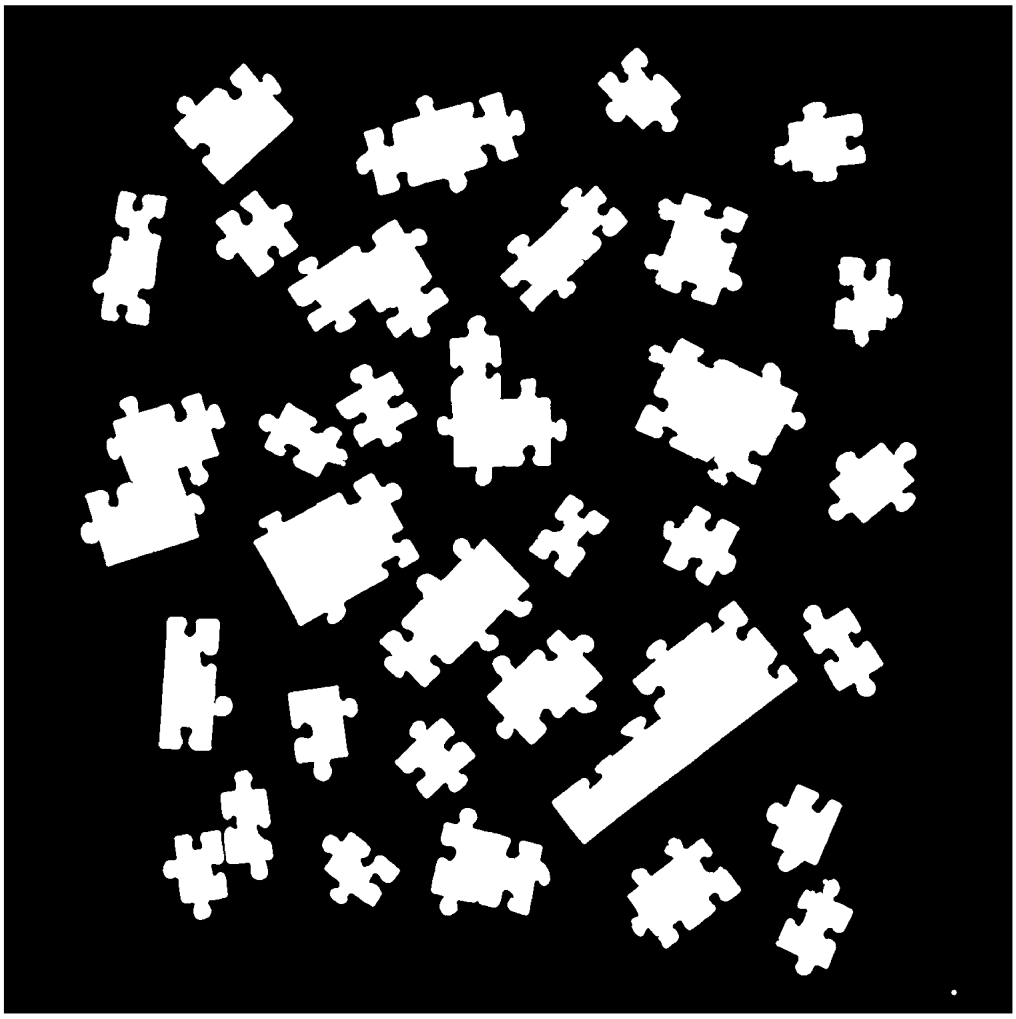
Analyze Puzzle Region Shape

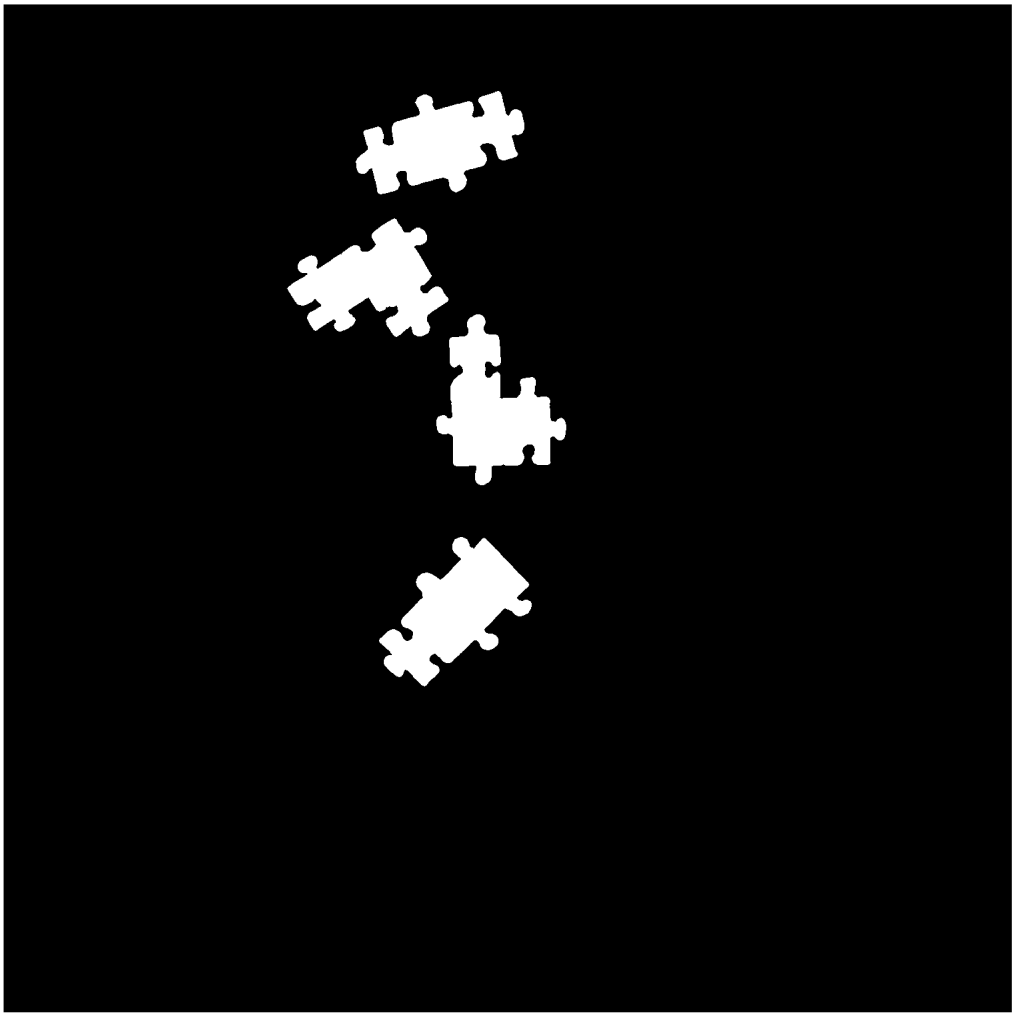
For this problem your code will need to do the following:

- Create a mask which will segment only the regions consisting of 3 joined puzzle pieces (use region filtering to modify the mask created by the provided function). Assign your answer to the variable name **`threePieceMask`**.
- Calculate the Area and Eccentricity of each region in **`threePieceMask`**. Provide the results in a table variable **`threePieceProperties`**.
- Filter the **`threePieceMask`** to include only the regions with three pieces connected in a straight line. Assign the result to **`threePieceLinearMask`**. **Hint:** You can use thresholds for Eccentricity with the bwpropfilt function to isolate the straight-line connected regions in **`threePieceMask`**.
- Find the coordinates of a bounding box for each of the regions in **`threePieceLinearMask`**. Provide the results in a table variable **`threePieceLinearBoxes`**. **Hint:** Use the **`regionprops`** function on **`threePieceLinearMask`** with the syntax shown in this example, and the `BoundingBox` property.
- To apply the bounding boxes to the original image and see the results, uncomment the provided template code.

