Parte 1

Desarrollar un programa que permita generar ruido de sal y pimienta y aplicar filtros para reducir dicho ruido. Para ello, deberá llevar a cabo:

1. Programar un método que genere un porcentaje de ruido de sal o pimienta en un video, considerando las dimensiones del mismo. Se deberá poder ingresar un porcentaje de ruido a través de dos trackbars (uno para sal y otro para pimienta).

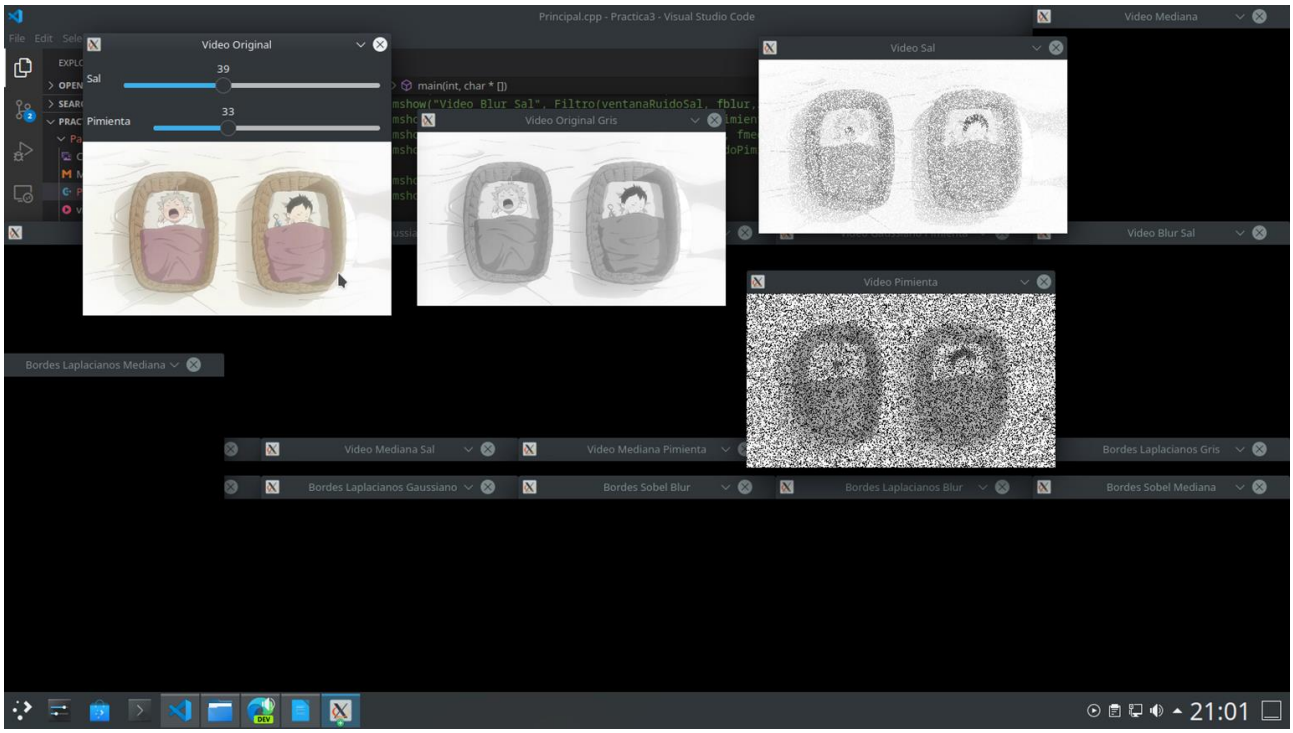
Código:

Principal.cpp

Parte1 > Principal.cpp > SalPimienta(Mat, int, int)

```
17  using namespace std;
18  using namespace cv;
19
20  int sal = 5;
21  int pimienta = 5;
22  int fmediana = 5;
23  int fblur = 5;
24  int fgaussiano = 5;
25
26  void functionTrackbar(int v, void *p){
27  }
28
29  Mat SalPimienta(Mat imagen, int valor, int tipo){
30      Mat res = imagen.clone();
31      double percent=valor/100.0;
32      int total = (int) (percent*res.rows*res.cols);
33      int cont=0, fila=0, columna=0;
34      while(cont < total){
35          res.at<uchar>(rand() % res.rows,rand() % res.cols) = (tipo==1) ? 255:0;
36          cont++;
37      }
38      return res;
39  }
40
```

Resultado



2. Programar una función para aplicar los siguientes filtros: mediana, blur, Gaussiano, probando con diferente tamaño de máscara.

