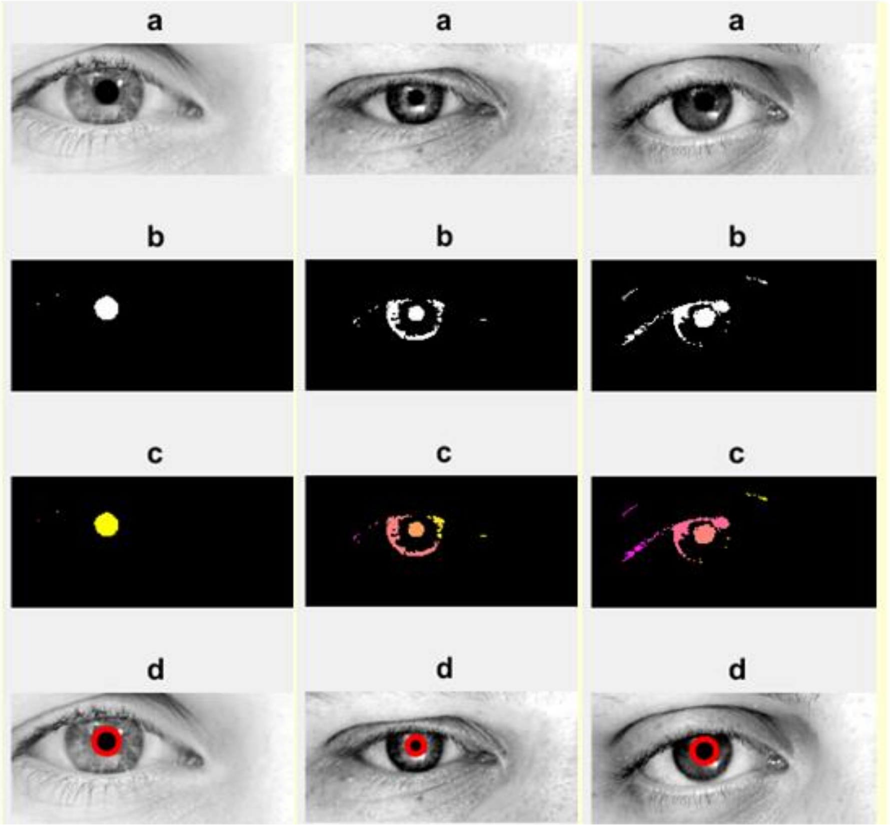


Figure 17:
Results of pupil detection steps: grayscale image (a), binary image after binarization (b), image labeled after connected component labeling (c), and image with pupil contour after centroid - radius calculation.

3. EXPERIMENTAL RESULTS

The performances of the proposed algorithm were analyzed and measured using 75 images (60 images having been randomly selected from the database and 15 downloaded images from the Internet). Detection rates for face area, eyes and pupil (when anisocoria case is not present) are shown in Figure 18 for the random images taken from the Siblings Image Database. Detection rates for images downloaded from the Internet (when anisocoria case is present) are shown in Figure 19.

Siblings Image Database	Total Number of Input Image	Correct result	Error result	**Detection rate %**
Face Area Detection	60	54	6	**90**
Eyes Region Detection	54	47	7	**87**
Right Pupil Detection	47	29	18	**61**
Left Pupil Detection	47	23	24	**48,9**

Figure 18:

Detection rates for face area, eyes and pupil (when anisocoria case is not present)

Internet	Total Number of Input Image	Correct result	Error result	**Detection rate %**
Face Area Detection	15	14	1	**93,3**
Eyes Region Detection	14	13	1	**92,8**
Right Pupil Detection	13	29	18	**61**
Left Pupil Detection	13	6	7	**46,2**

Figure 19:

Detection rates for face area, eyes and pupil (when anisocoria case is present)

The values of the possible results for the proposed algorithm are given in Figure 20. There are four possible outcomes:

- TP (True Positive) anisocoria is present and the test result is positive,
- TN (true negative) anisocoria is not present and the test result is negative
- FN (False Negative) anisocoria is present and the test result is negative,
- FP (False Positive) anisocoria is not present and the test result is positive.

