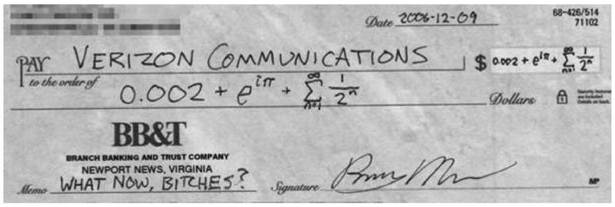
Activity 5: Feature Extraction Part 1 of 3: Image Segmentation

Submitted by: Jonabel Eleanor B. Baldres

Section: App Physics 157 WFY FX 2

Image Segmentation on Grayscale Images

```
I = imread('cropped_grayscale_check.jpg');
imshow(I);
```



[redacted]

Date 2006-12-09

68-426/514
71102

PAY to the order of VERIZON COMMUNICATIONS $ $0.002 + e^{i\pi} + \sum_{n=1}^{\infty} \frac{1}{2^n}$

$0.002 + e^{i\pi} + \sum_{n=1}^{\infty} \frac{1}{2^n}$ Dollars

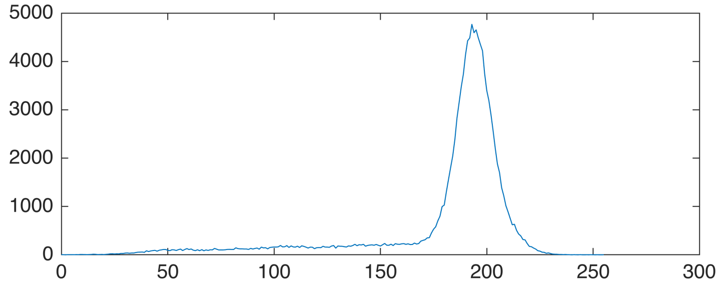
BB&T
BRANCH BANKING AND TRUST COMPANY
NEWPORT NEWS, VIRGINIA

Memo WHAT NOW, BITCHES?

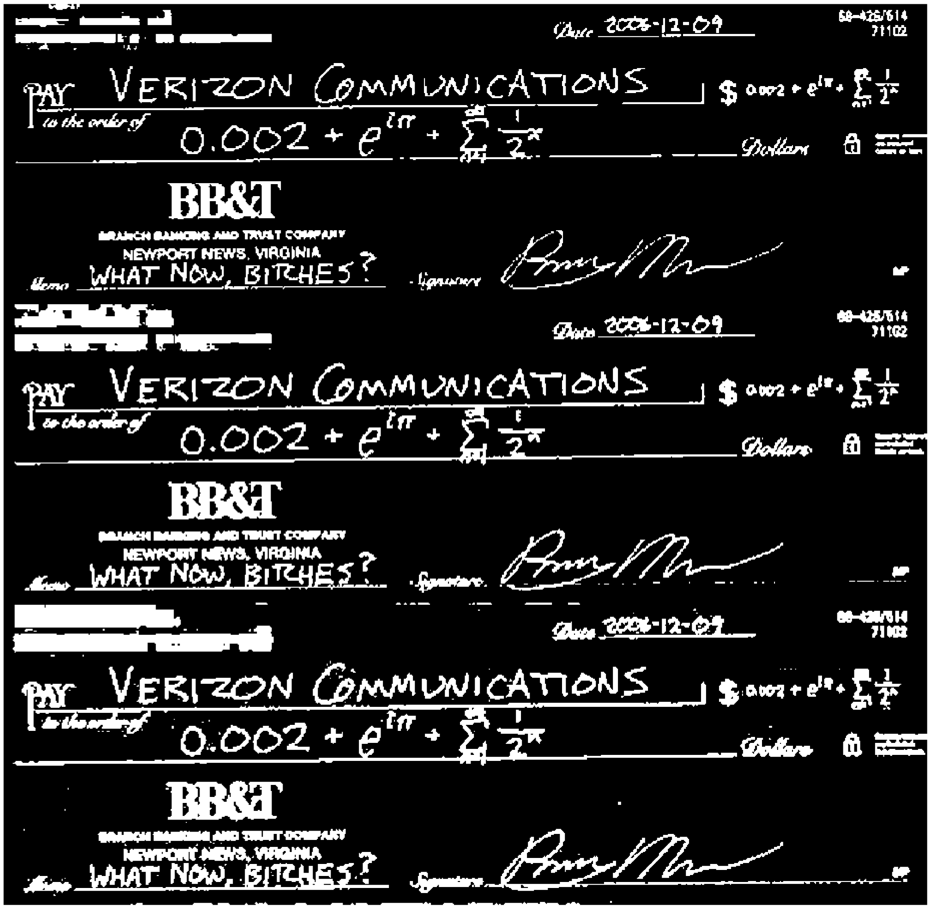
Signature

MP

```
[count, cells] = imhist(I,256);
plot(cells, count);
```



```
montage({threshold(125), threshold(150), threshold(170)}, 'size', [3 1])
saveas(gcf, "checque.png");
```



[redacted]

Date 2006-12-09

68-426/514
71102

PAY to the order of VERIZON COMMUNICATIONS $ $0.002 + e^{i\pi} + \sum_{n=1}^{\infty} \frac{1}{2^n}$

$0.002 + e^{i\pi} + \sum_{n=1}^{\infty} \frac{1}{2^n}$ Dollars

BB&T
BRANCH BANKING AND TRUST COMPANY
NEWPORT NEWS, VIRGINIA

Memo WHAT NOW, BITCHES? Signature

[redacted]

Date 2006-12-09

68-426/514
71102

PAY to the order of VERIZON COMMUNICATIONS $ $0.002 + e^{i\pi} + \sum_{n=1}^{\infty} \frac{1}{2^n}$

$0.002 + e^{i\pi} + \sum_{n=1}^{\infty} \frac{1}{2^n}$ Dollars

BB&T
BRANCH BANKING AND TRUST COMPANY
NEWPORT NEWS, VIRGINIA

Memo WHAT NOW, BITCHES? Signature

[redacted]

Date 2006-12-09

68-426/514
71102

PAY to the order of VERIZON COMMUNICATIONS $ $0.002 + e^{i\pi} + \sum_{n=1}^{\infty} \frac{1}{2^n}$

$0.002 + e^{i\pi} + \sum_{n=1}^{\infty} \frac{1}{2^n}$ Dollars

BB&T
BRANCH BANKING AND TRUST COMPANY
NEWPORT NEWS, VIRGINIA

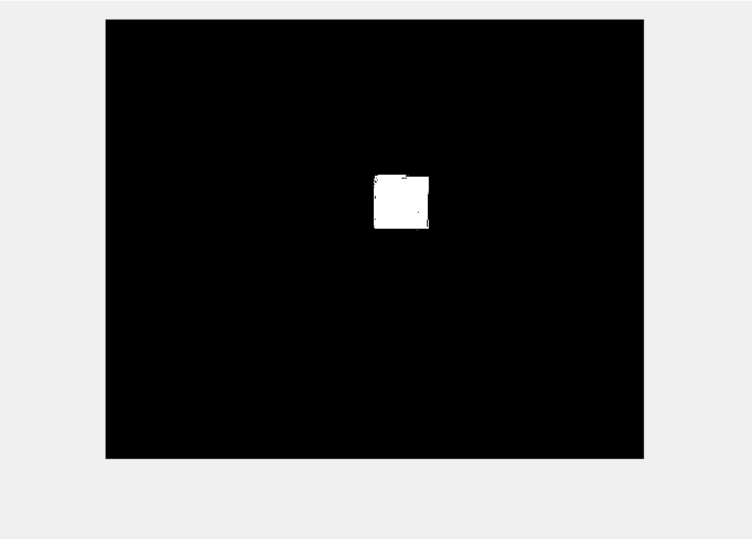
Memo WHAT NOW, BITCHES? Signature

Images used

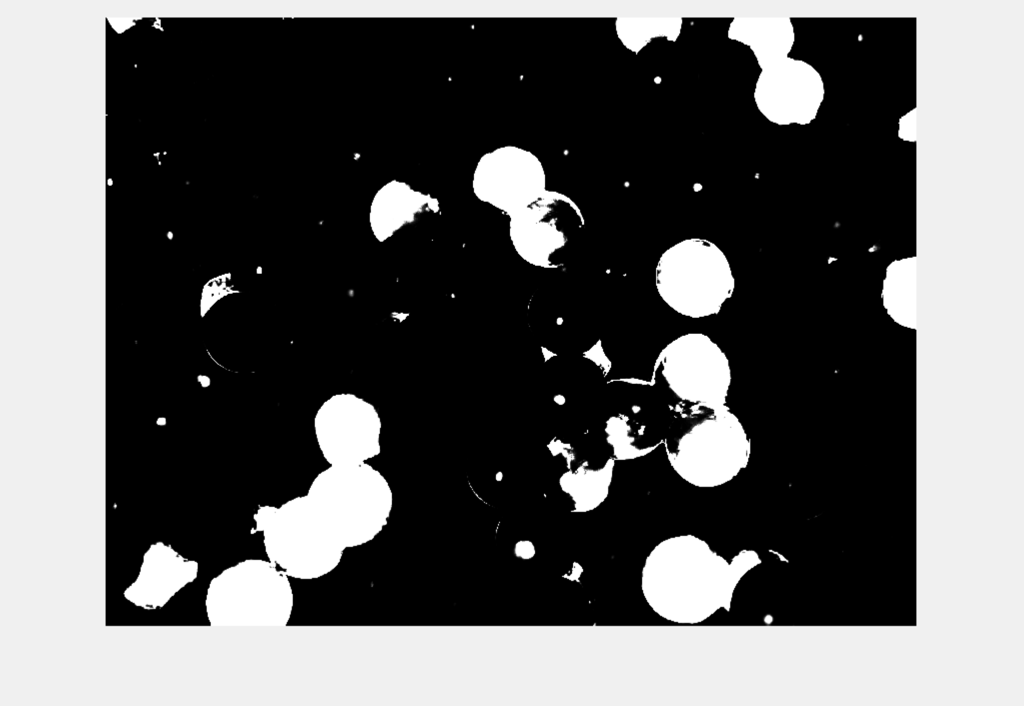
```
nips = imread('nips.jpg');
waddles =imread('waddles.jpg');
olive = imread('olive.jpeg');
barf = imread('barf.jpg');
macbeth = imread("Gretag-Macbeth_ColorChecker.jpg");
mango = imread("mango.jpg");
cells = imread("cells.jpg");
```

