

PROYECTO FIN DE GRADO

TÍTULO: Diseño y desarrollo de un sistema de conversión de imagen real a dibujo mediante técnicas de procesamiento de imagen.

AUTOR/A: Gema Serrano Domínguez

TITULACIÓN: Ingeniería de Sonido e Imagen

TUTOR/A: César Díaz Martín

DEPARTAMENTO: Ingeniería Audiovisual y Comunicaciones (IAC)

VºBº TUTOR/A

Miembros del Tribunal Calificador:

PRESIDENTE/A: Manuel Vázquez Rodríguez

TUTOR/A: César Díaz Martín

SECRETARIO/A: Martina Eckert

Fecha de lectura: 30/09/2024

Calificación:

El Secretario/La Secretaria,

Resumen

Este proyecto, titulado “Diseño y desarrollo de un sistema de conversión de imagen real a dibujo mediante técnicas de procesado de imagen”, trata sobre la conversión de imágenes reales a dibujos utilizando técnicas de procesamiento de imágenes mediante el lenguaje de programación Python.

Para ello se ha llevado a cabo el desarrollo de un programa que convierte una imagen a distintos estilos de dibujos entre varios efectos disponibles generando así una interfaz muy visual con la que poder trabajar.

Este sistema se caracteriza sobre todo por ser flexible y modulable en sus funcionalidades y factible para el manejo por parte del usuario. Se ha creado de forma que no cubra unas necesidades exactas, sino que aporte un abanico de opciones para que el usuario pueda probar y elegir entre todas ellas hasta conseguir el resultado que desea. Además, para conseguir que el programa sea aún más fácil de utilizar, se ha utilizado una interfaz gráfica que permite que todas estas opciones y pruebas se puedan realizar de forma intuitiva sin necesidad de estudios o conocimientos previos sobre la programación ni sobre el programa en cuestión.

Las diferentes pruebas y opciones que da el programa consisten en acciones como cargar la imagen que se quiere convertir, diferentes efectos a los que poder convertirla, una acción para retornar a la imagen original siempre que se quiera y por supuesto una acción que permita guardar los resultados. La opción más amplia y en la que realmente se lleva a cabo el proceso de conversión en función de lo que se desee obtener de la imagen es en los efectos disponibles.

Los efectos disponibles son cartoon, pixelado, grafiti, cómic, carboncillo, tintado, acuarela, sombreado y dithering. Para conseguir que el programa sea más flexible aún, cada uno de estos efectos cuenta con la opción de modificar uno o varios parámetros, consiguiendo así dentro de cada estilo una gama de tonalidades y formas que proporciona muchas posibilidades.

El proceso para conseguir este tipo de sistema se ha conseguido mediante el uso de técnicas como la detección de bordes de la imagen, aumento o disminución de la paleta de colores y la combinación de ambas.

Otro recurso importante que ha consistido en una de las bases para realizar este sistema ha sido el uso de bibliotecas. Han permitido crear mayor número de efectos y que el programa fuera consistente y funcional, ya que proporcionan multitud de funciones del campo del procesamiento de la imagen.

En conclusión, con este proyecto se han logrado los objetivos buscados, consiguiendo un programa funcional y útil para la conversión de imágenes a dibujos, además de estar constituido por una interfaz intuitiva.

También se ha comprobado que el funcionamiento del sistema cubre todas las necesidades deseadas para el usuario y que resulta sencillo utilizarlo mediante las pruebas llevadas a cabo. Debido a estos motivos se ha conseguido una solución plena que cumple las expectativas previas al comienzo del proyecto.

Abstract

This project, entitled "Design and development of a system for converting real images to drawings using image processing techniques", deals with the conversion of real images to drawings using image processing techniques using the Python programming language.

To this end, the development of a program has been carried out that converts an image to different styles of drawings among various available effects, thus generating a very visual interface with which to work.

This system is characterized above all by being flexible and modular in its functionalities and feasible for use by the user. It has been created in such a way that it does not cover exact needs, but rather provides a range of options so that the user can try and choose between all of them until they achieve the result they want. In addition, to make the program even easier to use, a graphical interface has been used that allows all these options and tests to be carried out intuitively without the need for previous studies or knowledge about programming or the program in question.

The different tests and options that the program gives consist of actions such as loading the image you want to convert, different effects to which you can convert it, an action to return to the original image whenever you want and of course an action that allows you to save the results. The widest option and the one in which the conversion process is really carried out depending on what you want to get from the image is in the available effects.

The effects available are cartoon, pixelated, graffiti, comic, charcoal, tinted, watercolor, shading and dithering. To make the program even more flexible, each of these effects has the option of modifying one or more parameters, thus achieving within each style a range of shades and shapes that provide many possibilities.

The process to achieve this type of system has been achieved by using techniques such as detecting the edges of the image, increasing or decreasing the color palette and combining both.

Another important resource that has been one of the bases for this system has been the use of libraries. They have made it possible to create a greater number of effects and for the program to be consistent and functional, as they provide a multitude of functions in the field of image processing.

In conclusion, with this project the objectives sought have been achieved, achieving a functional and useful program for the conversion of images to drawings, in addition to being constituted by an intuitive interface.

It has also been proven that the operation of the system covers all the desired needs of the user and that it is easy to use through the tests carried out. Due to these reasons, a full solution has been achieved that meets the expectations prior to the start of the project.

Índice de figuras

<i>Figura 1. Diagrama de bloques de la arquitectura del sistema</i>	<i>19</i>
<i>Figura 2. Proceso algoritmo K-Means</i>	<i>22</i>
<i>Figura 3.Imagen rana original</i>	<i>30</i>
<i>Figura 4.Imagen niña original</i>	<i>30</i>
<i>Figura 5. Imagen pingüinos original</i>	<i>30</i>
<i>Figura 6.Imagen rana con efecto cartoon</i>	<i>31</i>
<i>Figura 7.Imagen niña con efecto cartoon</i>	<i>31</i>
<i>Figura 8.Imagen pingüinos con efecto cartoon</i>	<i>31</i>
<i>Figura 9.Imagen rana con efecto pixelado</i>	<i>32</i>
<i>Figura 10.Imagen niña con efecto pixelado</i>	<i>32</i>
<i>Figura 11.Imagen pingüinos con efecto pixelado</i>	<i>32</i>
<i>Figura 12.Imagen rana con efecto grafiti</i>	<i>33</i>
<i>Figura 13.Imagen niña con efecto grafiti</i>	<i>33</i>
<i>Figura 14.Imagen pingüinos con efecto grafiti</i>	<i>33</i>
<i>Figura 15.Imagen rana con efecto cómic</i>	<i>34</i>
<i>Figura 16.Imagen niña con efecto cómic</i>	<i>34</i>
<i>Figura 17.Imagen pingüinos con efecto cómic</i>	<i>34</i>
<i>Figura 18.Imagen rana con efecto carboncillo</i>	<i>35</i>
<i>Figura 19. Imagen niña con efecto carboncillo</i>	<i>35</i>
<i>Figura 20.Imagen pingüinos con efecto carboncillo</i>	<i>35</i>
<i>Figura 21.Imagen rana con efecto tintado</i>	<i>36</i>
<i>Figura 22.Imagen niña con efecto tintado</i>	<i>36</i>
<i>Figura 23.Imagen pingüinos con efecto tintado</i>	<i>36</i>
<i>Figura 24.Imagen rana con efecto acuarela</i>	<i>37</i>
<i>Figura 25.Imagen niña con efecto acuarela</i>	<i>37</i>
<i>Figura 26.Imagen pingüinos con efecto acuarela</i>	<i>37</i>
<i>Figura 27.Imagen rana con efecto sombreado</i>	<i>38</i>
<i>Figura 28.Imagen niña con efecto sombreado</i>	<i>38</i>
<i>Figura 29.Imagen pingüinos con efecto sombreado</i>	<i>38</i>
<i>Figura 30.Imagen rana con efecto dithering</i>	<i>39</i>
<i>Figura 31.Imagen niña con efecto dithering</i>	<i>39</i>
<i>Figura 32.Imagen pingüinos con efecto dithering</i>	<i>39</i>
<i>Figura 33.Página web para descargar Python</i>	<i>40</i>
<i>Figura 34.Archivos obtenidos al descargar Python</i>	<i>40</i>
<i>Figura 35. Primera ventana al iniciar el entorno Python</i>	<i>41</i>
<i>Figura 36.Segunda ventana del entorno Python tras cargar código</i>	<i>41</i>
<i>Figura 37.Menú principal programa</i>	<i>42</i>
<i>Figura 38.Ventana emergente para seleccionar imagen inicial</i>	<i>42</i>
<i>Figura 39.Menú principal programa tras cargar una imagen</i>	<i>43</i>
<i>Figura 40.Ventana emergente para seleccionar los parámetros</i>	<i>43</i>

Índice de tablas

<i>Tabla 1. Gastos proyecto.....</i>	<i>46</i>
--------------------------------------	-----------

Lista de acrónimos

- **GIF:** Graphics Interchange Format (Estilo de formato de una imagen)
- **GUI:** Graphical User Interface (Interfaz gráfica de usuario)
- **JPEG:** Joint Photographic Experts Group (Estilo de formato de una imagen)
- **Numpy:** Numerical Python (biblioteca para cálculos matemáticos y científicos en Python)
- **ODS:** Objetivos de Desarrollo Sostenible
- **OpenCV:** Open Source Computer Vision Library (Biblioteca para el procesamiento de imagen en Python)
- **PIL:** Python Imaging Library (Biblioteca para el procesamiento de imagen en Python)
- **PNG:** Portable Network Graphics (Estilo de formato de una imagen)

Índice de contenidos

Resumen	i
Abstract	iii
Índice de figuras	v
Índice de tablas	vi
Lista de acrónimos.....	vi
1. Introducción	9
1.1 Marco y motivación del proyecto.....	9
1.2 Objetivos técnicos y académicos	9
1.3 Estructura del resto de la memoria	10
2. Marco tecnológico	13
2.1 Tecnología A: Procesamiento de imagen	13
2.1.1 Característica 1: Técnicas de Filtrado y Mejora de Imágenes.....	13
2.1.2 Característica 2: Detección de Bordes y Contornos	14
2.2 Tecnología B: Lenguaje de Programación Python	15
2.2.1 Bibliotecas Especializadas	15
2.2.2 Interfaz de Usuario	16
3. Especificaciones y restricciones de diseño	17
3.1 Especificaciones	17
3.2 Restricciones	18
4. Descripción de la solución propuesta	19
4.1 Técnicas empleadas	20
4.1.1 Cargar imagen, guardar imagen y opción original.....	24
4.1.2 Efectos disponibles	25
4.1.3 Resultados.....	30
4.2 Interfaz gráfica de usuario	40
5. Presupuesto	45
6. Impacto del proyecto.....	47
7. Conclusiones	49
8. Referencias	51
9. Agradecimientos	53

1. Introducción

1.1 Marco y motivación del proyecto

En el actual panorama tecnológico, el procesamiento de imágenes ha experimentado un crecimiento exponencial debido al incremento progresivo de su utilidad, proporcionando una base sólida para la creación de aplicaciones innovadoras incluyendo el desarrollo de herramientas que permiten automatizar el proceso. Dentro del mundo del procesado de imágenes, una tarea de interés es la conversión de imágenes reales a dibujos, que tiene varias aplicaciones como son el diseño gráfico, la publicidad e incluso para un fin puramente estético entre otras.

Este proyecto se enmarca en este contexto y tiene como propósito principal desarrollar un sistema de conversión de imágenes reales a dibujos utilizando técnicas de procesamiento de imagen con el lenguaje de programación Python.

El propósito de este proyecto es explorar y aprovechar las capacidades de la visión por computadora para permitir a todos los usuarios transformar sus imágenes en ilustraciones, ofreciendo una herramienta creativa que puede encontrar aplicaciones en campos como el entretenimiento, la publicidad y la educación.

