

Asignatura

Sistemas Interactivos Inteligentes

Práctica 1

Unidad IV

Calculadora basada en gestos

MEMORIA

LUCÍA NISTAL PALACIOS

El objetivo de esta práctica era la implementación de una calculadora basada en gestos a través de la cámara de nuestro ordenador, sin utilizar el teclado o el ratón.

Para ello lo primero que hice fue definir que gestos iba a implementar para los números, las operaciones y símbolos de mi calculadora.

Mi idea inicial era hacer **los números del 1 al 9** con los dedos de las manos y el 0 representarlo como un círculo hueco o un OK (juntando índice y pulgar). Inicialmente aunque usara dos manos para hacer un 8, solo me detectaba el 5 o el 3, pero añadí una función que devuelve la suma total de los dedos y así funciona.

Después pasé a las **operaciones**, suma resta multiplicación y división. Para las **operaciones del + y el -** pensé en hacerlo también con los dedos siendo el menos: un dedo índice horizontal; y el más: un dedo vertical y otro horizontal de manos diferentes superpuestos entre sí formando un + con los dedos, pero esto tenía bastantes errores a la hora de detectarse ya que al estar superpuestos sobre todo el símbolo del + se complicaba bastante y a veces solo detectaba el dedo que estaba por delante. Si era vertical detectaba un 1 y si era el horizontal detectaba un menos, aunque se estuviera realizando el símbolo del más. Así que probé a hacerlo con un movimiento de cabeza de vertical, pero solo me detectaba la resta, así que probé a implementarlo calibrando primero la cabeza en posición “neutra”, pero seguía sin funcionar, por lo que pasé a gestos corporales, levantar brazo derecho era la suma y levantar brazo izquierda la resta, al principio no lo detectaba bien porque estaba demasiado cerca de la cámara, al alejarme un poco más era completamente funcional y detectaba sin errores.

Después procedí a **la multiplicación y la división**, para la multiplicación pensé en una cruz con los dedos pero me pasó lo mismo que con la suma y era difícil de detectar, así que probé con un puño cerrado para la multiplicación y dos puños cerrados para la división y funcionó correctamente.

Finalmente para **borrar y limpiar**, pensé hacerlo con inclinación de la cabeza pero en vez de hacia arriba y hacia abajo (pitch) que me había dado errores anteriormente, decidí hacerlo con inclinaciones hacia los lados (roll), de forma que si se inclina la cabeza a la izquierda se limpia y si se inclina a la derecha se borra (solo lo último que esté escrito en la calculadora). Y para **el punto decimal o la coma**, lo hice con dos círculos huecos, (como para el 0 pero con la dos manos), me daba error y solo detectaba lo de una mano por lo que le añadí un poco de tiempo de espera para que diera tiempo a que detectara dos manos.

Como mejoras para la usabilidad añadí entre gesto y gesto un margen de **3 segundos** para que haya un espacio, y además pudiendo leer arriba un feedback de qué gesto se está interpretando, de forma que tienes tiempo para moverte o mostrarlo mejor en caso de error. Y si pasan **6 segundos** y no se detecta nada, esto se interpretará como un **igual (=)**.

El programa tiene **tres partes principales**: la parte que detecta los gestos, la que hace las operaciones matemáticas, y la interfaz que se ve en pantalla.

Lo primero que hace el programa es capturar la imagen de la cámara usando **OpenCV**, y posteriormente la procesa con **MediaPipe**.

Para detectar los **números**, está la una función llamada **contar_dedos** que cuenta cuántos dedos hay levantados. Para el pulgar se mira si está más a la derecha o izquierda que su nudillo (porque el pulgar se mueve de lado y depende de si es la mano derecha o izquierda), y para los otros dedos se mira si la punta está más arriba que el nudillo correspondiente (así se sabe si están levantados o no).

La función devuelve cuántos dedos están levantados en cada mano. Luego, si el programa detecta **dos manos** a la vez, suma los dedos de ambas manos. Así se pueden hacer **números del 6 al 9** usando las dos manos juntas.

Para el **0** se usa el **gesto del "OK"** (hacer un círculo con el pulgar y el índice). El programa mide la distancia entre la punta del pulgar y la del índice usando la fórmula de distancia euclidiana. Si están muy cerca (menos de 0.05) lo cuenta como un gesto OK válido.