Código:

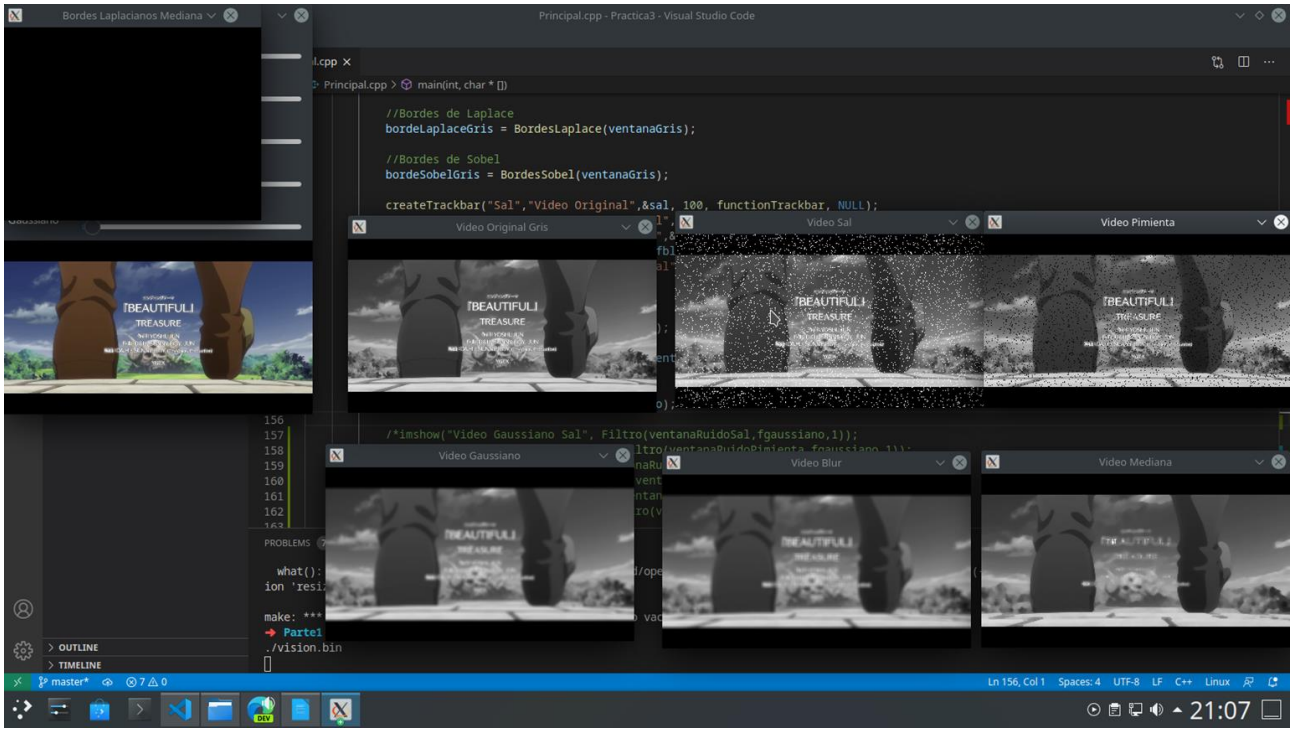
Help

Principal.cpp

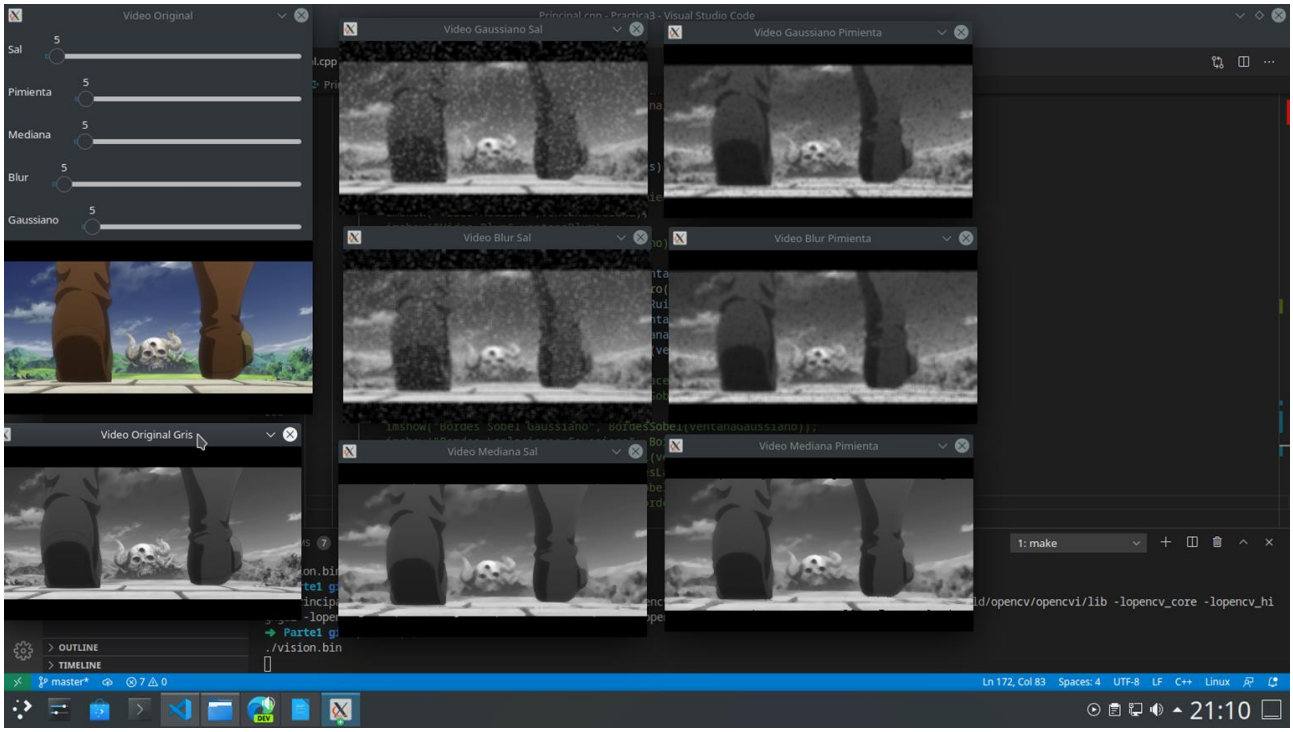
Parte1 > Principal.cpp > BordesSobel(Mat)

```
38      return res;
39  }
40
41  Mat Filtro(Mat imagen, int valor, int tipo){
42      valor = valor % 2 ==0 ? valor - 1 : valor;
43      valor = valor < 1 ? 1 : valor;
44      Mat res;
45      if (tipo==1){
46          GaussianBlur(imagen, res, Size(valor,valor),2,2);
47          return res;
48      }else if (tipo==2){
49          blur(imagen, res, Size(valor,valor));
50          return res;
51      }else if (tipo==3){
52          medianBlur(imagen, res, valor);
53      }
54      return res;
55  }
```