1.2 Objetivos técnicos y académicos

Entre los objetivos técnicos de este proyecto se encuentran:

- Llevar a cabo un programa que mediante técnicas de procesado de imagen realice una conversión de imagen a dibujo.
- Investigar en bibliotecas de Python para el uso de sus algoritmos y conseguir visualizar los resultados en pantalla, filtrar y detectar los bordes de las imágenes para operar con ellas, reducir la paleta de colores e incluso cargar y guardar las imágenes iniciales y los resultados.
- Generar un menú de programa con un aspecto sencillo para el manejo del usuario, pudiendo utilizarlo y probar tantas veces como necesite por medio de una interfaz gráfica de usuario (GUI).
- Realizar una serie de pruebas para corregir posibles errores e incluso mejorar el desarrollo del programa todo lo necesario.

Entre los objetivos académicos que se han conseguido adquirir de este proyecto se encuentran:

- Conseguir dominar técnicas avanzadas de procesamiento de imágenes y visualización por pantalla.

- Adquirir habilidad para programar en Python y utilizar bibliotecas especializadas como OpenCV (Open Source Computer Vision Library), PIL (Python Imaging Library) y Numpy (Numerical Python).
- Alcanzar la capacidad para diseñar, implementar y evaluar algoritmos complejos de forma efectiva y útil para este proyecto.
- Adquirir las competencias necesarias durante el desarrollo del programa y la interfaz gráfica de usuario para que tenga un manejo sencillo y práctico.
- Conseguir experiencia al realizar pruebas y optimizar los sistemas, incluyendo la capacidad de analizar los resultados y realizar todos los ajustes necesarios.
- Lograr mejorar la capacidad para gestionar proyectos, incluyendo la planificación, ejecución y presentación de resultados de manera profesional.

1.3 Estructura del resto de la memoria

La estructura del resto de la memoria se basa en los puntos clave que se han ido siguiendo para la creación de este proyecto en su debido orden.

El primer paso para ello, y por tanto el primer punto de esta estructura, se trata de ubicar el marco tecnológico de este proyecto, lo que consta de una visión general del campo de conocimiento en que se encuentra el problema abordado.

Una vez se tiene un enfoque general, se comienzan a dar detalles sobre las partes que se encuentran en relación directa con el trabajo, como son las especificaciones y restricciones de diseño. En esta parte se muestran las primeras ideas y pruebas para crear el programa que dé solución a este proyecto, así como especificando de forma clara aquellos puntos de diseño que hayan sido clave para todo ello.

Tras analizar las especificaciones y restricciones que el programa va a tener, se comienza a llevar a cabo la solución al completo, lo cual se refleja en la descripción de la solución propuesta. En este apartado se explica el proceso de creación de código de forma individual para cada una de las funciones creadas en el programa, así como los parámetros a tener en cuenta y el resultado esperado de cada una, y el desarrollo de la interfaz gráfica de usuario que envuelve todo haciendo que el programa sea fácil de usar.

Este último apartado da paso al siguiente en el que se muestran todos los resultados de cada parte del programa. Con ello se consigue entender de forma visual todo el proceso requerido y solución obtenida.

Una vez se ha expuesto todo el proceso de creación, se analiza todo el presupuesto gastado en ello, desde los gastos que generan el tiempo y trabajo de la persona que lo lleva a cabo hasta los equipos utilizados para ello. Además, de la mano de este apartado, se ponen en manifiesto las implicaciones sociales, de salud y seguridad, ambientales, económicas tecnológicas o industriales mostrando así el impacto del proyecto.

Al finalizar, se realiza una conclusión sobre el proyecto que incluya cada uno de estos pasos de forma resumida, incluyendo una resolución de resultados relacionando lo buscado y lo obtenido y añadiendo si el proyecto ha quedado cerrado o por lo contrario han quedado asuntos sin resolverse.

Por último, se añaden apartados de referencias, sobre cada información utilizada tanto para la redacción como para el aprendizaje, y anexos con un manual de usuario sobre el funcionamiento del programa como información adicional para completar la comprensión y facilitar el uso del programa.

2. Marco tecnológico

2.1 Tecnología A: Procesamiento de imagen

La tecnología principal que se va a utilizar es el procesamiento de imagen, lo cual engloba una amplia variedad de técnicas con las que perfeccionar, analizar y trabajar las imágenes. Esta tecnología ha tenido un crecimiento significativo debido al aumento de su utilidad y necesidad en diversas áreas como la seguridad, la publicidad y la medicina, aunque este proyecto se encuentra orientado al área del entretenimiento mediante la conversión de imágenes reales a dibujos.

El procesamiento de imágenes contiene numerosas técnicas cuyo fin es convertir imágenes con distintos motivos como eliminar el ruido, resaltar contrastes o detalles y mejorarlos e incluso prepararlas para modificaciones más avanzadas. Las funciones más importantes y las usadas para este proyecto se explican a continuación [1] [2].

2.1.1 Característica 1: Técnicas de Filtrado y Mejora de Imágenes

Dentro del procesamiento de imagen existen unas características principales como son las técnicas de filtrado y la mejora de imágenes. Estas técnicas incluyen la aplicación de filtros que se utilizan para reducir el ruido y mejorar la calidad de la imagen. Además, se aplican métodos de realce de contraste, ajuste de brillo para optimizar la visibilidad de los detalles en la imagen original, y una reducción de la paleta de colores para caracterizar mejor el dibujo, cumpliendo así algunas de las características de un dibujo.

Existen diferentes tipos de filtros, cada uno con unas funciones distintas:

- Filtro de Media (Promedio): Su función principal es suavizar la imagen, lo cual hace que disminuyan los cambios drásticos de color o la intensidad de la imagen. Para ello toma el valor promedio de los píxeles vecinos y asigna ese promedio al píxel central. Aunque como consecuencia pueda haber pérdida de detalles, consigue eliminar el ruido de la imagen.
 - o Ejemplo: Si se aplica este filtro en una imagen, los detalles como bordes o texturas leves pueden despreciarse, pero se consigue una imagen más suave y con menos ruido.
- Filtro Gaussiano: Se parece al filtro de media mencionado antes en cuanto a su uso para suavizar una imagen, pero lo lleva a cabo con más detalle. En este caso, se realiza una distribución gaussiana o campana de Gauss de forma que aquellos píxeles que estén más cerca del píxel vecino tendrán mayor valor que los alejados.
 - o Ejemplo: Si se aplica este filtro en las mismas condiciones que el ejemplo anterior, los detalles estarían más puestos en valor aún eliminando el ruido de la imagen.

- Filtro de Mediana: Este filtro es más utilizado para eliminar el ruido de la imagen provocado por intensidades muy altas o bajas en una imagen. En este caso, al aplicar el filtro se reemplaza cada píxel con el valor mediano de los píxeles vecinos en lugar del promedio. De todos los filtros es el que mejor consigue mantener los bordes de la imagen.
 - Ejemplo: En imágenes con ruido aleatorio, producido por la intensidad como se ha mencionado, consigue eliminar ese ruido puntual manteniendo los detalles.

Además de los filtros descritos, en el programa se utilizarán técnicas de mejora de imágenes como:

- Realce de contraste: Se trata de llevar a cabo una expansión de niveles de intensidad para que se distinga mejor las diferencias entre las zonas claras y oscuras de una imagen.
- Ajuste de brillo: Esta técnica cambia los valores del brillo total de la imagen para conseguir así mayor percepción de los detalles. De esta forma al aumentar ese brillo, se pueden resaltar las zonas oscuras e impedir la saturación en las claras.
- Reducción de la paleta de colores: Consiste en hacer más simple la imagen disminuyendo con ello el abanico de colores de la imagen inicial. Puede ser clave para la conversión de imagen en determinados estilos que requieran de mayor sencillez destacando más así las texturas y contornos de la imagen.

2.1.2 Característica 2: Detección de Bordes y Contornos

Existen más características que son fundamentales como la detección de bordes, que representan los cambios repentinos de intensidad en una imagen, y contornos para la conversión de imagen a dibujo. Consisten en generar de forma destacada los límites de la imagen para posteriormente combinarlos, así permiten extraer las líneas y contornos principales y con ello poder caracterizar un dibujo detallado y preciso en función del resultado deseado. Es otra de las características principales de un dibujo.

La detección de bordes se basa en analizar esos cambios repentinos. Algunas de las técnicas más utilizadas son:

- Detección de bordes de Sobel: Se lleva a cabo un cálculo en los cambios de intensidad tanto en direcciones verticales como horizontales, destacando así las zonas en las que los píxeles tienen cambios más drásticos.

- o Ejemplo: En imágenes con bordes gruesos es una técnica que los analizaría muy rápido, lo haría de forma efectiva.
- Detector de bordes de Canny: Es un detector de bordes más avanzado que el de Sobel ya que su proceso está formado por varias fases. Primero se aplica un filtro, generalmente el gaussiano y después se calculan los gradientes o cambios repentinos de bordes eliminando aquellos puntos cercanos que no formen parte de esos bordes principales.
 - o Ejemplo: En imágenes de edificios o planos se pueden distinguir de forma clara y nítida todas las líneas y contornos de cada parte de la imagen.

2.2 Tecnología B: Lenguaje de Programación Python

Otra tecnología a tener en cuenta y que resulta la base de todo el proyecto, es el lenguaje en el que se va a llevar a cabo el programa y la interfaz de usuario, siendo en este caso el lenguaje de programación Python [3] [4]. Constituye una herramienta de alto nivel fundamental para el procesamiento de imágenes ya que resulta sencillo de entender y utilizar y tiene a su disposición multitud de bibliotecas especializadas.

2.2.1 Bibliotecas Especializadas

Entre las bibliotecas especializadas que ofrece Python hay tres que son las que se van a utilizar a lo largo de este proyecto: OpenCV, PIL y Numpy.

- OpenCV: Es una biblioteca muy utilizada en Python para el procesamiento de imágenes que cuenta con una amplia gama de funciones para la manipulación de imágenes y detección de objetos, aunque sobre todo es utilizada para imprimir cosas por pantalla [5].
 - o Ejemplo: En casos en los que hay que aplicar filtros para suavizar una imagen o detectar sus bordes se utiliza mucho esta biblioteca
- PIL: Es otra biblioteca muy utilizada en Python para el procesamiento de imágenes. Está más centrada en la apertura, manipulación y guardado de varios formatos de imagen [6].
 - o Ejemplo: En casos en los que hay que abrir una imagen de entrada en un determinado formato antes de pasar a convertirla, se utiliza esta biblioteca.

- NumPy: Esta biblioteca sin embargo es clave para las operaciones matemáticas y cambios que haya que llevar a cabo en las imágenes, lo que es esencial para operaciones de filtrado o conversión de imágenes [7].
 - o Ejemplo: En casos en los que haya que analizar los datos de una imagen y manipular de forma numérica sus píxeles para la conversión a un determinado estilo.

2.2.2 Interfaz de Usuario

La interfaz de usuario a utilizar es esencial por ser hecha para facilitar su uso y entendimiento, debe ser intuitiva para poder interactuar y probar con el programa. Python ofrece diversas interfaces gráficas, pero para este proyecto se va a utilizar Tkinter. Con ello, se consigue crear una aplicación con un menú principal que funcione mediante botones, lo que le hace más accesible [8].

3. Especificaciones y restricciones de diseño

3.1 Especificaciones

1. Funciones principales:

Las funciones del programa deben ser múltiples para que el usuario que lo vaya a utilizar tenga la oportunidad de probar distintos efectos en los que convertir su imagen. Además, para una mayor versatilidad, cada efecto contará con la posibilidad de variar sus parámetros en función del valor que el usuario quiera darle. Con ello aumenta la variedad de posibilidades de convertir una imagen.

2. Compatibilidad:

El programa debe ser capaz de funcionar en distintos sistemas operativos mediante el uso del mismo código, sin realizar cambios ni instalaciones de ningún tipo para disfrutar de él. Esto también aumenta la flexibilidad del sistema puesto que está al alcance de cualquier persona con un ordenador.

3. Procesamiento de imágenes:

El proceso por el que se convierten las imágenes a distintos estilos de dibujo debe dar como resultado una imagen final con calidad, siempre se debe mantener una estética para que sea atractivo visualmente, tanto la imagen en global como cada detalle que la compone, sin perder la esencia de la imagen inicial.

