



Modelo de clasificación para productos no conformes de espumas florales a través de imágenes.

Yuly Nataly Leudo Gamboa
Sthefany Villarraga Diaz

Universidad del Valle
Facultad de Ingeniería
Escuela de Estadística
Cali, Colombia
2025

Modelo de clasificación para productos no conformes de espumas florales a través de imágenes.

Yuly Nataly Leudo Gamboa
Sthefany Villarraga Diaz

Trabajo de grado presentado como requisito parcial para optar al título de:
Estadística

Director: Álvaro José Flórez, Estadístico, MSc. en estadística, PhD en estadística
Co-director: Juan José Arias, Ingeniero de Materiales, MSc. en biotecnología

Universidad del Valle
Facultad de Ingeniería
Escuela de Estadística
Cali, Colombia
2025

Agradecimientos

No tengo palabras suficientes para agradecer a mi familia por apoyarme en mi decisión de retomar los estudios; a Fernando y a su mamá, Nhora, por creer en mí desde el principio; a mis amigos de la carrera, Luisa, Laura, Esteban, William y Mariet, por estar siempre ahí entre trabajos, evaluaciones y exposiciones, brindándome su apoyo incluso en los momentos más difíciles de mi vida; a mis profesores por enseñarme esta bonita profesión, en especial al profesor Jaime y a la profesora Jennyfer, quienes más que docentes me inspiraron con sus consejos; y a mi compañera de trabajo de grado y amiga Sthefany, por acompañarme en todo momento y dar siempre lo mejor de sí incluso cuando yo no podía. No puedo dejar de agradecer también a todas las personas que no mencioné, pero que con su granito de arena hicieron posible este logro, ¡los quiero mucho!

Yuly Nataly Leudo

Agradezco profundamente a mis amigos y compañeros por su apoyo, colaboración y compañía a lo largo de este proceso. A mis profesores, por su dedicación, orientación y valiosos conocimientos, así como por el interés demostrado durante mi formación y en el desarrollo de este trabajo. Y a mis familiares, por su constante respaldo, motivación y confianza, que fueron fundamentales para culminar con éxito esta etapa.

Sthefany Villarraga Diaz

Dedicatoria

*Para el amor más grande de mi vida: **mi mamá Nhora**. Soñamos este momento juntas y, aunque ya no estás conmigo, hoy hago este sueño realidad por las dos. Tu lucha, tu valentía y tu amor siguen guiando cada paso que doy. Este logro es también la respuesta de Dios a todas las oraciones que hiciste por mí para que este sueño se cumpliera.*

*A mi hermana **Carolina**, cuyo amor y apoyo han sido fundamentales en mi vida. Gracias por tu paciencia, por tu fe inquebrantable y por acompañarme incluso en los momentos más difíciles. Gracias por creer en mí cuando yo misma dudaba. **Este logro también te pertenece.***

Yuly Nataly Leudo

*A mis abuelos, **Martha y Alberto**, por su apoyo incondicional, sus consejos y todos los esfuerzos que han realizado para acompañarme en este camino. Su confianza y cariño han sido un pilar fundamental en mi formación personal y académica.*

*A mi mamá, **Yurani**, por su dedicación, fortaleza y compromiso inquebrantable. Su esfuerzo constante y su ejemplo han sido mi mayor motivación y fuente de inspiración para alcanzar cada meta. Este logro refleja su entrega y el inmenso valor de su apoyo en cada etapa de mi vida.*

Sthefany Villarraga Diaz

Resumen

Este trabajo propone un enfoque inicial para la detección automatizada de productos no conformes (productos que no cumplen con ciertos estándares de calidad, es decir, aquellos que presentan defectos) en una empresa productora de espumas fenólicas. Actualmente, la clasificación se realiza mediante inspección visual manual, la cual es susceptible a sesgos subjetivos y errores. Por ello, el objetivo de este estudio es desarrollar un modelo de clasificación supervisada basado en imágenes que permita identificar defectos en las espumas producidas.

Se realizó un diseño experimental para la adquisición de datos, mediante la captura de imágenes de 380 piezas de espuma fenólica, registrando dos vistas por pieza (lado A y lado B), lo que resultó en un total de 760 imágenes. Estas corresponden a cuatro clases: una categoría conforme, denominada espuma buena, y tres categorías no conformes: espuma con beta cristalizada, espuma con beta escurrida y espuma con poros. A partir de estas imágenes, se emplearon técnicas de extracción de características como la transformada discreta de wavelet y patrones locales binarios, junto con selección de características mediante Eliminación Recursiva de Características (RFE). Posteriormente, se entrenaron dos modelos de clasificación: máquinas de soporte vectorial y regresión logística multinomial, validados mediante validación cruzada.

Los resultados indican que las características extraídas fueron relevantes para discriminar entre las clases, destacándose el modelo de máquinas de soporte vectorial por su mejor desempeño global en la clasificación multiclase. En ambos modelos, la clase mejor identificada fue la de espuma conforme, que se refiere a la espuma que cumple con los estándares de calidad al no presentar ningún defecto visible. Se concluye que este enfoque constituye un primer avance hacia la automatización del proceso de inspección, y puede ser complementado con técnicas más avanzadas de visión por computador en futuros desarrollos.

Palabras clave: control de calidad, clasificación supervisada, transformada discreta de wavelet, patrones locales binarios, máquinas de soporte vectorial, regresión logística multinomial.

Abstract

This paper proposes an initial approach for the automated detection of non-compliant products (products that do not meet certain quality standards, i.e., those with defects) in a phenolic foam manufacturing company. Currently, classification is performed by manual visual inspection, which is susceptible to subjective bias and errors. Therefore, the objective of this study is to develop an image-based supervised classification model that allows defects in the produced foams to be identified.

An experimental design was carried out for data acquisition, capturing images of 380 pieces of phenolic foam, recording two views per piece (side A and side B), resulting in a total of 760 images. These correspond to four classes: one compliant category, called good foam, and three non-compliant categories: foam with crystallized beta, foam with drained beta, and foam with pores. From these images, feature extraction techniques such as discrete wavelet transform and local binary patterns were used, along with feature selection using Recursive Feature Elimination (RFE). Subsequently, two classification models were trained: support vector machines and multinomial logistic regression, validated by cross-validation.

The results indicate that the extracted features were relevant for discriminating between classes, with the support vector machine model standing out for its better overall performance in multi-class classification. In both models, the best-identified class was compliant foam, which refers to foam that meets quality standards by having no visible defects. It is concluded that this approach constitutes a first step toward automating the inspection process and can be complemented with more advanced computer vision techniques in future developments.

Keywords: quality control, supervised classification, discrete wavelet transform, local binary patterns, support vector machines, multinomial logistic regression.

Contenido

Resumen	v
Abstract	vi
1. Introducción y planteamiento del problema	5
1.1. Introducción	5
1.2. Planteamiento del Problema	7
1.2.1. Pregunta de Investigación	8
1.3. Justificación	9
1.4. Objetivos	10
1.4.1. Objetivo General:	10
1.4.2. Objetivos específicos	10
2. Antecedentes	11
3. Marco teórico	14
3.1. Espumas Florales	14
3.2. Preprocesamiento de imágenes	15
3.2.1. Imagen digital - Mapa de bits	16
3.2.2. Rango dinámico de la imagen	17
3.2.3. Análisis multiresolución	17
3.2.4. Filtro gaussiano	18
3.2.5. Análisis de Textura	19
3.3. Transformada Discreta de Wavelet (DWT)	19
3.3.1. Sistema de expansión wavelet	19
3.3.2. Wavelet de Haar	21
3.3.3. Proceso de descomposición	22

3.4. Local Binary Patterns (LBP)	24
3.4.1. Definición de LBP	24
3.4.2. Histograma de LBP	25
3.4.3. LBP Invariante a la Rotación	25
3.4.4. LBP y Patrones Uniformes	25
3.5. Máquinas de Soporte Vectorial (SVM)	26
3.5.1. Definición de Hiperplano	26
3.5.2. Clasificador de Margen Máximo	26
3.5.3. Clasificador de Soporte Vectorial	27
3.5.4. Extensión a Problemas No Lineales	27
3.6. Regresión Logística	28
3.6.1. Regresión logística multinomial (RLM)	29
3.7. Validación Cruzada	31
3.8. Recursive Feature Elimination (RFE)	32
3.8.1. Procedimiento General de RFE	32
3.9. Métricas de evaluación	33
3.9.1. Matriz de confusión	34
3.9.2. Métricas derivadas	35
4. Metodología	37
4.1. Captura de imágenes	38
4.1.1. Equipo Utilizado	39
4.1.2. Proceso de almacenamiento	40
4.2. Carga de imágenes	41
4.3. Preprocesamiento inicial	42
4.4. División del conjunto de datos	49
4.5. Validación cruzada k-fold	49
4.5.1. Preprocesamiento	50
4.5.2. Extracción de características	52
4.5.3. Selección de características	59
4.5.4. Selección de características e hiperparámetros - Máquina de Soporte Vectorial (SVM)	60
4.5.5. Selección de características e hiperparámetro C' - Regresión Logística Multinomial (RLM)	61
4.5.6. Entrenamiento de modelos para validación cruzada	62

4.5.7. Métricas de comparación	64
4.6. Entrenamiento final	66
4.6.1. Entrenamiento final de los modelos	66
4.6.2. Evaluación y comparación de resultados	66
5. Resultados	68
5.1. Resultados de los fold en la Validación Cruzada	68
5.1.1. Selección de características e hiperparámetros - SVM	68
5.1.2. Selección de características e hiperparámetro C - RLM	73
5.2. Resultado del entrenamiento final	76
6. Conclusiones	81
7. Recomendaciones	83
A. Detalles del proceso de captura de imágenes	85
B. Resultados detallados de las combinaciones evaluadas en los folds para SVM	89
C. Resultados detallados de las combinaciones evaluadas en los folds para RLM	96

Índice de Tablas

3-1.	Matriz de confusión para clasificación binaria	34
4-1.	Organización de las imágenes capturadas	41
4-2.	Nombre de las columnas del archivo CSV	41
4-3.	Número de características extraídas por DWT para clasificación	56
4-4.	Matriz de Confusión Multiclase para B (Espuma Buena)	65
5-1.	Resultados de la selección de características e hiperparámetros evaluados según los macro-promedios de las métricas en validaciones internas (Int.) y externas (Ext.).	69
5-2.	Subconjunto de 15 Características seleccionadas para cada Fold	72
5-3.	Características finales seleccionadas según su frecuencia y ranking promedio en los <i>folds</i>	73
5-4.	Resultados de la selección de características e hiperparámetro evaluados según los macro-promedios de las métricas en validaciones internas (Int.) y externas (Ext.).	74
5-5.	Características finales seleccionadas según su frecuencia y ranking promedio en los <i>folds</i> (modelo de regresión logística multinomial).	76
5-6.	Métricas finales de los modelos SVM y Regresión Logística Multinomial. Las celdas en verde indican las métricas con mejor desempeño, en rojo las de menor rendimiento y en gris aquellas con valores muy similares.	78
B-1.	Resultados completos (macro-promedios) para todas las combinaciones de número de características e hiperparámetros en SVM. Validaciones internas (Int.) y externas (Ext.).	89
C-1.	Resultados completos (macro-promedios) para todas las combinaciones de número de características e hiperparámetros. Validaciones internas (Int.) y externas (Ext.).	96

Índice de Figuras

3-1. Wavelet de Haar: función escala y función wavelet	22
3-2. Esquema del proceso de descomposición DWT en una imagen 2D	23
4-1. Metodología desarrollada	37
4-2. Imágenes según las diferentes clases (Lado A de la imagen)	39
4-3. Equipos utilizados en la captura de imagen.	40
4-4. Imagen RGB y canal HSV de una espuma (Lado A y Lado B). La columna izquier- da muestra la imagen original y los canales H, S y V para el lado A, mientras que la columna derecha muestra la misma información para el lado B.	44
4-5. Muestreo central para estimar el color de referencia en las imágenes.	45
4-6. Generación de la máscara binaria y su superposición sobre la imagen original para delimitar el objeto de interés (Lado A y Lado B).	46
4-7. Detección del contorno externo del objeto en las imágenes (Lado A y Lado B). . . .	47
4-8. Rectángulo delimitador calculado sobre la imagen original, antes de aplicar el re- corte final (Lado A y Lado B).	47
4-9. Imagen original y resultado del recorte (Lado A y Lado B) después del proceso de detección y delimitación.	48
4-10. Imágenes de las espumas antes y después del preprocesamiento (Lado A y Lado B). .	51
4-11. DWT - Primer Nivel (Subbandas LL, LH, HL, HH) para los lados A y B.	53
4-12. DWT - Segundo Nivel (Subbandas LL, LH, HL, HH) para los lados A y B.	54
4-13. Ejemplo del cálculo del código LBP ($P=8$, $R=1$): a la izquierda, intensidades en una vecindad 3×3 con píxel central $I_c=90$; a la derecha, binarización de vecinos (1 si $I_p \geq I_c$, 0 en otro caso) y lectura de bits en sentido horario. El patrón binario se convierte a entero mediante $\sum_{p=0}^7 b_p 2^p$ (código LBP = 19).	57

4-14. Comparación A vs B. Fila superior: imágenes originales. Fila intermedia: mapas LBP con $(P, R) = (8, 1)$, operador clásico. Fila inferior: histogramas normalizados de 256 bins (frecuencia relativa) para cada imagen; misma escala en el eje y para facilitar la comparación.	58
5-1. Curvas comparativas por número de características según los macro-promedio de las métricas.	71
5-2. Comparación de matrices de confusión de los modelos finales	77
A-1. Equipos utilizados en la captura de imagen.	85
A-2. Configuración de la cámara	86
A-3. Iluminación y posicionamiento	88

1. Introducción y planteamiento del problema

1.1 Introducción

El control de calidad es un aspecto fundamental en los procesos industriales, pues garantiza que los productos cumplan con los estándares establecidos para su correcto funcionamiento. Para ello, se emplean diversos métodos de evaluación que no comprometen la integridad del producto, entre los cuales destacan los ensayos no destructivos, como la inspección visual (Czimmermann et al., 2020). Este método es ampliamente utilizado para la detección de defectos superficiales; sin embargo, al depender de la percepción del operario, sus resultados suelen ser variables y subjetivos, lo que dificulta su estandarización y control. En consecuencia, se ha incrementado el interés por desarrollar sistemas automatizados de inspección que reduzcan la variabilidad humana y mejoren la reproducibilidad del proceso (Czimmermann et al., 2020).

En la producción de espumas fenólicas para arreglos florales, la inspección visual manual continúa siendo el principal método para la detección de defectos. Una empresa del sector realiza la clasificación apoyándose en operarios que identifican distintas anomalías superficiales asociadas a la no conformidad del producto. No obstante, este procedimiento presenta inconsistencias derivadas de la subjetividad del evaluador, la fatiga y las condiciones del entorno, lo que evidencia la necesidad de herramientas automáticas que estandaricen la inspección, reduzcan errores y proporcionen información confiable para la mejora continua del proceso.

Frente a estas limitaciones, este trabajo propone una metodología basada en el análisis digital de imágenes para la identificación automática de defectos visuales en espumas florales. Esta propuesta constituye una primera aproximación al control de calidad automatizado en la empresa, mediante el uso de técnicas de extracción de características y modelos de clasificación supervisada. Con ello se busca apoyar la toma de decisiones en producción, reducir los costos asociados a clasificaciones incorrectas y fortalecer la consistencia en la calidad del producto final (Coşkun et al., 2022; Czim-

mermann et al., 2020; Nazir et al., 2021).

A diferencia de enfoques más recientes basados en aprendizaje profundo, la metodología propuesta se centra en la extracción y análisis estadístico de características visuales mediante técnicas de procesamiento digital de imágenes (Gonzalez & Woods, 2018). Este enfoque permite evaluar su viabilidad dentro del proceso de inspección actual y servir como punto de partida para la incorporación de tecnologías más avanzadas en etapas posteriores.

El estudio se desarrolla en un entorno controlado de laboratorio y se limita a la clasificación automática de defectos visuales presentes en las panelas de espuma floral. No se abordan otros tipos de defectos ni técnicas más complejas, como las basadas en redes neuronales profundas, las cuales se contemplan como líneas de investigación futura.

El documento se organiza de la siguiente manera: el Capítulo 1 incluye el planteamiento del problema, la justificación y los objetivos del estudio. El Capítulo 2 presenta los antecedentes relacionados con la inspección visual automática en productos industriales. En el Capítulo 3 se expone el marco teórico sobre espumas fenólicas, procesamiento digital de imágenes, extracción de características y modelos de clasificación supervisada. El Capítulo 4 describe la metodología propuesta, desde la captura y procesamiento de imágenes hasta el entrenamiento de los modelos. En el Capítulo 5 se presentan los resultados obtenidos, mientras que los Capítulos 6 y 7 contienen las conclusiones y recomendaciones.

1.2 Planteamiento del Problema

Una empresa con sede en Cali, Colombia, que fabrica y comercializa espumas florales, espumas agrícolas y absorbentes industriales, ha logrado una posición de liderazgo en el mercado gracias a su capacidad para desarrollar espumas basadas en tecnología fenólica. Su enfoque principal está en la producción de espumas florales utilizadas en la creación de arreglos y decoraciones florales. Estas espumas tienen características físicas claves, como dureza, tiempo de retención, porcentaje de retención y resistencia a ruptura, que son fundamentales para cumplir con los estándares de calidad requeridos.

El proceso de control de calidad en la empresa se realiza de manera manual, a través de pruebas físicas y visuales. Si bien este enfoque ha sido tradicionalmente utilizado, presenta un margen de error significativo, lo que afecta la precisión y la consistencia del proceso. En este caso, dicho margen de error fue determinado en aproximadamente un **2 %**, valor obtenido a partir de experimentos realizados hace dos años, donde se revisaron seis estibas de productos no conformes (NC), cada una con cerca de 5000 panelas, para identificar falsos negativos (productos defectuosos clasificados como conformes), lo que implica que aproximadamente 1056 panelas de las 52800 producidas diariamente pueden ser clasificadas incorrectamente.

Este error de clasificación se distribuye de dos maneras:

1. **Clasificación de productos conformes como defectuosos:** genera un desperdicio de recursos, pues productos aptos son rechazados o sometidos a reprocesamiento innecesario, incrementando los costos de producción y reduciendo la eficiencia operativa.
2. **Clasificación de productos defectuosos como conformes:** permite que productos defectuosos lleguen al mercado, afectando la satisfacción del cliente y dañando la reputación de la empresa, lo que se traduce en pérdidas económicas por devoluciones y reclamos.

Adicionalmente, la empresa enfrenta un desafío relacionado con el porcentaje real de productos defectuosos. Este se calcula diariamente dividiendo el número de productos defectuosos entre el total de panelas producidas, correspondiendo al **5,5 %**. Esto equivale a aproximadamente 2904 panelas defectuosas al día. La alta cantidad de productos defectuosos, junto con los errores de clasificación mencionados, contribuyen a una disminución de la calidad y consistencia del producto final.

El proceso de control de calidad manual está sujeto a factores como fatiga, presión de tiempo, falta de estandarización en los criterios de evaluación y subjetividad en la interpretación de los resultados. Estas condiciones incrementan la variabilidad y la probabilidad de cometer errores, lo cual afecta la fiabilidad del sistema de control de calidad y dificulta la alineación con los estándares de calidad requeridos. Esta situación también limita la capacidad de la empresa para realizar ajustes rápidos en la producción y responder eficazmente a la demanda del mercado.