		ANISOCORIA		
		Present	Absent	**Total**
TEST RESULT	Positive	TP=11	FP=22	**33**
	Negative	FN=2	TN=25	**27**
	Total	**13**	**47**	**60**

Figure 20:
Probability table and performance measurements

The probability of the proposed algorithm to accurately determine the presence of anisocoria is 85% (Sensitivity). The likelihood of accurately determining the absence of anisocoria is 53% (Specificity).

The Positive Predictive Value (PPV) test result shows how many of the positive ones have anisocoria. PPV value for the proposed algorithm is 33%. Similarly, the Negative Predictive Value (NPV) shows how many of the negative ones do not have anisocoria. The NPV value for the proposed algorithm is 92%.

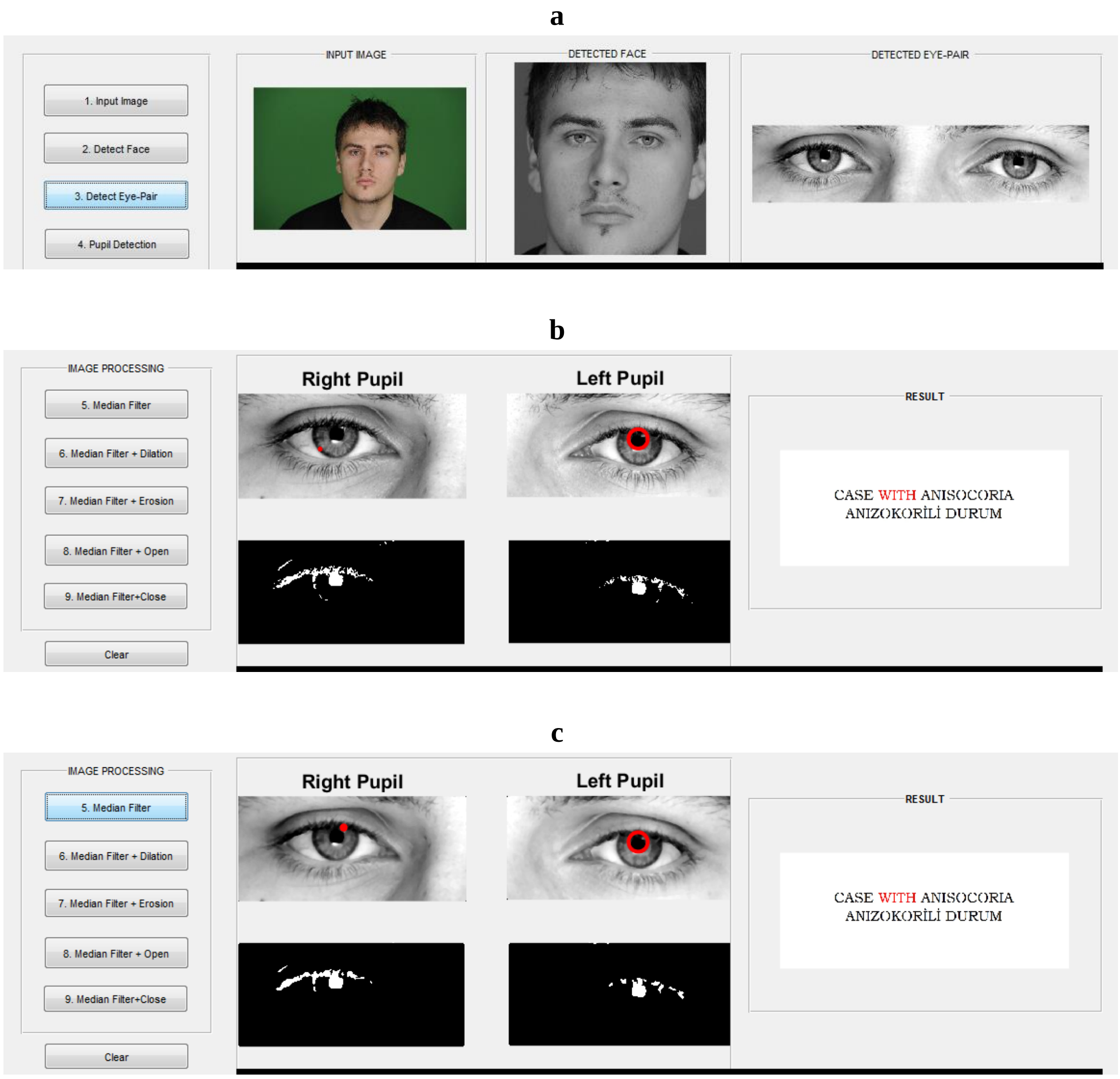
The False Positive Rate (FPR) is the number of cases tested for the absence of anisocoria but is detected as the present. The FPR value for the proposed algorithm is 46%. The False Negative Rate (FNR) is the number of cases tested in the absence of anisocoria and is determined as absent. The FNR value for the proposed algorithm is 46%.

