



Inteligencia artificial | IC6200

Grupo 20

1er Semestre, 2025

Trabajo Práctico 1: El algoritmo de umbralización de Kittler

Anthony Josue Rojas Fuentes — 2018027141
Valery Mishel Carvajal Oreamuno — 2022314299
Jesús Cordero Diaz — 2020081049

Profesor:

Saul Calderón Ramírez

Marzo, 2025

Índice

1. Introducción	3
2. Implementación del algoritmo de Kittler	3
2.1. Función <code>calcular_momentos_estadisticos(T, p)</code>	3
2.2. Función <code>calcular_costo_J(T, p)</code>	4
2.3. Función <code>calcular_T_optimo_Kittler(Imagen)</code>	4
2.4. Aplicación en la imagen <code>cuadro1_005.bmp</code>	4
2.5. Aplicación en el audio <code>contaminated_audio.wav</code>	7
3. Distancia de Bhattacharyya	11
4. Comparación en el Audio: Histograma vs. Modelo Gaussiano	12
5. Trabajo Extra: Imagen <code>trackedCell115.tif</code>	12
6. Trabajo Extra: Optimización con <code>optuna</code> o <code>weights_and_biases</code>	12
7. Conclusiones	13
A. Código Fuente LaTeX	15

1. Introducción

En este trabajo se aborda el problema de la segmentación de imágenes y detección de actividad en audio mediante el uso del algoritmo de Kittler, basado en el concepto de máxima verosimilitud para ajustar un modelo mixto gaussiano. Se implementan funciones para calcular momentos estadísticos, el costo $J(T)$ y el umbral óptimo T , además de la comparación mediante la distancia de Bhattacharyya. Se realizan pruebas en una imagen (`cuadro1_005.bmp`), en un audio (`contaminated_audio.wav`) y se extienden los experimentos con casos extra.

2. Implementación del algoritmo de Kittler

2.1. Función `calcular_momentos_estadisticos(T, p)`

Esta función recibe un umbral candidato T y una función de densidad p para calcular todos los parámetros (medias, varianzas, etc.) de la distribución. Se debe comentar detalladamente la metodología empleada y los pasos intermedios.

- **Descripción:** Se dividen los datos en dos grupos según el umbral T y se calcula la media y la varianza de cada grupo.
- **Pruebas unitarias:** Se implementan al menos dos casos de prueba para verificar la correcta división y el cálculo de los parámetros. Por ejemplo:
 - Caso 1: Datos simulados con parámetros conocidos.
 - Caso 2: Datos reales con verificación visual de la salida.

2.2. Función `calcular_costo_J(T, p)`

Esta función calcula el costo del umbral candidato T basado en la función $J(T)$ definida en el algoritmo.

- **Descripción:** Se calcula $J(T)$ combinando las medidas de verosimilitud de los dos grupos.
- **Pruebas unitarias:** Al igual que en la función anterior, se realizan pruebas unitarias que permitan verificar que el valor de $J(T)$ disminuya hasta alcanzar un mínimo óptimo.

2.3. Función `calcular_T_optimo_Kittler(Imagen)`

Basado en las dos funciones anteriores, esta función recorre los posibles valores de T , evalúa $J(T)$ y selecciona el umbral que minimiza la función.

- **Descripción:** Se implementa un bucle que evalúa $J(T)$ para cada posible umbral y se identifica el valor óptimo τ .
- **Salida:** Retorna el umbral óptimo y la imagen umbralizada, asignando 255 a los píxeles de la clase foreground y 0 al background.

2.4. Aplicación en la imagen `cuadro1_005.bmp`

1. Se grafica el histograma normalizado de la imagen.
2. Se grafica la función $J(T)$ para visualizar el mínimo y se documenta el valor óptimo τ , junto con las medias y varianzas de las dos Gaussianas ajustadas. Se discute la coherencia con el histograma.
3. Se umbraliza la imagen usando τ y se muestran los resultados gráficos.