Parametric Probability Distribution Estimation

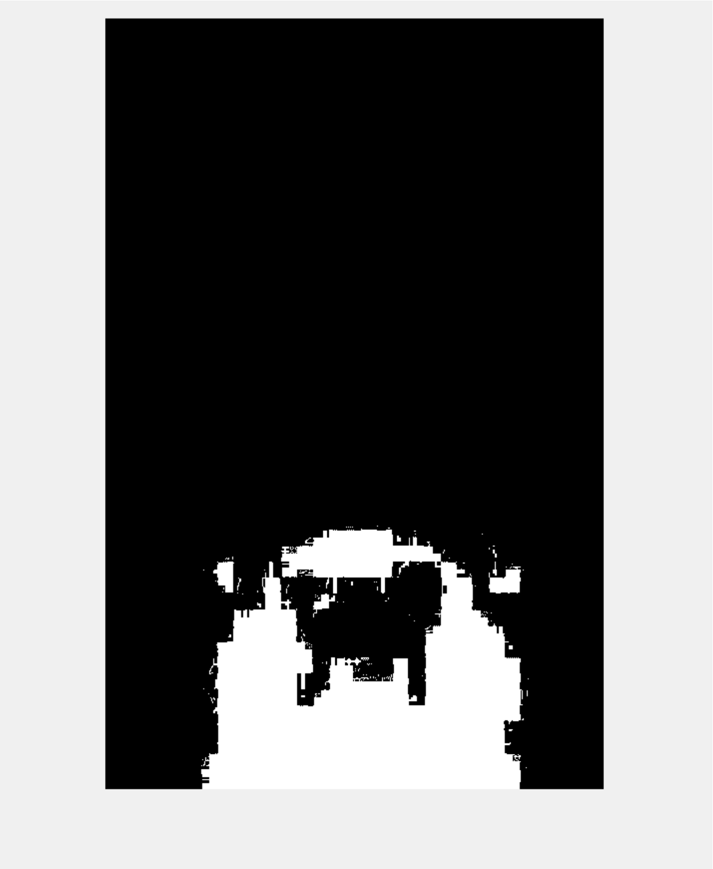
```
macbeth_parametric = parametric("Gretag-Macbeth_ColorChecker.jpg");
saveas(gcf, "macbeth_parametric.png");
```



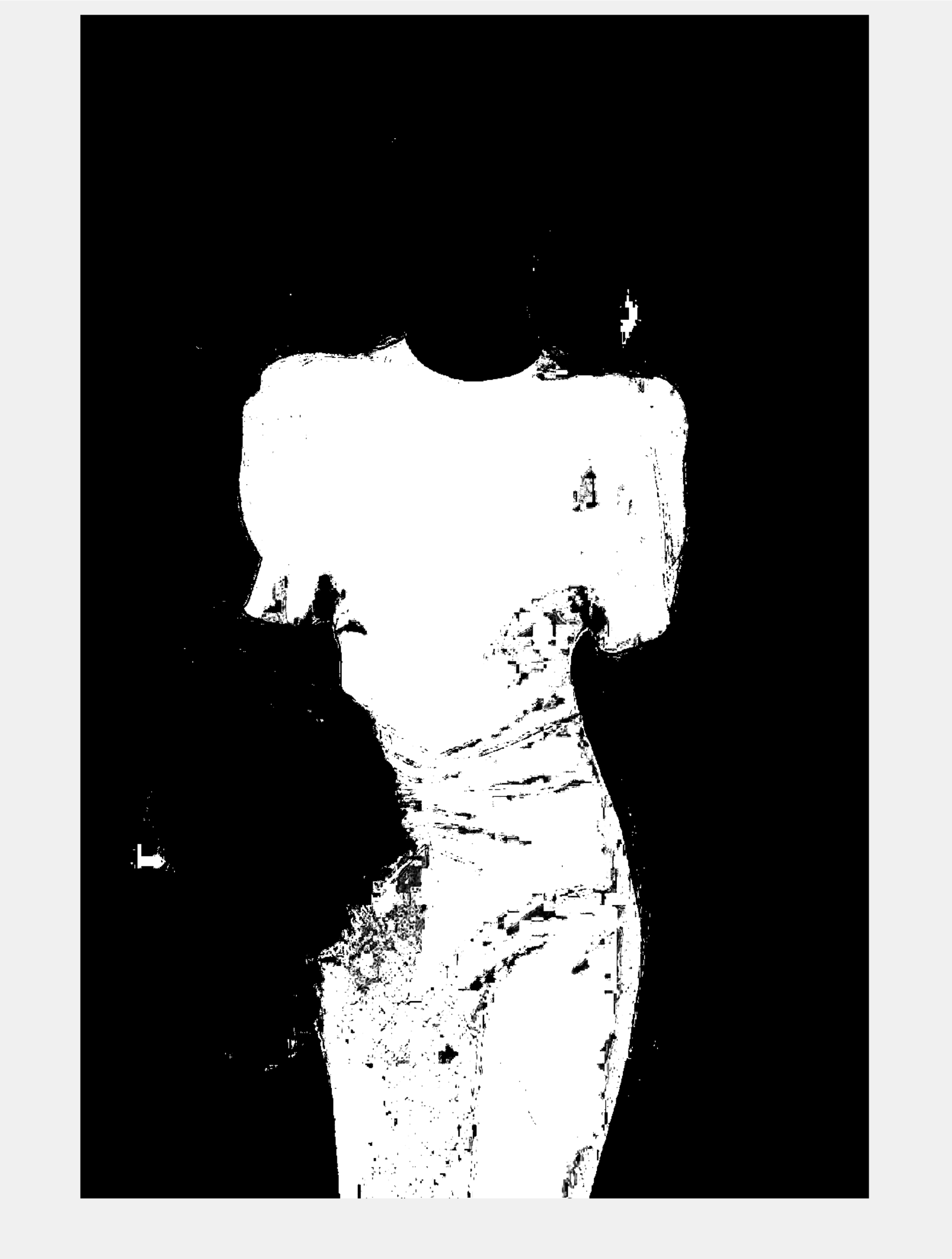
```
nips_parametric = parametric("nips.jpg");
saveas(gcf, "nips_parametric.png");
```



```
waddles_parametric = parametric("waddles.jpg");
saveas(gcf, "waddles_parametric.png");
```

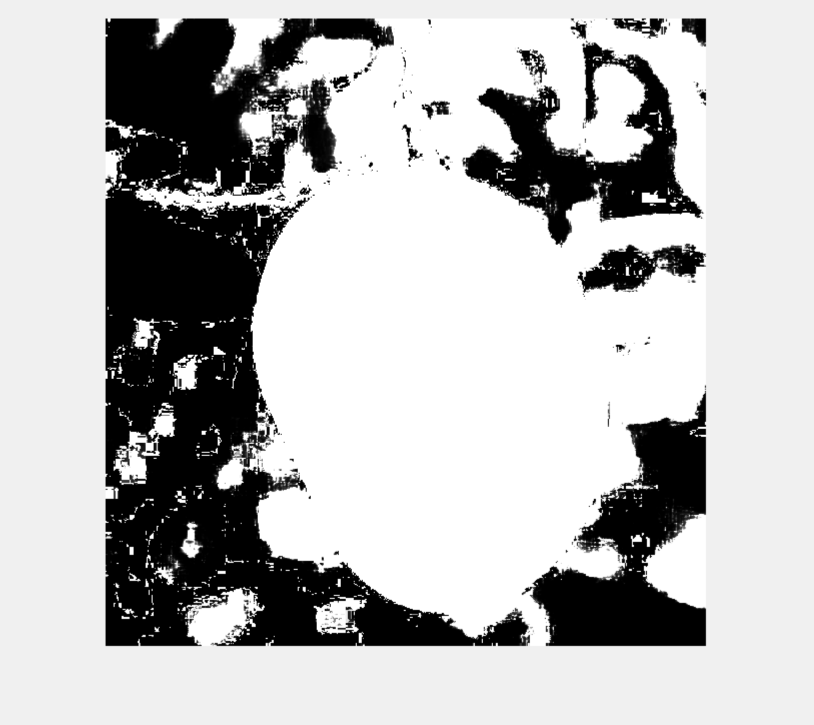


```
olive_parametric = parametric("olive.jpeg");
saveas(gcf, "olive_parametric.png");
```

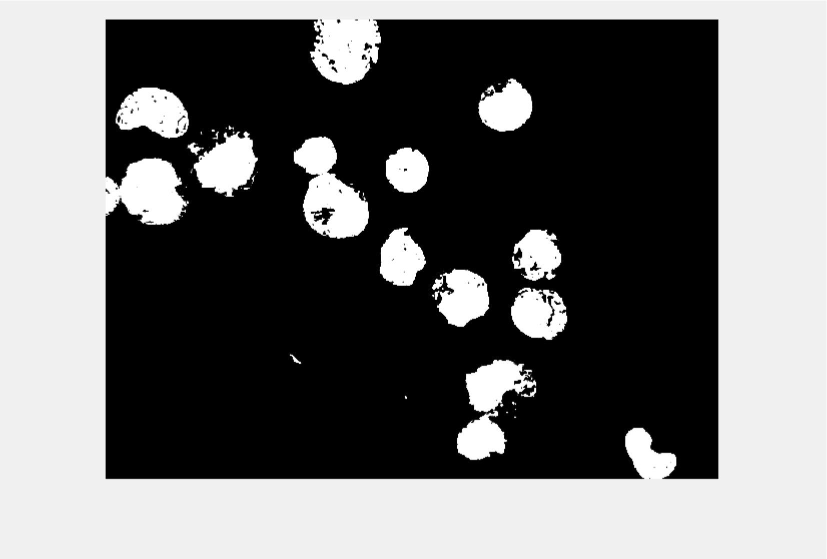


```
barf_parametric = parametric("barf.jpg");
saveas(gcf, "barf_parametric.png");
```

```
mango_parametric = parametric("mango.jpg");
saveas(gcf, "mango_parametric.png");
```

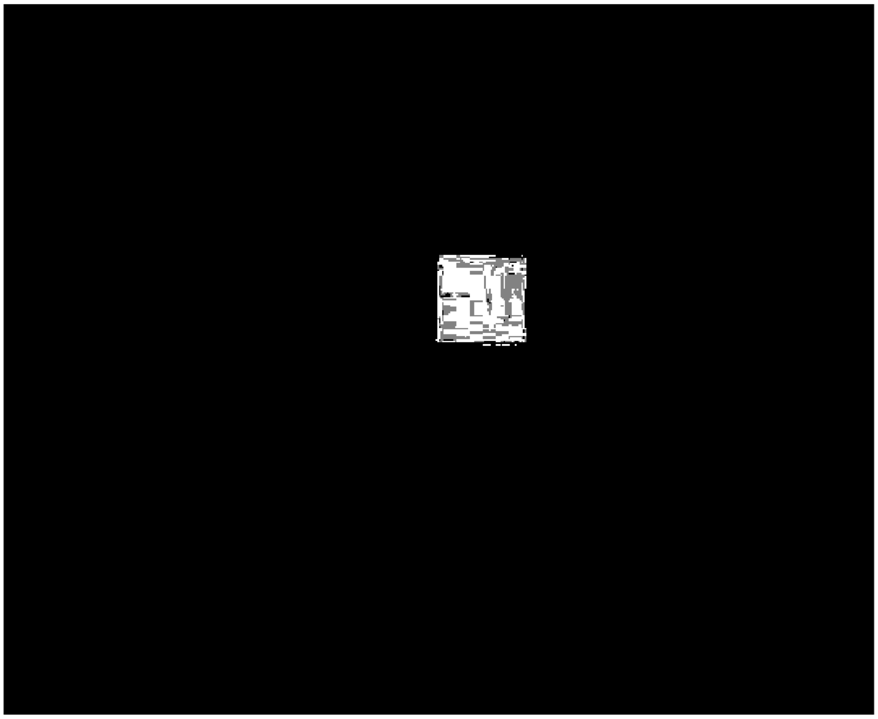


```
cells_parametric = parametric("cells.jpg");
saveas(gcf, "cells_parametric.png");
```

