Assignment 2

Question 1

#include<bits/stdc++.h>

#include<opencv2/opencv.hpp>

#include<opencv2/core/core.hpp>

#include<opencv2/highgui/highgui.hpp>

using namespace cv;

using namespace std;

int main()

{

//reading the image

Mat image = imread("Demo2.jpg",0);

//defining a matrix to store the image

Mat image0 = Mat::zeros(Size(500,380),CV\_8U);

for(int i=0;i<image.rows;i++)

{

for(int j=0;j<image.cols;j++)

{

image0.at<uchar>(i,j) = ((image.at<uchar>(i,j))/128)\*128;

}

}

namedWindow("image1",CV\_WINDOW\_FREERATIO);

imshow("image1",image0);

// Saving Image in file

imwrite("Q1o2.jpg",image0);

cout << "press enter " << endl;

waitKey(0);

Mat image1 = Mat::zeros(Size(500,380),CV\_8U);

for(int i=0;i<image.rows;i++)

{

for(int j=0;j<image.cols;j++)

{

image1.at<uchar>(i,j) = ((image.at<uchar>(i,j))/64)\*64;

}

}

namedWindow("image1",CV\_WINDOW\_FREERATIO);

imshow("image1",image1);

// Saving Image in file

imwrite("Q1o4.jpg",image1);

cout << "press enter " << endl;

waitKey(0);

Mat image2 = Mat::zeros(Size(500,380),CV\_8U);

for(int i=0;i<image.rows;i++)

{

for(int j=0;j<image.cols;j++)

{

image2.at<uchar>(i,j) = ((image.at<uchar>(i,j))/32)\*32;

}

}

namedWindow("image1",CV\_WINDOW\_FREERATIO);

imshow("image1",image2);

// Saving Image in file

imwrite("Q1o8.jpg",image2);

cout << "press enter " << endl;

waitKey(0);

Mat image3 = Mat::zeros(Size(500,380),CV\_8U);

for(int i=0;i<image.rows;i++)

{

for(int j=0;j<image.cols;j++)

{

image3.at<uchar>(i,j) = ((image.at<uchar>(i,j))/16)\*16;

}

}

namedWindow("image1",CV\_WINDOW\_FREERATIO);

imshow("image1",image3);

// Saving Image in file

imwrite("Q1o16.jpg",image3);

cout << "press enter " << endl;

waitKey(0);

Mat image4 = Mat::zeros(Size(500,380),CV\_8U);

for(int i=0;i<image.rows;i++)

{

for(int j=0;j<image.cols;j++)

{

image4.at<uchar>(i,j) = ((image.at<uchar>(i,j))/4)\*4;

}

}

namedWindow("image1",CV\_WINDOW\_FREERATIO);

imshow("image1",image4);

// Saving Image in file

imwrite("Q1o64.jpg",image4);

cout << "press enter " << endl;

waitKey(0);

return 0;

}

//g++-6 Q1.cpp `pkg-config --libs --cflags opencv`

Input image



Output images

2 levels



4 levels



8 levels



16 levels



64 levels



Conclusion

As the number of gray levels decrease, we lose all the gradients, and finer details.

As the number of gray levels decrease to very low values, we cannot observe any details, like the decorative drawing on the skin of the elephant, or the different coloured blocks on the curb.

Question 2

#include<bits/stdc++.h>

#include<opencv2/opencv.hpp>

#include<opencv2/core/core.hpp>

#include<opencv2/highgui/highgui.hpp>

using namespace cv;

using namespace std;

int main()

{

// Input -- Reading Image

Mat image = imread("Demo2.jpg",0);

// output -- final image

Mat imagef = Mat::zeros(Size(500,380),CV\_8U);

int k=0,l=0;

// we are choosing only those rows and columns when both of them will only odd and saving in our

// final image.

for(int i=0;i<(image.rows);i++)

{

for(int j=0;j<(image.cols)\*4;j++)

{

if(i%2 != 0 && j%2 != 0)

