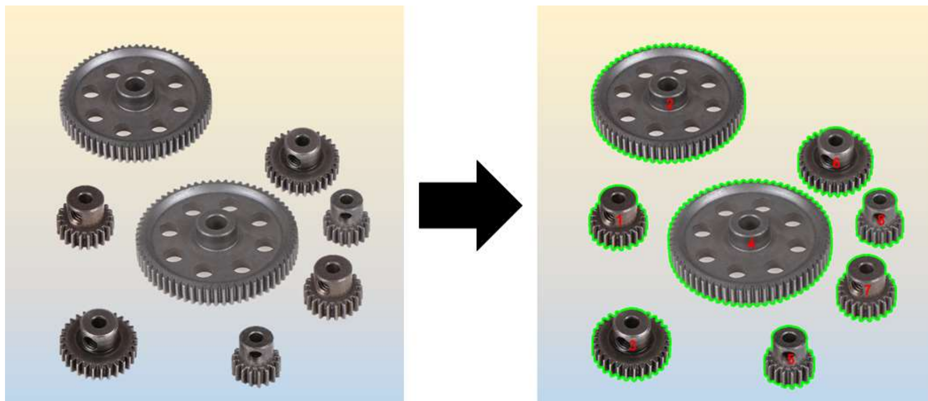
Task 1: MATLAB Image Analysis

a) Blob detection and boundary plotting

In the GearsAndWashers.tif image:

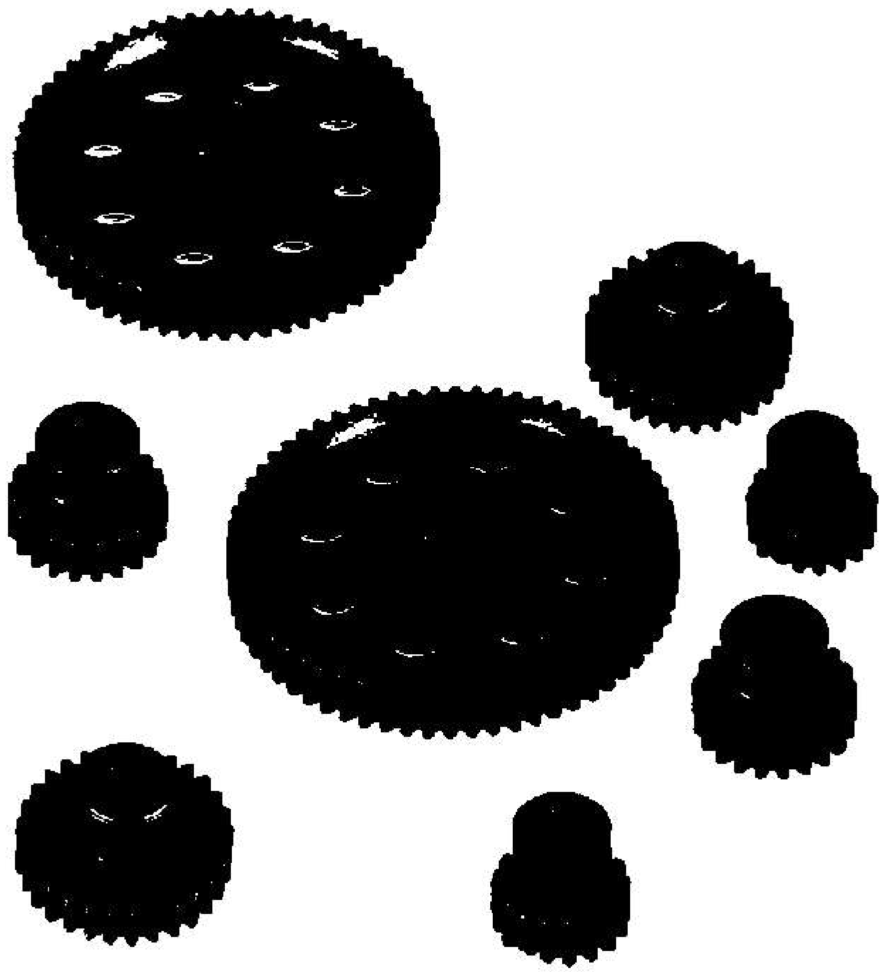
- Put a boundary on each of the object and a number label identifying them.
- Print the properties (mean intensity, area, perimeter, centroid, diameter) of each of the object.



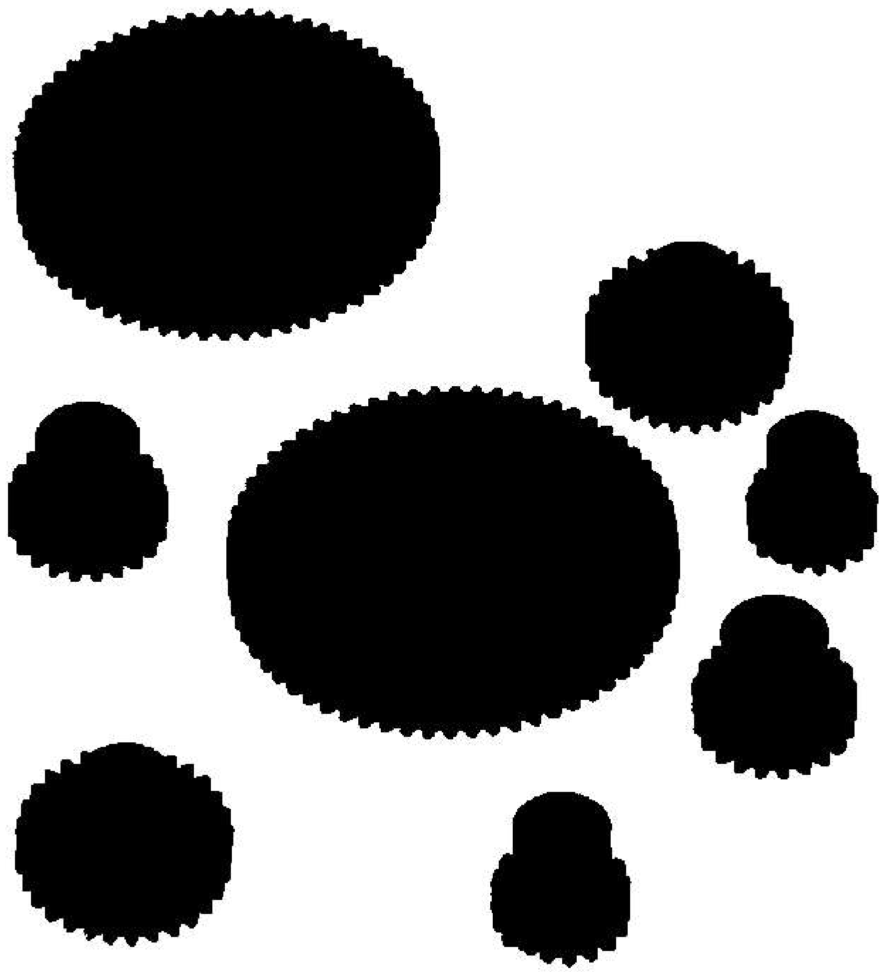
i. Convert the image to a clean binary image with no holes by thresholding and **`imfill`** function.

```
% Read and show the original image.
orig_gears = imread("Gears.tif");
imshow(orig_gears)
```

```
% Use a threshold value to convert from grayscale to a binary image and
% show the resulting image.
threshold = 0.8;
grey_gears = im2bw(orig_gears, threshold);
imshow(grey_gears);
```

