RYERSON UNIVERSITY

Faculty of Engineering, Architecture and Science

Department of Electrical and Computer Engineering

Course Number	CPS 843
Course Title	Introduction to Computer Vision
Semester/Year	F2023

Instructor	Dr. Guanghui Richard Wang

ASSIGNMENT No.	1

Assignment Title	Homework 1

Submission Date	October 9th, 2023
Due Date	October 9th, 2023

Student Name	Abdulrehman Khan
Student ID	500968727
Signature*	A.K.

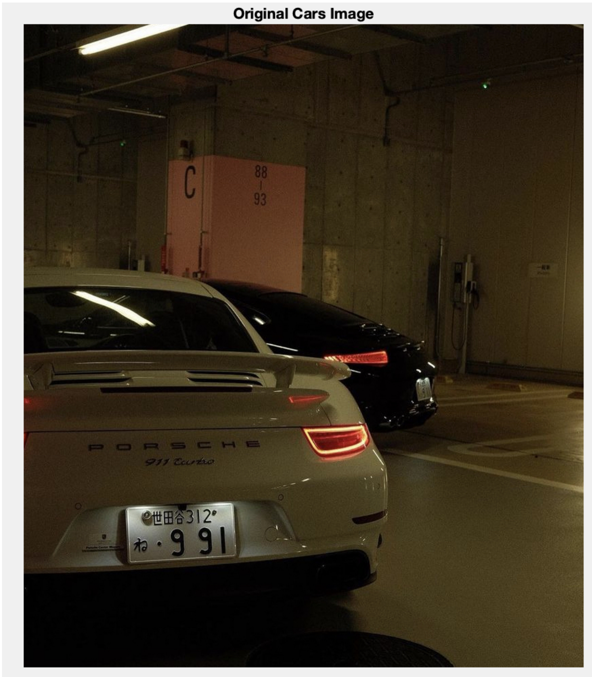
**By signing above you attest that you have contributed to this written lab report and confirm that all work you have swung the lab contributed to this lab report is your own work.*

Part 1:

Problem 1:

Function	***Equation***	***Effects***
Log	$s = c\, log\,(1 + r)\; where\; r \geq 0$	- Stretches low intensity levels - Compresses high intensity levels
Inverse Log	$s = c\, log^{-1}(r)$	- Stretches high intensity levels - Compresses low intensity levels
Power-law	$s = cr^{\gamma}$	- More versatile than log transformation - Performed by a lookup table

(A) Original cars:



(B) Grey-scaled cars:

(C) Power law $\gamma = 0.3$

(D) Power law $\gamma = 3$

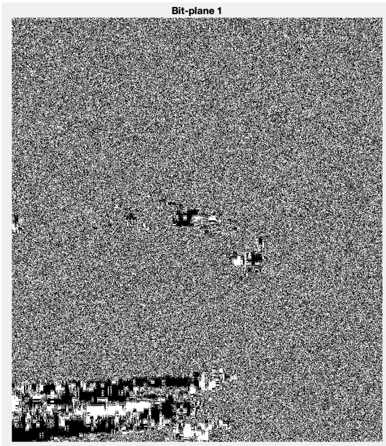
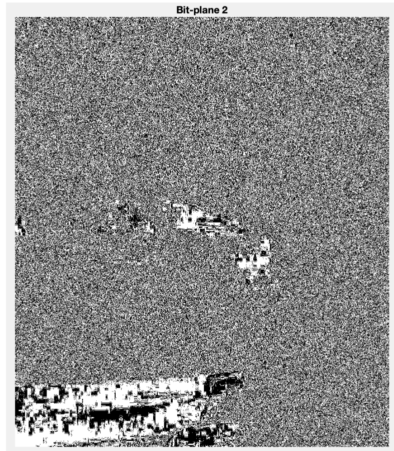
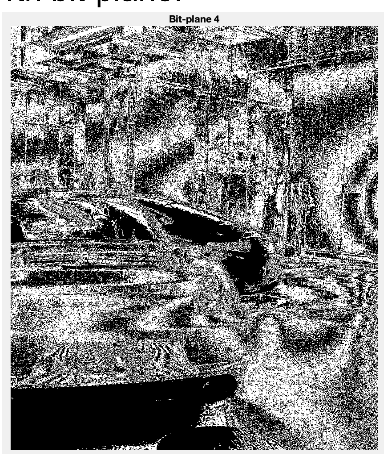
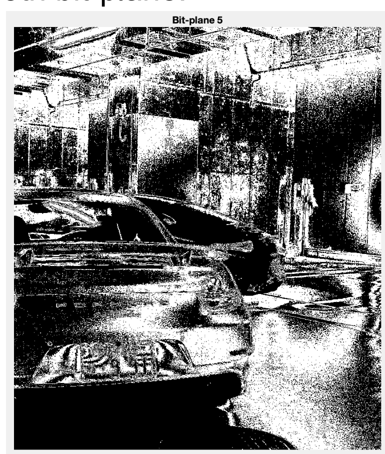
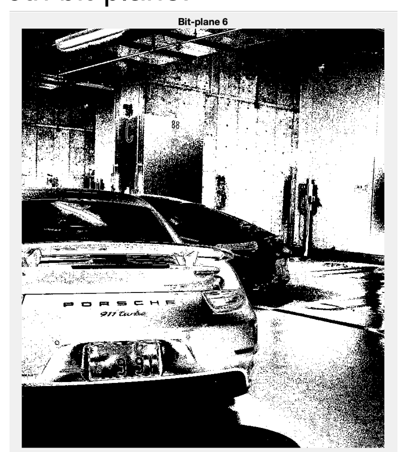
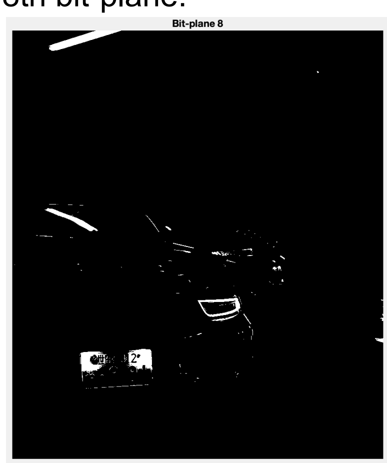
Brief Analysis:

Original cars (A) is a colorful image, and its grayscale counterpart is shown as grey-scaled cars (B). We apply Power-Law transformations with two different gamma values: gamma = 0.3 (C) and gamma = 3 (D). In the grayscale version, the green channel of the GB (Green Blue) image stands out as the most significant feature since all channels merge into one in grayscale, causing a loss of color information. When gamma = 0.3, low-intensity areas become significantly brighter than in the grayscale image, while gamma = 3 darkens high-intensity areas and enhances contrast compared to the original grayscale image (B).