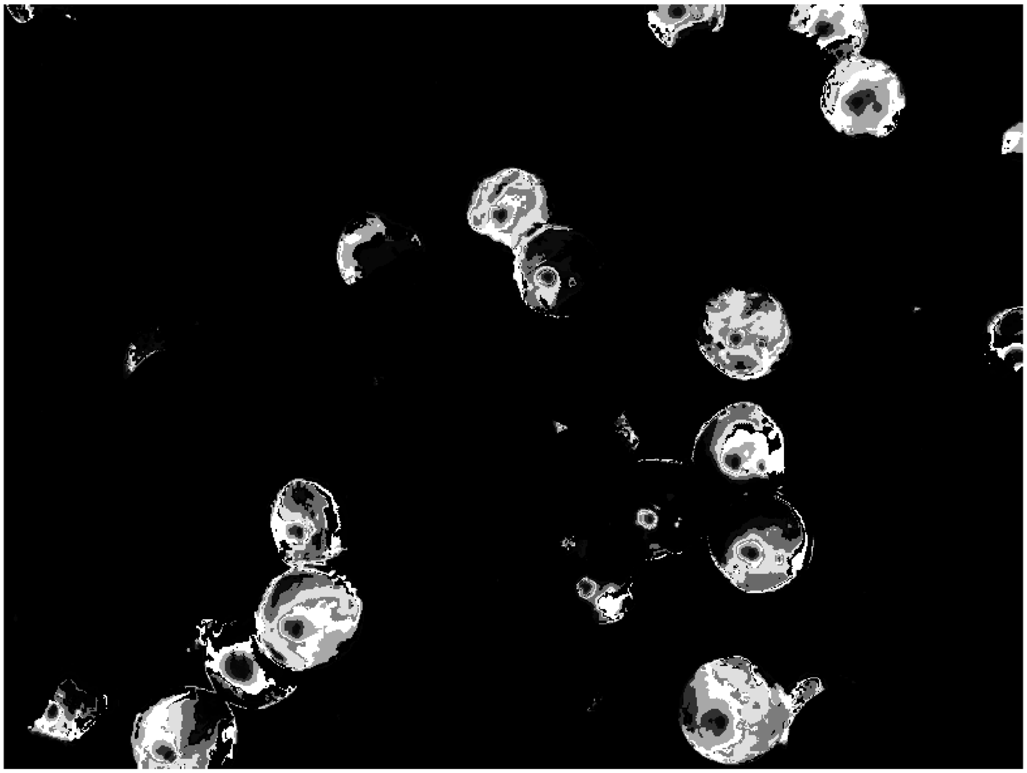


Nonparametric Probability Distribution Estimation

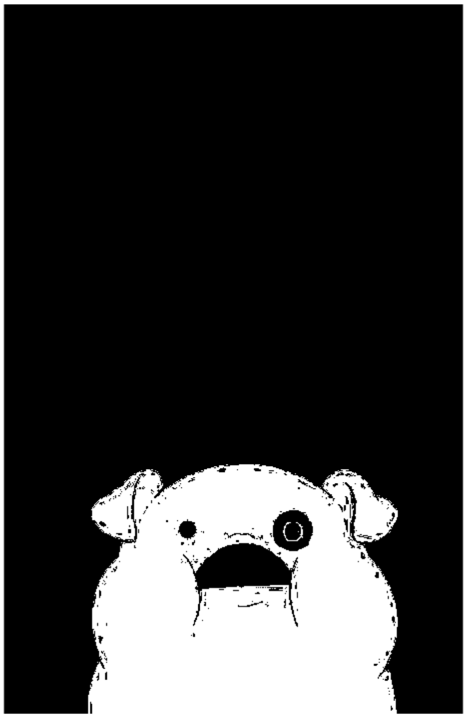
```
%32
macbeth_nonparametric_32 = nonparametric("Gretag-
Macbeth_ColorChecker.jpg",32);
```



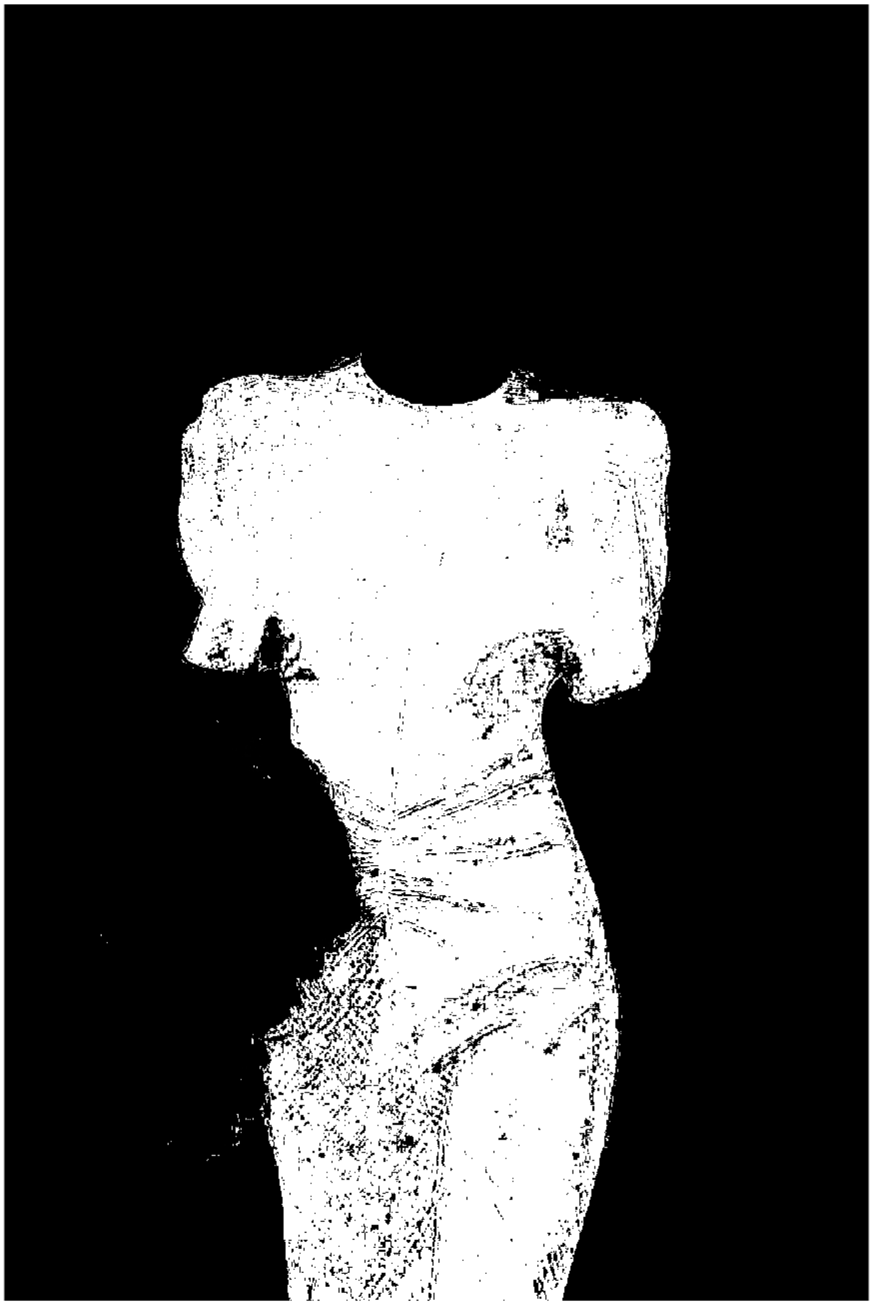
```
nips_nonparametric_32 = nonparametric("nips.jpg",32);
```



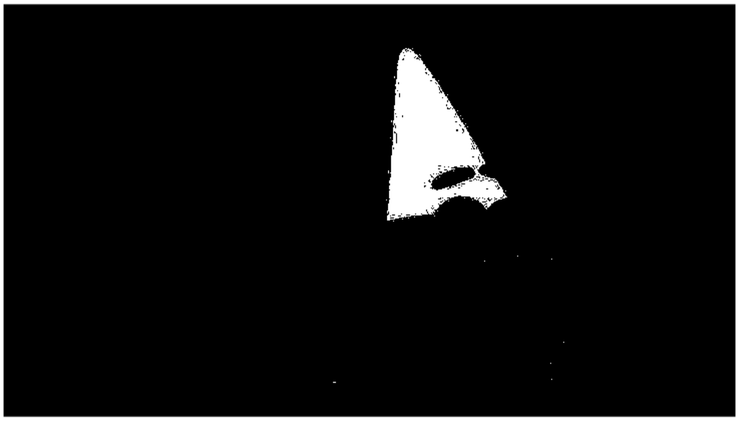
```
waddles_nonparametric_32 = nonparametric("waddles.jpg",32);
```



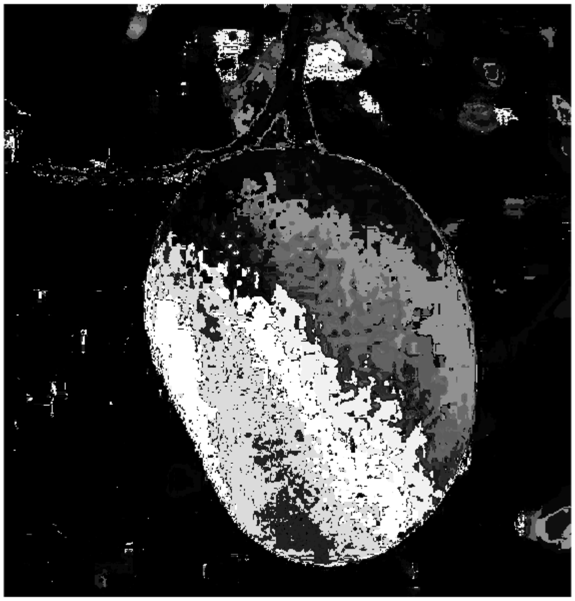
```
olive_nonparametric_32 = nonparametric("olive.jpeg",32);
```