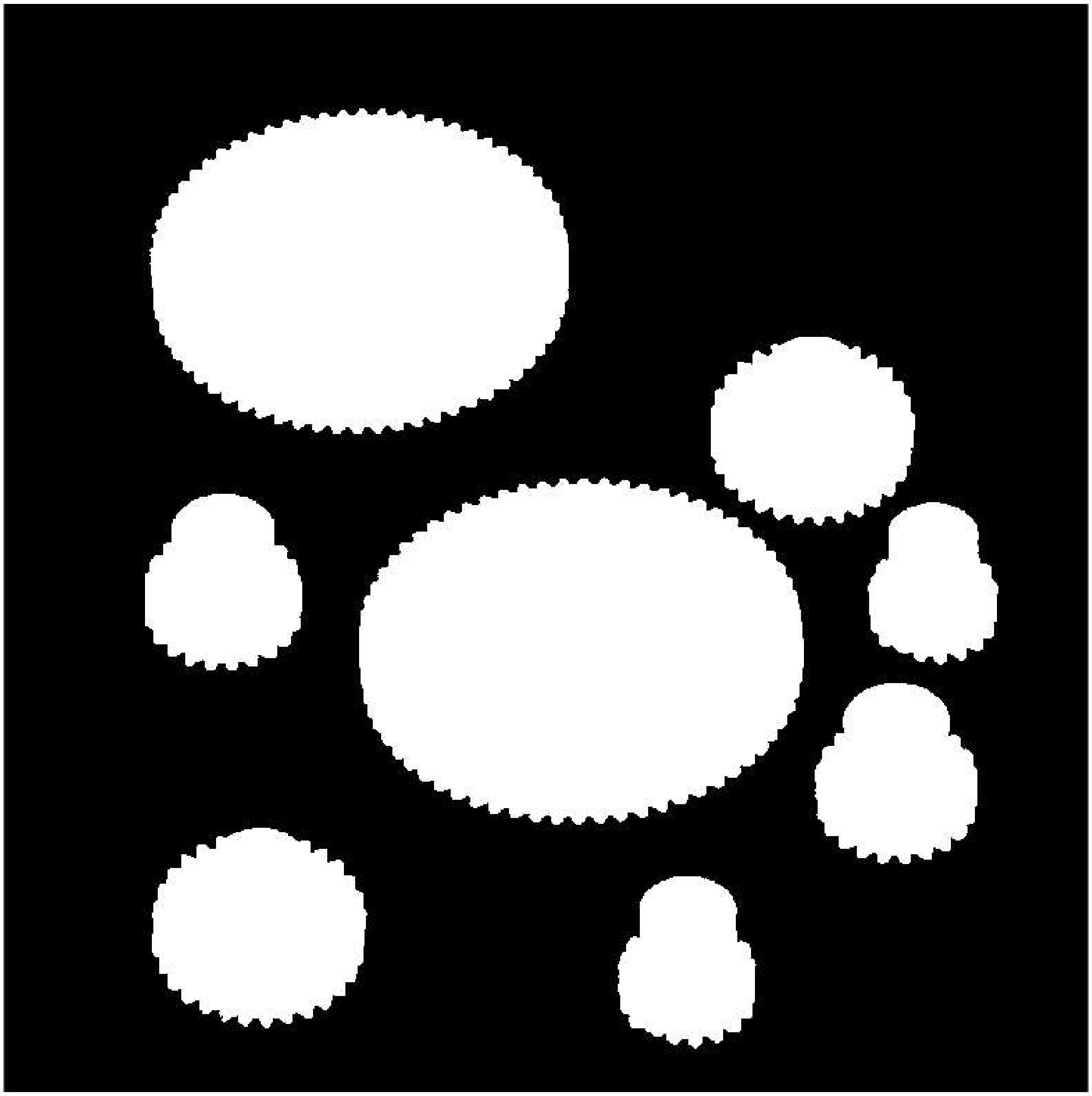


```
% Perform  a "hole fill" to get rid of holes in the gears and any background pixels
% inside the blobs and show the resulting image.
%gear_flipped = imcomplement(grey_gears);
gear_filled = imfill(~grey_gears, 'holes');
%gear_filled = imcomplement(gear_filled_flipped);
imshow(~gear_filled)
```

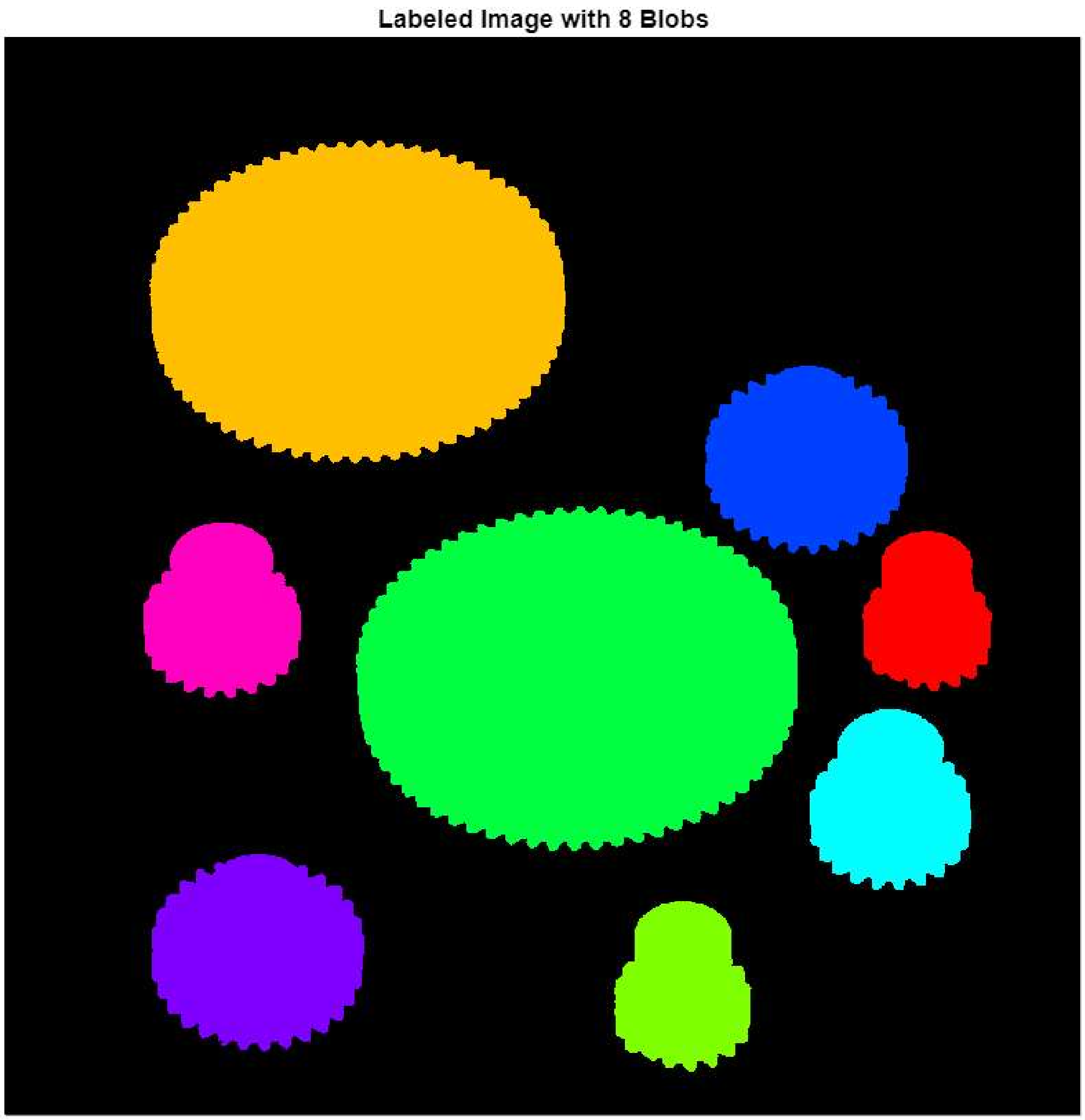


ii. Use the **bwlabel** command to generate a labeled image showing each of the blobs in the image.

```
% Label blobs in the image and show the resulting image.
[labeledImage, numBlobs] = bwlabel(gear_filled);

imshow(labeledImage)
```



```
% Show labels in pseudo random colors.

colormap = rand(numBlobs, 3);
NumGears = label2rgb(labeledImage, 'hsv', 'k', 'shuffle');
imshow(NumGears)
title(['Labeled Image with ', num2str(numBlobs), ' Blobs']);

xlim([1 1025])
ylim([1 1027])
```



iii. Use the **regionprops** command to get the nbwer of blobs found and the properties of each of the blobs.

```
% Get the number and properties of blobs found.

blobregions = regionprops(labeledImage, grey_gears, 'Area', 'Perimeter', 'Centroid', 'EquivDiameter', 'MeanIntensity')
```

blobregions = 8×1 struct

Fields	Area	Centroid	EquivDiameter	Perimeter	MeanIntensity
1	18909	[207.8482,548.8489]	155.1634	578.8780	0.0025
2	95632	[336.8806,252.5023]	348.9448	1.3772e+03	0.0242
3	29222	[241.7648,871.5010]	192.8901	762.6740	0.0059
4	108382	[545.6927,610.4708]	371.4785	1.4649e+03	0.0083
5	16007	[646.0342,905.8617]	142.7611	525.6770	7.4967e-04
6	26491	[763.9205,402.7607]	183.6556	722.2660	0.0031
7	19869	[843.3704,729.0863]	159.0534	595.3860	0.0011
8	14159	[878.1643,549.2070]	134.2676	489.5860	1.4125e-04

iv. Use the **bwboundaries** command to plot the boundary of each of the blobs found on the original grayscale image to highlight the objects.