Matlab code:

```
% Problem 1 Part 1

% Original image
image = imread("cars.jpeg");
imshow(image);
title ('Original Cars Image');

% Converting into greyscale image
imageGrey = rgb2gray(image);
imshow(imageGrey)
title ('Grey-scaled Cars Image');

% Problem 1 Part 2

imageLow = imadjust(imageGrey,[],[],0.3);
title ('Power Law gamma = 0.3');
imshow(imageLow);


imageHigh = imadjust(imageGrey,[],[],3);
title ('Power Law gamma = 3');
imshow(imageHigh);
```

Problem 2

1st bit-plane: 	2nd bit-plane: 	3rd bit-plane: 
4th bit-plane: 	5th bit-plane: 	6th bit-plane: 
7th bit-plane: 	8th bit-plane: 	Reconstructed image from highest 2 and 4-bit plane: 

Matlab Code for Problem 2

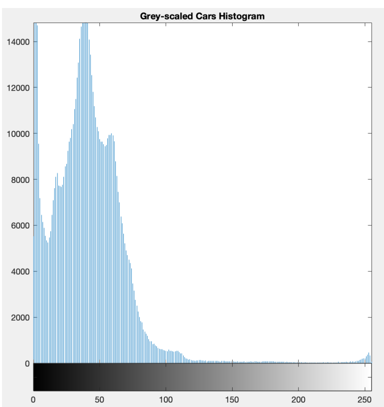
```
% Problem 2

b1 = double(bitget(imageGrey,1));
imshow(b1)
title ('Bit-plane 1');
b2 = double(bitget(imageGrey,2));
imshow(b2)
title ('Bit-plane 2');
b3 = double(bitget(imageGrey,3));
imshow(b3)
title ('Bit-plane 3');
b4 = double(bitget(imageGrey,4));
imshow(b4)
title ('Bit-plane 4');
b5 = double(bitget(imageGrey,5));
imshow(b5)
title ('Bit-plane 5');
b6 = double(bitget(imageGrey,6));
imshow(b6)
title ('Bit-plane 6');
b7 = double(bitget(imageGrey,7));
imshow(b7)
title ('Bit-plane 7');
b8 = double(bitget(imageGrey,8));
imshow(b8)
title ('Bit-plane 8');

c5 = b5 * 2^4;
c6 = b6 * 2^5;
c7 = b7 * 2^6;
c8 = b8 * 2^7;

high4 = uint8(c5 + c6 + c7 + c8);
imshow(high4)
title ('Reconstructed image from highest 2 and 4 bit-planes');
```

Brief Analysis:

The images above display eight dissected bit-planes, each corresponding to the grayscale image depicted in (B) Grey-scaled cars. In these bit-planes, the lower bits exhibit increased noise compared to the higher bits. Consequently, the 1st bit-plane through to the 4th bit-plane shows slices filled with noise, lacking distinct textures. However, the slices generated from the highest 4 bits contain the primary structural information of the original image, as exemplified in the reconstructed image from highest 2 and 4 bit-planes.

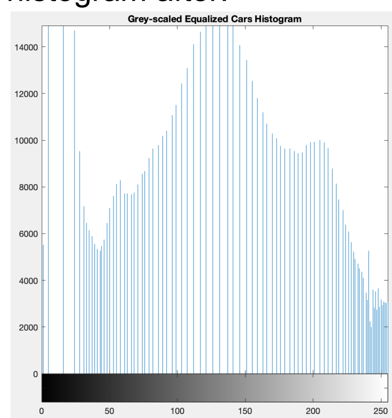
Problem 3

Grey-scaled cars histogram before:

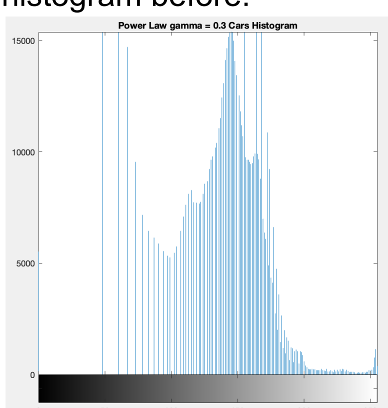


Equalized grey-scaled cars:

Grey-scaled cars histogram after:

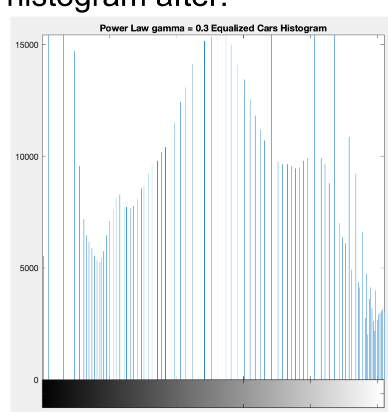


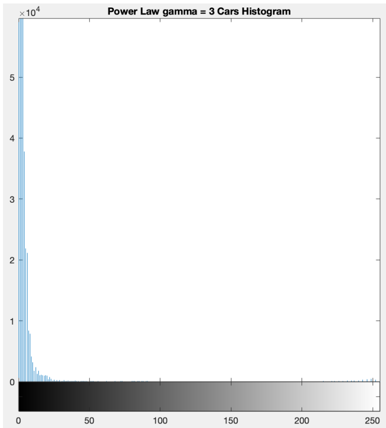
Power law $\gamma = 0.3$ cars histogram before:

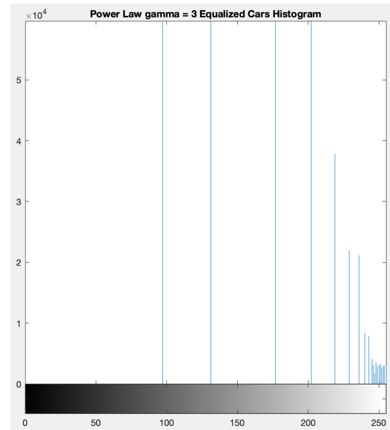


Equalized power law $\gamma = 0.3$ cars:

Power law $\gamma = 0.3$ cars histogram after:



<table>
<tr>
<td>Power law $\gamma = 3$ cars histogram before:

</td>
<td>Equalized power law $\gamma = 3$ cars:

</td>
<td>Power law $\gamma = 3$ cars histogram after:

</td>
</tr>
</table>

Brief Analysis:

As compared to the before histograms, the application of histogram equalization results in a uniform distribution of histograms across the entire intensity range. Consequently, all the images exhibit a similarity to one another.