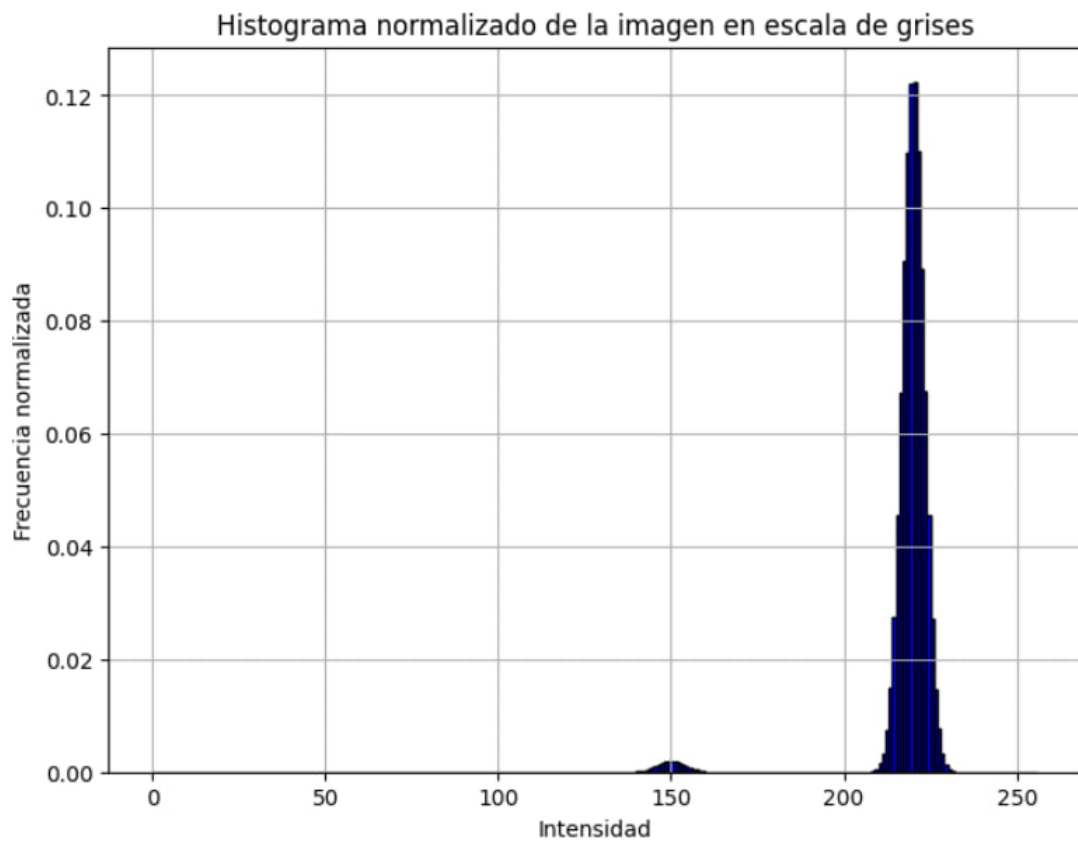


Figura 1: Histograma normalizado de la imagen cuadro1_005.bmp.

