

**T.C.**

**TEKİRDAĞ NAMIK KEMAL ÜNİVERSİTESİ**

**ÇORLU MÜHENDİSLİK FAKÜLTESİ**

**EHM-428 Yapay Görme ve Örüntü Tanımaya Giriş**

**MR Görüntüsünden Beyin Tümörü Tespiti**

**Uğur ILGIN**

**2150656815**

**BİLGİSAYAR MÜHENDİSLİĞİ**

# İÇİNDEKİLER

İÇİNDEKİLER.....	2
1. Proje hakkında Genel bilgiler.....	3
1.1. Projenin Konusu .....	3
1.2. Projenin Hedefleri .....	3
1.3. Projede Kullanılan Teknolojiler.....	4
2. Materyaller ve Metot .....	4
2.1. Görüntü Veriseti .....	4
2.2. Kullanılan Algoritmalar .....	5
2.2.1. Median Filter .....	5
2.2.2. Edge Detection (Kenar Algılama).....	6
2.2.3. Histogram .....	6
2.3. Projenin Tasarımı.....	7
3. Terimler .....	13
KAYNAKLAR .....	14
EKLER .....	15
4. ALGORİTMALAR.....	15
5. KODLAR.....	22
5.1. Mainmenu.m .....	22
5.2. Hizlitumor.m .....	23
5.3. Detaylitumor.m .....	25

## **1. PROJE HAKKINDA GENEL BİLGİLER**

### **1.1. Projenin Konusu**

Beyin tümörleri, kafatası içerisinde büyüyerek beyin üzerine yaptıkları baskı ile ilerleyen evrelerde yaşamı tehdit edebilmektedir. Kötü huylu beyin tümörleri son yıllarda insan ölümlerinin önemli nedenlerinden bir tanesi olmaya başlamıştır. Eğer tümör erken bir aşamada tespit edilerek doğru sınıflandırılabilirse, hastaların hayatta kalma şansı artırılabilir. Beyin kanseri için seçilecek en uygun tedavi, doktorun tümörün türünü, konumunu, büyüklüğünü ve sınırlarını kesin bir şekilde tespit etmesine bağlıdır. Bu sebeple, radyolog ve doktorların beyin tümörlerini yüksek başarıyla tespit edebilmesine yardımcı olacak bir bilgisayar destekli teşhis / tespit sistemi kullanması son derece önemlidir. Bu çalışmada, MR görüntülerini üzerinden MATLAB kullanılarak beyin tümörü analizi gerçekleştirilmesi hedeflenmiştir. Bu amaçla kullanıcı arayüzüne sahip bir uygulama yazılımı tasarlanmıştır. Böylece hekimlerin karar verme aşamalarında hem kolaylık sağlanarak oluşabilecek muhtemel hataların önüne geçilmesi hem de tasarlanan uygulamanın ikincil bir araç olarak kullanılması hedeflenmiştir. Tasarlanan uygulama yardımıyla beyin tümörlerinin başarılı bir şekilde tespit edildiği görülmüştür.

### **1.2. Projenin Hedefleri**

MATLAB kullanılarak görsel kullanıcı arayüzüne sahip bir uygulama geliştirmektir.

1. Görsel kullanıcı arayüzü sahip bir uygulama geliştirmek
2. MR Görüntüleri kullanılarak beyin tümörünün tespit edilmesi
3. Median Filter, Edge Detection, Segmentation algoritmalarının MR Görüntülerine uygulanması
4. Örnek resmin histogramını çizdirmek
5. Tüm resim formatlarının açabilen bir uygulama tasarlamak

Yapılmış olan kullanıcı arayüzlü MR Görüntülerinden Beyin Tümörü Tespit programı yukarıda hedefleri gerçekleştirmeyi hedeflemektedir.

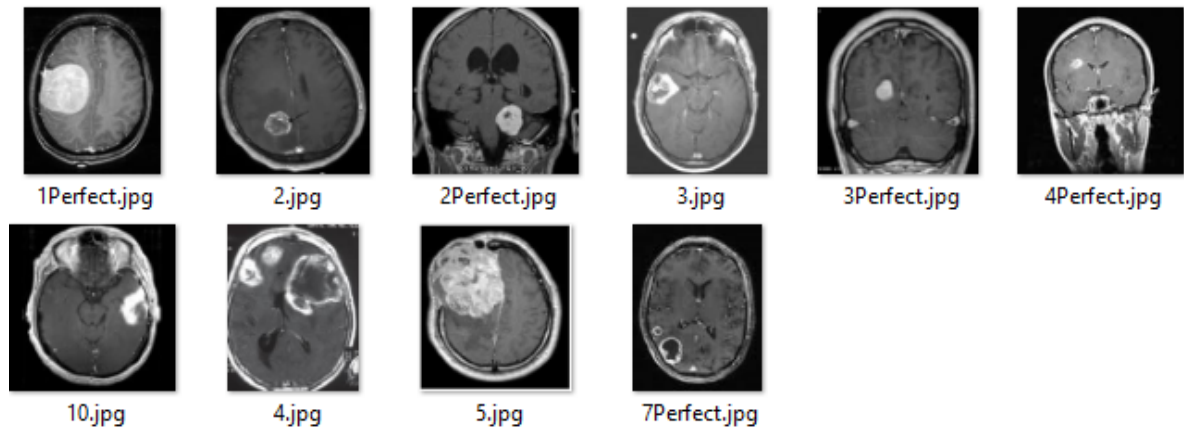
### 1.3. Projede Kullanılan Teknolojiler

Projede kullanılan programlama dili Yapay Zeka ,Sinyal İşleme ve Görüntü İşleme alanında sıklıkla tercih edilen MATLAB programlama dilidir .

## 2. MATERYALLER VE METOT

### 2.1. Görüntü Veriseti

Bu çalışma için 23 farklı MR görüntüsü kullanılmıştır.Bu görüntülerden 11 tanesi iyi huylu tümör kategorisindedir ve tümör bulunmayan MR görüntülerini kapsamaktadır.12 tanesi ise kötü huylu tümör kategorisine girmektedir.Görüntüler hastanedeki Siemens marka MR tarayıcılarından öncelikle DICOM formatı ile alınarak, daha sonra 2 boyutlu görüntü formatı şeklinde kaydedilmiştir. Görüntü verisetindeki beyin MR görüntüleri için bazı örnekler Şekil 2.1' de gösterilmiştir.



Şekil 2.1. Örnek MR görüntüleri

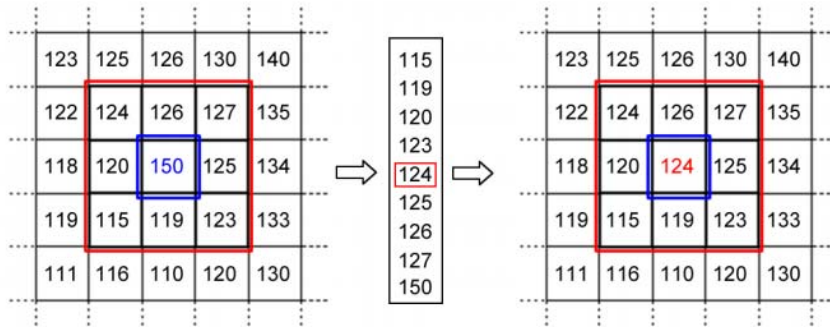
## 2.2. Kullanılan Algoritmalar

### 2.2.1. Median Filter

Median Filter (Medyan Filtresinin) yaygın kullanılan isimleri; Medyan filtresi (Orta Değer filtresi) (Median filtering), Sıralama filtresi (Rank filtering) Medyan filtresi, bir resimdeki gürültüyü azaltmak için kullanılır. Medyan filtresi her pikselin değerini hesaplamak için yakınındaki komşularına bakar. Medyan filtresinde piksel değeri komşu pikselleri sıralayıp sıranın ortasındaki değeri alır. Eğer incelenen bölge (şablonun içerisi) çift sayıda piksel varsa, orta değer olarak, ortada bulunan iki pikselin ortalaması kullanılır.