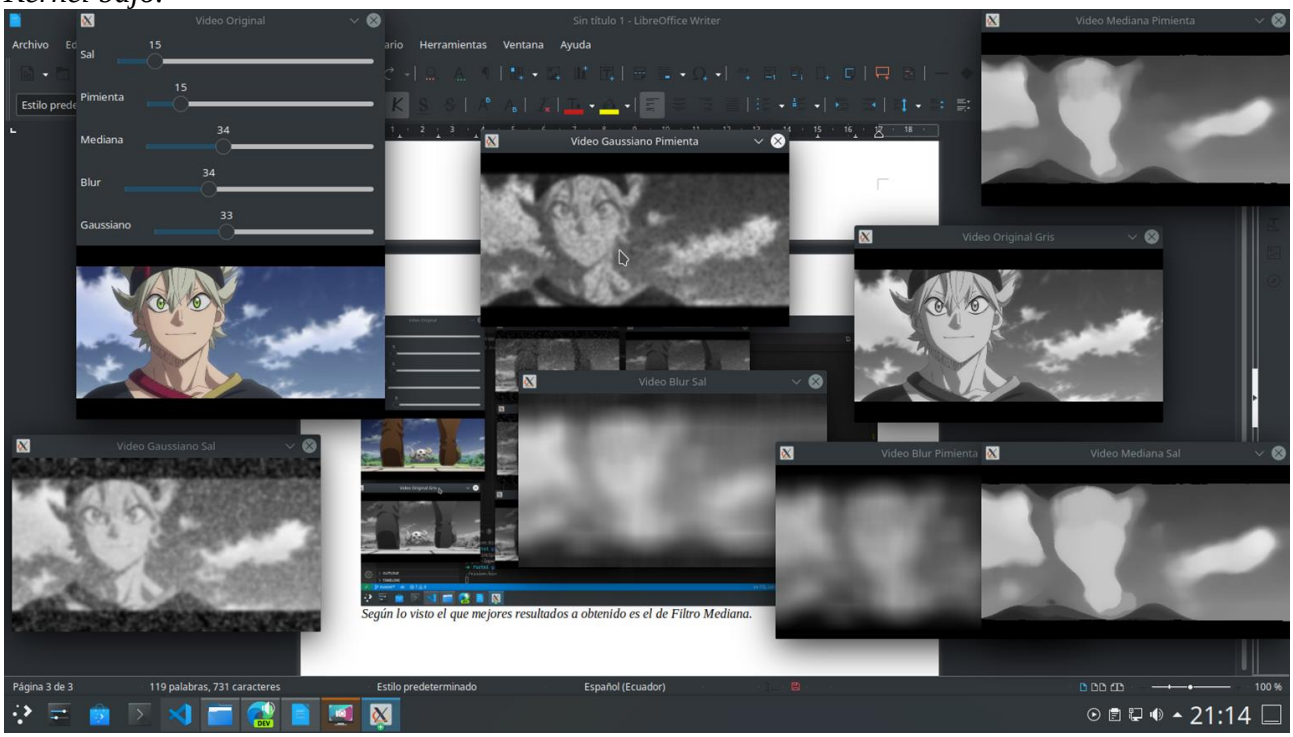
Resultado:



