**1. PSNR 코드**

**Main.cpp**

#include “RawInfoDig.h

#include “viewerDig.h”

Private:

Unsigned char\*\* Ori,, \*\*Dst;

Int nHeight \_ori , nWidth\_ori;

Int nHeight\_Dst, nWidth\_ost”

CEdit Edit\_File\_Ori, Edit\_File\_Dst, Edit\_PSNR;

Double GetPSNR(unsigned char\*\* ori, unsigned char \*\* Dst, int nHeight, int nWidth);

Public:

afx\_msg void OnOriopenButtonClicked();

afx\_msg void OnNoiseopenButtonClicked();

afx\_msg void OnGetpsnrButtonClicked();

afx\_msg void OnBnClickedButtonClose();

void CPSNRDig::OnOriopenButtonClicked()

{

CFileDialog dig(TRUE);

CRawinfoDig RawDig;

If (Ori !=NULL)

{

NemFree2D(ori, nHeight\_ori);

}

If(dig.GetFileExt() != “ raw” && dig. GetFileExt() !=”RAW”)

{

MessageBox(“파일 확장자가 raw 파일이 아닙니다. “);

Return;

}

FILE\* fp;

Fopen\_s(&fp, dig,GetFileName(), “rb”);

If (RawDig.DoModal() == IDOK)

{

nHeight\_ori = RawDig,GetRawHeight();

nWidth\_ori = RawDig.GetRawWidth();

if (nHeight\_ori == 0 || nWidth\_ori == 0 || RawDig.GetRawFormat() !=3)

{

MessageBox(“입력 값이 잘못되었습니다.”);

Return;

}

Edit\_File\_ori.SetWindowTextA(dig.GetFileTitle());

Ori = MemAlloc2D(nHeight\_Ori; h++)

{

fread(ori[h], sizeof(unsigned char) , nWidth\_Ori, fp);

}

}

Fclose(fp);

}

}

Void CPSNDig::OnNoiseopenButtonClicked()

{

CFileDialog dig(TRUE);

CRawinfoDig RawDig;

If(Dst !=NULL)

{

MemFree2D(Dst, nHeight\_Dst);

}

If (dig.DoModal() == IDOK)

{

If( dig.GetFileExt() != “raw” && dig.GetFileExt() != “RAW)

{

MessageBox(“파일 확장자가 raw 파일이 아닙니다.”);

Return;

}

FILE \*fp;

Fopen\_s(&fp, dig.GetFileName(), “rb”);

If (RawDig.DoModal() == IDOK)

{

nHeight\_Dst = RawDig.GetRawHeight();

nWidth\_Dst = RawDig.GetRawidth();

if(nHeight\_Dst =n= 0 || nWidth\_Dst == 0 || RawDig.GetRawFormat() !=3)

{

MessageBox(“입력 값이 잘못되었습니다.”);

Return;

}

Edit\_File\_Dst\_SetWindowTextA(dig.GetFileTitle());

Dst = MemAlloc2D(nHeight\_Ori, nWidth\_Ori, 0);

For(int h =0; h<nHeight\_Dst; h++)

{

Fread(Ori[h] , sizeof(unsigned char), nWidth\_Dst, fp);

}

}

Fclose(fp);

}

}

Void CPSNRDig :: OnGetpsnrButtonClicked()

{

If(Ori == NULL || Dst == NULL)

{

MessageBox(“원본 영상과 비교 영상을 불러오지 않았습니다.”);

Return;

}

If(nHeight\_Ori !=nHeight\_Dst || nWidth\_Ori !=nWidth\_Dst)

{

MessageBox(“원본 영상과 비교 영상의 크기가 다릅니다.”)’

Return;

}

CString strTemp;

Double dPSNR = GetPSNR(Ori, Dst, nHeight\_ori, nWidth\_ori);

strTemp.Format(\_T(“%f”), dPSNR);

Edit\_PSNR.SetWindowTextA(strTemp);

}

Double CPSNRDig::GetPSNR(unsigned char \*\* Ori, unsigned char \*\* Dst, int nHeight, int nWidth)

{

}

Void CPSNRDIG :: OnBnClickedButtonClose()

{

If(Ori !=NULL)

{

MemFree2D(Ori, nHeight\_Ori);

}

If (Dst !=NULL)

{

MemFree2D(Dst, nWidth\_Dst);

}

CDialog::OnCancel():

}

**viewerDig.h**

#include “PSNRDig.h”

**viewerDig.cpp**

void CViewerDig :: OnGetpsnr()

{

CPSNRDig PSNRDig;

PSNRDig.DoModal();

}

**2. 노이즈 이미지 생성**

Add 노이즈 새로 생성 및 tap Control 추가

AddNoiseTab\_1.h

Public:

CEdit Edit\_GN\_Mean;

CEdit Edit\_GN\_Stdev;

AddNoiseTab\_1.cpp

Void CAddNoiseTab\_1::DoDataExchange(CDataExchange\* pDx)