4. Interfaz gráfica de usuario:

La interfaz que engloba todo el sistema junto con el respectivo código debe dar lugar a un programa amigable y que resulte fácil de utilizar para los usuarios, como la biblioteca Tkinter. Para ello, se debe ofrecer un menú similar al de una aplicación en el que por medio de botones se pueda elegir la imagen de entre sus archivos, probar entre los estilos disponibles ajustando parámetros y guardar tantos resultados como se quieran. Al tratarse de un sistema útil para todo el mundo, debe de ser factible para ser usado por todo el mundo, y con ello debe tener un entorno sencillo.

5. Documentación técnica:

Todos los detalles técnicos se deben documentar para contar con un correcto desarrollo del sistema y con una descripción amplia de todas las funcionalidades disponibles, además de contar con un manual con las instrucciones de uso del programa.

3.2 Restricciones

1. Software abierto:

Para aumentar la flexibilidad y el acceso del código a cualquier persona, el código debe estar generado mediante software y bibliotecas abiertas únicamente. Por este motivo el lenguaje del código elegido es Python, debido a la multitud de bibliotecas, como Open que con cuentan con una amplia variedad de funciones para el procesamiento de imagen.

2. Programa modular:

Para poder generar un programa consistente y funcional es necesario que sea modular, de forma que cada función o característica se implementa de forma independiente. Gracias a esto, si se encuentra cualquier error o se quieren hacer mejoras se podrá llevar a cabo más fácilmente sin necesidad de reescribir e incluso borrar código que era útil.

3. Programa flexible:

El sistema debe ser flexible de forma que los usuarios, dentro de cada estilo de dibujo, tengan la libertad de probar y variar los parámetros modificando sus valores. Con esto a demás de conseguir una mayor variedad de resultados, se aseguran de que el resultado obtenido es justo el deseado.

4. Pruebas:

Es necesario llevar a cabo numerosas pruebas constantemente durante todo el proceso para ver si aquello que se está implementando funciona o por el contrario da fallos desde el inicio. De esta forma, cualquier fallo o mejora se puede reparar o añadir en cuanto se detecte el problema o la necesidad.

5. Accesible:

Otro factor importante es que el programa debe ser accesible para todo el mundo tenga los estudios o conocimientos que tenga. Para ello una restricción necesaria es que el manejo sea fácil y no sea proporcional a la habilidad del usuario.

4. Descripción de la solución propuesta

La solución propuesta se va a diferenciar claramente en una introducción sobre la arquitectura del sistema y dos apartados posteriores que son técnicas empleadas e interfaz gráfica de usuario.

La arquitectura del sistema secciona el proceso en fases esenciales que deben seguir un orden determinado y gracias a las cuales se asegura un programa modular con flexibilidad y un código seguro. Las fases en las que se divide son:

1. Entrada de imagen: El primer paso es que el sistema reciba la imagen digital en cualquier tipo de formato para poder llevar a cabo todas las transformaciones necesarias con el fin de conseguir la conversión.
2. Preprocesado: Como segundo paso, es necesario realizar unas preparaciones previas a la imagen que se recibió en la entrada. Estas preparaciones pueden ser la reducción de ruido o filtrado, ajustar el contraste o incluso la reducción de la escala de colores. Con ello se consigue resaltar todos los bordes.
3. Detección de bordes: El tercer paso es aplicar sobre la imagen un algoritmo o filtro que identifique todos los bordes resaltados en el paso anterior i analizar en ellos los contornos que más representan a la imagen o cambios de intensidad.
4. Transformación: El cuarto paso es coger esos contornos detectados en el anterior paso y modificarlos para transformarlos en el efecto que se quiera. Estas modificaciones pueden ser añadir sombreados, simular tipos de trazo y reducir o aumentar detalles.
5. Postprocesado: El quinto paso es, tras haber convertido la imagen a dibujo, aplicar filtros que mejoren la calidad para un mejor resultado final.
6. Salida de la imagen: El sexto paso y último, es recibir a la salida del sistema la imagen inicial convertida en dibujo para ver el resultado e incluso guardarlo si así se desea.

Para entender este proceso de forma más visual, se ha creado un diagrama de bloques, figura 1, que representa toda la arquitectura que debe seguir el programa.

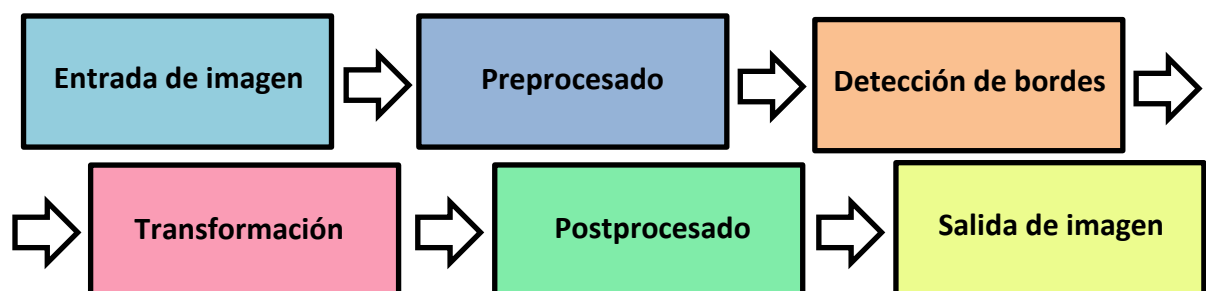


Figura 1. Diagrama de bloques de la arquitectura del sistema

4.1 Técnicas empleadas

Para este caso en el que se lleva a cabo un procesamiento de imágenes, se va a utilizar el lenguaje Python, con el cuál se cuenta también de una sintaxis clara y sencilla y multitud de bibliotecas que facilitan los procesos.

Para generar el efecto de dibujo de una imagen, se debe prestar atención sobre todo en dos factores que son los bordes y la paleta de colores [9]. Debido a esto se pueden declarar cuatro pasos básicos para este proceso:

1. Cargar la imagen que se quiere convertir.
2. Crear una máscara de borde a partir de esa imagen.
3. Reducir la paleta de colores
4. Combinar la máscara de borde creada con la imagen generada tras reducir la paleta

Antes de explicar cada paso de forma más específica, hay que tener en cuenta que el uso de bibliotecas es clave para llevar a cabo estos pasos. Las tres imprescindibles en este caso son OpenCV, PIL y numpy.

OpenCV, con su módulo cv2 en Python, y PIL, son dos bibliotecas creadas para el procesamiento de imágenes y su visión por pantalla [10]. Entre sus funciones comunes se encuentran:

- Lectura y escritura de imágenes, pudiendo así leer imágenes desde archivos y guardarlas.
- Transformación y manipulación de imágenes, de forma que te permite modificar la imagen rotándola, redimensionándola, recortándola e incluso cambiar su formato.
- Filtrado de características como el desenfoque, detección de borde y de rostros.

Aunque compartan todas estas funciones, cabe destacar que cv2 está más especializada y capacitada para el procesamiento en tiempo real y segmentación de imágenes, y PIL sobresale en el control y manejo de metadatos de imágenes.

Numpy, es una biblioteca para cálculos matemáticos y científicos capaz de operar con grandes matrices mediante multitud de funciones [11]. Entre sus funciones se encuentra:

- Crear y manipular arrays de N dimensiones.
- Operar dichos arrays realizando sumas, restas, divisiones y multiplicaciones con ellos, además de operaciones estadísticas.

Las tres bibliotecas se integran muy bien las unas con las otras sobre todo en este tipo de casos en los que se lleva a cabo un procesamiento de imagen con una posterior visualización por pantalla. De esta forma se lee la imagen mediante la biblioteca cv2 y PIL, y la imagen se compone en arrays mediante la biblioteca numpy para aplicar operaciones sobre la imagen y conseguir los resultados deseados.

El primer paso, cargar la imagen, se realiza en especial para poder seleccionarla entre los archivos que se tengan descargados y leerla, e incluso proporciona la opción de guardar la imagen tras los cambios pertinentes.

En el segundo paso se crea una máscara de bordes generada por todos los bordes y perfiles detectados que dan lugar a la imagen original y los engrosa, algo que es característico de un dibujo.

En el tercer paso se produce una reducción en la paleta de tonos que forman la imagen, ya que otro rasgo característico de los dibujos es que tienen un menor rango de variedad de colores que una imagen real. Para esta reducción se lleva a cabo el proceso de cuantización de color.

Para la cuantización de color se utiliza el algoritmo K-Means, un método de agrupamiento que analiza los datos que componen una imagen y los divide en k grupos o clusters. La norma que se rige para esta agrupación es juntar aquellos datos con información similar, siendo el número de grupos una variable que se designa por el parámetro k. Además, cada grupo se representa por un centroide que refleja la media aritmética de todos los puntos de datos que constituyen dicho grupo [12][13].

Los pasos que se deben seguir para llevar a cabo este algoritmo son:

1. Fijar el valor de k para definir cuantos grupos se quieren tener.
2. Seleccionar k puntos al azar de entre todos los puntos de datos para definir un centroide inicial para cada grupo y ajustar su posición en el centro del grupo.
3. El algoritmo analiza todo el conjunto de puntos de datos y organiza a cada punto dentro del grupo cuyo centroide se sitúe más cerca de él.
4. Al tener todos los valores definidos, se inicia un proceso iterativo en el que todos los puntos del grupo tratan de acercarse al centroide de su grupo.
5. Tras cada repetición del proceso, el centroide se recalcula para actualizar la nueva agrupación.

Con cada repetición del proceso se puede observar cómo se llevan a cabo los agrupamientos hasta que no se puedan agrupar más, así de esta forma se consigue reducir la paleta de colores de la imagen.

La totalidad del proceso se muestra en la figura 1 para comprender de forma visual el algoritmo [14].