Para la **coma** quería que fueran **dos gestos de OK** a la vez, uno con cada mano. Pero había un problema y es que a veces el programa procesaba las manos en momentos ligeramente diferentes y solo detectaba una mano haciendo OK, así que lo interpretaba como 0 en vez de la coma. Para solucionar esto, cuando detecta un solo gesto de OK, el programa espera 0.2 segundos más y vuelve a comprobar. Si en esa segunda comprobación encuentra dos manos haciendo OK, entonces es un punto decimal, pero si sigue habiendo solo una, entonces es un 0.

Para detectar si hay un brazo levantado, está la función **detectar_brazo_arriba** que compara las coordenadas verticales (el eje Y) de la muñeca, el codo y el hombro. Si la muñeca está significativamente más arriba que este y más arriba que el codo, y el otro brazo no está levantado, entonces lo cuenta como brazo levantado. También se comprueba la visibilidad del punto de la muñeca para asegurar que Mediapipe realmente detecta el brazo y no es simplemente una estimación.

Para detectar el puño cerrado, la función **detectar_puño** se comprueba que las puntas de los dedos índice, medio, anular y meñique estén todas por debajo de sus nudillos correspondientes (dobladas hacia abajo). Si los cuatro dedos están doblados, es un puño. Si detecta un puño es **multiplicación**, y si detecta dos puños a la vez es **división**.

Para las funciones de borrar y limpiar se usó la inclinación lateral de la cabeza. La función **inclinacion_cabeza** calcula el ángulo (roll) de la cabeza usando las coordenadas de los ojos. Básicamente calcula el ángulo de la línea que une ambos ojos usando la función atan2 y lo convierte a grados.

Si el ángulo es menor a -15 grados (cabeza inclinada hacia la izquierda), se interpreta como **"limpiar" (C)** y resetea toda la operación. Si el ángulo es mayor a 15 grados

(cabeza inclinada hacia la derecha), se interpreta como "**borrar**" (<) y elimina solo el último gesto interpretado. Esta solución con inclinación lateral funcionó mucho mejor que los intentos anteriores con movimientos verticales, que eran más inestables.

Usar simultáneamente Hands, Face Mesh y Pose es fundamental para tener suficientes gestos claramente distinguidos y completamente funcionales. Si solo hubiera usado Hands, hubiera tenido que usar gestos más complicados con dedos y menos fiables. Al distribuir los gestos entre manos, brazos y cabeza, cada uno de los gestos puede ser más simple y a la vez más "fiable".

El **flujo** del programa sigue estos pasos:

1. Dibuja la interfaz con los botones en pantalla, aparece la calculadora
2. Captura una imagen de la cámara
3. La procesa con Mediapipe (manos, cara y brazos)
4. Detecta qué gesto se está haciendo
5. Comprueba que el gesto sea estable (5 frames seguidos)
6. Muestra el gesto detectado (feedback de color verde arriba a la izquierda)
7. Si es válido y han pasado 3 segundos desde el último gesto, lo registra
8. Si han pasado 6 segundos sin detectar ningún gesto y hay una operación escrita, se interpreta automáticamente como igual (=)
9. Según el gesto registrado, actualiza la pantalla de operaciones y resultados de la calculadora

Esto se repite todo el rato hasta que se pulsa la tecla "q", que cierra el programa.

Durante el desarrollo, se intentó ejecutar la aplicación dentro de Docker. Sin embargo, en mi ordenador, Docker no tiene acceso directo a la cámara, lo que impedía la captura de vídeo. Para solucionar esto, la calculadora se ejecuta directamente en Python, manteniendo la misma estructura y funcionalidad. Esto permite que el proyecto funcione correctamente y garantiza que en un dispositivo donde Docker sí tenga acceso a la cámara, el código original pueda ejecutarse sin modificaciones.

La calculadora cumple su función de gestos, es fácil de usar, sabiendo los gestos a realizar. En el proceso he aprendido sobre las complejidades del diseño de interfaces gestuales y la importancia de equilibrar precisión técnica con usabilidad práctica.

A continuación dejo un enlace a un **video de GoogleDrive** donde se muestra la calculadora gestual siendo usada:

https://drive.google.com/drive/folders/1DbLZtCtpMWmI-oVibHym0_IJ-TuxxXhL?usp=share_link