En este contexto, es fundamental contar con un sistema confiable y automatizado que no solo permita identificar y clasificar correctamente los defectos, sino también proporcione una base de datos estructurada sobre las no conformidades. Estos defectos pueden presentarse de diversas formas, siendo los más comunes el defecto del caucho, asociado a la compactación o aplastamiento de la estructura celular de la espuma, lo que provoca la formación de zonas laminadas de mayor resistencia mecánica (este defecto no fue considerado en el presente trabajo, ya que se manifiesta principalmente como una variación de color y no como un patrón de textura, mientras que el análisis realizado se centró exclusivamente en características texturales); la beta escurrida, que genera líneas o marcas superficiales por un vertido tardío de la mezcla; la beta cristalizada, que altera la estructura interna del material debido a reacciones químicas localizadas; y los poros, que corresponden a cavidades o vacíos generados por aire atrapado durante la mezcla. Actualmente, la empresa no cuenta con información confiable sobre los tipos de defectos presentes en cada panela, lo que dificulta la trazabilidad del proceso de producción y la optimización del control de calidad.

La automatización del proceso de clasificación de defectos mediante un modelo de clasificación basado en imágenes puede ser una solución efectiva para reducir tanto el margen de error del 2 % en la clasificación como para asegurar que los productos defectuosos, que representan el 5,5 % de la producción, sean correctamente identificados. Esto no solo optimizará la eficiencia operativa, sino que también permitirá que la empresa tome decisiones informadas y oportunas para mejorar la calidad y consistencia del producto final, asegurando una mayor satisfacción del cliente y reduciendo pérdidas económicas.

1.2.1 Pregunta de Investigación

¿Cómo desarrollar un modelo de clasificación, basado en características obtenidas a partir de imágenes, que permita identificar productos no conformes en la línea de producción de espuma floral en la empresa?

1.3 Justificación

El control de calidad constituye un proceso fundamental dentro de la cadena productiva de la empresa, dado que garantiza que las panelas de espuma floral cumplan con los estándares establecidos para su comercialización. Sin embargo, el sistema actual, basado en inspección manual, presenta limitaciones que afectan la precisión y la consistencia de la clasificación, debido a la subjetividad del operario, la variabilidad en los criterios de evaluación y la ausencia de mecanismos sólidos de trazabilidad. Estas limitaciones incrementan el margen de error y dificultan la detección oportuna de defectos, lo que justifica la incorporación de procesos automatizados basados en imágenes. La automatización permitiría reemplazar evaluaciones subjetivas por procedimientos objetivos y repetibles, fortaleciendo la capacidad de detección de defectos visuales y garantizando un control de calidad más eficiente y confiable (Czimmermann et al., 2020).

En este trabajo se desarrolla un modelo de clasificación multiclase basado en imágenes, diseñado para identificar de manera automática los diferentes defectos visuales presentes en las panelas de espuma fenólica. El enfoque propuesto utiliza características extraídas de las imágenes para representar los patrones visuales de cada unidad y entrenar los algoritmos de clasificación. Con el propósito de determinar la alternativa más adecuada para esta tarea, se comparan dos métodos de aprendizaje supervisado ampliamente utilizados en aplicaciones industriales: la Regresión Logística Multinomial (RLM) y las Máquinas de Soporte Vectorial (SVM), evaluando su desempeño sobre un mismo conjunto de datos (imágenes).

La evaluación de los modelos se fundamenta en métricas que permiten analizar su capacidad para diferenciar entre los diversos tipos de defectos presentes en las panelas. En particular, se consideran la sensibilidad, que mide la capacidad del modelo para identificar correctamente las unidades defectuosas, y la especificidad, que refleja su habilidad para reconocer adecuadamente las unidades en buen estado (Rainio et al., 2024). Asimismo, se emplean métricas globales como la exactitud (accuracy), que indica la proporción total de clasificaciones correctas, y el F1-score, que combina precisión y sensibilidad para evaluar el rendimiento ante la presencia de falsos positivos y falsos negativos (Coşkun et al., 2022; Rainio et al., 2024). Debido a que el problema es de naturaleza multiclase, estas métricas se adaptan mediante promedios macro, lo que permite realizar comparaciones equitativas entre todas las categorías consideradas (Rainio et al., 2024). En conjunto, estas métricas ofrecen una evaluación integral del desempeño de cada modelo dentro de un sistema automatizado de inspección de calidad.

La elección de la RLM y la SVM responde a su amplia aplicación en problemas de clasificación industrial. La RLM constituye un modelo estadístico interpretable y adecuado para modelar relaciones lineales entre las variables explicativas y el predictor lineal (logaritmo de las probabilidades o *log-odds*) de cada categoría de la respuesta, en el marco de los modelos lineales generalizados (Bishop, 2006). Además, en aplicaciones industriales se ha mostrado como una herramienta eficiente para problemas de clasificación multiclase, alcanzando altos niveles de precisión e identificando las condiciones de operación más influyentes en la aparición de defectos (Vaz et al., 2021), mientras que la SVM proporciona mayor flexibilidad para capturar relaciones no lineales mediante el uso de funciones *kernel*, lo que la hace especialmente adecuada para representar la complejidad visual asociada a los defectos (Coşkun et al., 2022; Nazir et al., 2021). La comparación de sus resultados, basada en las métricas mencionadas, permitirá identificar el método con mejor equilibrio entre precisión, capacidad de generalización y viabilidad operativa, contribuyendo directamente a la automatización del control de calidad en la empresa.

1.4 Objetivos

1.4.1 Objetivo General:

Desarrollar un modelo de clasificación basado en características obtenidas a partir de imágenes, que permita identificar productos no conformes en la línea de producción de espuma floral en la empresa.

1.4.2 Objetivos específicos

- Capturar imágenes de las espumas florales (panelas) bajo condiciones controladas, garantizando la calidad visual de los datos.
- Extraer características de las imágenes que permitan obtener información relevante para el proceso de clasificación.
- Entrenar modelos de clasificación que utilizan la información extraída de las imágenes para identificar las no conformidades en la espuma floral.
- Evaluar el desempeño de los modelos de clasificación, determinando su capacidad para discriminar entre productos conformes y no conformes.

2. Antecedentes

La detección y clasificación de defectos en productos industriales ha sido un tema recurrente en la investigación debido a la necesidad de garantizar la calidad de los productos y mejorar la eficiencia de los procesos productivos (Czimmermann et al., 2020). Diversos estudios han explorado el uso de técnicas avanzadas de procesamiento de imágenes y aprendizaje automático para abordar esta problemática, sentando las bases para el desarrollo de este trabajo de grado.

Coşkun et al. (2022) desarrollaron un sistema de visión artificial para clasificar defectos en baldosas cerámicas, utilizando características geométricas y transformadas wavelet junto con máquinas de soporte vectorial (SVM). Este sistema logró identificar defectos como manchas, grietas y poros con una precisión del 100 % en algunas clases, destacando la eficacia de las wavelets para extraer características relevantes de las imágenes. La implementación del sistema se llevó a cabo tanto en laboratorio como en líneas de producción reales, demostrando que los métodos automáticos pueden superar las limitaciones de la inspección manual, que es más propensa a errores y menos consistente. Este enfoque resalta la importancia de integrar herramientas avanzadas para optimizar procesos industriales, una idea central para este trabajo, donde se busca clasificar defectos en espumas florales.

De manera complementaria, Ghazali et al. (2007) aplicaron la transformada discreta de wavelet bidimensional (2D-DWT) de tipo Haar para clasificar imágenes de malezas estrechas y anchas obtenidas en cultivos de palmas. Las imágenes fueron convertidas a escala de grises y descompuestas en sub-bandas mediante la 2D-DWT, lo que permitió representar la información en términos de escala y frecuencia. Además, se utilizó la técnica *Line Measure Technique* (LMT) para medir la continuidad de los valores de los píxeles en las sub-bandas bajo diferentes ángulos y longitudes. Esto simplificó los datos de entrada al clasificador, lo que permitió que las clases fueran linealmente separables, alcanzando precisiones del 86 % y 88 % para malezas estrechas y anchas, respectivamente. Este estudio evidencia que las transformadas wavelet no solo son útiles para extraer

características relevantes, sino también para simplificar los datos, lo que facilita el uso de clasificadores como las SVM en tareas de clasificación complejas.

Por su parte, Nazir et al. (2021) analizaron la clasificación de imágenes textiles con patrones de batik, caracterizados por su variabilidad en términos de textura y forma. Las imágenes fueron pre-procesadas mediante la detección de bordes con el método de Canny y la aplicación de la 2D-DWT, el patrón binario local (LBP) y los momentos invariantes (IM). Estas técnicas se combinaron para formar descriptores robustos que fueron posteriormente reducidos dimensionalmente con análisis de componentes principales (PCA). El uso de SVM como clasificador permitió alcanzar precisiones de entre el 93 % y el 97 %, mientras que sin la reducción dimensional, la precisión fue cercana al 80 %. Este estudio subraya la importancia del preprocesamiento de imágenes y la selección de características relevantes para mejorar el rendimiento de los clasificadores.

En una línea similar, Yousif et al. (2022) exploraron la clasificación de tumores cerebrales utilizando imágenes de resonancia magnética, aplicando un enfoque que combina la 2D-DWT y el análisis de componentes principales (PCA). En este trabajo, las señales de las imágenes fueron descompuestas en componentes de baja y alta frecuencia, utilizando solo los coeficientes de baja frecuencia, ya que contienen la mayor parte de la información relevante. La dimensionalidad de las características fue reducida con PCA, y se empleó un clasificador *Random Forest* (RF) configurado con 50 árboles y validación cruzada de 11 pliegues. Este modelo alcanzó una precisión del 96.83 %, demostrando que la combinación de wavelets y PCA puede mejorar significativamente la eficiencia y precisión de los modelos de clasificación, incluso en problemas multiclase. Estos hallazgos inspiran la posibilidad de aplicar un esquema similar para optimizar la clasificación de defectos en espumas florales.

En un contexto industrial más cercano al de este trabajo, Vaz et al. (2021) desarrollaron un modelo de regresión logística multinomial para predecir la aparición de defectos de burbujas de aire en un proceso de unión húmeda entre un cristal y una pantalla electrónica. La variable de respuesta se definió con cuatro categorías (producto sin burbujas, burbuja en la zona 1, burbuja en la zona A y burbuja en la zona D de la pantalla), mientras que las variables predictoras incluyeron 16 medidas de condiciones operativas del proceso, tales como errores en las estaciones de trabajo AOI y FLC, distancias y desplazamientos del robot. A partir de una muestra de 659 unidades ensambladas, el modelo alcanzó una precisión del 82.8 % en la clasificación de los escenarios de defecto, identificando como factores más influyentes la ocurrencia de errores en AOI y FLC, la distancia a la pantalla, la alineación en el eje R y el desplazamiento OffsetP4. Este antecedente evidencia que la

regresión logística multinomial puede emplearse de manera efectiva para modelar la probabilidad de diferentes tipos de defectos en procesos industriales reales, lo que respalda su potencial aplicación en la clasificación de defectos en espumas florales.

La selección de características es una etapa fundamental en el desarrollo de modelos de aprendizaje automático, ya que permite reducir la dimensionalidad de los datos, eliminar redundancias y conservar únicamente aquellas variables que aportan mayor relevancia predictiva. Entre las técnicas más utilizadas se encuentra el *Recursive Feature Elimination* (RFE), que selecciona iterativamente los predictores más significativos, mejorando la capacidad de generalización de los modelos y simplificando su estructura.

En el estudio de Guyon et al. (2002), se implementó RFE en conjunto con Support Vector Machine (SVM) en un estudio enfocado en la selección de genes para la clasificación de muestras de cáncer. El método permitió eliminar progresivamente las variables menos relevantes y retener las más significativas, lo que resultó en una mejora notable de la exactitud de las predicciones y en una mayor interpretabilidad del modelo. Estos hallazgos confirmaron la efectividad de RFE como técnica de selección en escenarios de clasificación de alta dimensionalidad.

De forma más reciente, Ingio et al. (2024) aplicaron RFE junto con la Regresión Lineal Múltiple (MLR) para predecir el rendimiento del arroz en Nigeria. La técnica permitió identificar las variables climáticas más influyentes, optimizando el desempeño del modelo, mejorando la capacidad predictiva y reduciendo la complejidad del análisis. Estos resultados evidencian la utilidad de RFE en modelos de regresión, lo que respalda su aplicación en enfoques como la regresión logística multinomial para problemas de clasificación.

Estos antecedentes son fundamentales para el presente trabajo, pues evidencian el impacto que tienen las técnicas de selección de características mediante *Recursive Feature Elimination* (RFE), las transformadas wavelet (2D-DWT) y las máquinas de soporte vectorial (SVM) en la mejora del rendimiento de modelos de clasificación. Con base en lo anterior, este proyecto se orienta al desarrollo de un modelo eficiente y preciso para la clasificación de espumas florales, un material poroso con propiedades singulares que introduce desafíos adicionales frente a los casos tradicionales. La aplicación conjunta de estas metodologías avanzadas permitirá automatizar la detección de defectos, incrementar la precisión del proceso y disminuir los costos asociados al control de calidad manual.

3. Marco teórico

Los antecedentes proporcionan una guía objetiva sobre los enfoques que se están utilizando actualmente para investigaciones o estudios cuyo objetivo es la clasificación de productos u objetos mediante modelos. Basados en esto, se definen los componentes teóricos principales que se tendrán en cuenta en el desarrollo de la metodología.

3.1 Espumas Florales

Es un material rígido y poroso elaborado a partir de resinas fenólicas, que son polímeros obtenidos mediante la reacción de fenol con formaldehído. Estas espumas están específicamente diseñadas para su uso con tallos gruesos y altos en arreglos florales, proporcionando soporte estructural, velocidad de absorción y un alto porcentaje de retención de agua, lo que ayuda a mantener las flores frescas durante más tiempo. Sus principales características son:

- **Dimensiones:** Cada panela mide 22.8 cm de largo, 10 cm de ancho y 7.6 cm de alto.
- **Unidad de embalaje:** Cada caja contiene 48 unidades de espuma floral.
- **Peso por unidad (panela):** Aproximadamente 30 gramos por cada panela.
- **Capacidad de absorción:** se refiere a la cantidad máxima de líquido que un material puede retener en relación con su peso o volumen seco. cada panela puede absorber más del 90 % de su capacidad en agua, lo que equivale aproximadamente a 1.5 litros (0.396 galones) de agua por panela, permitiendo que su peso seco aumente entre 40 y 50 veces después de la absorción.
- **Compatibilidad de fluidos:** La compatibilidad de fluidos se refiere a la capacidad de un material para resistir la degradación o reacción cuando entra en contacto con diferentes tipos de líquidos. Esta espuma es compatible con una amplia gama de líquidos, incluidos hidrocarburos, corrosivos y ácidos, lo que la hace versátil para diferentes tipos de arreglos.

- **Dureza:** es una propiedad de los materiales que indica su resistencia a la deformación permanente, generalmente medida por la capacidad de soportar indentaciones o rayaduras. La espuma posee una dureza que permite diseñar arreglos en cualquier ángulo e inclinación, proporcionando estabilidad y soporte a las flores, lo que es esencial para mantener la integridad y estética de los arreglos florales.
- **Estructura celular:** Capacidad de la espuma floral para resistir la presión y el peso ejercido por las flores sin romperse, asegurando la estabilidad estructural del arreglo floral y protegiendo la integridad de las flores colocadas en él. Esta resistencia garantiza que el arreglo se mantenga firme y atractivo durante su exhibición o transporte.

Entre los principales tipos de no conformidades (defectos) que se generan en el proceso de producción se encuentran:

1. **Beta escurrida:** Se refiere a la aparición de líneas o marcas en la superficie del material que ocurren cuando la mezcla se vierte en el molde después de haber comenzado a curarse parcialmente. Esto sucede porque la mezcla se vertió demasiado tarde, cuando ya había empezado a endurecerse, lo que puede provocar una falta de homogeneidad y uniones débiles en el material.
2. **Beta cristalizada:** Se refiere a zonas del material donde se han formado cristales o donde la estructura interna ha cambiado debido a reacciones químicas. Esto ocurre cuando hay concentraciones localizadas de ácido que causan la cristalización del material en esos puntos específicos. La presencia de ácido puede catalizar reacciones no deseadas que alteran la estructura del material.
3. **Poros:** Son pequeñas cavidades o vacíos en el material que pueden debilitar su estructura y afectar sus propiedades. Estos pueden formarse debido a una mezcla inadecuada de la resina, lo que deja burbujas de aire atrapadas en el material, así como por una falta de homogeneización adecuada de los solventes durante el proceso de mezclado. En el caso de las espumas florales fenólicas, este fenómeno es especialmente crítico, ya que el aire atrapado puede afectar la estructura y las propiedades de la espuma.

3.2 Preprocesamiento de imágenes

Es una fase fundamental del procesamiento de imágenes digitales que comprende una serie de técnicas y operaciones destinadas a mejorar la calidad y la utilidad de las imágenes antes de aplicar

algoritmos específicos para el análisis o reconocimiento de patrones. Este proceso incluye diversas tareas, como corrección de color, eliminación de ruido, mejora de contraste, ajuste de brillo, normalización de tamaño y formatos, Así como la eliminación de elementos no deseados (Azuela & Morales, 2011).

Su principal objetivo es optimizar las imágenes para asegurar que los algoritmos de procesamiento posterior puedan operar de manera más efectiva y precisa. Al realizar estas mejoras iniciales, se facilita la extracción de características relevantes y la reducción de la complejidad de los datos, lo que conduce a resultados más confiables en tareas avanzadas, como segmentación de objetos, clasificación de patrones, detección de objetos, reconocimiento facial, entre otras aplicaciones.

3.2.1 Imagen digital - Mapa de bits

Una imagen es la representación óptica de objetos iluminados por alguna fuente de radiación (luz visible, rayos X, infrarroja u otra). La física de la interacción de la radiación y el objeto puede ser representada como una función bidimensional de intensidad de luz $f(x,y)$. La luz reflejada es la imagen óptica de entrada al sistema de formación de una imagen. Tal sistema, según Azuela y Morales (2011) está constituido por:

- **Lentes ópticos o sistema óptico:** Sistema de desplazamiento que tiene una respuesta tipo impulso $h(x,y)$ que suprime las frecuencias (información de detalles) de $f(x,y)$ en una imagen de salida $s(x,y)$.
- **Captor óptico:** Componente que convierte la luz capturada de la imagen de salida $s(x,y)$ en señales eléctricas $n(x,y)$.
- **Digitalizador análogo/ digital de la imagen:** Convierte la señal analógica $n(x,y)$ continua a forma digital, involucrando procesos de cuantificación y muestreo.
 - **Muestreo de la imagen** Proceso que captura la señal $n(x,y)$ en puntos discretos de las coordenadas espaciales x e y .
 - **Cuantificación de la imagen** Proceso que asigna valores numéricos positivos de intensidad de luz a las muestras obtenidas según su amplitud para cuantificar los niveles de gris. Estos niveles se dan en relación binaria: $L = 2^B$, en donde B es el número de bits de la imagen.

A partir de este proceso se obtiene una imagen digital representada por una matriz de fila y columnas ($M_x \times N_y$) cuyos elementos son valores numéricos discretos. Cada elemento identifica de manera espacial un punto de la imagen y su correspondiente nivel de intensidad de luz. Los elementos de este arreglo digital se denominan elementos de la imagen o píxeles (Azuela & Morales, 2011).