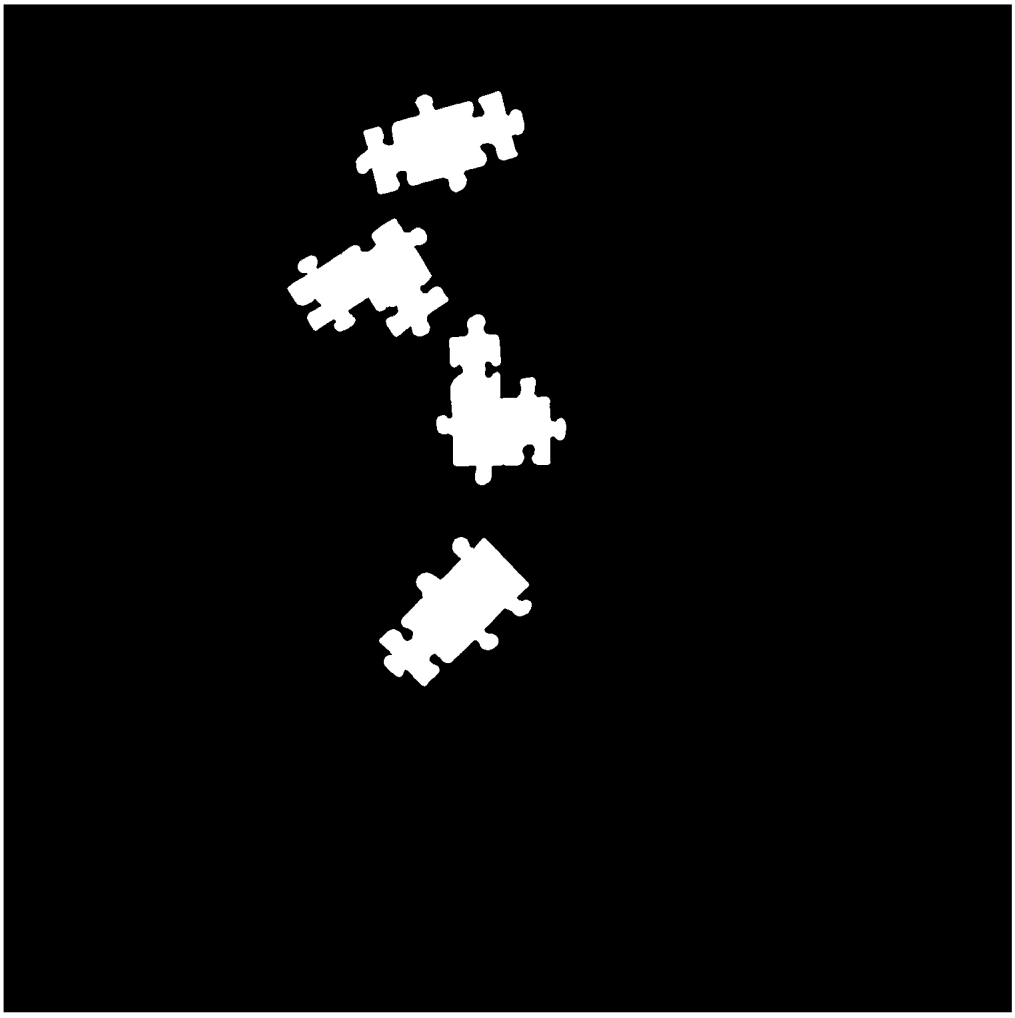
```
puzzleImage = imread("Puzzle_34.jpg");
puzzleMask = segmentPuzzle34(puzzleImage);

puzzleImage = imread("Puzzle_34.jpg");
puzzleMask = segmentPuzzle34(puzzleImage);
imshow(puzzleMask);
```



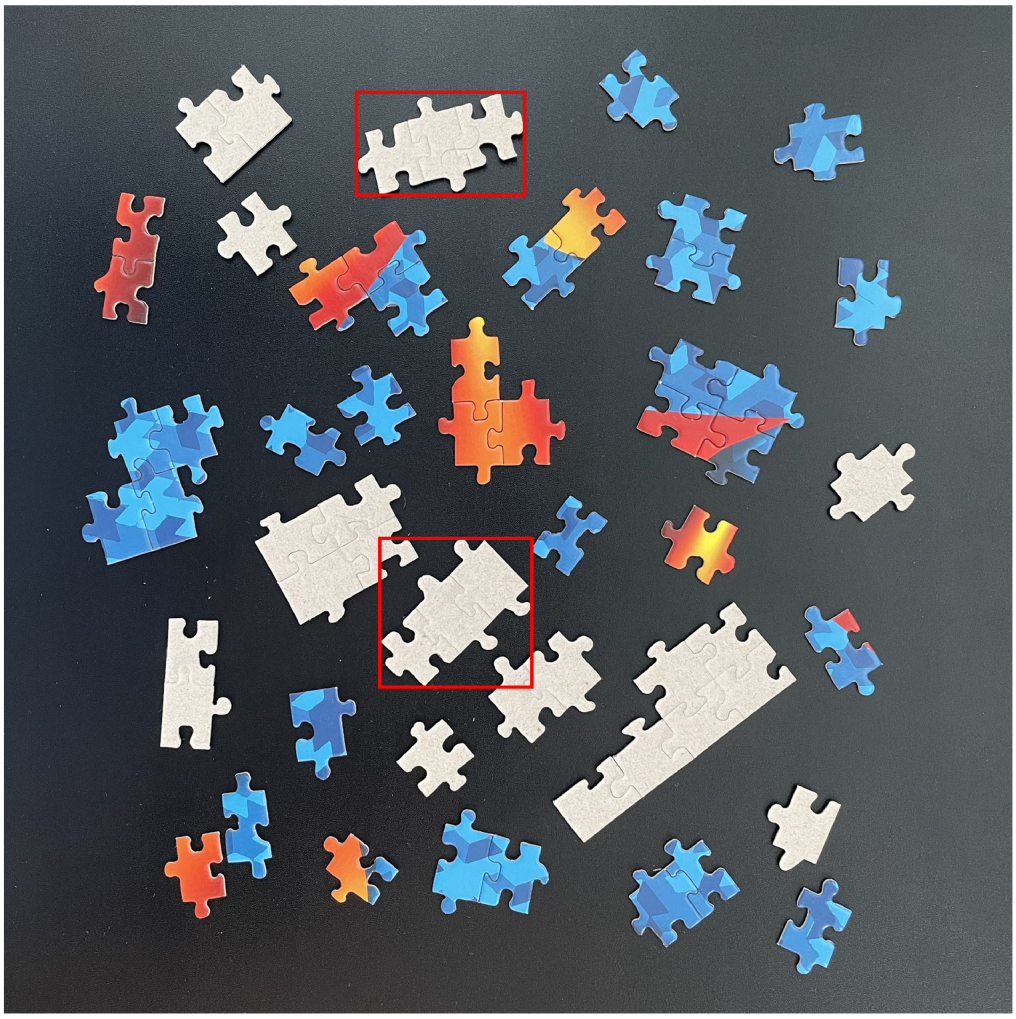
```
[threePieceMask,threePieceProperties] = threepfilter(puzzleMask);
imshow(threePieceMask);
```



```
puzzleImage = imread("Puzzle_34.jpg");
puzzleMask = segmentPuzzle34(puzzleImage);

[threePieceMask,threePieceProperties] = threepfilter(puzzleMask);
imshow(threePieceMask);
```



```
threePieceLinearMask = bwpropfilt(threePieceMask,"Eccentricity",[0.80,1]);

threePieceLinearBoxes = regionprops("table",threePieceLinearMask,"BoundingBox");


%% Uncomment the lines below to add your bounding boxes to the original image and visualize the
puzzleImageBoxed = insertShape(puzzleImage,"Rectangle",threePieceLinearBoxes.BoundingBox,"color
figure, imshow(puzzleImageBoxed);
```



```
function [threePieceMask,threePieceProperties] = threepfilter(BW_in)
%filterRegions Filter BW image using auto-generated code from imageRegionAnalyzer app.

% Auto-generated by imageRegionAnalyzer app on 31-Dec-2022
%-----------------------------------------------------------

threePieceMask = BW_in;

% Filter image based on image properties.
