Manual de Usuario

Roberto Solares

06-07-2020

Contents

I. Antes de empezar

Antes de empezar, verifica que tienes instaladas las siguientes dependencias:

- Git: `https://git-scm.com/downloads`
- .NET Core 3.0 `https://dotnet.microsoft.com/download`

Y que tienes una copia del código fuente del proyecto. Si no lo tienes puedes descargarlo corriendo el siguiente comando en tu consola:

```
git clone https://github.com/betoSolares/filters.git
```

Ten en cuenta que si el proyecto lo intentas correr en alguna carpeta que tenga las siguientes direcciones dará error:

- src/bin/Debug/netcoreapp3.0
- src/bin/Release/netcoreapp3.0

II. Ejecución

Hay tres maneras en las que puedes ejecutar el proyecto, estas son:

1. Linea de comandos (recomendada).
2. Visual Studio Code.
3. Visual Studio IDE.

Todas estas tienen en común las mismas dependencias, las cuales son necesarias para que el proyecto se ejecute de manera correcta.

II.I Dependencias

- AvaloniaUI `https://www.nuget.org/packages/Avalonia`.
- Citrus.Avalonia `https://www.nuget.org/packages/Citrus.Avalonia`.
- ReactiveUI `https://www.nuget.org/packages/reactiveui`.
- .NET Core standard library, esta viene con la instalación del compilador.

II.II Compilación

Esta parte solo es valida si tratas de ejecutar el proyecto mediante la linea de comandos. Si es así en el directorio raíz del proyecto ejecuta el siguiente comando:

```
dotnet build
```

Verifica que el mensaje emitido diga que la compilación fue exitosa y que hay 0 advertencias y 0 errores.

Si al momento de ejecutarlo encuentras un error trata de ejecutar el siguiente comando

```
dotnet restore && dotnet build
```

II.III Ejecución

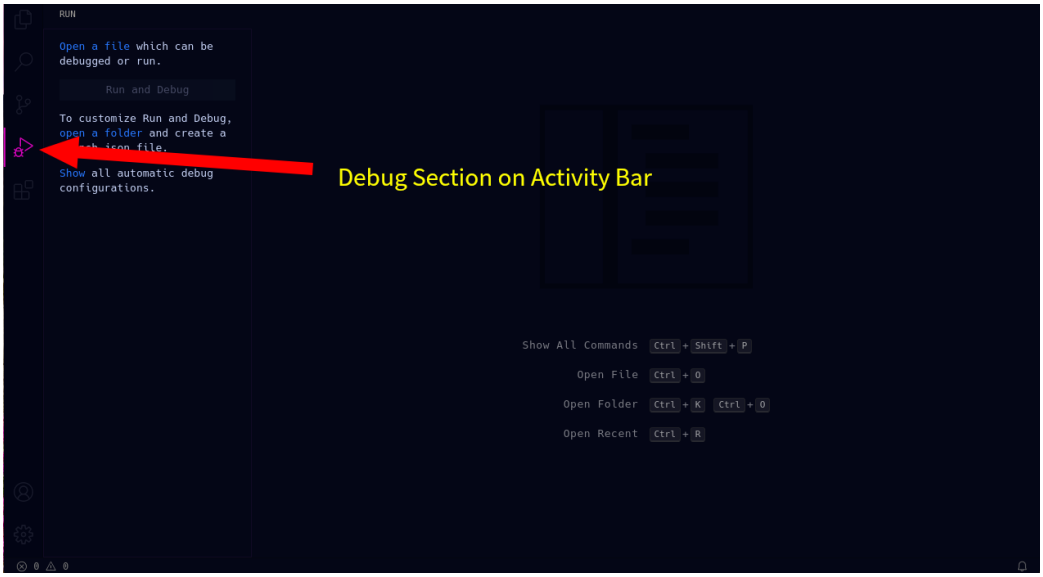
Linea de comandos

Para ejecutar el proyecto primero lo debes de haber compilado, después de haber hecho eso debes de ejecutar el siguiente comando en el directorio raíz del proyecto:

```
dotnet run -p src
```

Visual Studio Code

Para esto debes dirigirte a la sección de Ejecución o Debug en la barra de actividades.