3.2.2 Rango dinámico de la imagen

Es una medida de que tan buena es la apariencia de una imagen digital y se denota como el rango de valores del intervalo $[0, L-1]$, donde 0 es el nivel de gris más oscuro y $L-1$ es el más claro. En la práctica se utiliza una cuantificación de 8 bits, que son 256 niveles de grises (Azuela & Morales, 2011). Teniendo en cuenta esto, el número de bytes (un byte tiene 8 bits) para almacenar la imagen digital será: $(M_x \times N_y \times B)$.

3.2.3 Análisis multiresolución

Las señales son obtenidas mediante la medición de su intensidad en el tiempo, espacio, longitud o promedio en intervalos definidos. Esta medición es considerada una observación del comportamiento de la señal (Prieto, 1999). Para un conjunto de n mediciones de una señal, se puede expresar en vector como:

$$y = y(x_i, i = 1, 2, \dots, n)$$

En donde: $x_i = x_0 + i\delta x$

- x : desplazamiento o tiempo en el que se comienza a adquirir la señal.
- δx : intervalo de tiempo en el que se realiza cada observación.

δx define la resolución de la muestra, al variarlo se puede obtener una descripción más detallada o más burda de la señal (Prieto, 1999). En imágenes digitales, δx está determinado por la distancia entre la escena que se adquiere y el centro óptico de la cámara o dispositivo de adquisición, entre otros factores. Mientras más cercana esté la cámara de un objeto, el intervalo cubrirá un área más pequeña. A diferentes resoluciones, los detalles en una señal caracterizan diferentes eventos. En el

caso de una señal bidimensional empleada para una imagen digital, los detalles caracterizan diferentes estructuras físicas dentro de la escena. (Prieto, 1999)

El análisis de multi-resolución, también conocido como análisis multiescala (Nuñez Ramírez et al., 2010), permite examinar una señal en diferentes niveles de resolución, capturando tanto los detalles medios como los más finos. Este enfoque se diseñó para descomponer señales de tiempo discreto con el objetivo de obtener una representación tiempo-escala de las mismas (Prieto, 1999). Una metodología común para llevarlo a cabo utiliza filtros con diversas frecuencias de corte, donde la señal es procesada mediante un filtro de paso alto para extraer las componentes de alta frecuencia y un filtro de paso bajo para analizar las componentes de baja frecuencia. Estas operaciones modifican la resolución de la imagen y ajustan la escala mediante técnicas de interpolación y submuestreo.

3.2.4 Filtro gaussiano

Es un filtro de paso bajo utilizado en el procesamiento de imágenes para suavizar o desenfocar, con el objetivo de reducir el ruido y eliminar detalles finos no deseados, preservando las características más importantes. Este filtro se basa en una distribución gaussiana que asigna pesos a los píxeles según su distancia al centro del filtro. En el dominio espacial para imágenes 2D, el filtro gaussiano se describe matemáticamente mediante la siguiente fórmula: (Gonzalez & Woods, 2018, Sección 3.4).

$$G_0(x, y) = Ae^{-\left(\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2}\right)}$$

donde:

- (x, y) son las coordenadas del pixel.
- μ_x y μ_y son las medias o centros de la distribución en los ejes x e y.
- σ_x^2 y σ_y^2 son las varianzas o dispersión en los ejes x e y que determinan el tamaño del filtro.
- A es una constante de normalización para asegurar que la suma de todos los valores en el filtro sea 1.

3.2.5 Análisis de Textura

En Palomino et al. (2010) se da una definición figurativa de la textura como la repetición de un patrón espacial básico, cuya estructura puede ser periódica o parcialmente periódica (aleatoria). Para la caracterización de la textura en una imagen tiene en cuenta la distribución de valores de intensidad del color a nivel espacial, de manera que se evalúan aspectos como la rigurosidad, granulación, suavidad, fineza, etc.

El análisis de texturas intenta cuantificar las cualidades intuitivas descritas por términos como áspero, suave, sedoso o irregular en función de la variación espacial de las intensidades de píxeles. En ese sentido, la rugosidad o la irregularidad se refieren a las variaciones de los valores de intensidad o niveles de gris. (The MathWorks, 2025)

3.3 Transformada Discreta de Wavelet (DWT)

La Transformada Discreta de Wavelet (DWT) es una herramienta matemática utilizada para analizar y procesar señales e imágenes (Zhang, 2019). Su principal ventaja radica en su capacidad para representar señales de forma eficiente mediante una descomposición en componentes localizados tanto en el tiempo como en la frecuencia, lo que permite un análisis multiresolución adaptativo.

A diferencia de otras técnicas, como la Transformada de Fourier, que proporcionan una visión global de las frecuencias presentes en una señal, la DWT ofrece una representación jerárquica mediante diferentes niveles de resolución. Esto la hace especialmente útil para el análisis de señales no estacionarias, donde la información relevante puede variar a lo largo del tiempo o del espacio (Burrus et al., 1998).

3.3.1 Sistema de expansión wavelet

En este sistema, una señal o función $f(t)$ puede representarse de forma más efectiva si es expresada como una descomposición lineal en términos de funciones base:

$$f(t) = \sum_{\ell} a_{\ell} \psi_{\ell}(t)$$

Donde a_ℓ son los coeficientes de expansión y $\psi_\ell(t)$ las funciones del conjunto de expansión (Burrus et al., 1998). Si el conjunto es una base ortogonal, los coeficientes pueden calcularse mediante el producto interno:

$$a_k = \langle f(t), \psi_k(t) \rangle = \int f(t) \psi_k(t) dt$$

La expansión en wavelet consiste en usar un conjunto bidimensional de funciones base $\psi_{j,k}(t)$, donde j representa la escala y k la traslación. Esto conduce a una expresión del tipo:

$$f(t) = \sum_{jk} a_{j,k} \psi_{j,k}(t)$$

Cada función $\psi_{j,k}(t)$ se define como una versión escalada y trasladada de una función madre $\psi(t)$:

$$\psi_{j,k}(t) = 2^{\frac{j}{2}} \psi(2^j t - k) \quad j, k \in Z$$

Donde Z es el conjunto de los enteros y el factor $2^{\frac{j}{2}}$ mantiene una norma constante independientemente de la escala j . Esto permite representar una señal en distintos niveles de resolución y localización, lo que hace que las wavelets sean efectivas para señales transitorias, cuya información local cambia con el tiempo (Sweldens, 1996).

El análisis de wavelets se basa en el concepto de multiresolución, donde se definen espacios funcionales anidados:

$$\dots \subset V_{-1} \subset V_0 \subset V_1 \subset \dots \subset L^2$$

A partir de eso, se definen subespacios W_j wavelets como:

$$V_1 = V_0 \oplus W_0 \quad \text{que se extiende a:} \quad V_2 = V_0 \oplus W_0 \oplus W_1$$

Y en general:

$$L^2 = V_0 \oplus W_0 \oplus W_1 \oplus W_2 \oplus \dots$$

Donde V_0 es el espacio inicial generado por la función escala $\varphi(t - k)$ que representa la aproximación (contenido general) y los subespacios W_j son generados por $\psi(2^j t - k)$ que representa los detalles agregados al pasar de una resolución a otra.

La condición de multiresolución implica que si una señal puede representarse mediante una suma

ponderada de funciones base $\varphi(t - k)$ entonces al reducir estas funciones a la mitad de su tamaño y trasladarlas en pasos proporcionales $\varphi(2t - k)$, se logra una representación más precisa o una mejor aproximación de una mayor variedad de señales (Sweldens (1996); Daubechies (1992)).

Con esta base, una función $f(t)$ puede representarse como:

$$f(t) = \sum_k c_{j_0}(k) \varphi_{j_0,k}(t) + \sum_{j=j_0}^{\infty} \sum_k d_j(k) \psi_{j,k}(t)$$

Donde los coeficientes se calculan como:

$$c_j(k) = \int f(t) \varphi_{j,k}(t) dt \quad \text{Coeficiente de aproximación}$$

$$d_j(k) = \int f(t) \psi_{j,k}(t) dt \quad \text{Coeficientes de detalle}$$

Este sistema constituye la Transformada Discreta de Wavelet (DWT) y se ha demostrado que forma una base incondicional para una amplia clase de señales, lo cual significa que los coeficientes disminuyen rápidamente al aumentar j, k permitiendo una representación eficiente (Donoho (1993); Donoho et al. (1995))

3.3.2 Wavelet de Haar

Es el sistema de wavelet más simple y utilizado, introducida por Haar (1910). Se caracteriza por su soporte compacto, lo que implica que su acción se limita a un intervalo finito de la señal y la hace sensible a discontinuidades y cambios abruptos en la señal (Haar, 1910). Asimismo, destaca por su simplicidad al emplear operaciones básicas de promedio y diferencia (Zhang, 2019). Se construye a partir de una única función de escala $\varphi(t)$ y una wavelet madre asociada $\psi(t)$.

- **Escala** $\varphi(t)$: se utiliza para capturar las características globales (el promedio) de una señal en un intervalo. La Figura 3-1 (a) muestra el comportamiento de la función y se define como:

$$\phi(t) = \begin{cases} 1, & 0 \leq t < 1, \\ 0, & \text{en otro caso} \end{cases}$$

- **Wavelet (detalle)** $\psi(t)$: se utiliza para capturar las características locales (la diferencia) dentro del intervalo. La Figura 3-1 (b) muestra el comportamiento de la función y se define

como:

$$\psi(t) = \begin{cases} 1, & 0 \leq t < \frac{1}{2}, \\ -1, & \frac{1}{2} \leq t < 1 \\ 0, & \text{en otro caso} \end{cases}$$

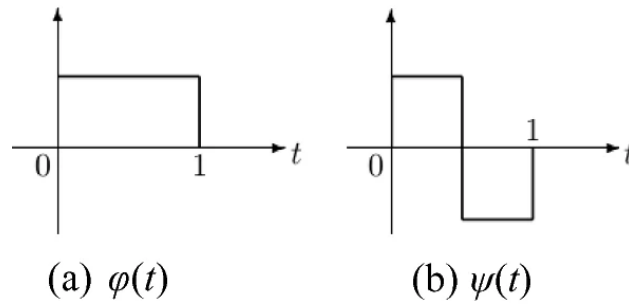


Figura 3-1.: Wavelet de Haar: función escala y función wavelet

En el caso de señales unidimensionales, como señales de audio, la DWT divide la señal en dos componentes en cada nivel: una **aproximación** (baja frecuencia) que contiene la información general, y un **detalle** (alta frecuencia) que refleja las variaciones rápidas o cambios locales (Zhang, 2019).

Para imágenes bidimensionales, se utiliza un árbol de descomposición en donde las filas y columnas se tratan como señales 1D. Gracias a la propiedad de separabilidad de la DWT, cada paso de la transformada se aplica de manera separada, primero a las filas y luego a las columnas (Zhang, 2019).

Para la Transformada discreta de wavelet bidimensional de Haar, sean x e y dos píxeles (puntos) vecinos en una imagen, las funciones de escala y wavelet están dadas por:

$$\textbf{Escala: } \varphi(x, y) = \frac{x + y}{2} \quad \textbf{Wavelet: } \psi(x, y) = \frac{x - y}{2}$$

3.3.3 Proceso de descomposición

Transformada Horizontal (Zhang, 2019): Para el nivel 1 de descomposición se procesa **cada fila** de la imagen:

- Se Aplica el promedio ponderado (filtro de paso bajo (escala) $\rightarrow$ L) y una diferencia ponderada (filtro de paso alto (Wavelet) $\rightarrow$ H)
- El resultado son dos mitades de la imagen que tienen el mismo ancho de la imagen original por fila:
 - Una contiene los coeficientes de escala (L)
 - La otra contiene los coeficientes de wavelet (H)

Transformada vertical (Zhang, 2019): Se aplica nuevamente el filtrado, pero ahora a cada **columna** de las **dos mitades obtenidas** en el paso anterior, el resultado es una imagen dividida en cuatro cuadrantes más pequeños como:

- LL (Superior izquierda): resumen general o aproximación.
- LH (Inferior izquierdo): detalles horizontales.
- HL (Superior derecha): detalles verticales.
- HH (inferior derecha): detalles diagonales.

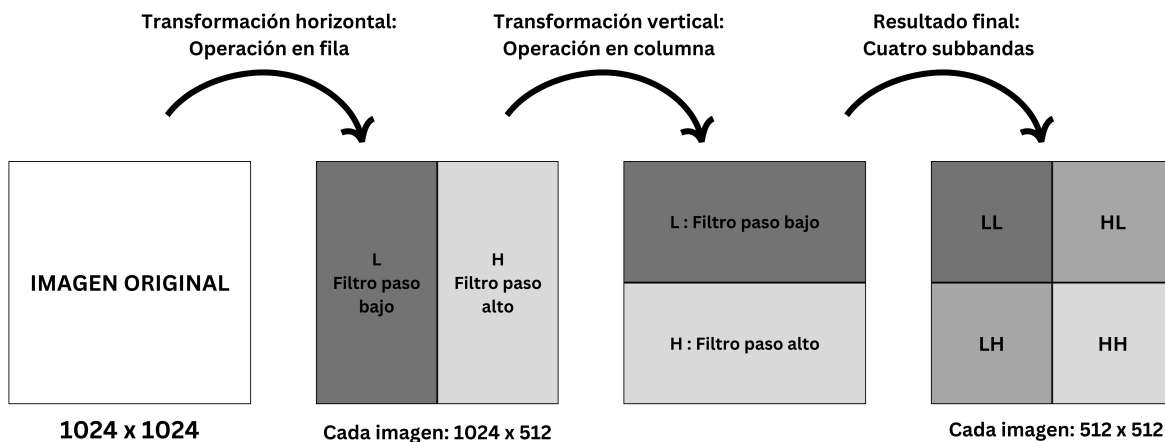


Figura 3-2.: Esquema del proceso de descomposición DWT en una imagen 2D

La Figura 3-2 ilustra gráficamente el proceso de descomposición explicado, a partir de una imagen original de tamaño $M \times N = 1024 \times 1024$, se realiza primero una transformación horizontal que

reduce a la mitad el número de columnas. Luego, se aplica una transformación vertical a las dos mitades obtenidas, lo que finalmente produce cuatro subimágenes o subbandas de tamaño 512×512 cada una, correspondientes a los diferentes niveles de detalle de la imagen original.

Este proceso es iterativo y se puede aplicar recursivamente a la imagen de la subbanda LL para obtener múltiples niveles de descomposición siempre que esta subimagen tenga tamaño suficiente para seguir dividiéndose. Esto es fundamental en aplicaciones como la compresión de imágenes, eliminación de ruido, análisis de texturas, entre otras (Zhang, 2019).

3.4 Local Binary Patterns (LBP)

Es un operador utilizado para describir la textura local de las imágenes. Introducido por Ojala et al. (2002), LBP compara los valores de los píxeles vecinos con el valor del píxel central en una vecindad circular y genera un patrón binario único que describe la textura local.

3.4.1 Definición de LBP

El valor de LBP para un píxel se calcula comparando el valor de intensidad del píxel central I_c con sus vecinos en una vecindad circular. El valor de cada vecino I_p se compara con I_c , y si el valor es mayor o igual, se asigna un 1; de lo contrario, se asigna un 0. Estos valores binarios se concatenan para formar un número binario único, que luego se convierte en un valor decimal.

La fórmula matemática para calcular el valor de LBP de un píxel es:

$$LBP(p) = \sum_{p=0}^{P-1} s(I_p - I_c) \cdot 2^p$$

Donde:

- I_c es el valor de intensidad del píxel central.
- I_p es el valor de intensidad de un píxel vecino en una vecindad circular.
- $s(x)$ es una función que devuelve 1 si $x \geq 0$ y 0 si $x < 0$.
- P es el número total de píxeles vecinos.

3.4.2 Histograma de LBP

Una vez calculados los valores de LBP para todos los píxeles de una imagen, se genera un **histograma** de las frecuencias de aparición de cada patrón LBP. Este histograma es utilizado como una *característica de textura*. La fórmula para calcular el histograma es:

$$H(LBP) = \sum_{i=0}^{N-1} f_i$$

Donde:

- f_i es la frecuencia de ocurrencia del patrón i -ésimo en la imagen.
- N es el número total de patrones posibles.

3.4.3 LBP Invariante a la Rotación

Para hacer que LBP sea *invariante a la rotación*, se realiza una **rotación cíclica** de los patrones binarios y se elige el patrón con el valor más pequeño, lo que asegura que un patrón rotado tenga siempre el mismo valor. Esto se puede expresar como:

$$LBP_{ri}(p) = \min_i (ROR(LBP(p), i))$$

Donde $ROR(x, i)$ es la rotación circular del valor x i veces.

3.4.4 LBP y Patrones Uniformes

Los *patrones uniformes* son aquellos que tienen un número limitado de transiciones entre 0 y 1, y son importantes porque ocurren con más frecuencia en las texturas naturales. Un patrón se considera uniforme si tiene dos o menos transiciones. La fórmula para identificar si un patrón es uniforme es:

$$U(LBP) = \sum_{p=0}^{P-1} |s(I_p - I_c) - s(I_{p-1} - I_c)|$$

Sí $U(LBP) \leq 2$, el patrón es considerado uniforme.

3.5 Máquinas de Soporte Vectorial (SVM)

Las Máquinas de Soporte Vectorial (SVM, por sus siglas en inglés) son un método de aprendizaje supervisado diseñado principalmente para tareas de clasificación y, en algunos casos, regresión. Según James et al. (2013), estas técnicas se centran en identificar un hiperplano que divida las clases en el espacio de características de manera óptima, maximizando la distancia entre las observaciones más cercanas de cada clase y el hiperplano, conocido como margen. Las SVM son altamente eficaces en problemas de alta dimensionalidad y pueden extenderse a problemas no lineales mediante el uso de kernels, que transforman los datos a espacios de características más elevados (James et al., 2013).

3.5.1 Definición de Hiperplano

Un hiperplano es una frontera que separa un espacio en dos regiones diferentes. En un espacio de p dimensiones, James et al. (2013) definen un hiperplano matemáticamente como:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0,$$

donde β_0 representa el término independiente, y β_j son los coeficientes que indican la influencia de cada predictor X_j en la frontera.

3.5.2 Clasificador de Margen Máximo

El clasificador de margen máximo, según James et al. (2013), tiene como objetivo encontrar el hiperplano que maximice la separación entre las dos clases. Este margen M se define como la distancia mínima entre el hiperplano y las observaciones de cada clase. Para garantizar esta separación, se resuelve el siguiente problema de optimización:

$$\text{maximizar } M, \quad \text{sujeto a } \sum_{j=1}^p \beta_j^2 = 1, \quad y_i(\beta_0 + \sum_{j=1}^p \beta_j x_{ij}) \geq M,$$

donde y_i corresponde a la clase de la observación i , x_{ij} es el valor de la característica j para el individuo i , y los coeficientes β determinan la orientación y posición del hiperplano.

3.5.3 Clasificador de Soporte Vectorial

Cuando las clases no son linealmente separables, James et al. (2013) introducen el uso de un término adicional ξ_i , que permite errores de clasificación. Este enfoque es controlado por el parámetro C , que regula el equilibrio entre la anchura del margen y la tolerancia a errores. Valores grandes de C fuerzan al modelo a clasificar correctamente la mayoría de las observaciones, aun si ello implica un margen estrecho, mientras que valores pequeños permiten más errores de clasificación, pero generan un margen más amplio y regularizado. La formulación de optimización queda entonces como:

$$\text{minimizar } \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^n \xi_i, \quad \text{sujeito a } y_i(\beta_0 + \sum_{j=1}^p \beta_j x_{ij}) \geq 1 - \xi_i, \quad \xi_i \geq 0.$$

De este modo, el parámetro C funciona como un regulador de la complejidad del modelo, y su ajuste se realiza comúnmente mediante validación cruzada.