{

CDialogEx::DoDataExchange(pDx);

DDX\_Control(pDx, IDC\_EDIT)GNMEAN, Edit\_GN\_Mean);

DDX\_Control(pDx, IDC\_EDIT, GNSTDEV, Edit\_GN\_Stdev);

}

AddNoiseTab\_2.h

Public:

CEdit Edit\_SN\_prob;

CEdit Edit\_PN\_prob;

AddNoiseTab\_2.cpp

Void CAdNoiseTab\_2::DoDataExchange(CDataExchange\* pDX)

{

CDialogEx::DoDataExchange(pDX);

DDX\_Control(pDX, IDC\_EDIT\_SPROB, Edit\_SN\_prob);

DDX\_Control(pDX, IDC\_EDIT\_PPROB, Edit\_PN\_Prob);

**ADDNoiseDig.h**

#pragma once

#include “AddNoiseTab\_1.h”

#include “AddNoiseTab\_2.h”

Private:

CAddNoiseTab\_1 Tab1;

CAddNoiseTab\_2 Tab2;

CWnd\* m\_pwndShow;

Float value\_1;

Float value\_2;

Int NoiseType;

Public:

Float GetVal\_1();

Float GetVal\_2();

Int GetNoiseType();

BOOL OninitDialog();

**AddNoiseDialog.cpp**

BOOL CAddNoiseDig::OninitDialog()

{

CDialog::OnInitDialog();

CString strTab = \_T("");

strTab.Format(\_T("Gaussian"));

this->m\_mainTab.Insertitem(1, strTab, 0);

strTab.Format(\_T("Salt&Pepper"));

this->m\_mainTab.InsertItem(1, strTab, 0);

CRect rect;

this->m\_mainTab.GetClientRect(&rect);

this->Tab1.Create(IDO\_ADDNOISE\_TAB!, &this->m\_mainTab);

this->Tab1.SetWindowPos(NULL, 5, 25, rect.Width() - 10, rect.Height() - 30, SWP\_SHOWWINDOW | SWP\_NOZORDER);

this->Tab2.Create(IDD\_ADDNOISE\_TAB2, &this->m\_mainTab);

this->Tab2.SetWindowPos(NULL, 5, 25, rect.Width() - 10, rect.Height() - 30, SWP\_NOZORDER);

this->m\_pwndShow = &this->Tab1;

return TRUE;

}

float CAddNoiseDig::GetVal\_1()

{

return value\_1;

}

float CAddNoiseDig::Getval\_2()

{

return value\_2;

}

int CAddNoiseDig::GetNoiseType()

{

return NoiseType;

}

void CAddNoiseDig::OnTcnSelchangeAddnoisesetab(NMHDR \*pNHDR, LRESULT \* pResult)

{

if (this->m\_pwndShow != NULL)

{

this->m\_pwndShow->ShowWindow(SW\_HIDE);

this->m\_pwndShow = NULL;

}

int tabindex = this->m\_mainTab.GetCursel();

switch (tabindex)

{

case 0:

this->tab1.SgowWindow(SW\_SHOW);

this->m\_pwndShow = &this->tab1;

break;

case 1:

this->Tab2.ShowWindow(SW\_SHOW);

this->m\_pwndShow = &this->Tab2;

break;

}

\*pResult = 0;

}

void CAddNoiseDig::OnOkAnButtonClicked()

{

CString str1, str2;

CAddNoiseTab\_1\* Tab\_1 = NULL;

CAddNoiseTab\_2\* Tab\_2 = NULL;

int tabindex = this->m\_mainTab.GetCurSel();

switch (tabindex)

{

case 0:

Tab\_1 = (CAddNoiseTab\_1\*)this->m\_pwndShow;

Tab\_1->Edit\_GN\_Mean, GetWindowTextA(str1);

Tab\_1->Edit\_GN\_Stdev.GetwindowTextA(str2);

NoiseType = 0;

break;

case 1:

Tab\_2 = (CAddNoiseTab\_2\*)this->m\_pwndShow;

Tab\_2->Edit\_SN\_prob.GetWindowTextA(str1);

Tab\_2->Edit\_PN\_prob.GetWindowTextA(str2);

NoiseType = 1;

break;

}

**ViewerDig**

#include "opencv2\opencv.hpp"

#include "RawinfoDig.h"

#include "imgScalingDig.h"

#include "rotation.h"

#include "FilterDig.h"

#include "AddNoiseDig.h"

enum FilterType

{

FILTER\_AVERAGE,

FILTER\_SMOOTH,

FILTER\_SHARPEN,

FILTER\_GAUSSIAN

enum NoiseType

{

NOISE\_GN,

NOISE\_SP

};

if (dig.doModal() == IDOK)

{

if (dig.GetFileExt() != "jpg" && dig.GetFileExt() != )

{

MessageBox("JPG 또는 YUV, Gray 파일이 아닙니다.")

return;