Matlab Code for Problem 3

```
% Problem 3

imhist(imageGrey, 256);
title('Grey-scaled Cars Histogram');

imhist(imageLow, 256);
title('Power Law gamma = 0.3 Cars Histogram');

imhist(imageHigh, 256);
title('Power Law gamma = 3 Cars Histogram');

histeq(imageGrey, 256);
title('Grey-scaled Cars Equalization');

histeq(imageLow, 256);
title('Power Law gamma = 0.3 Cars Equalization');

histeq(imageHigh, 256);
title('Power Law gamma = 3 Cars Equalization');

imhist(histeq(imageGrey, 256), 256);
title('Grey-scaled Equalized Cars Histogram');

imhist(histeq(imageLow, 256), 256);
title('Power Law gamma = 0.3 Equalized Cars Histogram');

imhist(histeq(imageHigh, 256), 256);
title('Power Law gamma = 3 Equalized Cars Histogram');
```

Problem 4

Abdulrehman Khan
500968727

$$S_k = T(r_k) = (L-1)\sum P_r(r_j)$$

$$S_0 = 7\left(\sum_{k=0}^{0} P_r(r_j)\right) = 7\,p_r(r_0) = 1.33 \rightarrow 1.00$$

$$S_1 = 7\left(\sum_{k=0}^{1} P_r(k)\right) = 7\left(P_r(r_0) + P_r(r_1)\right) = 3.08 \rightarrow 3.00$$

$$S_2 = 7\left(\sum_{k=0}^{2} P_r(k)\right) = 7\left(P_r(r_0) + P_r(r_1) + P_r(r_2)\right) = 4.55 \rightarrow 5.00$$

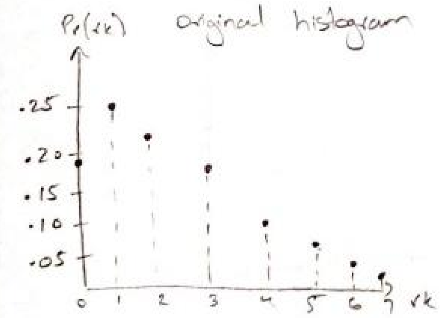
$$S_3 = 7\left(\sum_{k=0}^{3} P_r(k)\right) = 7\left(P_r(r_0) + P_r(r_1) + P_r(r_2) + P_r(r_3)\right) = 5.67 \rightarrow 6.00$$

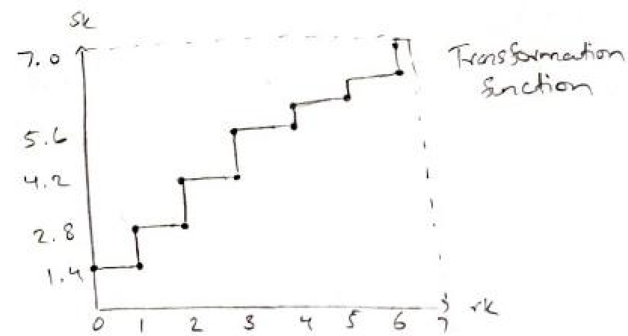
$$S_4 = 7\left(\sum_{k=0}^{4} P_r(k)\right) = 7\left(P_r(r_0) + P_r(r_1) + P_r(r_2) + P_r(r_3) + P_r(r_4)\right) = 6.23 \rightarrow 6.00$$

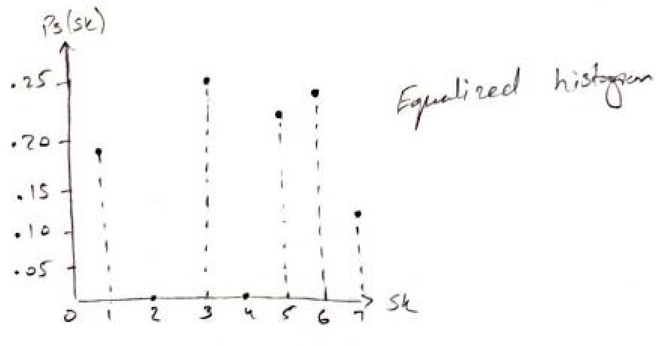
$$S_5 = 7\left(\sum_{k=0}^{5} P_r(k)\right) = 7\left(P_r(r_0) + P_r(r_1) + P_r(r_2) + P_r(r_3) + P_r(r_4) + P_r(r_5)\right) = 6.65 \rightarrow 7.00$$

$$S_6 = 7\left(\sum_{k=0}^{6} P_r(k)\right) = 7\left(P_r(r_0) + P_r(r_1) + P_r(r_2) + P_r(r_3) + P(r_4) + P(r_5) + P(r_6)\right) = 6.86 \rightarrow 7.00$$

$$S_7 = 7\left(\sum_{k=0}^{7} P_r(k)\right) = 7\left(P_r(r_0) + P_r(r_1) + P_r(r_2) + P(r_3) + P(r_4) + P(r_5) + P(r_6) + P(r_7)\right) = 7.00$$