3.5.4 Extensión a Problemas No Lineales

Para abordar problemas más complejos, las SVM utilizan funciones kernel que permiten transformar los datos a espacios de mayor dimensionalidad, donde las clases puedan separarse linealmente. Según James et al. (2013), entre los diferentes kernels se encuentran:

1. Kernel lineal:

$$K(x, x') = x^\top x',$$

utilizado cuando las clases pueden separarse aproximadamente por un hiperplano en el espacio original. Es computacionalmente eficiente y especialmente útil en problemas de alta dimensión como clasificación de texto.

2. Kernel polinómico:

$$K(x, x') = (1 + x^\top x')^d,$$

donde d es el grado del polinomio. Este kernel permite capturar interacciones no lineales de complejidad moderada entre los predictores.

3. Kernel radial (RBF):

$$K(x, x') = \exp(-\gamma \|x - x'\|^2),$$

donde γ controla el alcance de la influencia de cada observación. Valores grandes de γ producen fronteras muy complejas y ajustadas a los datos, mientras que valores pequeños generan fronteras más suaves y generalizadas.

3.6 Regresión Logística

La regresión logística es un modelo estadístico utilizado para estimar la probabilidad de que un evento binario ocurra, es decir, cuando la variable dependiente Y toma valores de 0 o 1. Este modelo pertenece a la familia de los modelos lineales generalizados (GLM) y es ampliamente empleado en contextos donde la variable de respuesta es binaria (Montgomery et al., 2012).

El modelo de regresión logística se define como:

$$P(Y = 1 | X) = \pi(X) = \frac{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p}}$$

Donde:

- X es el vector genérico de variables predictoras.
- $x_1, \dots, x_p$ son las variables predictoras observadas.
- β_0 es el intercepto del modelo.
- $\beta_1, \dots, \beta_p$ son los coeficientes asociados a las variables predictoras.
- e es la base del logaritmo natural.

El término $\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$ se conoce como el *predictor lineal* del modelo.

El *logit*, en cambio, es la función que relaciona la probabilidad con este predictor lineal y se define como:

$$\text{logit}(\pi) = \log \left(\frac{\pi(X)}{1 - \pi(X)} \right) = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$$

Los parámetros del modelo se estiman mediante el método de máxima verosimilitud, cuya expresión se resume en la siguiente función de verosimilitud:

$$L(\beta) = \prod_{i=1}^n \pi(x_i)^{y_i} [1 - \pi(x_i)]^{1-y_i}$$

Donde:

- $x_i = (1, x_{i1}, x_{i2}, \dots, x_{ip})$ es el vector de variables predictoras observado para la observación i , donde el primer elemento corresponde al término del intercepto.
- y_i es el valor observado de la variable respuesta para la observación i , donde $y_i \in \{0, 1\}$.

cuya versión logarítmica es la log-verosimilitud:

$$\ell(\beta) = \sum_{i=1}^n \left[y_i \log(\pi(x_i)) + (1 - y_i) \log(1 - \pi(x_i)) \right].$$

Regularización y parámetro C : En la regresión logística binaria, se suele añadir un término de regularización para evitar el sobreajuste. La función de pérdida penalizada puede escribirse como:

$$J(\beta) = -\ell(\beta) + \lambda \|\beta\|^2,$$

donde λ es el factor de regularización. En implementaciones modernas, este valor se parametriza como $C = 1/\lambda$. Un valor grande de C reduce la regularización y permite mayor flexibilidad del modelo, mientras que un valor pequeño la intensifica, favoreciendo modelos más simples y estables (Bishop, 2006).

3.6.1 Regresión logística multinomial (RLM)

El modelo logístico multinomial es una extensión de la regresión logística que se utiliza cuando la variable dependiente tiene más de dos categorías posibles. A diferencia de la regresión logística binaria, la versión multinomial estima la probabilidad de que una observación pertenezca a una de varias categorías mutuamente excluyentes. Este modelo es ampliamente utilizado en problemas de clasificación multiclase, donde se busca asignar cada observación a una de varias clases (Montgomery et al., 2012).

En esta regresión se selecciona una categoría de referencia (o base) para comparar las probabilidades de las demás categorías. Con esta elección, el modelo calcula las probabilidades relativas de

cada clase respecto a la referencia, lo que permite formular un modelo identificable y evita problemas de multicolinealidad (Bishop, 2006).

Para cada clase j , se asocia un vector de coeficientes β_j , y las probabilidades se modelan mediante la función softmax:

$$P(Y_i = j \mid \mathbf{x}_i) = \frac{\exp(\mathbf{x}_i^\top \beta_j)}{\sum_{k=1}^K \exp(\mathbf{x}_i^\top \beta_k)}.$$

Donde:

- $P(Y_i = j \mid \mathbf{x}_i)$ es la probabilidad de que la observación i pertenezca a la clase j .
- $\mathbf{x}_i = (1, x_{i1}, x_{i2}, \dots, x_{ip})$ es el vector de variables predictoras observado para la observación i , incluyendo el término de intercepto.
- $\beta_j = (\beta_{0j}, \beta_{1j}, \dots, \beta_{pj})$ es el vector de coeficientes asociado a la clase j .
- K es el número total de clases.

La estimación de los parámetros se realiza a partir de la función de verosimilitud multinomial:

$$L(\beta) = \prod_{i=1}^n \prod_{j=1}^K p_{ij}^{t_{ij}},$$

donde $p_{ij} = P(Y_i = j \mid \mathbf{x}_i)$. La correspondiente log-verosimilitud está dada por:

$$\ell(\beta) = \sum_{i=1}^n \sum_{j=1}^K t_{ij} \log(p_{ij}).$$

Donde:

- n es el número total de observaciones.
- t_{ij} es una variable indicadora que vale 1 si la observación i pertenece a la clase j , y 0 en caso contrario.
- p_{ij} es la probabilidad estimada de que la observación i pertenezca a la clase j .
- $\ell(\beta)$ es la log-verosimilitud que se maximiza para obtener los parámetros óptimos del modelo.

Para obtener los estimadores de máxima verosimilitud en la regresión logística multinomial es necesario recurrir a métodos de optimización iterativos, ya que la solución analítica no está disponible. Entre las técnicas más utilizadas se encuentran los algoritmos cuasi-Newton, como BFGS y su variante de memoria limitada L-BFGS, que aproximan la matriz hessiana sin calcularla explícitamente y permiten una optimización eficiente incluso en modelos de gran escala (Morales & Nocedal, 2011). Estos métodos se emplean de manera habitual en la resolución de problemas de máxima verosimilitud debido a su estabilidad numérica y a la eficiencia con la que ajustan los parámetros del modelo.

3.7 Validación Cruzada

La validación cruzada es una técnica estadística utilizada para evaluar el desempeño predictivo de un modelo y garantizar su capacidad de generalización a datos no observados. Este método es fundamental en problemas donde el objetivo es seleccionar o validar modelos predictivos, evitando fenómenos como el sobreajuste (*overfitting*) que ocurre cuando un modelo se ajusta demasiado a los datos de entrenamiento, perdiendo eficacia con nuevos datos.

Según Weisberg (2014), la validación cruzada implica dividir los datos en subconjuntos separados que se utilizan para entrenar y evaluar un modelo. El objetivo principal es medir el error de predicción de un modelo en datos que no se usaron para su construcción, proporcionando así una estimación realista de su desempeño. Por otro lado, Draper y Smith (1998) enfatizan que esta técnica también permite comparar diferentes configuraciones de modelos, proporcionando una base objetiva para seleccionar aquel que mejor generalice.

La validación cruzada puede implementarse de varias maneras, siendo el método más común el *k-fold cross-validation*:

1. Se divide el conjunto de datos en k subconjuntos o *folds*.
2. Se entrena el modelo usando $k - 1$ *folds* y se valida en el *fold* restante.
3. Este proceso se repite k veces, asegurando que cada *fold* se utilice una vez como conjunto de validación.
4. Finalmente, se calcula el promedio de las métricas de desempeño obtenidas en cada iteración (Weisberg, 2014).

3.8 Recursive Feature Elimination (RFE)

La Eliminación Recursiva de Características (RFE, por sus siglas en inglés, Recursive Feature Elimination) es un método de selección de características que permite reducir la dimensionalidad de un conjunto de datos y mejorar la precisión de los modelos de clasificación (Guyon et al., 2002). Su principio consiste en eliminar de manera iterativa aquellas características que aportan menor información al modelo, reteniendo únicamente las más relevantes para la predicción.

Sea un modelo de clasificación que aprende una función de decisión $D(\mathbf{x})$ a partir de un vector de características $\mathbf{x} = [x_1, x_2, \dots, x_n]$, de la forma:

$$D(\mathbf{x}) = f(\mathbf{w}, \mathbf{x})$$

donde $\mathbf{w} = [w_1, w_2, \dots, w_n]$ es el vector de pesos o coeficientes estimados por el modelo. Cada coeficiente w_i cuantifica la contribución de la característica correspondiente x_i en la decisión final del clasificador.

En un clasificador lineal, como un SVM, la función de decisión se expresa como:

$$D(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

y en regresión logística multinomial, la probabilidad de que la muestra $\mathbf{x}$ pertenezca a la clase c se calcula mediante:

$$P(y = c|\mathbf{x}) = \frac{e^{\mathbf{w}_c^T \mathbf{x}}}{\sum_k e^{\mathbf{w}_k^T \mathbf{x}}}$$

En ambos modelos, el vector de pesos $\mathbf{w}$ refleja la influencia de cada característica en el proceso de clasificación, lo que permite utilizar dichos coeficientes como criterio para evaluar su relevancia y, en consecuencia, realizar su eliminación de manera secuencial.

3.8.1 Procedimiento General de RFE

El procedimiento de *RFE* se realiza según los siguientes pasos (Guyon et al., 2002) :

1. **Entrenamiento del modelo inicial:** Se entrena el modelo utilizando todas las características disponibles.

2. **Evaluación de la importancia de cada característica:** La relevancia de cada característica se calcula mediante la magnitud del peso o coeficiente:

$$c_i = \|\mathbf{w}_i\|^2$$

3. **Eliminación de la característica menos importante:** Se elimina la característica x_f con menor contribución al modelo:

$$x_f = \arg \min_i c_i$$

4. **Actualización del subconjunto de características:** Se reentrena el modelo con el conjunto reducido y se repite el proceso hasta alcanzar un número óptimo de características.
5. **Ranking final de características:** Las características eliminadas al final son las más importantes, produciendo subconjuntos anidados de características:

$$F_1 \subset F_2 \subset \dots \subset F$$

Ventajas y Consideraciones

El principal beneficio de RFE es que permite seleccionar un subconjunto de características que maximiza la precisión del modelo y minimiza la complejidad, evitando el *overfitting* (Guyon et al., 2002). Además, es un método **multivariado**, ya que considera la interacción entre variables, a diferencia de los métodos univariados que evalúan cada característica de manera independiente (Kohavi & John, 1997). Esto lo hace aplicable a modelos como SVM, regresión logística multinomial u otros clasificadores capaces de asignar un peso a las variables de entrada.

En síntesis, RFE constituye una herramienta potente para la selección de características en problemas de alta dimensionalidad, proporcionando un marco sistemático para identificar las variables más relevantes y optimizar el desempeño de los modelos de aprendizaje automático.

3.9 Métricas de evaluación

Para evaluar y comparar la capacidad de clasificación de un modelo, se definen métricas estadísticas que cuantifican el rendimiento del modelo. Estas métricas se derivan de una matriz de confusión, que presentan los resultados de la clasificación.

3.9.1 Matriz de confusión

Es una herramienta para evaluar el desempeño de un modelo de clasificación. Esta matriz muestra cómo las predicciones del modelo se comparan con los valores verdaderos en una tabla que resume la frecuencia con que las observaciones fueron clasificadas correctamente o incorrectamente en cada categoría (Rainio et al., 2024).

Caso binario

Esta es la estructura básica, en la que se consideran únicamente dos categorías: una positiva y una negativa. En las tareas de clasificación binaria, la etiqueta positiva se interpreta como la presencia de una anomalía, mientras que la etiqueta negativa indica su ausencia (Rainio et al., 2024). Por lo tanto, cada etiqueta binaria puede asumir una de cuatro posibles designaciones:

- Verdadero positivo (VP): Predicción positiva cuando el valor real es positivo. (Clasificación correcta)
- Falso positivo (FP): Predicción positiva cuando el valor real es negativo. (Error Tipo I)
- Verdadero negativo (VN): Predicción negativa cuando el valor real es negativo. (Clasificación correcta)
- Falso negativo (FN): Predicción negativa cuando el valor real es positivo. (Error Tipo II)

En el caso de la clasificación binaria, se obtiene una matriz de 2×2 que contiene las observaciones VP, FP, VN y FN (Tabla 3-1), las cuales se utilizan para calcular las métricas de evaluación del clasificador binario.

Tabla 3-1.: Matriz de confusión para clasificación binaria

		Predicción	
		Positivo	Negativo
Valores Reales	Positivo	VP	FN
	Negativo	FP	VN

Caso multiclase

En los casos en que el clasificador debe distinguir entre $k \geq 3$ clases diferentes, los resultados pueden representarse mediante una matriz de confusión de tamaño $k \times k$, cuyos elementos n_{ij} corresponden a la intersección entre la i -ésima fila y la j -ésima columna, para $i, j = 1, 2, \dots, k$. Cada elemento n_{ij} indica el número de instancias de la clase i -ésima que han sido clasificadas como pertenecientes a la clase j -ésima (Rainio et al., 2024).

Para calcular las métricas de evaluación, se construye una matriz binaria de 2×2 para cada una de las k clases, en la que se definen las observaciones VP, FP, VN y FN correspondientes a cada clase (Rainio et al., 2024). A partir de la matriz de confusión multiclase, estos valores pueden obtenerse considerando cada clase i como positiva y agrupando las restantes como negativas, de acuerdo con las siguientes expresiones:

$$VP_i = n_{ii} \quad VN_i = \sum_{j \neq i} \sum_{h \neq i} n_{jh} \quad FN_i = \sum_{j \neq i} n_{ij} \quad FP_i = \sum_{j \neq i} n_{ji} \quad \text{Para } i = 1, 2, \dots, k.$$

3.9.2 Métricas derivadas

A partir de las observaciones (VP, FP, VN y FN), tanto para el caso binario como para el multiclase, es posible calcular diversas métricas que permiten evaluar el desempeño del clasificador.

Evaluación binaria

En el caso de la clasificación binaria, las métricas utilizadas son las siguientes:

- **Accuracy (exactitud global):** Fracción de clasificaciones correctas realizadas por el modelo.

$$Accuracy = \frac{VP + VN}{VP + FP + FN + VN} \in [0, 1]$$

- **Sensibilidad (Recall):** Capacidad del modelo para predecir correctamente los valores positivos, es decir, la proporción de verdaderos positivos identificados.

$$Sensibilidad = \frac{VP}{VP + FN} \in [0, 1]$$

- **Especificidad:** Capacidad del modelo para predecir correctamente los valores negativos, es decir, la proporción de verdaderos negativos identificados.

$$\text{Especificidad} = \frac{VN}{VN + FP} \in [0, 1]$$

- **F1-score:** Es la media armónica entre la precisión y la sensibilidad, que penaliza más fuertemente cuando alguna de las dos métricas es baja. Esto genera una métrica que mide el equilibrio entre ambas, permitiendo un balance adecuado entre falsos positivos y falsos negativos.

$$F1 - score = 2 \cdot \frac{\text{Precision} \cdot \text{sensibilidad}}{\text{Precision} + \text{sensibilidad}} \in [0, 1]$$

Métricas de clasificación multiclase

En el caso multiclase, se emplean las mismas métricas utilizadas en la clasificación binaria (Accuracy, Sensibilidad, Especificidad y F1-score), pero adaptadas para considerar las k clases.

- **Macro-Promedio:** Se calcula el valor de la métrica por separado para cada clase $i = 1, 2, \dots, k$ utilizando las observaciones de TP_i , TN_i , FP_i y FN_i definidos anteriormente y se calcula el valor medio de los k valores resultantes de la métrica. Mediante este proceso, en el resultado de cada métrica se da la misma importancia a cada clase independiente de su tamaño, lo que permite medir el equilibrio entre las clases (Rainio et al., [2024](#)).

4. Metodología

Con el fin de cumplir los objetivos planteados y garantizar un proceso sistemático, se diseñó una metodología basada en un flujo completo de aprendizaje automático en imágenes. La Figura 4-1 muestra el procedimiento que incluye etapas de preparación de datos, validación cruzada, extracción y selección de características, entrenamiento de modelos y evaluación comparativa de resultados.

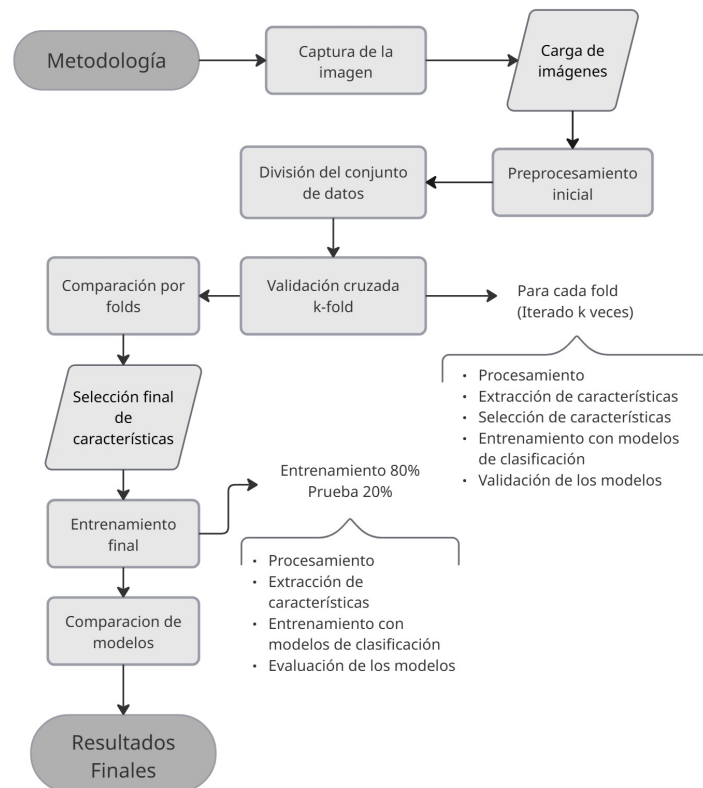


Figura 4-1.: Metodología desarrollada

4.1 Captura de imágenes

En este estudio, las imágenes fueron tomadas en el laboratorio de docencia de estadística. Se utilizaron espumas florales de un tamaño aproximado de 22.8 cm para la captura de imágenes. Como parte del proceso de captura de datos, se empleó un análisis exploratorio mediante curvas de aprendizaje, estas son funciones que relacionan el rendimiento de un modelo con la cantidad de recursos empleados en su entrenamiento. Este tipo de curvas permite observar cómo evoluciona el desempeño del modelo conforme se incrementa el tamaño del conjunto de entrenamiento, identificando el punto en el que el aprendizaje se estabiliza (Mohr & van Rijn, 2024).

Dicho análisis mostró que, a partir de aproximadamente 80 a 90 imágenes por clase, el desempeño del modelo tendía a estabilizarse, con mejoras marginales al incorporar más datos. En complemento a esto y considerando las dificultades en términos de recursos computacionales y logísticos disponibles, se estableció un tamaño de muestra de 95 imágenes por clase.

