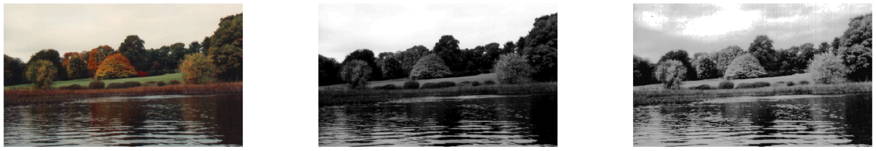
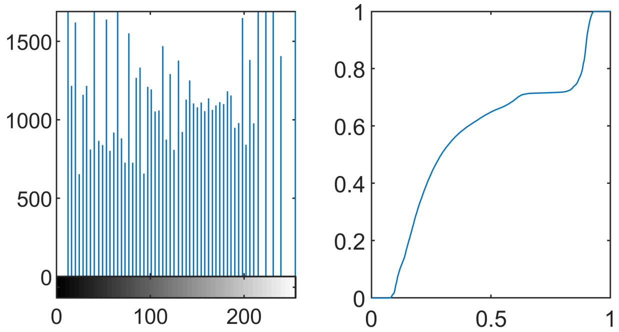
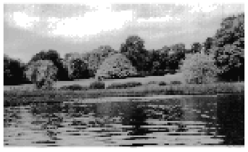
Histogram operations on colour images

```
%Consider the image moon.png. Develop a method to improve the contrast of the image.
% Show the result. (Show the transformation function).

%reading moon.png
moonimage=imread('autumn.tif');
%showing the original moon image
subplot(2,3,1),imshow(moonimage);

%moon.png is three dimensional image.So we can't use histogram equalization
% for the moon.png. Therefore
%change the image as gray scale image using the MATLAB function rgb2gray
moongrayimg = rgb2gray(moonimage);
% enhance contrast of image using the functions imadjust and stretchlim
adjustedmoonimage=imadjust(moongrayimg,stretchlim(moongrayimg, [0.05 0.95]));
%Display the image adjusted
subplot(2,3,2),imshow(adjustedmoonimage);
%get the histogram equalized image of modified image
moonhisteq=histeq(adjustedmoonimage);
%showing the histogram equalized image of modified image
subplot(2,3,3),imshow(moonhisteq);
%showing histogram of the histogram equalized image of modified image
subplot(2,3,4),imhist(moonhisteq);
%showing transformation function of output image of moon.png
normal=imhist(moongrayimg)./numel(moongrayimg);
cum=cumsum(normal);
x=linspace(0,1,256);
subplot(2,3,5),plot(x,cum);
```

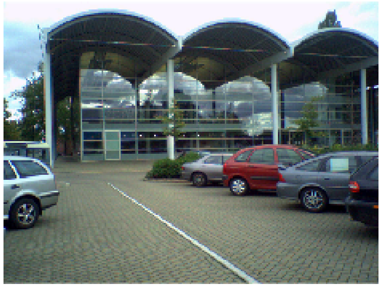


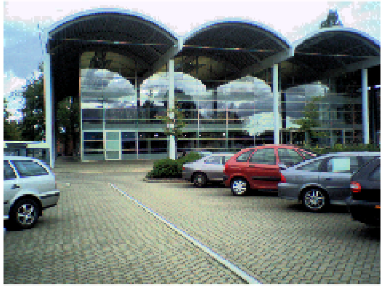


```
% Read in image
image=imread('images\carpark.png');
% Convert original to HSV image
imagehsv=rgb2hsv(image);
% Histogram equalise V (3rd) channel
V=histeq(imagehsv(:,:,3));
% Copy equalised V plane into (3rd) channel I2
imagehsv(:,:,3)=V;
% Convert imagehsv back to RGB form
outputimage=hsv2rgb(imagehsv);

subplot(1,2,1), imshow(image);
```



```
subplot(1,2,2), imshow(outputimage);
```





