

Práctica 1

Visión por computador

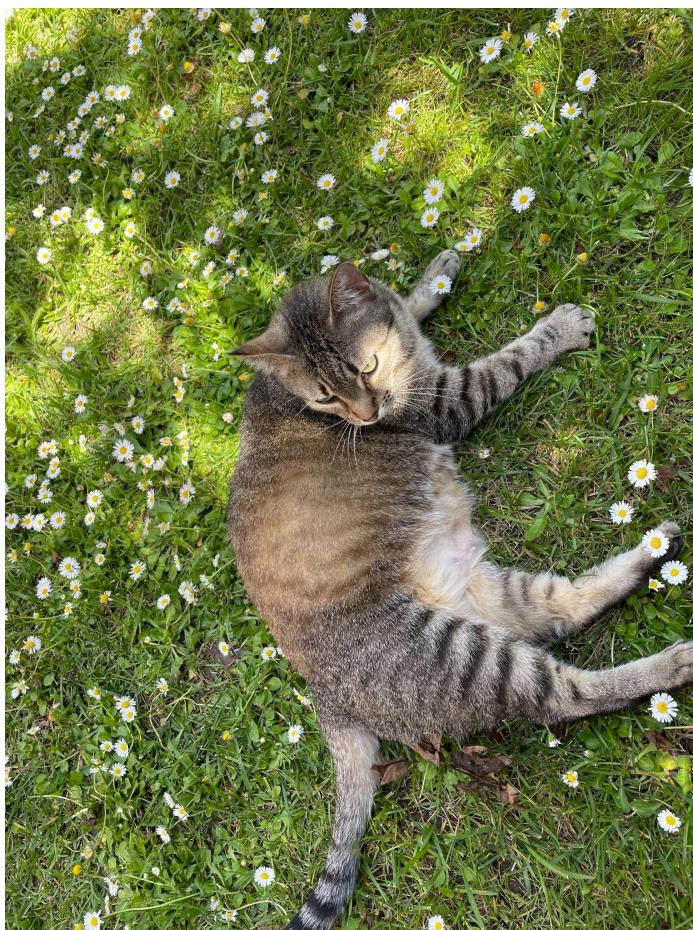
Efectos

César Moro Latorre - 815078

Alejandro Lalaguna Maza - 819860

Efectos especiales

En esta práctica hemos desarrollado una serie de efectos especiales que se pueden aplicar tanto en video en directo como en una imagen estática. Para ver los efectos individualmente los mostraremos con una foto de nuestra gata Mari.



1. Foto original de Mari

- Contraste:

El filtro de "contraste" implementado utiliza la función `convertScaleAbs()` de OpenCV, aplicando una transformación lineal a los píxeles de la imagen para ajustar su contraste y brillo mediante la fórmula $\text{imagen} * \alpha + \beta$ (hemos utilizado la operación de OpenCV por eficiencia, pero hay una implementación manual `manual_convertScaleAbs`). Los parámetros α y β controlan el aumento de contraste y el ajuste de brillo respectivamente. Tras esta transformación, la imagen se convierte a escala de grises y se aplica la ecualización del histograma para mejorar aún más el contraste, redistribuyendo los valores de intensidad de los píxeles a lo largo de todo el rango dinámico disponible.

La implementación de la ecualización del histograma obtenida a partir de ¹ incluye la conversión a escala de grises (si no lo está ya), el cálculo del histograma mediante la función calcHist() de OpenCV (por eficiencia, pero ha sido implementada por nosotros también), la obtención del histograma acumulativo normalizado para establecer una distribución de probabilidad acumulativa y el mapeo de los valores de intensidad de píxeles originales a nuevos valores de intensidad. Este proceso se aplica a la imagen en escala de grises para obtener la imagen ecualizada, que se devuelve como resultado final.