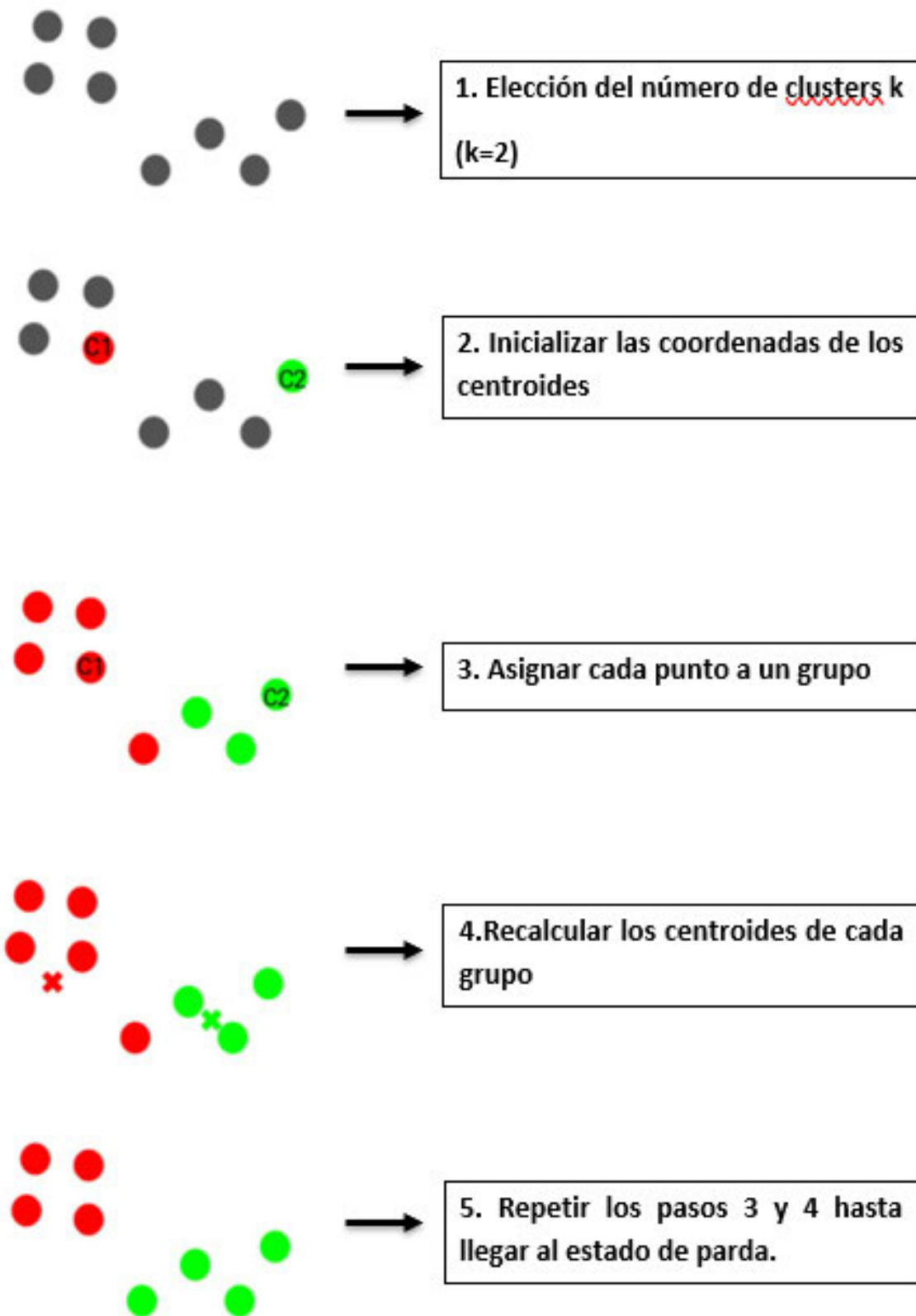


Figura 2. Proceso algoritmo K-Means

El fin del proceso, o el estado de parada, puede alcanzarse según tres criterios:

1. Se observa que los centroides han dejado de cambiar por muchas veces que se repita el proceso, por lo que se asume que se ha llegado a la convergencia.
2. Se observa que los puntos han dejado de cambiar de grupo puesto que los centroides han dejado de cambiar, de forma que se asume que se ha llegado a la convergencia.
3. Se limita el número de iteraciones que se quiere que dure el proceso, de forma que cuando estas se acaben se asume que se ha llegado a la convergencia que se buscaba.

Se pueden utilizar los tres criterios, dos de ellos e incluso solo uno, se gestionan de forma independiente puesto que son métodos de comprobación y no pasos obligatorios para que el algoritmo funcione como debe.

Tras la cuantización del color se utiliza un filtro bilateral. Este filtro no influye en el proceso de cuantización ni en el algoritmo de K-Means sino que reduce el ruido de la imagen produciendo así en ella un mayor desenfoque y una reducción de la nitidez. Estos también son rasgos característicos de un dibujo que aportarían en la conversión.

Por último, queda el cuarto paso que consta de combinar la máscara de bordes generada en el paso dos con la imagen formada por una paleta de colores más reducida generada en el paso tres.

Una vez se tiene clara la estructura general y los pasos que se deben seguir para desarrollar la parte principal del programa, se analiza, se estudia y se prueba de forma específica como conseguir el mayor rendimiento para contar con una amplia solución. Para ello, además de utilizar el lenguaje Python [15], se va a utilizar la interfaz Tkinter para dar lugar a un menú de opciones fácil de manejar para el usuario [16].

La solución propuesta para este proyecto es conseguir un programa que no solo convierta una imagen a un tipo de dibujo, sino que exista la opción de elegir qué tipo de dibujo se prefiere entre una amplia selección. Incluso, para mejorar el resultado que el usuario espera o desea obtener, se da la opción de elegir el valor que se le aplica a los parámetros clave que forman cada tipo de dibujo. De esta forma, se consigue un programa controlado por un menú en el que el usuario tiene como resultado un trabajo personalizado en función de las necesidades y preferencias.

Otras opciones con las que cuenta el menú del programa que son importantes para el proceso inicial y final de la conversión, son las opciones Cargar imagen y Guardar imagen. La primera es necesaria para poder seleccionar desde los archivos del ordenador la imagen que se quiere convertir. La segunda es necesaria para almacenar los resultados que se hayan ido obteniendo en los archivos del ordenador.

Además, se da una opción llamada Original, la cual permite al usuario retornar a la imagen original siempre que se quiera para observar cómo era la imagen inicial o para poder notar las diferencias entre la inicial y el dibujo conseguido.

La selección de efectos de dibujo está compuesta por los siguientes:

- Cartoon
- Pixelado
- Grafiti
- Cómic
- Carboncillo
- Tintado
- Acuarela
- Sombreado
- Dithering

En cuanto a la solución propuesta, ya se ha explicado el contenido de forma general, pero hay que especificar el desarrollo de cada una de las partes de forma más extensa y específica para mejorar con ello la comprensión del programa creado.

4.1.1 Cargar imagen, guardar imagen y opción original

La opción "Cargar imagen" ha sido creada para poder seleccionar las imágenes y visualizar dichos archivos desde su dispositivo o red. Este proceso se puede llevar a cabo por el uso de una interfaz gráfica que ofrece una navegación intuitiva por las carpetas de archivos del ordenador. Para ello se utiliza una función de sistema que abre una ventana de diálogo donde el usuario puede seleccionar entre archivos de distintos formatos como JPEG (Joint Photographic Experts Group), PNG (Portable Network Graphics) o GIF (Graphics Interchange Format).

Cuando se selecciona el archivo, este se verifica para asegurar que la imagen es válida y se puede llevar a cabo la carga. Este proceso se realiza mediante funciones internas que manejan archivos realizando comprobaciones de la extensión y tipos de datos. De esta forma se evitan errores durante la carga y posible futuros fallos.

Al igual que se necesita una opción para cargar la imagen, se necesita una para guardar el resultado, siendo esta la opción "Guardar imagen". Permite que los usuarios puedan almacenar los resultados obtenidos del programa. Al igual que con la opción cargar imagen, se utiliza una función de sistema que abre una ventana de diálogo donde el usuario puede seleccionar el formato y la ubicación.

También se realizan funciones internas para asegurar que la imagen se guarda de forma segura y efectiva, manteniendo la integridad de los datos y optimizando su tamaño en función de lo que prefiera el usuario.

Ambas opciones son esenciales para poder gestionar el proceso de inicio a fin de forma correcta, asegurando con ello una correcta lectura inicial y un correcto guardado final, todo ello de forma sencilla e intuitiva para el usuario.

Por último, al existir distintos tipos de efectos y la posibilidad de variar parámetros dentro de cada efecto, surge la necesidad de retornar a la imagen original ya sea para ver cómo ha cambiado al aplicar un efecto o porque se quiera volver a visualizar de cero, sin ningún tipo de modificación.

La opción "Original" permite la necesidad descrita. Se trata de volver a la imagen original eliminando todos los cambios o ajustes que se hayan realizado o probado sobre ella. Consiste en un punto de referencia clave para conocer en todo momento como era la imagen inicial desde la que se partió.

4.1.2 Efectos disponibles

Además de las opciones de cargar imagen y guardar imagen, se han generado distintos efectos disponibles entre los que el usuario puede elegir para conseguir los resultados que desee. Cada efecto ha sido creado en función de las características que se desean representar con él, y, por tanto, se han utilizado distintas funciones y métodos para cada uno.

Además, con la intención de que el usuario consiguiese un resultado más personalizado y específico, se han añadido uno o varios parámetros variables que puede elegir el usuario.

- **Cartoon**

El efecto cartoon, es un estilo de dibujo que transforma la imagen inicial haciendo que esta se vea más simple y simulando un aspecto gráfico como el de un dibujo animado. Para conseguir este efecto, se utilizan técnicas avanzadas de procesamiento de imagen.

Primero se adaptan los colores reduciendo su paleta mediante un filtro bilateral, y al mismo tiempo se destacan y guardan los bordes más significativos. De esta forma, se consigue dar claridad a los contornos y eliminar el ruido de la imagen.

Después, se utiliza un umbral adaptativo que permite ajustar los bordes destacados antes según el contenido de la imagen, llevando a cabo una transición suave entre las distintas zonas.

Ambos pasos son esenciales para conseguir que el efecto tenga una estética más acertada en relación con un dibujo animado, resaltando los bordes al mismo tiempo que se simplifican los detalles.

Los parámetros variables consisten en el tamaño del bloque y el valor C del filtro bilateral, que regulan la suavidad de los bordes y el contraste.

Parámetros variables:

- block_size: Tamaño del bloque (1-50)
- C: Valor C (1-50)

- Pixelado

El efecto pixelado, es un estilo de dibujo que transforma la imagen inicial en una imagen formada por bloques de píxeles, provocando así una estética retro o de baja resolución.

Para ello, se lleva a cabo un proceso en el que se redimensiona la imagen inicial y se ajusta al tamaño de píxel que el usuario desee.

Después se utiliza una técnica de interpolación para escalar la imagen a su tamaño original de nuevo, generando así bloques de píxeles que consiguen que la estructura de la imagen se simplifique.

Este efecto no solo se utiliza como estilo de dibujo con motivo artístico, sino que se utiliza para ocultar detalles específicos sin llegar a descartar la totalidad de la información visual y asegurando con ello la privacidad de la imagen.

El parámetro variable es el tamaño del píxel, que determina cuanto de pixelada se desea la imagen resultante.

Parámetros variables:

- pixel_size: Tamaño del pixel (1-100)

- Grafiti

El efecto grafiti, es un estilo de dibujo que transforma la imagen inicial en una representación artística que constituye arte urbano contemporáneo. Consiste en destacar sus líneas y detalles haciendo que se distingan por encima del resto de la imagen.

El proceso para conseguir este efecto comienza con el uso del algoritmo Canny, que reconoce los bordes y contornos y los realza.

Después, se utiliza una técnica para combinar todos los bordes identificados con el resto de la imagen, consiguiendo así un estilo más gráfico de la imagen dando lugar a un estilo similar al grafiti urbano.

Los parámetros variables constituyen los umbrales que se utilizan en el algoritmo de Canny para ajustar la sensibilidad de detección de bordes.

Parámetros variables:

- threshold1: Umbral 1 (0-255)
- threshold2: Umbral 2 (0-255)

- Cómic

El efecto cómic, es un estilo de dibujo que transforma la imagen inicial en una representación visual que imita el aspecto de las ilustraciones de los cómics originales.

El proceso para conseguir este efecto comienza ajustando la saturación de la imagen para conseguir una mayor intensidad de los colores, logrando con ello una paleta de colores que llame la atención.

Después, mediante el algoritmo K-Means, se separan los colores dentro de la imagen para destacar así las áreas principales. Otra característica significativa del estilo original de un cómic.

Con todo ello, se consigue captar la atención al igual que sucede en las historias de cómics donde las ilustraciones consiguen impactar.

El parámetro variable es la saturación, que varía la intensidad de los colores.

Parámetros variables:

- saturation: Saturación (0,5-2,0)

- Carboncillo

El efecto carboncillo, es un estilo de dibujo que transforma la imagen inicial en una representación artística que simula el clásico dibujo hecho a lápiz o carboncillo, caracterizado por sus líneas y trazos suaves que destacan cada detalle.

El proceso para conseguir este efecto comienza llevando a cabo una inversión de la imagen a escala de grises. Este paso es clave para resaltar los detalles que se deben destacar para caracterizar la imagen original.

Después se utiliza un filtro de desenfoque que suavice los cambios entre las zonas claras y oscuras, lo que genera textura y profundidad. Esto es una característica importante en aquellos dibujos hechos a lápiz o carboncillo.

Con todo ello se consigue transmitir la esencia del efecto carboncillo transmitiendo el estilo más clásico de dibujo.

El parámetro variable es el tamaño del kernel, que controla el grado de nitidez y el detalle de los trazos.

Parámetros variables:

- kernel_size: Tamaño del kernel (1-50)

- Tintado

El efecto tintado, es un estilo de dibujo que transforma la imagen inicial en una representación del arte pop o neón. Se caracteriza por generar una estética moderna con colores vibrantes e intensos.