3. Compare los resultados obtenidos por cada filtro, y reflexione cuál ha obtenido mejores resultados.

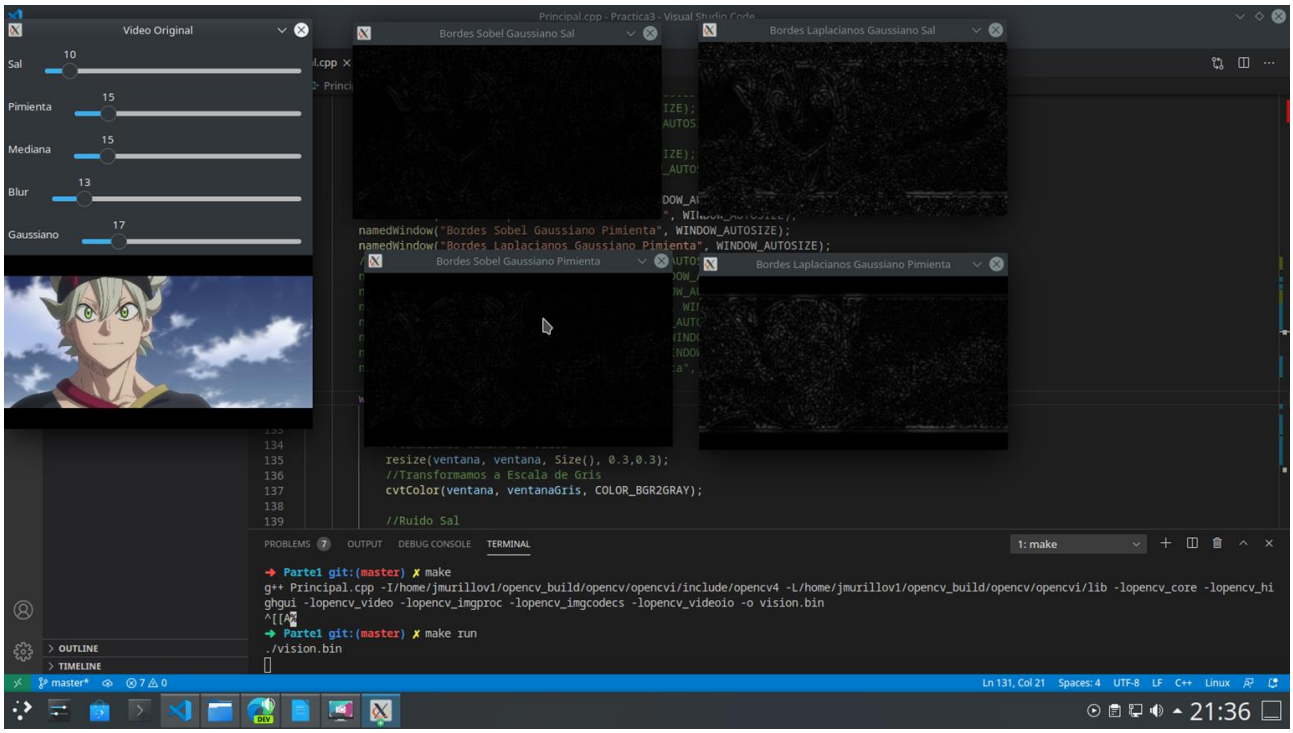


Según lo visto el que mejores resultados a obtenido es el de Filtro Mediana, pero con un valor de Kernel bajo.

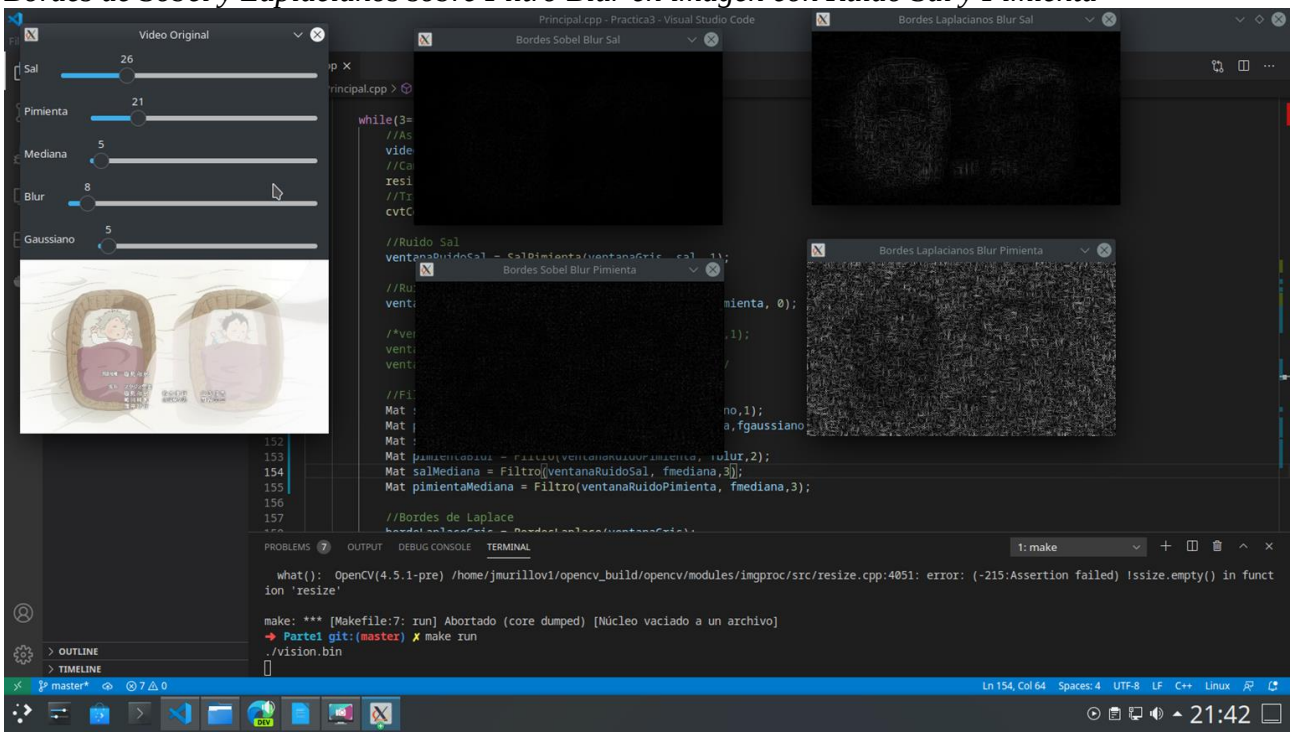


Al aumentar el kernel el que da mejor resultado es el Gaussiano.

