CSE464 – Digital Image Processing

Homework 3 – Report

Yasir Nacak – 161044042

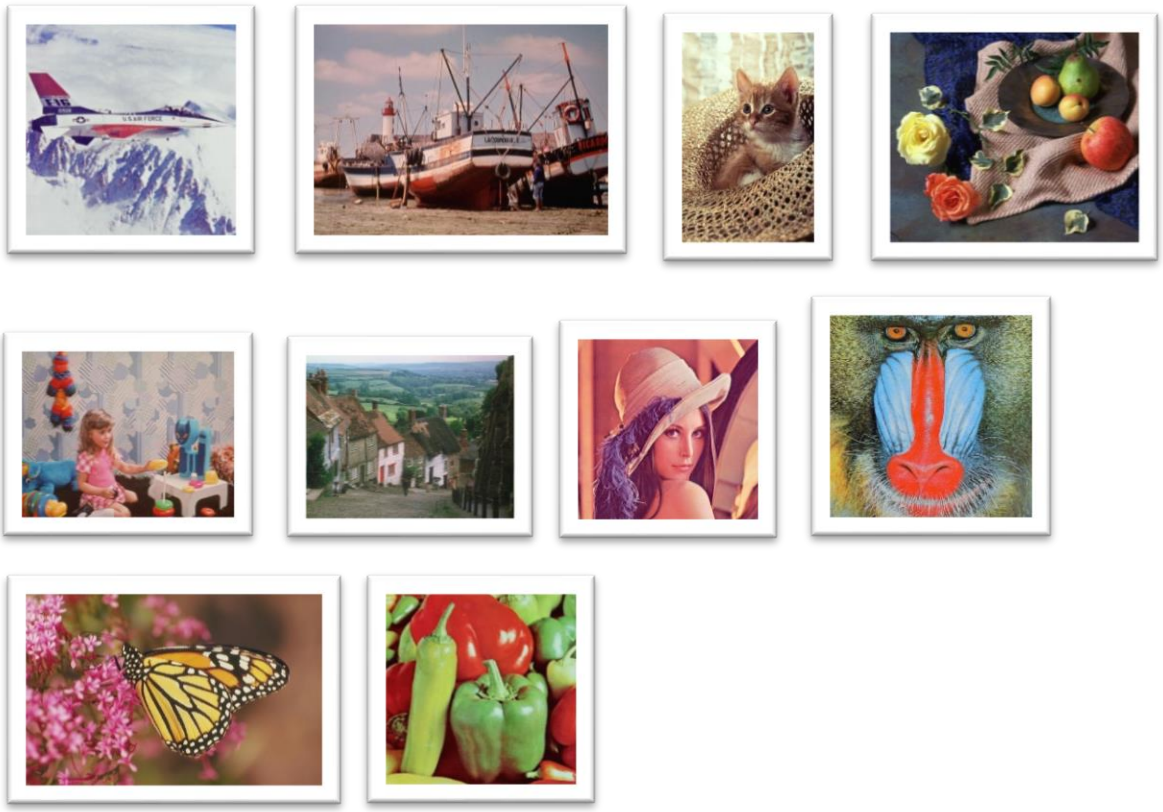
Q1.

In this part, I created four median filters that utilize three methods of pixel comparison and one marginal method that compares each channel separately. I then created a noise generator and applied the noise to a given image, then applied each median filter separately. I then collected the mean square error value of each filter. I tested my filters with given combinations of each noise percentage and filter size:

Noise Percentages = 10%, 25%, 50%, 75%

Filter Sizes = 5x5, 9x9, 11x11

Test Images:



From top left to bottom right: airplane.jpg, boats.jpg, cat.jpg, fruits.jpg, girl.jpg, goldhill.jpg, lena.jpg, mandril.jpg, monarch.jpg, peppers.jpg

Here is the table that shows the MSE values for 25% noise chance with 5x5 filter size:

	Lexicographical	Bitmix	Norm	Marginal
airplane.jpg	100514354	99360547	98216544	**93092661**
boats.jpg	229876580	224551674	222631549	**216869495**
cat.jpg	142711330	141144508	140570073	**138618811**
fruits.jpg	254381889	247360688	245272161	**236197356**
girl.jpg	106644217	103406168	101014368	**88750420**
goldhill.jpg	156481562	152589757	150596466	**143543942**
lena.jpg	56071915	53949430	52958610	**49976855**
mandril.jpg	488993560	465632286	461224994	**437076264**
monarch.jpg	212178231	209515386	206621358	**196800861**
peppers.jpg	73729582	61065989	59134944	**51408000**

Here is the table that shows the MSE values for 50% noise chance with 5x5 filter size:

	Lexicographical	Bitmix	Norm	Marginal
airplane.jpg	253072754	252254493	251220516	**247586771**
boats.jpg	578932400	575342978	573907942	**569516965**
cat.jpg	304790202	303516382	303048598	**301532915**
fruits.jpg	394493511	389029918	387450341	**380440308**
girl.jpg	344437336	342271248	340221587	**330862915**
goldhill.jpg	408559938	405951742	404244276	**398773973**
lena.jpg	185762143	184091132	183209054	**180865197**
mandril.jpg	684187277	667194430	663281345	**644508160**
monarch.jpg	568781589	566244747	564208143	**556419738**
peppers.jpg	196013672	187670064	186342598	**180568608**