The performance metric with their calculation equations and result values are given in Figure 21.

METRIC	**EQUATION**	**DETECTION RATE %**
Sensitivity	TP/(TP+FN)	85
Specificity	TN/(TN+FP)	53
Positive Predictive Value (PPV)	TP/(TP+FP)	33
Negative Predictive Value (NPV)	TN/(TN+FN)	92
False Positive Rate (FPR)	FP/(FP+TN)	47
The False Negative Rate	FP/(FP+TP)	67
Accuracy	(TN+TP)/(TN+TP+FN+FP)	60

Figure 21:
The performance metric with their calculation equations and result values

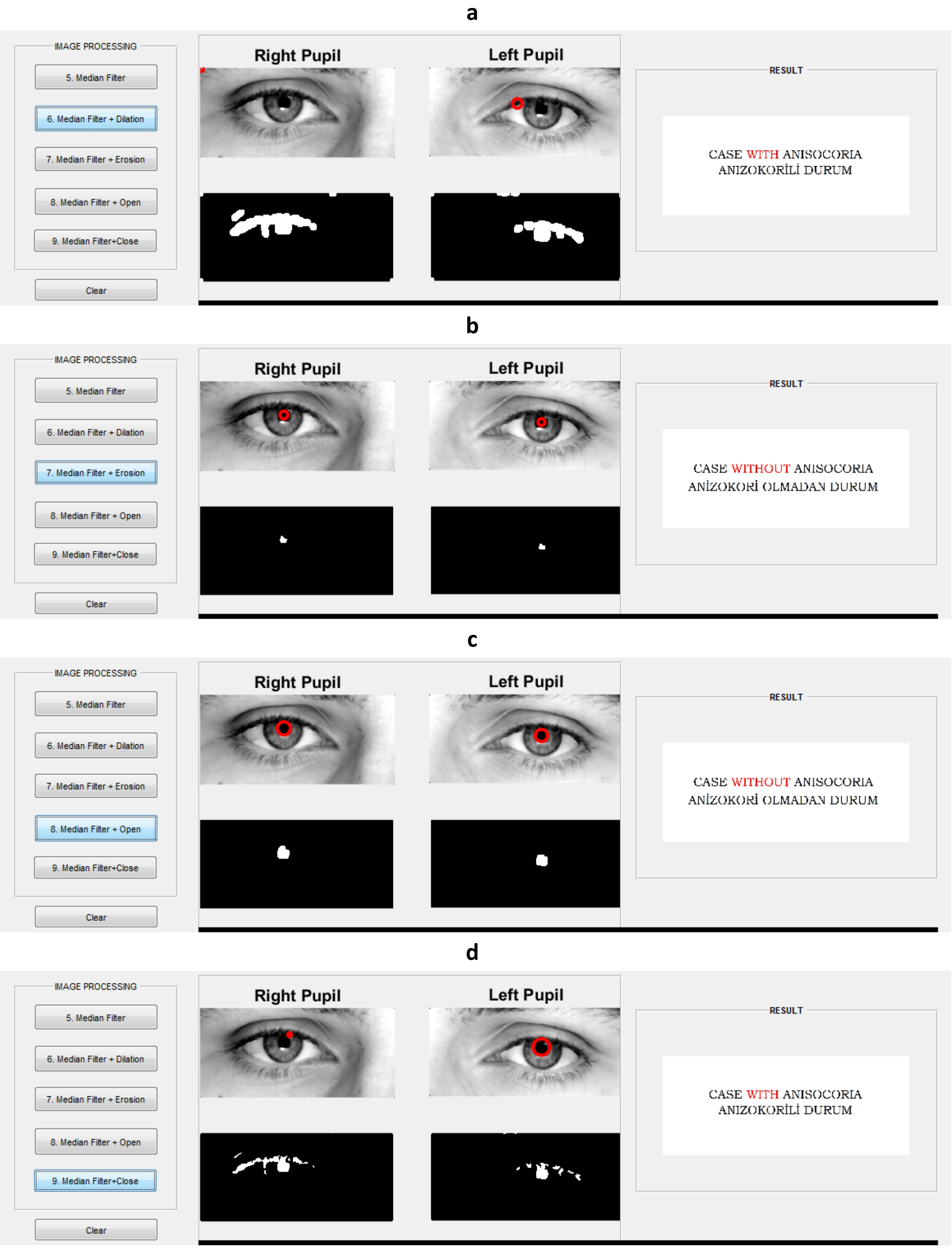
Image processing techniques can be used to improve the performance of the proposed algorithm. Figure 22 and Figure 23 show the results of right and left pupil detection by use of different methods. When the pupil area detection and measurement is correct, the anisocoria detection performance rate is 100%. From the results that are given in Figure 24, it was concluded that the proposed algorithm gives the best results when it is used in conjunction with the median filter and morphological opening. The detection rate for the right eye increased by 20% and for the left eye, it increased by 18%. Detection performances of different methods were analyzed for 50 images.