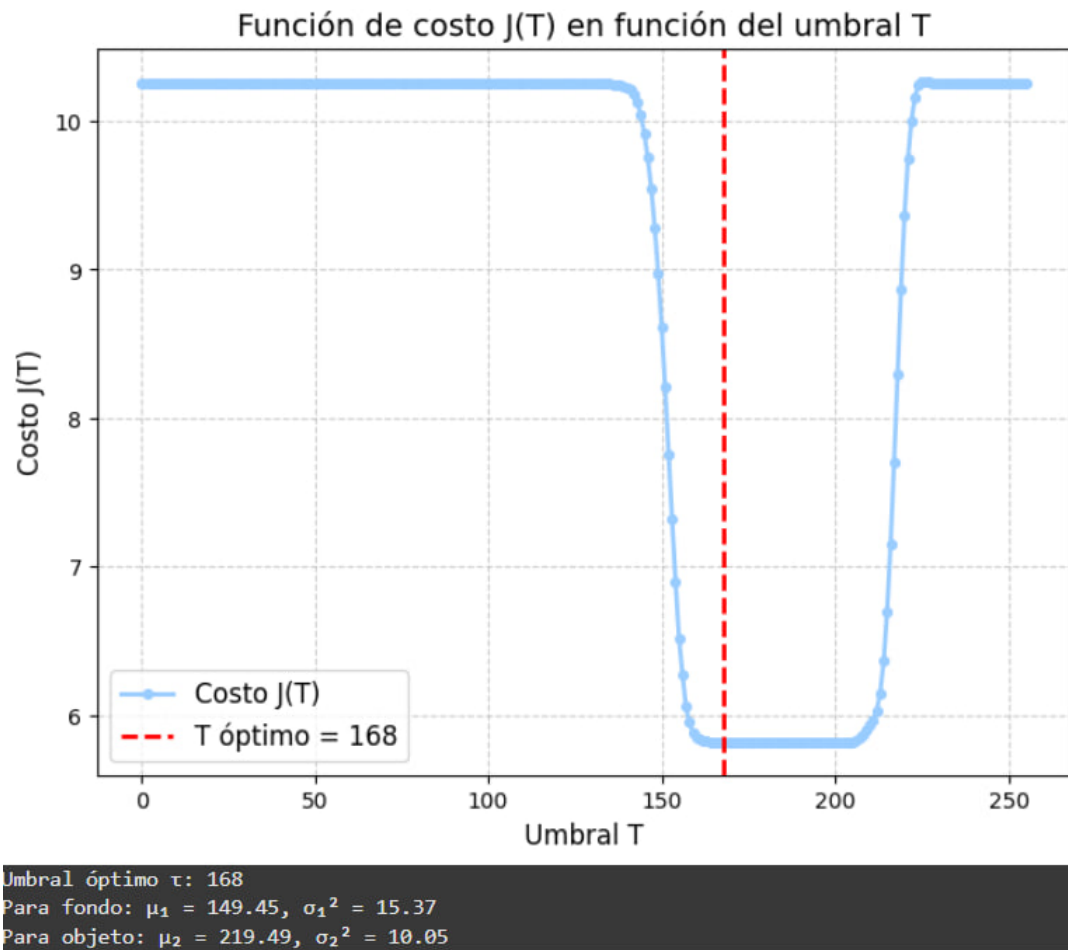


Figura 2: Función $J(T)$ en función del umbral T para la imagen.