Şekil 2.2' de ortadaki piksele göre işlemlerimizi yaparsak bu pikselin değeri olan 150 sayısı çevresindeki pikselleri iyi bir şekilde temsil etmediğini görebiliriz. Bu pikselin değerini değiştirirken öncelikle çevresindeki piksellerin değerini bir sıraya dizelim. Bunlar (115, 119, 120, 123, 124, 125, 126, 127,150) değerlerinden oluşur. Bu değerlerin en ortasında ise 124 sayısı vardır. Buna göre 150 rakamı 124 sayısı ile değiştirilir. Burada 124 sayısı medyan sayısı (orta değer) olmuş olur.

Burada kullanılan şablon 3x3 piksel boyutlarındadır. Daha büyük şablonların kullanılması daha fazla pürüzsüzleştirme (yumuşatma) etkisi üretir.



Şekil 2.2. Medyan Filtresi Örneği

Matematiksel ifadesi ise;

$$y = [m, n] = \text{median}\{x[i, j], (i, j), \in w\}$$

Şekil 2.3. Medyan Filtresi Matematiksel Gösterimi

### 2.2.2. Edge Detection (Kenar Algılama)

Kenar algılama ise temel bölütleme yöntemlerinden biridir. Kenar algılama, temel olarak görüntü yoğunluğundaki süreksizliklerin, özellikle yoğunluğun keskin bir biçimde değişme eğiliminde olduğu kenarlar boyunca tanımlanmasını amaçlamaktadır . Kenar tespiti, insan beyni üzerinde yapılan çalışmalarda özellik tanıma ve özellik çıkarma için kullanılan önemli bir araçtır. MR görüntüleri üzerinden hastalık teşhisi yapmak için kullanılan kenar algılama ve bölütleme üzerine birçok araştırma bulunmaktadır. Sobel Edge Detection en kolay ve en çok kullanılan kenar algılama algoritmasıdır. Şekil 2.4. ' teki formüller kullanılarak köşeler hesaplanır

$$\Delta f = mag(\Delta f) = \sqrt{G_x^2 + G_y^2}$$

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

Şekil 2.4. Sobel Edge Detection Matematiksel Gösterimi

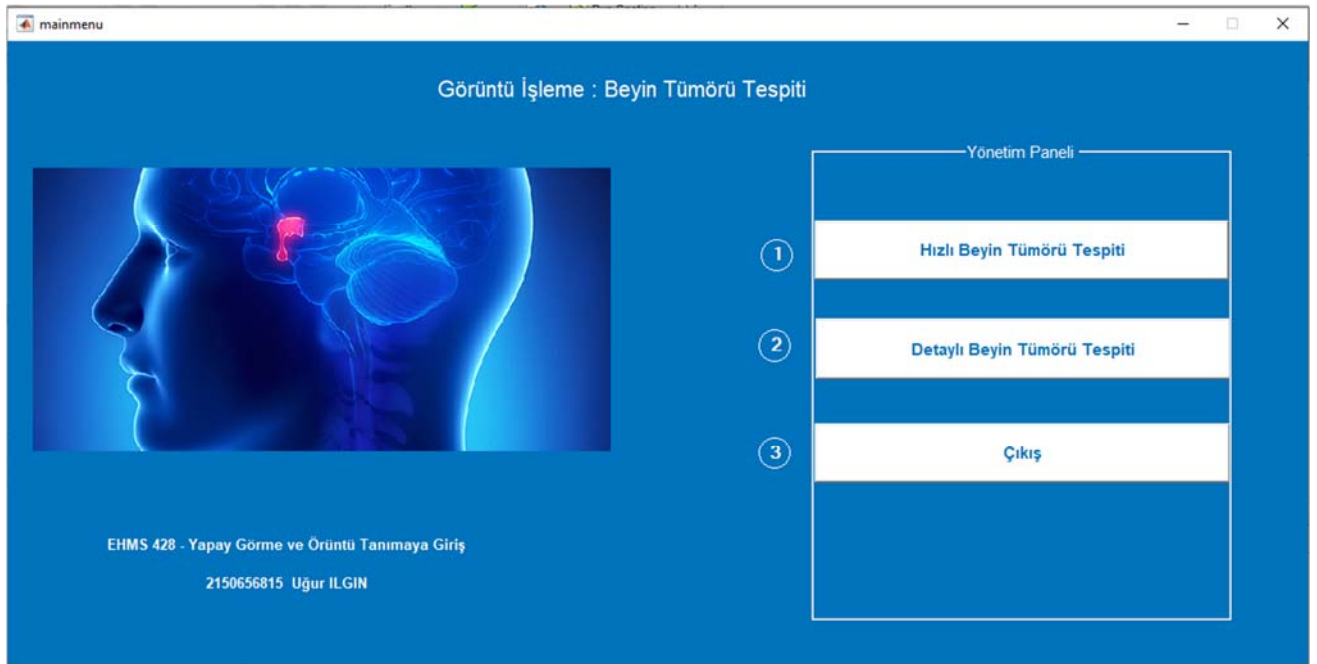
### 2.2.3. Histogram

Gruplandırılmış bir veri dağılımının sütun grafiğiyle gösterimine histogram adı verilir. Diğer bir ifadeyle tekrarlı sayılardan oluşan elimizdeki verileri, uygulanan işlemlerden sonra önce tabloya, tablodan yararlanarak grafiğe aktarılması yani veri gruplarının grafiğinin dikdörtgen sütunlar halinde gösterilmesine histogram denir. Bir görüntünün histogramı oluşturulurken piksel sayıları ve gri yoğunluğu dikkate alınarak histogram oluşturulur.

### 2.3. Projenin Tasarımı

Projenin tasarımı için araştırılan programlar ve algoritmalar sonucunda projede kullanılacak olan algoritmalar (Median Filter,Edge Detection) tespit edilmiştir. Ve bu algoritmalar sonucunda beyin tümör tespiti algoritması oluşturulmuştur.

İlk olarak anamenü arayüz tasarımı gerçekleştirilmiştir.Ana menüde olması gereken sayfa geçişleri butonlar yardımıyla gerçekleştirilmiştir.En alt kısma ders hakkında ve projeyi gerçekleştiren öğrenci bilgileri eklenmiştir.Yönetim Paneli kısmında Hızlı Beyin Tümörü Tespiti,Detaylı Beyin Tümörü Tespiti ve Çıkış butonları oluşturularak kullanıcının erişebileceği sayfalar belirlenmiştir.



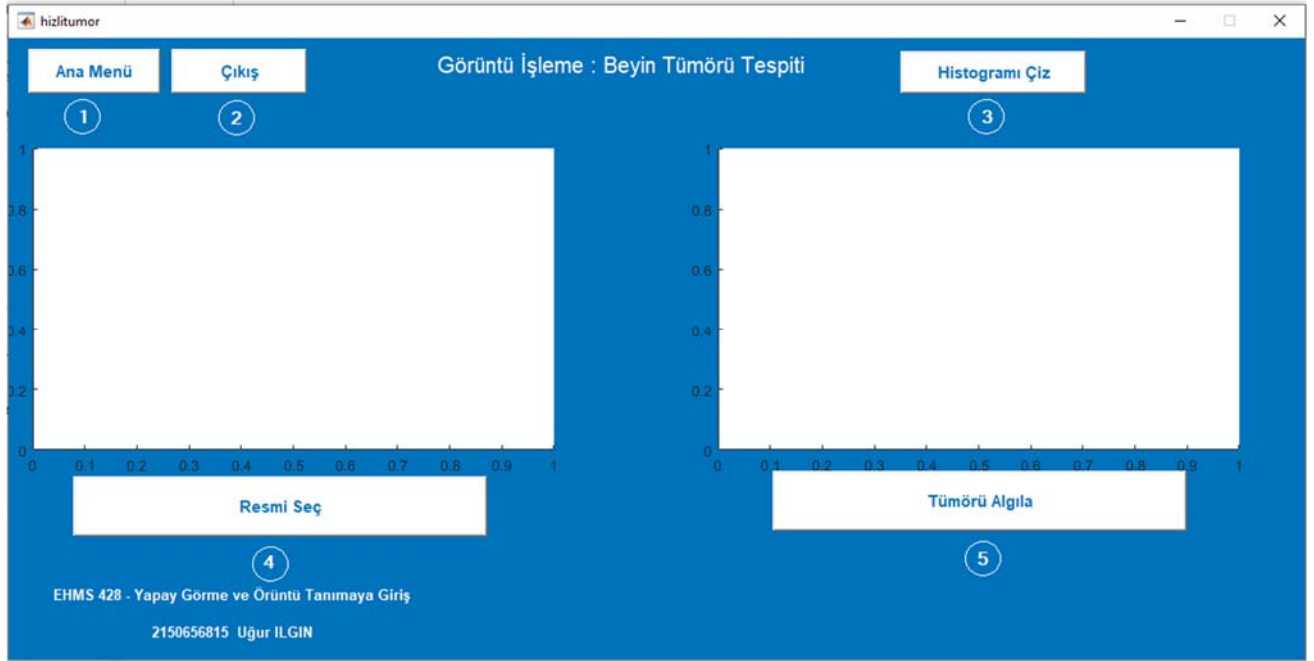
Şekil 2.5. Anamenü Kullanıcı Arayüzü

1-Hızlı Beyin Tümörü Tespiti penceresine geçiş yapar.