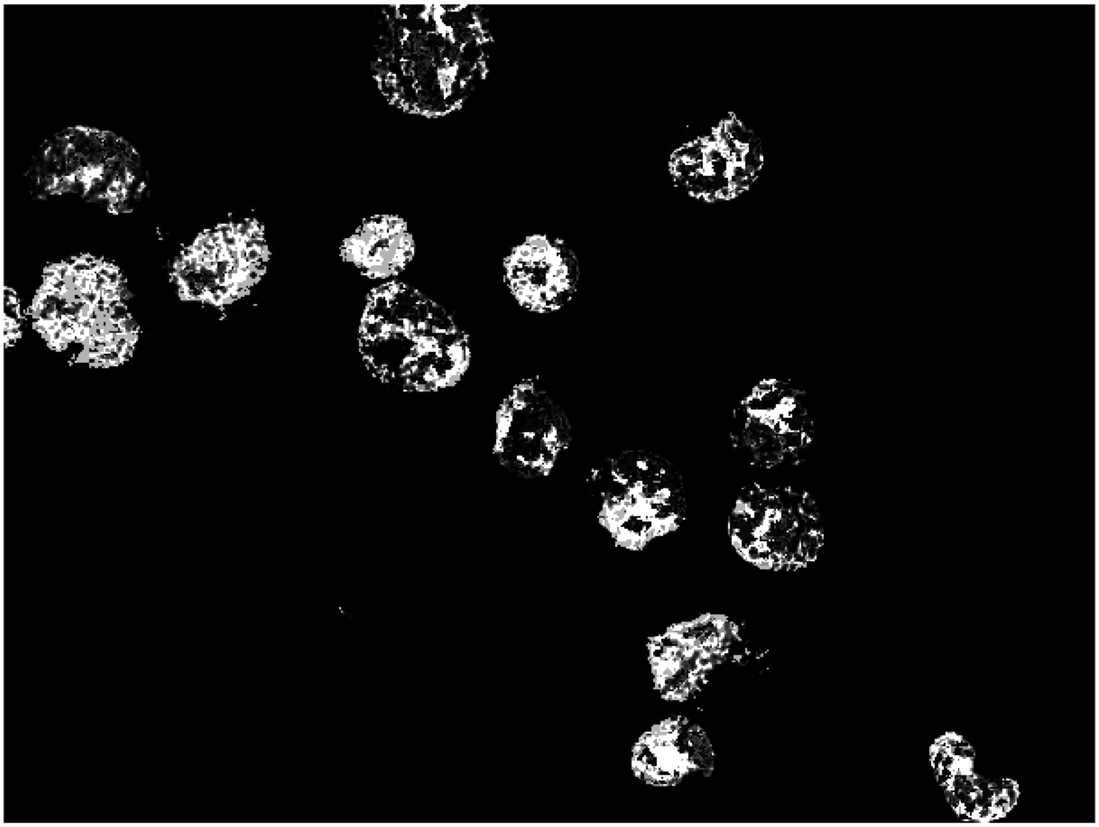
```
barf_nonparametric_32 = nonparametric("barf.jpg",32);
```



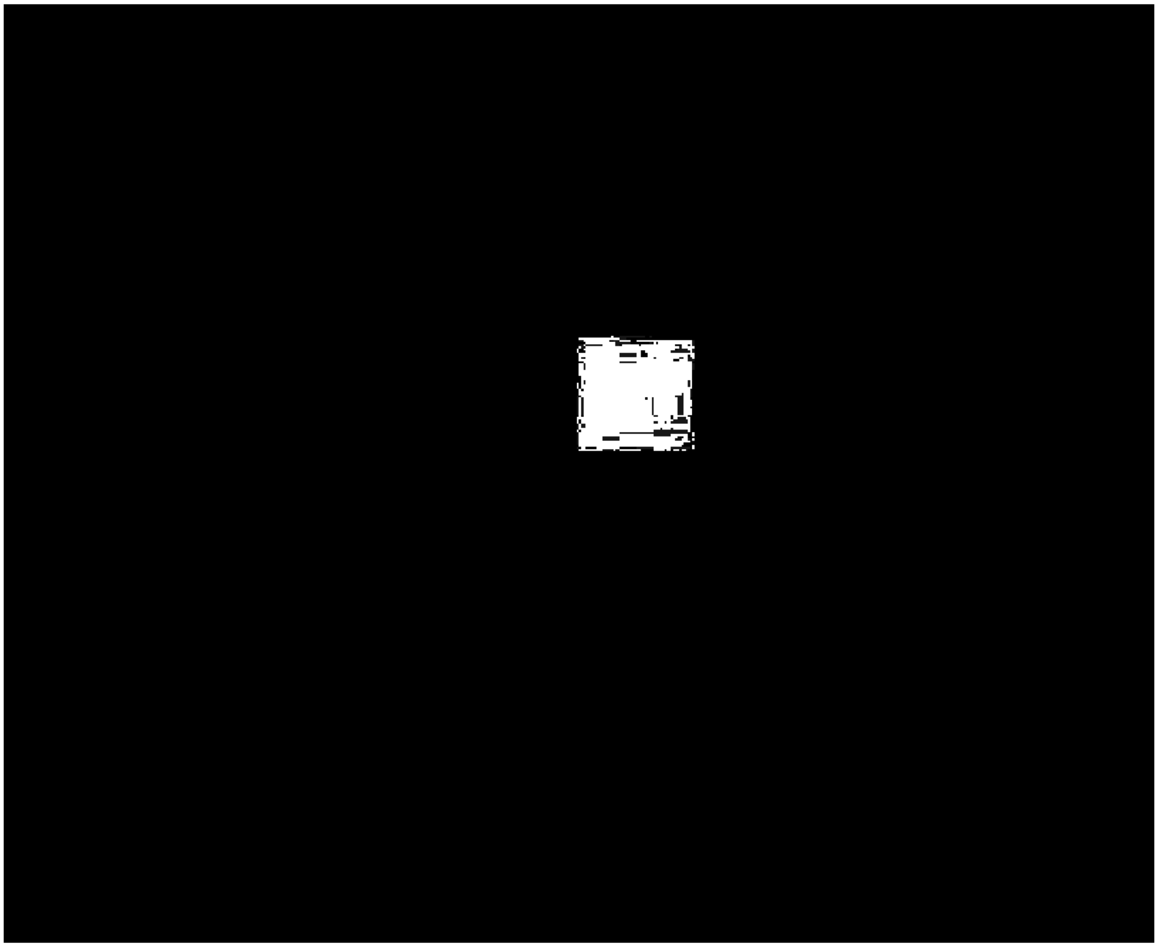
```
mango_nonparametric_32 = nonparametric("mango.jpg",32);
```



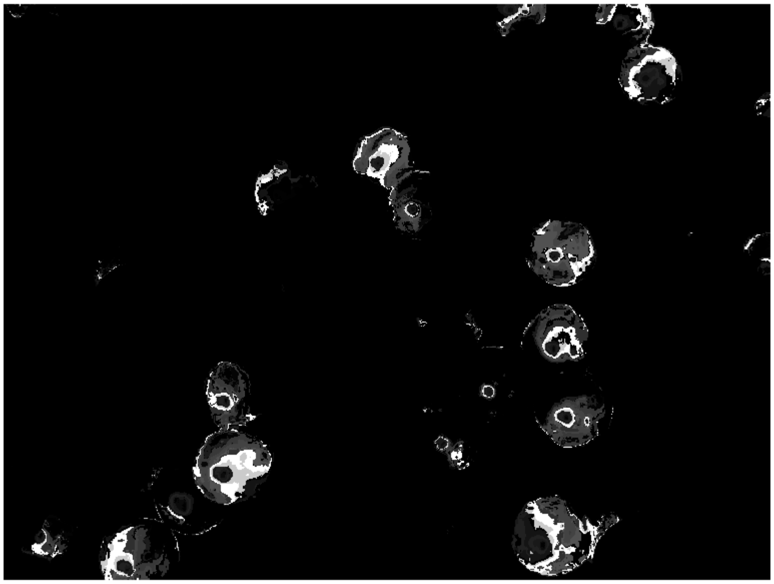
```
cells_nonparametric_32 = nonparametric("cells.jpg",32);
```



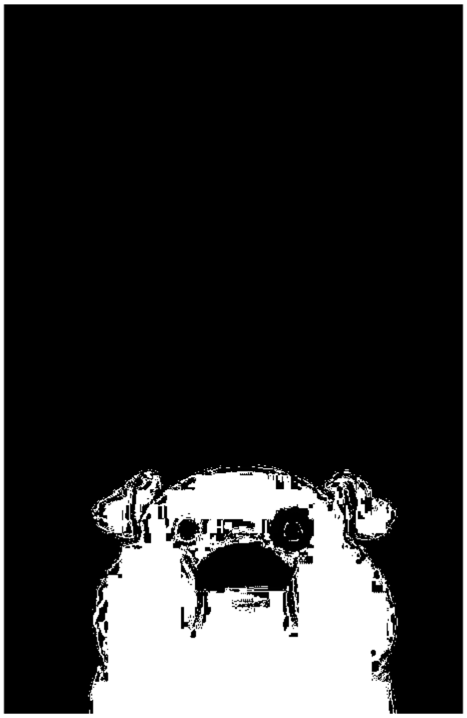
```
%50
macbeth_nonparametric_50 = nonparametric("Gretag-
Macbeth_ColorChecker.jpg",50);
```