{

imagef.at<uchar>(l,k) = image.at<uchar>(i,j);

k++;

}

}

if(l&1)

{

l++;

}

}

namedWindow("image1",CV\_WINDOW\_FREERATIO);

imshow("image1",imagef);

// Saving Image in file

imwrite("Q2i.jpg",imagef);

waitKey(0);

return 0;

}

//g++-6 Q2.cpp `pkg-config --libs --cflags opencv`

Input image



Output image



Assignment 3

Average of 5 images

#include<bits/stdc++.h>

#include<opencv2/opencv.hpp>

#include<opencv2/core/core.hpp>

#include<opencv2/highgui/highgui.hpp>

using namespace cv;

using namespace std;

int main()

{

// Input -- reading images

Mat image = imread("Cat0.pgm",0);

Mat image1 = imread("Cat1.pgm",0);

Mat image2 = imread("Cat2.pgm",0);

Mat image3 = imread("Cat3.pgm",0);

Mat image4 = imread("Cat4.pgm",0);

// Result image -- initially black

Mat image0 = Mat::zeros(Size(200,247),CV\_8U);

// Adding corresponding elements of arrays and saving in resulting array image0

for(int i=0;i<image.rows;i++)

{

for(int j=0;j<image.cols;j++)

{

image0.at<uchar>(i,j) = (image.at<uchar>(i,j) + image1.at<uchar>(i,j) + image2.at<uchar>(i,j) + image3.at<uchar>(i,j) + image4.at<uchar>(i,j) )\*0.2;

}

}

// Checking the output

namedWindow("image1",CV\_WINDOW\_FREERATIO);

imshow("image1",image0);

// Saving Image in file

imwrite("5ocat.jpg",image0);

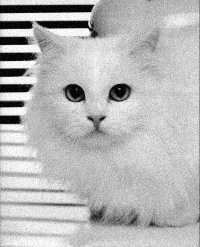
waitKey(0);

return 0;

}

//g++-6 5cat.cpp `pkg-config --libs --cflags opencv`

Output



Average of 10 images

#include<bits/stdc++.h>

#include<opencv2/opencv.hpp>

#include<opencv2/core/core.hpp>

#include<opencv2/highgui/highgui.hpp>

using namespace cv;

using namespace std;

int main()

{

// Input -- reading images

Mat image = imread("Cat0.pgm",0);

Mat image1 = imread("Cat1.pgm",0);

Mat image2 = imread("Cat2.pgm",0);

Mat image3 = imread("Cat3.pgm",0);

Mat image4 = imread("Cat4.pgm",0);

Mat image5 = imread("Cat5.pgm",0);

Mat image6 = imread("Cat6.pgm",0);

Mat image7 = imread("Cat7.pgm",0);

Mat image8 = imread("Cat8.pgm",0);

Mat image9 = imread("Cat9.pgm",0);

// Result image -- initially black

Mat image0 = Mat::zeros(Size(200,247),CV\_8U);

// Adding corresponding elements of arrays and saving in resulting array image0

for(int i=0;i<image.rows;i++)

{

for(int j=0;j<image.cols;j++)

{

image0.at<uchar>(i,j) = (image.at<uchar>(i,j) + image1.at<uchar>(i,j) + image2.at<uchar>(i,j) + image3.at<uchar>(i,j) + image4.at<uchar>(i,j) + image5.at<uchar>(i,j)+ image6.at<uchar>(i,j)+ image7.at<uchar>(i,j)+ image8.at<uchar>(i,j)+ image9.at<uchar>(i,j) )\*0.1;

}

}

// Checking the output

namedWindow("image1",CV\_WINDOW\_FREERATIO);

imshow("image1",image0);

// Saving Image in file

imwrite("10ocat.jpg",image0);

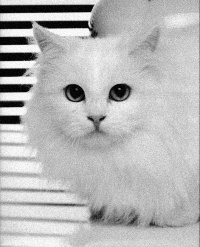
waitKey(0);

return 0;