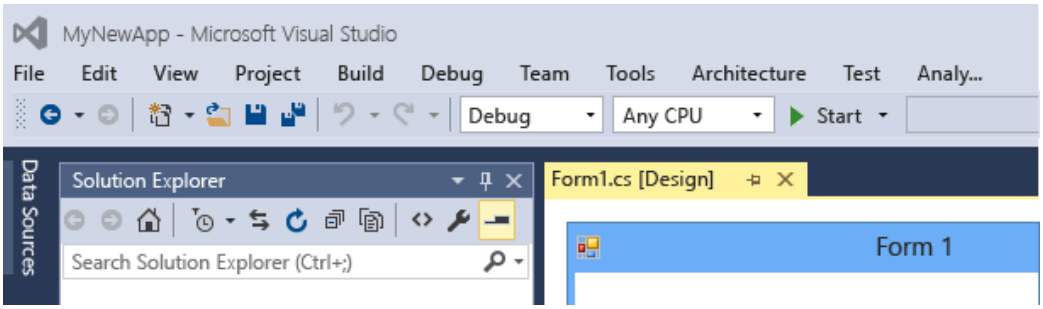


Luego debes hacer click al botón Run para inicializar el proyecto.

Ten en cuenta que puede ser que en algunas ocasiones y dependiendo de tu sistema operativo este proceso se pueda demorar demasiado tiempo. Es por ello que se recomienda utilizar una terminal integrada de Visual Studio Code y seguir el proceso para la línea de comandos.

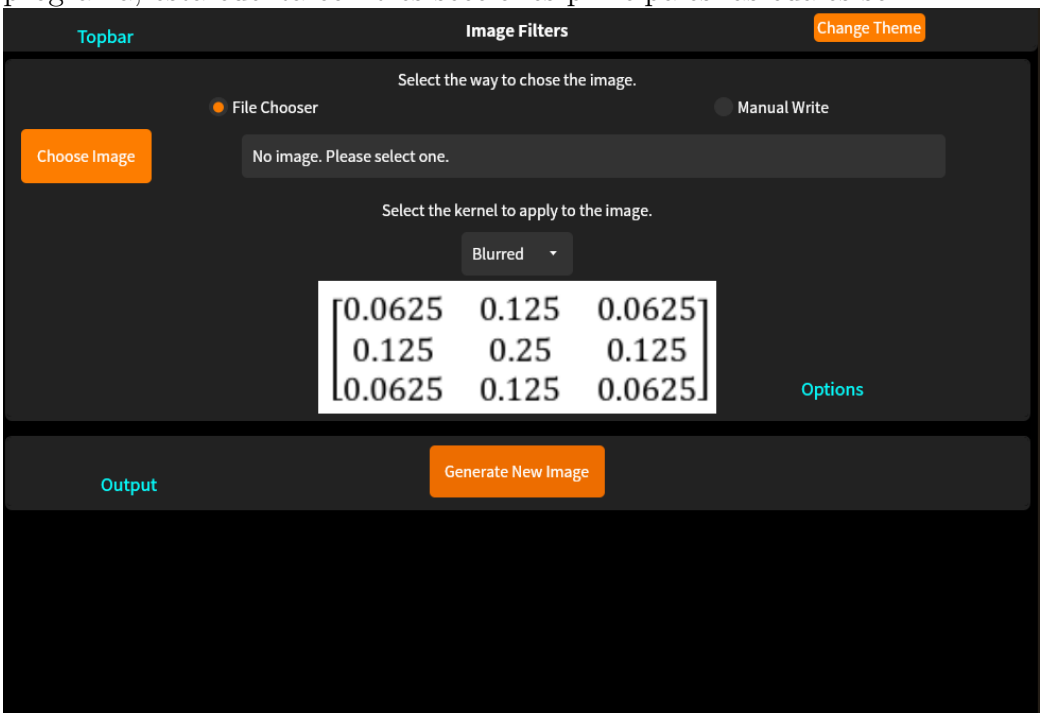
Visual Studio IDE

Para esta opción solo la puedes ejecutar en Windows, ya que necesitas Visual Studio IDE y este solo se puede ejecutar en ese sistema operativo. Aquí solo debes de darle click al botón de Start.



