Universidad Intercontinental de la Empresa

Asignatura

Sistemas Interactivos Inteligentes

Práctica 1. Unidad IV

Calculadora basada en gestos

Alumna: Carlota Fernández del Riego

Índice

1. Resumen

Esta práctica consistió en desarrollar una calculadora controlada por gestos, utilizando la cámara web como principal medio de entrada. El sistema debe ser capaz de detectar la mano del usuario, reconocer los gestos y traducirlos en comandos para escribir números, realizar operaciones y mostrar el resultado en pantalla.

El proyecto se ha construido en Python, combinando las librerías MediaPipe, para la detección de la mano, y OpenCV, para la interfaz visual. Durante el desarrollo, se ha diseñado una estructura modular y se ha implementado una lógica de reconocimiento gestual estable, evitando falsos positivos y priorizando la usabilidad.

2. Diseño e implementación

2.1. Detección de gestos

La base del sistema es la función **fingers_up()**, que analiza los 21 puntos clave llamados landmarks, detectados por MediaPipe, y que devuelve un patrón binario con el estado de cada dedo, True en caso de que esté levantado y False si no lo está. A partir de este patrón, se han definido los gestos personalizados para representar los diferentes comandos de la calculadora.

Cada gesto debe mantenerse durante 3 segundos para que pueda ser validado por el sistema, lo que evita activaciones accidentales al mover la mano. Los diferentes gestos creados son los siguientes:

Gesto	Patrón [Pulgar, Índice, Medio, Anular, Meñique]	Acción
Mano cerrada	[False, False, False, False, False]	Borrar la operación
Solo índice	[False, True, False, False, False]	Seleccionar número
Índice y corazón	[False, True, True, False, False]	Escribir el número 2
Índice y meñique	[False, True, False, False, True]	Escribir el signo +
Pulgar y meñique	[True, False, False, False, True]	Escribir el signo *
Cuatro dedos	[False, True, True, True, True]	Escribir el signo -
Mano abierta	[True, True, True, True, True]	Escribir el número 5
Pulgar hacia arriba	[True, False, False, False, False]	Resultado de la operación

Estos gestos se han elegido buscando claridad visual, facilidad de ejecución y diferenciación entre unos y otros. Además, durante las pruebas se ajustaron los tiempos y la lógica para lograr una respuesta estable y natural.

2.2. Interfaz gráfica (GUI)

La interfaz fue creada completamente con OpenCV, sin frameworks externos, e incluye:

- Una calculadora situada en la parte superior derecha, donde se muestra la operación actual, los números y signos, y el resultado de las operaciones realizadas.

- Un efecto hover que ilumina los botones cuando el dedo índice se posiciona sobre ellos con el fin de que el usuario tenga claro qué número o signo está seleccionando en ese determinado momento.
- Indicaciones de la cantidad de dedos levantados y de acciones que se están realizando (borrando y calculando) en la esquina superior izquierda.

El usuario puede seleccionar el botón que desea seleccionar manteniendo el dedo índice encima durante 3 segundos, lo que imita el comportamiento de una pulsación física. Además, se han incluido mensajes visuales como "*BORRANDO*..." al detectar la mano cerrada, y "*CALCULANDO*..." al reconocer el gesto de pulgar arriba. Ambos mensajes son diseñados con el fin de ofrecer feedback inmediato, mejorando la experiencia de uso y aclarando el significado del gesto para asegurarle al usuario que se está realizando esa acción.

2.3. Lógica del sistema

El bucle principal del programa realiza en cada frame:

1. Captura la imagen de la cámara.
2. Procesa la detección de la mano con MediaPipe.
3. Identifica los gestos mediante el patrón de dedos.
4. Dibuja los elementos visuales y ejecuta la acción correspondiente.

Se utilizan varias variables de control para garantizar la estabilidad:

- **click_time**: Controla el tiempo mínimo entre los diferentes gestos.
- **hover_start**: Mide cuánto tiempo está el dedo índice sobre un botón.
- **stable_frames**: Cuenta los frames consecutivos con el mismo gesto para confirmar su validez.