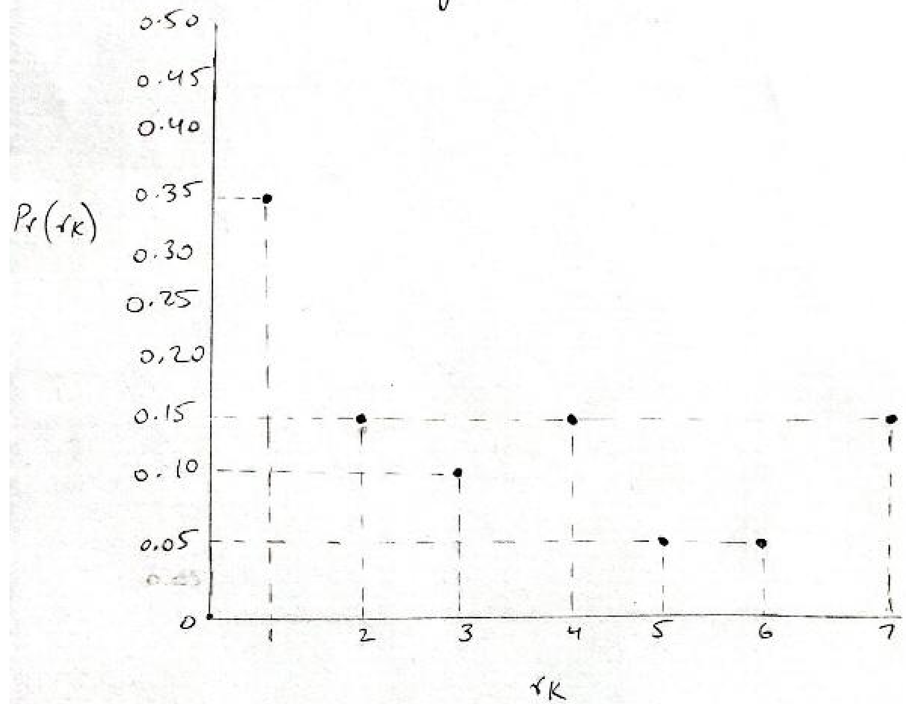




Problem 5

Abdulrehman Khan
500968727

① Histogram Before



② Equalization $\quad P_r(r_K) = \frac{n_K}{MN}$

$M = 4$
$N = 5$
$L = 8$
$L - 1 = 7$

r_K	n_K	$P_r(r_K)$	S_K	$P_s(S_K)$
$r = 0$	0	0	0	0
$r = 1$	7	0.35	1	0
$r = 2$	3	0.15	2	0.35
$r = 3$	2	0.10	3	0
$r = 4$	3	0.15	4	0.25
$r = 5$	1	0.05	5	0.15
$r = 6$	1	0.05	6	0.10
$r = 7$	3	0.15	7	0.15

frequency $\frac{nt}{(4)(5)}$ (probability)

3

Abdulrehman Khan
500968727

$S_k = T(r_k) = (L-1) \sum P_r(r_j)$

$S_0 = 7(0) = 0$

$S_1 = 7\left(\sum_{k=0}^{1} P_r(k)\right) = 7(0 + 0.35) = 2.45 \approx 2$

$S_2 = 7\left(\sum_{k=0}^{2} P_r(k)\right) = 7(0 + 0.35 + 0.15) = 3.5 \approx 4$

$S_3 = 7\left(\sum_{k=0}^{3} P_r(k)\right) = 7(0 + 0.35 + 0.15 + 0.1) = 4.2 \approx 4$

$S_4 = 7\left(\sum_{k=0}^{4} P_r(k)\right) = 7(0 + 0.35 + 0.15 + 0.1 + 0.15) = 5.25 \approx 5$

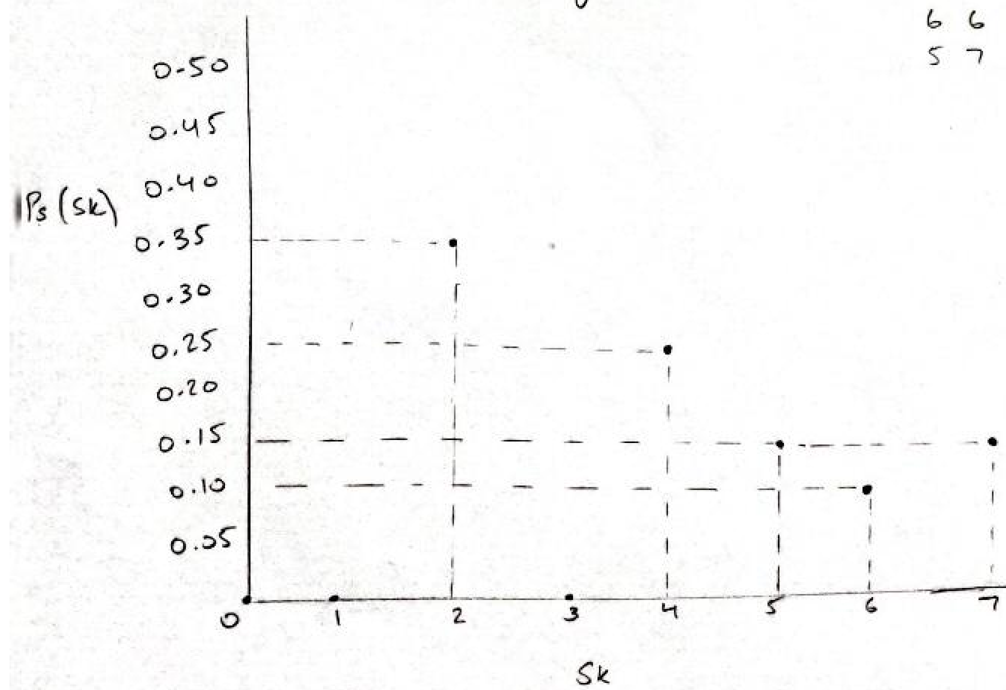
$S_5 = 7\left(\sum_{k=0}^{5} P_r(k)\right) = 7(0 + 0.35 + 0.15 + 0.1 + 0.15 + 0.05) = 5.6 \approx 6$

$S_6 = 7\left(\sum_{k=0}^{6} P_r(k)\right) = 7(0 + 0.35 + 0.15 + 0.1 + 0.15 + 0.05 + 0.05) = 5.95 \approx 6$

$S_7 = 7\left(\sum_{k=0}^{7} P_r(k)\right) = 7(0 + 0.35 + 0.15 + 0.1 + 0.15 + 0.05 + 0.05 + 0.15) = 7$