El proceso para conseguir este efecto comienza utilizando una técnica de mejora que destaca los detalles esenciales y los realza con colores llamativos.

Después se emplea un balance de blancos con el que se consigue establecer una paleta de colores que genere una vibración visual. Esto es una característica clave para poder generar un efecto tintado de forma correcta.

Los parámetros variables constituyen los ajustes de balance de blancos y los parámetros de mejora de detalles.

Parámetros variables:

- sigma_s: Sigma_s (10-200)
- sigma_r: Sigma_r (0,1-1,0)

- Acuarela

El efecto acuarela, es un estilo de dibujo que transforma la imagen inicial en una representación artística que simula la técnica de pintura conocida como acuarela.

Esta se caracteriza por trazos que simulan el pincel con pintura muy fluida y compuesta por alto porcentaje de agua, además de contar con colores más suaves que el resto de los efectos.

Para conseguir este efecto se utilizan distintos tipos de técnicas, entre ellas técnicas de filtros bilaterales con los que se consigue suavizar los detalles y los colores de la imagen, y técnicas de afinamiento que simulan las pinceladas y mezclas de color como ocurre en un dibujo de acuarelas original. Con ello se consiguen caracterizar en la imagen todos los rasgos principales de la técnica de dibujo acuarela.

Los parámetros variables constituyen los parámetros sigma_s y sigma_r del filtro bilateral, que regulan la suavidad del trazado y la variedad de colores fluidos que se van a usar.

Parámetros variables:

- sigma_s; Sigma_s (10-200)
- sigma_r: Sigma_r (0,1-1,0)

- Sombreado

El efecto sombreado, es un estilo de dibujo que transforma la imagen inicial en una representación artística que consiste en un dibujo en blanco y negro que destaca las áreas en las que hay luz o sombra de la imagen.

El proceso de este efecto se lleva a cabo mediante la técnica de umbral, con la que se consigue dividir la imagen en zonas claras y oscuras.

Después se emplea adicionalmente un procesamiento para mejorar el contraste y la definición de estas áreas.

Con todo ello se consigue un efecto en el que lo que destaca es la estructura y la forma de la imagen solo con el uso de dos colores, blanco y negro.

El parámetro variable es el umbral, que regula la intensidad y la superficie de las áreas sombreadas.

Parámetros variables:

- threshold: Umbral (0-255)

- Dithering

El efecto dithering, es un estilo de dibujo que transforma la imagen inicial en una representación de la misma imagen compuesta por patrones de puntos. Se caracteriza por el uso de estos patrones de puntos dispersos por la imagen simulando distintos grados de tonalidades.

El proceso de este efecto comienza convirtiendo la imagen a escala de grises, lo que simplifica la gestión de los tonos de la imagen.

Después se aplica el algoritmo Floyd-Steinberg que realiza un tramado, técnica por la que se genera profundidad en una imagen mediante el uso de una paleta de colores limitados, para la difusión del error de cuantificación de un píxel con sus vecinos. De esta forma se consigue componer la imagen en base a patrones de puntos y generar textura con ello, cumpliendo así con las características principales del efecto dithering.

El parámetro variable se trata de la selección del método: el algoritmo de dithering Floyd-Steinberg o su base, que es simplemente el algoritmo sin que la imagen esté compuesta por puntos, lo que sería la imagen en blanco y negro.

Parámetros variables:

- method: Método (Floyd-Steinberg/Base)

4.1.3 Resultados

Para llevar a cabo la demostración de resultados, se muestran las imágenes originales seleccionadas en las figuras 2, 3 y 4:



Figura 3. Imagen rana original



Figura 4. Imagen niña original

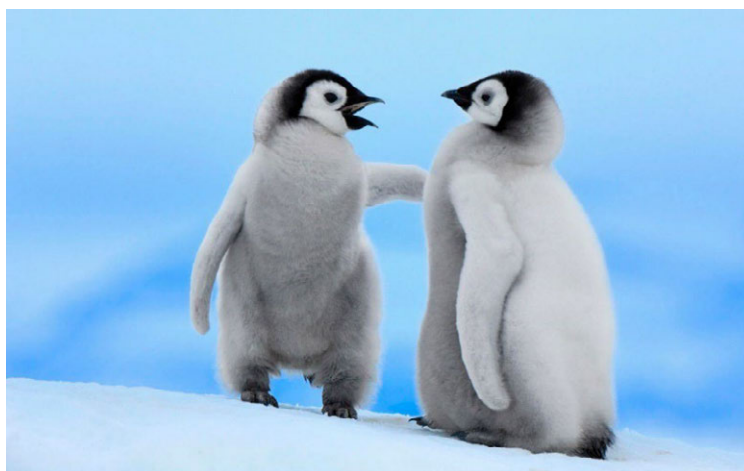


Figura 5. Imagen pingüinos original

- Cartoon

Parámetros a elección del usuario:

- Tamaño del bloque (1-50): Controla el tamaño de las pequeñas secciones en las que se dividirá la imagen para determinar los bordes. (Número menor = Bordes más suaves)
- Valor C (1-50): Ajuste fino que hace que los bordes sean más claros o más oscuros. (Número menor = Bordes más claros)

Para poder observar la diferencia entre aplicar el efecto cartoon con unos u otros valores, se van a mostrar en las figuras 5, 6 y 7 las tres imágenes seleccionadas con tres ejemplos de valores. Los valores de parámetros a utilizar, de izquierda a derecha, van a ser tamaño de bloque 10, 25 y 50, y valor C 10, 25 y 50.



Figura 6. Imagen rana con efecto cartoon



Figura 7. Imagen niña con efecto cartoon



Figura 8. Imagen pingüinos con efecto cartoon

- Pixelado

Parámetros a elección del usuario:

- Tamaño del pixel (1-100): Determina la relación cantidad / tamaño de los píxeles que forman la imagen. (Número mayor = Mayor cantidad / Menor tamaño de píxeles)

Para poder observar la diferencia entre aplicar el efecto pixelado con unos u otros valores, se van a mostrar en las figuras 8, 9 y 10 las tres imágenes seleccionadas con tres ejemplos de valores. Los valores de parámetros a utilizar, de izquierda a derecha, van a ser tamaño de pixel 10, 50 y 100.



Figura 9. Imagen rana con efecto pixelado



Figura 10. Imagen niña con efecto pixelado



Figura 11. Imagen pingüinos con efecto pixelado

- Grafiti

Parámetros a elección del usuario:

- Umbral 1 (0-255): Determina el primer nivel de sensibilidad para detectar los bordes a resaltar en la imagen. (Número menor = Mayor número de bordes resaltados)
- Umbral 2 (0-255): Determina el segundo nivel de sensibilidad para ajustar el trazado de los bordes. (Número menor = Trazados más largos y suaves)

Para poder observar la diferencia entre aplicar el efecto grafiti con unos u otros valores, se van a mostrar en las figuras 11, 12 y 13 las tres imágenes seleccionadas con tres ejemplos de valores. Los valores de parámetros a utilizar, de izquierda a derecha, van a ser umbral 1 20, 50 y 100, y umbral 2 100, 150 y 200.

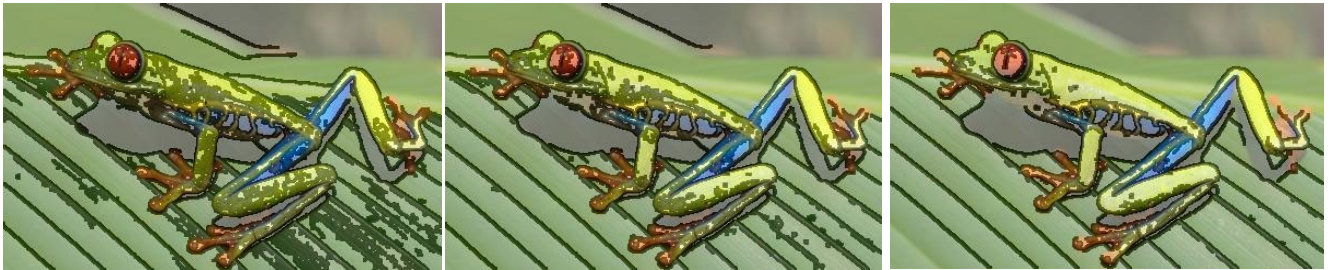


Figura 12. Imagen rana con efecto grafiti



Figura 13. Imagen niña con efecto grafiti



Figura 14. Imagen pingüinos con efecto grafiti

- Cómic

Parámetros a elección del usuario:

- Saturación (0,5-2,0): Ajusta la intensidad de los colores. (Número mayor = Colores más vivos)

Para poder observar la diferencia entre aplicar el efecto cómic con unos u otros valores, se van a mostrar en las figuras 14, 15 y 16 las tres imágenes seleccionadas con tres ejemplos de valores. Los valores de parámetros a utilizar, de izquierda a derecha, van a ser saturación 0,5 , 1,2 y 2.



Figura 15. Imagen rana con efecto cómic



Figura 16. Imagen niña con efecto cómic

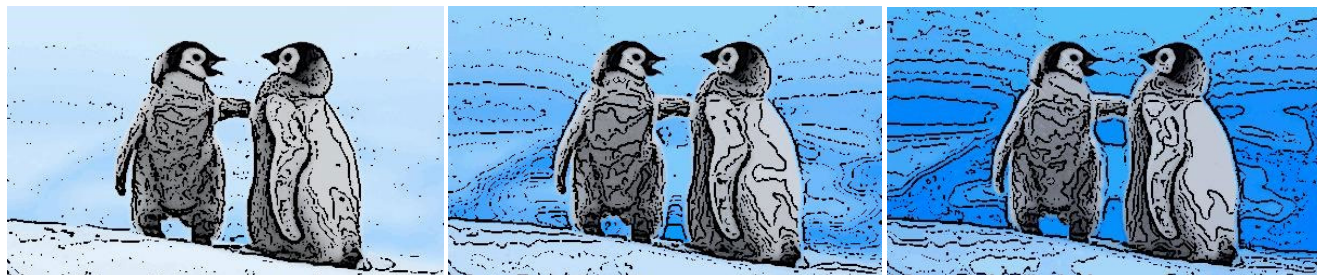


Figura 17. Imagen pingüinos con efecto cómic

- Carboncillo

Parámetros a elección del usuario:

- Tamaño del kernel (1-50): Controla el grado de detalle del trazado. (Número mayor = Trazo más detallado).

Para poder observar la diferencia entre aplicar el efecto carboncillo con unos u otros valores, se van a mostrar en las figuras 17, 18 y 19 las tres imágenes seleccionadas con tres ejemplos de valores. Los valores de parámetros a utilizar, de izquierda a derecha, van a ser tamaño del kernel 10, 25 y 50.

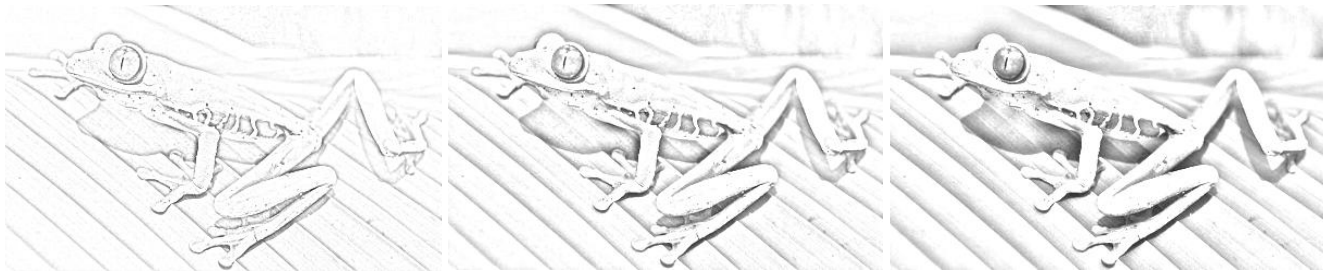


Figura 18. Imagen rana con efecto carboncillo



Figura 19. Imagen niña con efecto carboncillo



Figura 20. Imagen pingüinos con efecto carboncillo

- Tintado

Parámetros a elección del usuario:

- Sigma_s (10-200): Controla la extensión de los colores. (Número mayor = Extensión más lejana)
- Sigma_r (0,1-1,0): Controla la oscuridad de colores que se mezclan. (Número menor = Colores más claros)

Para poder observar la diferencia entre aplicar el efecto tintado con unos u otros valores, se van a mostrar en las figuras 20, 21 y 22 las tres imágenes seleccionadas con tres ejemplos de valores. Los valores de parámetros a utilizar, de izquierda a derecha, van a ser sigma s 50, 100 y 200, y sigma r 0,1, 0,5 y 1.