Gracias a estas tres variables, el sistema evita que un mismo gesto se repita varias veces seguidas y asegura que la detección sea consistente incluso con ligeros movimientos de la mano.

2.4. Decisiones de diseño

- **Reconocimiento visual con MediaPipe:** Se ha optado por utilizar la librería MediaPipe Hands por su alta precisión en la detección y el seguimiento de las manos en tiempo real. Además, ofrece una integración sencilla con Python y OpenCV, sin necesidad de un entrenamiento previo ni un uso de modelos personalizados.
- **Interfaz con OpenCV:** La interfaz visual se ha implementado con OpenCV, ya que es una herramienta ligera y muy versátil para el procesamiento de imágenes. Además, facilita la renderización directa de los botones, textos y resultados, y tiene una compatibilidad con el entorno Docker proporcionado para la práctica.
- **Retardo de 3 segundos:** Se ha añadido un pequeño retardo temporal de 3 segundos entre la detección de un gesto y su validación final para ayudar a reducir los falsos positivos provocados por movimientos involuntarios. De esta manera, se mejora la estabilidad y la fiabilidad del sistema de control por gestos.

- **Colores neutros:** El diseño visual emplea tonos grises y blancos, que ofrecen un buen contraste sobre el fondo del vídeo sin distraer al usuario. Esta elección busca mantener una interfaz limpia y funcional, centrando la atención en la posición de la mano y en los elementos interactivos principales.
- **Mensajes visuales de confirmación:** Para mejorar la experiencia del usuario, se han añadido indicadores visuales como palabras clave o cambios de color, que confirman la correcta detección de un gesto y la ejecución de una operación. Estos mensajes proporcionan feedback inmediato, reforzando la sensación de control e interacción natural con el sistema.

3. Dificultades y soluciones

Dificultad	Solución aplicada
Detección errónea de dedos	Se cambió el conteo de dedos en comparación con landmarks individuales.
Gestos que se ejecutaban varias veces seguidas	Se añadió el control temporal click_time.
Lentitud inicial en la ejecución	Se optimizó la frecuencia de dibujo y se limitó el procesamiento a una sola mano.
Dificultad para seleccionar botones pequeños	Se añadió confirmación por tiempo (*hover prolongado*).

4. Resultados obtenidos

El sistema final funciona de manera estable, con una precisión aproximada del 90-95% en condiciones normales de iluminación. Además, las respuestas del sistema tienen una latencia media de unos 150 ms, lo que resulta fluido a la vista.

Por otra parte, el reconocimiento de gestos es consistente, y la interfaz visual es intuitiva incluso sin instrucciones previas, por lo que la experiencia en general demuestra que los gestos simples y bien diferenciados son suficientes para controlar un sistema básico como es una calculadora.

Para que se puedan ver todos estos resultados, he incluido un vídeo demostrativo en el cual se puede observar el funcionamiento completo del sistema desarrollado:
https://drive.google.com/drive/folders/1P9AiL6KSaTIPeYBJPArhTvcLBKOmr5ny?usp=sharing

5. Conclusiones

En conclusión, esta práctica ha permitido integrar en un mismo proyecto los conceptos de visión por computador, interacción natural y diseño de interfaces, que han sido estudiados a lo largo de las sesiones de la unidad. Implementar la detección de gestos con MediaPipe me ayudó a comprender cómo se representan las posturas de la mano mediante coordenadas y cómo filtrarlas para conseguir estabilidad.

Este sistema cumple todos los requisitos de funcionalidad básica y añade mejoras en accesibilidad y usabilidad como la confirmación temporal antes de ejecutar gestos, el feedback visual claro y la interacción sin contacto físico. En futuras versiones se podría incorporar el reconocimiento por usuario con gestos personalizados o incluso comandos por voz para hacerlo más completo.

En resumen, el sistema demuestra que la interacción gestual es una alternativa viable y accesible para el control de interfaces simples, siempre que se cuide el diseño y la estabilidad del reconocimiento.