}

InpFileName = dig.GetFileTitle();

if (dig.GetFileExt() == "jpg" || dig.GetFileExit() == )

{

Ori\_img = imread(const char\*)dig.GetPathName());

nHeight\_in + Ori\_img.rows;

nWidth\_in = Ori\_img.cols;

}

};

void CviewerDig::OnNoisereductionAddnoiseClicked()

{

CAddNoiseDig AddNoiseDig;

BYTE\*\* Noiseimg = MemAlloc2D(nHeight\_in, nWidth\_in, 0);

if(isimageOpened && nFormat == FORMAT\_GRAY)

{

if (AddNoiseDig.DoModal() == IDOK)

{

CString OutFileName = inpFileName;

float Value\_1 = AddNoiseDig.Getval\_1();

float Value\_2 = AddNoiseDig.GetVal\_2();

int nNoiseType = AddNoiseDig.GetNoiseType();

if (nNoiseType == NOISE\_GN)

{

if (Value\_2 < 1)

{

MessageBox("표준편차 값이 잘못되었습니다. 표준편차 값은 1 이상이여야 합니다.");

MemFree2D(Noiseimg, nHeight\_in);

return;

}

inputGaussianNoise(ch\_in\_gray, Noiseimg, nHeight\_in, nWidth\_in, Value\_1, Value\_2);

OutFileName += " \_GN\_raw";

}

else

{

if (Value\_1 <= 0 && Value\_2 <= 0)

{

MessageBox(" 두 개의 확률 값 중 적어도 하나는 0보다 커야 합니다.");

MemFree2D(Noiseimg, nHeight\_in);

return;

}

inputSaltPepperNoise(ch\_in\_gray, Noiseimg, nHeight\_in, nWidth\_in, Value\_1, Value\_2);

OutFileName += "SPN.raw";

}

FILE \* fp;

fope\_s(&fp, OutFileName, "wb");

for (int h = 0; h < nHeight\_in; h++)

{

fwrite(Noiseimg[h], sizeof(unsigned char), nWidth\_in, fp);

}

fclose(fp);

MemFree2D(Noiseimg, nHeight\_in);

}

void InputGaussianNoise(unsigned char\*\* In, unsigned char\*\* Out, int nHeight, int nWidth, float fMean, float fStdev)

{

float fTemp = 0, fPDF[256] = { 0.0f };

GetGaussianPDF(fPDF, 256, fMean, fStdev);

srand(GetTickCount());

for (int h = 0; h < nHeight; h++)

{

for (int w = 0; w < nWidth; w++)

{

fTemp = (float)In[h][w] + GetNoise(fPDF, 256);

Out[h][w] = static\_cast<unsigned char>(fTemp);

}

}

}

void GetGaussianPDF(float\* EmptyPDF, int nLength, float fMean, float fStDev)

{

int n;

int Center = nLength / 2;

float x;

for (n = 0; n<nLength; n++)

{

x = (float)(n - Center);

EmptyPDF[n] = (1 / ((float)sqrt(2 \* PI) \* fStDev)) \* exp(-pow(x - fMean, 2) / (2 \* fStDev \* fStDev));

}

}

float GetNoise(float\* PDF, int nLength)

{

int n;

int Center = nLength / 2;

float fRand, fComp, fTemp = 0;

float x = 0, fDiff;

float\* CDF = new float[nLength];

CDF[0] = 0;

fRand = (float)rand() / (RAND\_MAX + 1); // 0~1 Uniform Distribution

for (n = 1; n<nLength; n++)

{

CDF[n] = (PDF[n] + PDF[n - 1]) / 2 + CDF[n - 1];

fDiff = fRand - CDF[n];

if (fDiff < 0)

{

x = ((float)n - Center);

break;

}

}

fComp = (fRand - CDF[n - 1]) / (CDF[n] - CDF[n - 1]);

delete[] CDF;

void InputSaltPepperNoise(unsigned char\*\* In, unsigned char\*\* Out, int nHeight, int nWidth, float fSProb, float fPProb)

{

float Low = fSProb;

float High = 1.0f - fPProb;

float fRand;

srand(GetTickCount());

for (int h = 0; h < nHeight; h++)

{

for (int w = 0; w < nWidth; w++)

{

fRand = ((float)rand() / RAND\_MAX);

if (fRand < Low)

{

Out[h][w] = 255;

}

else if (fRand > High)

{

Out[h][w] = 0;

}

else Out[h][w] = In[h][w];

}

}

}

**VIEWERDig.h**

Void inputGaussianNoise(unsigned char \*\* in, unsigned char\*\* out, int nHeight, int nWidth, float FMean, float fStdev);

Void inputSaltPepperNoise(unsigned char\*\* in, nsigned char\*\* out, int nHeight, int nWidth, float fSProb, float,FPProb);

Void GetGaussianPDF(float\* EmptyPDF, int nLength , float fMean, float fStDev);

Float GetNoise(float\* PDF, int nLength);