Para la creación del modelo se considerarán cuatro tipos de clases, entre ellas, tres son clases que constituyen un defecto visual de la espuma, que son: Beta cristalizada, Beta escurrida y poros, y una clase relaciona con el estado adecuado que es la espuma buena.

Además, dado que la espuma tiene una forma en bloque rectangular, y con el objetivo de visualizar sus seis caras, se optó por tomar dos fotografías por cada espuma, de manera que en cada una se visualizan tres caras. En la Figura 4-2 se observan las clases consideradas y su conformidad.



Comforme - Espuma Buena



No Conforme - Espuma Beta cristalizada



No Conforme - Espuma Beta escurrida



No Conforme - Espuma Poros

Figura 4-2.: Imágenes según las diferentes clases (Lado A de la imagen)

En total, se contó con 380 panelas de espumas florales, previamente clasificadas según su condición (conforme, beta escurrida, beta cristalizada y poros), con 95 espumas por cada clase. Esta selección no se realizó de forma aleatoria, sino de manera intencional, con el fin de garantizar un conjunto de datos balanceado para el entrenamiento y evaluación del modelo. De cada panela se capturaron 190 imágenes, obteniéndose así un total de 760 imágenes para procesar.

4.1.1 Equipo Utilizado

Para la captura de imágenes se utilizó una cámara Canon EOS M50 Mark II con lente EF-M 15–45 mm f/3.5–6.3 IS STM, montada sobre un trípode para garantizar estabilidad. La configuración de la cámara incluyó una apertura de f/16, velocidad de obturación de 0.6 s, ISO y balance de blancos automáticos, y un estilo de imagen neutro, priorizando la nitidez y el control de la exposición.

La iluminación se proporcionó mediante un bombillo LED de 7 W (600 lm, 6500 K), proyectado desde dos ángulos para resaltar texturas y minimizar reflejos. La composición se realizó con la cámara a 45° sobre la panela y un fondo blanco, ubicando las espumas de forma que se destacaran sus irregularidades superficiales.



Figura 4-3.: Equipos utilizados en la captura de imagen.

Los detalles técnicos completos del equipo, configuración y entorno de captura se presentan en el [Anexo A](#).

4.1.2 Proceso de almacenamiento

Las imágenes capturadas fueron guardadas en una carpeta de Google Drive en formato JPG, con dimensiones de 6000×3368 píxeles. Dentro de esta carpeta, las imágenes se organizaron en subcarpetas, las cuales fueron nombradas de acuerdo al tipo de defecto y el lado de la espuma, como se detalla en la Tabla 4-1. Además, cada imagen se guardó individualmente bajo un nombre que incluye el tipo de defecto, el número de toma y el lado correspondiente. Por ejemplo, la imagen nombrada **B1A** corresponde a la categoría B (espuma buena), con el número de toma 1, y el lado A.

Tabla 4-1.: Organización de las imágenes capturadas

Nombre de las carpetas	Descripción
B - A	Espuma Buena - Lado A
B - B	Espuma Buena - Lado B
BTC - A	Espuma con Beta Cristalizada - Lado A
BTC - B	Espuma con Beta Cristalizada - Lado B
BTE - A	Espuma con Beta Ecurrida - Lado A
BTE - B	Espuma con Beta Ecurrida - Lado B
PO - A	Espuma con Poro - Lado A
PO - B	Espuma con Poro - Lado B

4.2 Carga de imágenes

El trabajo fue realizado completamente en Google Colab utilizando el lenguaje de programación Python, dada la cantidad de imágenes y la necesidad de realizar diversos procesos computacionales sobre ellas. Para facilitar el acceso a las imágenes y optimizar el uso de recursos, estas fueron cargadas en Google Drive.

Considerando las limitaciones de memoria RAM del entorno de ejecución de Google Colab, se optó por no cargar las imágenes directamente en el script. En su lugar, se generó un archivo CSV almacenado en Google Drive con el nombre `pares_imagenes`, que contiene la información por fila de cada par de imágenes. Para ello, se utiliza la librería `pandas` en la manipulación, lectura y guardado de los archivos (Team, 2021).

Las rutas de acceso a las imágenes fueron construidas basándose en la estructura de carpetas mencionada anteriormente, y los nombres de las imágenes fueron registrados en el archivo CSV con las etiquetas *name_A* o *name_B*, según corresponda.

El archivo CSV está organizado según columnas que incluyen las rutas y nombres de las imágenes, así como su etiqueta numérica correspondiente. En la Tabla 4-2 se muestra la estructura por columnas del archivo.

Tabla 4-2.: Nombre de las columnas del archivo CSV

ruta_A	name_A	ruta_B	name_B	label
--------	--------	--------	--------	-------

De esta forma, se accede a las imágenes mediante las rutas especificadas sin necesidad de cargarlas completamente en memoria. Esto optimiza el rendimiento y reduce el consumo de recursos durante el procesamiento.

4.3 Preprocesamiento inicial

En las imágenes capturadas (Ver Figura 4-2), se observa la espuma floral sobre un fondo blanco, que ocupa una parte considerable del espacio en la imagen. Este fondo puede ser problemático durante la extracción de características, ya que podría generar interferencias en el análisis del objeto principal. Por esta razón, se realiza un procesamiento inicial que consiste en recortar la imagen original, eliminando parte de este fondo y enfocando la atención en el objeto de interés: la espuma.

De manera general, el proceso comenzó leyendo las rutas de las imágenes desde el archivo CSV (`pares_imagenes`), usando la librería `pandas` mencionada anteriormente. Adicionalmente, se utilizaron las siguientes librerías en el proceso:

- **cv2 (OpenCV):** Utilizada para tareas de procesamiento de imágenes (O. Contributors, 2021a).
- **numpy:** Para manejar arrays y realizar operaciones matemáticas sobre las matrices de imágenes (N. Developers, 2021).
- **PIL:** Usada para abrir, guardar y convertir las imágenes (P. Contributors, 2021).

El procedimiento consiste en encontrar y recortar el bloque (espuma) según un color predominante (en este caso, el “verde”) en la imagen, utilizando un rango en el espacio de color HSV calculado automáticamente a partir del color dominante del centro de la imagen.

Convertir del espacio de color RGB a HSV: Se utilizan las funciones de `Open CV`: `cv2.cvtColor` y `cv2.COLOR_BGR2HSV` para este proceso.

- **Normalización:** Para la imagen de 8 bits (valores de pixel 0-255) se convierte los valores de cada canal R G B a números decimales entre 0 y 1 para tener una escala estándar.
- **Conversión a HSV:** Es usado para segmentar una imagen, ya que este espacio describe los colores en términos de tono (H), saturación (S) y valor (V), lo que facilita la detección de objetos (O. Contributors, 2021b).

- **Cálculo de V (Brillo):** Valor que indica el color más fuerte de los tres canales. Indica que tan brillante es el pixel.

$$V = \max(R, G, B)$$

- **Cálculo de S (Saturación):** Mide que tan vivo o apagado es el color. Si todos los canales son iguales, la saturación es 0.

$$S = \frac{V - \min(R, G, B)}{V}.$$

- **Cálculo de H (Tono):** es el color básico en el círculo cromático (0° a 360°) que se calcula según cuál canal predomina, usando el valor V (Brillo). El resultado es un ángulo según el color: 0° = Red, 120° = Green y 240° = Blue. Y ángulos intermedios como combinaciones de estos colores.

$$H = \begin{cases} \frac{60(G - B)}{V - \min(R, G, B)} & \text{Si } V = R, \\ \frac{120 + 60(B - R)}{V - \min(R, G, B)} & \text{Si } V = G, \\ \frac{240 + 60(R - G)}{V - \min(R, G, B)} & \text{Si } V = B, \\ 0 & \text{Si } R = G = B. \\ H = H + 360 & \text{Si } H < 0 \text{ (negativo), para que este dentro del círculo.} \end{cases}$$

- Con lo anterior, los valores de los canales HSV se encuentran en los rangos $0 \leq V \leq 1$, $0 \leq S \leq 1$ y $0 \leq H \leq 360$. Estos valores se convierten luego al rango de 8 bits utilizado en las imágenes, es decir, en una escala de valores de píxel de 0 a 255.

La conversión se realiza de la siguiente manera: $V = 255V$, $S = 255S$ y $H = \frac{H}{2}$.

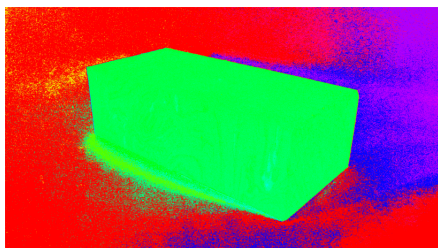
La Figura 4-4 ilustra este proceso de conversión HSV, sobre el cual se llevará a cabo el proceso posterior de segmentación y detección.



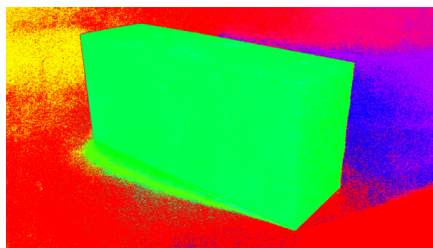
Imagen original RGB - A



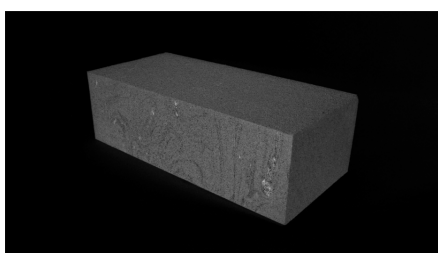
Imagen original RGB - B



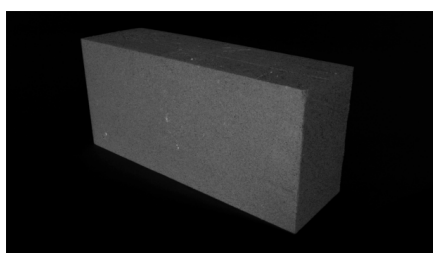
Canal H (Tono) - A



Canal H (Tono) - B



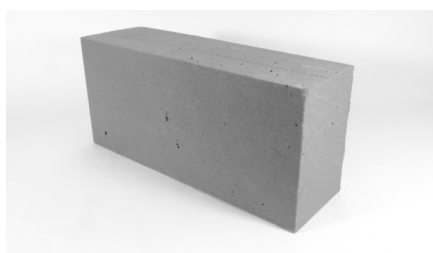
Canal S (Saturación) - A



Canal S (Saturación) - B



Canal V (Brillo) - A

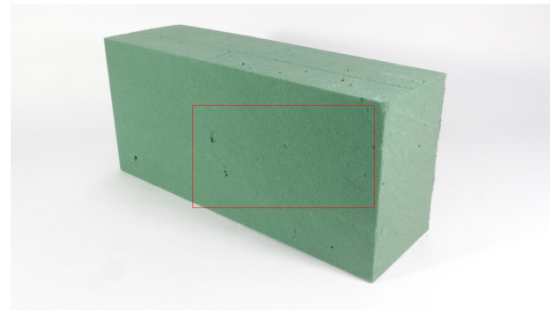
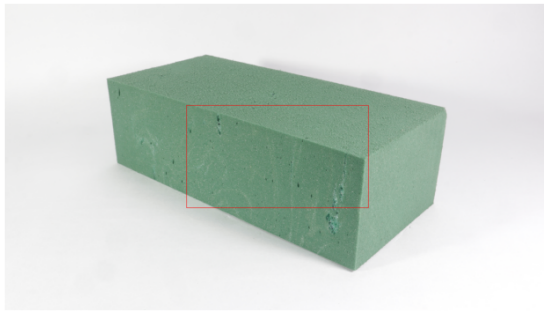


Canal V (Brillo) - B

Figura 4-4.: Imagen RGB y canal HSV de una espuma (Lado A y Lado B). La columna izquierda muestra la imagen original y los canales H, S y V para el lado A, mientras que la columna derecha muestra la misma información para el lado B.

Segmentación y Detección: Se sigue un procedimiento que permite detectar automáticamente en cada imagen las coordenadas del objeto (espuma) y calcular las dimensiones del recorte, para ello los pasos seguidos fueron (OpenCV, [s.f.](#)):

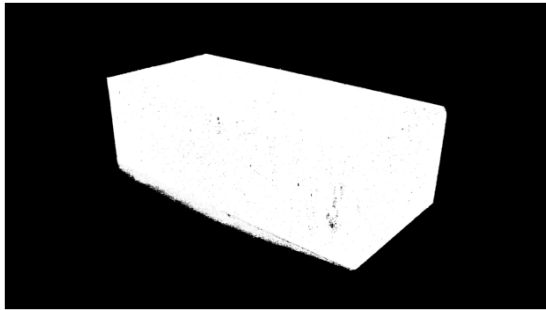
1. Se obtienen las dimensiones de la imagen original presentada en la Figura 4-4 con el fin de calcular umbrales de área relativa y establecer los límites espaciales del procesamiento.
2. Se realiza un muestreo en la región central de la imagen utilizando un rectángulo, como se muestra en la Figura 4-5. Esto se debe a que, debido a las condiciones de captura, se garantiza que el objeto de interés (la espuma) está cercano al centro de la imagen. Este rectángulo se emplea para estimar el color de referencia.



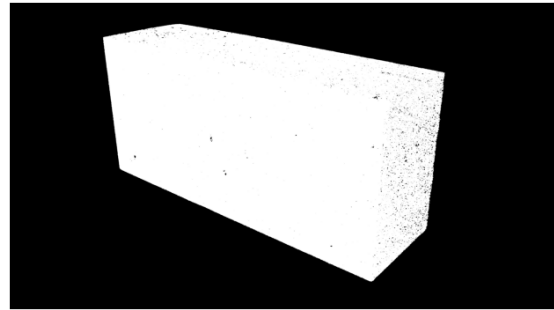
Muestreo en el centro de la imagen (Lado A) Muestreo en el centro de la imagen (Lado B)

Figura 4-5.: Muestreo central para estimar el color de referencia en las imágenes.

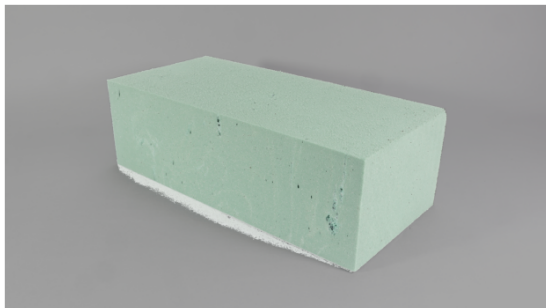
3. El color central del objeto se estima mediante la mediana de los valores en el espacio HSV, lo que permite reducir el efecto de valores atípicos y ruido en los canales de tono, saturación y brillo.
4. A partir de la mediana obtenida, se define un rango dinámico en HSV considerando tolerancias superiores e inferiores para capturar las variaciones naturales del color del objeto. Se genera una máscara binaria con la función `cv2.inRange`, donde los píxeles dentro del rango definido se marcan como blancos (valor 255) y los fuera del rango como negros (valor 0). Esta máscara se superpone a la imagen original para delimitar el objeto de interés, como se muestra en la Figura 4-6.



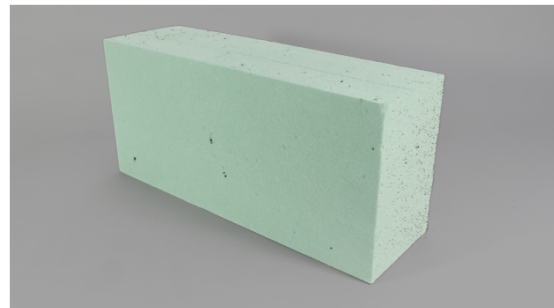
Máscara binaria en la imagen (Lado A)



Máscara binaria en la imagen (Lado B)



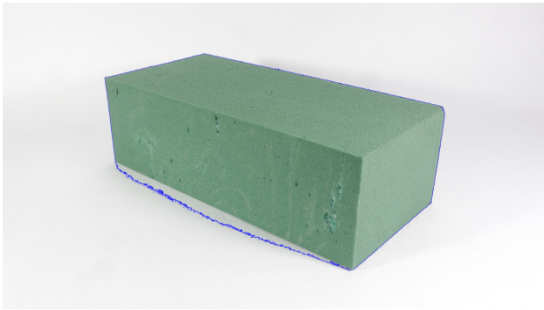
Máscara superpuesta sobre la imagen original (Lado A)



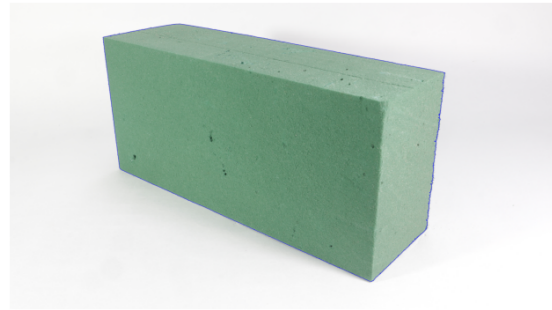
Máscara superpuesta sobre la imagen original (Lado B)

Figura 4-6.: Generación de la máscara binaria y su superposición sobre la imagen original para delimitar el objeto de interés (Lado A y Lado B).

5. Sobre la máscara resultante, se aplicó la detección de contornos externos utilizando la función `cv2.findContours` en modo `RETR_EXTERNAL`, que recupera únicamente los contornos externos. De los contornos obtenidos, se selecciona el de mayor área (contorno principal) y se aplica una restricción de área mínima equivalente al 1 % del área total de la imagen, con el fin de descartar falsos positivos, como se muestra en la Figura 4-7.



Contorno externo de la imagen (Lado A)



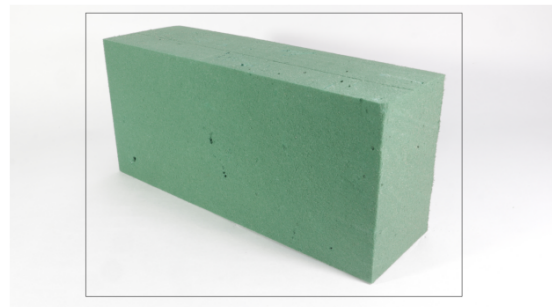
Contorno externo de la imagen (Lado B)

Figura 4-7.: Detección del contorno externo del objeto en las imágenes (Lado A y Lado B).

6. Finalmente, se calcula el **rectángulo delimitador** del contorno principal. Dicho rectángulo se expande en un 5 % sobre cada eje para evitar recortes demasiado ajustados, y se acota a los límites de la imagen. En la Figura 4-8 se muestra el rectángulo delimitador sobre la imagen original, antes de aplicar el recorte final.



Rectángulo delimitador sobre la imagen original (Lado A)



Rectángulo delimitador sobre la imagen original (Lado B)

Figura 4-8.: Rectángulo delimitador calculado sobre la imagen original, antes de aplicar el recorte final (Lado A y Lado B).

En la Figura 4-9 se muestra la imagen original junto con el resultado del recorte, realizado tras aplicar el proceso descrito anteriormente.



Imagen original Lado A



Imagen original Lado B



Imagen recortada Lado A



Imagen recortada Lado B

Figura 4-9.: Imagen original y resultado del recorte (Lado A y Lado B) después del proceso de detección y delimitación.

Formato de guardado: El archivo resultante se guarda en Google Drive y se genera un archivo CSV (`rutas_recortes`) con las rutas correspondientes, siguiendo la misma estructura utilizada para la carga de las imágenes originales. Para preservar la calidad de la imagen recortada, se opta por utilizar el formato PNG, que es un formato de compresión sin pérdida y garantiza que la calidad original de la imagen se mantenga intacta (Gonzalez & Woods, 2018, p. 554). Las imágenes recortadas serán las que se utilicen a partir de este momento en el proceso.