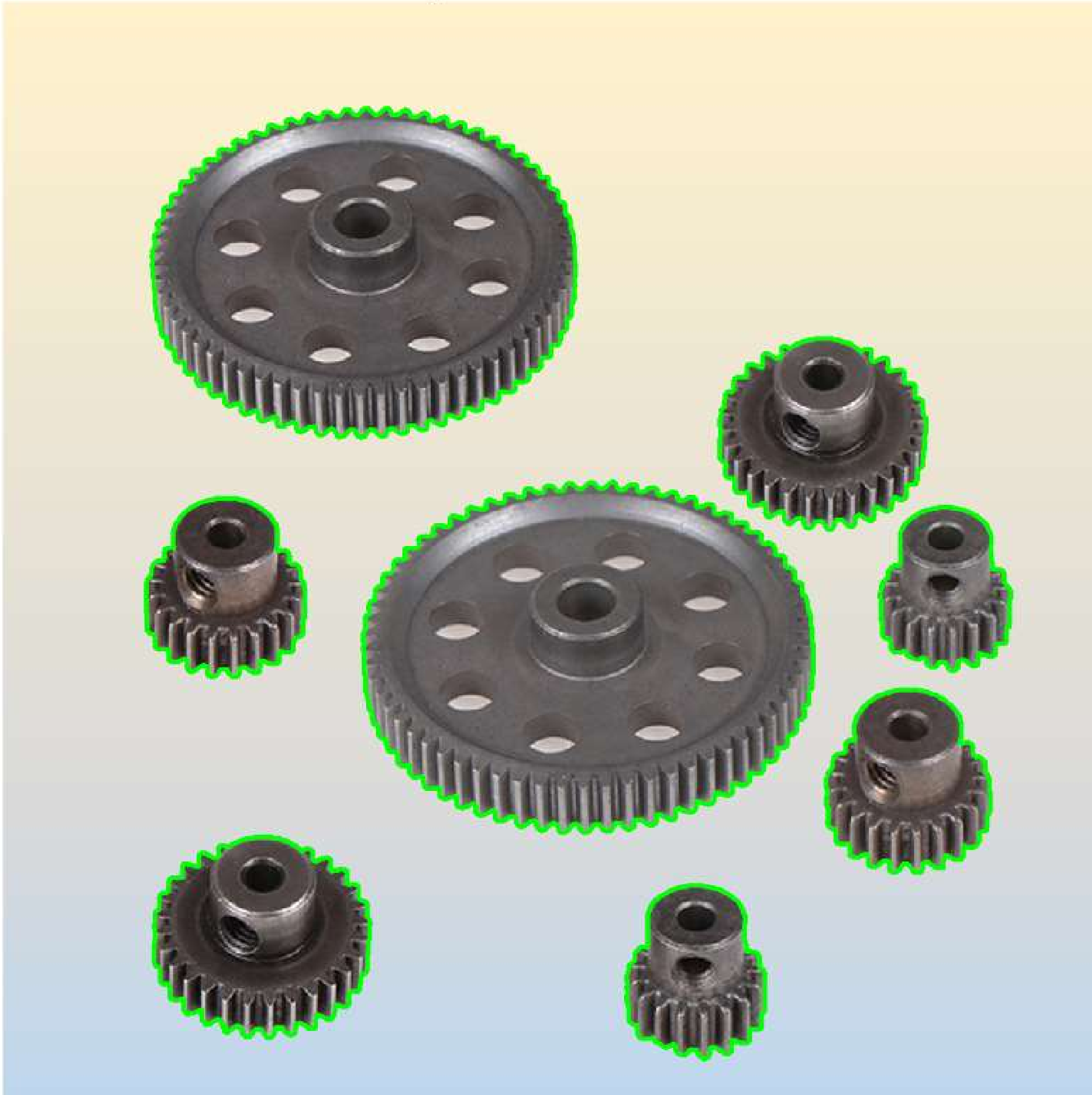
```
% bwboundaries returns a cell array where each cell contains the
% row/column coordinates for an object in the image.

gearBoundries = bwboundaries(labeledImage)
```

gearBoundries = 8×1 cell

	1
1	545×2 double
2	1282×2 double
3	716×2 double
4	1364×2 double
5	490×2 double
6	678×2 double
7	556×2 double
8	457×2 double

```
% Plot the borders of all the objects on the original grayscale image
% using the coordinates returned by bwboundaries.

figure;
imshow(orig_gears);
title('Original Picture w/ Borders');

hold on;

for k = 1:length(gearBoundries)
    boundary = gearBoundries{k};
    plot(boundary(:, 2), boundary(:, 1), 'g', 'LineWidth', 2);
end

hold off;
```

Original Picture w/ Borders



v. Print the blob properties (mean intensity, area, perimeter, centroid, diameter) of each blob found and plot blob number labels on the objects in the outlined grayscale image.

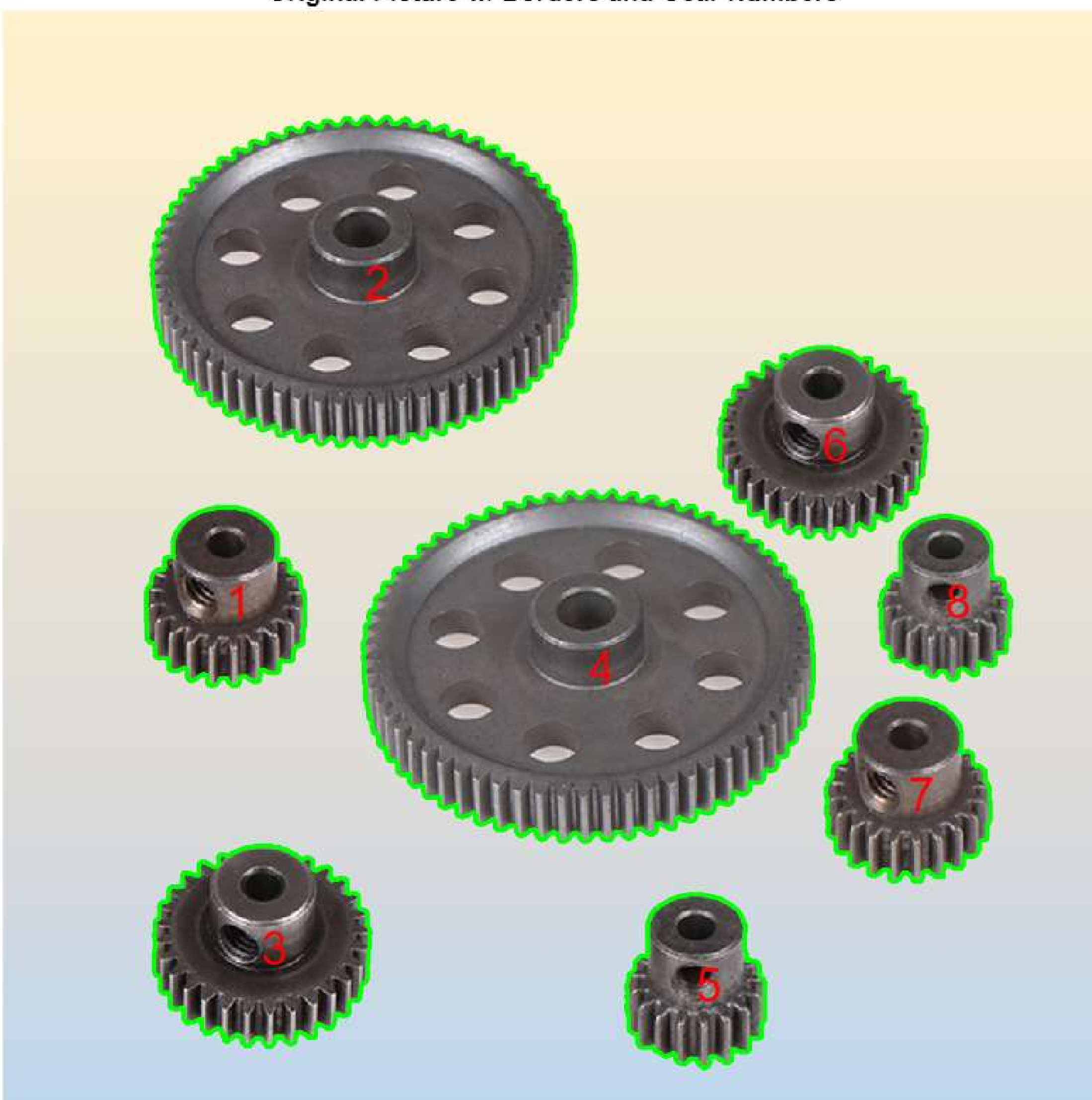
```
% Printing out the properties of each blob

%creating the table to print values to
properties = cell(numel(blobregions), 6);
for i = 1:numel(blobregions)
    properties{i, 1} = i;
    properties{i, 2} = (blobregions(i).Area);
    properties{i, 3} = (blobregions(i).Perimeter);
    properties{i, 4} = blobregions(i).Centroid(1); blobregions(i).Centroid(2);
    properties{i, 5} = (blobregions(i).EquivDiameter);
    properties{i, 6} = blobregions(i).MeanIntensity;
    % Calculating Mean Intensity
end

propTable = cell2table(properties, 'VariableNames', {'Blob Num', 'Area', 'Perimeter', 'Centroid', 'Diameter', 'Mean Intensity'});
disp(propTable)
```