2-Detaylı Beyin Tümörü Tespiti penceresine geçiş yapar.

3-Çıkış rogramdan çıkış yapar

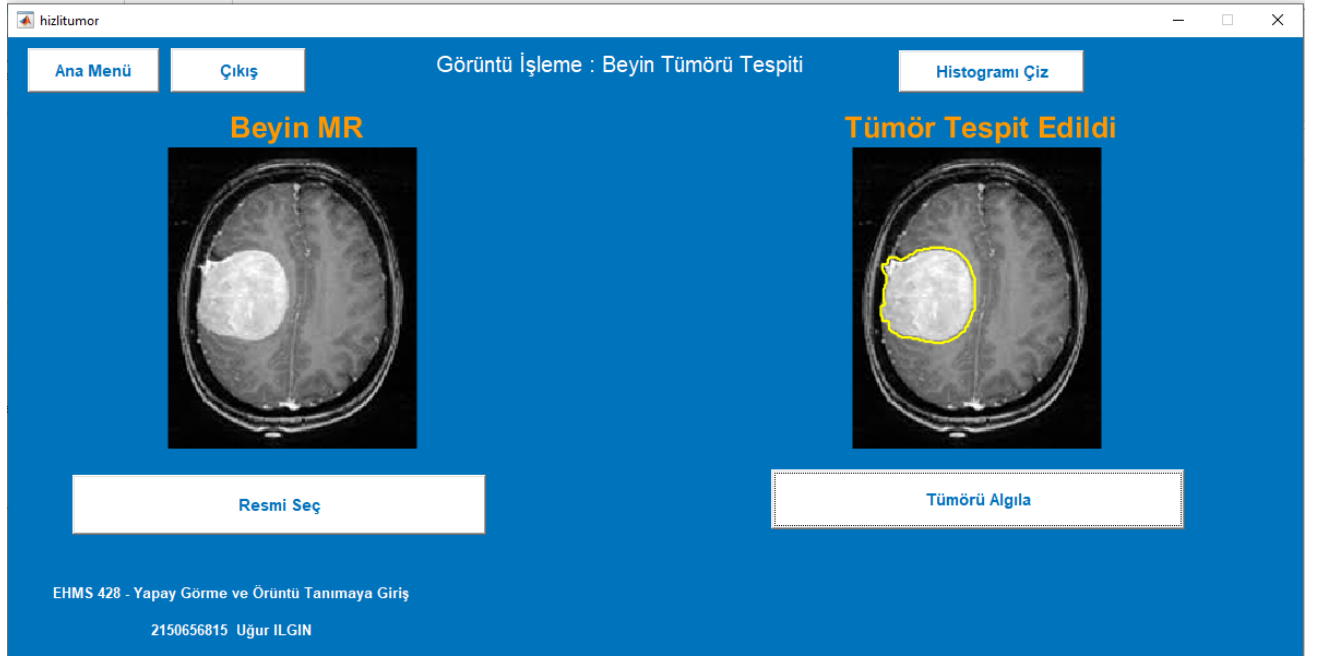
Hızlı Beyin Tümörü Tespiti butonuna tıklanılarak Hızlı Tümör Tespiti yapılabilen pencereye geçilir bu pencerede kullanıcı Resmi Seç butonunu kullanarak bilgisayarından bir MR görüntüsü seçer. Histogramı Çiz butonunu kullanarak seçilen resmin histogram grafiğini görebilir. Tümörü Algıla butonunu kullanarak MR görüntüsündeki tümörü tespit edebilir. Ana Menü butonunu kullanarak ana menüye geri dönüş yapar.Çıkış butonunu kullanarak programdan çıkış yapabilir.



Şekil 2.6. Hızlı Beyin Tümörü Tespiti Kullanıcı Arayüzü

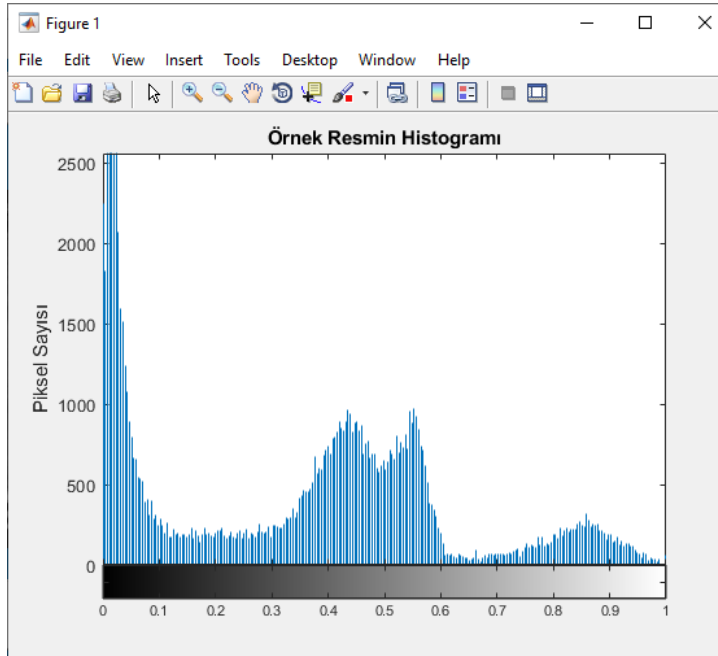
- 1-Anamenü penceresine geçiş yapar
- 2-Programdan çıkış yapar
- 3-Örnek resmin Histogram Grafiğini oluşturur
- 4-Bilgisayardan yeni bir resim seçme işlemini gerçekleştirir
- 5-Örnek MR üzerinden tümörün tespit edilmesini sağlar





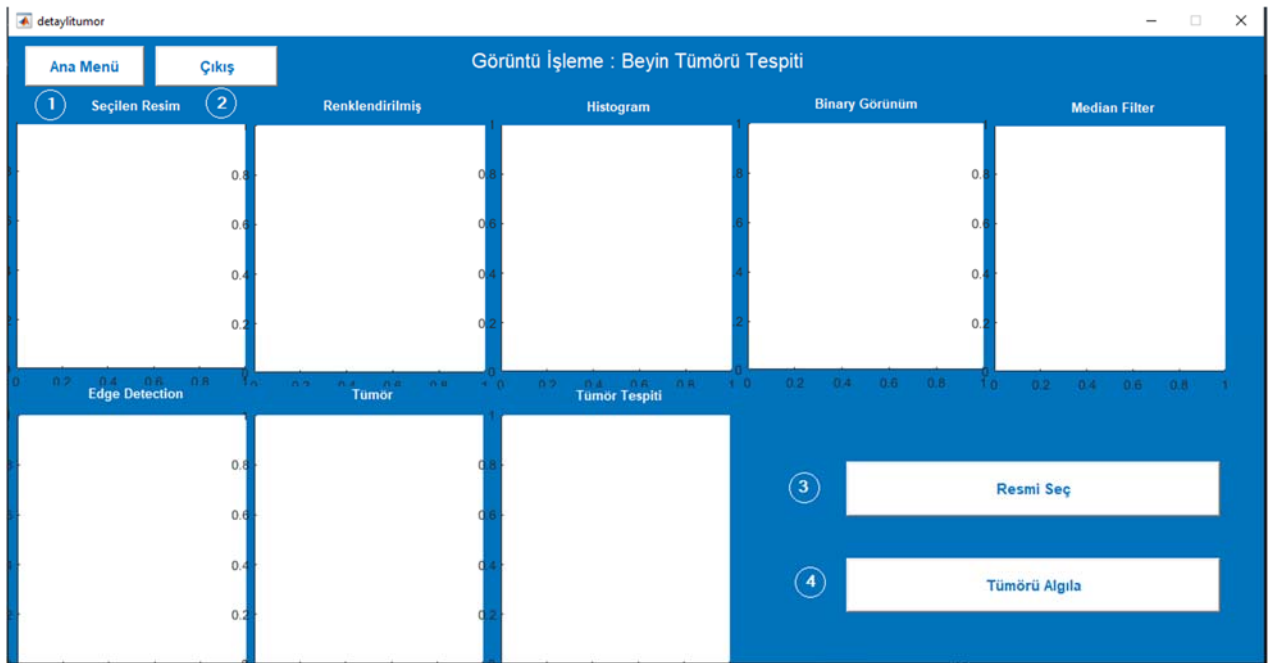
Şekil 2.7. Hızlı Beyin Tümörü Tespiti Tümör Tespiti Gerçekleştirilen Görüntü

