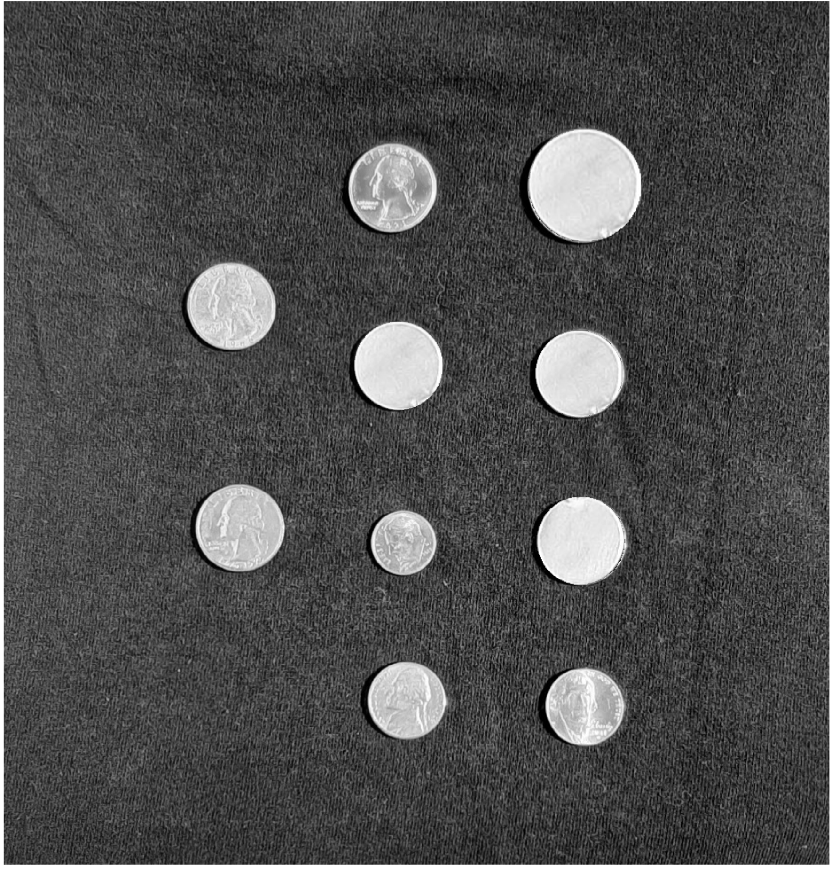
Final Analysis

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

- Accurately determine the number of each coin type present. Use variable names **nDimes**, **nNickels**, **nQuarters**, and **nFiftyCents**.
- Calculate the total $ value of coins present. Use variable name **USD**.

```
testCoinImage = imread("testCoinImage2.png");
imshow(testCoinImage);
title("Original Coin Image");
```