4.4 División del conjunto de datos

Para la división del conjunto de datos se tomaron las rutas del archivo CSV `rutas_recortes`, se asignó el 80 % de las imágenes para el entrenamiento (304 pares) y el 20 % restante para la prueba final (76 pares). Este proceso se hace utilizando la librería `scikit-learn` que es usada para aplicar modelos de machine learning con herramientas para preprocesamiento, entrenamiento, evaluación y selección de modelos (S.-I. Developers, 2021a). Para garantizar una representación equilibrada de las cuatro clases, se emplea la función `train_test_split` del módulo `sklearn.model_selection`. Esta función incluye el parámetro `stratify`, que permite realizar una estratificación según la clase (S.-I. Developers, 2021c). De este modo, el conjunto de datos se divide de manera proporcional, obteniéndose 76 pares para entrenamiento y 19 pares para prueba en cada clase.

Para esta división de datos, también se generan archivos CSV de `rutas_train` y `rutas_test`, cada uno de ellos contiene la información correspondiente a las rutas de las imágenes siguiendo la estructura de las secciones anteriores.

4.5 Validación cruzada k-fold

Para asegurar un mejor control y precisión en los resultados, se implementa una estrategia de validación cruzada con $k=5$ *folds* para los datos de entrenamiento, que representan el 80 % del total de las imágenes. Para ello, se utilizó nuevamente la librería `scikit-learn` y el módulo `sklearn.model_selection`; en este caso, la función `StratifiedKFold`, que permite realizar una validación cruzada estratificada, garantizando una representación adecuada de cada clase en los *folds*.

Las 304 parejas de imágenes del conjunto de entrenamiento se dividieron en cinco *folds* estratificados. En cada partición, entre 243 y 244 parejas se emplearon para entrenamiento y entre 60 y 61 parejas para validación, manteniendo la proporción de las cuatro clases. Asimismo, cada *fold* conservó una distribución equilibrada por clase, con aproximadamente 60 o 61 parejas por clase en el conjunto de entrenamiento y entre 14 y 16 parejas por clase en el conjunto de validación, presentando únicamente variaciones mínimas propias del proceso de estratificación.

El objetivo es realizar, para cada *fold*, un procesamiento completo de las imágenes, con el fin de identificar qué características de los métodos de extracción utilizados son las que mejor discriminan

entre las clases.

4.5.1 Preprocesamiento

Para cada *Fold* del proceso, se lleva a cabo un preprocesamiento de las imágenes, que consiste en:

- Convertir la imagen original de formato RGB (con 3 canales de color) a una escala de grises, lo que reduce la imagen a solo 2 canales.
- Aplicar un filtro gaussiano para suavizar la imagen, reduciendo el ruido y resaltando las características importantes de la misma.

Este proceso de preprocesamiento se ejecuta utilizando las bibliotecas `pandas`, `numpy`, `OpenCV` y `PIL` mencionadas anteriormente. Además, se hace uso de la librería `matplotlib.pyplot` para la visualización de los resultados obtenidos después de cada transformación, permitiendo comparar las imágenes originales con las preprocesadas (Hunter & Contributors, 2021).

Conversión a grises:

Dado que la imagen original está en formato RGB, es decir, cuenta con tres canales de color, en los cuales, como se observa en la Figura 4-2, predomina un color unificado en tonos de verde, y considerando que este color no constituye una característica relevante para la clasificación en el planteamiento del modelo, se decidió convertir la imagen a escala de grises.

Para ello, se utilizó la librería `OpenCV`, empleando la función `cv2.cvtColor` y el parámetro de conversión `cv2.COLOR_BGR2GRAY`, que aplica una fórmula de conversión utilizando una ponderación para cada canal BGR, de la siguiente manera (O. Contributors, 2021b):

$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B$$

Filtro gaussiano

Se utiliza un desenfoque gaussiano para reducir el ruido y resaltar características en las imágenes (Gonzalez & Woods, 2018, Sección 3.4). Para ello, se emplea la librería `OpenCV` con la función `cv2.GaussianBlur`, que realiza una convolución de la imagen con un kernel gaussiano. Este kernel es una versión discretizada de la función gaussiana, que pondera los píxeles de la imagen

según su distancia al centro del filtro (O. Contributors, 2021c).

La aplicación de estos kernels debe ser en forma de cuadrados con tamaños de valores impares. Se realizaron pruebas con kernels de tamaños: 5x5, 9x9 y 11x11, con el objetivo de observar cuál proporcionaba los mejores resultados en la posterior extracción de características. Finalmente, se seleccionó el kernel 11X11. En la Figura 4-10 se observa los resultados de los dos preprocesamientos.

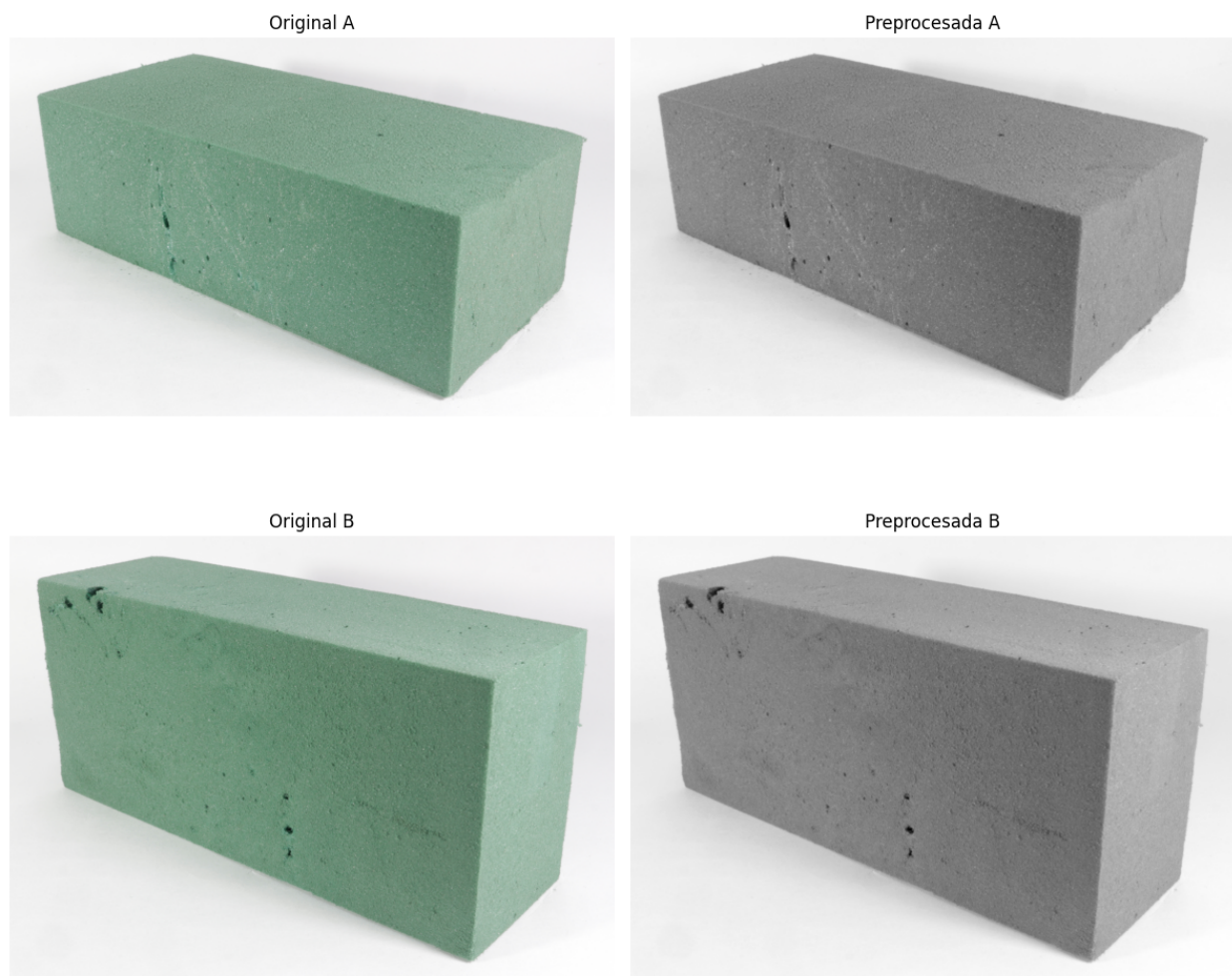


Figura 4-10.: Imágenes de las espumas antes y después del preprocesamiento (Lado A y Lado B).

4.5.2 Extracción de características

Con el objetivo de identificar los atributos que representan a cada una de las clases, se retoman los enfoques propuestos por Coşkun et al. (2022) y Nazir et al. (2021), quienes emplean la transformada discreta de wavelet para extraer características en sistemas de visión por computador aplicados a problemas de clasificación. Además, se considera el trabajo de Yousif et al. (2022), que propone complementar las transformadas wavelet con técnicas adicionales como la detección de bordes mediante el método de Canny, momentos invariantes y patrones locales binarios (LBP), con el fin de obtener descriptores más robustos.

Sin embargo, teniendo en cuenta las limitaciones computacionales y con el objetivo de mantener la complejidad del trabajo en un nivel manejable, se opta por utilizar la transformada discreta de wavelet como técnica principal de extracción de características, complementándola con la aplicación de patrones locales binarios (LBP).

Transformada discreta de wavelet

Se realiza la descomposición de la imagen preprocesada mediante la transformada discreta de wavelet (DWT), la cual permite descomponer la señal de la imagen en las subbandas LL, LH, HL y HH, extrayendo características en diferentes direcciones de la imagen (Zhang, 2019). Para este proceso, se utilizó la librería `PyWavelets` (Lee & et al., 2019), empleando la función `dwt2` (P. contributors, 2023). Se eligió la función de wavelet de Haar debido a su estructura simple y soporte compacto, lo que permite operar sobre vecindarios pequeños de píxeles y realizar un análisis estrictamente local de la imagen. Esta característica la hace adecuada para detectar cambios abruptos en la señal, facilitando la identificación de patrones granulares, como los poros, y texturas irregulares asociadas a betas. Además, esta wavelet emplea operaciones básicas de promedio y diferencia, lo que reduce el tiempo de procesamiento y el uso de recursos computacionales (Coşkun et al., 2022; Haar, 1910).

Se aplicó la DWT de primer nivel en ambas imágenes, A y B, como se muestra en la Figura 4-11. Pero dado que algunos detalles en las subbandas no son suficientemente perceptibles, se aplicó una descomposición de segundo nivel sobre la subbanda LL para obtener detalles más finos y mejorar la resolución de la imagen en las subbandas. Los resultados de esta segunda descomposición se muestran en la Figura 4-12.

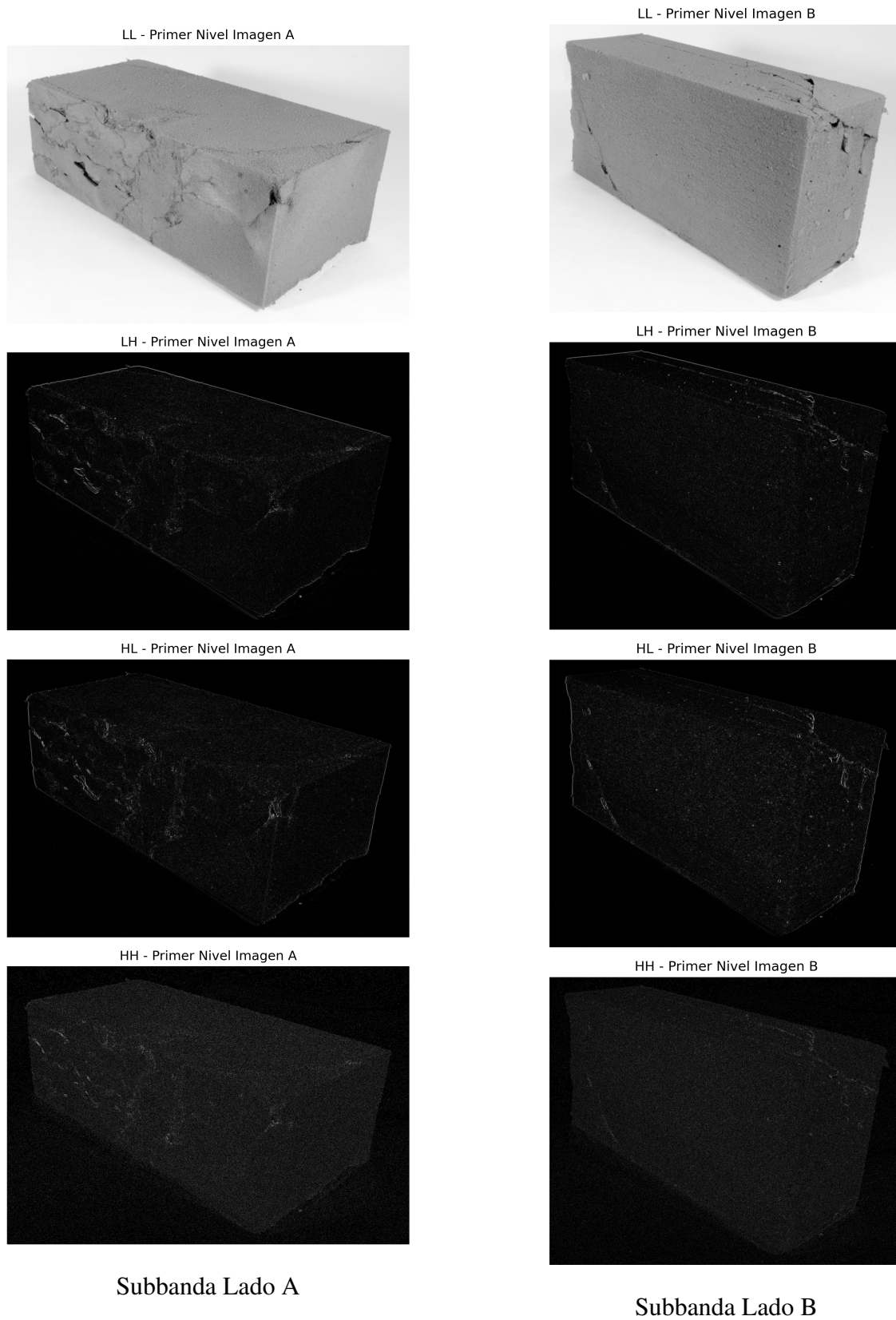


Figura 4-11.: DWT - Primer Nivel (Subbandas LL, LH, HL, HH) para los lados A y B.

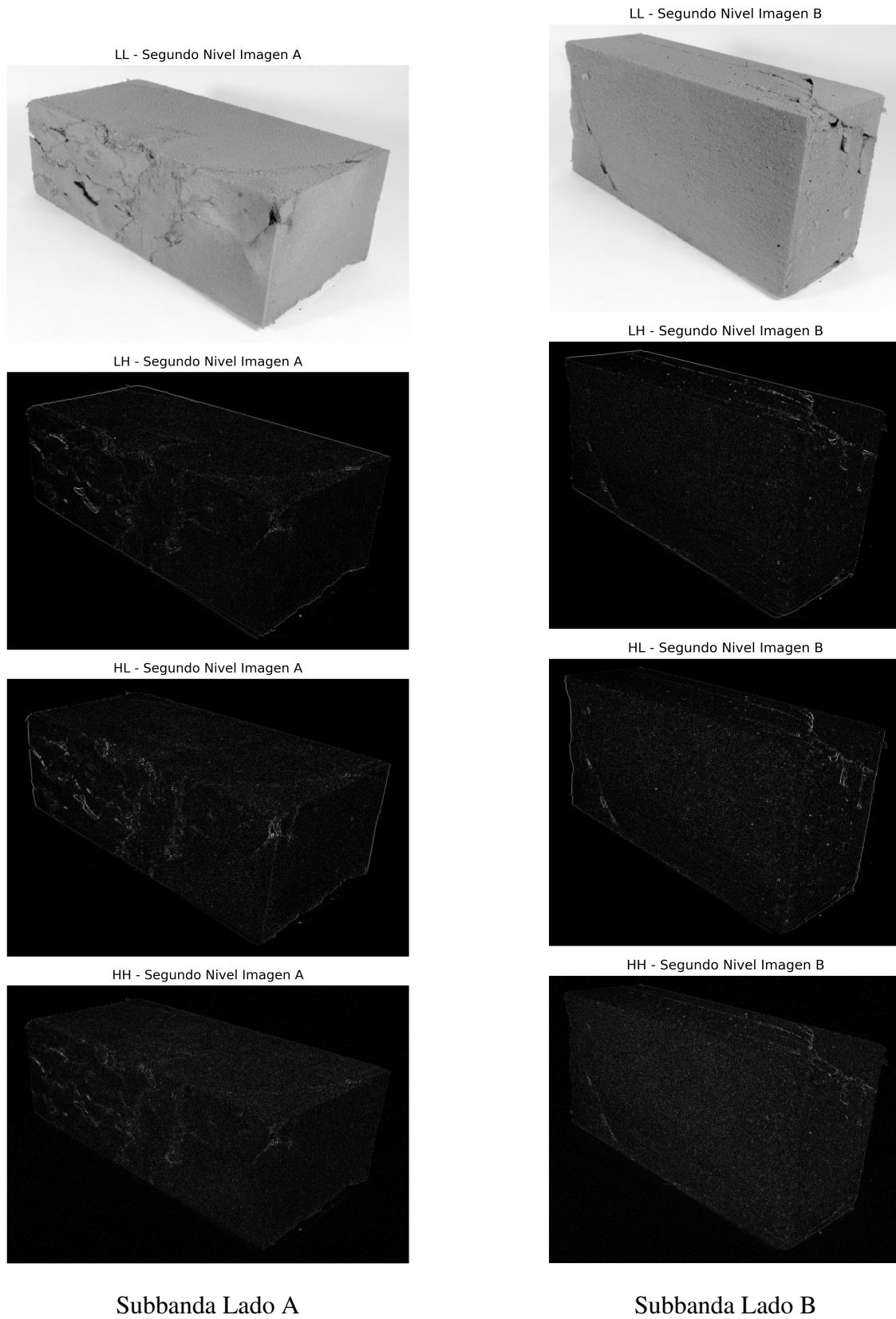


Figura 4-12.: DWT - Segundo Nivel (Subbandas LL, LH, HL, HH) para los lados A y B.

Extracción de características para DWT

A partir de las subbandas de segundo nivel, se utilizan los coeficientes de cada una para calcular métricas estadísticas globales como características descriptivas, tales como: media, varianza, energía y entropía. Los trabajos de Bhat et al. (2015) y Gonzalez y Woods (2018, Cap. 12) detallan la extracción de estas características para una subbanda de tamaño $M \times N$, donde C_{ij} representa el coeficiente en la posición i, j :

- Media: Nivel promedio de los coeficientes según la subbanda.

$$\mu = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N C_{ij}$$

- Varianza: Medida de dispersión respecto a la media.

$$\sigma^2 = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (C_{ij} - \mu)^2$$

- Energía: cantidad total de información (o potencia) contenida en la sub-banda.

$$E = \sum_{i=1}^M \sum_{j=1}^N C_{ij}^2$$

- Entropía: mide la cantidad de desorden o aleatoriedad en la distribución de coeficientes.

$$H = - \sum_{k=1}^K p_k \cdot \log_2(p_k)$$

En donde: p_k es el histograma de los coeficientes que mide la probabilidad de que un coeficiente tenga un valor k en el intervalo de valores de coeficientes.