CASE WITH ANISOCORIA
ANIZOKORİLİ DURUM

CASE WITH ANISOCORIA
ANIZOKORİLİ DURUM

Figure 22:
Results of right and left pupil detection with use of different image processing techniques: input image, face and eye pair detection (a), proposed algorithm without image processing techniques (b), proposed algorithm +median filter(c).



a: CASE WITH ANISOCORIA
ANIZOKORİLİ DURUM

b: CASE WITHOUT ANISOCORIA
ANİZOKORİ OLMADAN DURUM

c: CASE WITHOUT ANISOCORIA
ANİZOKORİ OLMADAN DURUM

d: CASE WITH ANISOCORIA
ANIZOKORİLİ DURUM

Figure 23:

Results of right and left pupil detection with use of different image processing technique (median filter + different morphological operators): dilation(a), erosion (b), opening (c) and closing(d)

	Pupil Detection	**Correct result**	**Error result**	**Not detected**	**Detection rate %**
Proposed algorithm	Right Pupil	31	19	0	62
	Left Pupil	25	25	0	50
Proposed algorithm +Median filter	Right Pupil	32	17	1	64
	Left Pupil	26	23	1	52
Proposed algorithm +Median filter +Erosion	Right Pupil	27	0	23	54
	Left Pupil	23	4	23	46
Proposed algorithm +Median filter + Opening	Right Pupil	41	4	5	82
	Left Pupil	34	11	5	68

Figure 24:
Results of right and left pupil detection with use of different image processing technique

4. CONCLUSION

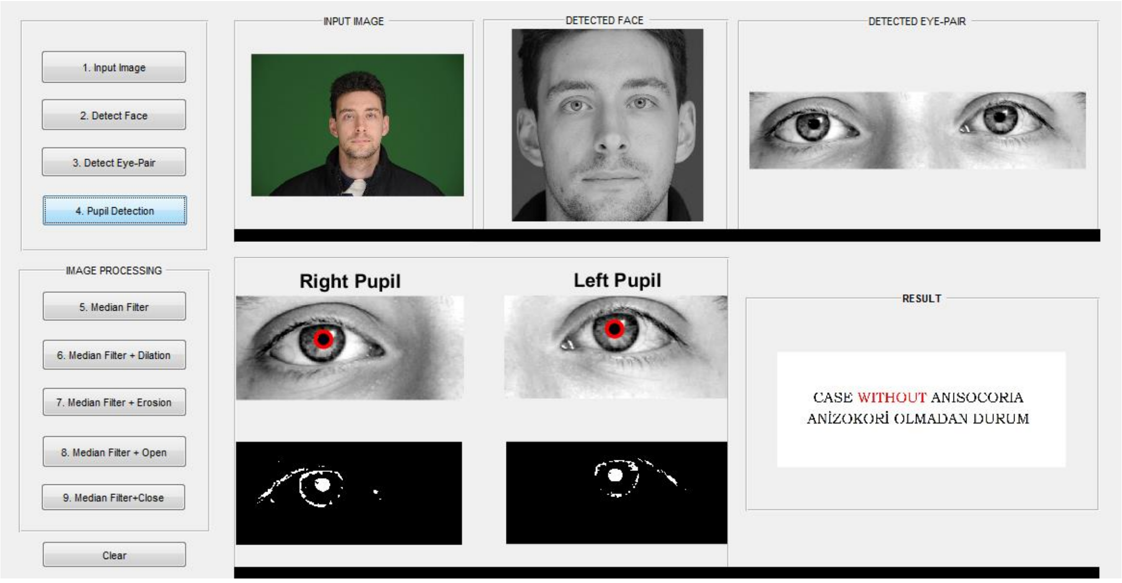
This paper provides information about the proposed automatic algorithm for the detection of anisocoria from a digital image using image processing techniques. This algorithm combines face detection, eye detection, and pupil detection and measurement from a digital image of a person that may have anisocoria.