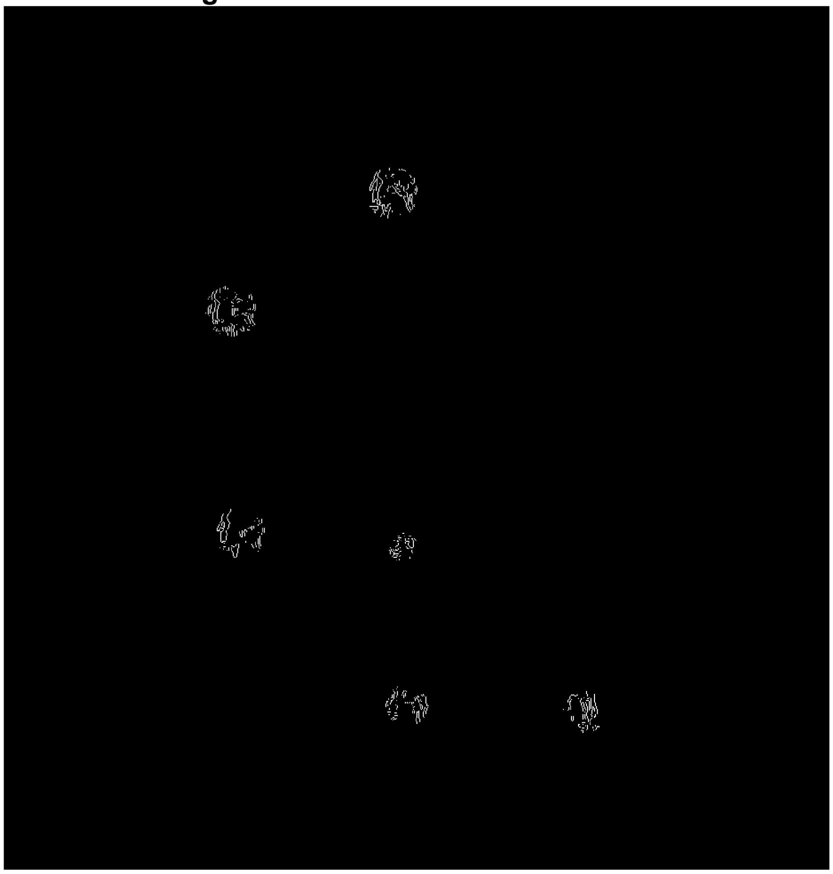
Original Coin Image



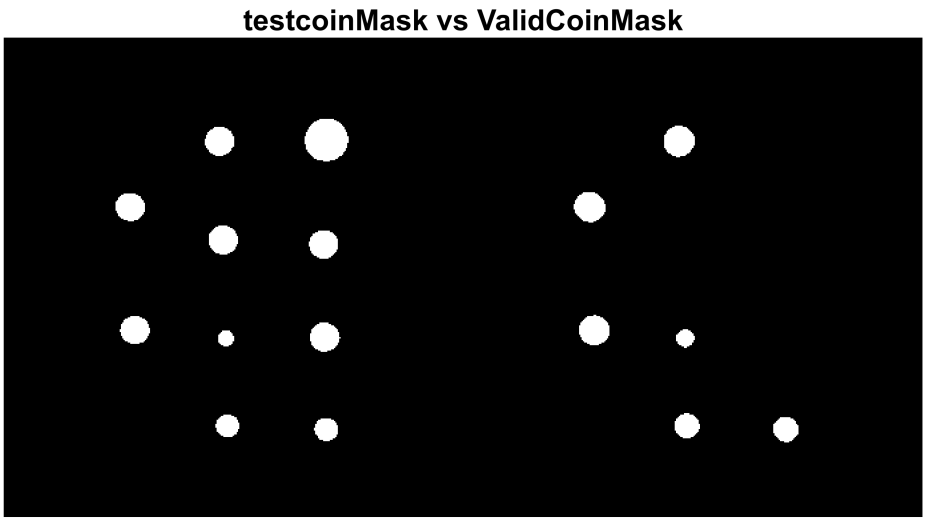
```
[testcoinMask,MaskedtestCoin] = segmentCoin(testCoinImage);
% Shrink the coin mask.
se = strel('disk', 20, 0);
testcoinMask = imfill(testcoinMask, 'holes'); % Fill any holes in it.
testcoinMask = imerode(testcoinMask, se); % Shrink by 3 layers of pixels.

% Find edges using original poster's code.
imgFilt = imgaussfilt(MaskedtestCoin,0.5,...
    Padding="circular",FilterDomain="frequency",FilterSize=3);
```

```
faceEdgeMask = edge(imgFilt,"sobel",0.05,"both");
% Erase outside the shrunken coin mask to get rid of outer boundary.
faceEdgeMask(~testcoinMask) = false;
imshow(faceEdgeMask);
title("Edge Mask Detection for Valid Coins")
```

Edge Mask Detection for Valid Coins



```
see = strel("disk",25,0);
fb = imfill(faceEdgeMask,"holes");
Bw2 = imdilate(fb,see);
%imshow(Bw2)
validCoinMask = Bw2 & testcoinMask;
set = strel("disk",2,0);
validCoinMask = imdilate(validCoinMask,set);
montage({testcoinMask,validCoinMask});
title("testcoinMask vs ValidCoinMask");
```



testcoinMask vs ValidCoinMask

```
%imshow(validCoinMask);

coinSizes = regionprops("table",validCoinMask,"Area");
nDimes = coinSizes.Area < 1100;
nDimes = sum(nDimes);
nNickels = (coinSizes.Area > 1100 & coinSizes.Area <2200);
nNickels = sum(nNickels);
nQuarters = (coinSizes.Area > 2200 & coinSizes.Area < 3200);
nQuarters = sum(nQuarters);
nFiftyCents = coinSizes.Area > 3200;
nFiftyCents = sum(nFiftyCents);
USD = (nDimes * 0.10) + (nNickels * 0.05) + ...
(nQuarters * 0.25) + (nFiftyCents * 0.50)
```

```
USD = 0.9500
```

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

% Auto-generated by imageSegmenter app on 31-Dec-2022
%----------------------------------------------------
X = im2gray(X);

% Threshold image - manual threshold
testcoinMask = im2gray(X) > 150;

% Close mask with default
radius = 12;
```

```
decomposition = 4;
se = strel('disk', radius, decomposition);
testcoinMask = imclose(testcoinMask, se);

% Create masked image.
MaskedtestCoin = X;
MaskedtestCoin(~testcoinMask) = 0;
end
```