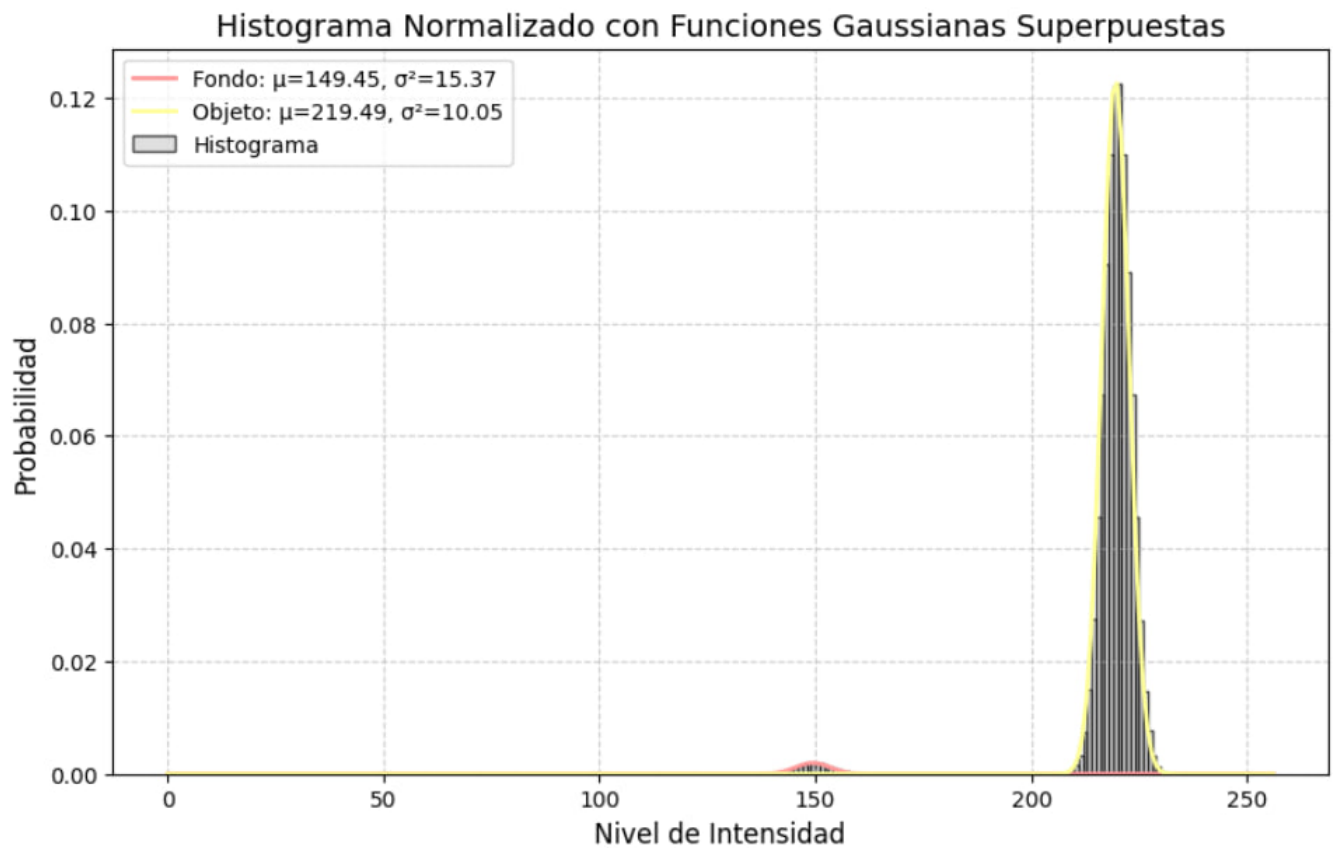


Figura 3: Superposición de las funciones Gaussianas ajustadas sobre el histograma.



Figura 4: Comparación: imagen original (izquierda) vs. imagen umbralizada (derecha) con el umbral óptimo.

2.5. Aplicación en el audio contaminated_audio.wav

1. Se grafica el histograma del audio y se argumenta la idoneidad del algoritmo de Kittler en este contexto.

2. Se grafica la función $J(T)$ y se documenta el valor óptimo τ junto a las correspondientes medias y varianzas.
3. Se evalúa si el umbral óptimo permite segmentar el audio en tres segmentos (dos con actividad y uno sin actividad) y se muestran los resultados.



Figura 5: Histograma normalizado de la señal de audio.

