CS543 Assignment 2

Name: Huey-Chii Liang

NetID: hcliang2

Part 1 Fourier-based Alignment:

A: Channel Offsets

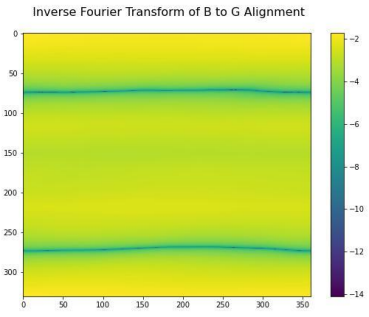
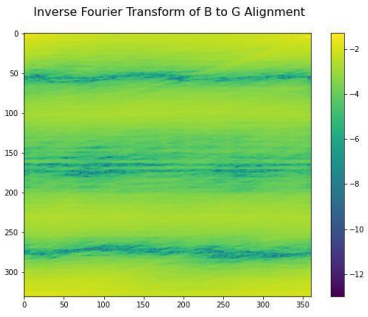
Low-resolution images (using channel <G> as base channel):

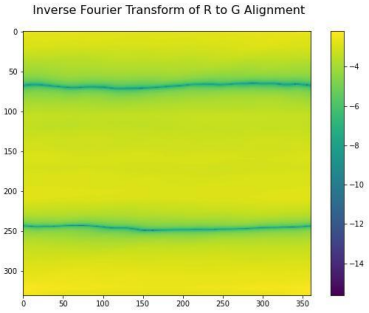
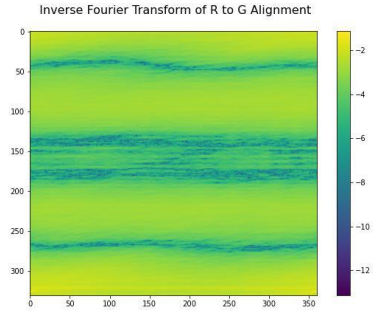
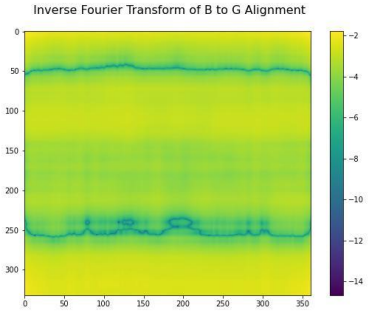
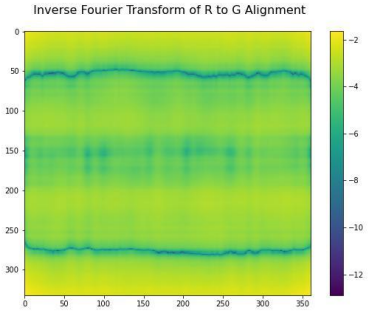
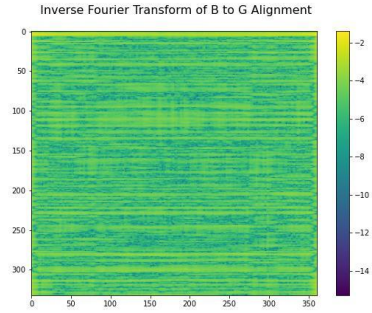
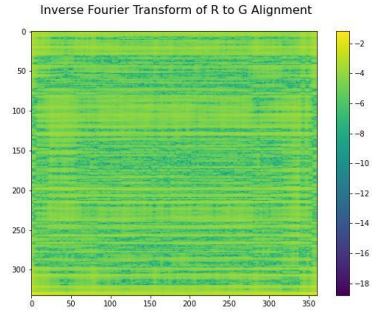
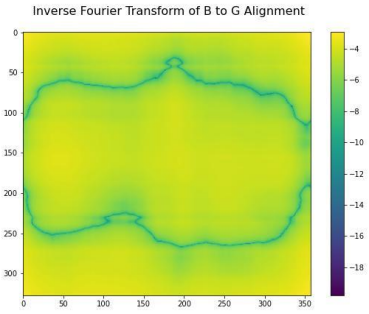
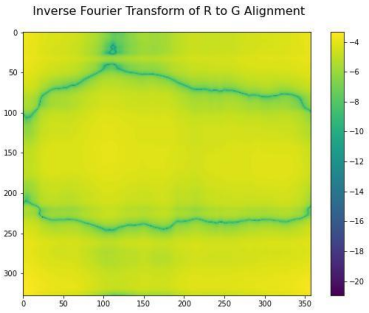
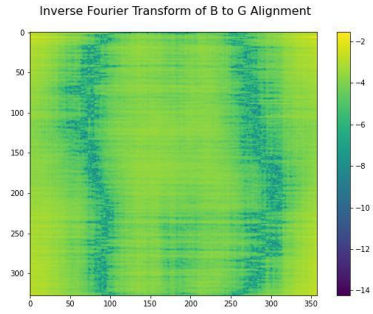
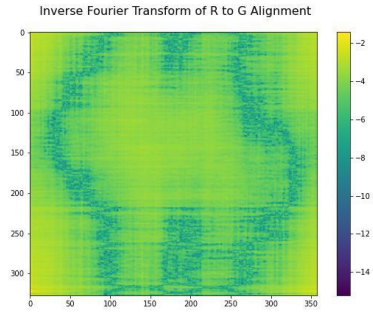
Image	<B> (h,w) offset	<R> (h,w) offset
00125v.jpg	[4, -2]	[-6, -1]
00149v.jpg	[4, -2]	[-3, 0]
00153v.jpg	[6, -3]	[-6, 2]
00351v.jpg	[14, 0]	[-9, 1]
00398v.jpg	[5, -2]	[-4, 1]
01112v.jpg	[10, 0]	[-5, 1]