Blob Num	Area	Perimeter	Centroid	Diameter	Mean Intensity
1	18909	578.88	207.85	155.16	0.0025385
2	95632	1377.2	336.88	348.94	0.024166

```
         3          29222       762.67       241.76       192.89        0.0058518
         4     1.0838e+05       1464.9       545.69       371.48        0.0083409
         5          16007       525.68       646.03       142.76       0.00074967
         6          26491       722.27       763.92       183.66        0.0030576
         7          19869       595.39       843.37       159.05        0.0010569
         8          14159       489.59       878.16       134.27       0.00014125
```

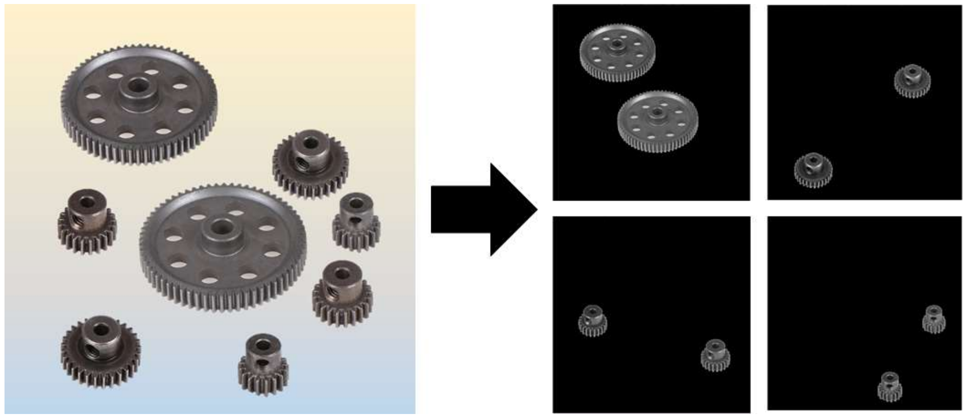
```
% Put the blob number labels on the boundaries grayscale image.
figure;
imshow(orig_gears);
title('Original Picture w/ Borders and Gear Numbers');

hold on;

for k = 1:length(gearBoundries)
    boundary = gearBoundries{k};
    plot(boundary(:, 2), boundary(:, 1), 'g', 'LineWidth', 2);
end

hold on
for i = 1:numel(blobregions)
    text(blobregions(i).Centroid(1), blobregions(i).Centroid(2), num2str(i), 'Color', 'Red', 'FontSize', 20);
end
hold off
```

Original Picture w/ Borders and Gear Numbers



b) Find the different object types in the image

Identify the different object types (large gears, medium-large gears, medium-small gears, and small gears) in the image using the blob properties obtained in part (a) and extract them in separate images.



i. Check blob properties from part (a) and identify which criteria to use to separate the four different types of the objects.

```
% Check criteria of blob properties from part(a)(v) to select the best one to
% separate object types

%Can I just assume to use Area seeing as the image we're using isn't
%changing for now?
```

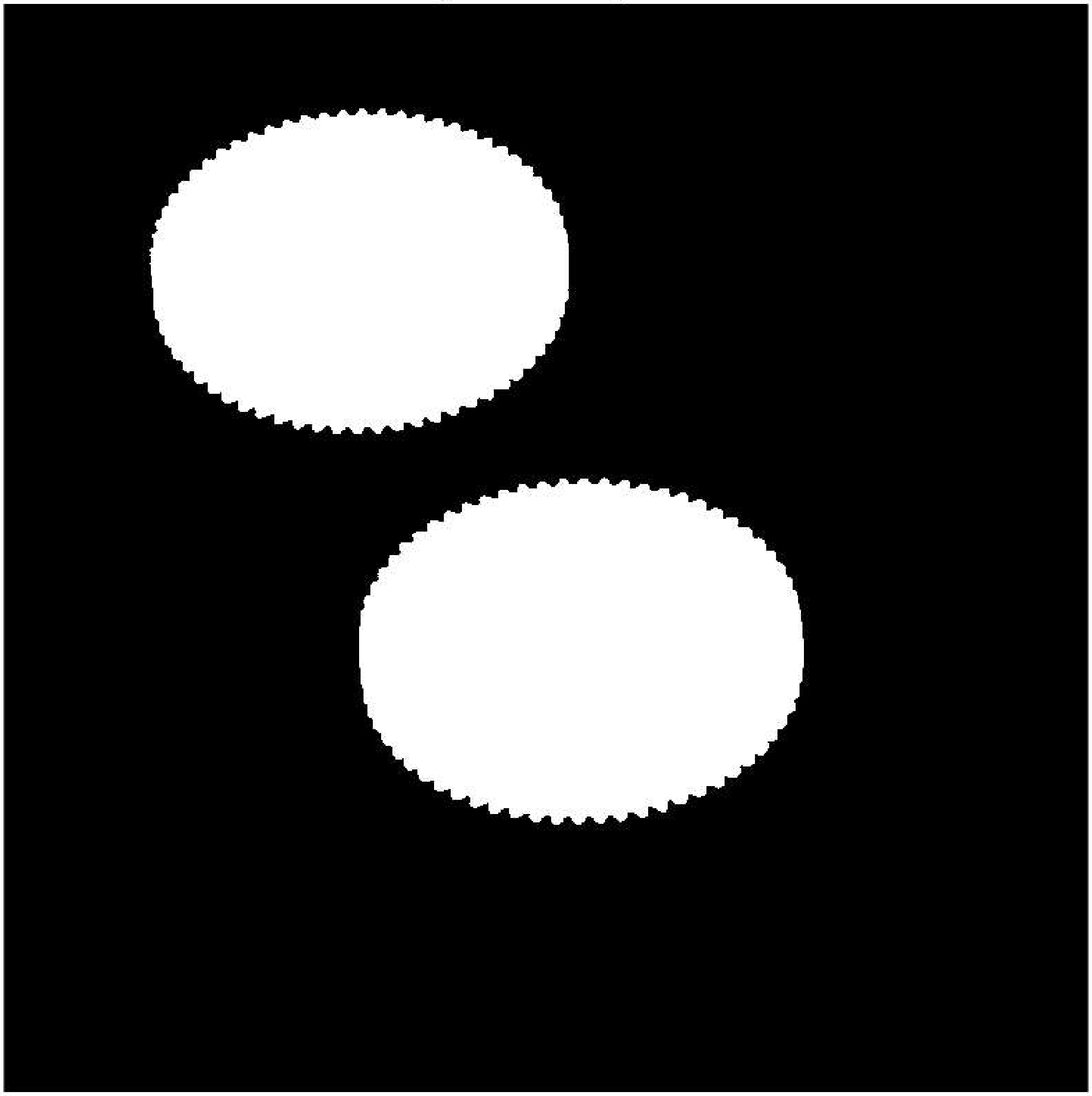
ii. Use the **`ismember`** function to extract the selected blobs in the labeled image that was generated in part (a) using the **`bwlabel`** command. Create a binary mask from the labeled image and filter each type of the object for display in the original grayscale image.

```
% Extract large gears by selecting the blobs in the labeled image that meet
% the criteria and show the binary image mask with separated gears.
blobArea = regionprops(labeledImage, 'Area');

minAreaBig = 95000;
maxAreaBig = 110000;

ValidBlobsBig = [blobArea.Area] >= minAreaBig & [blobArea.Area] <= maxAreaBig;
validLabelsForBig = find(ValidBlobsBig)
```

```
validLabelsForBig = 1×2
     2     4
```

```

binaryMaskBig = ismember(labeledImage, validLabelsForBig);

figure;
imshow(binaryMaskBig);
title('Binary Mask for Large Gears');
```