Histogram After

2	4	5	7	4
4	5	7	4	2
6	6	4	2	2
5	7	2	2	2



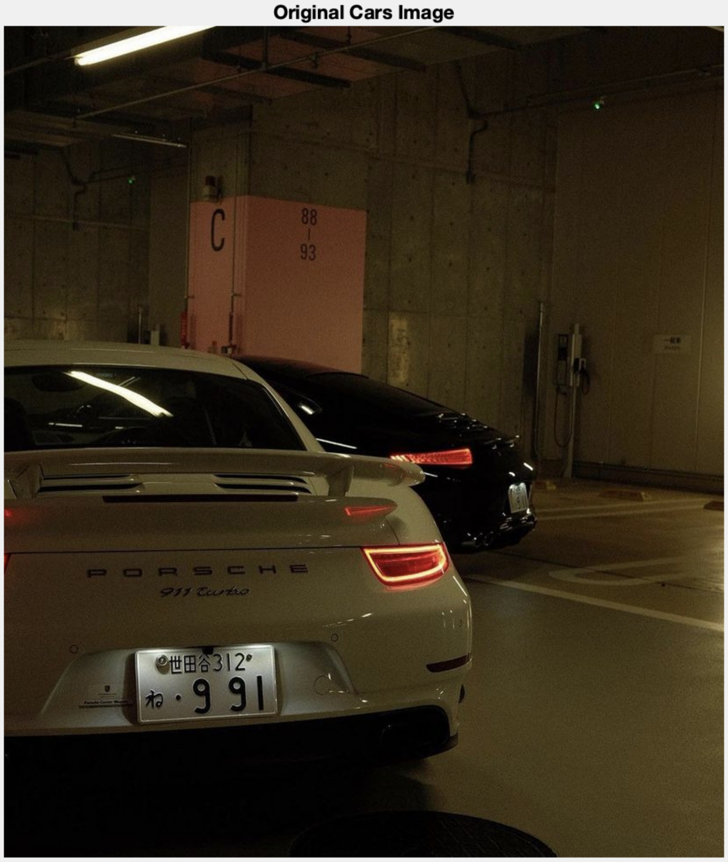
Part 2:

Following the instructions from the examples, referencing and, adjusting the code to match my images requirements the following was produced:

Original Image:

```
a = 0.45;
T = maketform('affine', [1 0 0; a 1 0; 0 0 1] );

image = imread('cars.jpeg');
h1 = figure;
imshow(image);
title('Original Cars Image');
```

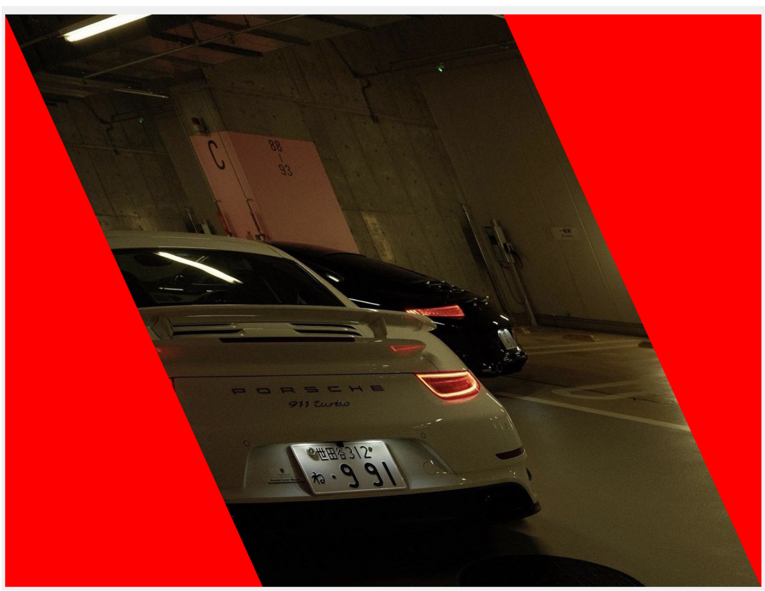
Original Cars Image



Sheared Image:

Chose red as the color

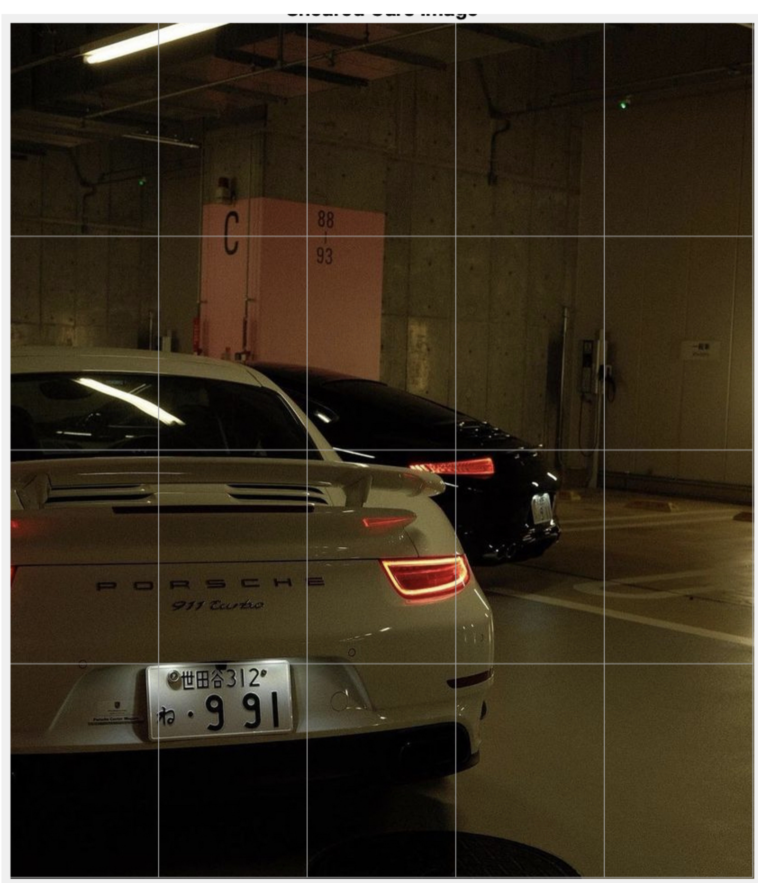
```
red = [255 0 0]';

R = makeresampler({'cubic','nearest'},'fill');
B = imtransform(image,T,R,'FillValues',red);
h2 = figure; imshow(B);
title('Sheared Cars Image');
```



Original Image:

Our cars image the height is 950px and the width is 826px so adjustments were made and will continue to be made from here on out based on this: [U, V] = meshgrid(0:(image_width-1)/5:(image_width-1), 0:(image_height-1)/4:(image_height-1));

```
[U,V] = meshgrid(0:165:825, 0:237.25:949);
[X,Y] = tformfwd(T,U,V);
gray = 0.65 * [1 1 1];

figure(h1);
hold on;
line(U, V, 'Color',gray);
line(U',V','Color',gray);
```



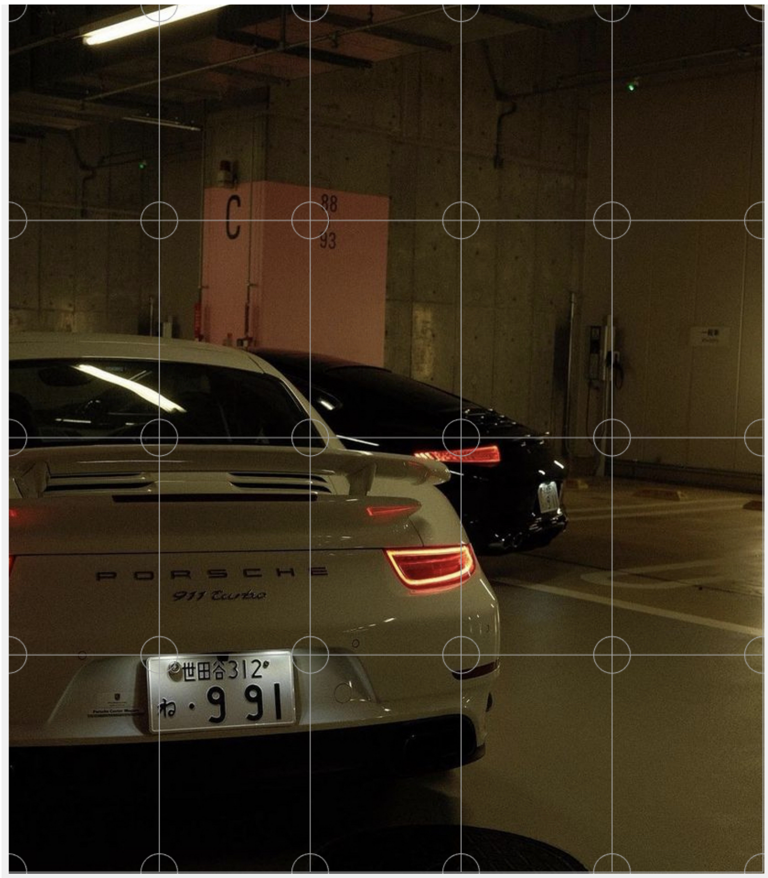
Sheared Image:

```
figure(h2);
hold on;
line(X, Y, 'Color',gray);
line(X',Y','Color',gray);
```

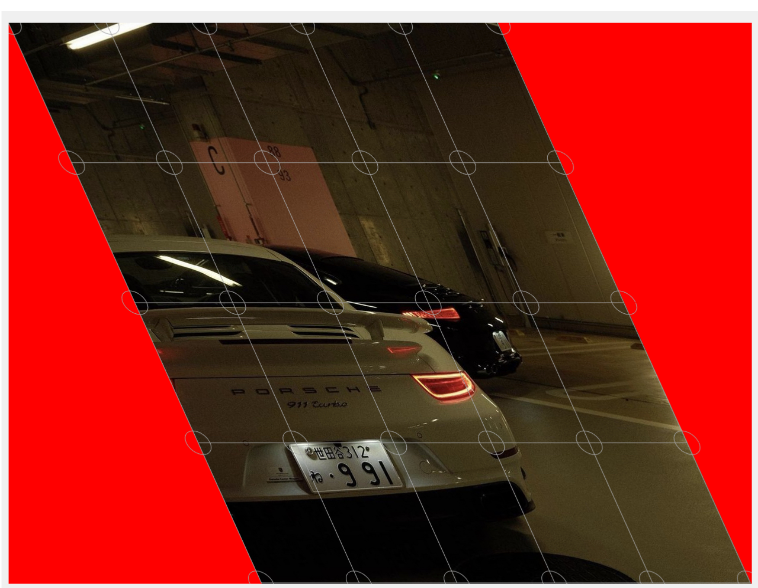
Original Image:

Adjustments made for my image

```
gray = 0.65 * [1 1 1];
for u = 0:165:825
    for v = 0:237.25:949
        theta = (0 : 82.5)' * (2 * pi / 82.5);
        uc = u + 20*cos(theta);
        vc = v + 20*sin(theta);
        [xc,yc] = tformfwd(T,uc,vc);
        figure(h1); line(uc,vc,'Color',gray);
        figure(h2); line(xc,yc,'Color',gray);
    end
end
```