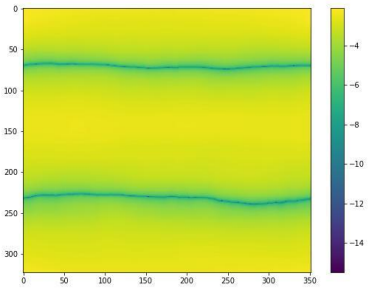
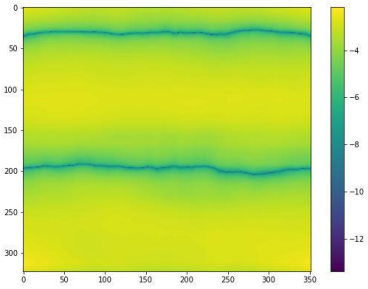
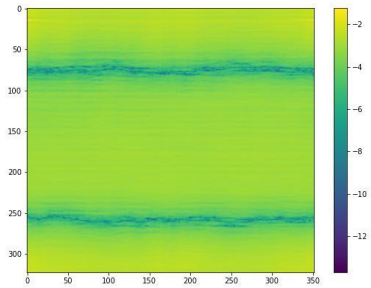
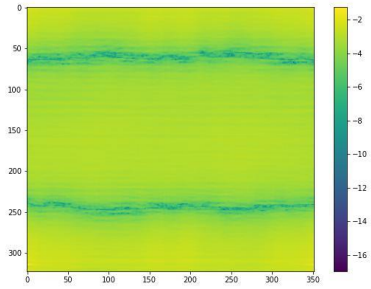
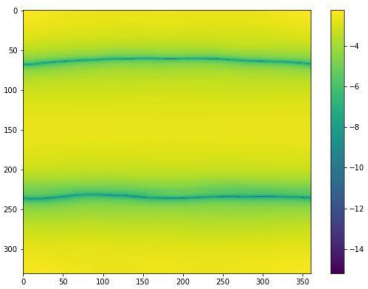
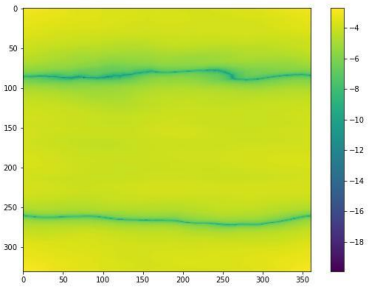
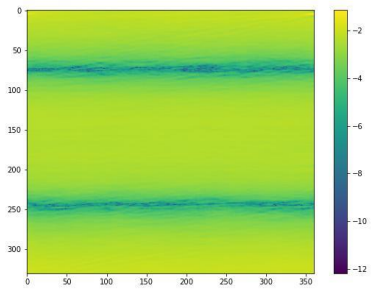
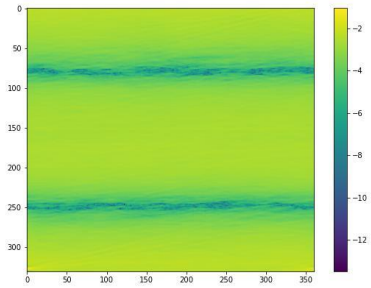
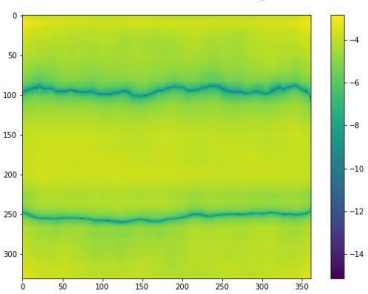
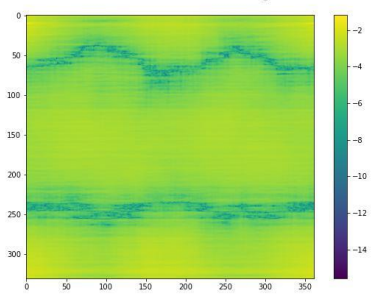
High-resolution images (using channel G as base channel):

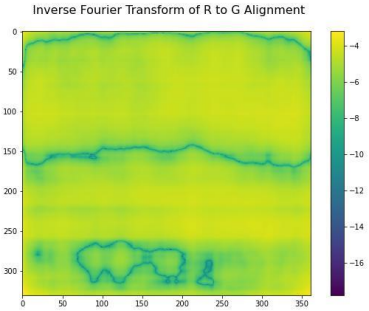
Image	<B> (h,w) offset	<R> (h,w) offset
01047u.tif	[65, -21]	[-44, 14]
01657u.tif	[24, -8]	[-18, 3]
01861a.tif	[23, -37]	[-17, 24]

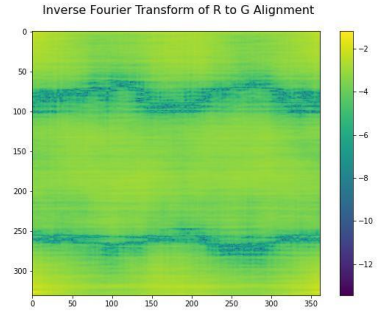
B: Output Visualizations

Aligned image	Inverse Fourier transform output **without** preprocessing	Inverse Fourier transform output **with** preprocessing
00125v.jpg 		

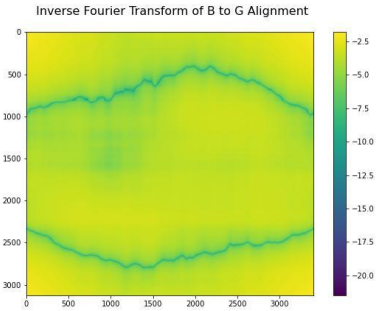
		
00149v.jpg 	 	 
00153v.jpg 	 	 

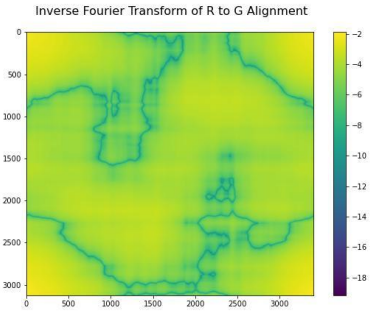
00351v.jpg 	Inverse Fourier Transform of B to G Alignment  Inverse Fourier Transform of R to G Alignment 	Inverse Fourier Transform of B to G Alignment  Inverse Fourier Transform of R to G Alignment 
00398v.jpg 	Inverse Fourier Transform of B to G Alignment  Inverse Fourier Transform of R to G Alignment 	Inverse Fourier Transform of B to G Alignment  Inverse Fourier Transform of R to G Alignment 
01112v.jpg	Inverse Fourier Transform of B to G Alignment 	Inverse Fourier Transform of B to G Alignment 