Binary Mask for Large Gears



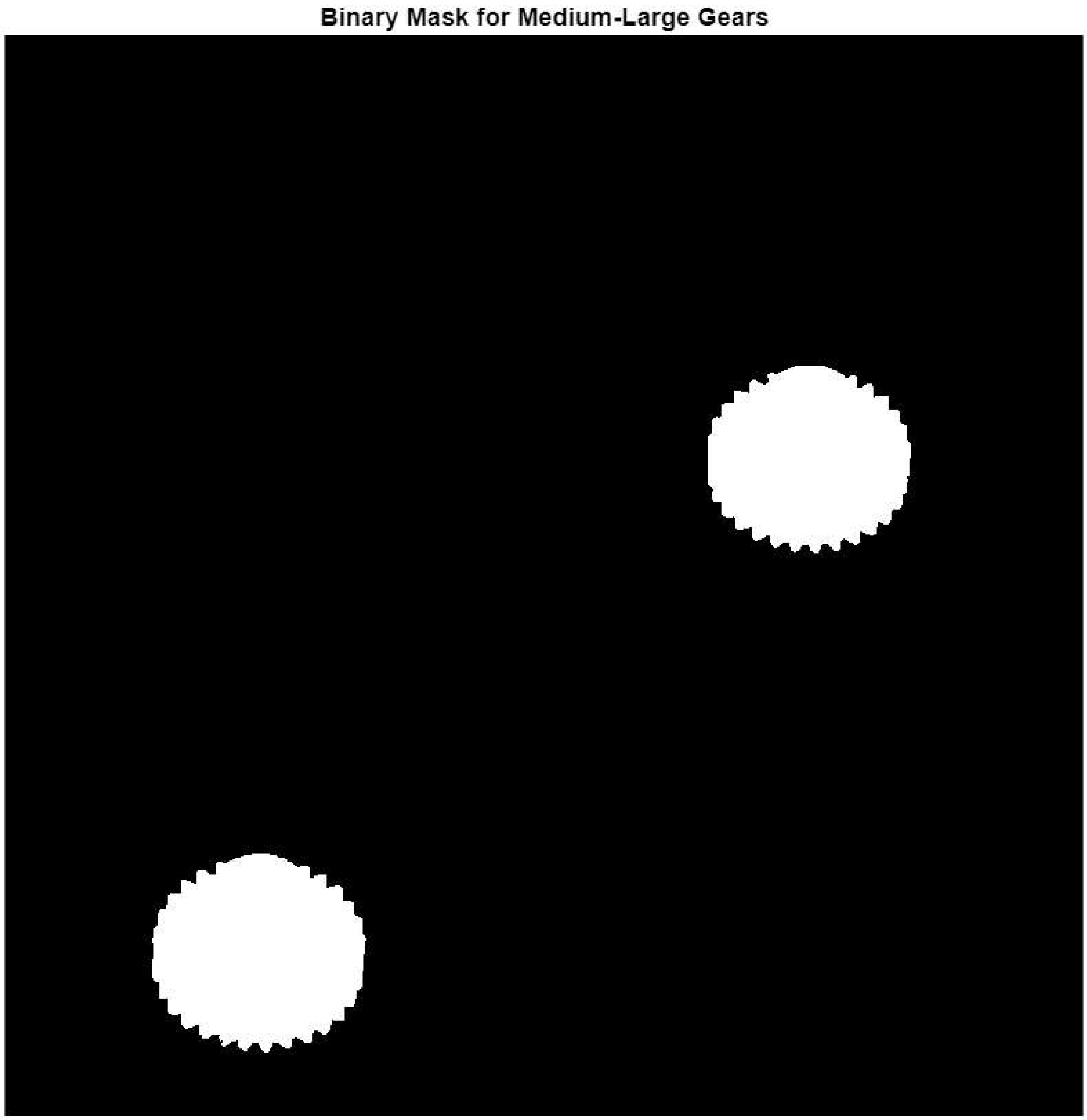
```
% Use the binary image mask to filter out the large gears from the grayscale image
% and show the original image with only the separated gears.

% Big Gears
gearImageBig = orig_gears .* uint8(binaryMaskBig);

figure;
imshow(gearImageBig)
title('Isolated Large Gears');
```

Isolated Large Gears

```
% Extract medium-large gears by selecting the blobs in the labeled image that meet
% the criteria and show the binary image mask with separated gears.

%Med-Big Gears

minAreaMedBig = 26000;
maxAreaMedBig = 30000;
ValidBlobsMedBig = [blobArea.Area] >= minAreaMedBig & [blobArea.Area] <= maxAreaMedBig;
validLabelsForMedBig = find(ValidBlobsMedBig)
```

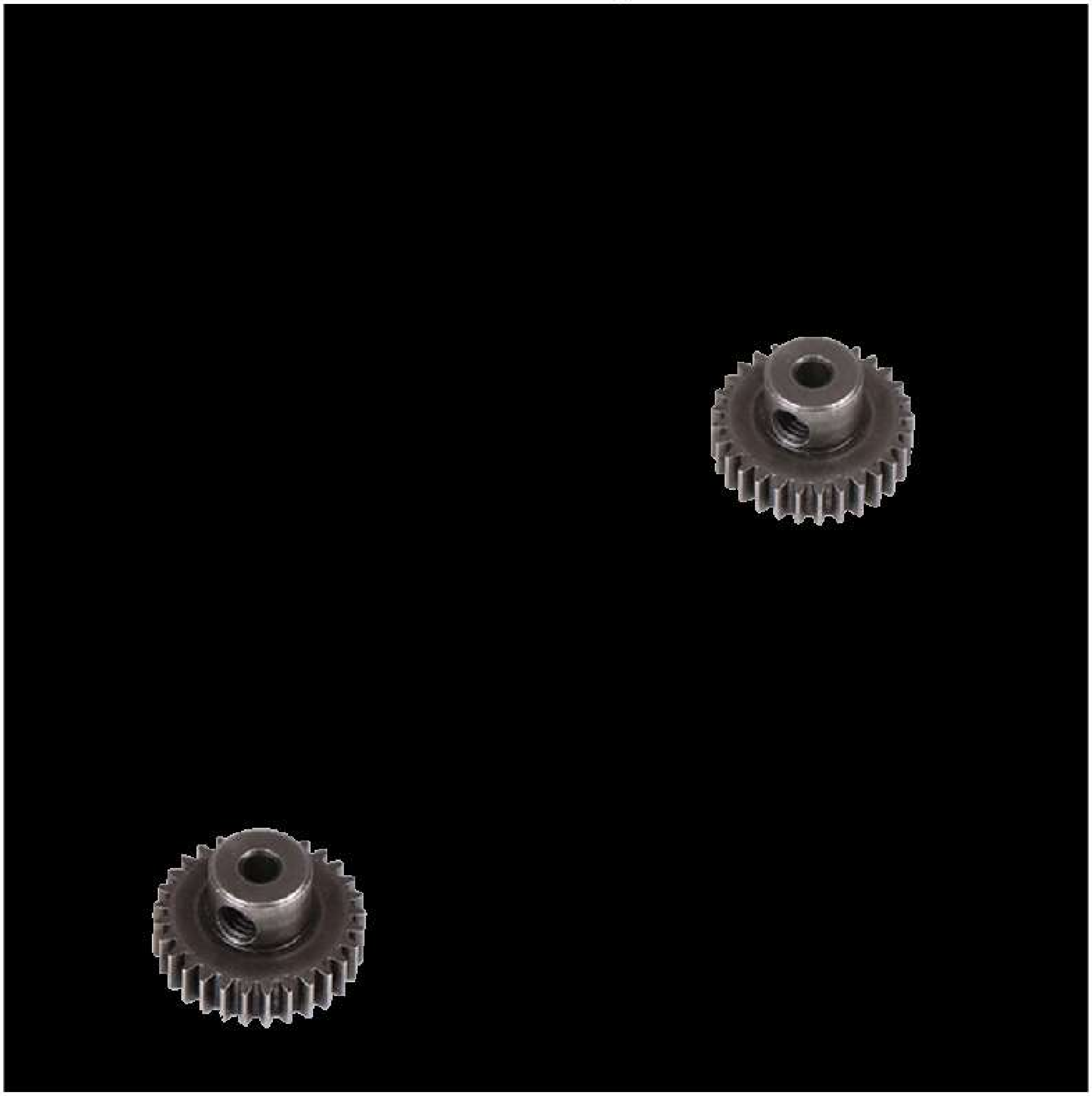
```
validLabelsForMedBig = 1×2
     3     6
```

```
binaryMaskMedBig = ismember(labeledImage, validLabelsForMedBig);

figure;
imshow(binaryMaskMedBig);
title('Binary Mask for Medium-Large Gears');
```




```
% Use the binary image mask to filter out the medium-large gears from the grayscale image
% and show the original image with only the separated gears.

gearImageMedBig = orig_gears .* uint8(binaryMaskMedBig);

figure;
imshow(gearImageMedBig)
title('Isolated Medium-Large Gears');
```

