Nathan Bemus

ECCS 4361

Homework 5

10/11/17

This is the image I manipulated throughout the assignment after changing it to greyscale.

The following Matlab code is used to initialize the image as a variable and displays it.

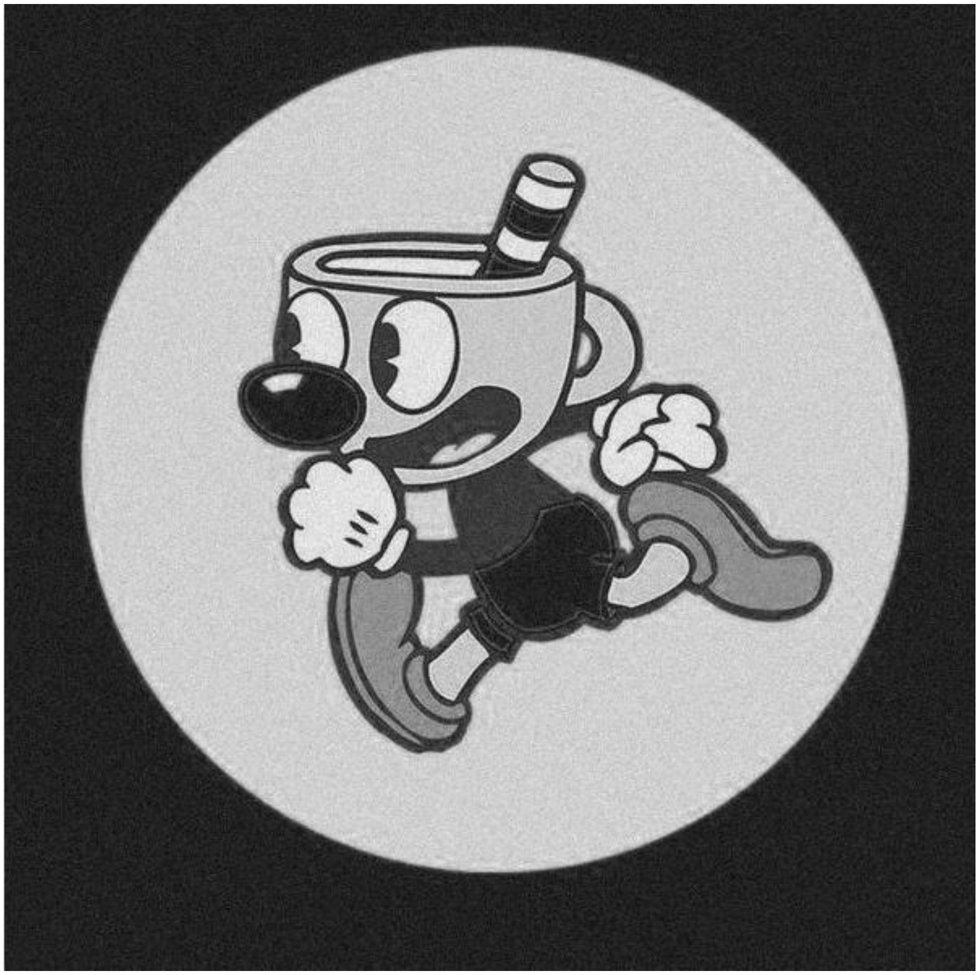
```
close all;
clear;
clc;

im =imread('Mugman.jpg');
im = im(:, :, 1);
figure(1);
imshow(im);
```

Next, I converted my image into a double and created the variables rows and columns. Their values are equivalent to the values of the pixels in the image. I then set the value of the standard deviation for the Gaussian Filter. I added noise to the image and then displayed the results. This is the Matlab code used to do what was described.

```
im = double(im);
rows = 600;
cols = 600;
```

```
gauss_std = 100;

imNoise = (gauss_std .* randn(rows, cols)) + im;
figure(2);
imshow(imNoise, [min(min(imNoise)), max(max(imNoise))]);
```

This is the resulting image the first time it was run through with a standard deviation of 10.



Next I extended the image and ran through it with a filtering window of different sizes. The following matlab code shows how I did this.

