

**UNIVERSIDAD AUTONOMA DE YUCATAN**

**FACULTAD DE MATEMATICAS**

**“Procesamiento Digital de Imágenes (2025)”**

**ELABORADO POR:**

**Roger Aguilar Uicab.**

**Docente:**

**DR: José Luis**

***Proyecto Final de Procesamiento de Imágenes Digitales:*** *un filtro de desenfoque de fondo en tiempo real a través de una cámara web*

**INTRODUCCIÓN**

En el mundo actual de las videollamadas, transmisiones en vivo y creación de contenido digital, mejorar la calidad visual en tiempo real es una necesidad creciente. Este proyecto propone una solución utilizando técnicas de procesamiento de imágenes digitales y visión por computadora para implementar un **filtro de retrato** que difumina el fondo y mantiene el sujeto enfocado, aplicable a múltiples contextos como reuniones en línea, streaming o fotografía profesional.

**OBJETIVO GENERAL**

Desarrollar una aplicación en Python que permita aplicar un **filtro de desenfoque de fondo** en tiempo real usando una cámara web, integrando técnicas de segmentación y filtros visuales adicionales como ajustes de color, brillo y contraste.

**TECNOLOGÍAS UTILIZADAS**

* **Lenguaje de programación:** Python
* **Librerías:**
  + OpenCV: Para procesamiento de imágenes.
  + NumPy: Para manejo de arreglos y datos.
  + MediaPipe: Para la segmentación del cuerpo humano en tiempo real.
* **IDE:** PyCharm
* **Hardware:** Cámara web estándar
* **Sistema Operativo:** Windows.

**DESARROLLO DEL PROYECTO:**

Utilizando OpenCV, se accede a la cámara web para capturar el flujo de video en tiempo real. Cada frame es procesado de manera individual.

**segmentación del Sujeto**

La biblioteca MediaPipe permite detectar el cuerpo humano en tiempo real. Se crea una máscara binaria para separar el sujeto (primer plano) del fondo.

**Desenfoque del Fondo**

Una vez aislado el fondo mediante la máscara, se le aplica un **filtro Gaussiano** u otros métodos de desenfoque, y luego se superpone al sujeto no procesado.

**Filtros Adicionales**

Se desarrollaron controles para ajustar:

* **Brillo**: Modificando la intensidad de los pixeles.
* **Contraste**: Aumentando la diferencia entre tonos claros y oscuros.
* **Tono/Color**: Alterando los canales de color para distintos efectos visuales.
* **Interfaz de Usuario (GUI)**
* La aplicación cuenta con una interfaz básica creada con cv2.createTrackbar() para modificar los parámetros de los filtros en tiempo real.

**RESULTADOS**

* Se logró una segmentación eficiente del sujeto en tiempo real.
* El filtro de desenfoque se aplicó sin afectar el rendimiento.
* Los filtros adicionales funcionan en vivo y pueden ajustarse manualmente.
* La calidad de video es fluida y estable para cámaras estándar.

**Conclusión:**

* Este proyecto demostró la efectividad de las técnicas de visión por computadora aplicadas a casos reales como videollamadas y transmisión. Usar MediaPipe junto con OpenCV facilita la creación de aplicaciones con resultados profesionales. La modularidad del sistema permite extenderlo fácilmente, por ejemplo, para reemplazar fondos o integrar funciones de inteligencia artificial.