Isolated Medium-Large Gears

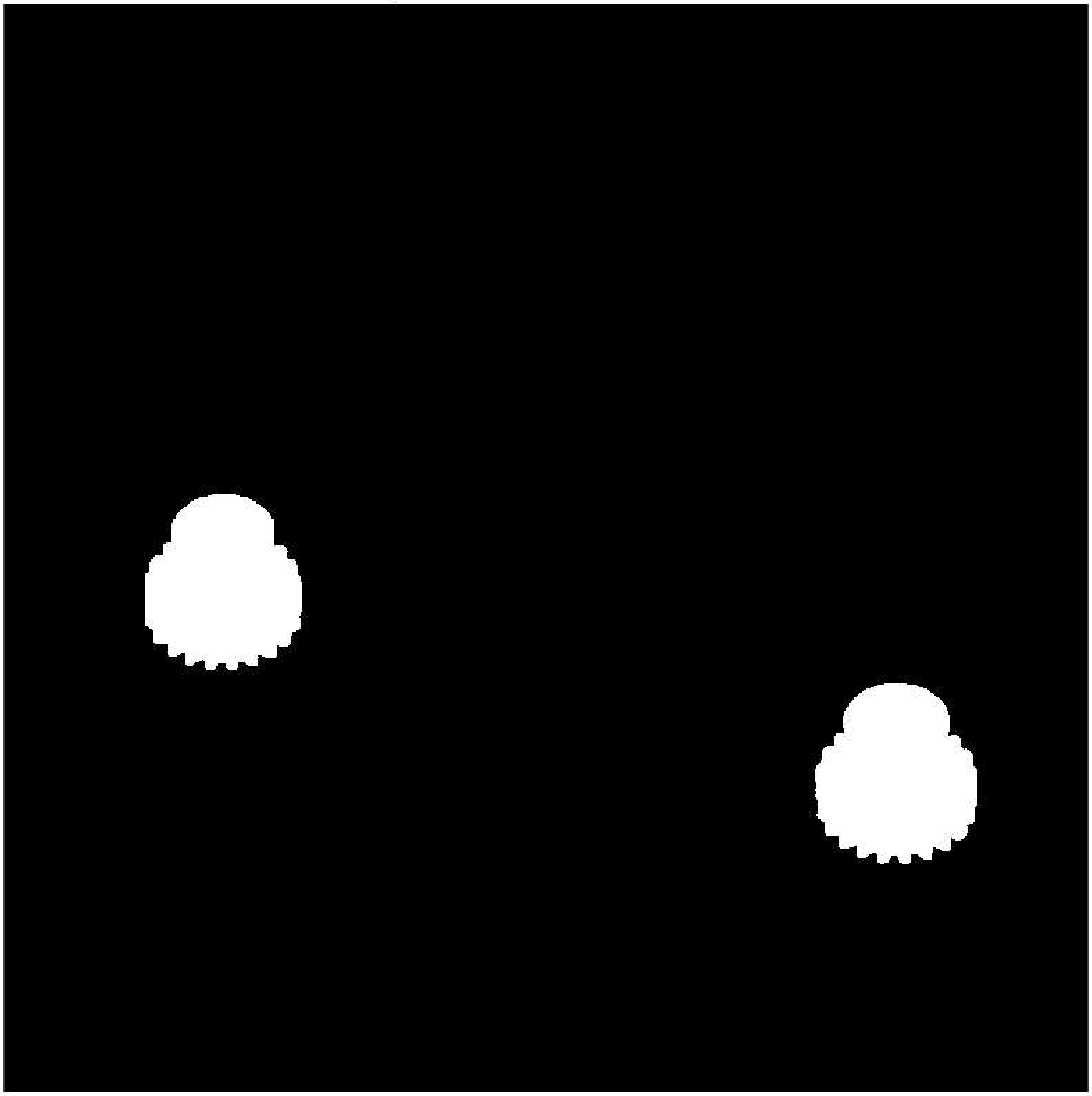


```
% Extract medium-small gears by selecting the blobs in the labeled image that meet
% the criteria and show the binary image mask with separated gears.

% Med-Small Gears

minAreaMedSmall = 18500;
maxAreaMedSmall = 20000;
ValidBlobsMedSmall = [blobArea.Area] >= minAreaMedSmall & [blobArea.Area] <= maxAreaMedSmall;
validLabelsForMedSmall = find(ValidBlobsMedSmall)
```

```
validLabelsForMedSmall = 1×2
     1     7
```

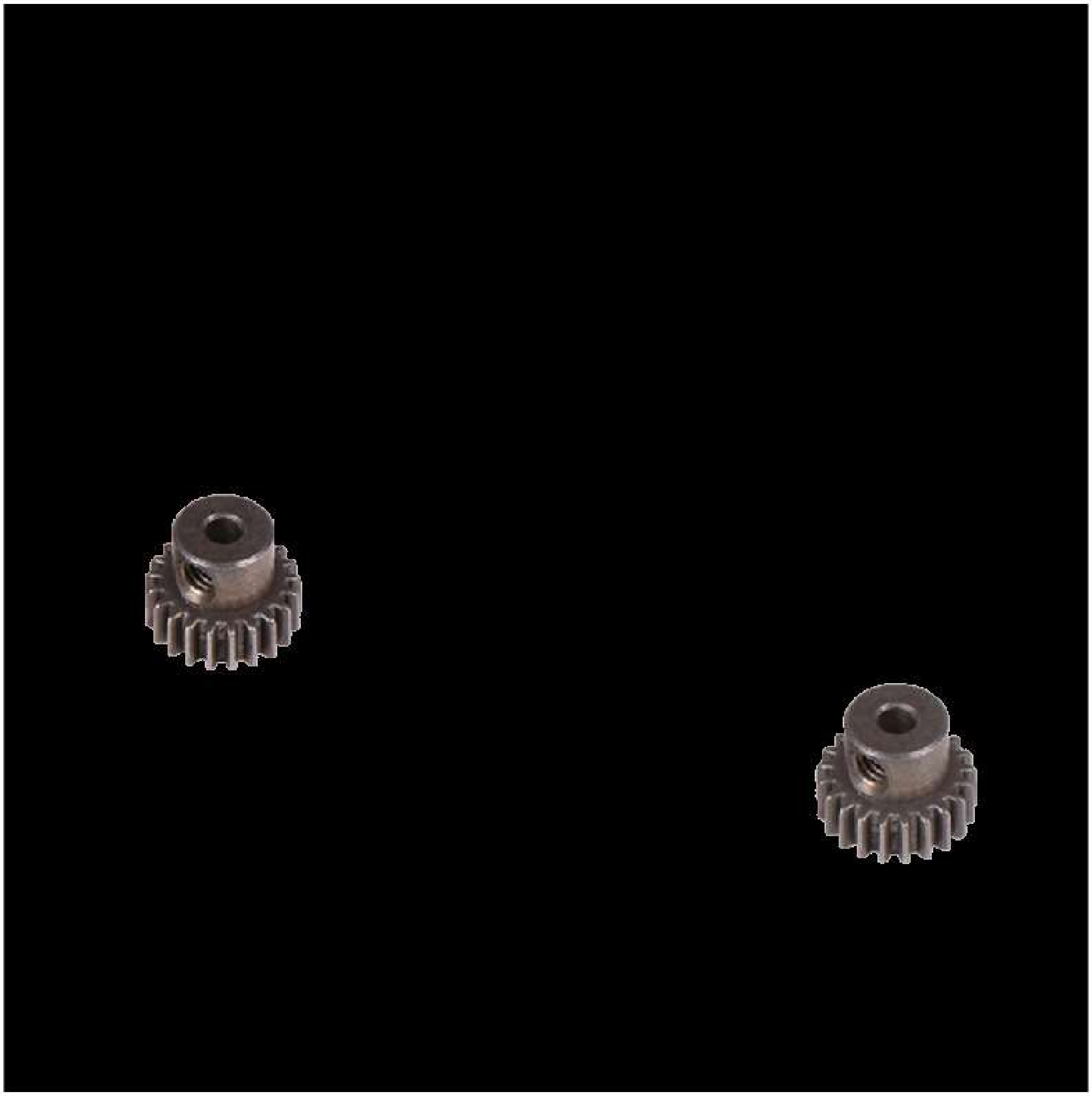
```
binaryMaskMedSmall = ismember(labeledImage, validLabelsForMedSmall);

figure;
imshow(binaryMaskMedSmall);
title('Binary Mask for Medium-Small Gears');
```