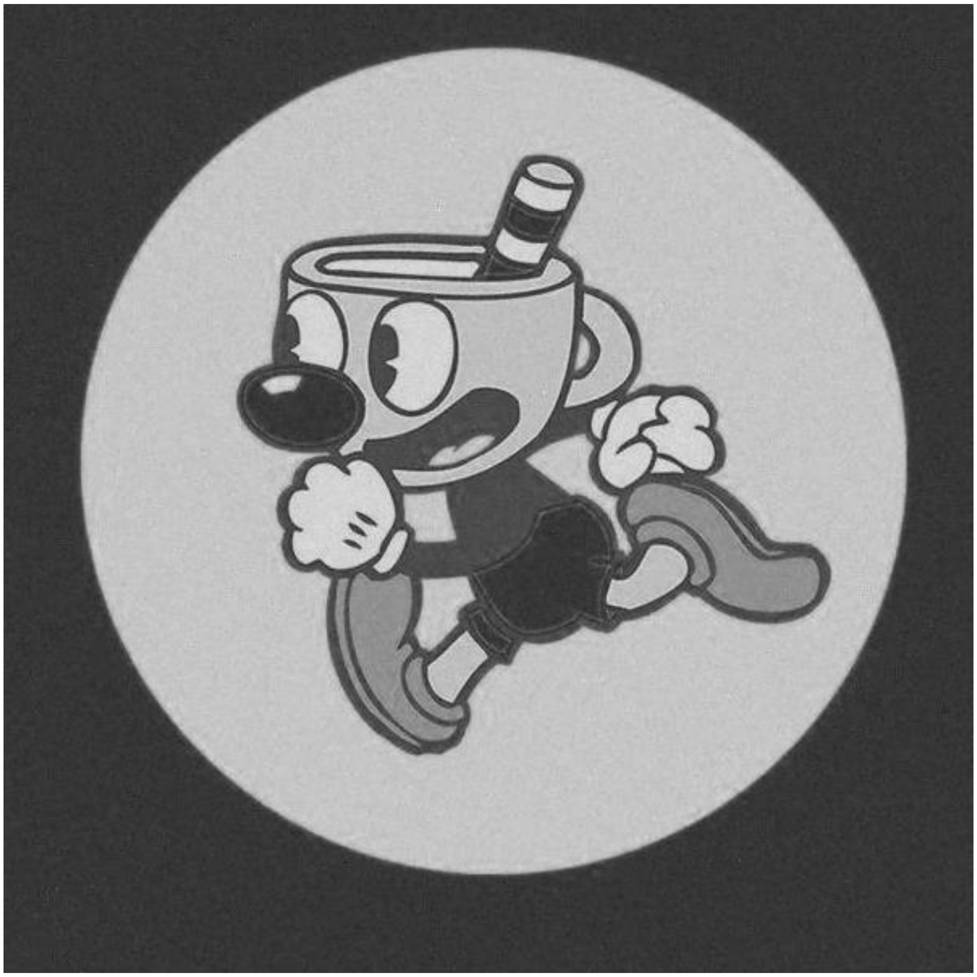
```
extend = zeros([rows*3, cols*3]);

for i = 1:rows
    for j = 1:cols
        extend(rows + i, cols + j) = imNoise(i,j);
        extend(rows + i, j) = imNoise(i, cols + 1 - j);
        extend(rows + i, 2*cols + j) = imNoise(i, cols + 1 -j);
    end
end

for i = 1:rows
```

```
    for j = 1:cols*3
        extend(i, j) = extend(rows + 1 + rows - i, j);
        extend(i + 2*rows, j) = extend(2*rows + 1 - i, j);
    end
end

windowSize = 11;

a = (windowSize - 1) /2;

means = zeros(rows, cols);
std = zeros(rows, cols);

for i = 1:rows
    for j = 1:cols
        window = extend(rows + i - a: rows + i + a, cols + j - a: cols + j +
a);
        means(i, j) = 1 / (windowSize*windowSize) * sum(sum(window));

        win1 = window - means(i,j);
        win2 = win1 .* win1;

        std(i, j) = 1/ (windowSize * windowSize) * sum(sum(win2));
    end
end

enhance = zeros(rows, cols);

for i = 1:rows
    for j = 1:cols
        enhance(i, j) = imNoise(i, j) - (gauss_std * gauss_std / std(i, j) *
(imNoise(i, j) - means(i, j)));
    end
end

figure(3);
imshow(enhance, [min(min(enhance)) , max(max(enhance))]);
```

