

Sumplete mediante reconocimiento de voz e imagen

Este proyecto permite jugar al juego Sumplete utilizando tecnologías de reconocimiento de patrones para identificar las coordenadas (fila y columna) de las celdas seleccionadas, ya sea mediante entrada por voz o por análisis de imágenes. Está implementado en **MATLAB** y combina el uso de **Modelos Ocultos de Markov (HMM)** y **técnicas avanzadas de procesamiento de imágenes**.

El objetivo es hacer el juego más accesible y dinámico, proporcionando una interfaz que interprete las instrucciones del jugador tanto desde la voz como desde imágenes capturadas en tiempo real

Contenido

- [Características](#)
- [Requisitos](#)
- [Estructura del proyecto](#)
- [Problemas](#)
- [Uso](#)
- [Participantes](#)

Características

- Reconocimiento de voz para obtener las coordenadas utilizando Modelos Ocultos de Markov (HMM).
- Reconocimiento de patrones visuales para identificar las coordenadas en imágenes.
- Interfaz interactiva para seleccionar el método deseado.
- Configurable para distintos números de centroides y estados en el HMM.

Requisitos

- MATLAB R2020b o superior.
- Toolboxes necesarios:
 - Signal Processing Toolbox
 - Image Processing Toolbox

- Archivos de modelos y codebooks previamente entrenados.
- Micrófono funcional para entrada de voz.
- Cámara funcional para entrada visual.

Estructura del proyecto

Reconocimiento de imagen

Esta parte tiene como objetivo capturar una imagen en tiempo real desde una cámara, procesarla para detectar y extraer una cuadrícula, y realizar un análisis sobre la misma. Las funciones desarrolladas abarcan desde la adquisición de imágenes hasta la corrección de su orientación y su binarización:

1. leerCuadrícula

Esta es la función principal, se encarga de capturar video en tiempo real desde la cámara y permite procesar la imagen cuando se presiona una tecla. El flujo de trabajo incluye:

1. Captura de imágenes en tiempo real.
2. Preprocesamiento de la imagen mediante la función `preprocesado`.
3. Extracción de la cuadrícula con la función `imprimirMatriz`.
4. Devuelve una matriz con los números de la cuadrícula y su tamaño.

2. preprocesado

Se encarga de transformar la imagen capturada en una versión binaria optimizada para su análisis. Sus pasos incluyen:

1. Conversión a escala de grises.
2. Ajuste de los niveles de intensidad.
3. Generación de una imagen binaria basada en un umbral.
4. Aplicación de máscaras mediante la función `aplicarMascara`.
5. Enderezamiento de la imagen mediante la función `enderezarImagen`.
6. Detección y extracción de la región más grande de interés (que debe ser la cuadrícula).

3. aplicarMascara

Aplica una máscara para rellenar las regiones en blanco dentro de la imagen binaria y ajustar los bordes. Su lógica incluye:

1. Detectar regiones donde todos los valores dentro de la máscara son `1` y establecerlos en `0`.
2. Asegurar que los bordes de la imagen se ajusten a negro.

4. enderezarImagen

Corrige la orientación de la imagen identificando las líneas verticales. Los pasos son:

1. Aplicación del método de Canny para detectar bordes.
2. Uso de la Transformada de Hough para identificar líneas rectas.
3. Selección de la línea vertical más larga y cálculo del ángulo respecto al eje vertical para rotar la imagen.

5. imprimirMatriz

Identifica y organiza los números presentes en una cuadrícula NxN.

1. Detecta las regiones conectadas en la imagen binaria.
2. Filtra las regiones según su área.
3. Compara las regiones con las plantillas de números almacenadas en la carpeta `NUMEROS`.
4. Asocia los números a las regiones y organiza los resultados en una matriz.
5. Devuelve la matriz con los valores correspondientes a la cuadrícula.

Lectura y procesamiento de una tarjeta

Esta sección tiene una implementación muy parecida a la anterior, pero aquí captura una tarjeta en tiempo real desde una cámara para procesarla y extraer el número que contiene. Esta son las funciones que abaraca:

1. leerTarjeta

Es la función principal, se encarga de capturar video en tiempo real desde la cámara y permite procesar la imagen cuando se presiona una tecla. El flujo de trabajo incluye:

1. Captura de imágenes en tiempo real.
2. Preprocesamiento y extracción del número de la imagen mediante la función `preprocesadoExtraccion`.
3. Devuelve el número que ha identificado en la tarjeta.