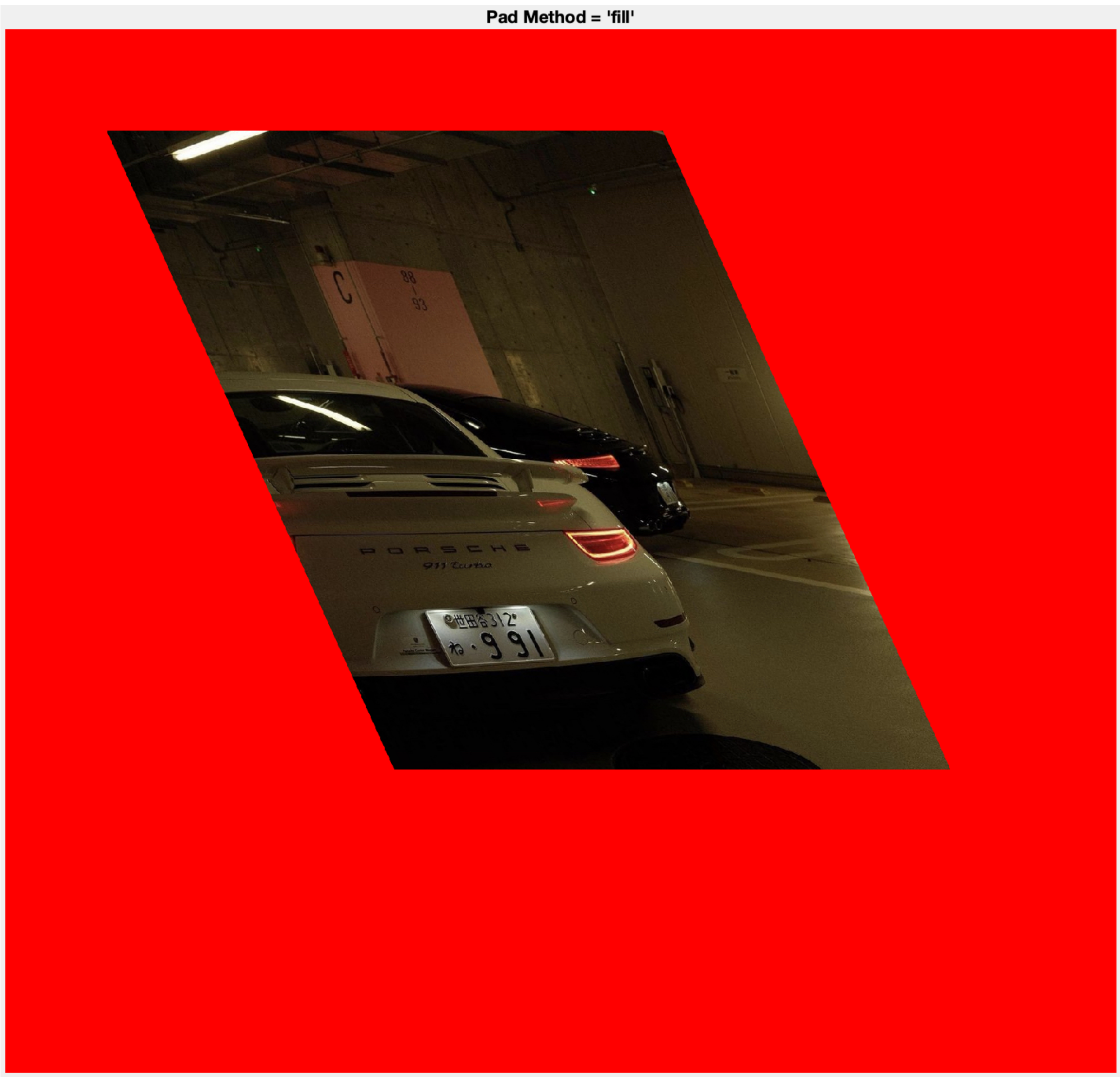
Sheared Image:



Padding Image (fill):

Adjustments made for my image

```
R = makeresampler({'cubic','nearest'},'fill');
Bf = imtransform(A,T,R,'XData',[-149 1500],'YData',[-149 1400],...
                     'FillValues',red);

figure, imshow(Bf);
title('Pad Method = ''fill''');
```



Padding Image (replicate):

Adjustments made for my image

```
R = makeresampler({'cubic','nearest'},'replicate');
Br = imtransform(image,T,R,'XData',[-149 1500],'YData', [-149 1400]);

figure, imshow(Br);
title('Pad Method = ''replicate''');
```

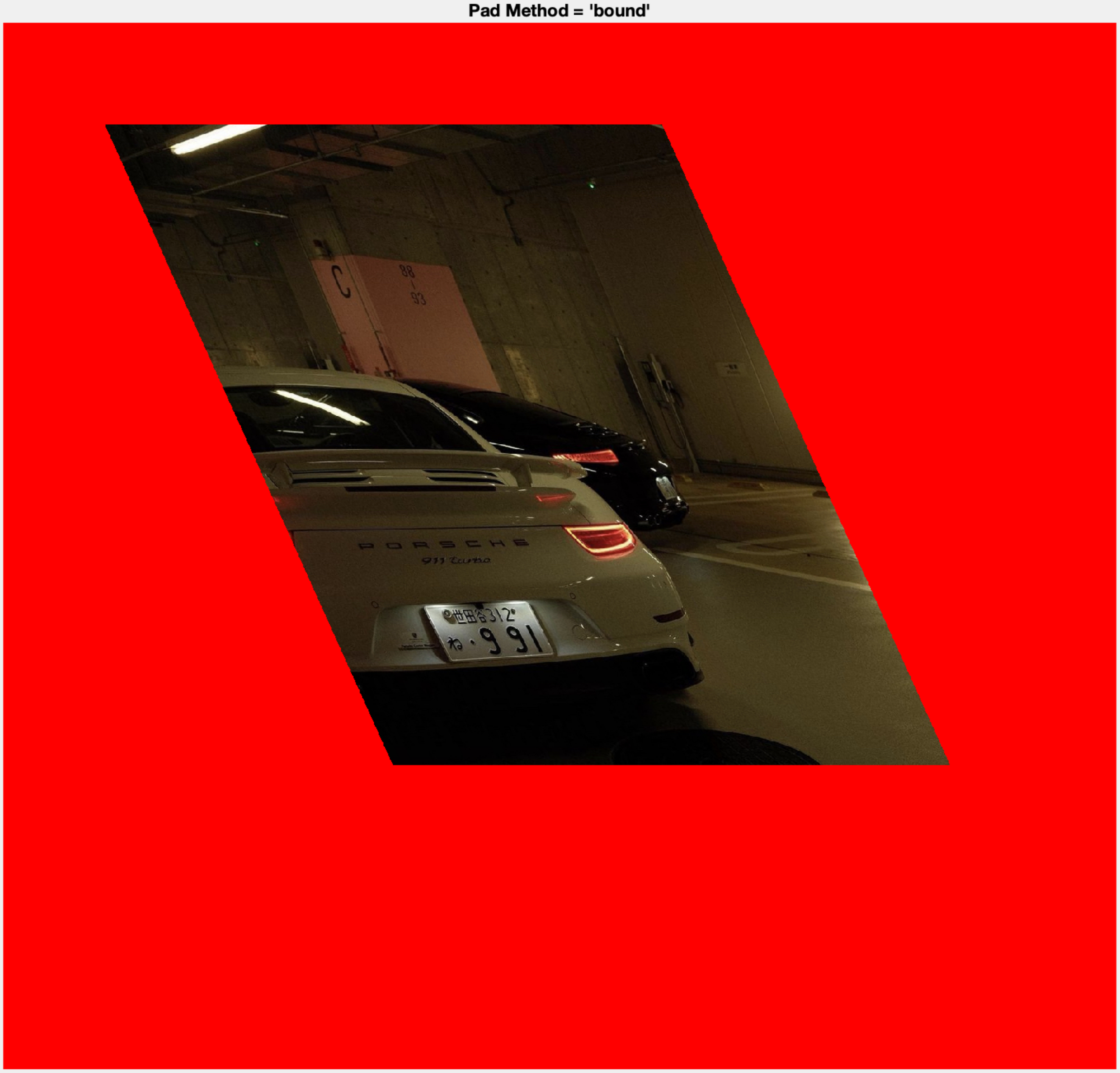
Pad Method = 'replicate'