The proposed algorithm determines the anisocoria state using the size information from the adult human pupil. The main factors that have an effect on anisocoria determination are the accuracy of the detection and visualization of the pupil. When the pupil area detection and measurement is correct, the anisocoria detection performance rate is 100%. For future studies, it will be useful to focus on pupil detection.

One of the factors that also affect the result is that not all images have the same quality. The proposed algorithm tests were carried out on 75 images with different features. The test result showed that the proposed algorithm had an accuracy rate of 60%. Image processing techniques could be used to improve the performance of the proposed algorithm. The proposed algorithm was used in conjunction with a median filter and basic morphology operations to improve pupil detection. After testing, it was concluded that the proposed algorithm gives the best results in combination with the median filter and morphological opening.

In the future database that will contain images of person with and without anisocoria should be created. Images will be taken under controlled conditions that will for sure, affect testing result accuracy and improve this algorithm performance.

All algorithms steps, described in this paper, are done in MATLAB R2016 computing environment. Graphical user interface (GUI) for proposed algorithm in this paper was developed to make its use easier for the user because pupil detection and size measurement is the crucial step of many applications in different fields such as ophthalmology, computer vision, psychology, etc.



CASE WITHOUT ANISOCORIA
ANİZOKORİ OLMADAN DURUM

Figure 25:
Graphical user interface (GUI) for proposed algorithm

REFERENCES

1. Basit, A., Javed, M. Y., & Masood, S. (2008). Non-circular Pupil Localization in Iris Images. 2008 4th International Conference on Emerging Technologies. DOI: 10.1109/ICET.2008.4777505

2. Dey, S., & Samanta, D. (2007). An Efficient Approach for Pupil Detection in Iris Images. 15th International Conference on Advanced Computing and Communications (ADCOM 2007). DOI: 10.1109/ADCOM.2007.79

3. Kaiser,P.K., Friedman, N. J., & Pineda, R. (2014). The Massachusetts Eye and Ear Infirmary Illustrated Manual of Ophthalmology (4th Edition) Elsevier Saunders. eBook ISBN: 9780323225274

4. Iacoviello, D. (2006). Analysis of pupil fluctuations after a light stimulus by image processing and neural network. Computers & Mathematics with Applications, Volume 53, Issue 8 p.1260-1270. DOI: 10.1016/j.camwa.2006.05.022

5. Lin, L., Lin, P., LiFang, W., & Lun, Y. (2010). A robust and accurate detection of pupil images. 3rd International Conference on Biomedical Engineering and Informatics. Yantai, China: IEEE. DOI: 10.1109/BMEI.2010.5639646

6. Mahmood, N. H., Uyop, N., Mansor, M. M., & Jumadi, A. M. (2011). Measurement of the Area and Diameter of Human Pupil Using Matlab. 5th Kuala Lumpur International Conference on Biomedical Engineering (p. vol 35.). Berlin, Heidelberg: Springer. DOI:10.1007/978-3-642-21729-6_167

7. Nowak, W., Zarowska, A., Szul-Pietrzak, E., & Misiuk-Hojło, M. (2014). System and measurement method for binocular pupillometry to study pupil size variability. BioMedical Engineering OnLine. DOI: 10.1186/1475-925X-13-69.

8. Pasarica, A., Cehan, V., Bozomitu, R. G., & Lupu, R. (2015). Pupil detection algorithms for eye tracking applications. 2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging. DOI: 10.1109/SIITME.2015.7342317

9. Viola, P., & Jones, M. (2004). Robust real-time object detection. International Journal of Computer Vision, 58(2):137-154. DOI: 10.1023/B:VISI.0000013087.49260.fb