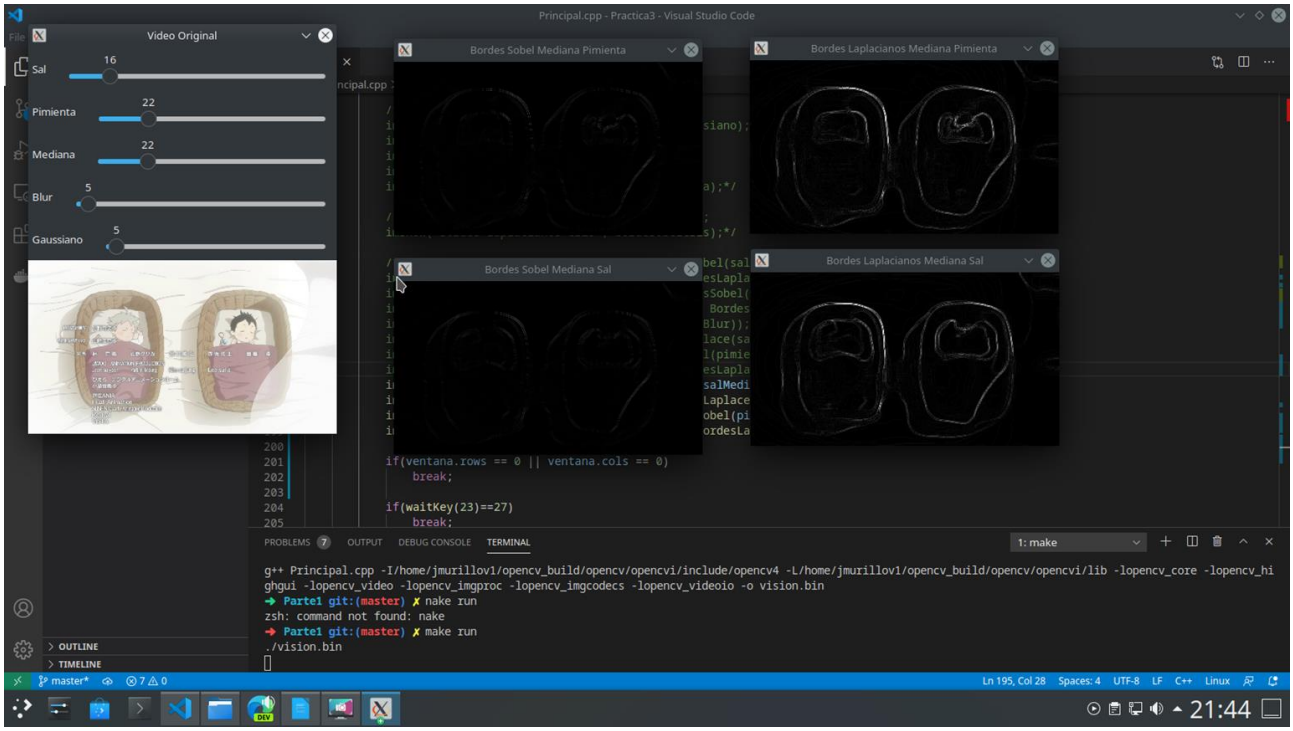
4. Aplicar al menos 2 algoritmos de detección de bordes y comparar los resultados de usar o no filtros de suavizado.
Bordes de Sobel y Laplacianos sobre Filtro Gaussiano en imagen con Ruido Sal y Pimienta.



Bordes de Sobel y Laplacianos sobre Filtro Blur en imagen con Ruido Sal y Pimienta



Bordes de Sobel y Laplacianos sobre Filtro Mediana en imagen con Ruido Sal y Pimienta



Como se puede ver al utilizar filtros podemos reducir el ruido, pero en algunos casos la identificación de bordes se complica.