Örnek görüntü üzerinden beyin tümörü tespiti yapıldıktan sonra veya incelenecek MR görüntüsü seçildikten sonra Histogramı Çiz butonu kullanılarak Histogram Grafiği oluşturulabilir.



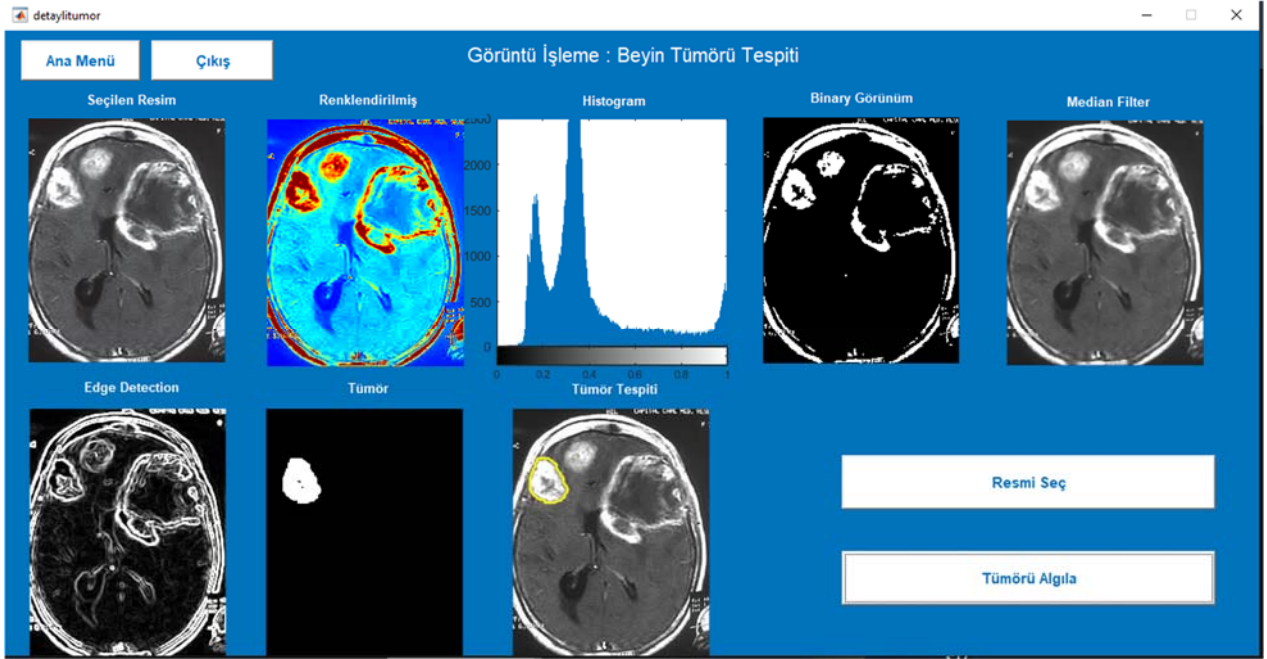
Şekil 2.8. Örnek Resmin Histogramı

Detaylı Beyin Tümörü Tespiti butonuna tıklanılarak Detaylı Beyin Tümörü Tespiti yapılabilen pencereye geçilir bu pencerede kullanıcı Resmi Seç butonunu kullanarak bilgisayarından bir MR görüntüsü seçer.Seçilen Resim ,Renklendirilmiş(Colorized ) Resim ve Histogram ilgili alanlarda otomatik olarak oluşturulur. Tümörü Algıla butonunu kullanarak Resmin Binary Görünümü,Median Filter(Medyan Filtresi) uygulanmış hali,Edge Detection(Kenar Algılama) algoritması uygulanmış hali,Tespit Edilen Tümörün görüntüsü ,Örnek Resim Üzerinde Tespit Edilen Tümörün görüntüsü ilgili alanlarda oluşturulur.Ana Menü butonunu kullanarak ana menüye geri dönüş yapar.Çıkış butonunu kullanarak programdan çıkış yapabilir.



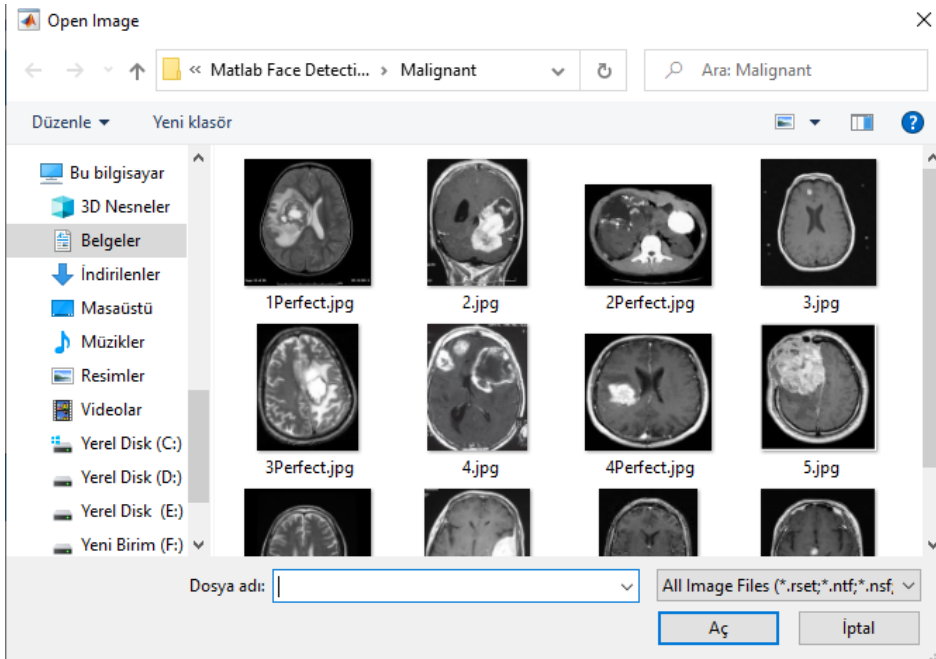
Şekil 1.9. Detaylı Beyin Tümörü Tespiti Kullanıcı Arayüzü

- 1-Anamenü penceresine geçiş yapar
- 2-Programdan çıkış yapar
- 3- Bilgisayardan yeni bir resim seçme işlemini gerçekleştirir.Histogram,Renklendirme algoritmaları uygulanır
- 4-Örnek MR üzerinden Binary Görüntü,Median Filter,Edge Detection algoritmaları uygulanır.Yalnızca Tümör ve Resim üzerinden tespiti yapılmış tümör görüntülenir.