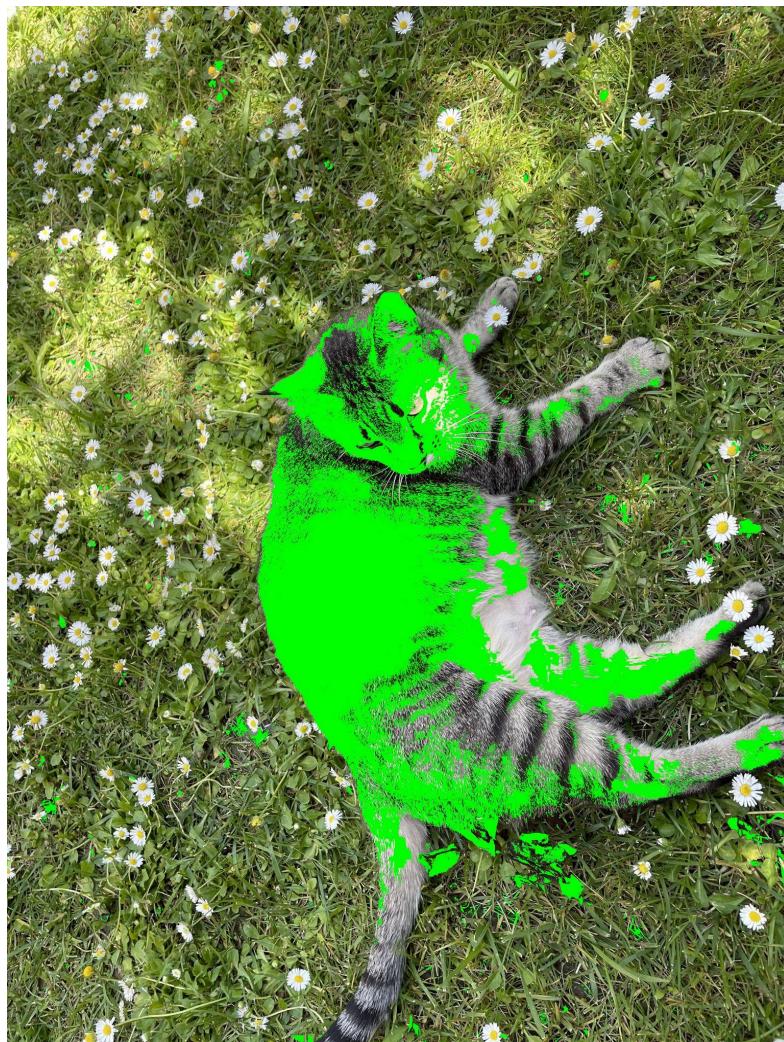


2. Filtro de contraste

- Alien: cambiar el color de la piel a color rojo, verde o azul

Este filtro define un rango en el espacio HSV que representa los tonos de piel, utilizando los valores mínimos y máximos de los componentes H (matiz), S (saturación) y V (valor). Luego, se aplica una máscara utilizando cv2.inRange() para identificar los píxeles que están dentro de este rango de tonos de piel. Finalmente, los píxeles que coinciden con la máscara se modifican para cambiar su color a verde ([0, 255, 0] en BGR), lo que produce el efecto de piel alienígena en la imagen.

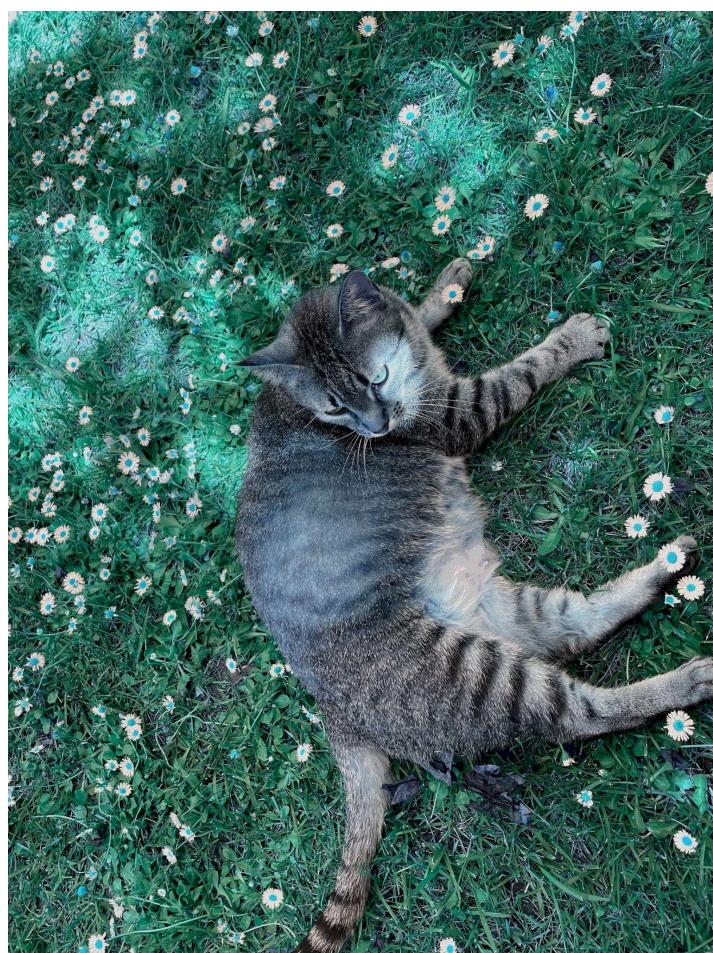
* En el video final se puede ver como afecta en nuestra piel.