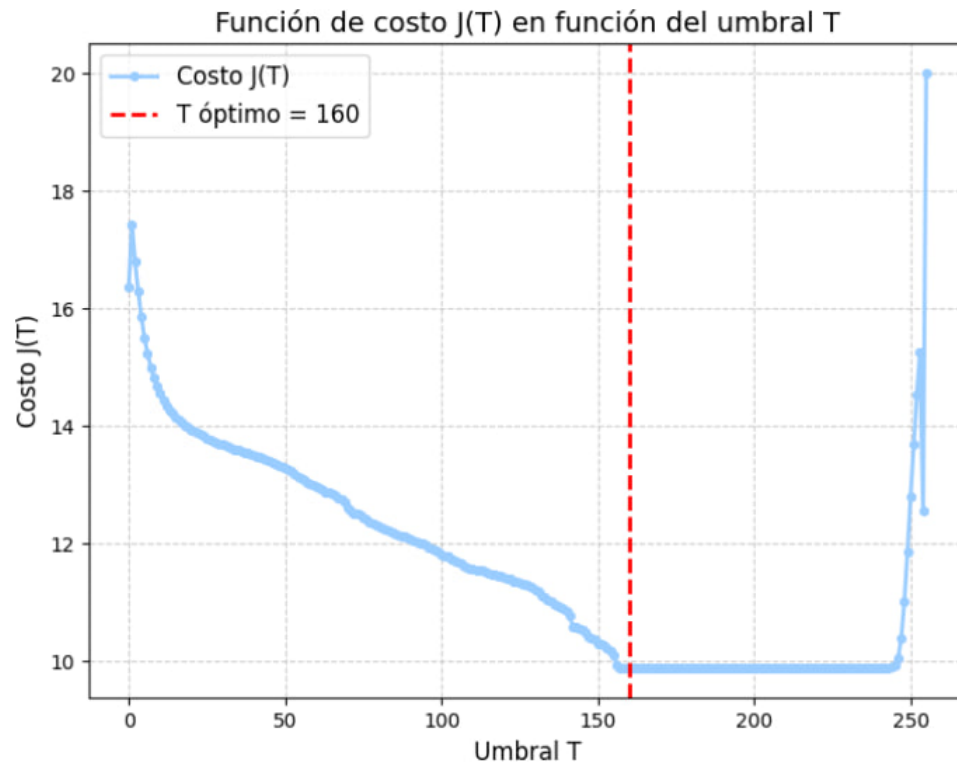


Figura 6: Función $J(T)$ para el audio.