2. preprocesadoExtraccion

Esta función realiza el preprocesamiento de una imagen para detectar y extraer un número presente en una tarjeta. Sus pasos incluyen:

1. Conversión a escala de grises de la imagen.
2. Ajuste de los niveles de intensidad en el rango deseado para mejorar la posterior binalización mediante `imadjust`.
3. Generación de una imagen binaria mediante un umbral y se eliminan las columnas de los extremos (primer y último 25%) para reducir el ruido y centrarse en la región de interés.
4. **Detección de regiones conectadas:**
 - Se identifican las regiones conectadas en la imagen binaria usando `bwlabel`.
 - Las propiedades de estas regiones (como sus límites) se obtienen mediante `regionprops`.
 - Se filtran las regiones para eliminar aquellas que son más anchas que altas o que no contienen información relevante.
5. **Carga de plantillas:**
 - Las plantillas de números (`T_1.png` a `T_9.png`) se cargan desde una carpeta llamada `TARJETAS`.
 - Si alguna plantilla falta, se genera un error.
6. Para cada región detectada, se calcula la autocorrelación normalizada (`normxcorr2`) con cada una de las plantillas. Si las dimensiones de la región y la plantilla no coinciden, se redimensiona la más pequeña para que ambas tengan el mismo tamaño.
7. Al final, se determina la plantilla con la mayor correlación con la región analizada. Se actualiza la variable `mejorPlantilla` con el índice de la plantilla que presenta la mayor correlación y la función lo devuelve.

Reconocimiento de voz

Esta parte del proyecto se encarga de utilizar grabaciones de audio para reconocer números que corresponden a las filas y columnas de la cuadrícula. El reconocimiento se realiza mediante el uso de Modelos Ocultos de Markov (HMM) y técnicas de procesamiento de audio. A continuación, se describen las funciones principales:

1. obtenerFilaColumnaVoz

Esta función principal interactúa con el usuario para reconocer las coordenadas (fila y columna) a partir de grabaciones de audio. El flujo incluye:

1. **Interacción con el usuario:**
 - Solicita al usuario grabar su voz para indicar las coordenadas.

- Utiliza mensajes interactivos para confirmar si el número reconocido es correcto.

2. Grabación de audio:

- Usa `audiorecorder` para capturar el audio en tiempo real.

3. Extracción de características:

- Las características de la palabra grabada se obtienen mediante `obtenerCaracteristicasPalabra`.

4. Reconocimiento con HMM:

- Se comparan las características con los modelos HMM entrenados para determinar el número más probable.

5. Confirmación y ajustes:

- Si el número reconocido es incorrecto o no se reconoce, permite al usuario introducirlo manualmente.
- Incluye un límite de intentos para mejorar la robustez del sistema.

2. obtenerGrabacionesAudio

Captura múltiples grabaciones de audio para entrenamiento y prueba. Permite a los usuarios grabar palabras correspondientes a números en un rango específico.

3. extraerCaracteristicas

Procesa cada grabación de audio para extraer las características que se utilizarán en el entrenamiento y el reconocimiento. Incluye:

1. Preprocesamiento:

- **Preénfasis:** Aumenta las altas frecuencias de la señal para mejorar la robustez del reconocimiento.
- **Segmentación:** Divide la señal en tramas para análisis individual.
- **Enventanado:** Aplica una ventana para reducir discontinuidades en los bordes de las tramas.

2. Extracción de características:

- **Coeficientes Mel-Frecuencia Cepstrales (MFCC):** Obtiene las características principales de la señal en el dominio cepstral.

- **Delta y Delta-Delta:** Calcula las diferencias de primer y segundo orden para capturar características dinámicas.
- **Log-Energía:** Calcula la energía logarítmica de las tramas para añadir robustez a los MFCC.

4. obtenerCaracteristicasPalabra

Similar a `extraerCaracteristicas`, pero específicamente diseñada para procesar una sola grabación (una palabra). Es utilizada en la fase de reconocimiento.