Binary Mask for Medium-Small Gears



```
% Use the binary image mask to filter out the medium-small gears from the grayscale image
% and show the original image with only the separated gears.

gearImageMedSmall = orig_gears .* uint8(binaryMaskMedSmall);

figure;
imshow(gearImageMedSmall)
title('Isolated Medium-Small Gears');
```

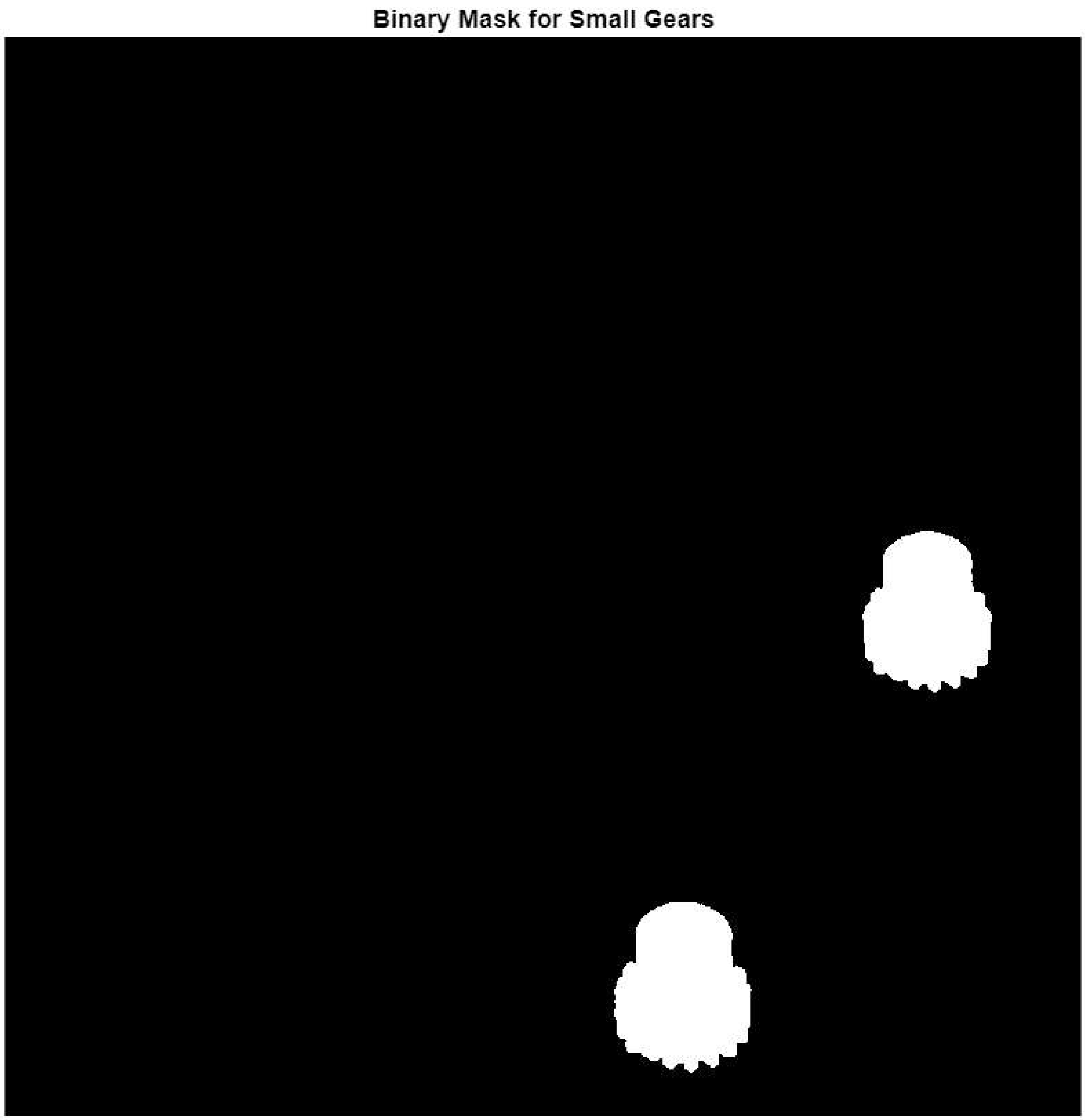
Isolated Medium-Small Gears



```
% Extract small gears by selecting the blobs in the labeled image that meet
% the criteria and show the binary image mask with separated gears.

% Small Gears

minAreaSmall = 14000;
maxAreaSmall = 16500;
ValidBlobsSmall = [blobArea.Area] >= minAreaSmall & [blobArea.Area] <= maxAreaSmall;
validLabelsForSmall = find(ValidBlobsSmall)
```