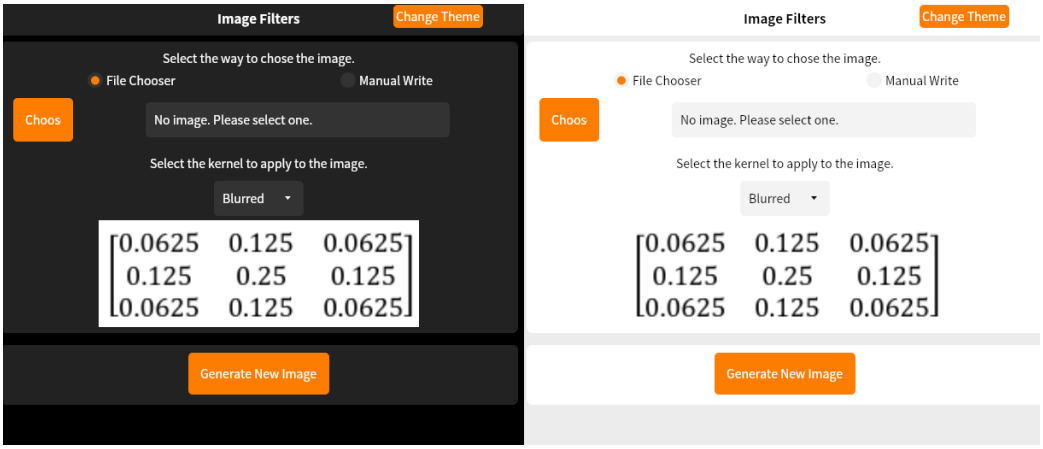
III. Explicación del proyecto

Al momento de ejecutar el proyecto se te presentara la ventana principal del programa, esta cuenta con tres secciones principales las cuales son:



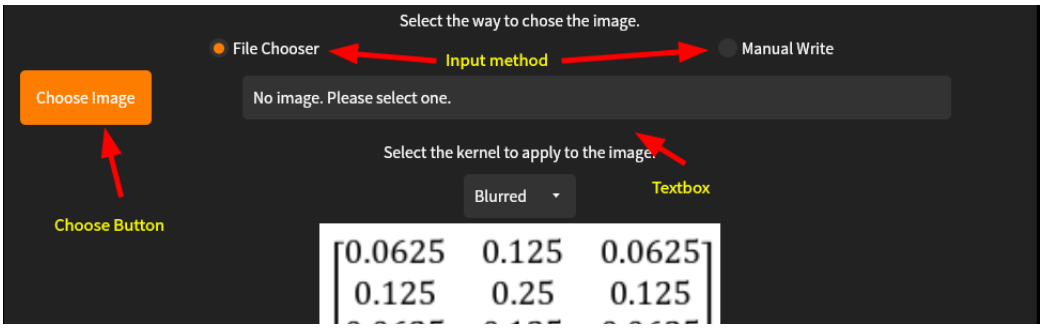
III.I Topbar

Aquí se encuentra el titulo de la aplicación y un botón con el que se puede cambiar el tema de la aplicación.

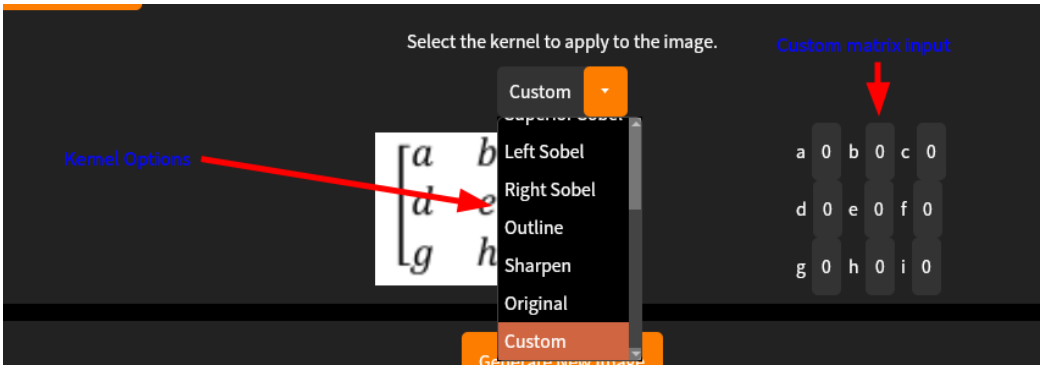


III.II Options

Aquí puedes seleccionar la imagen a la que deseas aplicarle el filtro, para ello solo debes de hacer click en el botón Choose, es posible que si estas ejecutando la aplicación en Linux o macOS esta funcionalidad haga a la aplicación fallar, para ello solo debes de seleccionar el método de entrada Manual Write y después escribir la dirección a la imagen de manera manual en el textbox.