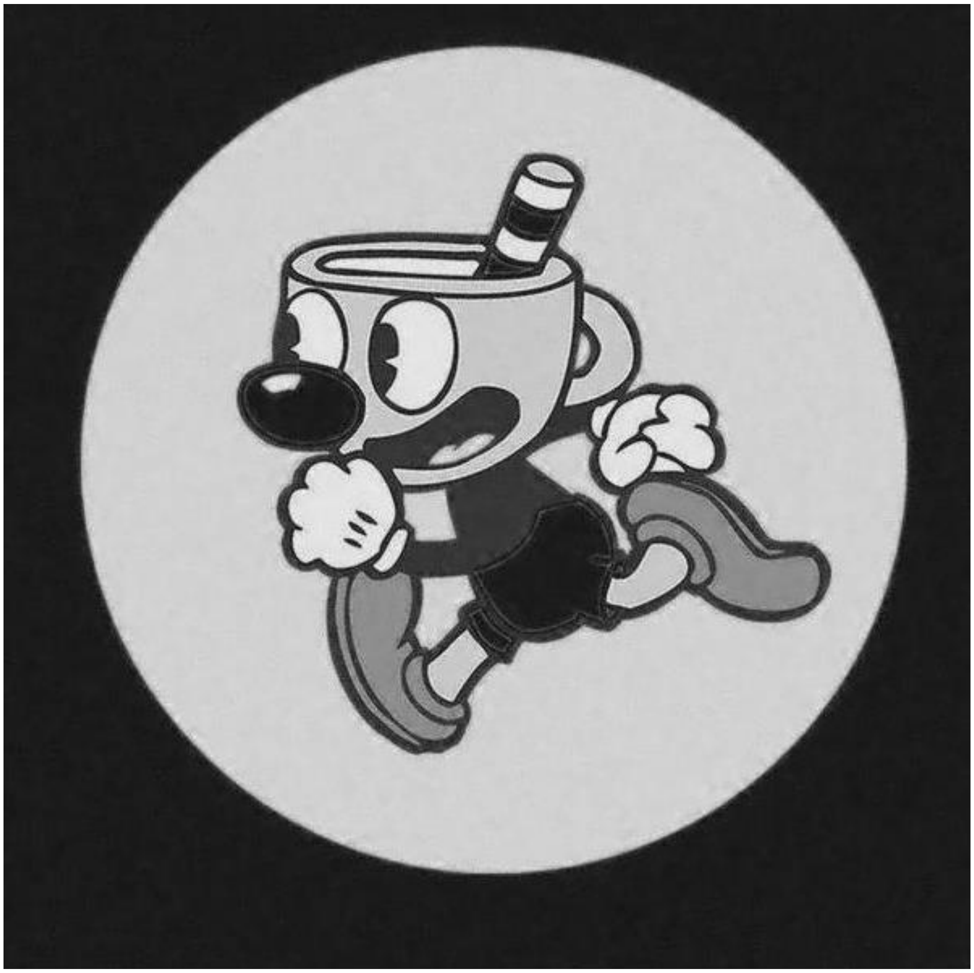
These are the results based off of the different window sizes used to filter through the image.

Standard Deviation of 10 (with filtering windows):

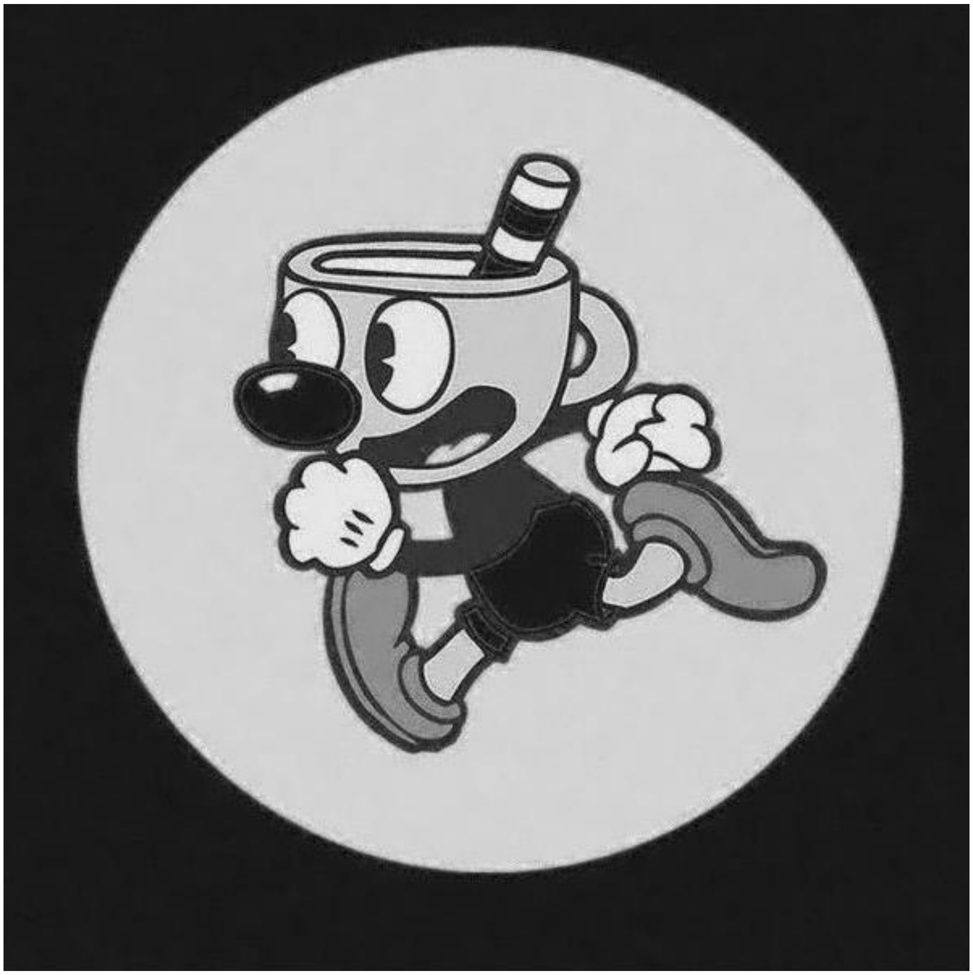
Window Size of 3:



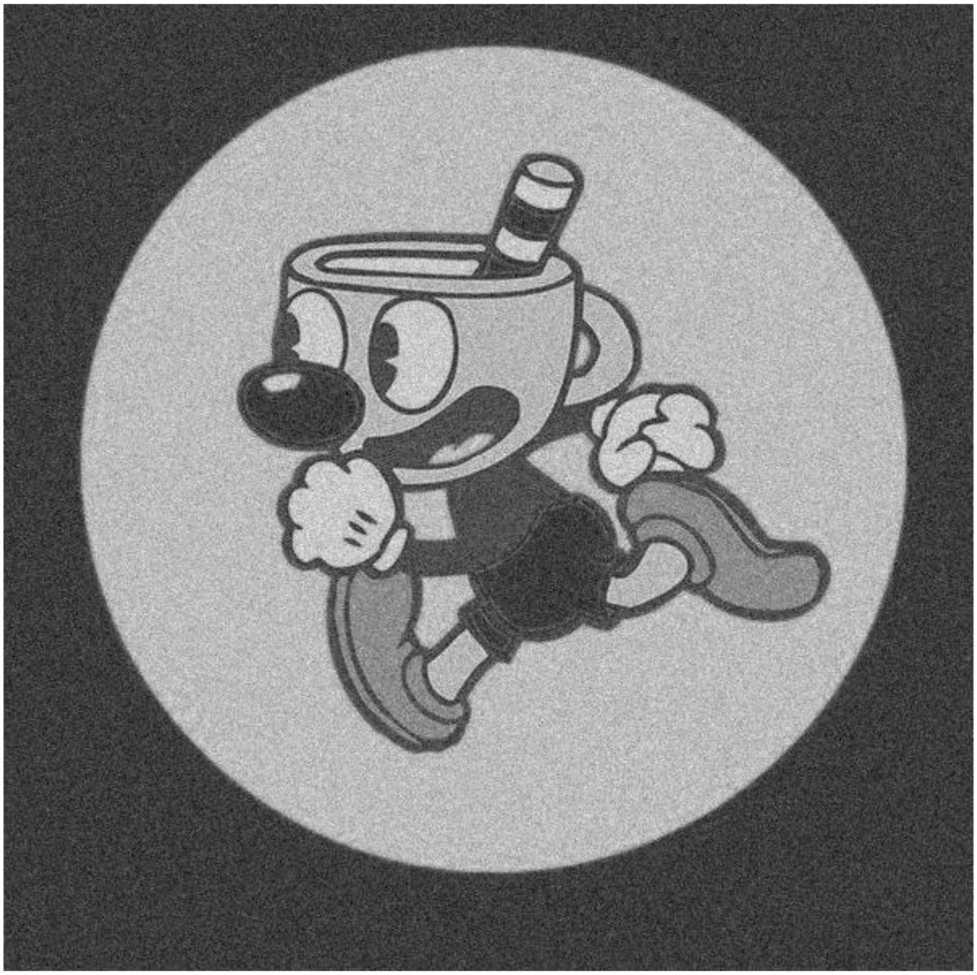
Window Size 7:



Window Size 11:

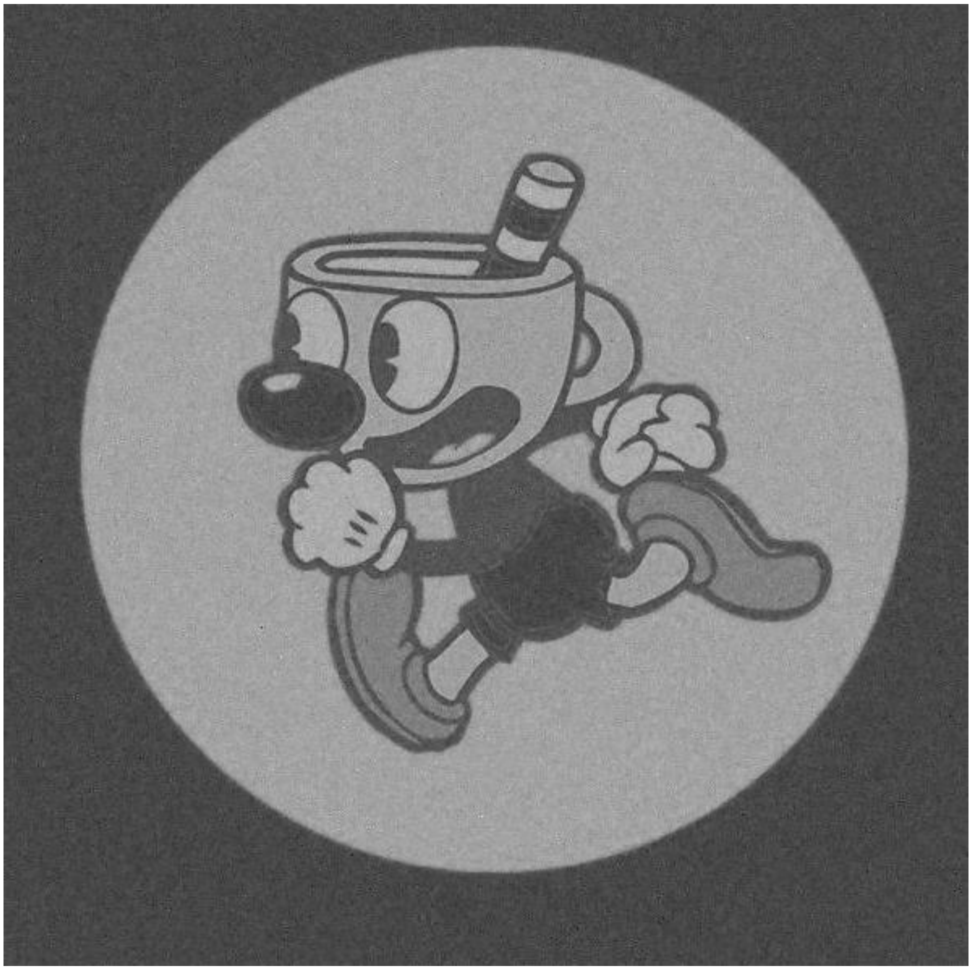


Standard Deviation of 25 (without a filtering window):

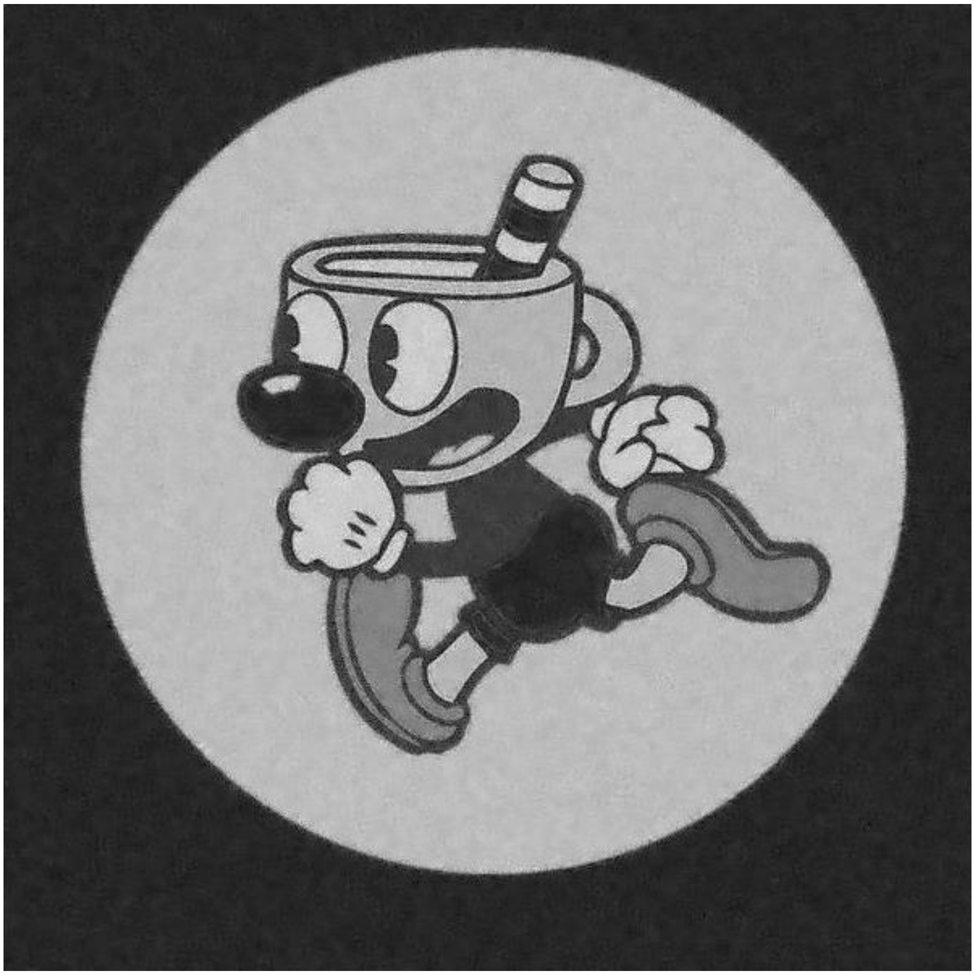


Standard Deviation of 25(with filtering windows):