Este cálculo se realizó usando la librería de `numpy`, ya que su manejo de arrays facilita las operaciones matemáticas de media, varianza y energía. Para el cálculo de la entropía, se empleó la librería `skimage` (van der Walt et al., 2014) específicamente la función `shannon_entropy` del módulo `measure` (S.-I. contributors, 2020). La Tabla 4-3 muestra el número de características obtenidas por subbanda y lado.

Tabla 4-3.: Número de características extraídas por DWT para clasificación

Lado	Coficiente	Número de características
A / B	LL	4
	LH	4
	HL	4
	HH	4
Total características por lado		16
Total de características para clasificación		32

Patrón Local Binario (LBP)

Una vez preprocesadas las imágenes, se calcularon las características *Local Binary Patterns* (LBP) usando la librería `scikit-image`, específicamente la función `local_binary_pattern` del módulo `skimage.feature`, configurada con el operador clásico (`method="default"`) y parámetros $(P, R) = (8, 1)$. Este operador corresponde al enfoque original de LBP propuesto por Ojala et al. (1996) y posteriormente extendido en trabajos posteriores para dotarlo de invarianza a escala y rotación (Ojala et al., 2002). En este procedimiento, cada píxel central I_c se compara con la intensidad de sus vecinos I_p ubicados sobre una circunferencia de radio $R = 1$; se asigna un 1 si $I_p \geq I_c$ y un 0 en caso contrario. Los ocho bits generados, ordenados en sentido horario, se interpretan como un entero en el rango $[0, 255]$, produciendo así el mapa LBP (imagen de etiquetas) que refleja los patrones binarios locales en cada región.

El uso del operador clásico por *default* se justifica porque constituye la formulación original del método, ampliamente empleada como referencia en la literatura para la descripción de texturas. A diferencia de variantes más avanzadas, este enfoque básico permite una interpretación directa del patrón binario local y proporciona un punto de comparación estandarizado para evaluar el desempeño de los clasificadores en estudios posteriores (Ojala et al., 1996, 2002).

Un ejemplo paso a paso del cálculo de LBP se ilustra en la Figura 4-13, tomada Ojala et al. (2002).

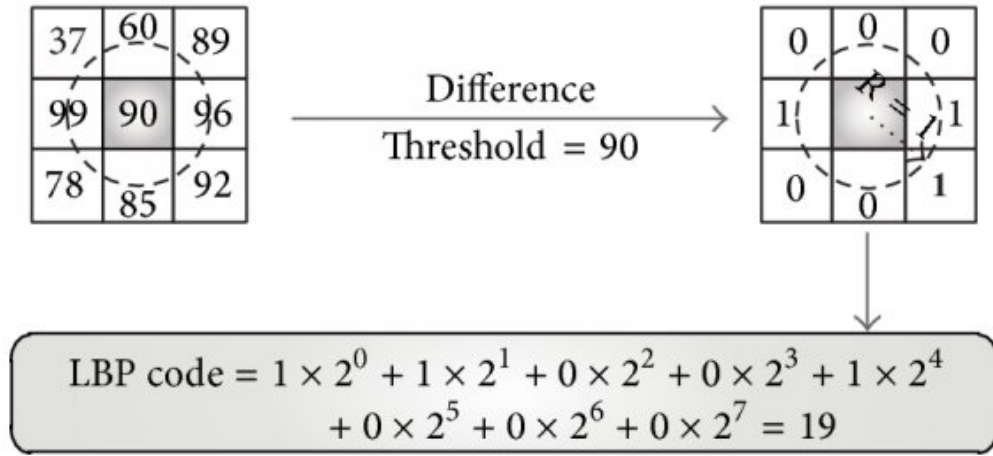


Figura 4-13.: Ejemplo del cálculo del código LBP ($P=8$, $R=1$): a la izquierda, intensidades en una vecindad 3×3 con píxel central $I_c=90$; a la derecha, binarización de vecinos (1 si $I_p \geq I_c$, 0 en otro caso) y lectura de bits en sentido horario. El patrón binario se convierte a entero mediante $\sum_{p=0}^7 b_p 2^p$ (código LBP = 19).

Extracción de características LBP: construcción del histograma

Para resumir la distribución de patrones se construyó, con NumPy (`numpy.histogram`), un histograma de 256 intervalos alineados con el rango $[0, 255)$ y normalizado por densidad (ancho de bin unitario). Con esta configuración, cada componente h_k corresponde a la frecuencia relativa del patrón k (equivalente a $h_k = n_k/N$), cumpliéndose $\sum_{k=0}^{255} h_k \approx 1$ (posibles diferencias por redondeo). El vector $\mathbf{h} = (h_0, \dots, h_{255})$ se utilizó como descriptor de textura por imagen. Una comparación visual de dos muestras (imágenes originales, mapas LBP y sus histogramas normalizados) se presenta en la Figura 4-14.

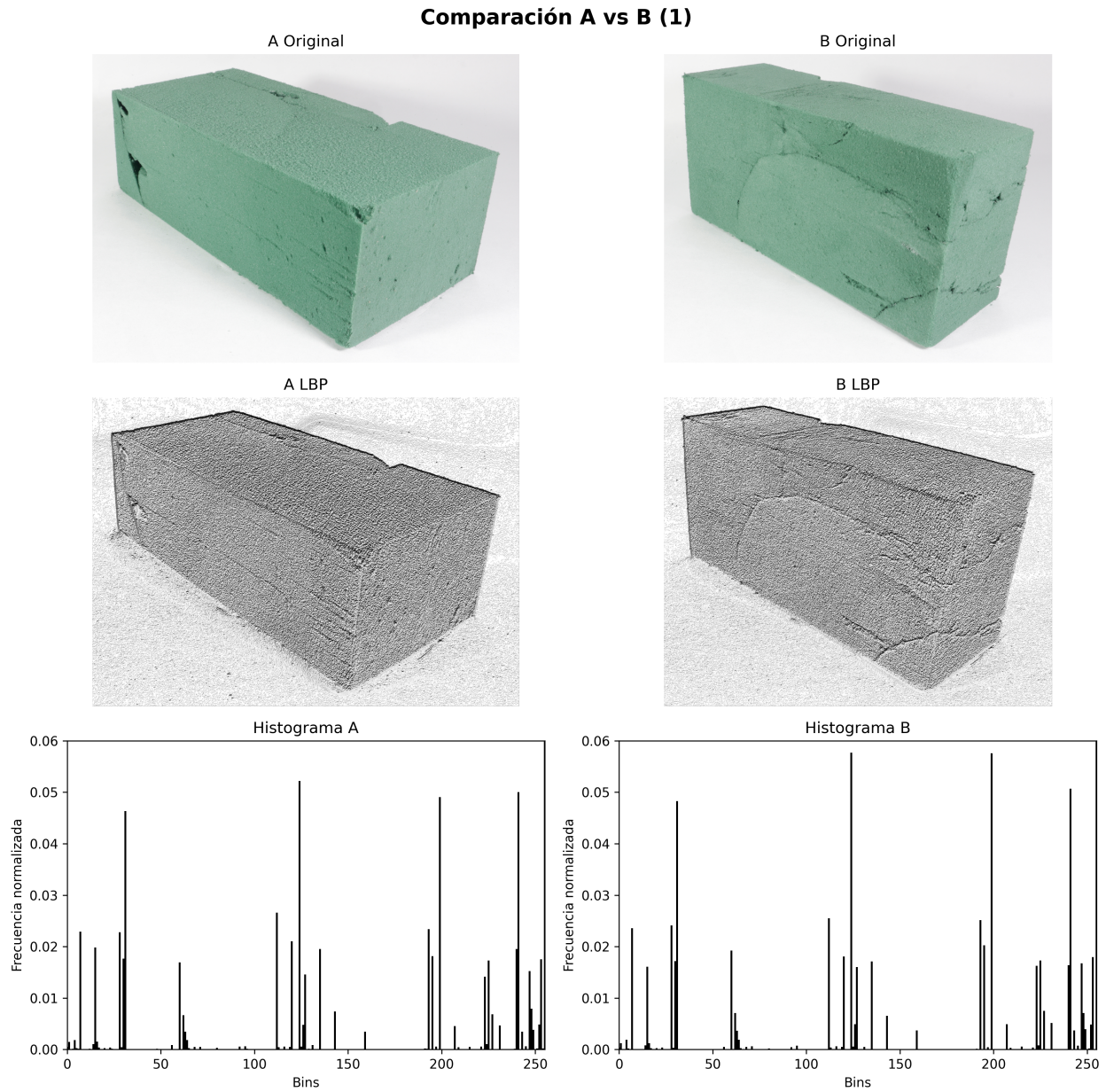


Figura 4-14.: Comparación A vs B. Fila superior: imágenes originales. Fila intermedia: mapas LBP con $(P, R) = (8, 1)$, operador clásico. Fila inferior: histogramas normalizados de 256 bins (frecuencia relativa) para cada imagen; misma escala en el eje y para facilitar la comparación.

4.5.3 Selección de características

La selección de características se llevó a cabo mediante el método *Recursive Feature Elimination* (RFE), una técnica que elimina de forma iterativa las variables menos relevantes, evaluando su importancia según la capacidad predictiva del modelo. A diferencia de otros procedimientos de selección, el RFE considera en cada iteración tanto el aporte individual de cada variable como su interacción con el resto del conjunto (Guyon et al., 2002).

Las características empleadas en este proceso provinieron de dos fuentes: *Transformada Wavelet* y *Local Binary Patterns (LBP)*. En total, se conformó un conjunto de 544 características, distribuidas de la siguiente manera: 512 provenientes de LBP (histogramas de 256 bins por cada lado A/B) y 32 de Wavelet (4 subbandas $\times$ 4 métricas para cada lado A/B). Posteriormente, las características fueron estandarizadas mediante la función `StandardScaler` del módulo `preprocessing` de la librería `Scikit-learn`, garantizando que todas presentaran media cero y desviación estándar igual a uno. Este paso fue esencial para que ninguna variable dominara las métricas de importancia debido a diferencias en su escala.

Metodología de selección (RFE)

La selección de características se implementó mediante la función *RFE*, disponible en el módulo `feature_selection` de la librería `Scikit-learn` (S.-I. Developers, 2021a). Este procedimiento requiere definir un modelo base (*estimator*), a partir del cual se determina la importancia relativa de cada variable. El proceso puede resumirse en los siguientes pasos:

- Ajustar un modelo base sobre el conjunto completo de características.
- Estimar la importancia de cada variable a partir de los coeficientes o pesos asignados por el modelo.
- Eliminar de forma iterativa las variables menos relevantes hasta alcanzar el número de características definido.

El algoritmo se aplicó de manera independiente en cada uno de los cinco *folds* de la validación cruzada, generando subconjuntos de **10, 15 y 20 características**. Estos tamaños se establecieron considerando tanto las limitaciones computacionales como la cantidad de observaciones disponibles.

4.5.4 Selección de características e hiperparámetros - Máquina de Soporte Vectorial (SVM)

La selección de características se llevó a cabo mediante la función `RFE`, utilizando como estimador un modelo SVM implementado con la clase `SVC` del módulo `svm` de `Scikit-learn` (Pedregosa et al. (2011); S.-l. Developers (s.f.)). Para este procedimiento se empleó exclusivamente un **kernel lineal**, dado que es el único que permite obtener de manera directa los coeficientes del hiperplano separador, indispensables para calcular la importancia de cada variable y, en consecuencia, realizar la selección de características (Guyon et al., 2002).

El procedimiento de selección se aplicó en cada *fold* de la validación cruzada, obteniéndose tres subconjuntos de características por cada partición. En cada caso, se generó un archivo `.csv` que contenía las 544 características originales e indicaba cuáles de ellas fueron seleccionadas, así como el ranking asignado por el método *RFE*. Dicho ranking es definido de la siguiente manera:

- **Ranking = 1:** variables seleccionadas e incluidas en el subconjunto final.
- **Ranking > 1:** variables no seleccionadas, definidas y ordenadas según el momento en que fueron eliminadas durante el proceso iterativo *RFE*.

El proceso de decisión se estructuró en dos etapas:

1. **Determinación del número de características:** Para cada subconjunto generado se entrenaron modelos en la validación cruzada, evaluando su desempeño con diferentes valores de los hiperparámetros ($C = \{0.01, 0.1, 1, 10, 100\}$ y $\text{kernel} = \{\text{lineal}, \text{polinómico}, \text{radial}\}$). Los subconjuntos fueron comparados y se seleccionó el tamaño de características que mostró el mejor rendimiento en las métricas de macro-promedio.
2. **Selección de las características finales:** Con el número de variables definido en el paso anterior, se consolidó la información de los archivos de ranking. La decisión sobre qué variables integrar se basó en los siguientes criterios:
 - a) **Frecuencia de aparición:** se priorizaron las variables seleccionadas de manera consistente en la mayor cantidad de *folds*, considerándolas como más estables y representativas.
 - b) **Promedio del ranking:** en caso de empate en la frecuencia, se seleccionaron aquellas variables con menor valor promedio en el ranking asignado por el *RFE*.

4.5.5 Selección de características e hiperparámetro C - Regresión Logística Multinomial (RLM)

En la regresión logística multinomial, los hiperparámetros involucrados incluyen el coeficiente de regularización C , el tipo de penalización, la estrategia multiclase y el solucionador numérico. En este estudio, algunos de estos hiperparámetros se fijaron de manera previa según criterios teóricos y de estabilidad del modelo: se empleó penalización ℓ_2 , equivalente a la regularización Ridge, para controlar la magnitud de los coeficientes y evitar sobreajuste; la estrategia `multi_class='multinomial'`, que modela directamente las probabilidades de cada clase mediante la función *softmax*; y el solucionador `lbfgs`, seleccionado por su buen desempeño en problemas multiclase y su estabilidad numérica. El único hiperparámetro del modelo ajustado durante la validación fue el coeficiente de regularización C , cuyo valor óptimo se determinó junto con el número de características seleccionadas. Esta configuración resulta adecuada para la tarea de clasificación abordada y permite mantener la interpretabilidad de los coeficientes asociados a cada variable.

El hiperparámetro C y el número de características se determinaron de forma conjunta mediante una búsqueda en rejilla con validación cruzada interna, acoplando RFE al clasificador sobre la matriz ya estandarizada. En cada partición interna se exploraron simultáneamente valores de $C = \{0.01, 0.1, 1, 10, 100\}$ y tamaños de subconjunto (10, 15 y 20), empleando el F1-Score, exactitud (accuracy), sensibilidad y especificidad a nivel macro como métricas para comparar configuraciones. Esta configuración se aplicó únicamente en la etapa de selección de variables, con el fin de identificar la combinación óptima de regularización y número de características antes del entrenamiento final del modelo.

Para construir el ranking final en la regresión logística multinomial, se siguió un procedimiento orientado a garantizar la estabilidad y relevancia de las características seleccionadas a través de los distintos *folds* de validación. La idea fue identificar un conjunto de variables que no solo resultaran importantes para el clasificador en un *fold* particular, sino que mantuvieran su relevancia de manera consistente a lo largo de todas las particiones, reduciendo así la dependencia del modelo a posibles fluctuaciones en los datos de entrenamiento.

El procedimiento aplicado fue el siguiente:

- En cada *folds*, RFE produjo un ranking y se registró una variable binaria (seleccionada/no seleccionada).

- Se ajustó un modelo en cada *fold* para obtener las importancias mediante la norma ℓ_2 de los coeficientes asociados a cada variable (agregada sobre clases).

Con las importancias obtenidas en los cinco *folds*, se calcularon *frecuencia de aparición*, *mediana*, *media* y *desviación estándar* para cada característica. Estos indicadores se priorizaron en el ranking global de la siguiente forma:

1. **Frecuencia de aparición:** se privilegió que la característica fuese seleccionada en la mayoría de los *folds*, pues esto refleja estabilidad y consistencia en diferentes particiones de los datos.
2. **Mediana de importancia:** entre las variables con igual frecuencia, se ordenaron según la mediana de sus importancias, ya que esta medida es robusta frente a valores atípicos y refleja el comportamiento típico de la variable.
3. **Media de importancia:** se consideró como complemento de la mediana, ya que aporta información sobre la **magnitud global** de la relevancia de cada característica.
4. **Desviación estándar:** finalmente, se dio preferencia a las variables con menor variabilidad entre *folds*, lo que indica una importancia más estable a lo largo de las validaciones.

Este procedimiento permitió consolidar un conjunto de 15 características estables y relevantes, junto con un valor de regularización $C = 10$, garantizando un modelo final consistente y con alta capacidad discriminatoria.

4.5.6 Entrenamiento de modelos para validación cruzada

Entrenamiento con Máquina de Soporte Vectorial

Siguiendo los trabajos de Coşkun et al. (2022) y Nazir et al. (2021), que emplean SVM para la clasificación de defectos en sistemas de visión multiclase, se adoptó un enfoque similar para nuestro conjunto de datos. Las SVM son algoritmos supervisados que buscan un hiperplano que separe de forma óptima las clases, maximizando el margen entre los elementos más cercanos de cada clase James et al. (2013). En particular, se sigue el enfoque de Coşkun et al. (2022), que aplica la estrategia “Uno contra el resto” (One vs Rest) para gestionar el problema multiclase. Este enfoque consiste en clasificar de forma iterativa, donde en cada paso se selecciona una clase como referencia y se compara con las demás clases restantes.

Se entrenaron modelos SVM considerando los diferentes números de características seleccionadas (10, 15 y 20) en cada uno de los cinco *folds*. En cada entrenamiento, se realizó una búsqueda sistemática de hiperparámetros para definir el tipo de kernel y el valor de penalización C con el mejor rendimiento. Se evalúa el desempeño con kernels lineal y no lineal como el polinómico y radial, dado que estos kernels pueden capturar relaciones más complejas entre variables. Asimismo, se probaron distintos valores del parámetro de regularización $C = [0.1, 1, 10, 100]$, lo que permitió analizar la sensibilidad del modelo frente a márgenes de error más estrictos o más flexibles.

Para la optimización de hiperparámetros se empleó la función `GridSearchCV` de la librería `scikit-learn` (Pedregosa et al. (2011); S.-l. Developers (2021c)). Esta función implementa un proceso de validación cruzada interna en cada *fold*, utilizando el parámetro `cv=k` para dividir el conjunto de entrenamiento en $k - 1$ particiones para el ajuste del modelo y una partición para su validación. Posteriormente, el mejor modelo resultante de esta búsqueda se evalúa en el *fold* de validación externo. En este trabajo se definió `cv = 3` para la optimización de hiperparámetros, lo que permitió estimar el rendimiento de los modelos en diferentes particiones del conjunto de datos de entrenamiento.

Los resultados de cada entrenamiento se registraron en archivos `.csv`, incluyendo toda la información necesaria para comparar el desempeño en función del número de características seleccionadas, el tipo de kernel y el valor de C evaluado.

Entrenamiento el Modelo logístico multinomial

Definidos el subconjunto específico de 15 características y el valor de $C = 10$ mediante la optimización conjunta anterior, se evaluó el comportamiento del clasificador con validación cruzada estratificada de cinco *folds* (etapa no final). En cada iteración, se restringió la matriz estandarizada a esas 15 variables, se entrenó la regresión logística multinomial con el valor de C ya fijado usando únicamente los datos del *fold* de entrenamiento y se evaluó en el *fold* de validación. Las métricas por clase (Sensibilidad, Especificidad, F1-Score y Accuracy global) se registraron por *fold*, calculando su macropromedio al final, sin realizar otras agregaciones estadísticas.