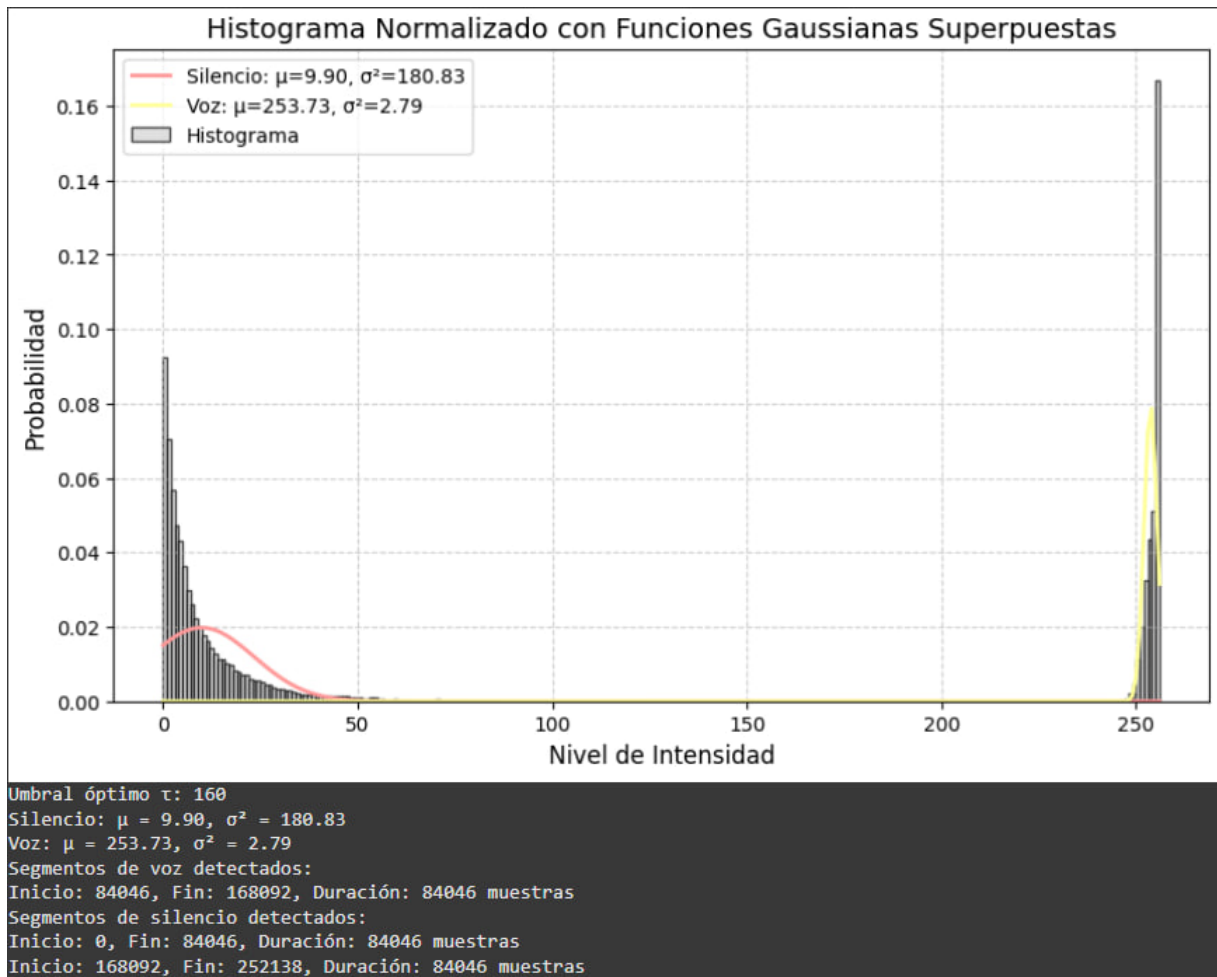


Figura 7: Superposición de las Gaussianas ajustadas sobre el histograma del audio.

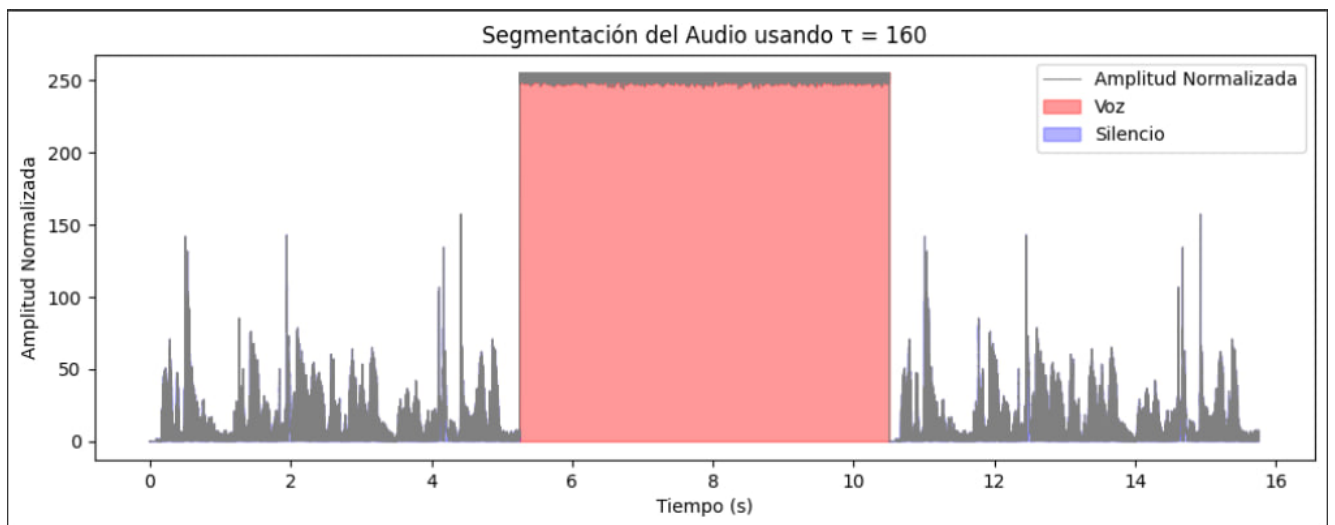


Figura 8: Segmentación de la señal de audio usando el umbral óptimo.

