Working formats in MATLAB

A 2D filter, F_1 has *a filter mask* or *coefficients* as defined below

1	2	1
2	4	2
1	2	1

In MATLAB this filter can be applied to an image using *out* = *conv*2(*pic,* F_1, '*same'*); The last argument ensures MATLAB extends the image with zeros when border pixels need to be output. The first argument is the picture and the middle argument is the mask of filter coefficients to be applied. The coefficients can be arranged in a 2D mask as in this case i.e. *F*1 or a row or column vector. Remember that in MATLAB if *g* is a row vector then *g'* is the transpose of this, i.e. a Column vector.

Preparation: Adding Noise to Images

We start off by preparing some "noisy" images, by artificially adding noise to these images. The Matlab function imnoise is able to add 3 types of noise – Gaussian, Salt-and-Pepper, and Speckle noise. As we have already been introduced to the first two types in lecture, the speckle noise is also known as *multiplicative noise*, where noise is modeled by random values multiplied by pixel values. This kind of noise is a major problem in some radar applications.

Sample usage for:

Adding salt-and-pepper noise:

```
>> x1 = imnoise(x, 'salt & pepper',0.05);
```

Adding Gaussian noise:

```
>> x2 = imnoise(x, 'gaussian',0.01);
```

Adding speckle noise:

```
>> x3 = imnoise(x, 'speckle',0.04);
```

Try different parameter values for each type of noise to see the different degree of effect. You may also add a combination of 2 or more types of noises into a same image (but those are very often unrealistic!)

Removing Noise from Images

Before we attempt to filter the noise created in step 1 earlier, we need to know a few types of filters that we can use (at least using Matlab's toolbox functions). The following are some filters that may be useful to remove noise:

1. Median filter – Sample code:

```
>> f1 = medfilt2(x, [3 3]);
```

2. Min/Max/Median filter – This uses a more general approach with the function, *ordfilt2*, which can be used to specify the rank-order of your choice. This line gives you the median filter (which can also be created with *medfilt2*):

```
>> f2 = ordfilt2(x,5,[1 1 1; 1 1 1; 1 1 1]);
```

Note: The second parameter specifies the rank order chosen. The third parameter specifies the mask with the neighbors specified by the nonzero elements in it.

Task: Figure out how to create Min or Max filters using this function.

3. Adaptive filter: Wiener filter

Adaptive filters are a class of filters that change their characteristics according to the values of the grayscales under the mask, making use of local statistical properties. Wiener filter, a nonlinear spatial filter, is a type of adaptive filter. Sample usage:

```
>> f3 = wiener2(x,[3 3]);
```