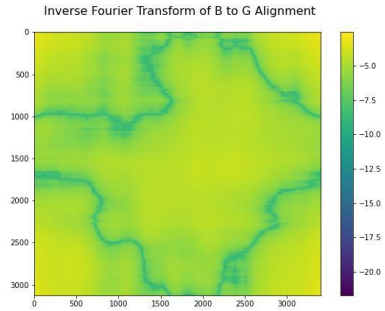


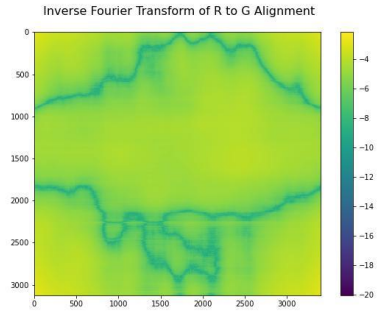


01047u.tif

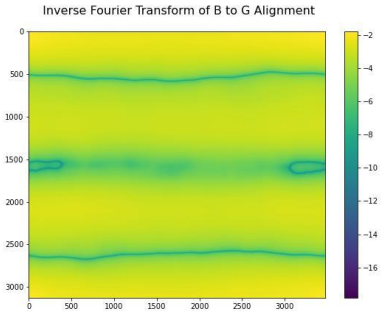


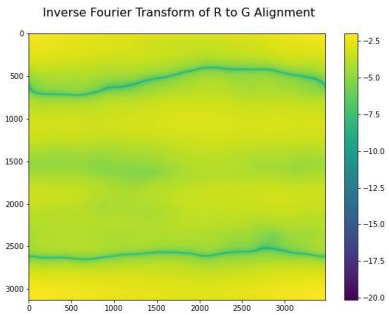


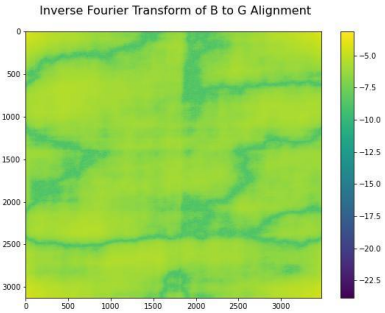


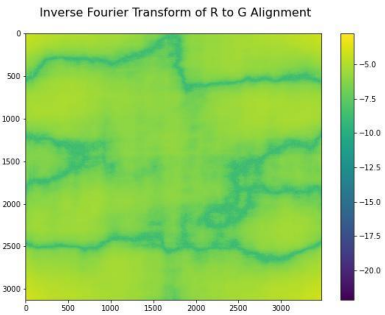


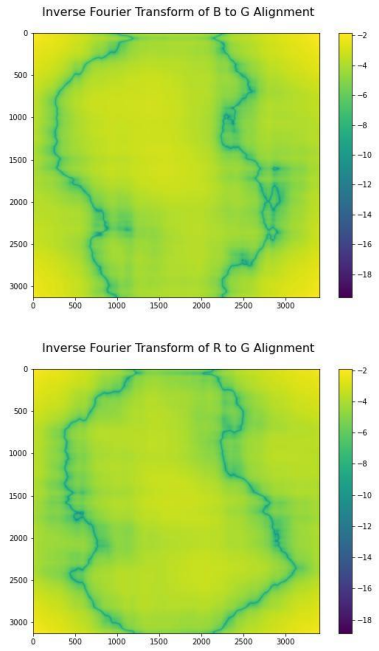
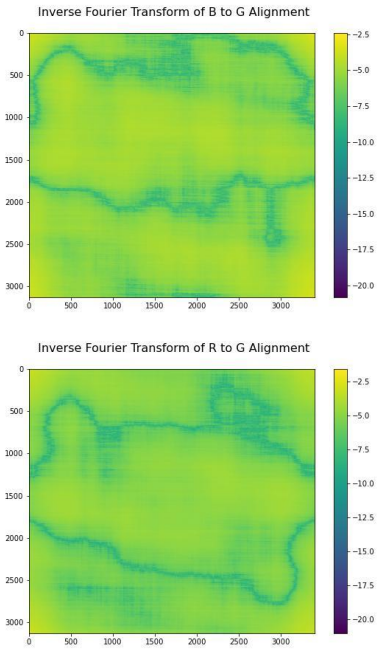
01657u.tif









01861a.tif 		

C: Discussion and Runtime Comparison

I used a Laplacian of Gaussian filter with sigma = 1 to highlight edges in each color channel, and it really helped achieve better Fourier-based alignment results. For the image “01112v.jpg”, it helped to generate a better aligned image when using B and R as base channels. The comparison is shown in the table below:

	Using <B> as base channel	Using <R> as base channel
Without LoG filtering		

With LoG filtering		

Running time improvement:
Fourier-based alignment average runtime = 6.7 seconds
Multiscale alignment average runtime = 170 seconds
Single-scale alignment average runtime > 1 hour

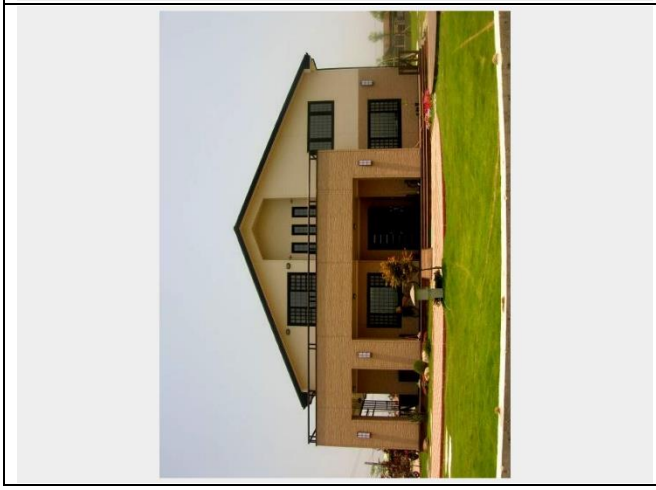
Part 2 Scale-Space Blob Detection:

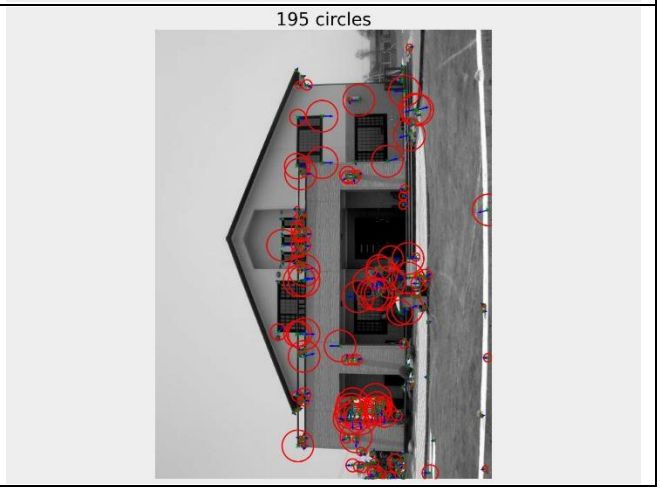
Example 1:

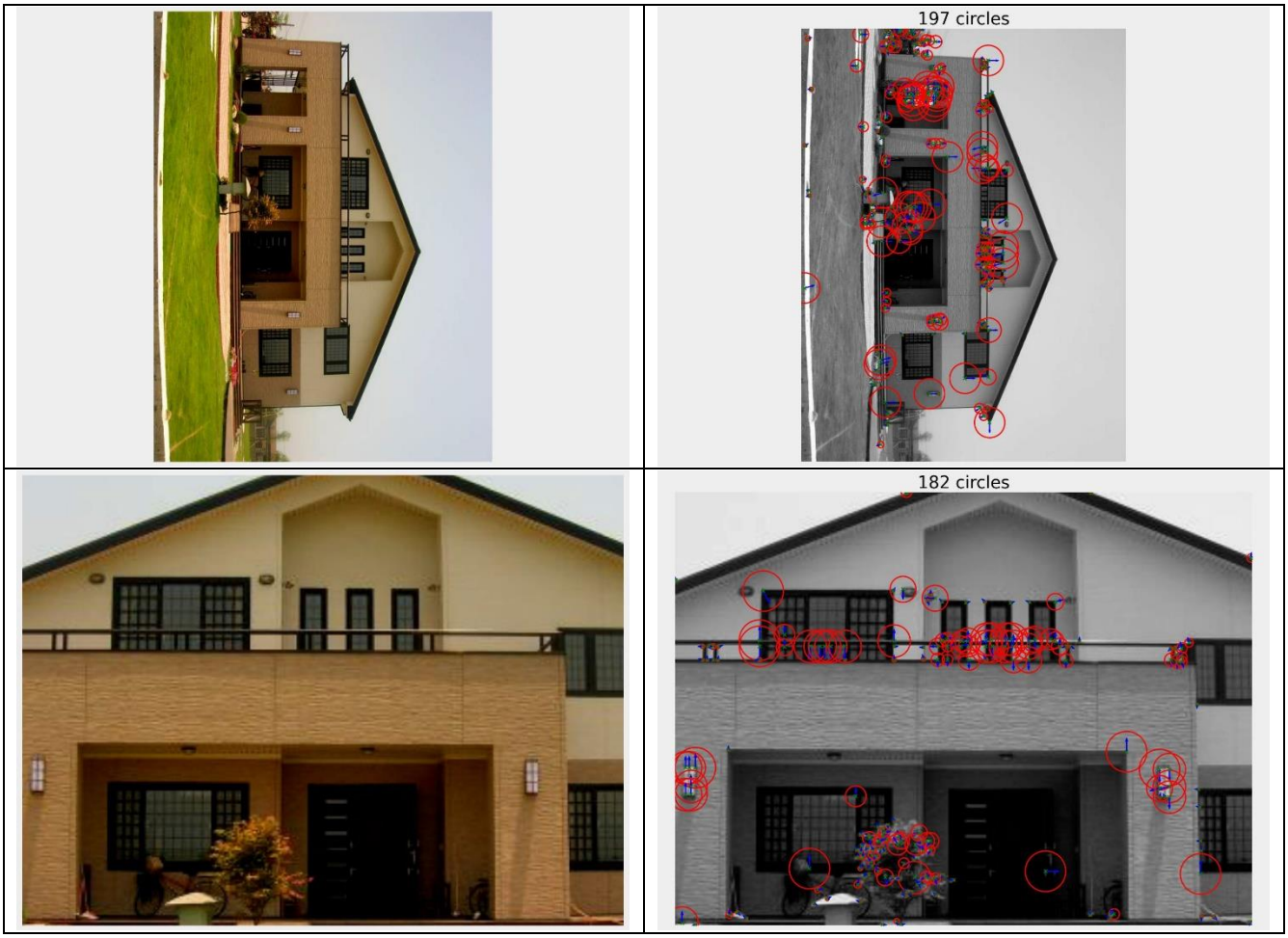
Raw images	Blob detection
	


123 circles

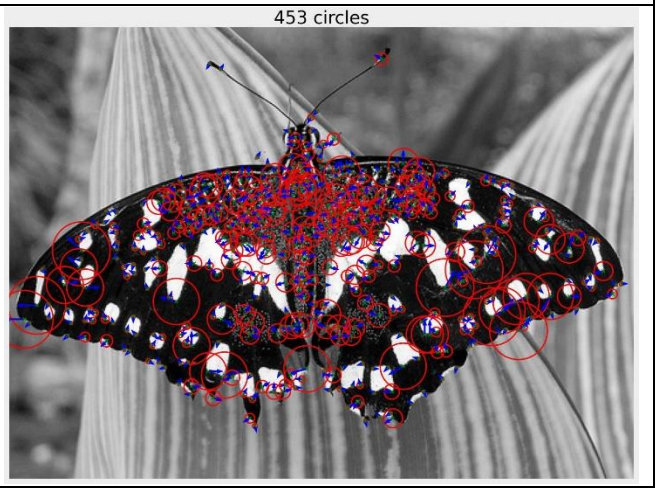

165 circles

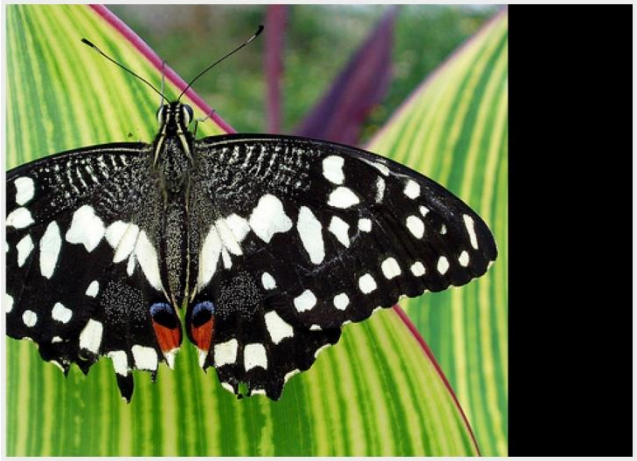