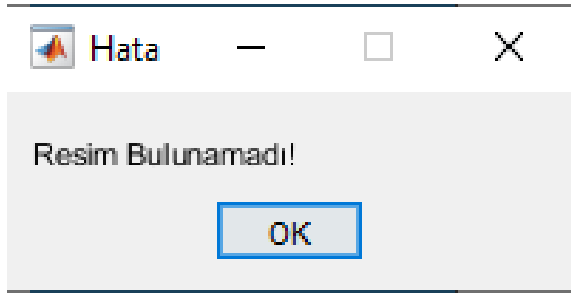


Şekil 2.10. Detaylı Beyin Tümörü Tespiti Tumor Tespiti Yapılan Resim

Resim Seçme Butonuna basılarak açılan pencereden MR görüntüsü seçilmezse program kullanıcıya hata mesajı göstermektedir.



Şekil 2.11. Resim Seçim Penceresi



Şekil 2.12. Hata Mesajı

### 3. TERİMLER

İmgetfile():Herhangi bir uzantıya sahip görüntü (Image) dosyası seçilmesi için Windows Dosya Görüntüleyici penceresini açar.

İmread():Seçilen dosya adresindeki görselin istenilen değikene aktarılmasını sağlar

İm2double(): fonksiyonu giriş datasını double sınıf dataya dönüştürür

İmshow():Görüntünün görüntülenmesini sağlar.

İmhist():Resmin histogramının görüntülenmesini sağlar.

rgb2gray():Resmin renklerini griye çevirir.

size():Resmin boyutunun bulunmasını sağlayan fonksiyondur.

medfilt2():Resme median filter uygular.

sqrt():Karekök alma işlemini yapan fonksiyondur.

İm2bw():Resmi Siyah Beyaza çevirir.

bwlabel(): işlemi ile bağlı bileşenlerin bilgileri bir değışkene atanır.

regionprops(): ile bölgelerin özellikleri çıkarılır.

max():Maximum değerini bulur.

find():Verilen değeri bulur.

Strel(): Strel, morfolojik işlemlerde kullanılan yapısal filtre elemanıdır. Morfolojik işlemleri hangi şekil ve parametrelerle uygulanacağı strel ile belirlenir.

İmdilate(): Açma işlemini yapar.

Bwboundaries(): binary modda bölgelerin sınırlarını belirler

## KAYNAKLAR

Hmc.Edu, [http://fourier.eng.hmc.edu/e161/lectures/smooth\\_sharpen/node2.html](http://fourier.eng.hmc.edu/e161/lectures/smooth_sharpen/node2.html), Median Filter

İbrahim Çayiroğlu, [www.ibrahimcayiroglu.com](http://www.ibrahimcayiroglu.com) ,Görüntü İşleme Ders Notları,5.Hafta, Median Filter

Edge Detection , <https://www.imageprocessing.com/2011/12/sobel-edge-detection.html>, Sobel Edge Detection

Wikipedia , <https://tr.wikipedia.org/wiki/Histogram>,Histogram

MathWork,<https://www.mathworks.com/help/matlab/ref/matlab.graphics.chart.primitive.histogram.html>,Histogram Kullanımı

Alec Jacobson, <https://www.alecjacobson.com/weblog/?p=1655>,Colorized

Youtube, <https://www.youtube.com/watch?v=UeFRo7uALhM&t=320s>, Brain Tumor Detection

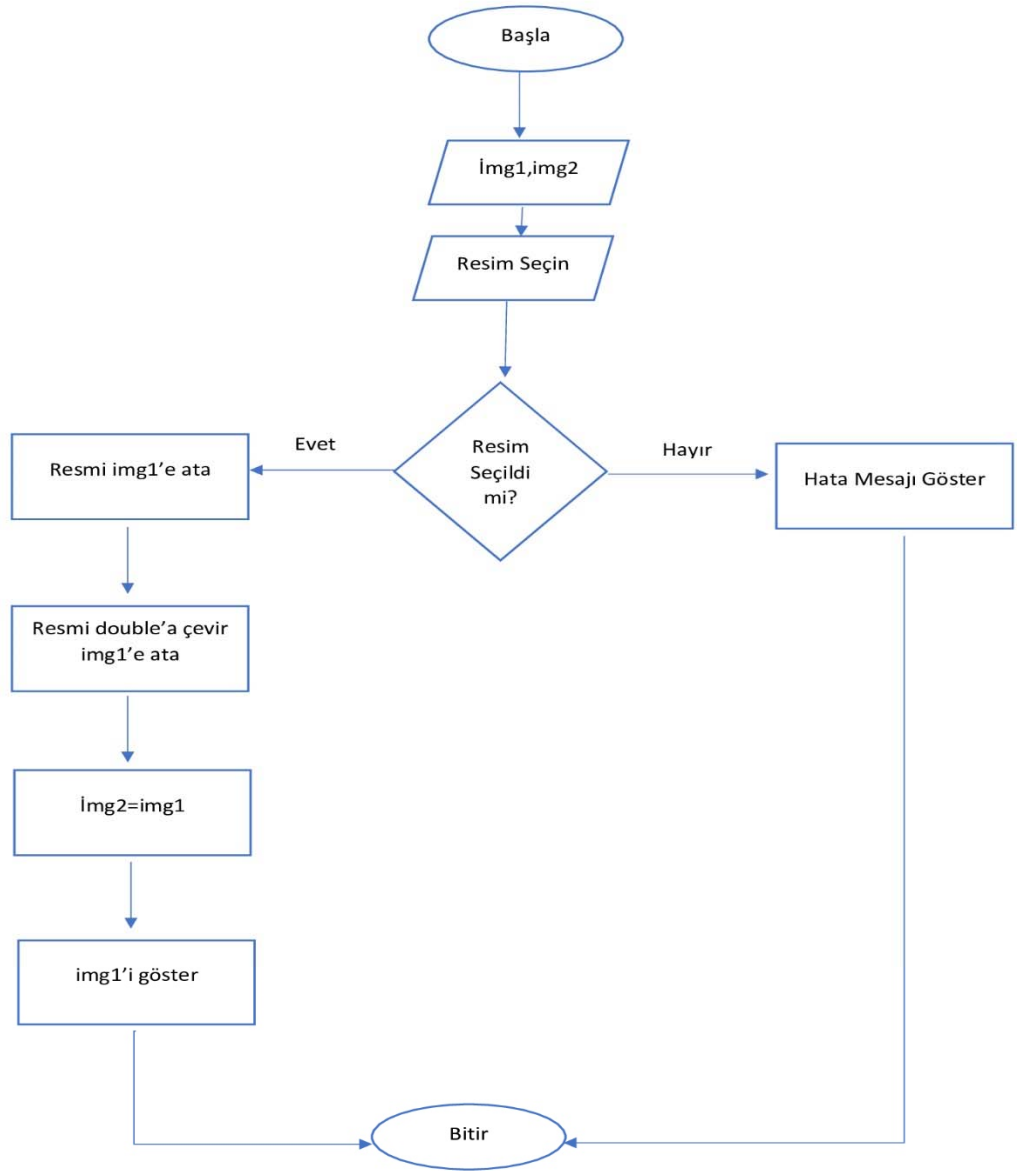
ElektrikPort,<https://www.elektrikport.com/teknik-kutuphane/matlab-ile-goruntu-isleme-uygulamasi-elektrikport-akademi/8434#ad-image-0>,Terimler

ElektrikPort,<https://www.elektrikport.com/universite/matlab-ile-goruntu-isleme-2-elektrikport-akademi/8197#ad-image-0>,Terimler