Figura 21. Imagen rana con efecto tintado



Figura 22. Imagen niña con efecto tintado



Figura 23. Imagen pingüinos con efecto tintado

- Acuarela

Parámetros a elección del usuario:

- Sigma_s (10-200): Controla la definición del trazado. (Número mayor = Trazado menos definido).
- Sigma_r (0.1-1.0): Controla el rango de colores que se mezclan. (Número menor = Colores más oscuros).

Para poder observar la diferencia entre aplicar el efecto acuarela con unos u otros valores, se van a mostrar en las figuras 23, 24 y 25 las tres imágenes seleccionadas con tres ejemplos de valores. Los valores de parámetros a utilizar, de izquierda a derecha, van a ser sigma s 50, 100 y 200, y sigma r 0,1, 0,5 y 1.



Figura 24. Imagen rana con efecto acuarela



Figura 25. Imagen niña con efecto acuarela

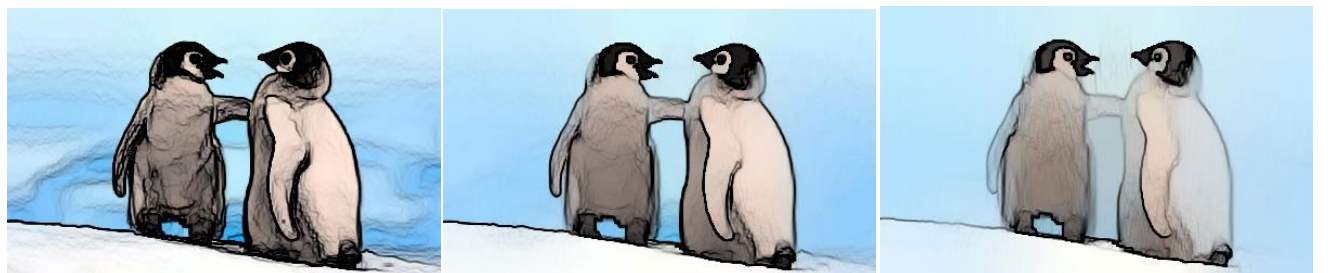


Figura 26. Imagen pingüinos con efecto acuarela

- Sombreado

Parámetros a elección del usuario:

- Umbral (0-255): Determina el grado de definición y extensión de las sombras de la imagen. (Número mayor = Mayor extensión de sombras oscuras).

Para poder observar la diferencia entre aplicar el efecto sombreado con unos u otros valores, se van a mostrar en las figuras 26, 27 y 28 las tres imágenes seleccionadas con tres ejemplos de valores. Los valores de parámetros a utilizar, de izquierda a derecha, van a ser umbral 50, 120, 200.



Figura 27. Imagen rana con efecto sombreado



Figura 28. Imagen niña con efecto sombreado



Figura 29. Imagen pingüinos con efecto sombreado

- Dithering

Parámetros a elección del usuario:

- Método (Floyd-Steinberg/Base): Elige el método de dithering: Floyd-Steinberg o Base. (Floyd-Steinberg = Blanco y Negro con puntillismo / Base = Blanco y Negro sin puntillismo).

Para poder observar la diferencia entre aplicar el efecto dithering con unos u otros valores, se van a mostrar en las figuras 29, 30 y 31 las tres imágenes seleccionadas con dos ejemplos de valores. Los valores de parámetros a utilizar, de izquierda a derecha, van a ser método Floyd-Steinberg y Base.

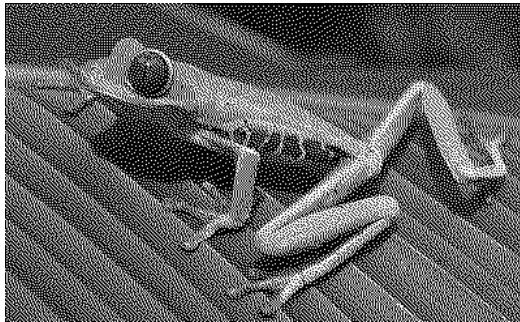


Figura 30. Imagen rana con efecto dithering



Figura 31. Imagen niña con efecto dithering

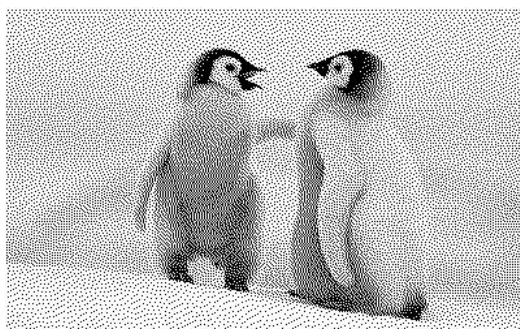


Figura 32. Imagen pingüinos con efecto dithering

4.2 Interfaz gráfica de usuario

Para poder entender mejor el funcionamiento del programa se proporciona el siguiente manual de usuario al que se puede recurrir en cualquier momento. En él se explica paso a paso todo lo necesario para obtener los resultados que se deseen.

1. Lo primero que se debe hacer es instalar el entorno sobre el que se va a trabajar. Para ello se debe acceder a la página web [17], tras lo que se abrirá la página que aparece en la figura 32, y pinchar en el botón de descarga:



Figura 33. Página web para descargar Python

2. Una vez se ha descargado e instalado el entorno, se debe comprobar que se cuentan con los 4 archivos que aparecen en la figura 33.

☐ Nombre	Tipo	Tamaño
 IDLE (Python 3.12 64-bit)	Acceso directo	3 KB
 Python 3.12 (64-bit)	Acceso directo	2 KB
 Python 3.12 Manuals (64-bit)	Acceso directo	2 KB
 Python 3.12 Module Docs (64-bit)	Acceso directo	3 KB

Figura 34. Archivos obtenidos al descargar Python

1. Para iniciar el uso del entorno se debe abrir el primer archivo llamado IDLE, tras lo que se abrirá la ventana que aparece en la figura 34.

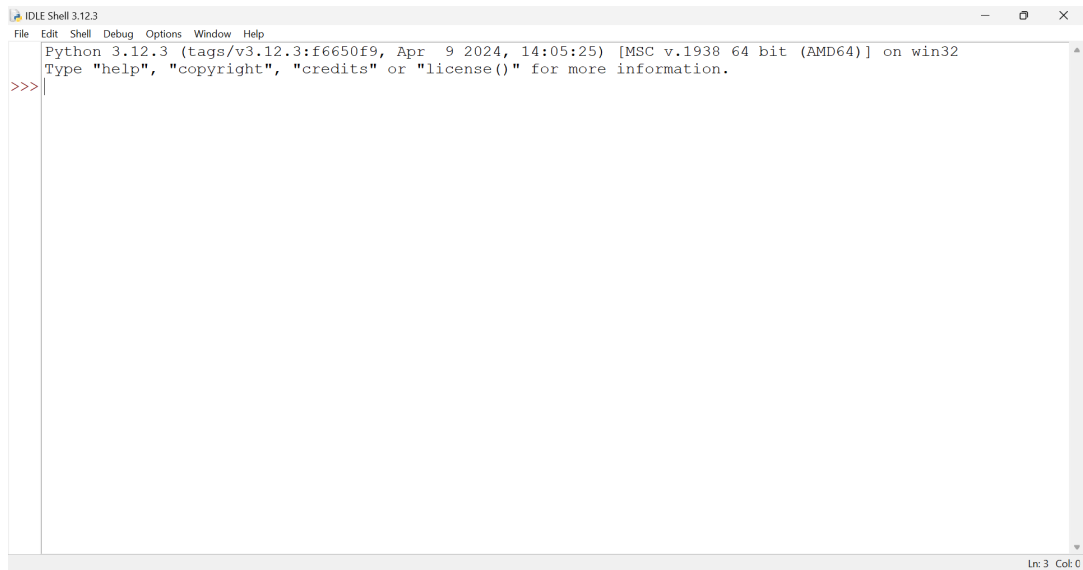


Figura 35. Primera ventana al iniciar el entorno Python

2. Dentro de la ventana que se ha abierto antes, hay que pinchar en la pestaña File que aparece arriba a la derecha. Se desplegarán una serie de opciones donde hay que seleccionar la opción Open que abrirá una ventana de archivos para seleccionar el archivo de programa con extensión .py que se quiera utilizar. Tras esto, se podrá visualizar la figura 35, en la que se ha cargado el programa de este proyecto.

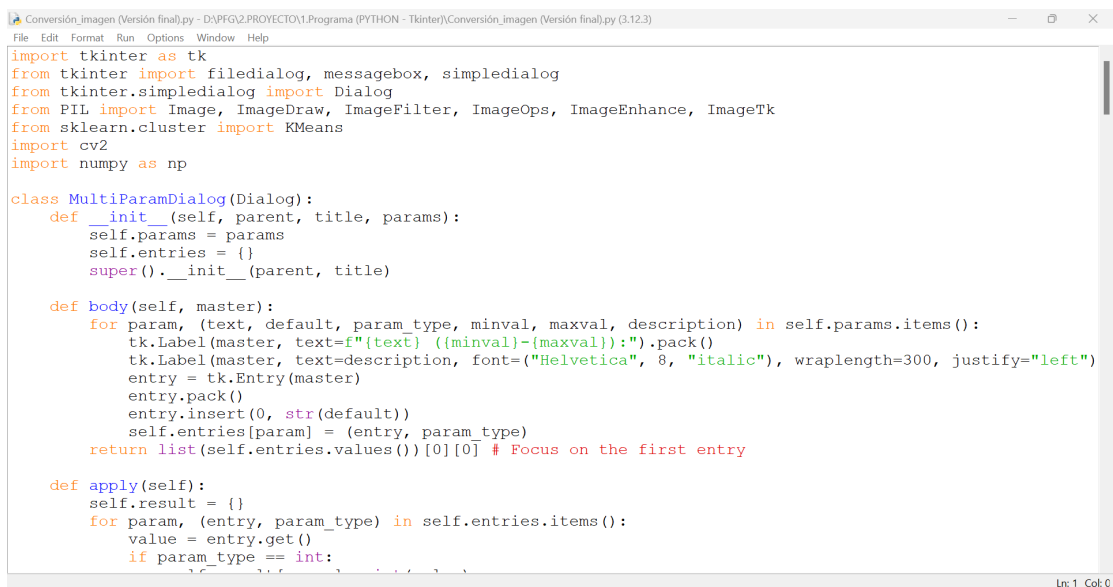


Figura 36. Segunda ventana del entorno Python tras cargar código

1. Dentro de la ventana anterior, con el código ya cargado, se debe pinchar en la pestaña Run, tras lo que se abrirá un desplegable con opciones entre las que se debe elegir la opción Run module. Esto permitirá poner en funcionamiento el programa y se cargará el menú principal como se observa en la figura 36.



Figura 37. Menú principal programa

2. Una vez se abre el menú principal del programa, se comienza con la conversión. Para ello lo primero es pinchar en Cargar imagen lo que abrirá una ventana como la de la figura 37 para seleccionar de entre los archivos del ordenador la imagen que se desee convertir.

