Sesión 12-Tema 9: Reconocimiento de Objetos

Tema 9 (parte 2): Reconocimiento de imágenes– Resumen

Antes de entrar en detalle, debemos diferencias los siguientes términos:

Clasificación vs localización vs detección de imágenes

- Clasificación de imágenes: Dado una imagen, etiquetarla según a la clase a la que pertenece (coche, avión, perro, etc.).
- Localización de imágenes: Localizar un objeto en concreto dentro de la imagen y marcarlo con un Bounding Box.
- Detección de objetos: Localizar dónde se encuentra un objeto específico en la imagen e informarnos del tipo de objeto que es (a que clase pertenece el objeto).

Reconocimiento de Objetos

Existen diferentes técnicas en el reconocimiento de objetos, y cada una de ellas es igual de válida que el resto. Según el problema al que nos enfrentemos, el reconocimiento tendrá una complejidad u otra, por lo que debemos aplicar algoritmos complejos adecuados a problemas complejos, y aplicar algoritmos sencillos a problemas triviales o más sencillos.

A continuación, entraremos en detalle en 2 técnicas de las más usadas, pero antes comentaremos brevemente otras técnicas.

- Coincidencia de plantillas: Partimos de una imagen cómo plantilla (véase una lata de coca cola), y el algoritmo debe buscar esa plantilla en una imagen.
- Segmentación de imágenes y análisis de blobs.

Reconocimiento de características

Una de las técnicas más usadas es el reconocimiento de objetos a partir de las características del mismo. Partimos de una imagen modelo, de la cual obtenemos el vector de características (a partir del algoritmo Sift y similares). Obtenemos el mismo vector de la imagen a evaluar, si las características coinciden, hemos acertado.

Eso es lo que dice la teoría, pero la práctica se complica un poco más. Cómo imagen modelo tenemos una base de datos con vectores de características, cada uno por imagen de ejemplo. Calculamos la distancia euclídea entre el vector a evaluar y los de prueba, si es menor a cierto umbral preestablecido, coinciden, pero no es exactamente igual.

Ahora debemos hacer una verificación geométrica, la cual comprueba si existe una transformación geométrica desde la imagen nuevo a la modelo o viceversa. Esto consiste en aplicar una rotación, escalado y transición a la imagen para comprobar si ambas son iguales.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Figura 1: Verificación Geométrica

Matriz M representa la rotación y escalado a aplicar. Matriz t de traslación, 'x' e 'y' la imagen modelo y 'u' y 'v' la imagen nueva. Si coincide (Puntos iguales), pertenece a la misma clase. Esto se puede agrupar en la siguiente ecuación:

$$Ax = b \text{ è } x = [A^t A]^{-1} A^t b$$

De este modo, despejamos los valores de rotación, escalado y traslación, y si encontramos la incógnita, es que la característica coincide.

Machine Learning / Deep Learning

La principal diferencia entre ambas es el funcionamiento interno. Machine Learning, ML para resumir, se basa en un aprendizaje supervisado (cómo Adaboost), en el cual, primero recopilamos las imágenes que vamos a usar para entrenar (ejemplos), después indicamos las características más relevantes que debe extraer (se lo indicamos nosotros: Sift, adaboost – clasificadores, etc.), por último, dividir las imágenes en las diferentes categorías. Podemos usar Adaboost, Redes neuronales, etc.

ML se usa sobre todo en el reconocimiento de objetos simples o no muy complejos, ya que debemos indicar previamente que características debe buscar en las imágenes, en contraposición del Deep Learning, DL para resumir, el cual está enfocado en el reconocimiento de objetos complejos.

DL normalmente es implementado con redes neuronales convolucionales, las cuales intentan aprender las características de las imágenes por su cuenta, analizando las imágenes y obteniendo sus datos de manera autónoma.

Para funcionar correctamente, debe tener una base de datos de ejemplo muy grande para detectar bien las características, por lo que necesita tener una buena GPU para procesar todas las imágenes.

Reconocimiento de caras

El algoritmo más utilizado es el de Viola&Jones, el cual localiza las caras dentro de una imagen. Es muy, muy rápido, fiable y tiene pocos falsos positivos. En contraposición, sólo localiza caras frontales o poco giradas, ya que deben verse los 2 ojos, además, se ve muy afectado según la iluminación de la imagen, ya que trabaja con las diferencias luminosas para localizar caras.

Se basa en la estructura de Adaboost, un conjunto de clasificadores débiles crea una cascada de clasificadores (clasificador fuertes) a aplicar a la imagen.

Para la extracción de características, se basa en las diferencias luminosas en regiones rectangulares vecinas, las cuales pueden tener 2, 3 o 4 regiones. Un ejemplo de esto es en el puente de la nariz (entrecejo), que es más claro que los ojos, de este modo localiza las características en todo el rostro. Si la zona tiene 2 regiones, calcula la diferencia de ambas zonas, si tiene 3, pondera la zona central y calcula la zona central, si tiene 4, hace la diferencia por pares.

Esto lo aplica a una imagen integral, la cual es la representación de la imagen original, pero cada píxel es la suma del mismo, más todos los que tiene a la izquierda y arriba, de modo que los cálculos de los rectángulos (diferencia de píxeles en áreas) se hacen de manera super rápida.

Reconocimiento de movimiento: Gestos, poses, etc. (Articulaciones)

Existen varias tecnologías para reconocer el movimiento:

- Vídeo: Sólo para imágenes en 2D en movimiento. No es de las más usadas debido a la limitación que tiene no poder analizar la imagen en 3 dimensiones.

- Sistema de capturas de movimientos: Usado principalmente en la industria del cine y el videojuego, se basa en que un individuo se pone un traje con puntos reflectantes, y mediante la utilización de cámaras en todos los ángulos del habitáculo y los puntos reflectantes del traje, detectan el movimiento en tiempo real de la persona.

  De este modo capturan los movimiento y animaciones de los personajes que van a incluir en los videojuego o películas. Es un sistema que presenta resultados excelentes, pero el equipo de captación de movimiento es muy caro, por lo que sólo lo utilizan en producciones con gran presupuesto.

- Nubes de puntos / Mapas de profundidad: Los más utilizados, tienen la ventaja de tener en cuenta la tercera dimensión (coordenada z). Los mapas de profundidad usan una escala de grises, miden la profundidad de la imagen con respecto a la cámara, cuanto más blanco, más cerca de la cámara.

  Un ejemplo de tecnología que aplica las nubes de puntos es el Kinect de Microsoft, el cual consta de un sensor de infrarrojos para la detección de profundidad, capas de detectar hasta 6 personas (en su última versión) y devolviendo 25 puntos del cuerpo humano (x, y, z) para la correcta detección del cuerpo humano.

Figura 2: Detección articulaciones Kinect

Kinect, un accesorio para jugar que se ha convertido en herramienta para artistas (lavanguardia.com)

Índice WIki individual