Después tienes la opción de seleccionar varios filtros que se le pueden ser aplicados a la imagen, si seleccionas la opción Custom te aparecerá una tabla en la que podrás ingresar los valores para cada una de las posiciones que se te pide, ten en cuenta que en esta solo podrás ingresar números y cualquier letras que ingreses sera descartada.



III.III Output

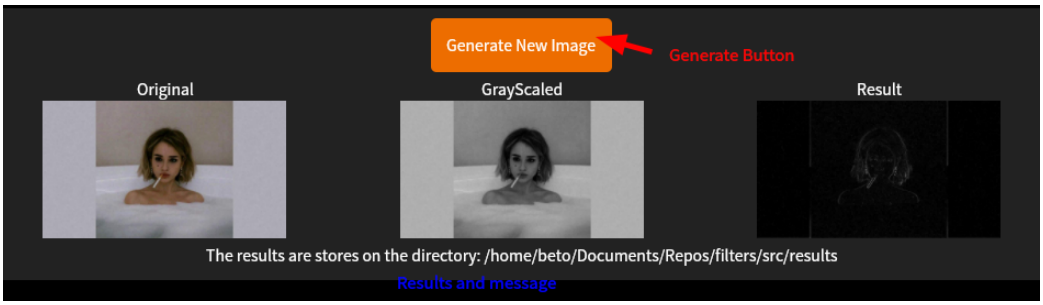
Después de que has seleccionado la imagen y el kernel que se le aplicara, debes de darle click al botón de Generate New Image, el cual aplicara el algoritmo de filtrado a la imagen. Las matrices que se utilizan para cada filtro son las siguientes:

Difuminado $\begin{bmatrix} 0.0625 & 0.125 & 0.0625 \\ 0.125 & 0.25 & 0.125 \\ 0.0625 & 0.125 & 0.0625 \end{bmatrix}$	**Realzar** $\begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$
Sobel Inferior $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	**Sobel Superior** $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
Sobel Izquierdo $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	**Sobel Derecho** $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
Contorno $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	**Afilar** $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$
Original $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	**Personalizado** $\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$

Con la única diferencia de que para el filtro Blur se utiliza la matriz no

normalizada, ya que para esta es mucho mas fácil aplicar el algoritmo al no tener que balancear los pesos de la matriz. Para resolver el problema de los bordes en cada filtro se utilizo el algoritmo de Extensión, el cual utiliza los pixeles mas cercanos y los extiende tanto como sea necesario. Los pixeles se extienden en lineas rectas.

Luego de que se ha generado la imagen veras los resultados de todo el proceso así como un mensaje indicando en donde han sido guardadas las imágenes.



References

[1] Prakhar Ganesh. *Types of Convolution Kernels: Simplified*. URL: https://towardsdatascience.com/types-of-convolution-kernels-simplified-f040cb307c37.

[2] GIMP. *Convolution Matrix*. URL: https://docs.gimp.org/2.8/en/plug-in-convmatrix.html.

[3] Jamie Ludwing. *Image Convolution*. URL: qute://pdfjs/web/viewer.html?filename=tmpyq1ds5la_Ludwig_ImageConvolution.pdf&file=&source=http://web.pdx.edu/~jduh/courses/Archive/geog481w07/Students/Ludwig_ImageConvolution.pdf.

[4] Victor Powell. *Image Kernels*. URL: https://setosa.io/ev/image-kernels/.

[5] A blog about shaders. *Computer Graphics*. URL: http://ablogaboutshaders.blogspot.com/2013/04/2d-gaussian-blur.html.

[6] Wikipedia. *Kernel (image processing)*. URL: https://en.wikipedia.org/wiki/Kernel_(image_processing).

[7] Tianxu Zhang. *Guidance Information Processing Methods in Airbone Optical Imaging Seeker*. Springer Nature, 2019.