CODIGO FUENTE:

```
#include <iostream>
#include <cstdlib>
#include <math.h>

#include <opencv2/core/core.hpp>

#include <opencv2/imgproc/imgproc.hpp>

#include <opencv2/imgcodecs/imgcodecs.hpp>

#include <opencv2/highgui/highgui.hpp>

#include <opencv2/video/video.hpp>

#include <opencv2/videoio/videoio.hpp>

using namespace std;
using namespace cv;

int sal = 5;
int pimienta = 5;
int fmediana = 5;
int fblur = 5;
int fgaussiano = 5;

void functionTrackbar(int v, void *p){
}

Mat SalPimienta(Mat imagen, int valor, int tipo){
   Mat res = imagen.clone();
   double percent=valor/100.0;
   int total = (int) (percent*res.rows*res.cols);
   int cont=0, fila=0, columna=0;
   while(cont < total){
      res.at<uchar>(rand() % res.rows,rand() % res.cols) = (tipo==1) ? 255:0;
      cont++;
   }
   return res;
}

Mat Filtro(Mat imagen, int valor, int tipo){
   valor = valor % 2 ==0 ? valor - 1 : valor;
   valor = valor < 1 ? 1 : valor;
   Mat res;
   if (tipo==1){
      GaussianBlur(imagen, res, Size(valor,valor),2,2);
      return res;
   }else if (tipo==2){
      blur(imagen, res, Size(valor,valor));
```

```
        return res;
    }else if (tipo==3){
        medianBlur(imagen, res, valor);
    }
    return res;
}

Mat BordesSobel(Mat imagen){
    Mat gX, gY, gXAbs, gYAbs, bordeSobel, bordeSobelAbs;
    Sobel(imagen, gX, CV_16S, 1, 0, 3);
    Sobel(imagen, gY, CV_16S, 0, 1, 3);
    Sobel(imagen, bordeSobel, CV_16S, 1, 1, 3);
    convertScaleAbs(gX,gXAbs);
    convertScaleAbs(gY,gYAbs);
    convertScaleAbs(bordeSobel,bordeSobelAbs);
    return bordeSobelAbs;
}

Mat BordesLaplace(Mat imagen){
    Mat bordeLaplace, bordeLaplaceAbs;
    Laplacian(imagen, bordeLaplace, CV_16S, 3);
    convertScaleAbs(bordeLaplace, bordeLaplaceAbs);
    return bordeLaplaceAbs;
}

int main (int arcg, char *argv[]){

    VideoCapture video("video.mp4");

    if (video.isOpened()){

        //Declaración de variables a usar
        Mat ventana;
        Mat ventanaGris;
        Mat ventanaRuidoSal;
        Mat ventanaRuidoPimienta;

        Mat salGaussiano;
        Mat pimientaGaussiano;
        Mat salBlur;
        Mat pimientaBlur;
        Mat salMediana;
        Mat pimientaMediana;

        namedWindow("Video Original", WINDOW_AUTOSIZE);
        namedWindow("Video Original Gris", WINDOW_AUTOSIZE);

        namedWindow("Video Gaussiano Sal", WINDOW_AUTOSIZE);
        namedWindow("Video Gaussiano Pimienta", WINDOW_AUTOSIZE);
        namedWindow("Video Blur Sal", WINDOW_AUTOSIZE);
        namedWindow("Video Blur Pimienta", WINDOW_AUTOSIZE);
        namedWindow("Video Mediana Sal", WINDOW_AUTOSIZE);
```

```
namedWindow("Video Mediana Pimienta", WINDOW_AUTOSIZE);

namedWindow("Bordes Sobel Gaussiano Sal", WINDOW_AUTOSIZE);
namedWindow("Bordes Laplacianos Gaussiano Sal", WINDOW_AUTOSIZE);
namedWindow("Bordes Sobel Gaussiano Pimienta", WINDOW_AUTOSIZE);
namedWindow("Bordes Laplacianos Gaussiano Pimienta", WINDOW_AUTOSIZE);
namedWindow("Bordes Sobel Blur Sal", WINDOW_AUTOSIZE);
namedWindow("Bordes Laplacianos Blur Sal", WINDOW_AUTOSIZE);
namedWindow("Bordes Sobel Blur Pimienta", WINDOW_AUTOSIZE);