3. Filtro alien

- Póster: reducir el número de colores presentes en la imagen

Este filtro lo hemos implementado reduciendo el número de colores en la imagen, lo que produce un efecto similar al de un póster. Primero, se define el número deseado de colores a utilizar. Después, se cuenta el número de colores dividiendo cada componente de color por el tamaño del intervalo entre los colores nuevos. Esto tiene el efecto de reducir el rango de colores en la imagen original, lo que produce una versión simplificada de la imagen con una cantidad limitada de colores, dando así el efecto de un póster. La cantidad de colores en la imagen final depende del valor especificado en num_colores.



4. Filtro Poster

- Distorsión: añadir distorsión de barril y de cojín ajustables

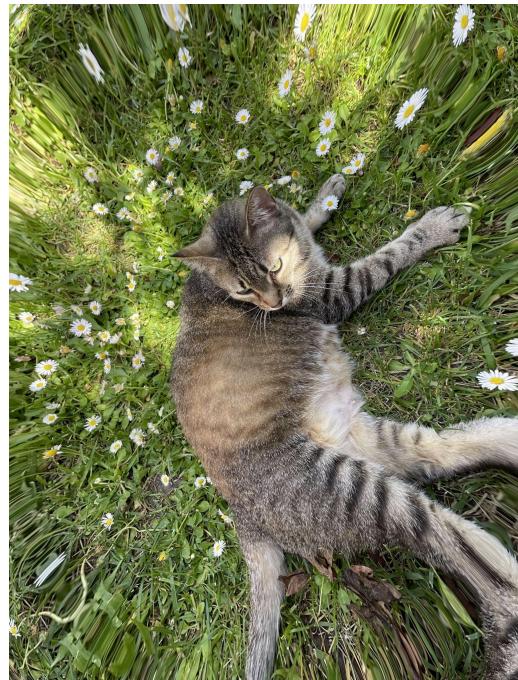
El filtro de distorsión implementa una distorsión en la imagen, lo que resulta en un efecto de deformación. Se utilizan dos coeficientes, **k1** y **k2**, que controlan la magnitud de la distorsión radial. Estos coeficientes se utilizan para calcular un nuevo radio en función del radio original r . Luego, se normalizan las coordenadas y se calculan las nuevas coordenadas ($x_{distortion}$, $y_{distortion}$) de cada píxel de la imagen en función de la distorsión radial aplicada. Estas coordenadas se desnormalizan después y son las que representan las nuevas coordenadas x e y de cada píxel después de la distorsión.

Finalmente, se utiliza la función `cv2.remap()` para aplicar la distorsión a la imagen original. Esta función toma la imagen original y realiza una interpolación para mapear los píxeles de la imagen original a sus nuevas posiciones distorsionadas, produciendo así la imagen final con el efecto de distorsión radial.

** En el video no se ve este efecto porque es costoso calcularlo y no es capaz de mostrar en tiempo real la transformación (va muy despacio)*



5. Filtro barril



6. Filtro cojín

Video

El video que hemos hecho en el cual se pueden apreciar los diferentes filtros implementados se puede consultar en el siguiente enlace:

[Video_Vision_P1.mp4](#)

Ejecución

Hemos hecho dos programas, *P1_imagen.py* y *P1_video.py*.

p1_imagen.py

El programa permite aplicar diferentes efectos a una imagen. Se puede ejecutar desde la línea de comandos, especificando la ruta de la imagen y el efecto deseado.

Opciones:

- image_path: Ruta de la imagen a procesar. Obligatorio.
- efecto: Filtro a aplicar. Opciones disponibles: contraste, ecualizacion, alien, poster, barril, cojin, blur. Valor por defecto: contraste.
- save: Indica si se debe guardar la imagen resultante. Opcional.

Ejemplo de ejecución:

```
$python p1_imagen.py --image_path imagen.jpg --efecto alien --save
```

p1_video.py

El programa permite aplicar diferentes efectos a una transmisión de cámara en vivo. Se puede ejecutar desde la línea de comandos, especificando el efecto deseado.

Opciones:

- filtro: Filtro a aplicar. Opciones disponibles: contraste, ecualizacion, alien, poster, barril, cojin, blur. Valor por defecto: contraste.

Ejemplo de ejecución:

```
python p1_video.py --efecto alien
```

Bibliografía

1. Ng, Samuel Samsudin. "Introduction to Histogram Equalization for Digital Image Enhancement." Medium, 11 Jan. 2021,
levelup.gitconnected.com/introduction-to-histogram-equalization-for-digital-image-enhancement-420696db9e43.