}

//g++-6 10cat.cpp `pkg-config --libs --cflags opencv`

Output Image



Average of 20 images

#include<bits/stdc++.h>

#include<opencv2/opencv.hpp>

#include<opencv2/core/core.hpp>

#include<opencv2/highgui/highgui.hpp>

using namespace cv;

using namespace std;

int main()

{

// Input -- reading images

Mat image = imread("Cat0.pgm",0);

Mat image1 = imread("Cat1.pgm",0);

Mat image2 = imread("Cat2.pgm",0);

Mat image3 = imread("Cat3.pgm",0);

Mat image4 = imread("Cat4.pgm",0);

Mat image5 = imread("Cat5.pgm",0);

Mat image6 = imread("Cat6.pgm",0);

Mat image7 = imread("Cat7.pgm",0);

Mat image8 = imread("Cat8.pgm",0);

Mat image9 = imread("Cat9.pgm",0);

Mat image10 = imread("Cat10.pgm",0);

Mat image11 = imread("Cat11.pgm",0);

Mat image12 = imread("Cat12.pgm",0);

Mat image13 = imread("Cat13.pgm",0);

Mat image14 = imread("Cat14.pgm",0);

Mat image15 = imread("Cat15.pgm",0);

Mat image16 = imread("Cat16.pgm",0);

Mat image17 = imread("Cat17.pgm",0);

Mat image18 = imread("Cat18.pgm",0);

Mat image19 = imread("Cat19.pgm",0);

// Result image -- initially black

Mat image0 = Mat::zeros(Size(200,247),CV\_8U);

// Adding corresponding elements of arrays and saving in resulting array image0

for(int i=0;i<image.rows;i++)

{

for(int j=0;j<image.cols;j++)

{

image0.at<uchar>(i,j) = (image.at<uchar>(i,j) + image1.at<uchar>(i,j) + image2.at<uchar>(i,j) + image3.at<uchar>(i,j) + image4.at<uchar>(i,j) + image5.at<uchar>(i,j)+ image6.at<uchar>(i,j)+ image7.at<uchar>(i,j)+ image8.at<uchar>(i,j)+ image9.at<uchar>(i,j) + image10.at<uchar>(i,j)+ image11.at<uchar>(i,j)+ image12.at<uchar>(i,j)+ image13.at<uchar>(i,j)+ image14.at<uchar>(i,j)+ image15.at<uchar>(i,j)+ image16.at<uchar>(i,j)+ image17.at<uchar>(i,j)+ image18.at<uchar>(i,j)+ image19.at<uchar>(i,j) )\*0.05;

}

}

// Checking the output

namedWindow("image1",CV\_WINDOW\_FREERATIO);

imshow("image1",image0);

// Saving Image in file

imwrite("20ocat.jpg",image0);

waitKey(0);

return 0;

}

//g++-6 20cat.cpp `pkg-config --libs --cflags opencv`

Output Image



Average of 30 images

#include<bits/stdc++.h>

#include<opencv2/opencv.hpp>

#include<opencv2/core/core.hpp>

#include<opencv2/highgui/highgui.hpp>

using namespace cv;

using namespace std;

int main()

{

// Input -- reading images

Mat image = imread("Cat0.pgm",0);

Mat image1 = imread("Cat1.pgm",0);

Mat image2 = imread("Cat2.pgm",0);

Mat image3 = imread("Cat3.pgm",0);

Mat image4 = imread("Cat4.pgm",0);

Mat image5 = imread("Cat5.pgm",0);

Mat image6 = imread("Cat6.pgm",0);

Mat image7 = imread("Cat7.pgm",0);

Mat image8 = imread("Cat8.pgm",0);

Mat image9 = imread("Cat9.pgm",0);

Mat image10 = imread("Cat10.pgm",0);

Mat image11 = imread("Cat11.pgm",0);

Mat image12 = imread("Cat12.pgm",0);

Mat image13 = imread("Cat13.pgm",0);

Mat image14 = imread("Cat14.pgm",0);