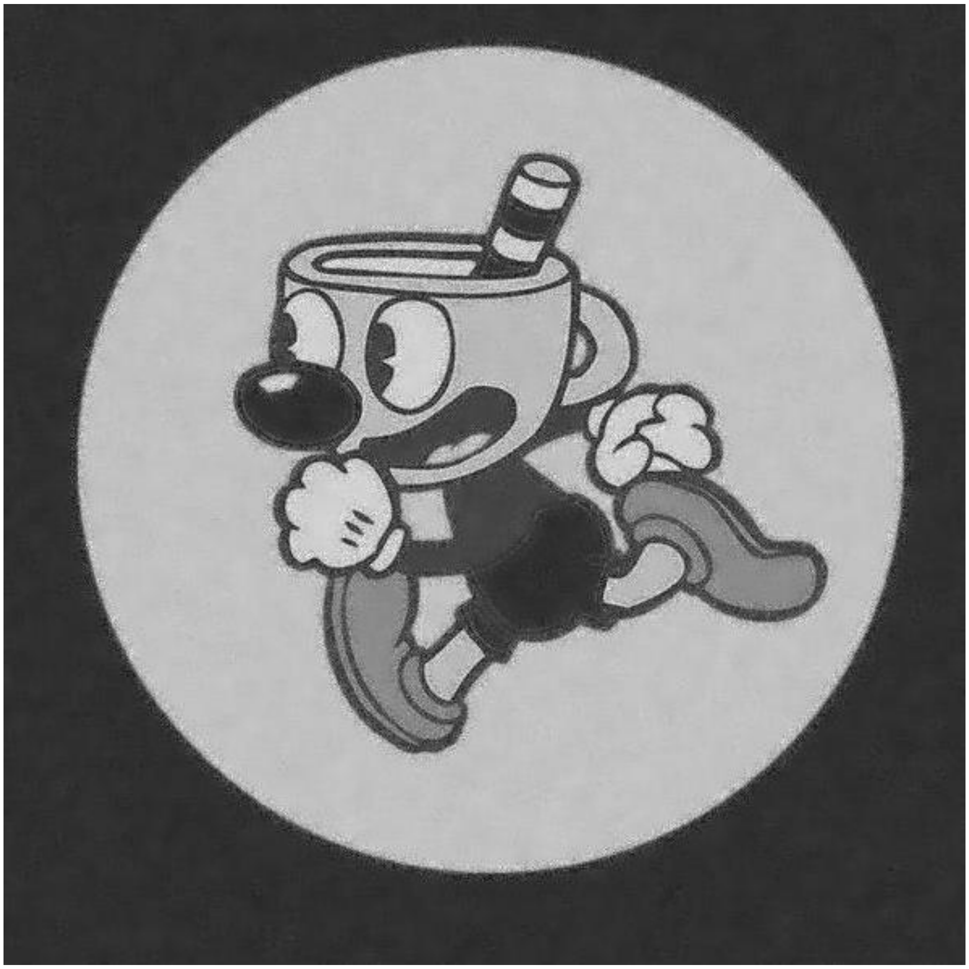
Window Size of 3:



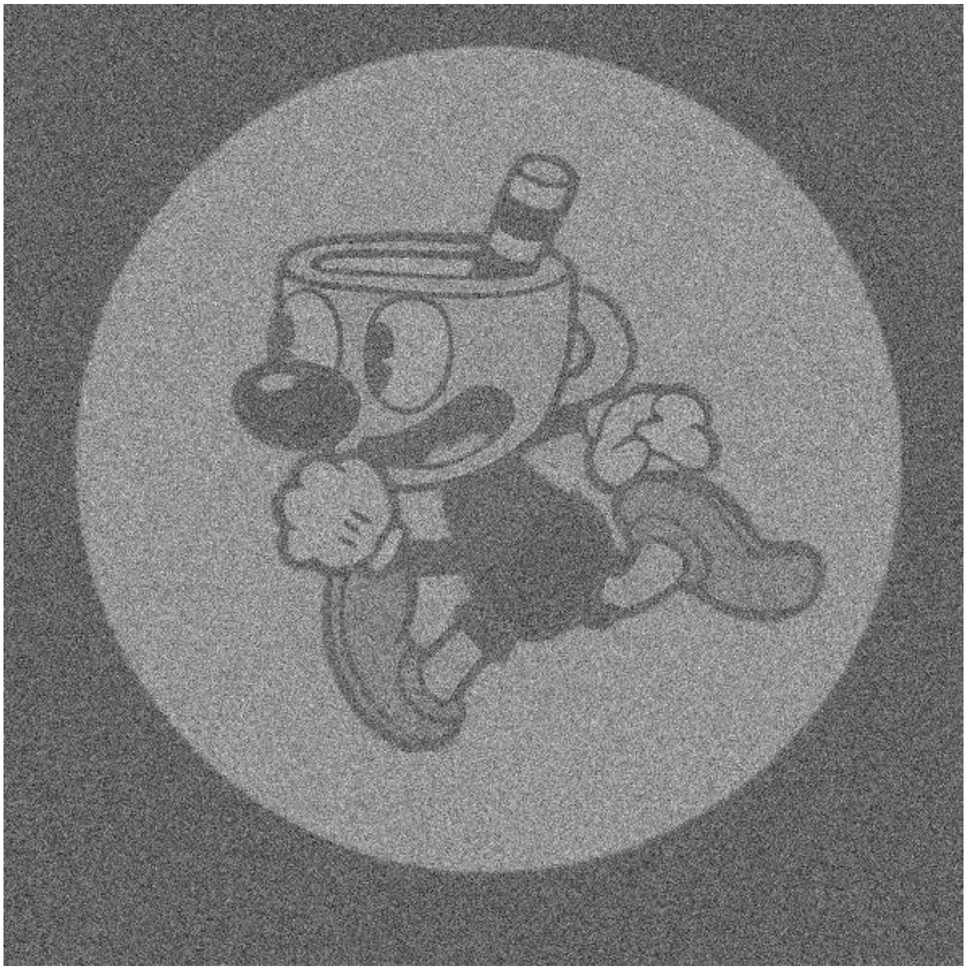
Window Size of 7:



Window Size of 11:

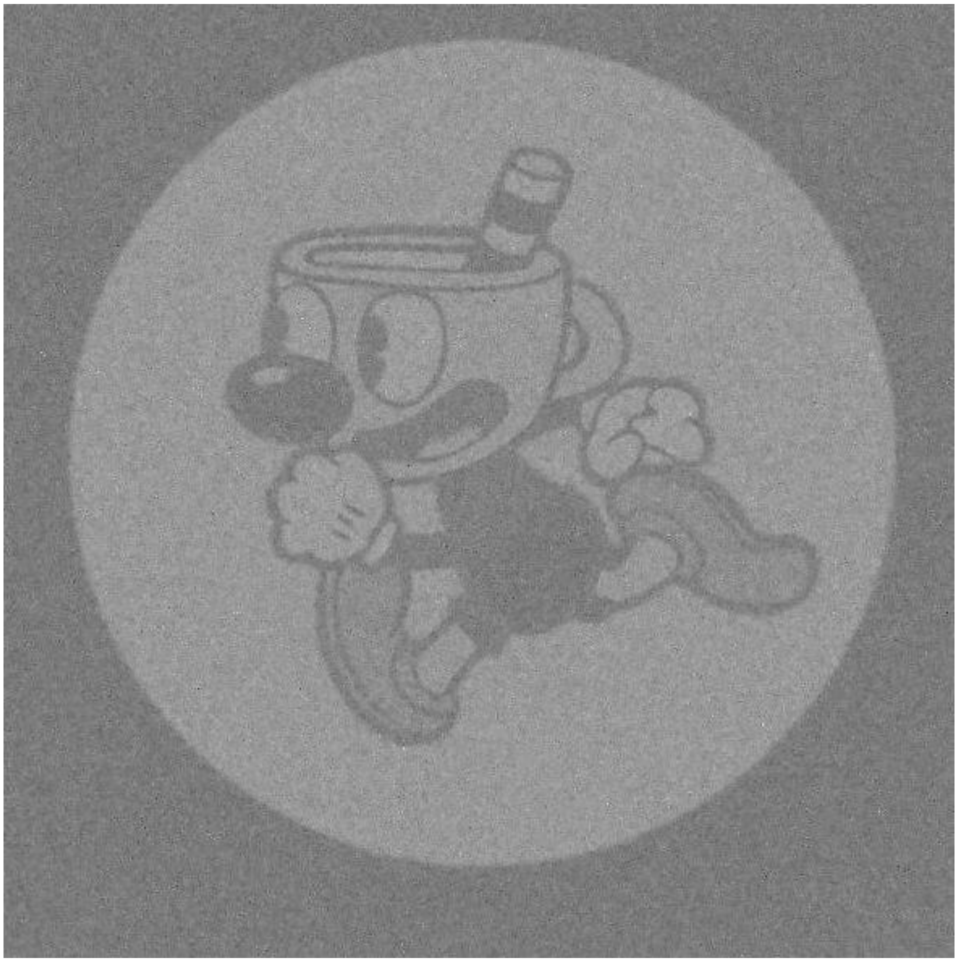


Standard Deviation of 100 (Without Filtering Window):