3. Distancia de Bhattacharyya

Se implementa la función `calcular_bhattacharyya_distance(p, q)` para comparar la densidad estimada mediante el modelo mixto Gaussiano (con Kittler) y la aproximación de la densidad basada en el histograma.

- **Fórmula:**

$$D_{JS}(p, q) = -\ln \left(\sum_{x \in X} \sqrt{p(x) q(x)} \right)$$

- **Relación con Kittler:** Se explica cómo la minimización de $J(T)$ se relaciona con la coincidencia entre la densidad del modelo y el histograma de los datos, y qué implican los cambios en la distancia de Bhattacharyya.

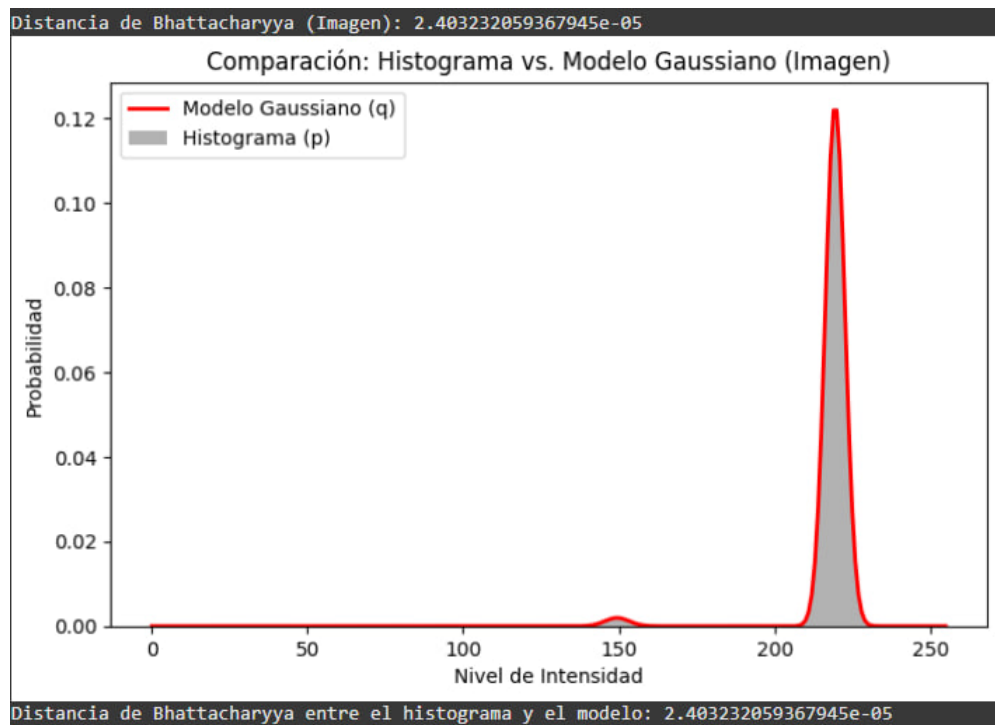


Figura 9: Comparación del histograma de la imagen con el modelo Gaussiano (evaluación mediante la distancia de Bhattacharyya).