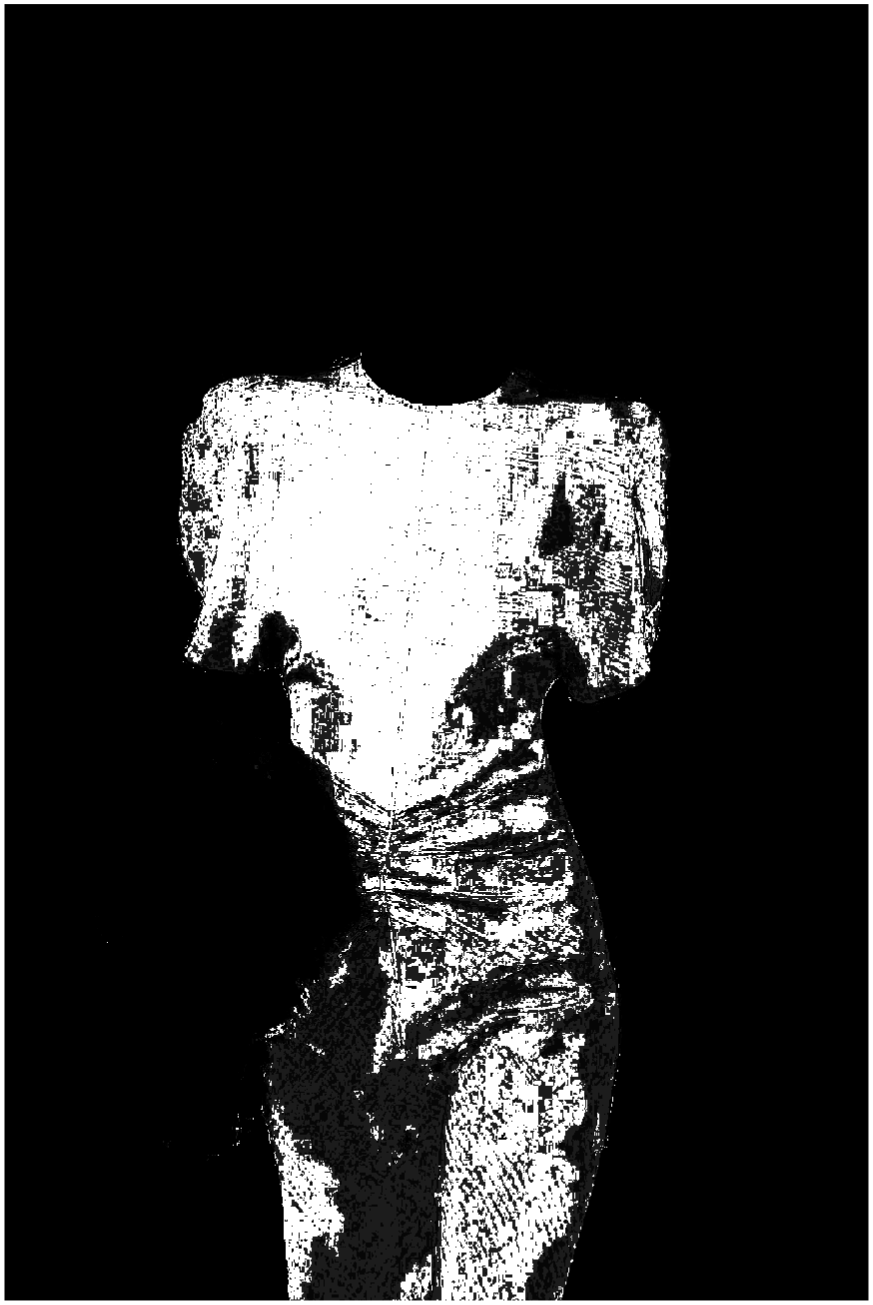
```
nips_nonparametric_50 = nonparametric("nips.jpg",50);
```



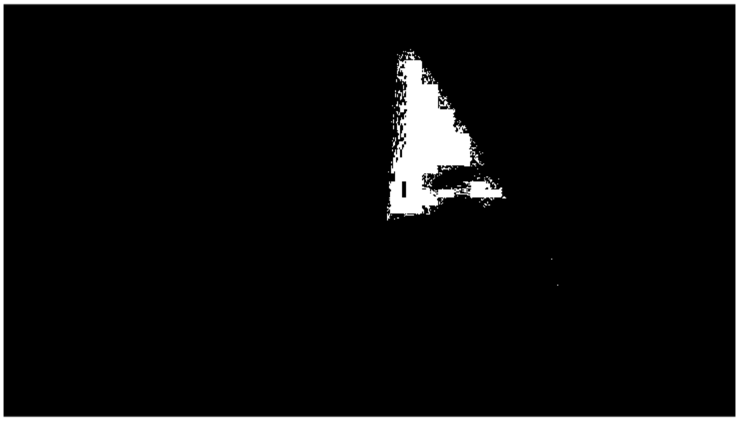
```
waddles_nonparametric_50 = nonparametric("waddles.jpg",50);
```



```
olive_nonparametric_50 = nonparametric("olive.jpeg",50);
```

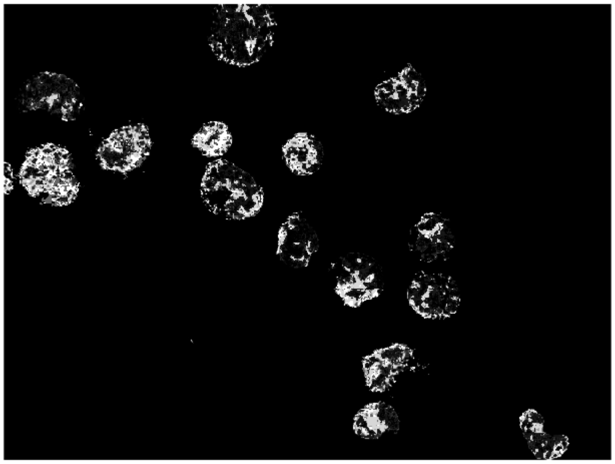


```
barf_nonparametric_50 = nonparametric("barf.jpg",50);
```

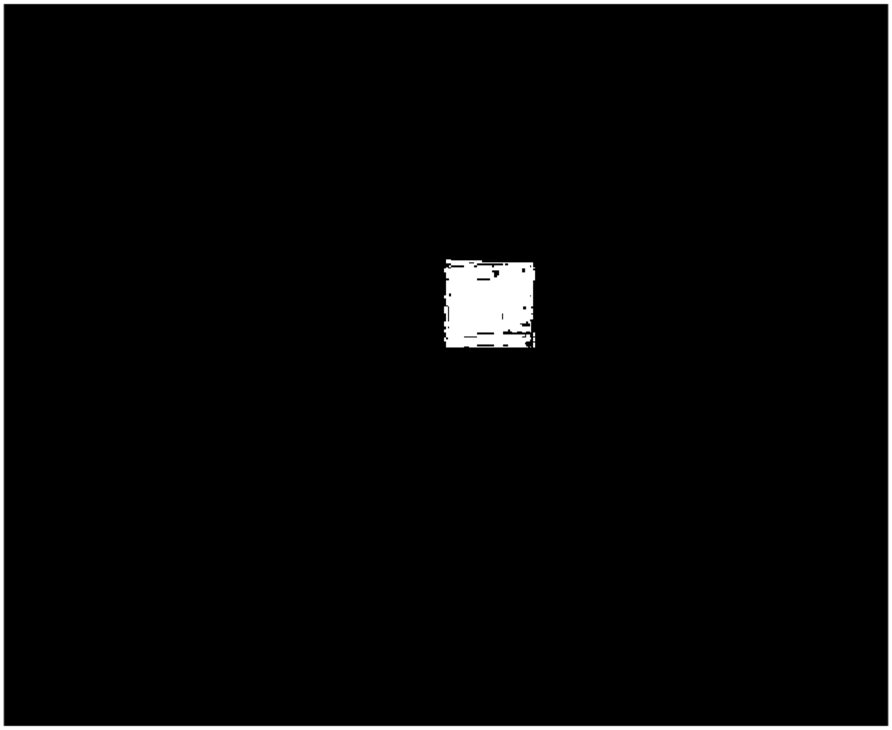


```
mango_nonparametric_50 = nonparametric("mango.jpg",50);
```

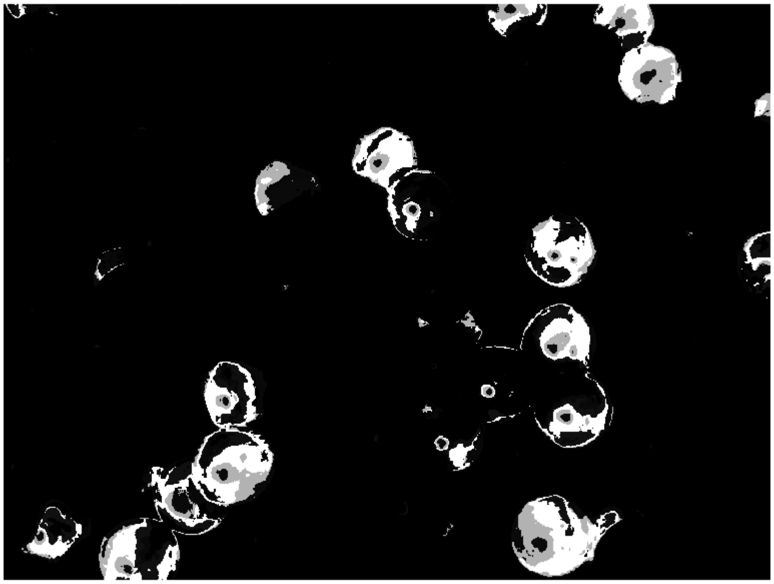
```
cells_nonparametric_50 = nonparametric("cells.jpg",50);
```



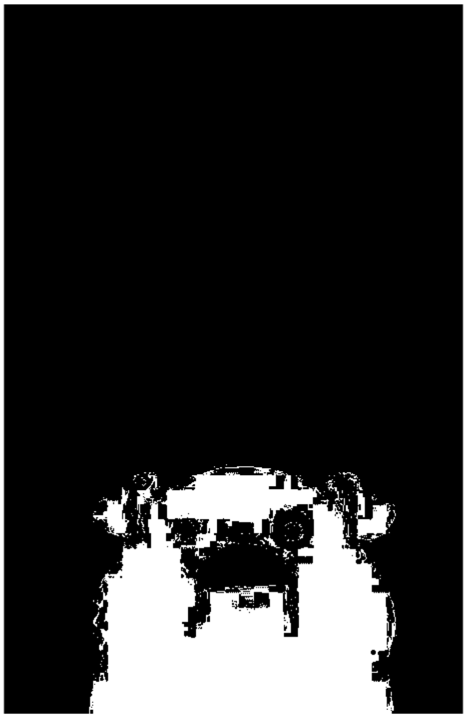
```
%20
macbeth_nonparametric_20 = nonparametric("Gretag-
Macbeth_ColorChecker.jpg",20);
```



