Reporte Académico Detallado: Fundamentos y Calidad en Imagen Médica

Sesiones 1 y 2 - Especialización en Procesamiento Avanzado de Imágenes Médicas (PAIM)

Experto en Procesamiento de Imágenes Médicas

2026-01-19

Table of contents

1 Introducción

En el ámbito de la Ingeniería Biomédica, el procesamiento de imágenes no es simplemente una manipulación estética, sino un proceso de extracción de información biológica fidedigna. Este reporte detalla los contenidos de las sesiones 1 y 2, centrados en cómo la representación digital y las métricas de calidad impactan la interpretabilidad diagnóstica y la confiabilidad de los sistemas de Inteligencia Artificial.

---

2 Sesión 1: Representación Digital y Percepción Visual

La primera sesión establece el modelo matemático de la imagen y analiza cómo las limitaciones del Sistema Visual Humano (SVH) condicionan el diseño de algoritmos de procesamiento.

2.1 Modelado de la Digitalización

Una imagen médica continua $f(x, y)$ debe ser transformada en una matriz discreta $I[m, n]$. Este proceso se divide en:

1. **Muestreo Espacial:** Determina la resolución espacial. En radiografía digital, el tamaño del píxel suele oscilar entre 100 y 200 μm.
2. **Cuantización de Intensidad:** Define la resolución de contraste. Mientras que las imágenes comerciales usan 8 bits, los estándares clínicos requieren entre 12 y 16 bits para representar adecuadamente la escala de unidades Hounsfield (HU) en Tomografía o niveles de gris en Rayos X.

2.1.1 Implementación en Python: Simulación de Profundidad de Bits

El siguiente código demuestra el efecto de la reducción de la profundidad de bits (cuantización) sobre una imagen médica, evidenciando el artefacto de "falso contorneo".

```python
import numpy as np
import matplotlib.pyplot as plt
from skimage import data, img_as_float, color

def simulate_quantization(image, bits):
    """
    Simula el efecto de reducir la profundidad de bits.
    """
    levels = 2**bits
    return np.round(image * (levels - 1)) / (levels - 1)

# Cargamos una imagen de prueba y la normalizamos
image = img_as_float(data.camera())

# Simulamos diferentes profundidades
q_8bit = simulate_quantization(image, 8)
q_4bit = simulate_quantization(image, 4)
q_2bit = simulate_quantization(image, 2)

fig, axes = plt.subplots(1, 3, figsize=(18, 6))
axes[0].imshow(q_8bit, cmap='gray')
axes[0].set_title('Resolución Estándar (8-bit)')
axes[1].imshow(q_4bit, cmap='gray')
axes[1].set_title('Cuantización 4-bit (16 niveles)')
axes[2].imshow(q_2bit, cmap='gray')
axes[2].set_title('Cuantización 2-bit (4 niveles)')
for ax in axes: ax.axis('off')
plt.tight_layout()
plt.show()
```

2.2 Espacios de Color en Aplicaciones Biomédicas

Aunque el diagnóstico radiológico es monocromático, la patología digital y la endoscopia dependen críticamente del color. El espacio RGB no es adecuado para el análisis cuantitativo debido a su falta de uniformidad perceptual. Se prefiere el espacio **CIELAB**, donde la luminosidad (L^*) está desacoplada de la información cromática (a^*, b^*).

2.2.1 Implementación en Python: Conversión Perceptual

```
from skimage.color import rgb2lab, lab2rgb

# Ejemplo con imagen de retina
retina_rgb = data.retina()
retina_lab = rgb2lab(retina_rgb)

# Extracción de componentes
L_channel = retina_lab[:, :, 0] # Luminosidad (información estructural)

plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.imshow(retina_rgb)
plt.title('Imagen Original (RGB)')
plt.subplot(1, 2, 2)
plt.imshow(L_channel, cmap='gray')
plt.title('Componente de Luminosidad (L*)')
plt.show()
```

---

3 Sesión 2: Caracterización de la Calidad en Entornos Clínicos

La calidad de una imagen médica se define por su capacidad para permitir un diagnóstico preciso. Esto se cuantifica mediante tres pilares: Ruido, Contraste y Resolución.

3.1 Naturaleza Estocástica del Ruido

En imagenología médica, el ruido no es una señal aditiva simple; es un proceso dependiente de la física de adquisición:

- **Ruido de Poisson (Cuántico):** Domina en Rayos X y TC. Surge debido a la llegada aleatoria de fotones al detector. Su varianza es proporcional a la intensidad de la señal.
- **Ruido Gaussiano (Electrónico):** Originado por la instrumentación del detector y la digitalización.