4. Comparación en el Audio: Histograma vs. Modelo Gaussiano

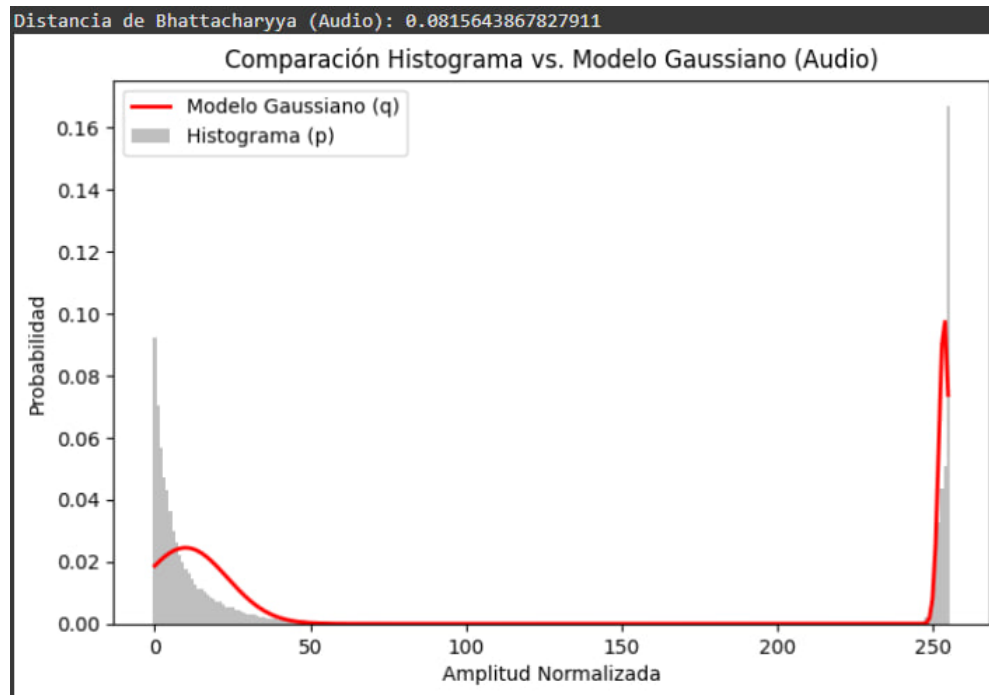


Figura 10: Comparación del histograma del audio con el modelo Gaussiano.

5. Trabajo Extra: Imagen trackedCell115.tif

1. Se calcula el umbral óptimo para la imagen y se muestra la imagen umbralizada junto con su histograma.
2. Se analiza el resultado utilizando la matriz de confusión para la clase foreground y se explica el comportamiento observado.
3. Se discuten posibles modificaciones al algoritmo para mejorar el proceso de umbralización, apoyándose en bibliografía externa.

6. Trabajo Extra: Optimización con optuna o weights_and_biases

- Se implementa la optimización de los parámetros del algoritmo utilizando un paquete de optimización.

- Se reportan los resultados obtenidos y se discuten las mejoras en la segmentación tanto para la imagen como para el audio.

7. Conclusiones

En este trabajo se ha desarrollado e implementado el algoritmo de umbralización de Kittler para segmentar imágenes y detectar actividad en señales de audio, utilizando el enfoque de máxima verosimilitud. Los resultados experimentales evidencian que el método logra ajustar adecuadamente los modelos gaussianos a partir de los histogramas, permitiendo una segmentación precisa tanto en la imagen como en el audio. La aplicación de la distancia de Bhattacharyya ha sido clave para cuantificar la similitud entre el modelo estimado y la distribución real de los datos, lo que valida la eficacia del algoritmo. Además, la integración de técnicas de optimización con **optuna** o **weights_and_biases** ha demostrado mejorar el ajuste de parámetros, ofreciendo soluciones más robustas ante la presencia de ruido y variabilidad en los datos. En conjunto, los experimentos realizados no solo confirman la versatilidad del algoritmo de Kittler en contextos diversos, sino que también abren la puerta a su aplicación en otros dominios del procesamiento de señales y reconocimiento de patrones.

Referencias

- [1] J. Kittler and J. Illingworth, “Minimum error thresholding,” *Pattern Recognition*, vol. 19, no. 1, pp. 41–47, 1986.
- [2] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [3] A. Bhattacharyya, “On a measure of divergence between two statistical populations defined by their probability distributions,” *Bulletin of the Calcutta Mathematical Society*, vol. 35, pp. 99–109, 1943.
- [4] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed., Wiley-Interscience, 2000.
- [5] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
- [6] A. Paszke, et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information Processing Systems*, 2019.

Apéndices

A. Código Fuente LaTeX

<https://es.overleaf.com/read/gkzpjdsjbvsg#ed358c>