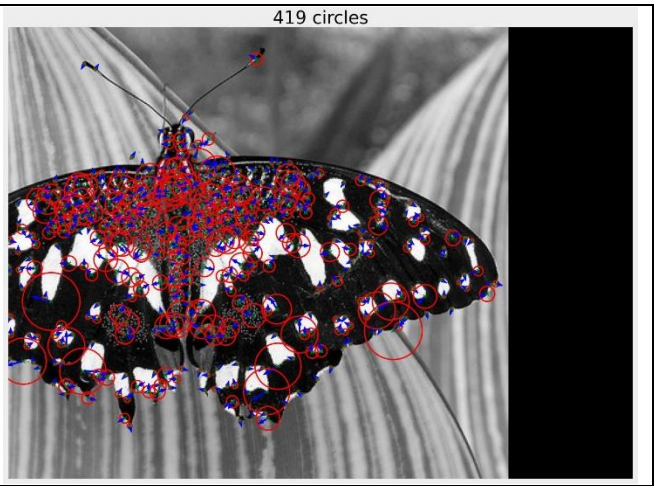

195 circles

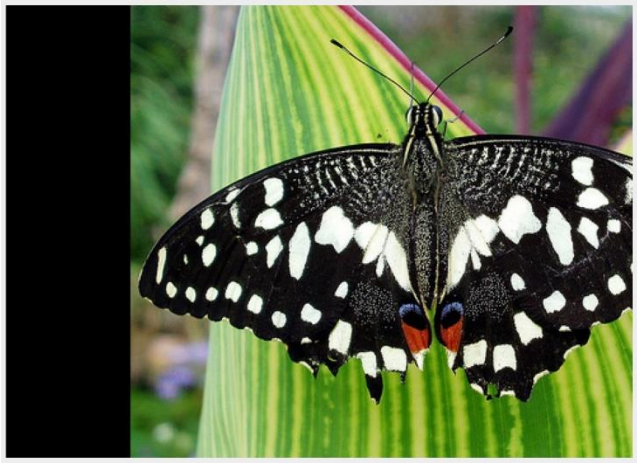


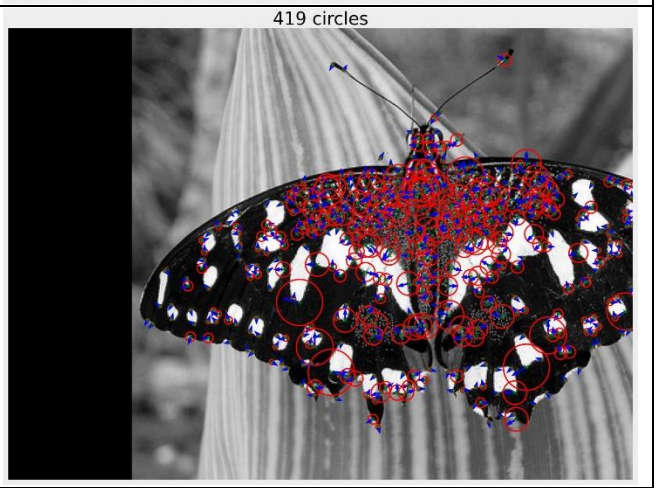
Example 2:

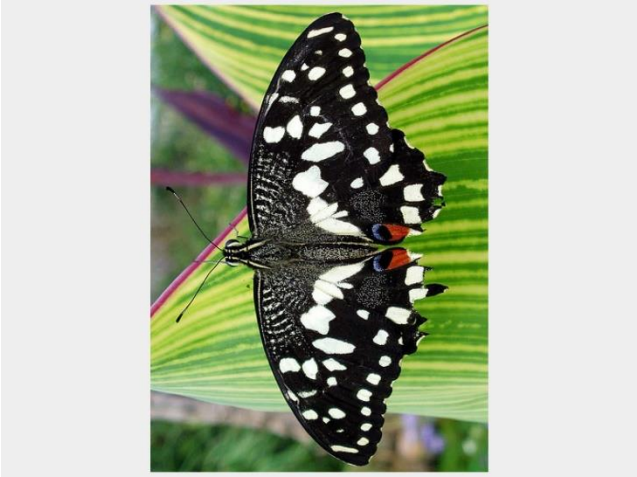
Raw images	Blob detection
	

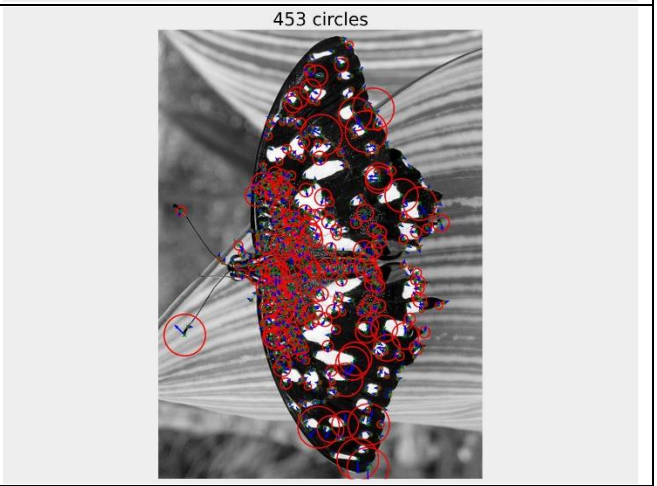


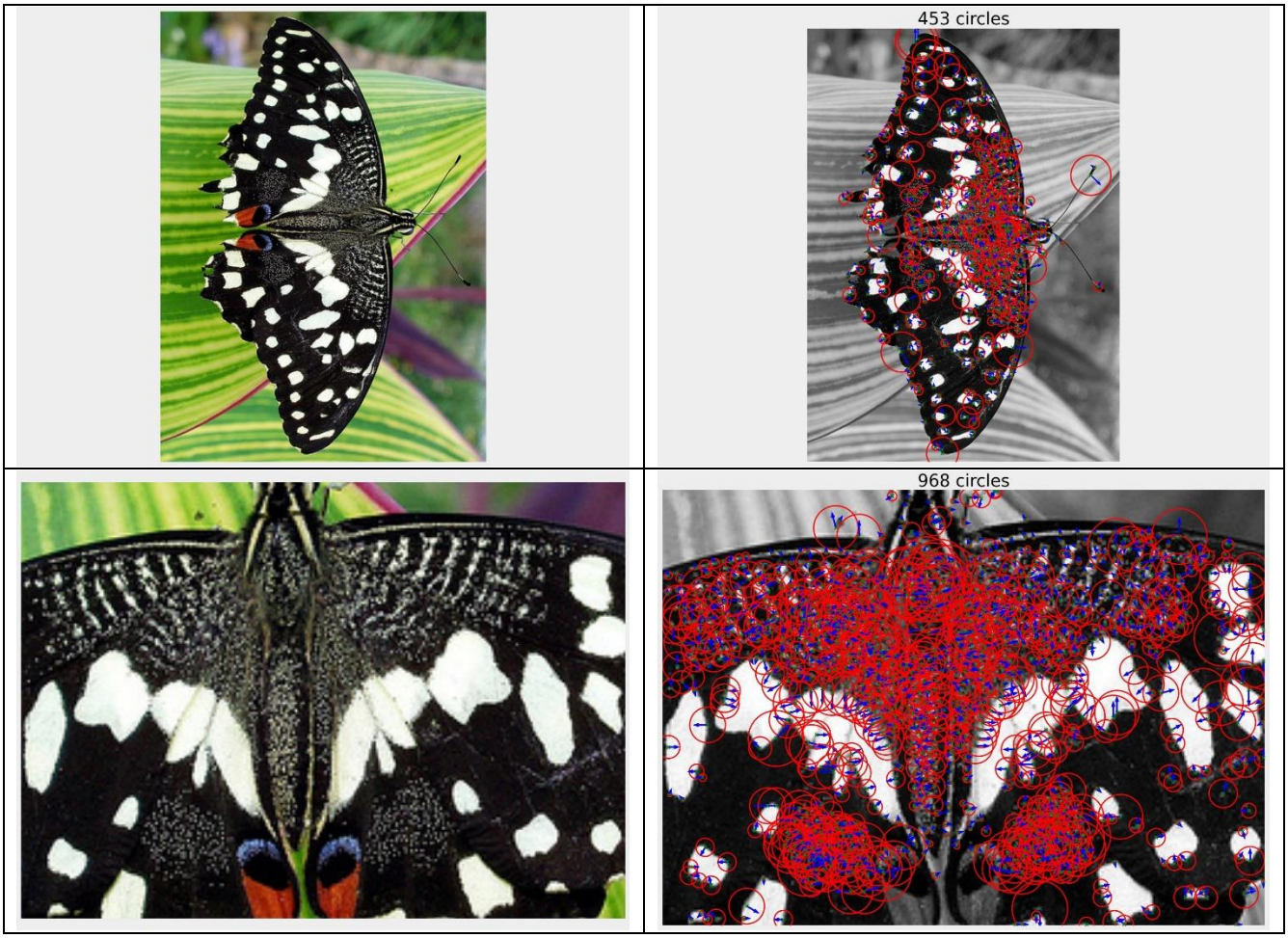

419 circles



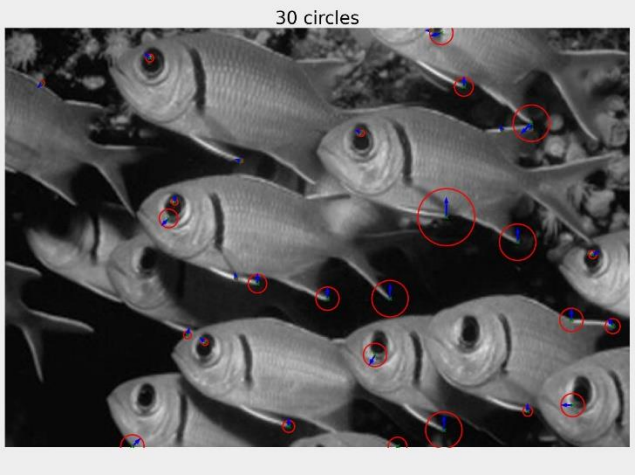

419 circles



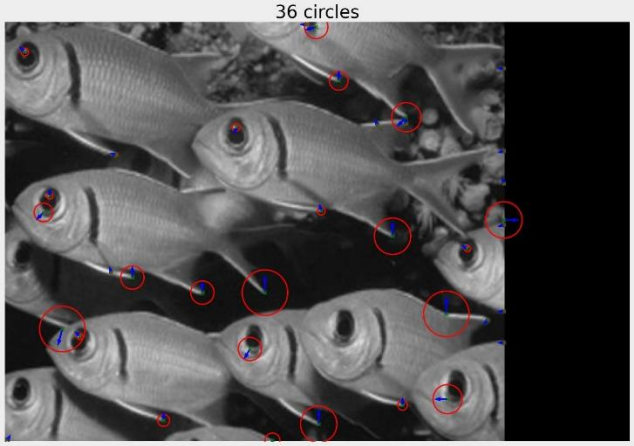

453 circles



Example 3:

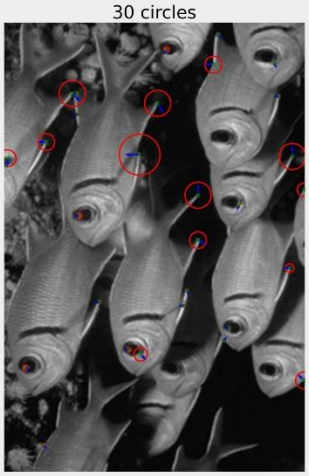
Raw images	Blob detection
	

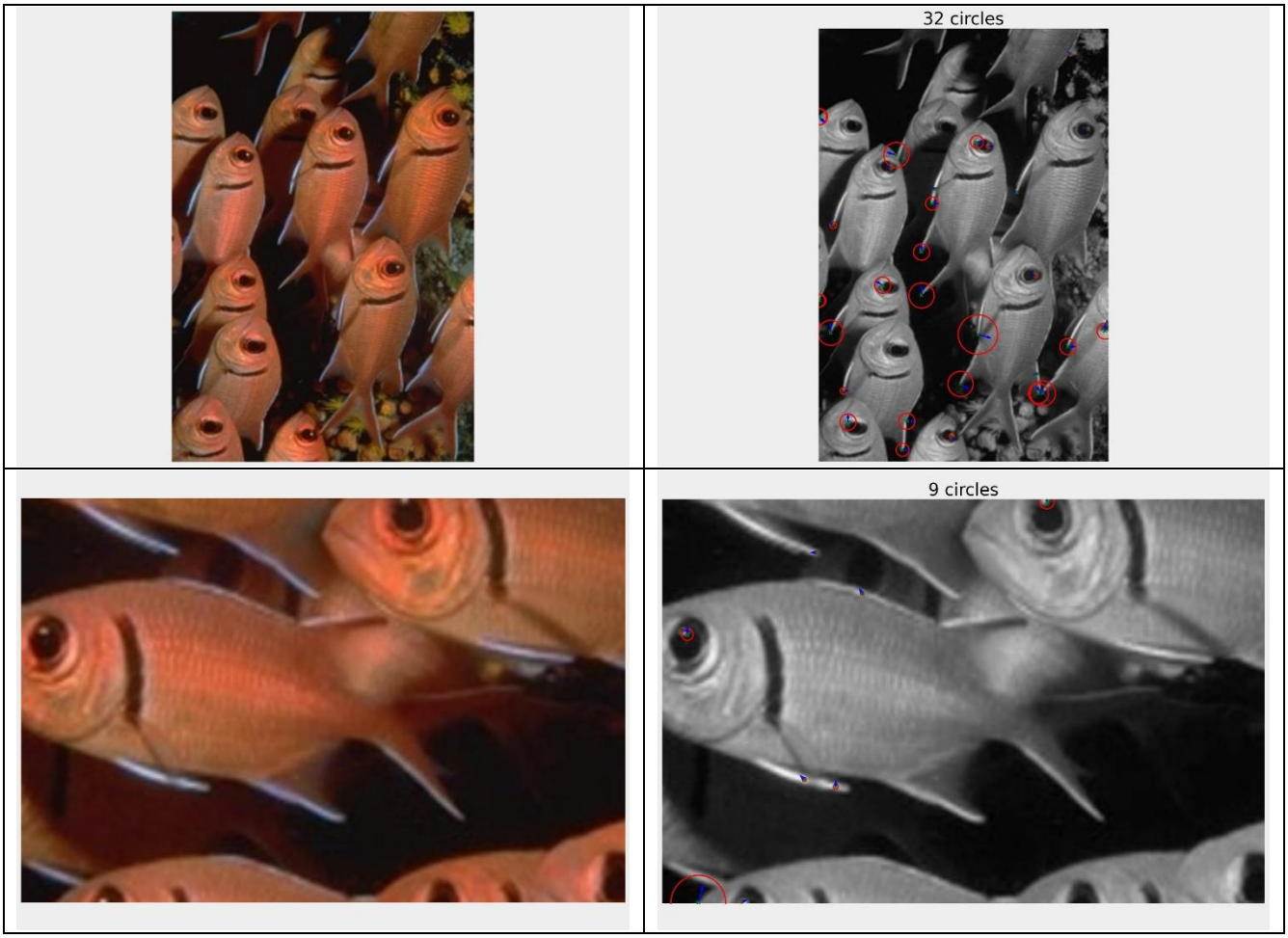
36 circles



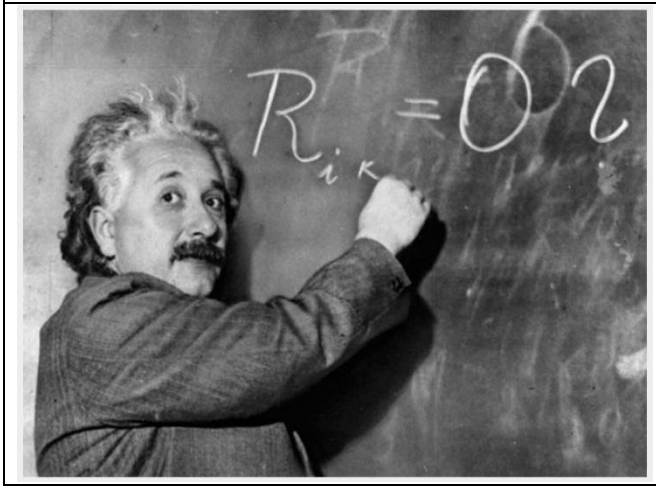
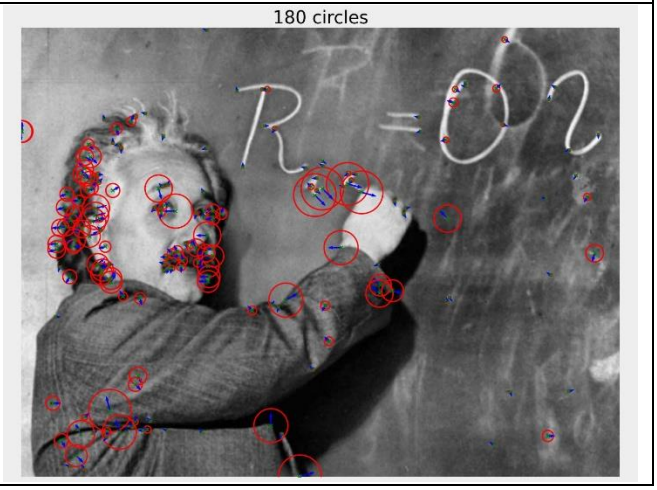
25 circles