3.1.1 Implementación en Python: Modelado de Ruido Clínico

```
def add_poisson_noise(image):
    """
    Simula el ruido cuántico típico de adquisiciones de Rayos X.
    """
    # Ajuste de escala para simular flujo de fotones
    peak = 50
    noisy = np.random.poisson(image * peak) / peak
    return np.clip(noisy, 0, 1)

phantom = data.shepp_logan_phantom()
noisy_phantom = add_poisson_noise(phantom)

plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.imshow(phantom, cmap='gray')
plt.title('Fantoma Ideal')
plt.subplot(1, 2, 2)
plt.imshow(noisy_phantom, cmap='gray')
plt.title('Fantoma con Ruido de Poisson (Cuántico)')
plt.show()
```

3.2 Métricas de Desempeño: SNR y CNR

Para que un algoritmo de IA o un radiólogo identifique una patología, la lesión debe ser distinguible del fondo ruidoso.

1. **SNR (Signal-to-Noise Ratio):** Relación entre la intensidad media de la señal y la desviación estándar del ruido en una región de interés (ROI).
2. **CNR (Contrast-to-Noise Ratio):** Evalúa la diferencia de intensidad entre dos tejidos (ej. tumor vs. parénquima sano) normalizada por el ruido.

$$CNR = \frac{|\mu_{tumor} - \mu_{sano}|}{\sigma_{ruido}}$$

3.2.1 Implementación en Python: Cálculo Automático de Métricas

```python
def get_metrics(img, roi_sig, roi_bg):
    """
    Calcula SNR y CNR basadas en coordenadas [y1, y2, x1, x2].
    """
    signal = img[roi_sig[0]:roi_sig[1], roi_sig[2]:roi_sig[3]]
    background = img[roi_bg[0]:roi_bg[1], roi_bg[2]:roi_bg[3]]

    mu_s = np.mean(signal)
    mu_b = np.mean(background)
    sigma_b = np.std(background)

    snr = mu_s / sigma_b
    cnr = np.abs(mu_s - mu_b) / sigma_b

    return snr, cnr

# Definición de ROIs sobre el fantoma (ejemplo ilustrativo)
# ROI señal: elipse central; ROI fondo: área negra superior
snr, cnr = get_metrics(noisy_phantom, [180, 220, 180, 220], [10, 40, 10, 40])

print(f"Métricas de Calidad:\nSNR: {snr:.4f}\nCNR: {cnr:.4f}")
```

3.3 Resolución Espacial y la MTF

3.3.1 Introducción a la Calidad de Imagen desde la Frecuencia

En el procesamiento avanzado de imágenes médicas (PAIM), la resolución espacial suele malinterpretarse como el simple número de píxeles o el tamaño de la matriz de adquisición. Sin embargo, para un ingeniero biomédico, la resolución es una propiedad dinámica que depende de la capacidad del sistema para transferir el contraste del objeto real a la imagen digital.

La **Función de Transferencia de Modulación (MTF)** es la métrica reina para caracterizar esta capacidad. Representa la fidelidad de un sistema de imagen en función de la frecuencia espacial, cuantificando cómo se degrada el contraste a medida que las estructuras se vuelven más pequeñas y densas.

Este reporte desglosa la MTF desde sus fundamentos matemáticos hasta su implementación computacional, proporcionando una visión integral esencial para el desarrollo de algoritmos de restauración e IA confiable.

3.3.2 Fundamentos Matemáticos: De la PSF a la MTF

Para entender la MTF, primero debemos definir la **Función de Dispersión de Punto (Point Spread Function - PSF)**.

3.3.2.1 La Función de Dispersión de Punto (PSF)

Si capturamos la imagen de un punto infinitesimal (una delta de Dirac $\delta(x, y)$), el sistema no devolverá un punto perfecto, sino una mancha borrosa debido a la difracción, el tamaño del foco en Rayos X o el movimiento del paciente. Esta mancha es la PSF.

Matemáticamente, la imagen $g(x, y)$ es la convolución del objeto $f(x, y)$ con la PSF $h(x, y)$:

$$g(x, y) = f(x, y) * h(x, y)$$

3.3.2.2 La Función de Transferencia Óptica (OTF)

Al aplicar la Transformada de Fourier a la PSF, pasamos del dominio espacial al dominio de la frecuencia. El resultado es la OTF:

$$OTF(u, v) = \mathcal{F}\{h(x, y)\}$$

Donde (u, v) son las frecuencias espaciales (ciclos/mm o lp/mm).

3.3.2.3 Definición de MTF

La MTF es simplemente la magnitud (módulo) de la OTF, usualmente normalizada a la unidad en frecuencia cero:

$$MTF(u, v) = \frac{|OTF(u, v)|}{|OTF(0, 0)|}$$

3.4 Visualización de la PSF y su relación con la resolución

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.fft import fft2, fftshift

def generate_psf(sigma, size=128):
    """Genera una PSF Gaussiana que simula el desenfoque del sistema."""
    x = np.linspace(-size//2, size//2, size)
    y = np.linspace(-size//2, size//2, size)
    x, y = np.meshgrid(x, y)
```