Mat image15 = imread("Cat15.pgm",0);

Mat image16 = imread("Cat16.pgm",0);

Mat image17 = imread("Cat17.pgm",0);

Mat image18 = imread("Cat18.pgm",0);

Mat image19 = imread("Cat19.pgm",0);

Mat image20 = imread("Cat20.pgm",0);

Mat image21 = imread("Cat21.pgm",0);

Mat image22 = imread("Cat22.pgm",0);

Mat image23 = imread("Cat23.pgm",0);

Mat image24 = imread("Cat24.pgm",0);

Mat image25 = imread("Cat25.pgm",0);

Mat image26 = imread("Cat26.pgm",0);

Mat image27 = imread("Cat27.pgm",0);

Mat image28 = imread("Cat28.pgm",0);

Mat image29 = imread("Cat29.pgm",0);

// Result image -- initially black

Mat image0 = Mat::zeros(Size(200,247),CV\_8U);

// Adding corresponding elements of arrays and saving in resulting array image0

for(int i=0;i<image.rows;i++)

{

for(int j=0;j<image.cols;j++)

{

image0.at<uchar>(i,j) = (image.at<uchar>(i,j) + image1.at<uchar>(i,j) + image2.at<uchar>(i,j) + image3.at<uchar>(i,j) + image4.at<uchar>(i,j) + image5.at<uchar>(i,j)+ image6.at<uchar>(i,j)+ image7.at<uchar>(i,j)+ image8.at<uchar>(i,j)+ image9.at<uchar>(i,j) + image10.at<uchar>(i,j)+ image11.at<uchar>(i,j)+ image12.at<uchar>(i,j)+ image13.at<uchar>(i,j)+ image14.at<uchar>(i,j)+ image15.at<uchar>(i,j)+ image16.at<uchar>(i,j)+ image17.at<uchar>(i,j)+ image18.at<uchar>(i,j)+ image19.at<uchar>(i,j) + image20.at<uchar>(i,j) + image21.at<uchar>(i,j)+ image22.at<uchar>(i,j)+ image23.at<uchar>(i,j)+ image24.at<uchar>(i,j)+ image25.at<uchar>(i,j)+ image26.at<uchar>(i,j)+ image27.at<uchar>(i,j)+ image28.at<uchar>(i,j)+ image29.at<uchar>(i,j))\*0.0333;

}

}

// Checking the output

namedWindow("image1",CV\_WINDOW\_FREERATIO);

imshow("image1",image0);

// Saving Image in file

imwrite("30ocat.jpg",image0);

waitKey(0);

return 0;

}

//g++-6 30cat.cpp `pkg-config --libs --cflags opencv`

Output image



Average of 40 images

#include<bits/stdc++.h>

#include<opencv2/opencv.hpp>

#include<opencv2/core/core.hpp>

#include<opencv2/highgui/highgui.hpp>

using namespace cv;

using namespace std;

int main()

{

// Input -- reading images

Mat image = imread("Cat0.pgm",0);

Mat image1 = imread("Cat1.pgm",0);

Mat image2 = imread("Cat2.pgm",0);

Mat image3 = imread("Cat3.pgm",0);

Mat image4 = imread("Cat4.pgm",0);

Mat image5 = imread("Cat5.pgm",0);

Mat image6 = imread("Cat6.pgm",0);

Mat image7 = imread("Cat7.pgm",0);

Mat image8 = imread("Cat8.pgm",0);

Mat image9 = imread("Cat9.pgm",0);

Mat image10 = imread("Cat10.pgm",0);

Mat image11 = imread("Cat11.pgm",0);

Mat image12 = imread("Cat12.pgm",0);

Mat image13 = imread("Cat13.pgm",0);

Mat image14 = imread("Cat14.pgm",0);

Mat image15 = imread("Cat15.pgm",0);

Mat image16 = imread("Cat16.pgm",0);

Mat image17 = imread("Cat17.pgm",0);

Mat image18 = imread("Cat18.pgm",0);