threePieceMask = bwpropfilt(threePieceMask,'Area',[21000, 25000]);

% Get properties.
threePieceProperties = regionprops(threePieceMask, {'Area', 'Eccentricity'});
threePieceProperties = struct2table(threePieceProperties);
end
```

```
function [BW,maskedImage] = segmentPuzzle34(X)
%segmentImage Segment image using auto-generated code from imageSegmenter app
%  [BW,MASKEDIMAGE] = segmentImage(X) segments image X using auto-generated
%  code from the imageSegmenter app. The final segmentation is returned in
%  BW, and a masked image is returned in MASKEDIMAGE.

% Auto-generated by imageSegmenter app on 29-Aug-1997
%----------------------------------------------------

X = im2gray(X);

% Adjust data to span data range.
X = imadjust(X);

% Threshold image - adaptive threshold
BW = imbinarize(X, 'adaptive', 'Sensitivity', 0.500000, 'ForegroundPolarity', 'bright');

% Erode mask with disk
radius = 2;
decomposition = 0;
se = strel('disk', radius, decomposition);
BW = imerode(BW, se);
% BW = bwpropfilt(BW,"Area",[300,inf]);

% Dilate mask with disk
radius = 4;
decomposition = 0;
se = strel('disk', radius, decomposition);
BW = imdilate(BW, se);

% Clear borders
BW = imclearborder(BW);

% Fill holes
BW = imfill(BW, 'holes');

% Create masked image.
maskedImage = X;
maskedImage(~BW) = 0;
end
```