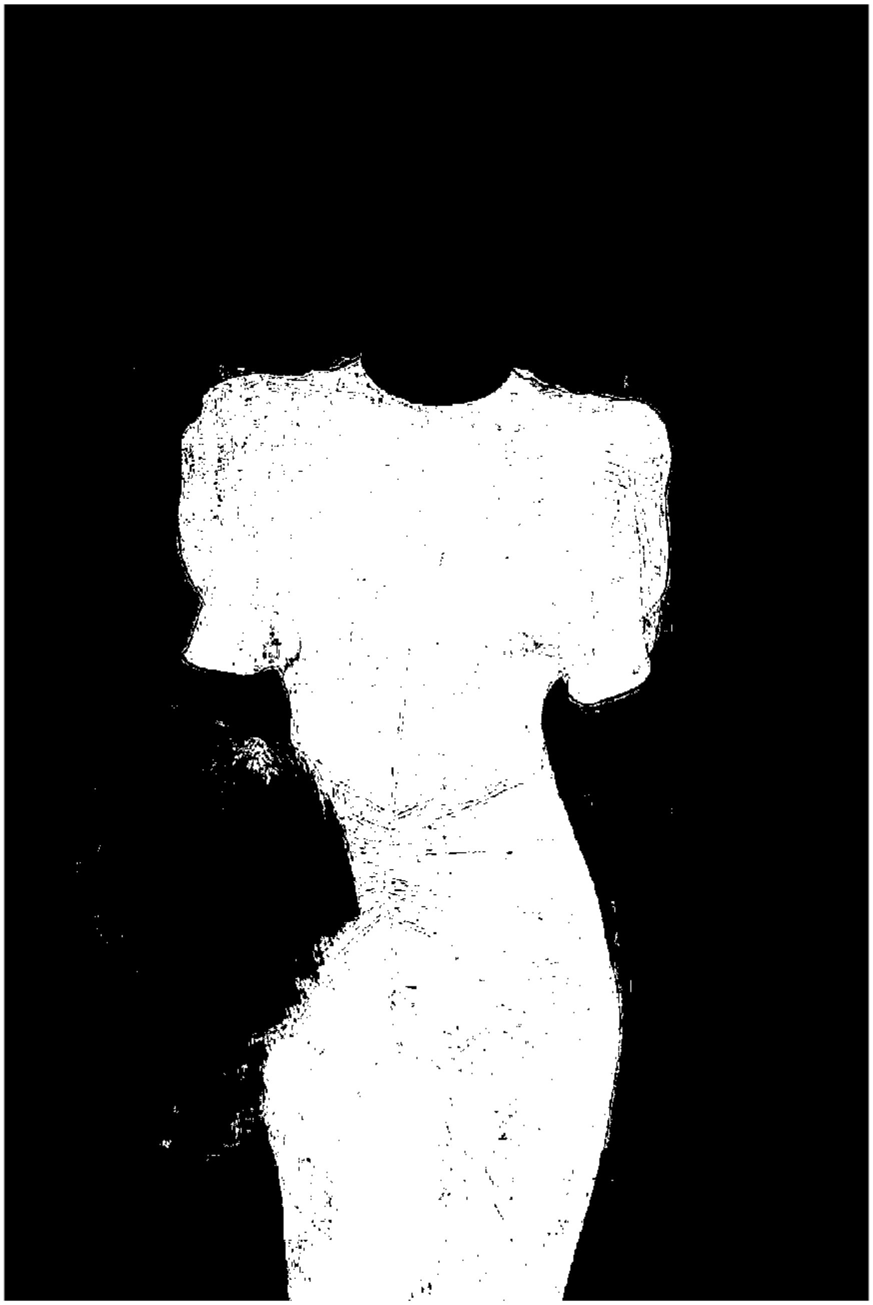
```
nips_nonparametric_20 = nonparametric("nips.jpg",20);
```



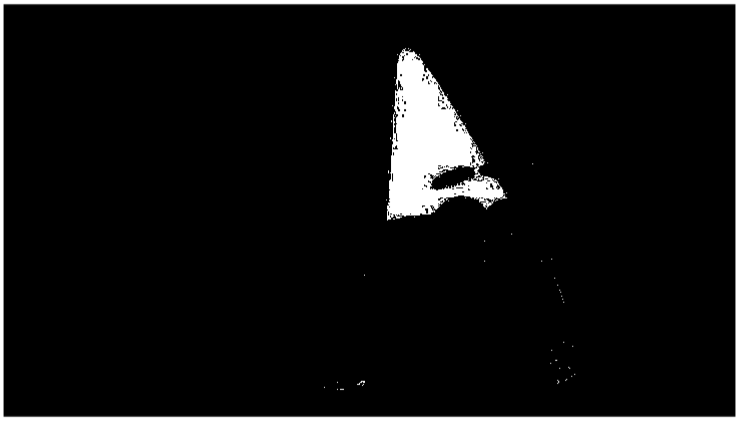
```
waddles_nonparametric_20 = nonparametric("waddles.jpg",20);
```



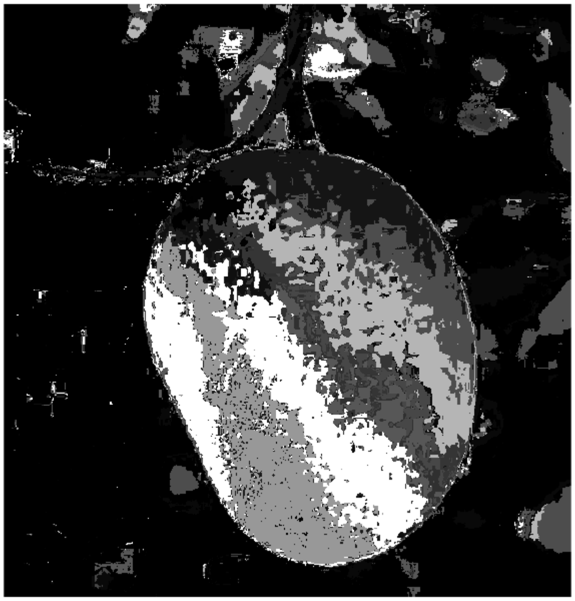
```
olive_nonparametric_20 = nonparametric("olive.jpeg",20);
```



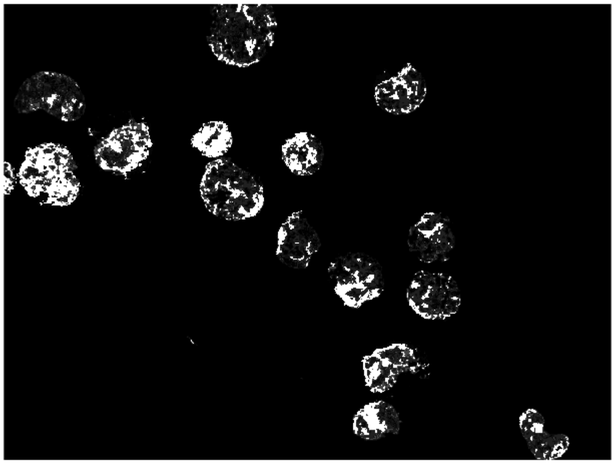
```
barf_nonparametric_20 = nonparametric("barf.jpg",20);
```



```
mango_nonparametric_20 = nonparametric("mango.jpg",20);
```

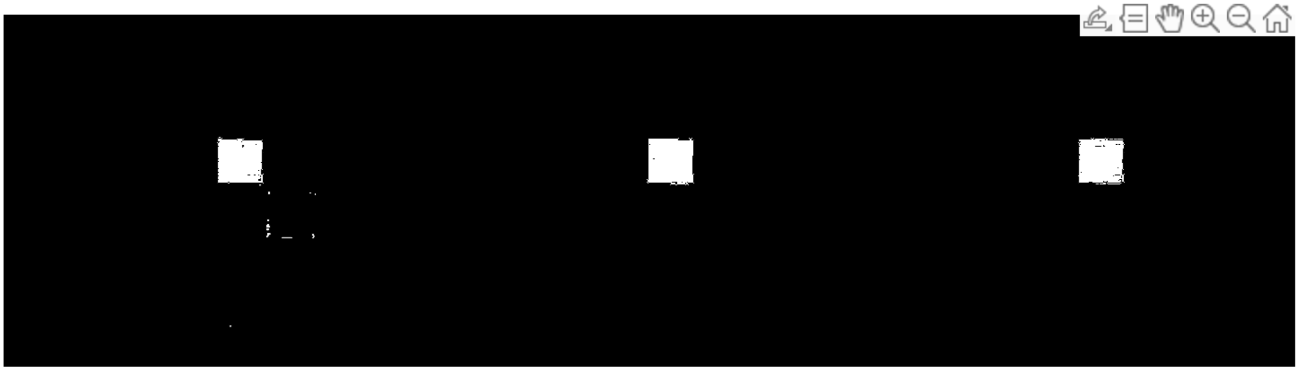


```
cells_nonparametric_20 = nonparametric("cells.jpg",20);
```

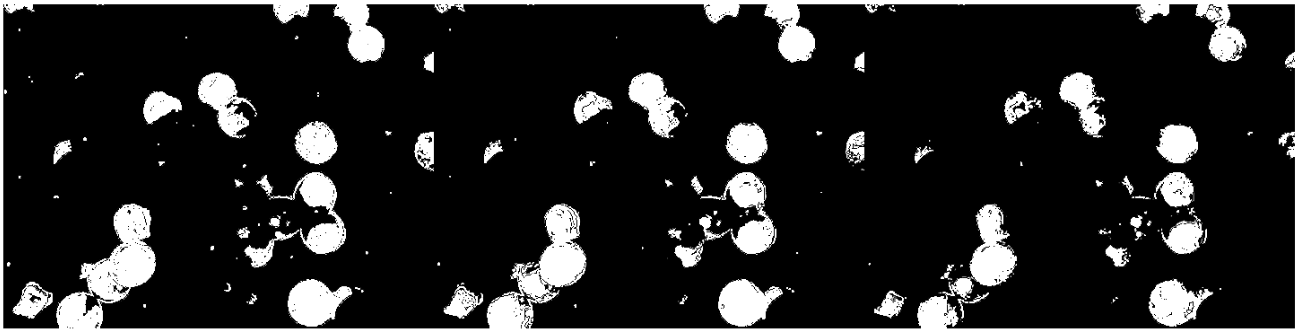


Comparison of Nonparametric Probability Distribution Function with Varying Bin Sizes

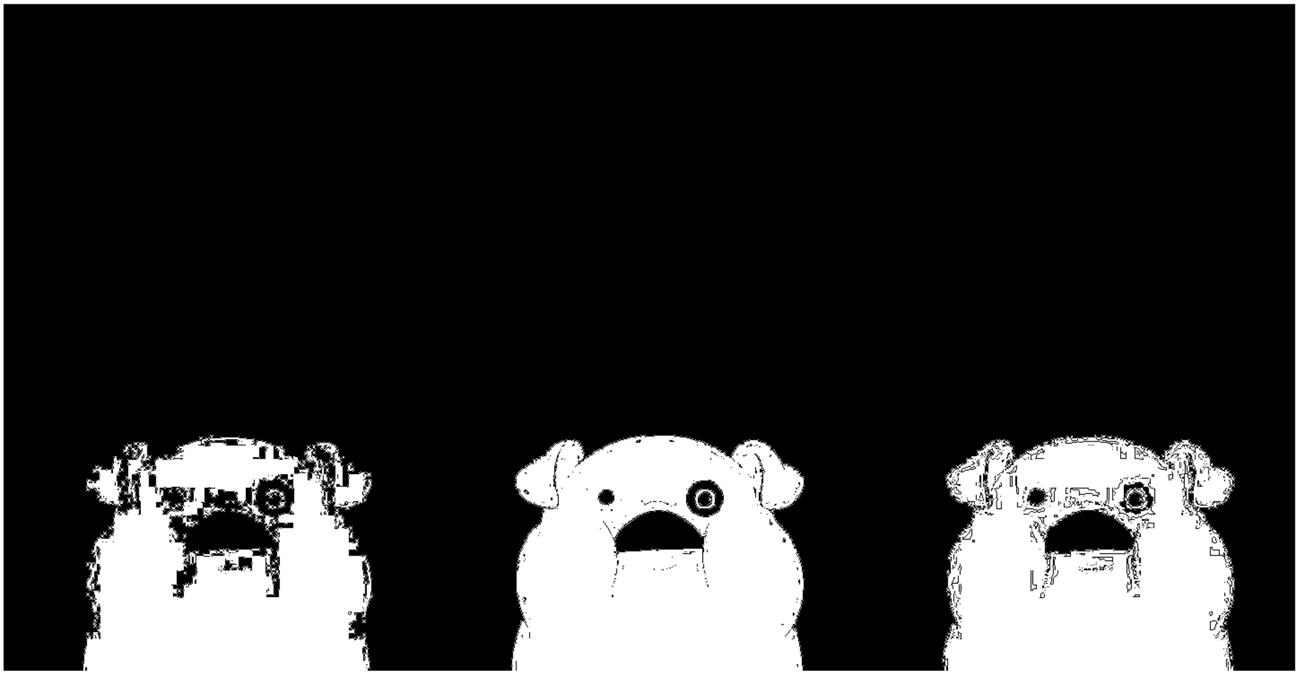
```
montage({macbeth_nonparametric_20,macbeth_nonparametric_32,macbeth_nonparame
tric_50}, 'size', [1 NaN]);
saveas(gcf, "macbeth_nonparametric.png");
```