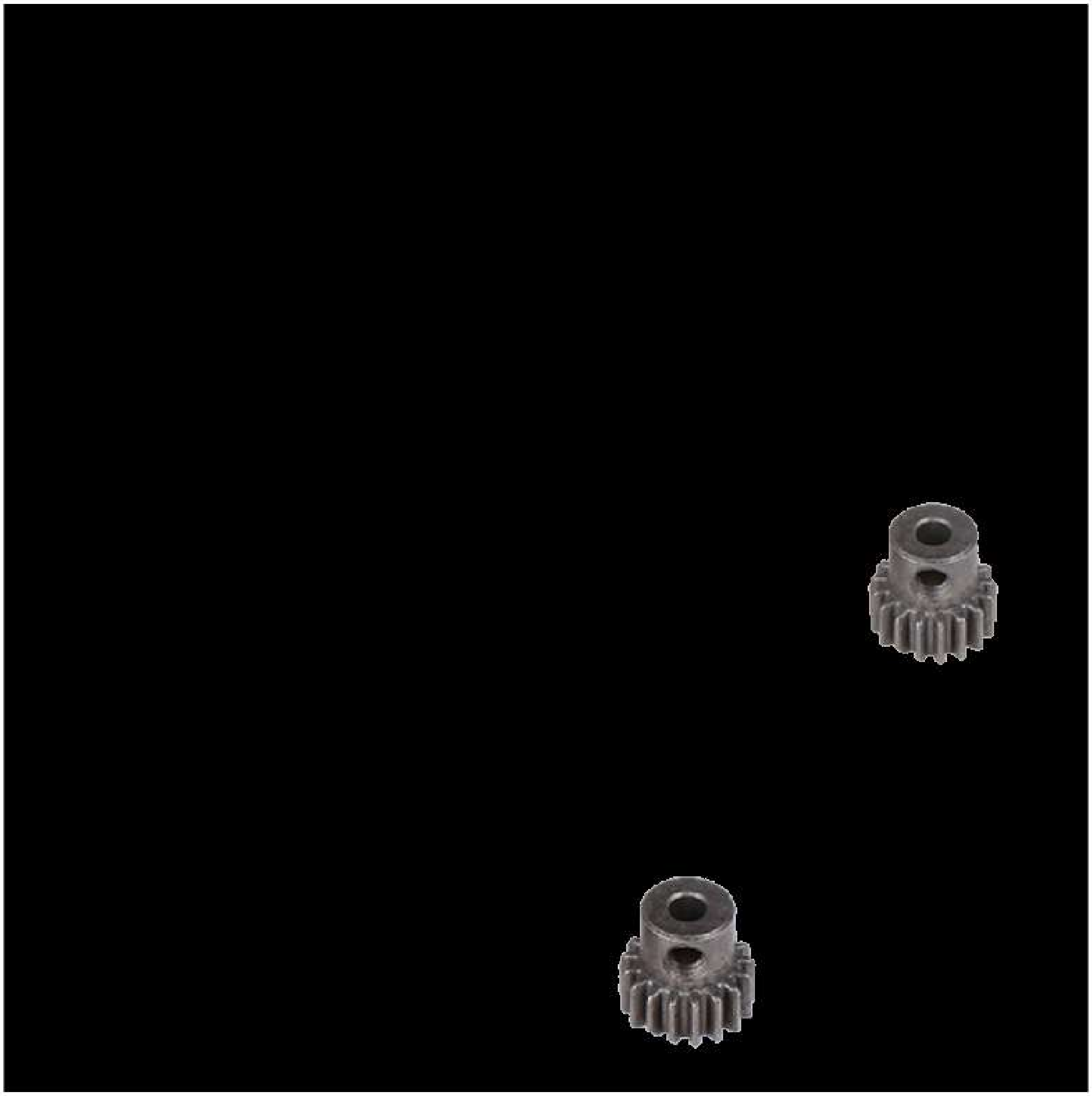
```
validLabelsForSmall = 1×2
     5     8
```

```
binaryMaskSmall = ismember(labeledImage, validLabelsForSmall);

figure;
imshow(binaryMaskSmall);
title('Binary Mask for Small Gears');
```




```
% Use the binary image mask to filter out the small gears from the grayscale image
% and show the original image with only the separated gears.

gearImageSmall = orig_gears .* uint8(binaryMaskSmall);

figure;
imshow(gearImageSmall)
title('Isolated Small Gears');
```

Isolated Small Gears



c) Find the brightest medium-large gear in the image

Identify the brightest medium-large gear in the image using the blob intensity and other properties obtained in part (a) and extract it in a separate image.

i. Check blob properties from part (a) and identify which criteria to use to separate the brightest medium-large gear.

```
% Compare the area of the gears using the blob properties to separate
% medium-large gears and set up the criteria to find the brightest one.

minAreaMedBig = 26000; % From Part B. Min threshold used to determine MedBig Gears


% Extracts the labels of the gears from the blobs within the given region.
medBigMask = ismember(labeledImage, find([blobregions.Area] >= minAreaMedBig & [blobregions.Area] < minAreaBig));
medBigGearLabels = unique(labeledImage(medBigMask))
```

```
medBigGearLabels = 2×1
     3
     6
```

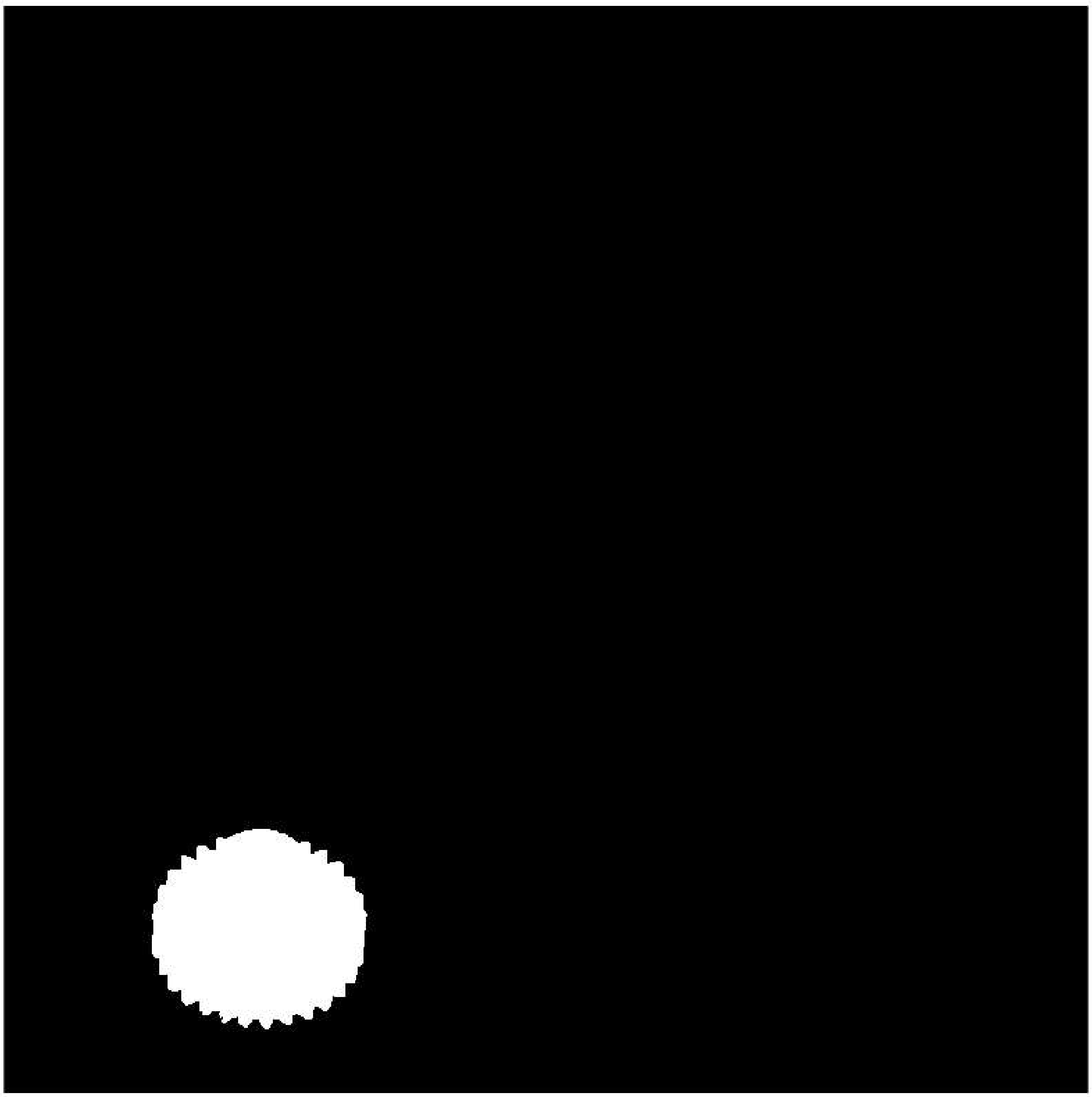
```
brightestMeanIntensity = 0;  % Initialize with a very small value
brightestMediumLargeGearLabel = 0;


for i = 1:numel(medBigGearLabels)
    gearLabel = medBigGearLabels(i);
    rowofLabel = propTable{:, 1} == gearLabel;
    meanIntensity = propTable{rowofLabel, 6};
```

```
    if max(meanIntensity) > brightestMeanIntensity
        brightestMediumLargeGearLabel = gearLabel;
        brightestMeanIntensity = max(meanIntensity);
    end
end
fprintf("Brightest Gear is %d\n", brightestMediumLargeGearLabel);
```

```
Brightest Gear is 3
```

ii. Use the `ismember` function to extract the identified blobs in the labeled image that was generated in part (a) using the `bwlabel` command. Create a binary mask from the labeled image and filter the brightest medium-large gear for display in the original grayscale image.

```
% Extract the brightest medium-large gear by selecting the blobs in the labeled image that
% meet the criteria and show the binary image mask with separated gear.

brightestMedBigGear = ismember(labeledImage, brightestMediumLargeGearLabel);

imshow(brightestMedBigGear);
```



```
% Use the binary image mask to filter out the brightest medium-large gear from the
% grayscale image and show the original image with only that gear.

imshow(brightestMediumLargeGearGray)
```