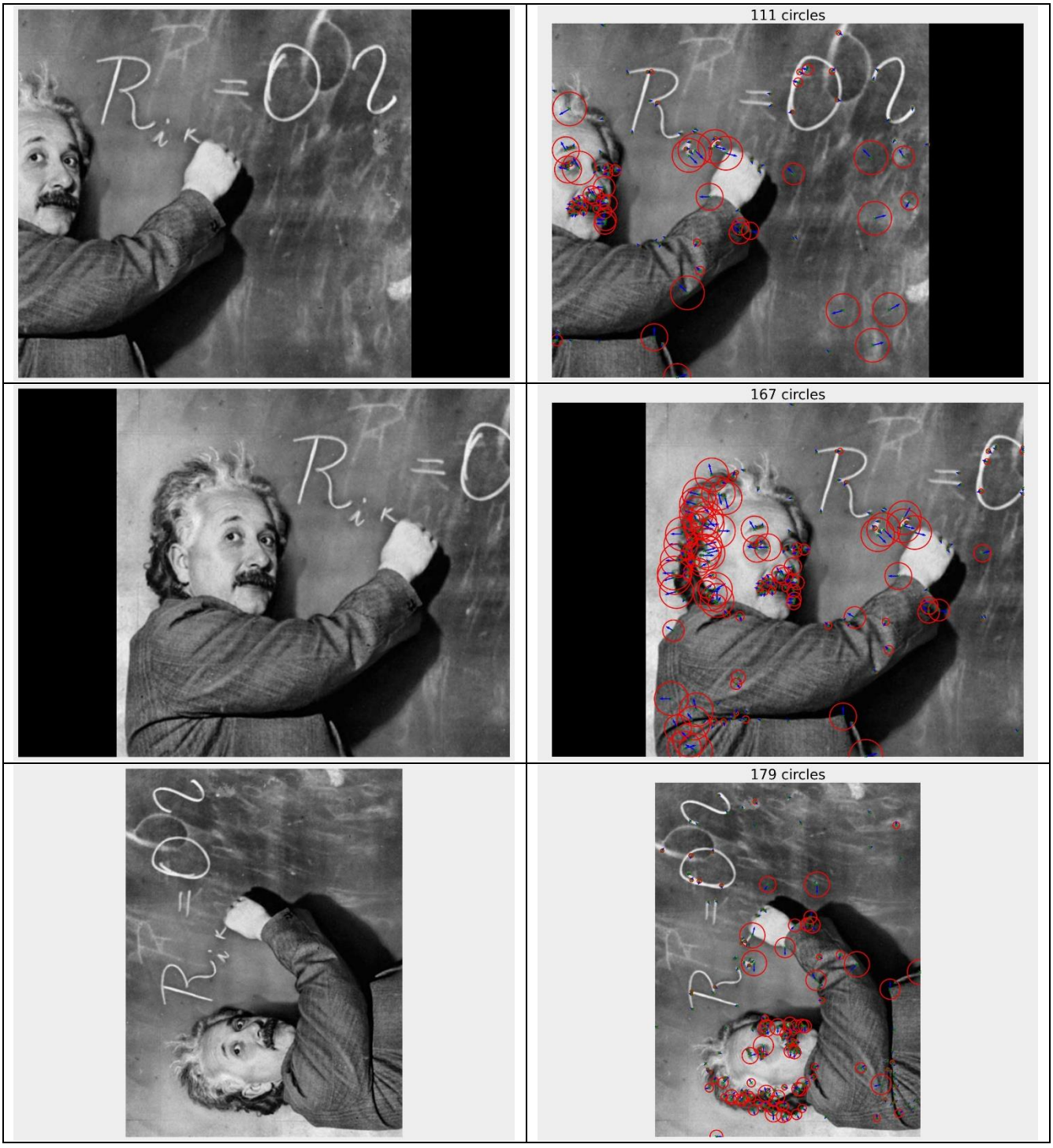
30 circles

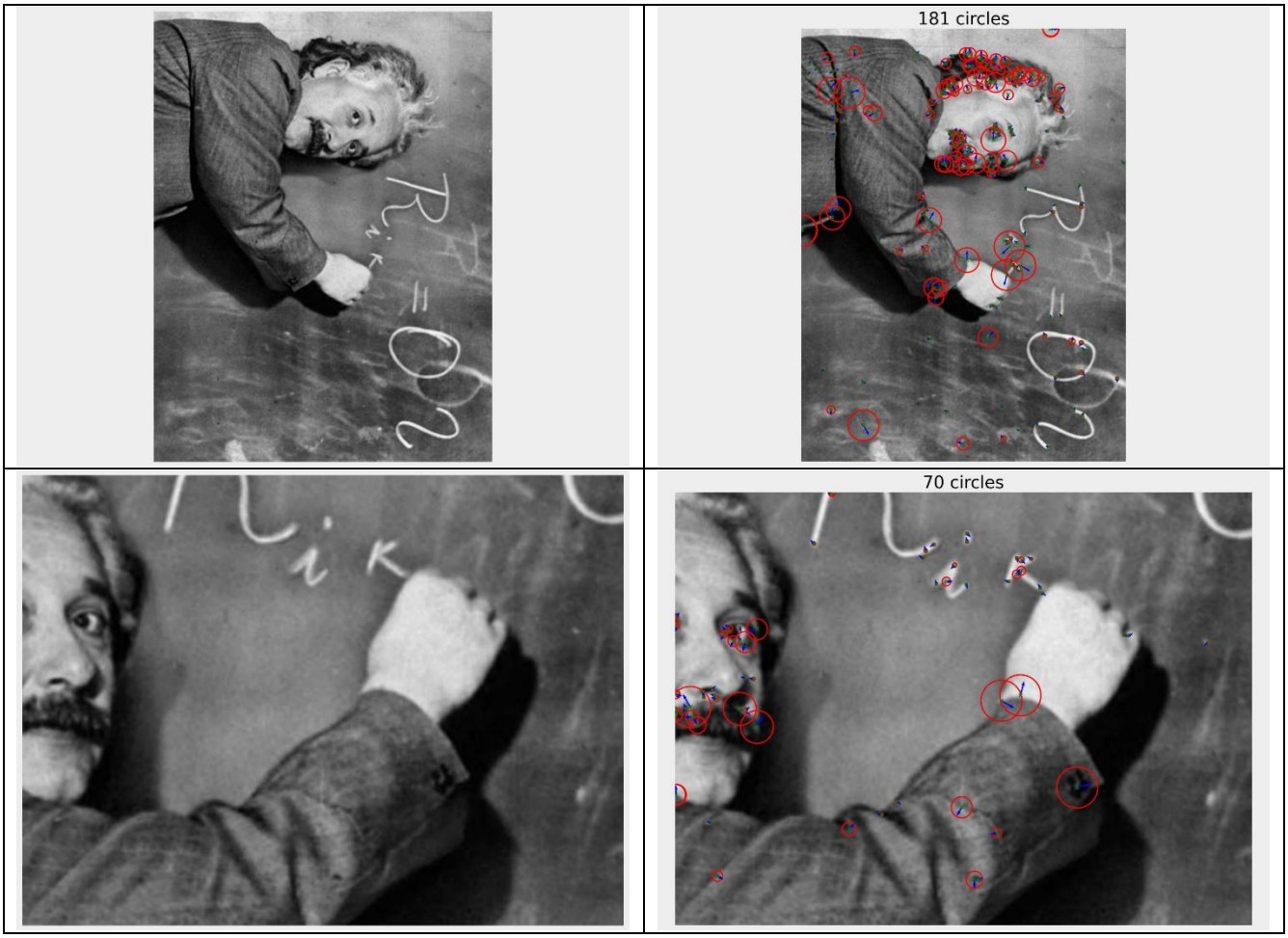




Example 4:

Raw images	Blob detection
	


111 circles
167 circles
179 circles



Discussion:

Explanation of any “interesting” implementation choices that you made.

Before finding scale space, I filtered the Harris corner detector output “dst” with maximum filter and then find pixels such that the output of the maximum filter “mx” and the “dst” is the same. Also, I set a threshold equals to 2% of the maximum pixel value of “dst” to get a better corner detection result.