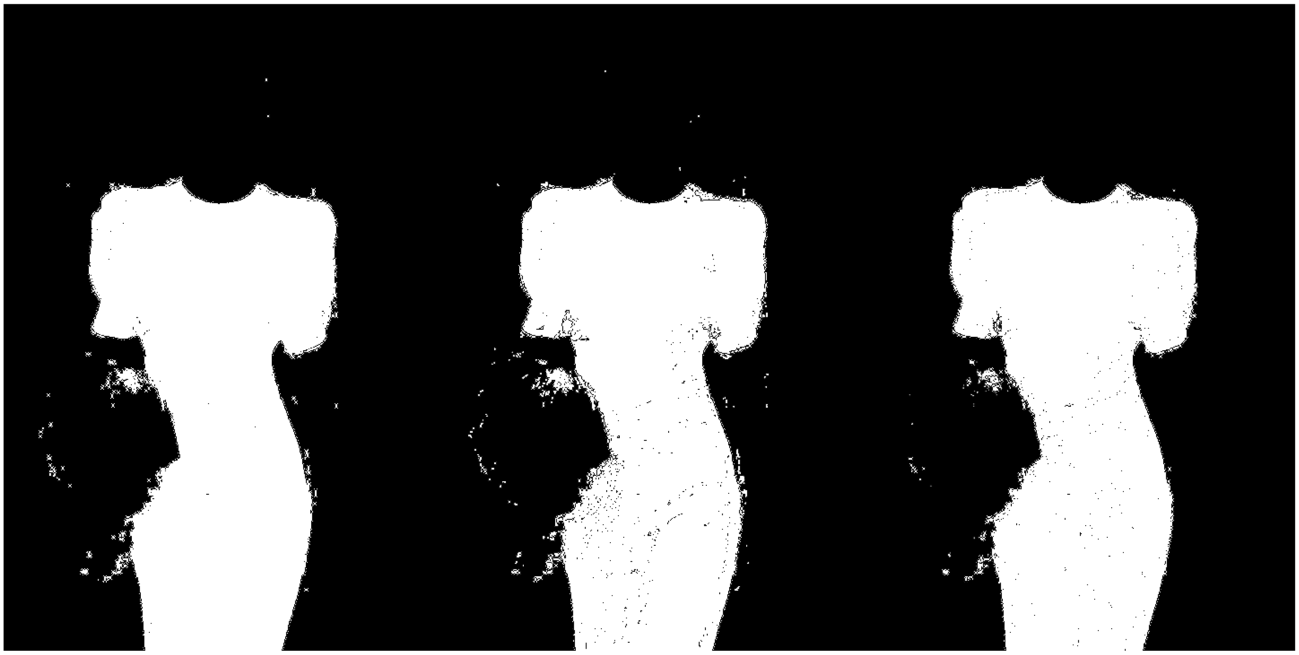
```
montage({nips_nonparametric_20,nips_nonparametric_32,nips_nonparametric_50},
 'size', [1 NaN]);
saveas(gcf, "nips_nonparametric.png");
```



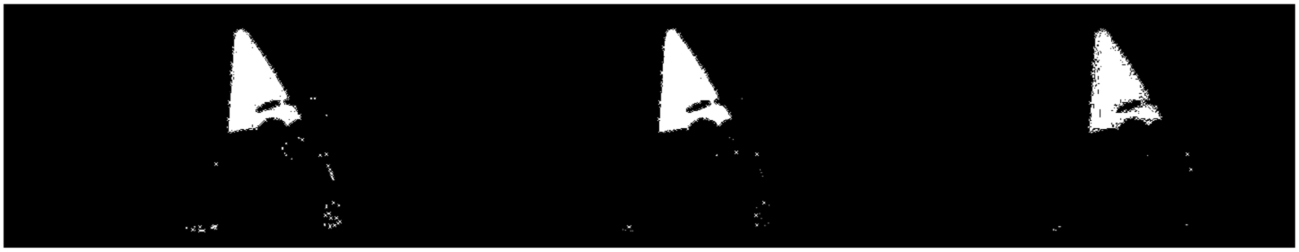
```
montage({waddles_nonparametric_20,waddles_nonparametric_32,waddles_nonparametric_50}, 'size', [1 NaN]);
saveas(gcf, "waddles_nonparametric.png");
```



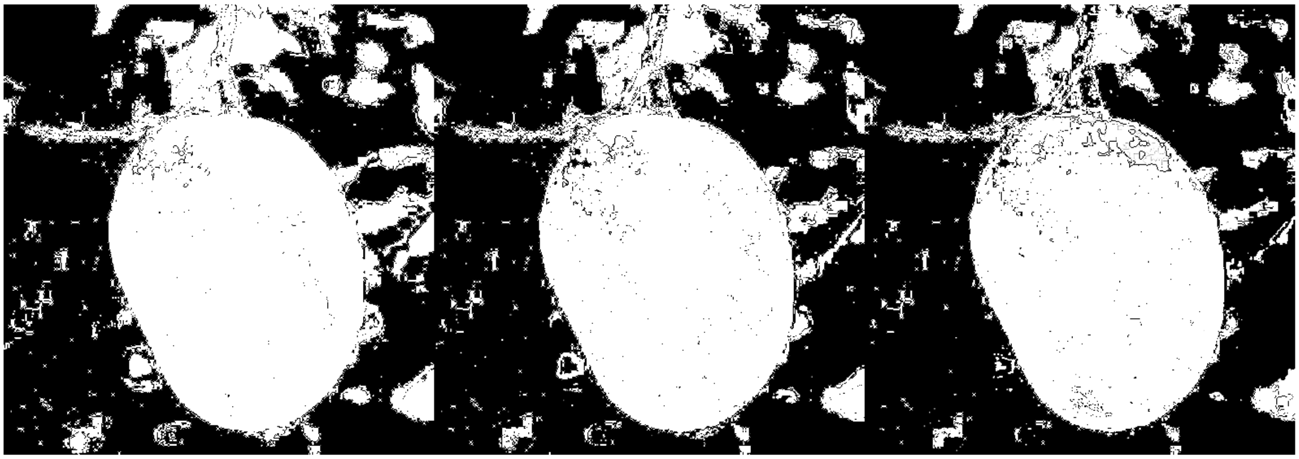
```
montage({olive_nonparametric_20,olive_nonparametric_32,olive_nonparametric_50}, 'size', [1 NaN]);
saveas(gcf, "olive_nonparametric.png");
```



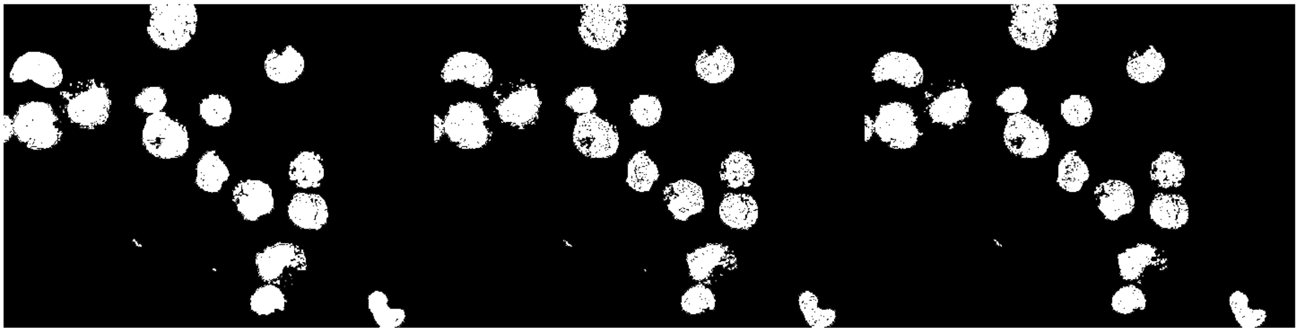
```
montage({barf_nonparametric_20,barf_nonparametric_32,barf_nonparametric_50},
 'size', [1 NaN]);
saveas(gcf, "barf_nonparametric.png");
```



```
montage({mango_nonparametric_20,mango_nonparametric_32,mango_nonparametric_5
0}, 'size', [1 NaN]);
saveas(gcf, "mango_nonparametric.png");
```

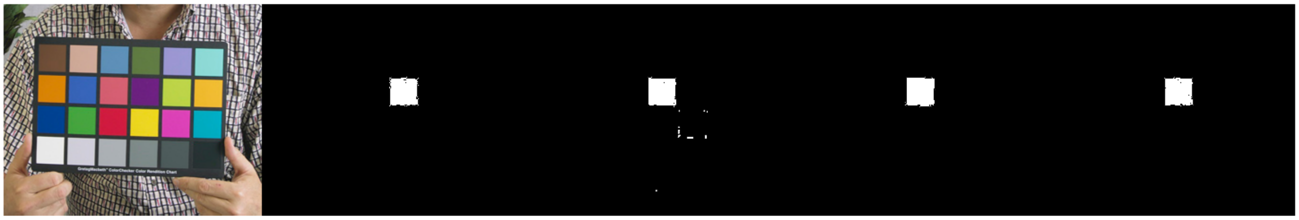


```
montage({cells_nonparametric_20,cells_nonparametric_32,cells_nonparametric_5
0}, 'size', [1 NaN]);
saveas(gcf, "cells_nonparametric.png");
```