Mat image19 = imread("Cat19.pgm",0);

Mat image20 = imread("Cat20.pgm",0);

Mat image21 = imread("Cat21.pgm",0);

Mat image22 = imread("Cat22.pgm",0);

Mat image23 = imread("Cat23.pgm",0);

Mat image24 = imread("Cat24.pgm",0);

Mat image25 = imread("Cat25.pgm",0);

Mat image26 = imread("Cat26.pgm",0);

Mat image27 = imread("Cat27.pgm",0);

Mat image28 = imread("Cat28.pgm",0);

Mat image29 = imread("Cat29.pgm",0);

Mat image30 = imread("Cat30.pgm",0);

Mat image31 = imread("Cat31.pgm",0);

Mat image32 = imread("Cat32.pgm",0);

Mat image33 = imread("Cat33.pgm",0);

Mat image34 = imread("Cat34.pgm",0);

Mat image35 = imread("Cat35.pgm",0);

Mat image36 = imread("Cat36.pgm",0);

Mat image37 = imread("Cat37.pgm",0);

Mat image38 = imread("Cat38.pgm",0);

Mat image39 = imread("Cat39.pgm",0);

// Result image -- initially black

Mat image0 = Mat::zeros(Size(200,247),CV\_8U);

// Adding corresponding elements of arrays and saving in resulting array image0

for(int i=0;i<image.rows;i++)

{

for(int j=0;j<image.cols;j++)

{

image0.at<uchar>(i,j) = (image.at<uchar>(i,j) + image1.at<uchar>(i,j) + image2.at<uchar>(i,j) + image3.at<uchar>(i,j) + image4.at<uchar>(i,j) + image5.at<uchar>(i,j)+ image6.at<uchar>(i,j)+ image7.at<uchar>(i,j)+ image8.at<uchar>(i,j)+ image9.at<uchar>(i,j) + image10.at<uchar>(i,j)+ image11.at<uchar>(i,j)+ image12.at<uchar>(i,j)+ image13.at<uchar>(i,j)+ image14.at<uchar>(i,j)+ image15.at<uchar>(i,j)+ image16.at<uchar>(i,j)+ image17.at<uchar>(i,j)+ image18.at<uchar>(i,j)+ image19.at<uchar>(i,j) + image20.at<uchar>(i,j) + image21.at<uchar>(i,j)+ image22.at<uchar>(i,j)+ image23.at<uchar>(i,j)+ image24.at<uchar>(i,j)+ image25.at<uchar>(i,j)+ image26.at<uchar>(i,j)+ image27.at<uchar>(i,j)+ image28.at<uchar>(i,j)+ image29.at<uchar>(i,j)+ image30.at<uchar>(i,j)+ image31.at<uchar>(i,j)+ image32.at<uchar>(i,j)+ image33.at<uchar>(i,j)+ image34.at<uchar>(i,j)+ image35.at<uchar>(i,j)+ image36.at<uchar>(i,j)+ image37.at<uchar>(i,j)+ image38.at<uchar>(i,j)+ image39.at<uchar>(i,j))\*0.025;

}

}

// Checking the output

namedWindow("image1",CV\_WINDOW\_FREERATIO);

imshow("image1",image0);

// Saving Image in file

imwrite("40ocat.jpg",image0);

waitKey(0);

return 0;

}

//g++-6 40cat.cpp `pkg-config --libs --cflags opencv`

Output image



Conclusion

The amount of noise in the output image will reduce when the number of images that are combined together are increased. As all input images have different noise patterns, averaging all them will result in a smoother image.

The difference in the amount of noise removed will decrease as we increase the number of images. If we take a small number of images, we will still have noise in the output. If we take a number that is too high, we will get a smooth image, but we can get a similar quality of image, by using a smaller number of images. The returns in quality of the output will decrease as we increase the number of images.

In this example, combining 20 images will give us a slightly noisy image. On combining 30 images, we are not able to observe noise. Here, on combining 40 images, the quality of the image obtained is similar to 30 images.