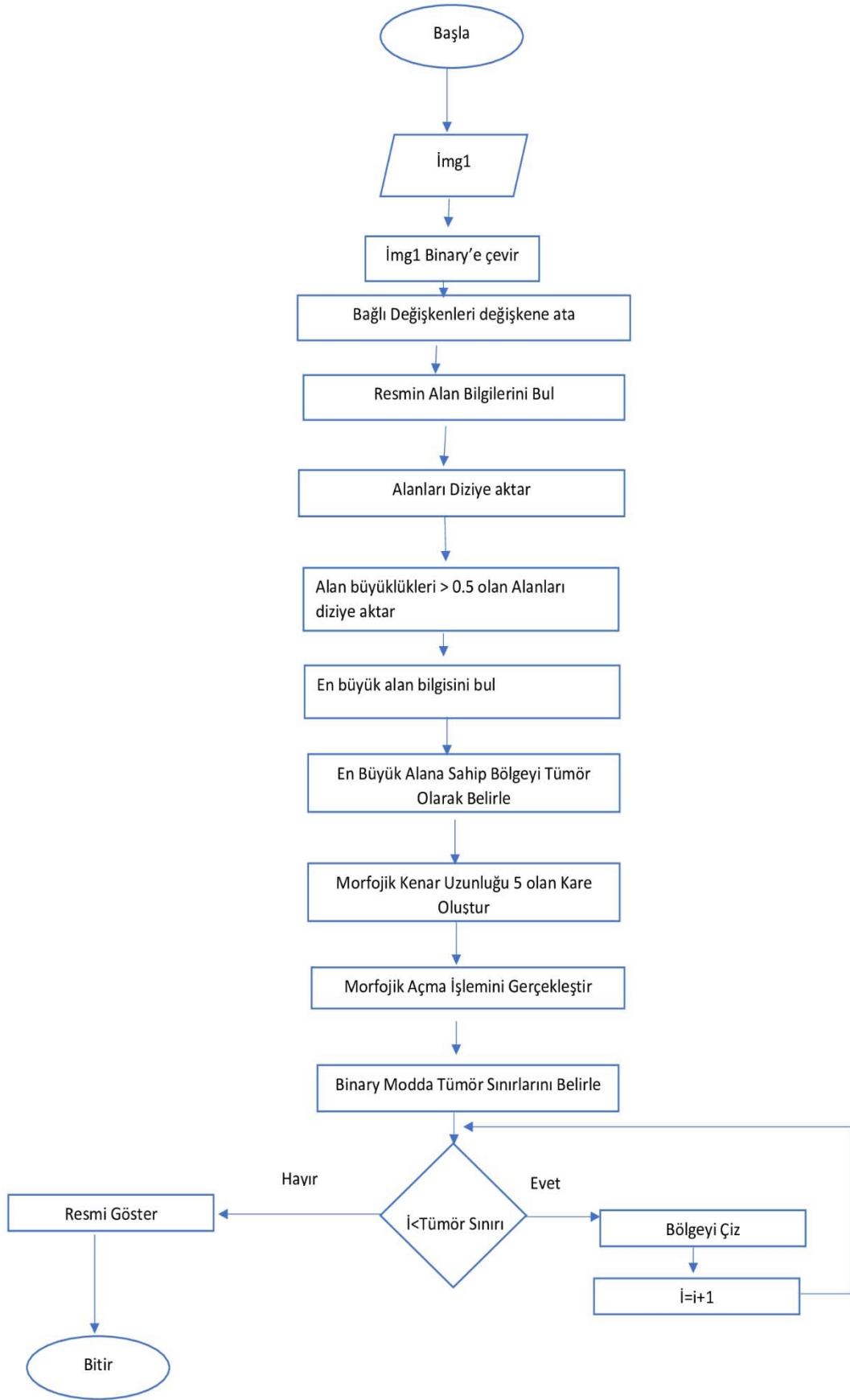
FıratÜniversitesi,<http://bil.muh.firat.edu.tr/sites/bil.muh.firat.edu.tr/files/Görüntü%20İşleme%20Uygulamaları.pdf>,bwboundaries komutu