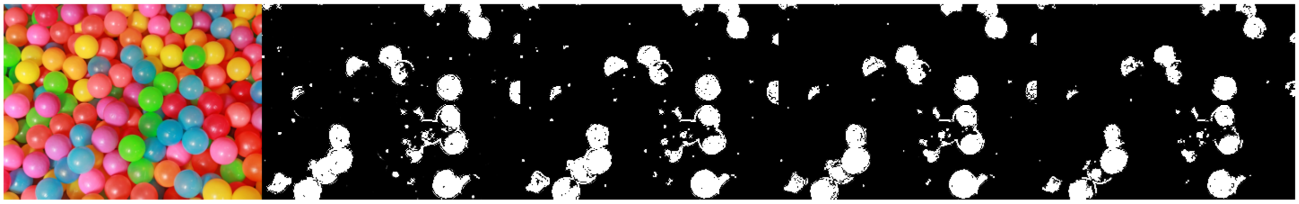


Comparison between Parametric and Nonparametric

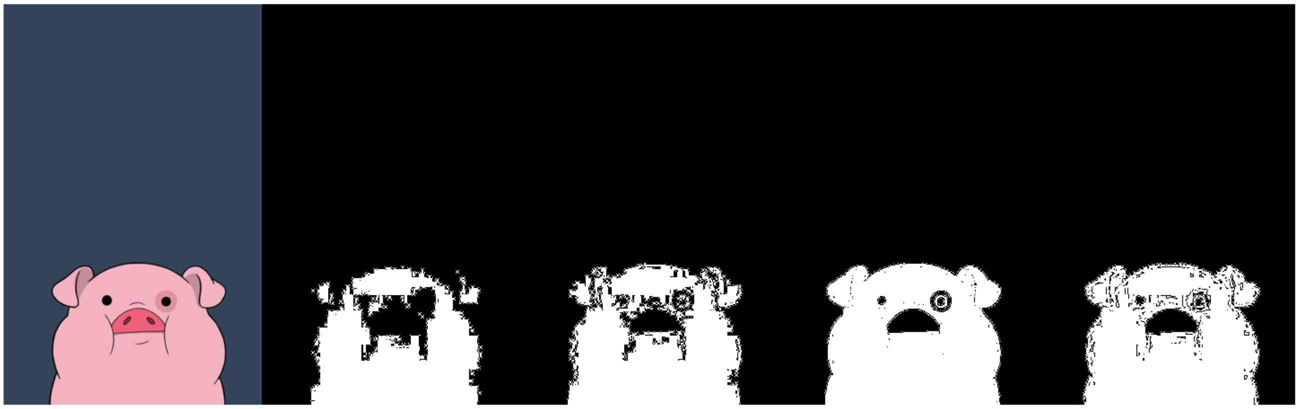
```
montage({macbeth, macbeth_parametric,
macbeth_nonparametric_20,macbeth_nonparametric_32,macbeth_nonparametric_50},
 'size', [1 NaN]);
saveas(gcf, "macbeth_comparison.png");
```



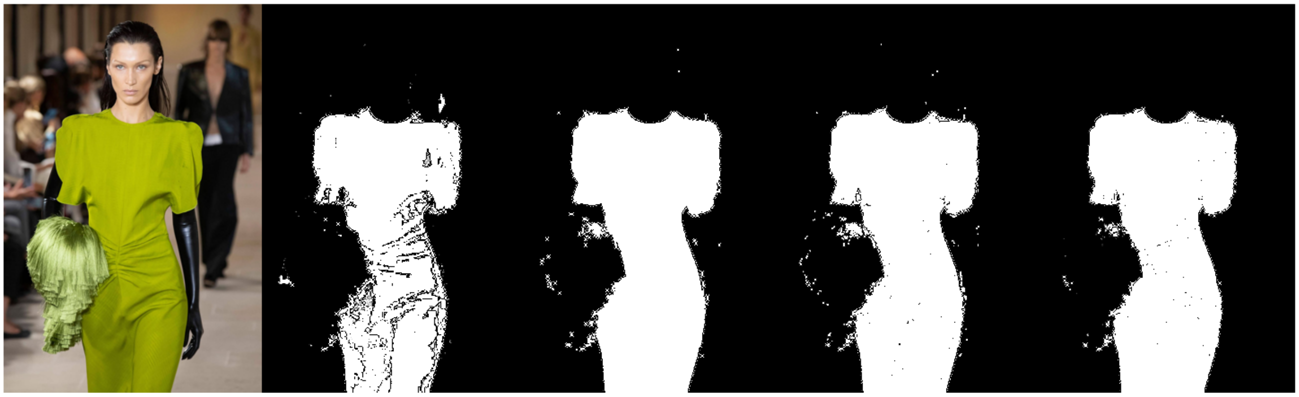
```
montage({nips, nips_parametric,
nips_nonparametric_20,nips_nonparametric_32,nips_nonparametric_50}, 'size',
[1 NaN]);
saveas(gcf, "nips_comparison.png");
```



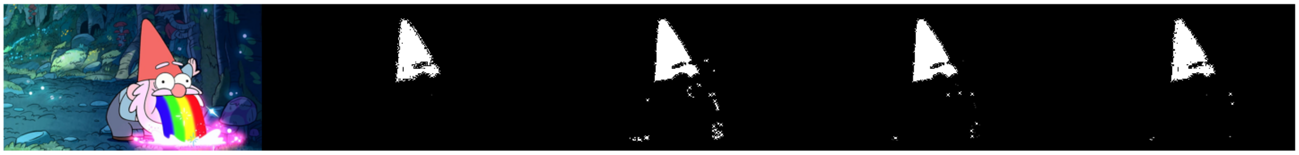
```
montage({waddles,
waddles_parametric,waddles_nonparametric_20,waddles_nonparametric_32,waddles
_nonparametric_50}, 'size', [1 NaN]);
saveas(gcf, "waddles_comparison.png");
```



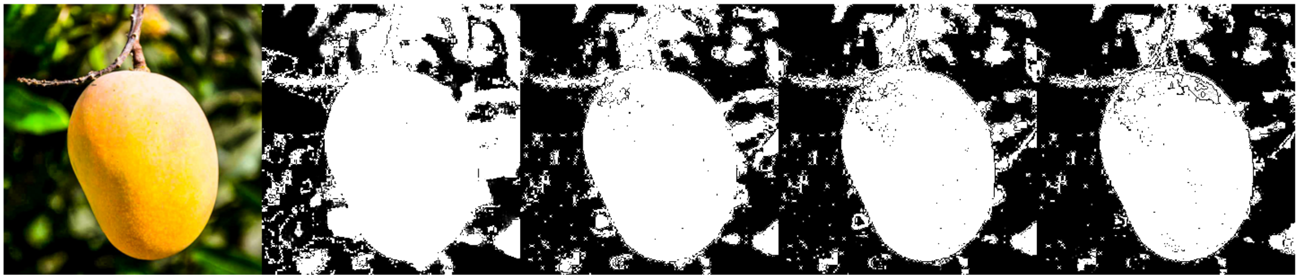
```
montage({olive,
olive_parametric,olive_nonparametric_20,olive_nonparametric_32,olive_nonpara
metric_50}, 'size', [1 NaN]);
saveas(gcf, "olive_comparison.png");
```



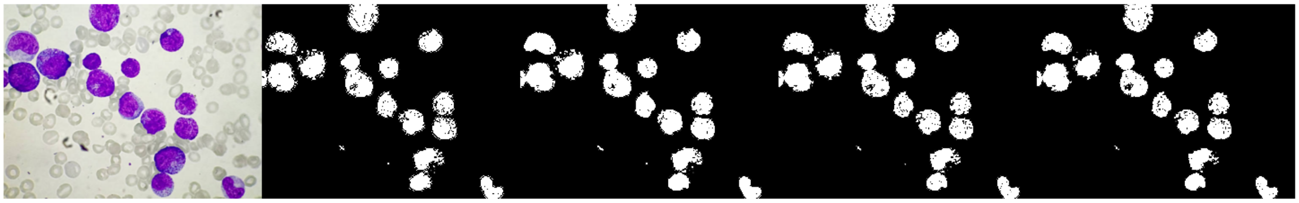
```
montage({barf,
barf_parametric,barf_nonparametric_20,barf_nonparametric_32,barf_nonparametr
ic_50}, 'size', [1 NaN]);
saveas(gcf, "barf_comparison.png");
```



```
montage({mango,
mango_parametric,mango_nonparametric_20,mango_nonparametric_32,mango_nonpara
metric_50}, 'size', [1 NaN]);
saveas(gcf, "mango_comparison.png");
```



```
montage({cells,
cells_parametric,cells_nonparametric_20,cells_nonparametric_32,cells_nonpara
metric_50}, 'size', [1 NaN]);
saveas(gcf, "cells_comparison.png");
```



Functions Used

```
function bw = threshold(x)
    I = imread('cropped_grayscale_check.jpg');
    bw = I < x;
    imshow(bw);
end

% Parametric Probability Distribution Estimation
function prob_combined = parametric(x)
    image = im2double(imread(x));
    [R,G,B] = imsplit(image);
    I = R + G + B;
    I(I==0) = 100000;
    r = double(R./I) ; g = double(G./I) ;

    image_roi = im2double(imcrop(image));
    [R_roi, G_roi, B_roi] = imsplit(image_roi);
    I_roi = double(R_roi + G_roi + B_roi) ;
    r_roi = double(R_roi ./ I_roi) ;
    g_roi = double(G_roi ./ I_roi);

    %red
    image_mu_red = mean2(r_roi);
    image_sigma_red = std2(r_roi);
    image_prob_red = (1 ./ (image_sigma_red) .* sqrt(2.*pi)) .* exp(- (r-
image_mu_red).^2 ./ (2.* image_sigma_red.^2));

    %green
    image_mu_green = mean2(g_roi);
    image_sigma_green = std2(g_roi);
    image_prob_green = (1 ./ (image_sigma_green) .* sqrt(2.*pi)) .* exp(-
(g-image_mu_green).^2 ./ (2.* image_sigma_green.^2));

    prob_combined = image_prob_green .* image_prob_red;

    % Display the original image and the joint probability map side by side
    imshow(prob_combined);
end
```

```
% Non Parametric
function HBImage = nonparametric(x, bins)
    BINS = bins;
    J = im2double(imread(x));
    figure; imshow(J);
    rect = getrect();
    I = imcrop(J, rect);
    %% Get the r g of the whole image
        J = double(J);
        R_roi = J(:,:,1); G_roi = J(:,:,2); B_roi = J(:,:,3);
        Int= R_roi + G_roi + B_roi;
        Int(Int==0)=100000; %to prevent NaNs
        rJ = R_roi./ Int; gJ = G_roi./Int;

    %% Crop the region of interest in the rg space

        r_roi = imcrop(rJ, rect);
        g_roi = imcrop(gJ, rect);
        rint = round( r_roi*(BINS-1) + 1);
        gint = round (g_roi*(BINS-1) + 1);
        colors = gint(:) + (rint(:)-1)*BINS;

    %% Compute rg-histogram
    % This is the 1-d version of a 2-d histogram
        hist = zeros(BINS*BINS,1);
        for row = 1:BINS
        for col = 1:(BINS-row+1)
        hist(col+(row-1)*BINS) = length( find(colors==( ((col +
(row-1)*BINS)))));
        end
        end

    %% Backproject histogram
        rJint = round( rJ*(BINS-1) + 1);
        gJint = round (gJ*(BINS-1) + 1);
        colorsJ = gJint(:) + (rJint(:)-1)*BINS;
        HB = hist(colorsJ);
        HBImage = reshape(HB,size(J,[1,2]));
        figure (2); imagesc(HBImage);
        axis off;
        axis image;
        colormap (gray);

end
```