namedWindow("Bordes Laplacianos Blur Pimienta", WINDOW_AUTOSIZE);
namedWindow("Bordes Sobel Mediana Sal", WINDOW_AUTOSIZE);
namedWindow("Bordes Laplacianos Mediana Sal", WINDOW_AUTOSIZE);
namedWindow("Bordes Sobel Mediana Pimienta", WINDOW_AUTOSIZE);
namedWindow("Bordes Laplacianos Mediana Pimienta", WINDOW_AUTOSIZE);

while(3==3){
   //Asignamos el video a una ventana
   video >> ventana;
   //Cambiamos tamaño de video
   resize(ventana, ventana, Size(), 0.3,0.3);
   //Transformamos a Escala de Gris
   cvtColor(ventana, ventanaGris, COLOR_BGR2GRAY);

   //Ruido Sal
   ventanaRuidoSal = SalPimienta(ventanaGris, sal, 1);

   //Ruido Pimienta
   ventanaRuidoPimienta = SalPimienta(ventanaGris, pimienta, 0);

   //Filtros SAL y PIMIENTA
   Mat salGaussiano = Filtro(ventanaRuidoSal,fgaussiano,1);
   Mat pimientaGaussiano = Filtro(ventanaRuidoPimienta,fgaussiano,1);
   Mat salBlur = Filtro(ventanaRuidoSal, fblur,2);
   Mat pimientaBlur = Filtro(ventanaRuidoPimienta, fblur,2);
   Mat salMediana = Filtro(ventanaRuidoSal, fmediana,3);
   Mat pimientaMediana = Filtro(ventanaRuidoPimienta, fmediana,3);

   // Creaci´on de los Trackbar Necesarios
   createTrackbar("Sal","Video Original",&sal, 100, functionTrackbar, NULL);
   createTrackbar("Pimienta","Video Original",&pimienta, 100, functionTrackbar, NULL);
   createTrackbar("Mediana","Video Original",&fmediana, 100, functionTrackbar, NULL);
   createTrackbar("Blur","Video Original",&fblur, 100, functionTrackbar, NULL);
   createTrackbar("Gaussiano","Video Original",&fgaussiano, 100, functionTrackbar, NULL);

   //Mostramos los resultados
   imshow("Video Original",ventana);
   imshow("Video Original Gris",ventanaGris);

   //Mostramos los filtros
   imshow("Video Gaussiano Sal", salGaussiano);
   imshow("Video Gaussiano Pimienta", pimientaGaussiano);
   imshow("Video Blur Sal", salBlur);
```

```
        imshow("Video Blur Pimienta", pimientaBlur);
        imshow("Video Mediana Sal", salMediana);
        imshow("Video Mediana Pimienta", pimientaMediana);

        // Mostramos los Bordes
        imshow("Bordes Sobel Gaussiano Sal", BordesSobel(salGaussiano));
        imshow("Bordes Laplacianos Gaussiano Sal", BordesLaplace(salGaussiano));
        imshow("Bordes Sobel Gaussiano Pimienta", BordesSobel(pimientaGaussiano));
        imshow("Bordes Laplacianos Gaussiano Pimienta", BordesLaplace(pimientaGaussiano));
        imshow("Bordes Sobel Blur Sal", BordesSobel(salBlur));
        imshow("Bordes Laplacianos Blur Sal", BordesLaplace(salBlur));
        imshow("Bordes Sobel Blur Pimienta", BordesSobel(pimientaBlur));
        imshow("Bordes Laplacianos Blur Pimienta", BordesLaplace(pimientaBlur));
        imshow("Bordes Sobel Mediana Sal", BordesSobel(salMediana));
        imshow("Bordes Laplacianos Mediana Sal", BordesLaplace(salMediana));
        imshow("Bordes Sobel Mediana Pimienta", BordesSobel(pimientaMediana));
        imshow("Bordes Laplacianos Mediana Pimienta", BordesLaplace(pimientaMediana));

        if(ventana.rows == 0 || ventana.cols == 0)
            break;

        if(waitKey(23)==27)
            break;
      }
      destroyAllWindows();
   }
   return 0;
}
```