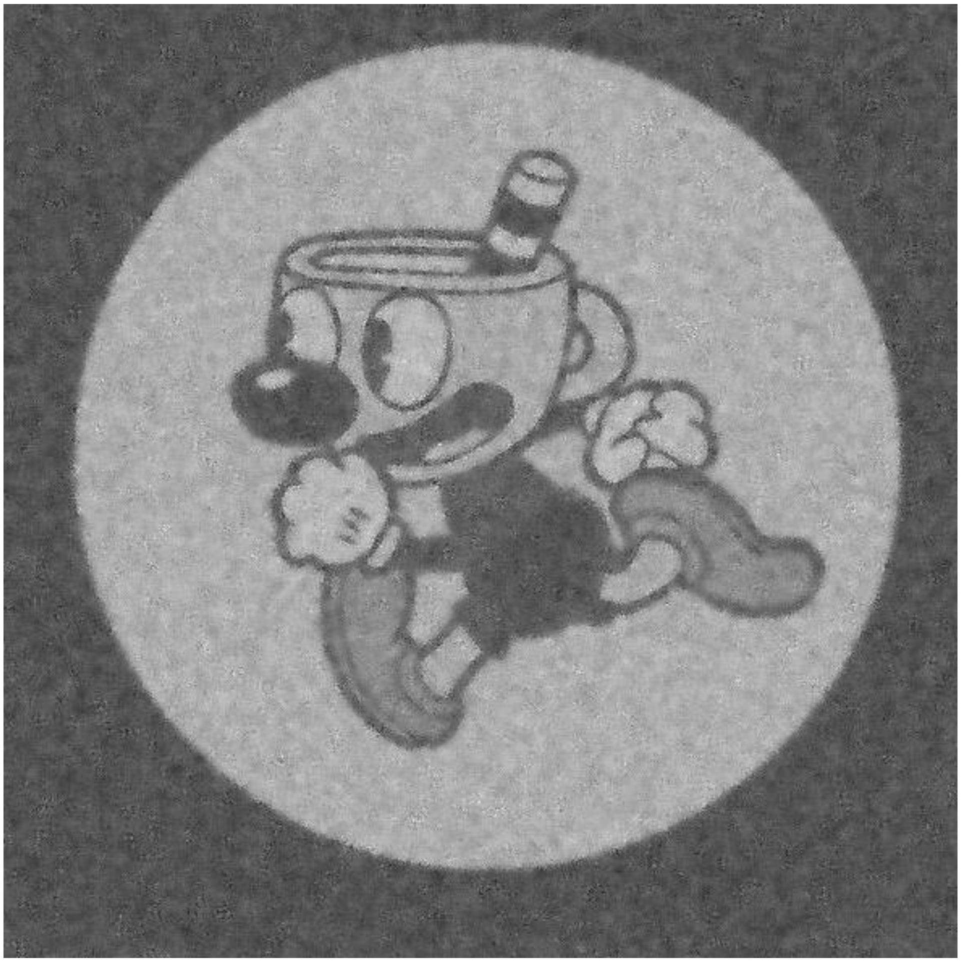


Standard Deviation of 100 (With Filtering Window):

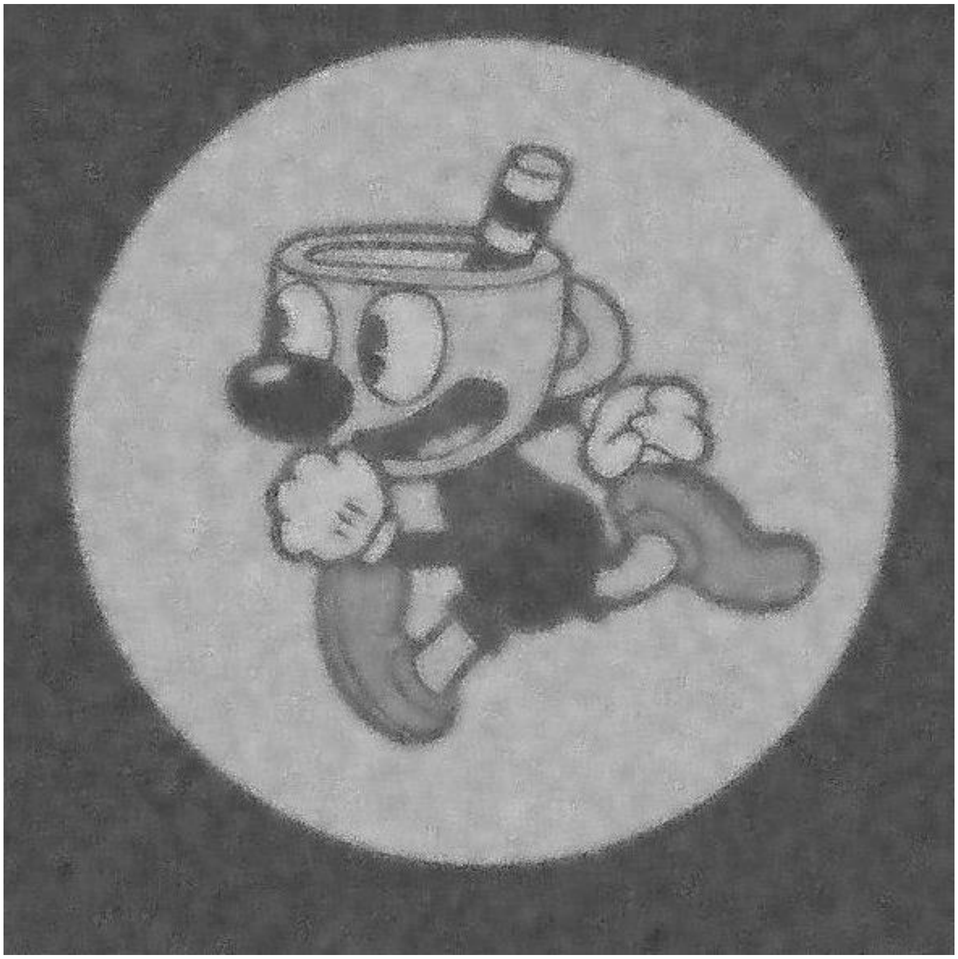
Window Size of 3:



Window Size of 7:



Window Size of 11:



The images look better when the standard deviation of the noise is lower. As the standard deviation gets higher the image got darker and darker without the filtering window. As the filtering window got bigger, the image got rid of the noise but would get more and more blurry. When there was more noise to get rid of in the image there was a loss of sharpness around the edges.