## EKLER

### 4. ALGORİTMALAR

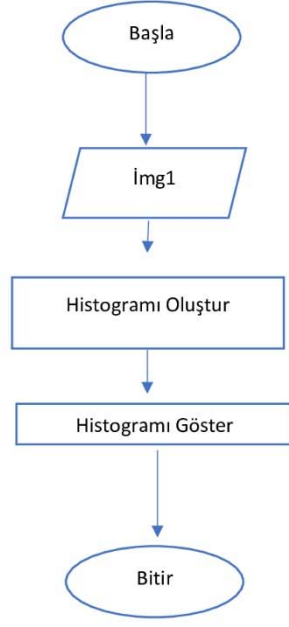


Şekil 4.1. Hızlı Beyin Tümörü Tespiti Resim Seç Butonu Algoritması

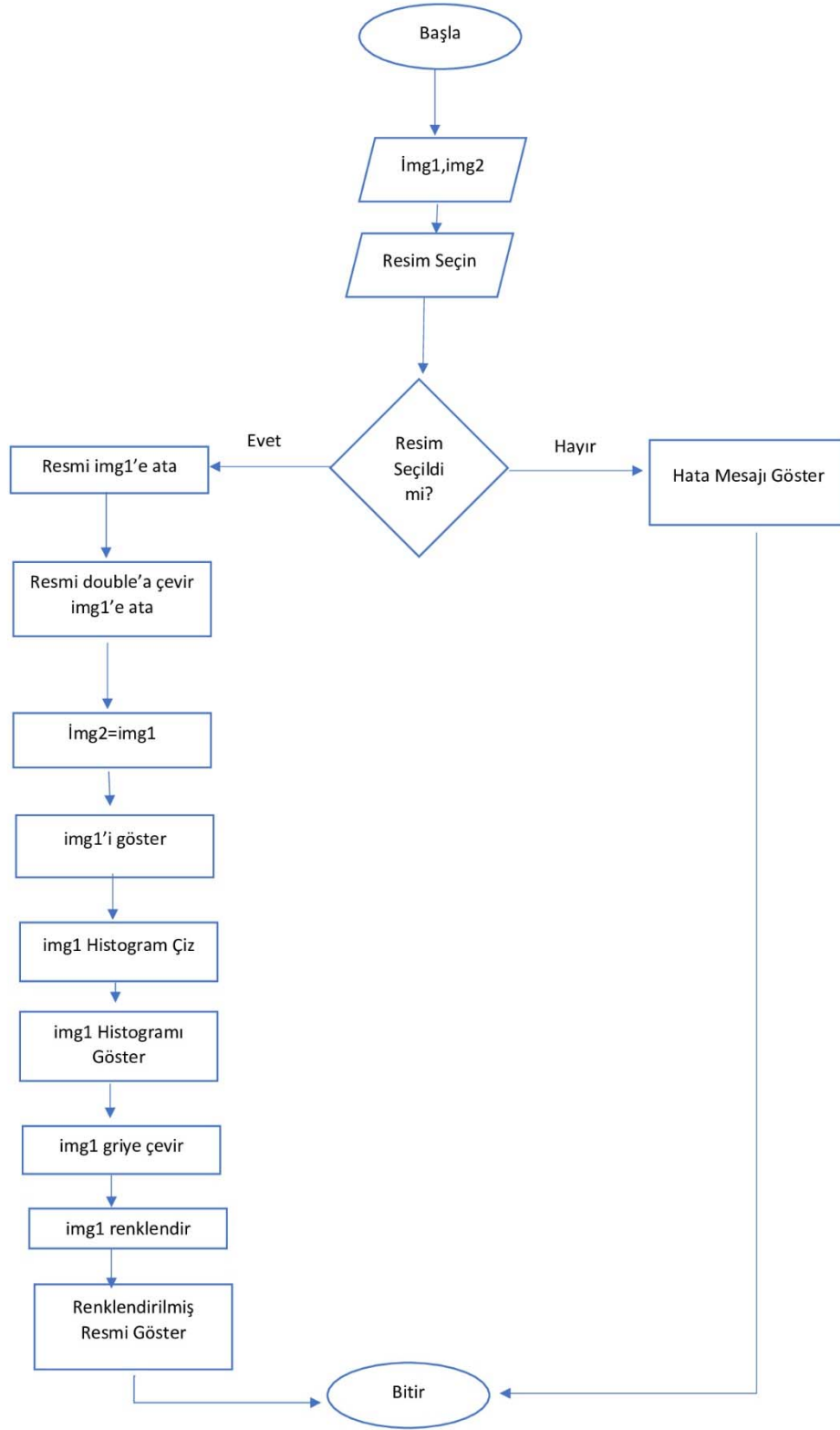


Şekil 4.2. Hızlı Beyin Tümörü Tespiti Tümör Bul Butonu Algoritması

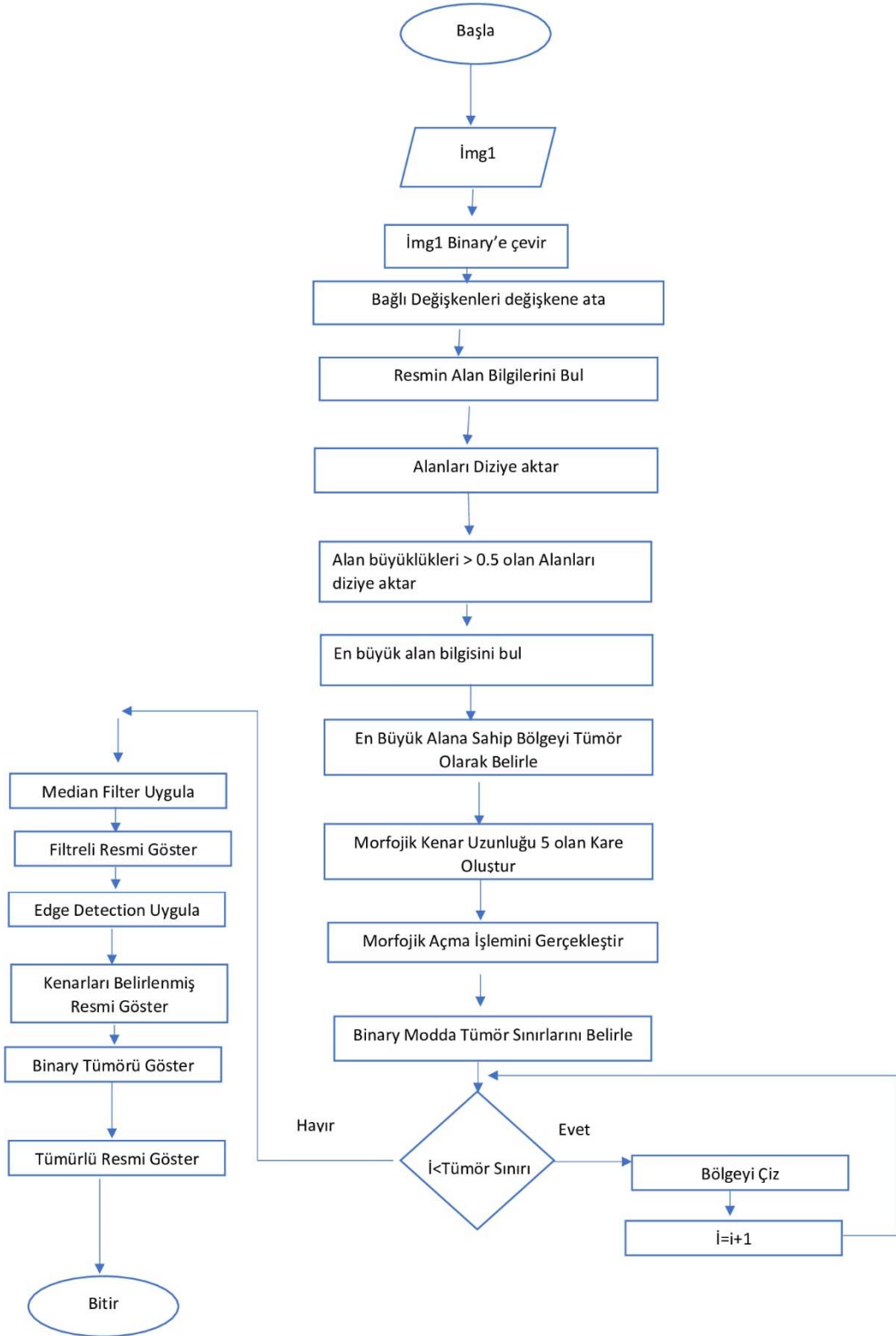




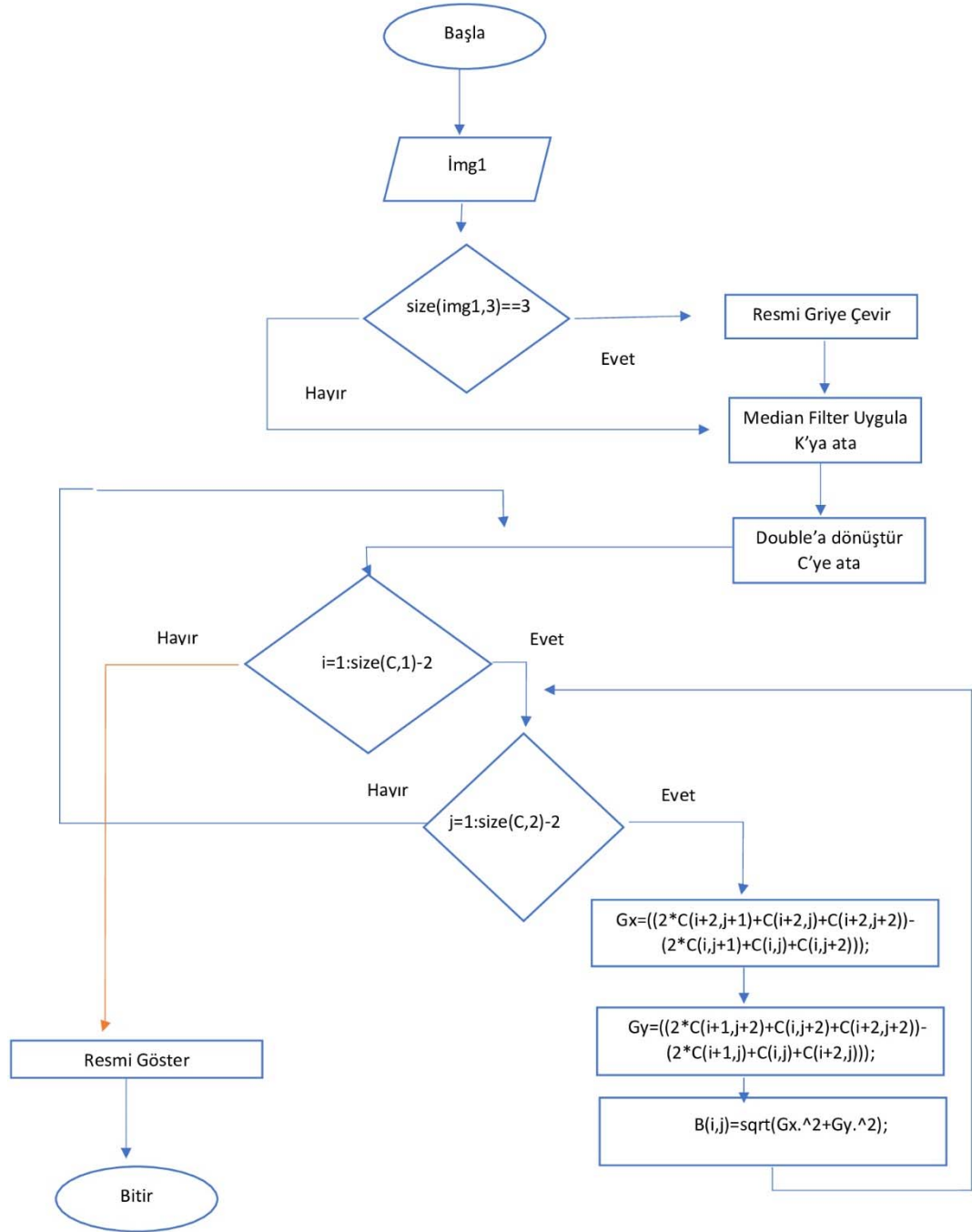
Şekil 4.3. Hızlı Beyin Tümörü Tespiti Histogram Çiz Butonu Algoritması