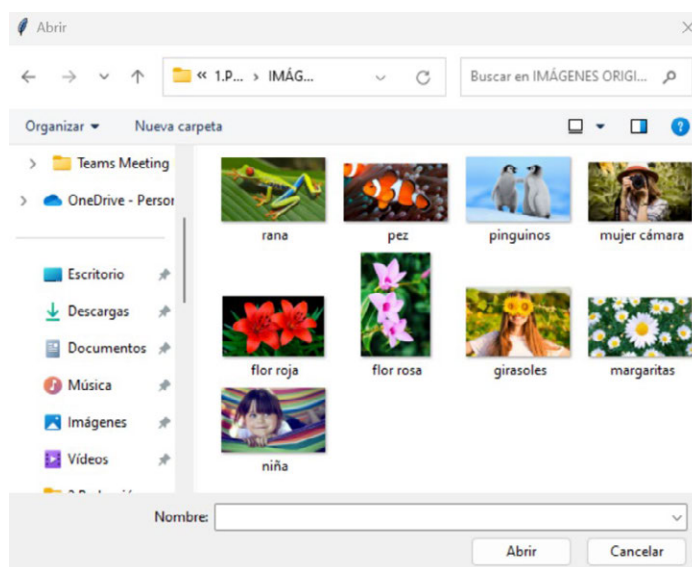


Figura 38. Ventana emergente para seleccionar imagen inicial

1. Al seleccionar una imagen, esta se visualiza en la parte superior del menú principal como se puede observar en la figura 38, así se puede ver de forma continua cada cambio.

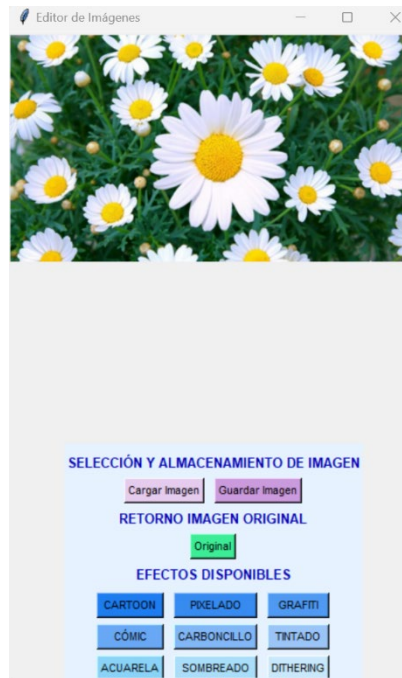


Figura 39. Menú principal programa tras cargar una imagen

2. Una vez se ha seleccionado la imagen ya se puede operar entre todas las opciones del menú con cada uno de los efectos disponibles, incluso la opción original para deshacer el efecto y retornar a la imagen inicial. Además, se puede modificar cada uno de los parámetros que se solicitan al pinchar en cada efecto, como por ejemplo en cartoon que sale una ventana como en la figura 39, aunque por defecto al abrirse la ventana aparecen introducidos unos valores con los que el resultado es adecuado.

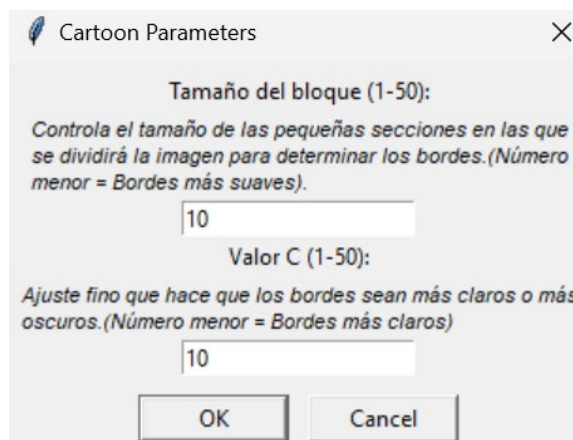


Figura 40. Ventana emergente para seleccionar los parámetros

1. Para finalizar el proceso y guardar el trabajo realizado cuando se ha probado con todos los efectos y se obtiene el resultado deseado, se clica en Guardar imagen. Para ello se abre una ventana como la de la figura 37, la cual apareció para seleccionar la imagen inicial, pero en este caso para almacenar el resultado entre los archivos del ordenador.
2. El proceso de conversión se puede repetir y probar con él tantas veces como se quiera hasta obtener todos los resultados que se buscan siguiendo los mismos pasos una y otra vez.

5. Presupuesto

El presupuesto para un proyecto se desglosa en varios aspectos a tener cuenta como son los gastos en el personal dedicado a la creación del proyecto, los gastos de equipos utilizados para ello y los gastos de material que se ha necesitado.

En cuanto a los gastos en el personal, se ha calculado una estimación del salario promedio de una persona graduada en Ingeniería Informática o Ingeniería de Telecomunicaciones que comienza su vida laboral, para tener en cuenta el mínimo que se debería cobrar. En este caso, el salario anual promedia entre los 20.000 y los 25.000, trabajando un total de 18.000 horas aproximadamente al año. Esto supone que el sueldo por hora oscilaría entre 11,11 y 13,89, por lo que se tiene en cuenta un promedio, siendo esto 12,50 euros la hora.

Hay que tener en cuenta el 30% correspondiente a la seguridad social y otros beneficios, lo que supone un incremento del sueldo por hora:

Sueldo por hora = 12,50 euros/hora x 1,30 = 16,25 euros/hora

La totalidad de horas que se han invertido en este proyecto son 170 horas de trabajo, las cuales se han destinado de forma equilibrada a cada etapa siguiendo el siguiente desglose:

- Investigación en profundidad: 40 horas
- Diseño y planificación: 30 horas
- Implementación del sistema: 40 horas
- Optimización y pruebas: 20 horas
- Memoria del proyecto: 30 horas
- Evaluación y ajustes finales: 10 horas

Teniendo en cuenta el total de 170 horas de trabajo y el sueldo por hora del trabajador requerido, el coste total de personal es:

170 horas x 16,25 euros/hora = 2.762,50 euros

En cuanto a los gastos de equipos, el proyecto se ha llevado a cabo únicamente mediante el uso de un ordenador Lenovo Yoga 7i, cuyo precio en el mercado es de aproximadamente 1.000 euros.

Para realizar un cálculo del gasto de forma más precisa, se realiza una proporción en función del tiempo que se ha utilizado. Para ello se estima que la duración del ordenador será de unos 4 años, lo que significaría un gasto de 250 euros anuales. Teniendo en cuenta que para este proyecto se utilizaría 4 meses dentro del año, se calcula un gasto mensual de 21 euros. Esto suma un total de 84 euros

- Ordenador Lenovo Yoga 7i: 84 euros

En cuanto a los gastos de material, el proyecto se ha desarrollado con programas y herramientas de software abiertos y gratuitos, por lo que no se incluyen dichos gastos.

Tabla 1. Gastos proyecto

Gastos	Costes (euros)
Gastos en personal	2.762,50
Gastos de equipos	84,00
Gastos de material	0
TOTAL	2.846,50

Teniendo en cuenta cada tipo de gasto de este proyecto, lo cual incluye todos los gastos que tendría en cuenta una empresa o institución que quisiese realizar este desarrollo, se calcula un total aproximado de 3,762,50 euros.

6. Impacto del proyecto

En este apartado se analizan las distintas implicaciones sociales, de salud y seguridad, ambientales, económicas, tecnológicas e industriales, así como la Contribución a los Objetivos de Desarrollo Sostenible (ODS) que conllevan este proyecto sobre la conversión de imagen real a dibujo [18].

1. Implicaciones Sociales

Este proyecto cuenta con diversas implicaciones sociales:

-Acceso a nuevas tecnologías: El poder acceder a herramientas tecnológicas avanzadas, como las relacionadas con este proyecto que pueden convertir imágenes reales en dibujos, promueve la creación de nuevas tecnologías y herramientas y con ello la creatividad.

-Inclusión social: El software con el que se lleva a cabo este proyecto es libre y gratuito, totalmente disponible para todo el mundo, lo que hace que personas con distintos niveles económicos puedan acceder a ello de igual forma.

-Formación y educación: Se trata de un proyecto que se puede utilizar con fines educativos en cursos de programación, informática, procesamiento de imagen e incluso diseño gráfico, promoviendo con todo ello el aprendizaje de nuevas técnicas y métodos.

2. Implicaciones de Salud y Seguridad

Este proyecto no cuenta con implicaciones de salud y seguridad ya que se trata de utilizar un software, por lo que no existe ningún riesgo para los usuarios que lo vayan a utilizar. Sin embargo, sí que puede existir un problema de salud y es el uso prolongado e intenso de los ordenadores, lo que provoca problemas con la ergonomía y la visión debido al tiempo ante la pantalla. Ante esto, se deben intentar reducir los riesgos con descanso y llevando a cabo movilidad física cada cierto tiempo.

3. Implicaciones Ambientales

Este proyecto cuenta con unas mínimas implicaciones ambientales:

-Utilizar un software libre: Al usarse un software libre y gratuito no hay necesidad de licencias ni trámites costosos, por lo que se reduce el consumo de recursos adicionales.

-Reducción del uso de papel: Este proyecto promueve la manipulación de archivos e imágenes digitales por lo que se evita la necesidad de imprimir y utilizar papel, contribuyendo así al medio ambiente

4. Implicaciones Económicas

Este proyecto cuenta con diversas implicaciones económicas:

- Reducción de costes: Este proyecto utiliza un software libre y gratuito, lo que disminuye en gran medida los costes totales evitando con ello tener que pagar licencias o gastos adicionales que son necesarios para otro tipo de programas.
- Nuevas oportunidades de negocio: Toda la tecnología innovadora que se desarrolla para dar lugar a servicios o necesidades genera una línea de negocio nueva que se puede seguir desarrollando y mejorando con el tiempo.

5. Implicaciones Tecnológicas e Industriales

Este proyecto cuenta con diversas implicaciones tecnológicas e industriales:

- Técnicas innovadoras en procesamiento de imágenes: Proyectos como este contribuyen al avance e innovación de nuevas técnicas para procesar imágenes, incluso mejorar las ya creadas para con el tiempo aumentar su calidad y eficiencia.
- Aplicación en diversas industrias: Existen distintos sectores en los que este proyecto puede ser útil además de la informática, como por ejemplo en publicidad, entretenimiento, educación o diseño.

6. Contribución a los Objetivos de Desarrollo Sostenible [19].

Este proyecto cuenta con diversos objetivos de desarrollo sostenible:

- ODS 4: Educación de Calidad: Facilita recursos educativos en el ámbito de la tecnología y la informática, promoviendo el aprendizaje y la formación continua.
- ODS 9: Industria, Innovación e Infraestructura: Fomenta la innovación tecnológica y puede ser aplicado en la industria para mejorar procesos y productos.
- ODS 12: Producción y Consumo Responsables: Promueve el uso de herramientas digitales que reducen la necesidad de recursos físicos como el papel, fomentando prácticas de producción y consumo más sostenibles.

7. Conclusiones

En este proyecto se ha llevado a cabo el desarrollo de un programa que realiza la conversión de imágenes reales a dibujos mediante técnicas de procesamiento de imagen en lenguaje Python. Se ha realizado una implementación de diversas técnicas avanzadas asociadas a la transformación y manipulación de las imágenes, y todas ellas han demostrado ser viables y eficaces dando lugar a la solución propuesta.

El objetivo principal de este proyecto era generar un programa que fuese capaz de transformar una imagen real en distintos estilos de dibujos. Para ello la clave son técnicas de procesado de imagen con las que se detectan los bordes, se reducen o modifican las paletas de colores y se aplican de distintas formas cada uno de los efectos. El sistema creado permite la conversión a varios efectos, pero también ofrece un menú intuitivo generado con una interfaz gráfica, Tkinter, que facilita el uso de todos los usuarios que quieran o necesiten utilizar este programa.

Para poder obtener una solución que funcionase tal como se buscaba ha sido necesario llevar a cabo numerosas pruebas tanto para arreglar errores como para mejorar el desarrollo del programa. En estas pruebas se han investigado diversos estilos de dibujo hasta tener claro cuáles eran los más usados, factibles y atractivos visualmente. Además, en estas pruebas se añadió la opción de que cada efecto contase con parámetros variables por parte del usuario, resultados que se han mostrado en este proyecto, para lo que se tuvieron que hacer mayor cantidad de pruebas. Todos los pasos y pruebas en su totalidad demuestran la capacidad del sistema para devolver resultados de alta calidad y todos los ajustes llevados a cabo para facilitar lo máximo posible al usuario su experiencia con este programa, lo que ha generado un programa muy preciso.

