Script de Aprendizaje: Método DiffCut

[Tu Nombre]

18 de marzo de 2025

1. Introducción

En este documento aprenderemos en profundidad el método **DiffCut: Catalyzing Zero-Shot Semantic Segmentation with Diffusion Features and Recursive Normalized Cut**. Este método combina modelos de difusión con algoritmos de partición de grafos para realizar segmentación semántica sin necesidad de anotaciones (zero-shot). A lo largo del script se explican los conceptos teóricos y prácticos involucrados.

1.1. Definiciones Clave

- **Segmentación semántica:** Proceso de dividir una imagen en regiones, donde cada región corresponde a un concepto o clase.
- **Zero-shot:** Capacidad para segmentar imágenes sin haber entrenado previamente con ejemplos etiquetados.
- **Modelos de difusión:** Modelos generativos que transforman ruido en imágenes realistas mediante un proceso iterativo de denoising.

2. Modelos de Difusión y Extracción de Características

2.1. Modelos de Difusión

Los modelos de difusión generan imágenes a partir de ruido. En particular, en la modalidad de **latent diffusion** se utiliza un codificador (VQ-encoder)

para transformar la imagen original en una representación latente de menor dimensión. Esto:

- Reduce la carga computacional.
- Permite trabajar en un espacio más compacto.
- **Latent:** Representación compacta de la imagen.
- **VQ-encoder:** Codificador basado en vector quantization que convierte la imagen en un código discreto o continuo en un espacio latente.

2.2. Encoder UNet y Self-Attention

Una vez obtenida la representación latente z, se añade ruido gaussiano (simulando el proceso de difusión) y se procesa mediante el **encoder UNet** del modelo de difusión.

- **UNet:** Arquitectura con una parte de contracción (encoder) y otra de expansión (decoder) conectadas por **skip connections**.
- **Self-Attention:** Mecanismo que permite a cada posición del input .ªtender.ª todas las demás, capturando relaciones globales.

En DiffCut se extraen las características de la *última capa de self-attention*, denotadas como $\hat{z}$, ya que retienen información semántica muy rica a nivel de parches.

Diagrama (conceptual):
Imagen $I \rightarrow$ VQ-encoder $\rightarrow$ Latent $z \rightarrow$ Añadir ruido $\rightarrow$ Encoder UNet $\rightarrow$ Última capa de Self-Attention $\rightarrow \hat{z}$.

3. Construcción de la Matriz de Afinidad

Para agrupar los parches en función de su similitud semántica se construye una matriz de afinidad W.

3.1. Similitud Coseno

Para dos vectores $\hat{z}_i$ y $\hat{z}_j$ se define la similitud coseno:

$$\text{sim}(i,j) = \frac{\langle \hat{z}_i, \hat{z}_j \rangle}{\|\hat{z}_i\|_2 \, \|\hat{z}_j\|_2}$$

El valor se normaliza en el intervalo $[0, 1]$.

3.2. Aplicación del Exponente α

Para enfatizar las similitudes altas y suprimir las bajas, se eleva cada valor de similitud a la potencia α:

$$W_{ij} = \left(\frac{\langle \hat{z}_i, \hat{z}_j \rangle}{\|\hat{z}_i\|_2 \, \|\hat{z}_j\|_2} \right)^{\alpha}$$

Este proceso actúa como un *soft thresholding* en la matriz de afinidad.

4. Algoritmo de Normalized Cut (NCut) y Problema Eigen

4.1. Normalized Cut (NCut)

El objetivo del NCut es dividir el grafo formado por los parches en clusters que minimicen la disimilitud entre ellos y maximicen la similitud interna. Se define el corte como:

$$\text{NCut}(A,B) = \frac{\text{cut}(A,B)}{\text{assoc}(A,V)} + \frac{\text{cut}(A,B)}{\text{assoc}(B,V)}$$

donde:

- $\text{cut}(A,B)$ es la suma de los pesos de las aristas que unen el conjunto A con el conjunto B.
- $\text{assoc}(A,V)$ es la suma de los pesos de las aristas que unen A con todos los nodos V.

4.2. Formulación del Problema Eigen

Se define la matriz diagonal D con elementos:

$$d(i) = \sum_j W_{ij}$$

El problema se formula de la siguiente manera:

$$(D - W)x = \lambda Dx$$

Buscamos el eigenvector x asociado al segundo menor eigenvalor (el *Fiedler vector*), el cual nos proporciona una solución continua para la partición del grafo.

4.3. Bipartición

Con el Fiedler vector, se evalúan varios puntos de corte y se elige aquel que minimice el valor de NCut, obteniendo así una partición en dos grupos.

5. Partición Recursiva y Control de Granularidad

En lugar de limitarse a una sola partición, se aplica el proceso de NCut de manera recursiva:

- Cada subgrafo resultante se somete nuevamente al NCut.
- Se define un umbral τ que detiene la recursión si el costo de partición supera este valor.

Un τ bajo generará menos particiones (menos segmentos) y un τ alto permitirá particiones más finas (más segmentos).

Proceso: Repetir el NCut en cada subgrafo hasta que NCut(subgrafo) $>$ τ.

6. Asignación de Conceptos en Alta Resolución

El mapa de segmentación resultante tras la partición recursiva tiene una resolución baja (por ejemplo, 32×32). Para obtener una segmentación a resolución completa se realiza lo siguiente:

6.1. Masked Spatial Marginal Mean (SMM)

Para cada segmento se colapsa la dimensión espacial de las características para obtener un **embedding semántico** representativo.

6.2. Upsampling Bilineal

El mapa de segmentación se reescala a la resolución original mediante interpolación bilineal.

6.3. Asignación de Píxeles y Refinamiento

- Para cada píxel del mapa upsampled, se calcula la similitud con los embeddings de cada segmento (usando similitud coseno) y se asigna el segmento con mayor similitud (operación de `argmax`).
- Se aplica un módulo de **Pixel-Adaptive Refinement (PAMR)** para afinar los contornos y suavizar la segmentación.

7. Resumen del Método DiffCut

A continuación se resume el flujo completo:

1. **Entrada y Codificación:**
   Imagen $I \rightarrow$ VQ-encoder $\rightarrow$ Latent $z \rightarrow$ Añadir ruido $\rightarrow$ UNet Encoder $\rightarrow$ Última capa de Self-Attention $\rightarrow \hat{z}$.
2. **Construcción de la Matriz de Afinidad:**
   Calcular la similitud coseno entre parches y aplicar el exponente α para formar W.

3. **NCut Recursivo:**
   Resolver el problema eigen $(D - W)x = \lambda Dx$ y realizar partición recursiva hasta que NCut $> \tau$.

4. **Asignación en Alta Resolución:**
   Emplear Masked SMM, upsampling bilineal, asignación mediante `argmax` y refinamiento con PAMR para obtener el mapa final de segmentación.

8. Conclusiones y Reflexiones Finales

Ventajas del Método DiffCut:

- No requiere anotaciones para segmentar (zero-shot).
- Se adapta a la complejidad visual de cada imagen.
- Eficiencia computacional al utilizar únicamente el encoder de un modelo de difusión.

Desafíos y Futuras Líneas:

- Ajuste fino de los hiperparámetros α y τ.
- Adaptación a dominios específicos, como imágenes biomédicas.
- Integración con técnicas de open-vocabulary para la asignación de etiquetas semánticas.

9. Preguntas y Reflexiones

Utiliza este script para repasar cada componente del método DiffCut. Asegúrate de comprender tanto la teoría (por ejemplo, la formulación del problema eigen y la interpretación del Fiedler vector) como la aplicación práctica en la segmentación de imágenes.