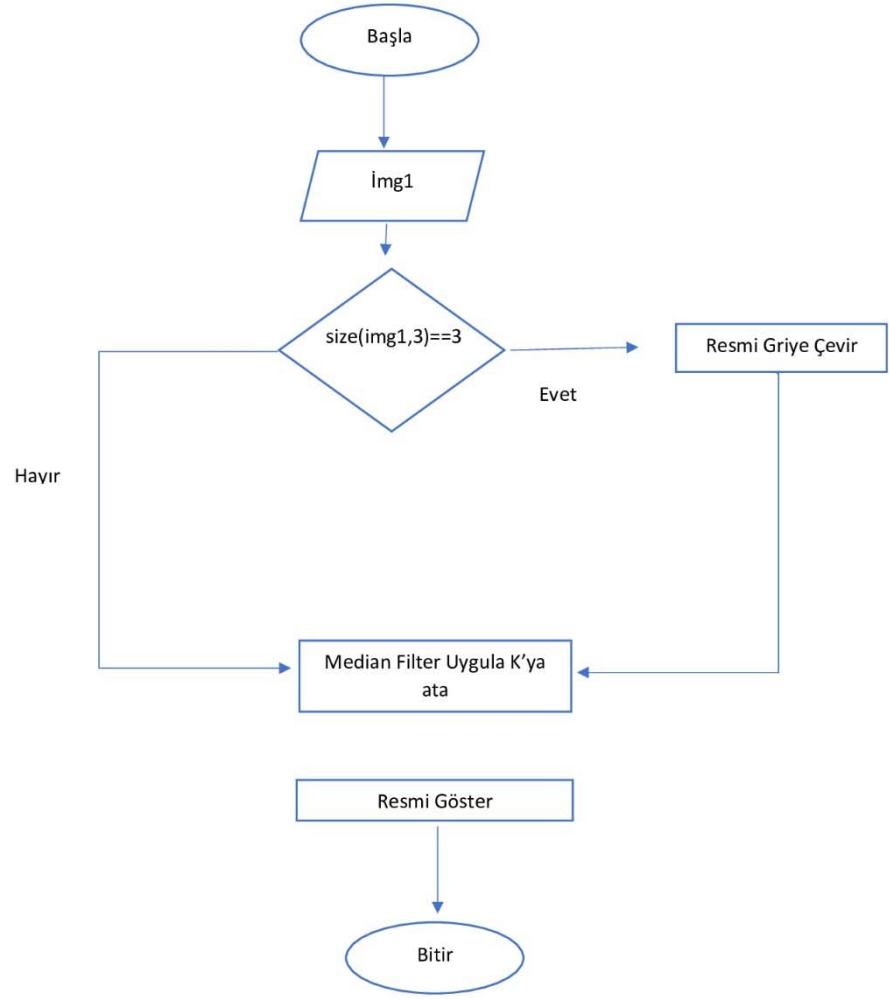
Şekil 4.4. Detaylı Beyin Tümörü Tespiti Resim Seç Butonu Algoritması



Şekil 4.5. Detaylı Beyin Tümörü Tespiti Tümör Bul Butonu Algoritması



Şekil 4.6. Detaylı Beyin Tümörü Tespiti Edge Detection Algoritması



Şekil 4.7. Detaylı Beyin Tümörü Tespiti Median Filter Algoritması

## 5. KODLAR

### 5.1. Mainmenu.m

```
function varargout = mainmenu(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',   @mainmenu_OpeningFcn, ...
                  'gui_OutputFcn',    @mainmenu_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function mainmenu_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
axes(handles.axes1);
imshow('C:\Users\futbo\OneDrive\Belgeler\Matlab Face Detection\face.jpg');
guidata(hObject, handles);

function varargout = mainmenu_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function hizliTumor_Callback(hObject, eventdata, handles)
hizlitumor();
closereq();

function detayliTumor_Callback(hObject, eventdata, handles)
detaylitumor();
closereq();

function Cikis_Callback(hObject, eventdata, handles)
closereq();
```

## 5.2. Hizlitumor.m

```
function varargout = hizlitumor(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @hizlitumor_OpeningFcn, ...
                  'gui_OutputFcn',  @hizlitumor_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function hizlitumor_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);

function varargout = hizlitumor_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function anamenu_Callback(hObject, eventdata, handles)
mainmenu();
closereq();

function secim_Callback(hObject, eventdata, handles)
global img1 img2
[path,nofile]=imgetfile();
if nofile
    msgbox(sprintf('Resim Bulunamad!'),'Hata','Warning');
    return
end
img1=imread(path);
img1=im2double(img1);
img2=img1;
axes(handles.axes1);
imshow(img1);
title('\fontsize{20}\color{rgb}{0.996,0.592,0.0} Beyin MR')

function Cikis_Callback(hObject, eventdata, handles)
closereq();
```

```

function tumor_Callback(hObject, eventdata, handles)
global img1
axes(handles.axes3);
bw=im2bw(img1, 0.7);
label=bwlabel(bw);

stats=regionprops(label,'solidity','Area');
density=[stats.Solidity];
area=[stats.Area];
high_dense_area=density>0.5;
max_area = max(area(high_dense_area));
tumor_label=find(area == max_area);
tumor=ismember(label,tumor_label);

se=strel('square',5);
tumor = imdilate(tumor,se);

Bound=bwboundaries(tumor,'noholes');

imshow(img1);
hold on

for i=1:length(Bound)
    plot(Bound{i}(:,2), Bound{i}(:,1),'y','linewidth',1.75)
end
title('\fontsize{20}\color{rgb}{0.996,0.592,0.0} Tmespit Edildi')
hold off
axes(handles.axes3);

function histogram_Callback(hObject, eventdata, handles)
global img1
figure;
imhist(img1);
title("nek Resmin Histogram");
xlabel("Gri Oran");
ylabel("Piksel Says");

```



### 5.3. Detaylitumor.m

```
function varargout = detaylitumor(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @detaylitumor_OpeningFcn, ...
                  'gui_OutputFcn',  @detaylitumor_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function detaylitumor_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);

function varargout = detaylitumor_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function anamenu_Callback(hObject, eventdata, handles)
mainmenu();
closereq();

function secim_Callback(hObject, eventdata, handles)
global img1 img2
[path,nofile]=imgetfile();
if nofile
    msgbox(sprintf('Resim Bulunamad!'),'Hata','Warning');
    return
end
img1=imread(path);
img1=im2double(img1);
img2=img1;
axes(handles.axes1);
imshow(img1);
axes(handles.axes1);

axes(handles.axes6);
imhist(img1);
axes(handles.axes6);
imggri=rgb2gray(img1);

axes(handles.axes5);
imshow(imggri,'Colormap','jet(255));
axes(handles.axes5);

function Cikis_Callback(hObject, eventdata, handles)
```

```

closereq();

function tumor_Callback(hObject, eventdata, handles)
global img1
% --- Medyan Filtresi.

axes(handles.axes7);
if size(img1,3)==3
    img1=rgb2gray(img1);
    Seg=medfilt2(img1);
end
K=medfilt2(img1);
axes(handles.axes7);
imshow(K);
% --- KoseAlglama.

axes(handles.axes8);
C=double(K);
for i=1:size(C,1)-2
    for j=1:size(C,2)-2
        Gx=((2*C(i+2,j+1)+C(i+2,j)+C(i+2,j+2))-
(2*C(i,j+1)+C(i,j)+C(i,j+2)));
        Gy=((2*C(i+1,j+2)+C(i,j+2)+C(i+2,j+2))-
(2*C(i+1,j)+C(i,j)+C(i+2,j)));
        B(i,j)=sqrt(Gx.^2+Gy.^2);
    end
end
axes(handles.axes8);
imshow(B);

% --- Tumoriglama .

axes(handles.axes10);

bw=im2bw(img1, 0.7);
label=bwlabel(bw);

stats=regionprops(label,'solidity','Area');
density=[stats.Solidity];
area=[stats.Area];
high_dense_area=density>0.5;
max_area = max(area(high_dense_area));
tumor_label=find(area == max_area);
tumor=ismember(label,tumor_label);

se=strel('square',5);
tumor = imdilate(tumor,se);

Bound=bwboundaries(tumor,'noholes');

imshow(img1);
hold on

for i=1:length(Bound)
    plot(Bound{i}(:,2), Bound{i}(:,1),'y','linewidth',1.75)
end
hold off
axes(handles.axes10);

```

```
axes(handles.axes9);  
imshow(tumor,[]);  
axes(handles.axes9);
```

```
axes(handles.axes3);  
imshow(bw);  
axes(handles.axes3);
```