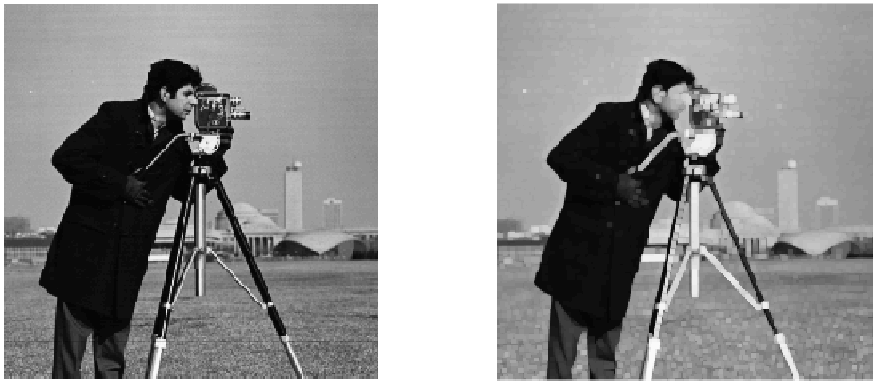
```
%The functions rgb2hsv and hsv2rgb are used to convert
% between the RGB and HSV colour spaces.
```

Enhancement via image filtering

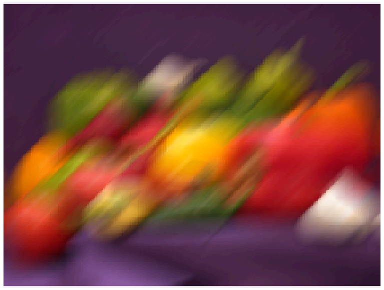
Here we specify func() as the max() filter function to apply over each and every 3�3 neighbourhood of the image. This replaces every input pixel in the output image with the maximumpixel value of the input pixel neighbourhood.Youmay wishtoexperiment with the effects of varying the neighbourhood dimensions and investigating the Matlab min() and mean() (for the latter, a type conversion will be required to display the double output type of the Matlab mean() function as an 8-bit image – specify the filter function as uint8(mean())).

The basic syntax is `function_name = @(variable_name) matlab_expression;`

```
%read an image
image1=imread("images\cameraman.tif");
%Display the original image
subplot(1,2,1),imshow(image1);
%Set a max filter using anonymous function
myFilterFunc=@(val) max(val(:));
%apply 3x3 neighborhood
%nlfilter erforms general sliding-neighborhood operations
filteredImage=nlfilter(image1,[3 3],myFilterFunc);
%display the filtered image
subplot(1,2,2),imshow(filteredImage);
```



```
%Read an image
image2=imread('peppers.png');
%Display the image
subplot(1,2,1), imshow(image2);
%Create a motion-blur filter using the fspecial function.
%Create a motion blur convolution kernel/filter
%fspecial creates predefined 2-d filter
% len specifies the length of the motion and
% theta specifies the angle of motion in degrees in a counter-clockwise direction
kernelMotion=fspecial('motion', 50, 45);

% imfilter filters the multidimensional array A with
% the multidimensional filter h and returns the result in B.
%Apply using symmetric mirroring at edges
filteredImage2=imfilter(image2,kernelMotion, 'symmetric');
%Display resultant image
subplot(1,2,2), imshow(filteredImage2);
```



Here we specify the fspecial() function to construct a kernel that will mimic the effect of motion blur (of specified length and angle) onto the image. Option 3) from our earlier discussion is used to deal with image edges during filtering. You may wish to investigate the use of other kernel filters that can be generated with the fspecial() function and the edge region filtering options available with the imfilter() function. Type doc imfilter at the Matlab prompt for details). How do they effect the image filtering result?

```
xy=@(x,y) (x.^2+y.^2);
xy(2,3)
```

```
ans = 13
```

```

myeq=@(x,y,z) (x.^3+2*x*y+z);
myeq(2,3,1)
```

```
ans = 21
```