```
    psf = np.exp(-(x**2 + y**2) / (2 * sigma**2))
    return psf / np.sum(psf)

psf_sharp = generate_psf(sigma=1)
psf_blur = generate_psf(sigma=5)

fig, ax = plt.subplots(1, 2, figsize=(12, 5))
ax[0].imshow(psf_sharp, cmap='hot'); ax[0].set_title('PSF Sistema Alta Resolución')
ax[1].imshow(psf_blur, cmap='hot'); ax[1].set_title('PSF Sistema Baja Resolución')
plt.show()
```

4

3 Cascada de Modulación: El Sistema como Filtro

En un entorno clínico (ej. un tomógrafo), la MTF total no depende de un solo componente, sino que es el producto de las MTF individuales de cada etapa de la cadena de adquisición:

$$MTF_{total}(u) = MTF_{foco}(u) \cdot MTF_{detector}(u) \cdot MTF_{movimiento}(u) \cdot MTF_{algoritmo}(u)$$

Esta propiedad es fundamental. Si el detector tiene una resolución excelente pero el foco del tubo de Rayos X es grande, la MTF total se verá limitada por el componente más débil.

4.1 Determinación Práctica de la MTF: El Método del Borde

Medir una PSF directamente es difícil porque no existen "puntos perfectos" en la práctica clínica. Por ello, utilizamos la **Función de Respuesta al Borde (Edge Response Function - ERF)**.

4.2 El Proceso de Cálculo

1. **Adquisición:** Se toma la imagen de un borde afilado (ej. una placa de plomo).
2. **ERF:** Se extrae el perfil de intensidades perpendicular al borde.
3. **LSF (Line Spread Function):** Se deriva la ERF para obtener la respuesta a una línea.

$$LSF(x) = \frac{d}{dx} ERF(x)$$

4. **MTF:** Se aplica la Transformada de Fourier a la LSF.

4.2.1 Implementación en Python: Cálculo de MTF desde un Borde

```
import numpy as np
from scipy.ndimage import gaussian_filter1d
from scipy.fft import fft

def calculate_mtf_from_edge(edge_profile):
    """
    Calcula la MTF a partir de un perfil de borde (ERF).
    """
    # 1. Derivada para obtener la LSF
    lsf = np.diff(edge_profile)

    # 2. Ventaneo (Hanning) para reducir ruido en los extremos
    lsf = lsf * np.hanning(len(lsf))

    # 3. Transformada de Fourier
    mtf = np.abs(fft(lsf))

    # 4. Normalización
    mtf = mtf / mtf[0]

    return mtf[:len(mtf)//2]

# Simulación de un perfil de borde con ruido
x = np.linspace(0, 100, 1000)
erf_ideal = np.where(x < 50, 0, 1)
erf_real = gaussian_filter1d(erf_ideal, sigma=5) + np.random.normal(0, 0.01, 1000)

mtf_result = calculate_mtf_from_edge(erf_real)
```

```
plt.figure(figsize=(10, 5))
plt.plot(mtf_result, label='MTF Sistema Simulado')
plt.xlabel('Frecuencia Espacial (unidades relativas)')
plt.ylabel('Modulación (Contraste)')
plt.grid(True)
plt.legend()
plt.title('Curva MTF calculada vía Método del Borde')
plt.show()
```

4.3 Interpretación Clínica de la Curva MTF

Una curva MTF se lee de la siguiente manera:

- **Frecuencia 0 (DC):** Siempre es 1.0. Representa objetos infinitamente grandes donde el contraste se preserva totalmente.
- **Frecuencia de Corte (f_c):** La frecuencia donde la MTF cae a cero. Es el límite físico de resolución.
- **MTF al 50% (f_{50}):** Indica la frecuencia donde el sistema pierde la mitad de su contraste original. Es un buen indicador de la nitidez percibida.
- **MTF al 10% (f_{10}):** Se considera a menudo el límite de resolución detectable por el ojo humano en condiciones clínicas.

4.3.1 Aplicación en Mamografía vs. TC Corporal

En mamografía, donde se buscan microcalcificaciones ($\sim 100\mu m$), se requiere una MTF alta en frecuencias elevadas (ej. hasta 10-15 lp/mm). En un TC de abdomen, nos interesan frecuencias más bajas para distinguir órganos

---

5 Conclusión de las Sesiones 1 y 2

El análisis de estas sesiones demuestra que el procesamiento avanzado de imágenes médicas requiere una comprensión profunda de la física subyacente. La **cuantización** afecta la sensibilidad del contraste, mientras que el **ruido cuántico** impone límites fundamentales a la detectabilidad. El uso de métricas objetivas como **CNR** es indispensable para validar cualquier etapa posterior de segmentación o clasificación mediante IA.

6 Referencias Académicas

1. **González, R. C., & Woods, R. E.** (2002). *Digital Image Processing*. Prentice Hall.
2. **Deserno, T. M.** (2011). *Biomedical Image Processing*. Springer.
3. **Haidekker, M. A.** (2011). *Advanced Biomedical Image Analysis*. John Wiley & Sons.