Padding Image (bound):

Adjustments made for my image

```
R = makeresampler({'cubic','nearest'}, 'bound');
Bb = imtransform(image,T,R,'XData',[-149 1500],'YData',[-149 1400],...
                 'FillValues',red);
figure, imshow(Bb);
title('Pad Method = ''bound''');
```

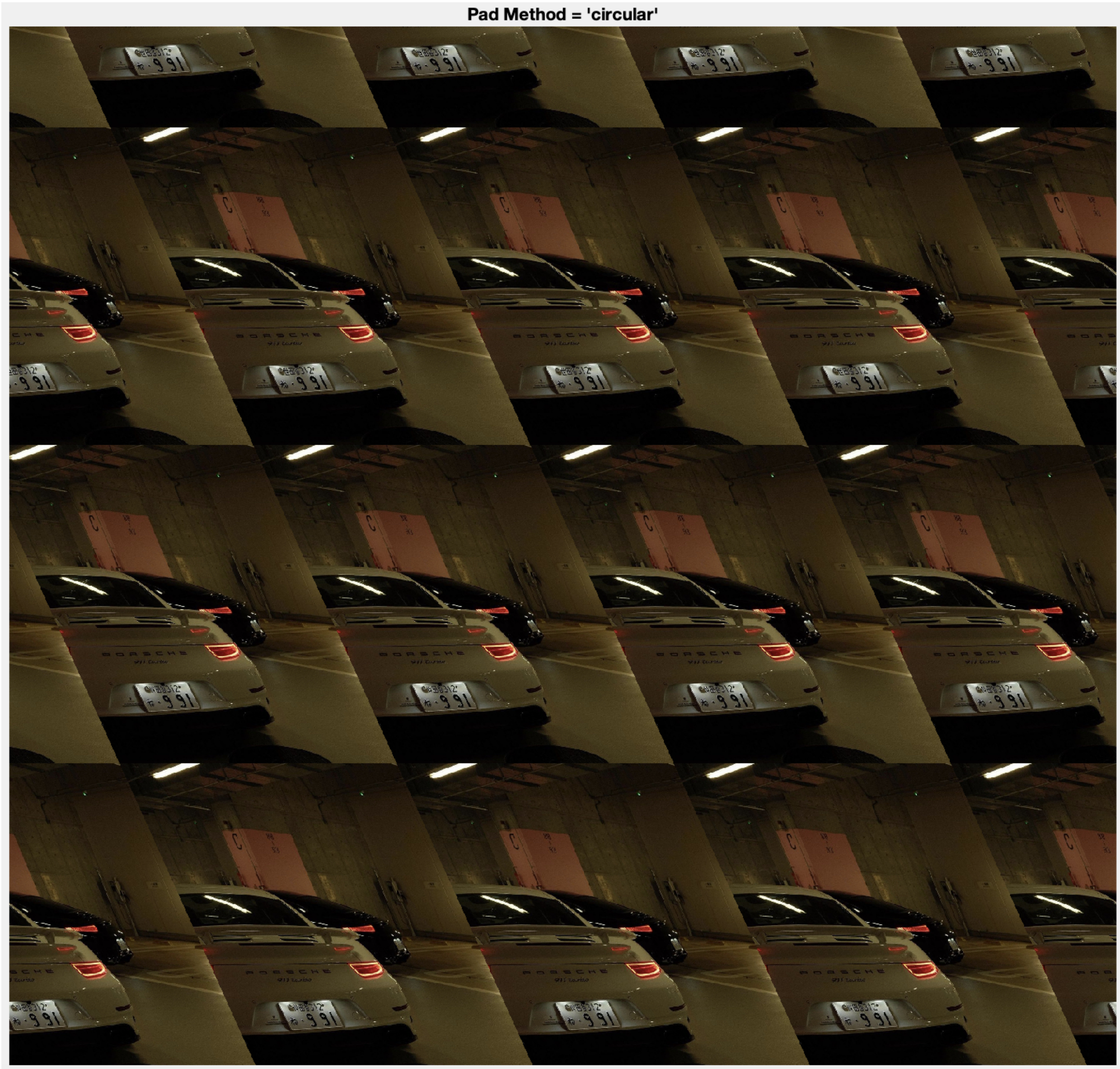
Pad Method = 'bound'



Padding Image (circular):

Adjustments made for my image

```
Thalf = maketform('affine',[1 0; a 1; 0 0]/2);

R = makeresampler({'cubic','nearest'},'circular');
Bc = imtransform(image,Thalf,R,'XData',[-149 1500],'YData',[-149 1400],...
                 'FillValues',red);
figure, imshow(Bc);
title('Pad Method = ''circular''');
```



Padding Image (symmetric):

Adjustments made for my image

```
R = makeresampler({'cubic','nearest'},'symmetric');
Bs = imtransform(image,Thalf,R,'XData',[-149 1500],'YData',[-149 1400],...
                  'FillValues',red);
figure, imshow(Bs);
title('Pad Method = ''symmetric''');
```

Pad Method = 'symmetric'

<u>References:</u>

MathWorks. (n.d.). Padding and Shearing an Image Simultaneously. In Image Processing Toolbox User's Guide (Version R2021b). MathWorks.

Toronto Metropolitan University. 2023. Course Content: Week 2. In Computer Vision, CPS 843. Toronto Metropolitan University's Learning Management System. https://courses.torontomu.ca/d2l/le/content/797209/viewContent/5350565/View

Toronto Metropolitan University. 2023. Course Content: Week 3. In Computer Vision, CPS 843. Toronto Metropolitan University's Learning Management System. https://courses.torontomu.ca/d2l/le/content/797209/viewContent/5360127/View