5. cargarCodebooksModelos

Carga los modelos HMM y los codebooks necesarios para el reconocimiento. Estos modelos son previamente entrenados y se almacenan en archivos.

6. Funciones auxiliares

Estas funciones se utilizan para el preprocesamiento y análisis de las grabaciones de audio:

1. **preenfasis:** Realiza el filtro de preénfasis sobre la señal.
2. **segmentacion:** Divide la señal en tramas de tamaño fijo con superposición.
3. **enventanar:** Aplica una ventana de Hamming a las tramas segmentadas.
4. **inicioFin:** Detecta los límites de la palabra en la señal para eliminar ruido inicial y final.
5. **coeficientesMel:** Calcula los MFCC utilizando un banco de filtros Mel.
6. **liftering:** Aplica un filtro cepstral para destacar las características más relevantes.
7. **logEnergia:** Calcula la energía logarítmica de las tramas.
8. **MCCDelta:** Calcula las características delta y delta-delta.

Lógica

Esta parte del proyecto es la encargada de la interacción de los demás componentes del proyecto con el usuario, así como de calcular la puntuación en cada momento.

La puntuación se calcula sobre 10 en base al número de columnas y filas completadas.

Además se encarga de una visualización correcta del tablero de juego, mostrando en azul las filas y columnas completadas.

Problemas

Reconocimiento de imagen

1. Identificar los números de la cuadrícula:

- **Problema:** El sistema no podía clasificar bien los números, antes usaba los momentos invariantes de Hu, pero no servía para nada. Como cada imagen era distinta a la anterior no tenía mucho sentido su uso. Más tarde se usó la autocorrelación y las medidas que proporciona la función `regionProps` para poder identificarlos bien, pero seguía sin funcionar.
- **Solución:** Se implementó un enfoque basado en plantillas para mejorar la precisión en la identificación. Cada dígito fue almacenado como una plantilla de referencia. Luego, se utilizó la autocorrelación normalizada para comparar las plantillas con los números detectados en las imágenes. Este método permitió una mejor correspondencia entre las características de los números. Además, se complementó el proceso con el cálculo del número de Euler, proporcionado por la función `regionProps`, lo que ayudó a reforzar la clasificación al analizar la topología de las formas.

Reconocimiento de voz

1. Precisión del reconocimiento de voz:

- **Problema:** A veces el sistema puede no reconocer correctamente los números debido a acentos, pronunciación imprecisa, ruido de fondo, falta de entrenamiento, mal micrófono, etc.
- **Solución:** Pronunciar las palabras claramente, con el acento más neutro posible y en un lugar sin ruido. Además de buscar un mejor micrófono, varias muestras de audio, más centroides y estados. En general, eso mejoraría la precisión del HMM.

2. Grabaciones con palabras cortadas:

- **Problema:** Algunas grabaciones tienen números entrecortados. Por ejemplo, en lugar de escuchar "cero", se escucha "ero".
- **Solución:** Regrabar aquellas grabaciones.

3. Retraso en el reconocimiento:

- **Problema:** El jugador puede seleccionar la opción de fila y columna por voz, pero no decir nada.
- **Solución:** Que el jugador pulse un botón para hablar y comprobar rangos de fila o columna.

4. Detección incorrecta de número:

- **Problema:** El sistema de voz podría no reconocer ni fila ni columna bien de manera prolongada.
- **Solución:** Poner un máximo de intentos y que indique ambos de forma manual.

Uso

1. Ejecuta el programa principal en MATLAB.
2. Captura una imagen de la cuadrícula del juego Sumplete.
3. Verifica si el sistema reconoció correctamente la cuadrícula.
4. Selecciona una celda indicando una fila y una columna:
 - Utiliza comandos de voz para decir las coordenadas.
 - O utiliza plantillas de números frente a la cámara.
5. Confirma si el sistema reconoció correctamente tu selección.
6. El sistema actualizará la cuadrícula eliminando las celdas seleccionadas.
7. Repite el proceso hasta que todas las sumas coincidan con los valores objetivo.

Pariticipantes

- Ivan Cantos Piñero
- Rubén Tocón Argudo
- Rafael Caro de los Reyes