Here is the table that shows the MSE values for 50% noise chance with 11x11 filter size:

	Lexicographical	Bitmix	Norm	Marginal
airplane.jpg	257836122	252877970	251608524	**242657094**
boats.jpg	633381478	615273592	609023584	**593330175**
cat.jpg	429973123	425145790	423516027	**423516027**
fruits.jpg	397082156	387199393	383415465	**365758976**
girl.jpg	310178726	292294933	293079521	**249942637**
goldhill.jpg	333090427	323493241	318156431	**302979211**
lena.jpg	138926995	132920256	130542388	**123725629**
mandril.jpg	678446076	640709112	633094322	**584954713**
monarch.jpg	752429968	741467296	732229108	**700259542**
peppers.jpg	198683212	151588886	142685942	**121208181**

Here is the table that shows the MSE values for 75% noise chance with 11x11 filter size:

	Lexicographical	Bitmix	Norm	Marginal
airplane.jpg	430947898	428190924	425857829	**418225106**
boats.jpg	1017976142	1006330482	1002731243	**990115914**
cat.jpg	638412682	634939475	633574462	**629962849**
fruits.jpg	553138337	544917108	542498261	**529753944**
girl.jpg	582370000	570872216	571887082	**540010043**
goldhill.jpg	604037371	596487282	592854305	**580810163**
lena.jpg	290485243	285926103	284176407	**278699522**
mandril.jpg	868031437	838885827	832878119	**794043973**
monarch.jpg	1224960678	1216738711	1208225761	**1181414418**
peppers.jpg	346811735	311973634	306174505	**290145231**

Each table has the minimum mean square error value highlighted for each image.

To compare our four different methods from these results, we can see that the Marginal filter has the best success in denoising images. Every possible change of variable has it as the least errored method. After that, the method yields a good result is the norm based median filters. A visual observation of the results also show that the lexicographical filter seems to have lots of artifacts. This is usually visible with low noise percentage and small filter sizes.

To analyze the general error values for each of the given cases, we can see that more the noise, harder to restore as a general case. This is apparent from the increase of the mean square error values for increasing noise percentage.

To analyze the effect of the filter size, we can see from these results that the filter size is a double-edged sword. While it helps us clear more noise from the image, it also causes detail to deteriorate the bigger it gets. But in general, to get an image that don't have any noise artifacts, bigger filter sizes are preferable.

To access all MSE values:

https://drive.google.com/open?id=1M6D0r8As4iWQY1EjoEcKcuXIF0-xoQ9O

To access all filtered images:

https://drive.google.com/open?id=1s94z-l7WT_TVvxBRm0mvr0WDDkW-drKO

Q2.

For this part, I wrote classifier class that loads up the trained data, take a K value, then loads each image to compare it with the others to find out which class does the image belong to. I wrote a histogram equalization method and a local binary pattern detection method. I then applied these methods to my test data and train data, then calculated the distances between each test data to each train data and selected the K closest distanced images to the test data. I then picked the most occurring class among those K train data.

1st Approach: No filtering

This approach yields very poor results. Since it is both varied by illumination and rotation, it does not serve our purpose well. For the 000 data, it gives 14.7% success and for the 001 data, it gives 16.8% success.

2nd Approach: Only Histogram Equalization

This approach seems not to work well. It yields almost no success. Since we are only comparing histograms of each image, this approach doesn't take much into account. It is also worse than no filtering. For the 000 data, it gives 4.9% success and for the 001 data, it gives 5.3% success.

3rd Approach: Only LBP – 8 points, 1 radius

In this approach, I used Local Binary Patterns method to get a more descriptive values of the pixels. This method is a lot more successful than the previous two methods. For the 000 data, it gives 51.9% success rate and for the 001 data, it gives 53.1% success. This method seems crude, but it is the fastest among all methods I experimented that has a 50%+ success rate on both datasets.

4th Approach: LBP – P=8, R=1 + Histogram Equalization

For this approach, the results are same with the 000 dataset while there is an improved result for the 001 dataset. The reason behind this is our approach is now closer to illumination invariance. For the 000 data, it still gives 51.9% success but for the 001 data, it gives 54.1% success.

5th Approach: LBP – P=8, increasing radii + Histogram Equalization

In this method, I increased the R value of the LBP operation and observed the results. There seems to be rise of success at start but after a certain value, the success rate drops. For 000 data, R=3 gives 56.3% success, R=5 gives 57.5%, R=8 gives 59.2%, R=16 gives 53.8%.

6th Approach: LBP – increasing points, R=8 + Histogram Equalization

For this final approach, I increased the number of points to be considered and observed the results. This also seems to increase the success rate. For 000 data, P=12 gives 62.4%, P=16 gives 64.0% and this is the best result I can get from this whole task. The more points to sample, the more running time increases so the results become harder to obtain.