Durante todo el desarrollo del proyecto se han adquirido grandes conocimientos sobre técnicas de procesamiento de imágenes, su manipulación y aplicación en la conversión a distintos efectos. Se han obtenido una totalidad de 9 efectos disponibles que son cartoon, pixelado, grafiti, cómic, carboncillo, tintado, acuarela, sombreado y dithering, además de los conocimientos adquiridos sobre la interfaz gráfica Tkinter que permite englobar todos los efectos con las funcionalidades y usos necesarios por los que se creo el programa. Todo ello en global ha dado lugar a altos conocimientos sobre todas las técnicas y métodos investigados y empleados para obtenerla solución buscada.

El proyecto se ha llevado a cabo con un presupuesto total de 3.762,50 euros. Este presupuesto cubre todos los aspectos necesarios para el desarrollo y la implementación de la solución:

- Gastos en Personal: Se estimaron 2.762,50 euros. Este cálculo se hizo en función de un sueldo promedio de 16,25 euros por hora, que incluye el salario y beneficios asociados, y en que el trabajo realizado se realizó en un total de 170 horas, incluyendo investigación, diseño, implementación, optimización, y la redacción de la memoria del proyecto.

- Gastos de Equipos: Se tuvo un gasto de 84,00 euros para la utilización de un ordenador Lenovo Yoga 7i, necesario para el desarrollo del proyecto.
- Gastos de Material: No se han tenido gastos adicionales de material, ya que se utilizaron herramientas y software de código abierto y gratuito.

Este desglose del presupuesto demuestra los costes relacionados con la realización del proyecto en un contexto profesional, mostrando que se han optimizado los recursos para alcanzar los objetivos establecidos.

En conclusión, el proyecto ha alcanzado sus objetivos de manera satisfactoria, dando lugar a un programa que proporciona una solución útil y efectiva para la conversión de imágenes a dibujos en lenguaje Python. Tras todas las pruebas implementadas y todos los escenarios que se han tenido en cuenta, se puede declarar que se ha completado con éxito el proyecto y garantizando con él el funcionamiento buscado.

Además, se puede asegurar que se han adquirido múltiples conocimientos sobre el tema, incluido todo lo relacionado con programar en lenguaje Python y todas las bibliotecas que este proporciona y que han sido utilizadas en este proyecto. Académicamente se han cubierto todas las expectativas tanto por el aprendizaje conseguido como por haber logrado un programa que puede ser una herramienta útil de forma educativa y en otros muchos sectores como la publicidad y el entretenimiento.

Puesto que todos los resultados obtenidos son satisfactorios y cumplen con las especificaciones buscadas de forma previa a la realización del proyecto, se considera que el proyecto ha sido completado con éxito en su totalidad. Por ello, no se necesitan propuestas de nuevas líneas de trabajo futuras en este momento, el sistema cumple y funciona como se pretendía.

8. Referencias

- [1] R. C. Gonzalez y R. E. Woods, Procesamiento digital de imágenes, 4ta ed., Pearson, 2018. [En línea]. Disponible: <https://www.pearson.com/store/p/digital-image-processing/P100000732934>
- [2] W. Burger y M. J. Burge, Procesamiento digital de imágenes: Una introducción algorítmica usando Java, 2da ed., Springer, 2016. [En línea]. Disponible: <https://link.springer.com/book/10.1007/978-1-4471-6684-9>
- [3] Coursera, "Python for Everybody," [En línea]. Disponible: <https://www.coursera.org/learn/python>.
- [4] Coursera, "Introduction to Data Science in Python," [En línea]. Disponible: <https://www.coursera.org/learn/python-data>.
- [5] Imagina Formación, "OpenCV en Python," [En línea]. Disponible: <https://imaginaformacion.com/tutoriales/opencv-en-python>.
- [6] Códigos Python, "Procesamiento de imágenes con Pillow (PIL) en Python," [En línea]. Disponible: <https://codigospython.com/procesamiento-de-imagenes-con-pillow-pil-en-python/>.
- [7] DataScientest, "NumPy: la biblioteca Python," [En línea]. Disponible: <https://datascientest.com/es/numpy-la-biblioteca-python>.
- [8] Real Python, "Python GUI Programming With Tkinter," [En línea]. Disponible: <https://realpython.com/python-gui-tkinter/>.
- [9] Okoro, "Turn Photos into Cartoons using Python," Towards Data Science, 11-Jun-2019. [En línea]. Disponible: <https://towardsdatascience.com/turn-photos-into-cartoons-using-python-bb1a9f578a7e>.
- [10] R. Raj, "OpenCV vs Pillow," Educative, 01-Jun-2021. [En línea]. Disponible: <https://www.educative.io/blog/opencv-vs-pillow>.
- [11] Dive into Python, "Numpy," [En línea]. Disponible: <https://diveintopython.org/es/learn/data-science/numpy>.
- [12] IEBSchool, "Algoritmo k-means: qué es y cómo funciona," [En línea]. Disponible: <https://www.iebschool.com/blog/algoritmo-k-means-que-es-y-como-funciona-big-data/>.

- [13] K. Onum, "What is K-Means in Clustering in Machine Learning?" Kindsonthegenius.com, 07-Jun-2021. [En línea]. Disponible: https://kindsonthegenius.com/blog/what-is-k-means-in-clustering-in-machine-learning/#google_vignette.
- [14] The Machine Learners, "K-Means," [En línea]. Disponible: <https://www.themachinelearners.com/k-means/>.
- [15] Dive Into Python 3, "Dive Into Python 3," [En línea]. Disponible: <https://diveintopython3.net>.
- [16] GeeksforGeeks, "GeeksforGeeks," [En línea]. Disponible: <https://www.geeksforgeeks.org>.
- [17] Python Software Foundation, "Downloads," Python.org, [En línea]. Disponible: <https://www.python.org/downloads/>
- [18] Fernández Aller, Celia; Miñano, Rafael. "Guía para trabajar la Responsabilidad Social y Ambiental (GRSA)". ETSI Sistemas Informáticos, UPM. (Versión 2.0 Mayo 2015) [En línea] . Disponible: https://oa.upm.es/35542/1/Guia_Responsabilidad_Social_y_Ambiental-V2-1.pdf
- [19] Naciones Unidas. Objetivos de Desarrollo Sostenible [En línea]. Disponible: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>

9. Agradecimientos

Este apartado es con diferencia el que más me ha costado escribir, no por falta de gente a la que agradecer, sino porque no sé cómo agradecer tanto. Por suerte he tenido a mi lado a mucha gente que me ha apoyado durante toda mi etapa en la universidad, cuya ayuda ha sido clave, y aunque sé que unas palabras no son suficientes, se las merecen.

Para mí la carrera ha sido un proceso duro que ha consistido en una carrera de fondo en la que no habría conseguido llegar hasta la meta sin cada una de las personas que se han quedado conmigo aguantando mis días buenos y malos y aconsejándome en cada bache que me encontraba por el camino. Está claro que es importante luchar por tus sueños y trabajar por aquello que se quiere, algo que creo que he logrado con creces y por lo que me considero una persona aún más fuerte, pero cuando no te quedan fuerzas ni sabes de donde sacarlas para continuar ahí están todas las personas que te quieren empujándote y ayudándote a respirar para seguir con todo hasta conseguirlo.

Primero quiero dar las gracias a mi tutor del PFG por todas las explicaciones y recursos que me ha dado, por no agobiarme en ningún momento y tener paciencia conmigo y sobre todo por estar ahí cada vez que tenía una duda. Cuando comencé a pensar en mi proyecto no sabía lo que quería hacer, pero tenía claro que quería que él me guiase y no me confundí. Gracias por todo.

Querría seguir dando gracias a todos aquellos que en algún momento durante la carrera han compartido conmigo su tiempo ya sea para alguna explicación o simplemente para distraerte y darte esas risas que tanta falta hacen algunas veces. Sobre todo, gracias a aquellos que me llevo como amigos y que estoy segura de que después de esta etapa seguiremos en contacto. Gracias por todo.

Por supuesto tengo que dar las gracias a mi familia, porque, aunque el nombre de familia muchas veces no significa nada, ellos lo significan todo. Por cada llamada para preguntarme que tal ha ido un examen, por cada llamada para consolarme sabiendo que algo había ido mal, por todas las velitas gastadas esperando conseguir algún pequeño empujón, pero sobre todo por estar ahí siempre. No toda mi familia ha podido estar de forma presencial a mi lado durante todo el proceso. Aunque la empezaron estando físicamente y luego un día en el que tuvieron que irse, han estado conmigo cada día y en cada momento, y sé que por mucho tiempo que pase los voy a tener presentes toda la vida. Gracias por todo.

Quiero dar las gracias a mi primo, mi hermano de padres distintos, que, aunque es familia merece un agradecimiento especial. Gracias porque, aunque te has graduado de Veterinaria siempre me has preguntado y te has preocupado por mis cosas de Teleco, incluso cuando no entendías nada. Sobre todo, darte las gracias de corazón por estar ahí, aunque ni si quiera supieras que me pasaba. Gracias por todo.

No puede faltar un espacio dedicado a mis amigos, mi grupo de toda la vida. Tengo que agradecerles cada gesto que directa o indirectamente me han ofrecido durante todo el proceso, sobre todo aquellos momentos en los que más he necesitado despejar mi mente o desahogarme y ahí han estado, aunque sea para bajar a tomar algo al bar de siempre. Es una suerte tener un grupo de amigos como vosotros. Gracias por todo.

No podría mencionar mi grupo de toda la vida sin destacar a mi mejor amiga. En las malas, gracias por estar cada vez que he necesitado quedar, aunque fuera para llorar, y en las buenas, gracias por cada aventura y risa que me das. Ojalá todo el mundo consiga en la vida al menos una relación tan especial como la nuestra. Gracias por todo.

A lo largo de este apartado he mencionado muchos pilares en mi vida, pero hay un pilar que se merecería miles de apartados. No lleva conmigo desde el inicio del proceso, pero el tiempo que me ha dedicado desde que le conocí vale mucho más que el que no estuvo presente. A raíz de conocerle sentí un punto de inflexión en la carrera, las cosas empezaban a ir mejor. No sé si es por todos los intentos que ya llevaba en algunas asignaturas, porque ya cada vez me quedaba menos o porque el me daba una paz que no conseguía darme nada más, pero sé que es mi suerte. Gracias de verdad por aparecer en mi vida y apoyarme y creer tanto en mí desde el minuto 0. Gracias por todo.

Me queda dar las gracias a las personas más importantes de mi vida, a las personas que han hecho posible esto. No solo quiero agradecerlos el haberme dado la oportunidad de estudiar una carrera, algo que ya es una suerte, sino mucho más. Habéis estado conmigo desde que empecé esta carrera de fondo, levantándose en cada caída, llevándose de la mano en cualquier momento que lo necesitara y por supuesto esperándome en la meta con una sonrisa para celebrar que he podido, que lo he conseguido. Habéis creído en mí siempre, incondicionalmente, incluso cuando mucha gente pensaba que lo dejaría ahí estabais vosotros para recordarme que estabais seguros de que yo podía. Sé que no tendría forma de agradecerlos todo esto ni en mil vidas, pero espero que de verdad sintáis lo agradecida que os estoy porque os lo debo todo. Gracias por todo.

Gracias a todos de verdad por haber creído en mí y por haber estado conmigo de una forma u otra, gracias por haber llegado a mi vida para quedaros y gracias por quererme como soy aun cuando no me aguanto ni yo.

A la Gema del pasado decirle que hoy puedo asegurarla que tomó una buena decisión eligiendo esta carrera, que está trabajando en una empresa en la le gusta lo que hace y donde está muy a gusto y que enhorabuena porque ha podido con todo, aunque pensara mil veces que no era así.