Este procedimiento tuvo como propósito validar la estabilidad y consistencia del modelo ya configurado, más que volver a explorar configuraciones de hiperparámetros. En otras palabras, se buscó observar el comportamiento del clasificador en cada partición de los datos, asegurando que su rendimiento no dependiera excesivamente de un conjunto específico de entrenamiento. Con esta

evaluación se obtuvo una visión más detallada del desempeño del modelo previo al entrenamiento final con todo el conjunto de datos disponible (Bishop, 2006).

4.5.7 Métricas de comparación

Se comparan los resultados obtenidos en los *folds* de entrenamiento de los modelos **SVM** y **logístico multinomial** según sus hiperparámetros considerados. El objetivo en cada caso es definir las configuraciones definitivas para el entrenamiento de los modelos finales.

Para evaluar y comparar la capacidad de clasificación de cada modelo en los *folds* y en los modelos finales, se emplean métricas estadísticas que cuantifican el rendimiento en la detección de defectos. Estas métricas se derivan de la matriz de confusión, que define cómo se evalúa la correcta clasificación de cada defecto. Dada la característica multiclase de los modelos entrenados, la matriz de confusión extiende el concepto de la matriz de confusión binaria para manejar, en este caso, cuatro clases.

La matriz de confusión binaria solo contempla dos categorías: positivas o negativas. En la matriz multiclase, sin embargo, se tiene una tabla de dimensiones $k \times k$, donde $k = 4$ es el número total de clases. Esto significa que no solo se comparan dos categorías, sino que se evalúan todas las clases posibles dentro del conjunto de datos. Cada celda de la matriz indica cuántas veces el modelo ha asignado una clase predicha a una clase real específica.

Celda (i, j) : Indica el número de veces que el modelo ha predicho la clase i cuando la clase real era j .

- Si $i = j$, indica una correcta clasificación en la clase i . (Verdadero Positivo).
- Si $i \neq j$, indica un error de clasificación, es decir, el modelo predijo la clase i cuando en realidad la instancia pertenecía a la clase j .

Para la clasificación de los cuatro defectos en espumas florales (B, BTC, BTE, PO), la Tabla 4-4 presenta un ejemplo para el defecto B (Espuma Buena) y sus respectivos valores de VP, FN, FP y VN. Estos valores se calculan utilizando la función `confusion_matrix` del módulo `metrics` de la librería `scikit-learn` (S.-I. Developers, 2021b).

Tabla 4-4.: Matriz de Confusión Multiclase para B (Espuma Buena)

		Predicción			
		B	BTC	BTE	PO
Valores Reales	B	VP	FN	FN	FN
	BTC	FP	VN	VN	VN
	BTE	FP	VN	VN	VN
	PO	FP	VN	VN	VN

En la 4-4, los valores observados en la matriz corresponde a:

- Verdadero positivo (VP): Predicción positiva cuando el valor real es positivo. (Clasificación correcta)
- Falso positivo (FP): Predicción positiva cuando el valor real es negativo. (Error Tipo I)
- Verdadero negativo (VN): Predicción negativa cuando el valor real es negativo. (Clasificación correcta)
- Falso negativo (FN): Predicción negativa cuando el valor real es positivo. (Error Tipo II)

A partir de estos valores obtenidos para cada defecto, se calculan las métricas de evaluación para cada clase utilizando el módulo `metrics` de la librería `scikit-learn` (S.-I. Developers, 2021b). Las métricas utilizadas serán:

Métricas binarias

- **Accuracy (exactitud global):** Proporción de predicciones correctas respecto al total de observaciones.
- **Sensibilidad (Recall):** Capacidad del modelo para identificar correctamente los casos positivos.
- **Especificidad:** Capacidad del modelo para identificar correctamente los casos negativos.
- **F1-score:** Media armónica entre precisión y sensibilidad, útil para evaluar el equilibrio entre falsos positivos y falsos negativos.

Métricas multiclase: Dada la naturaleza multiclase del problema, las métricas de evaluación originalmente definidas para clasificación binaria se extienden mediante los siguientes enfoques:

- **Macro-promedio:** Permite evaluar el desempeño del modelo otorgando igual importancia a cada clase, independientemente de su frecuencia. Esta métrica facilita identificar si el modelo mantiene un rendimiento equilibrado entre categorías o si existen clases con un comportamiento significativamente inferior.

La evaluación de los modelos en la validación se realizó inicialmente utilizando métricas multiclase con promedio macro, calculando cada métrica por clase y promediándolas para obtener una valoración global y equilibrada del desempeño del modelo. Este enfoque es especialmente adecuado dado que las clases se encuentran balanceadas, y permite identificar de manera general si el modelo mantiene un rendimiento uniforme entre todas las categorías. En casos donde se requiera un análisis más detallado, las métricas también se calcularon de manera individual por clase, lo que facilita detectar posibles diferencias de desempeño específicas entre categorías.

4.6 Entrenamiento final

4.6.1 Entrenamiento final de los modelos

Con los hiperparámetros ya definidos y los subconjuntos de características determinados para cada estimador, se procedió al entrenamiento de los dos modelos finales de manera independiente. En ambos casos, el ajuste se realizó utilizando la totalidad del conjunto de entrenamiento (304 pares de imágenes) y replicando de forma estricta la configuración establecida en las fases previas, incluyendo el procesamiento de datos, la extracción de características, los parámetros de entrenamiento y el número y tipo de variables seleccionadas.

El entrenamiento final de **SVM** se realizó empleando únicamente sus 15 variables seleccionadas para dicho modelo; de forma análoga, el entrenamiento final de la **regresión logística multinomial** se llevó a cabo con las 15 variables específicas seleccionadas para este estimador.

4.6.2 Evaluación y comparación de resultados

La selección del mejor clasificador se realizó mediante una evaluación integral que consideró todas las métricas definidas anteriormente (*Accuracy*, *sensibilidad*, *especificidad* y *f1-score*), tanto a

nivel macro como por clase, complementada con el análisis de la matriz de confusión. Dado que se evaluaron cuatro clases, la decisión se basó en el modelo que mostró el mejor desempeño conjunto y más equilibrado en las clases, evitando sesgos hacia alguna categoría y minimizando las confusiones entre ellas.

5. Resultados

En esta sección se presentan los resultados obtenidos mediante validación cruzada con 5 *folds* para la selección del número de características (10, 15 y 20) utilizando el método *Recursive Feature Elimination* (RFE), aplicado a los modelos de Máquina de Soporte Vectorial (SVM) y Regresión Logística Multinomial. Asimismo, se muestran los resultados que sustentan la elección de la configuración óptima de hiperparámetros para cada modelo final. Finalmente, se exponen los resultados del entrenamiento definitivo de ambos modelos y se compara su rendimiento en la clasificación de defectos en espumas florales.

5.1 Resultados de los fold en la Validación Cruzada

5.1.1 Selección de características e hiperparámetros - SVM

A partir del conjunto inicial de 544 características, se realizó un proceso de selección de variables mediante el método RFE (Eliminación Recursiva de Características) empleando un clasificador SVM con *kernel* lineal, obteniéndose subconjuntos de 10, 15 y 20 características más relevantes. Con estos subconjuntos, se aplicó la función `GridSearchCV` para evaluar diferentes configuraciones del modelo, considerando los tipos de *kernel* (lineal, radial y polinómico) y valores del parámetro de penalización del margen C (0.1, 1, 10 y 100). Los resultados correspondientes a todas las combinaciones posibles evaluadas se presentan en [Anexo B](#).

En la Tabla [5-1](#), se muestran los mejores desempeños obtenidos en cada uno de los *folds*, con base en el rendimiento macro de las métricas de evaluación. Se observa de manera general que la combinación de hiperparámetros más frecuente entre los *folds* corresponde al *kernel* radial con un valor de penalización $C = 10$. Las diferencias entre los valores de las métricas están principalmente asociadas al número de características seleccionadas. Además, la métrica macro de especificidad es la más estable y con valores más altos, alcanzando promedios cercanos a 0.93 tanto en la validación

interna como externa, lo que indica que el modelo mantiene una buena capacidad para identificar correctamente los casos negativos (evitando falsos positivos), independientemente del número de características consideradas.

Tabla 5-1.: Resultados de la selección de características e hiperparámetros evaluados según los macro-promedios de las métricas en validaciones internas (Int.) y externas (Ext.).

Macro-Promedio				Accuracy		Sensibilidad		Especificidad		F1-score	
Nº	Fold	C	Kernel	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.
10	1	10	Radial	0,8560	0,8689	0,8563	0,8677	0,9519	0,9563	0,8541	0,8680
10	2	10	Radial	0,8971	0,9016	0,8976	0,9000	0,9657	0,9671	0,8971	0,9005
10	3	10	Radial	0,8848	0,9672	0,8851	0,9677	0,9616	0,9891	0,8849	0,9672
10	4	10	Radial	0,9136	0,9180	0,9133	0,9198	0,9712	0,9726	0,9135	0,9183
10	5	10	Radial	0,9096	0,9333	0,9101	0,9332	0,9699	0,9777	0,9087	0,9345
15	1	10	Radial	0,8930	0,8852	0,8933	0,8844	0,9643	0,9620	0,8928	0,8852
15	2	10	Radial	0,9342	0,9016	0,9351	0,9000	0,9781	0,9671	0,9346	0,9005
15	3	10	poly	0,9095	0,9344	0,9099	0,9344	0,9699	0,9781	0,9100	0,9343
15	4	10	Radial	0,9177	0,9344	0,9177	0,9354	0,9726	0,9781	0,9182	0,9349
15	5	10	Radial	0,9220	0,9167	0,9228	0,9153	0,9741	0,9721	0,9215	0,9162
20	1	10	Radial	0,8807	0,8852	0,8808	0,8854	0,9601	0,9620	0,8808	0,8857
20	2	10	Radial	0,9177	0,8852	0,9183	0,8833	0,9726	0,9617	0,9177	0,8832
20	3	10	Radial	0,8971	0,9344	0,8972	0,9344	0,9657	0,9781	0,8979	0,9329
20	4	10	Radial	0,9136	0,9016	0,9133	0,9031	0,9712	0,9673	0,9137	0,9019
20	5	10	Radial	0,8769	0,9167	0,8780	0,9153	0,9591	0,9721	0,8768	0,9162

A partir de los resultados mostrados, se realizó un análisis comparativo del comportamiento del modelo, considerando tanto el rendimiento y la estabilidad entre *folds* como las variaciones según el número de características seleccionadas.

En conjunto, los valores de las métricas en la Tabla 5-1 presentan una alta similitud tanto dentro de cada *fold* como entre los diferentes *folds*, lo que evidencia un comportamiento estable del modelo y un equilibrio entre los distintos macro-promedios de las métricas de desempeño (accuracy (exactitud), sensibilidad, especificidad y F1-score). Los valores de las métricas se mantienen en rangos estrechos, con gran parte de los resultados internos y externos entre aproximadamente 0.85 y 0.93.

Esta consistencia sugiere que el modelo logra mantener un rendimiento uniforme y balanceado entre clases, sin mostrar una dependencia marcada hacia una métrica o subconjunto específico de datos.

A partir de estos resultados globales, se realizó un análisis más detallado por número de características, cuyos principales hallazgos se resumen a continuación:

- Para 10 características se observa, particularmente en el *fold 3*, un comportamiento variable en donde para la mayoría de las métricas los valores en entrenamiento son considerablemente menores a los obtenidos en validación (por ejemplo, F1 interno = 0.8849 frente a F1 externo = 0.9672). De manera general, para este número de características los valores de las métricas son menores que los de los demás grupos en la mayoría de los *folds*.
- Con 15 características, los resultados en el *fold 2* muestran un leve sobreajuste en la mayoría de las métricas (Sensibilidad interno = 0.9351 frente a Sensibilidad externa = 0.9000). En el *fold 3* se observó además que el mejor desempeño se obtuvo con el *kernel* polinómico, aunque con diferencias mínimas respecto a los demás. En general, este grupo de características presentó resultados altos, estables y consistentes entre la evaluación interna y externa.
- Con las 20 características, los resultados presentan un comportamiento más variable. Por ejemplo, en los *folds 5* se evidencia un comportamiento variable (Accuracy interno = 0.8769, externo = 0.9167), mientras que en el *fold 2* se presenta un leve sobreajuste (F1 interno = 0.9177, externo = 0.8832). Por lo anterior, este grupo de características resultó ser el más inestable, además de mostrar un rendimiento promedio ligeramente inferior al de 15 características.

Teniendo en cuenta lo anterior, la configuración con 15 características, *kernel* radial y valor de penalización $C = 10$ muestra ser la más estable y con mejor rendimiento general. Aun así, en la Figura 5-1 se presenta un gráfico comparativo que resume estos resultados considerando únicamente el número de características.

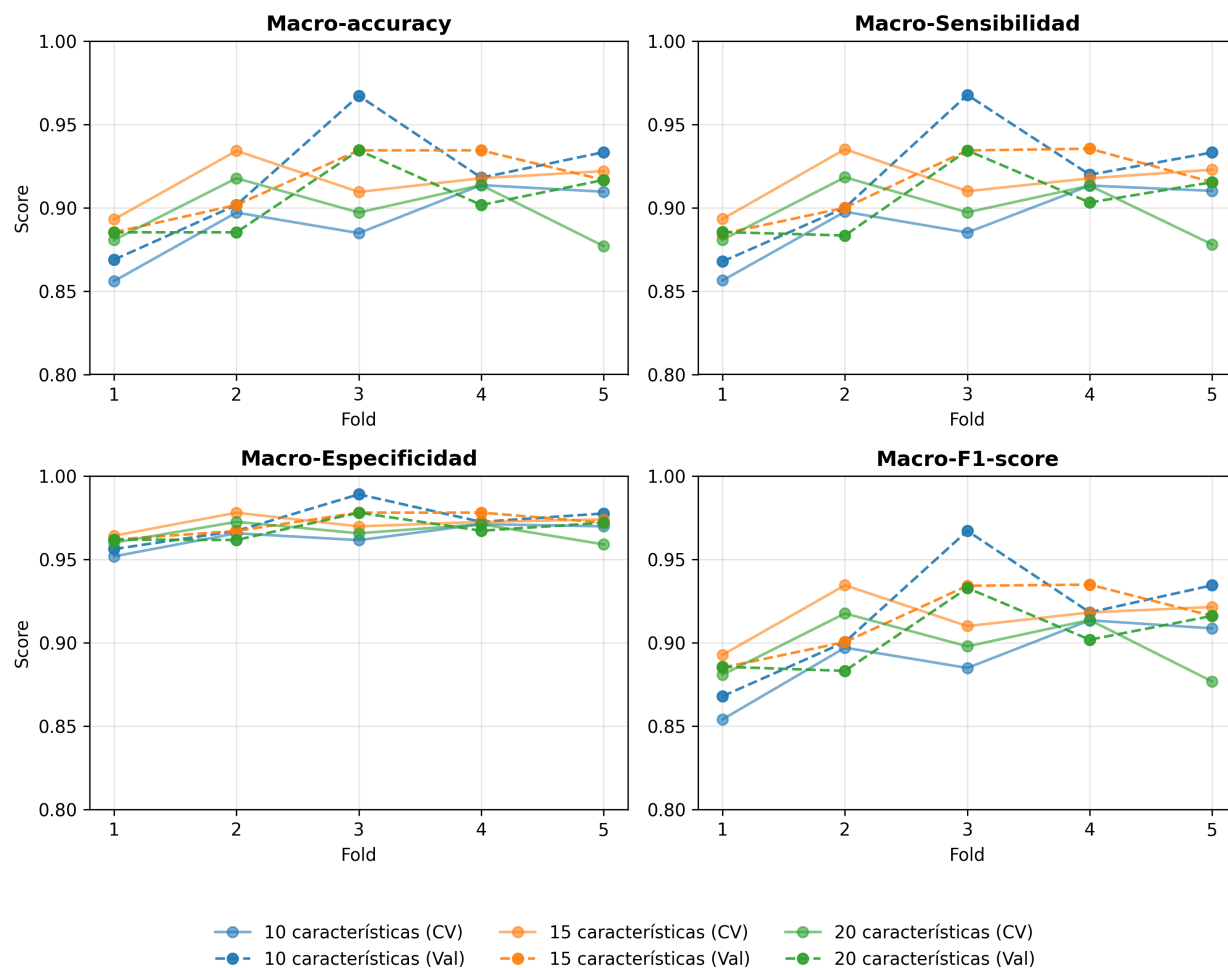


Figura 5-1.: Curvas comparativas por número de características según los macro-promedio de las métricas.

Esta figura 5-1 reafirma lo descrito anteriormente, mostrando que la métrica macro de especificidad es la más estable entre los grupos de características y *folds*. Y que de manera general, el grupo de 10 características es estable, pero presenta los menores rendimientos entre los mismos *folds*; el de 15 características es estable y alcanza los mejores promedios globales (Accuracy 0.93 y F1-score 0.93); y el de 20 características resulta ser el más inestable, con una ligera disminución en el desempeño promedio.

Selección del subconjunto de características

Como resultado del proceso de selección y validación, el entrenamiento final del modelo SVM se realizó utilizando un **kernel radial**, con un parámetro de penalización $C = 10$ y un subconjunto final de **15 características**. La conformación de este subconjunto se determinó con base en la consistencia de aparición de las variables a lo largo de los diferentes *folds*. La Tabla 5-2 presenta las características seleccionadas por cada partición mediante el procedimiento *RFE* (Eliminación Recursiva de Características), evidenciando aquellas variables que el método identificó como más relevantes para la clasificación con el modelo SVM. Dichas características incluyen información proveniente tanto del lado A como del lado B de las imágenes, lo que refleja una selección equilibrada entre ambas perspectivas. Además, se observa una diversidad en los tipos de descriptores, al incluir tanto características derivadas de la transformada *wavelet* (DWT) como *bins* del histograma LBP.

Tabla 5-2.: Subconjunto de 15 Características seleccionadas para cada Fold

N°	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
1	A_LL_energia	A_LL_varianza	A_LL_energia	A_LL_energia	A_LL_varianza
2	A_LL_entropia	A_LL_energia	A_LL_entropia	B_LL_energia	A_LL_energia
3	B_LL_varianza	A_LL_entropia	B_LL_varianza	B_LL_entropia	A_LL_entropia
4	B_LL_energia	B_LL_varianza	B_LL_entropia	A_LH_media	B_LL_media
5	B_LL_entropia	B_LL_entropia	A_LH_media	A_LH_entropia	B_LL_entropia
6	B_LH_media	A_LH_media	A_LH_entropia	A_HH_entropia	A_LH_media
7	B_HL_entropia	B_LH_media	A_HL_energia	LBP_A_bin_4	A_HL_media
8	A_HH_entropia	B_HL_media	B_HL_media	LBP_A_bin_28	A_HL_energia
9	LBP_B_bin_0	A_HH_entropia	A_HH_entropia	LBP_A_bin_112	A_HH_entropia
10	LBP_A_bin_28	LBP_A_bin_4	LBP_A_bin_64	LBP_A_bin_159	LBP_A_bin_13
11	LBP_A_bin_112	LBP_A_bin_28	LBP_A_bin_112	LBP_A_bin_207	LBP_A_bin_28
12	LBP_B_bin_240	LBP_A_bin_112	LBP_A_bin_125	LBP_A_bin_224	LBP_A_bin_64
13	LBP_A_bin_241	LBP_A_bin_159	LBP_A_bin_211	LBP_A_bin_241	LBP_A_bin_112
14	LBP_B_bin_248	LBP_A_bin_193	LBP_A_bin_241	LBP_A_bin_249	LBP_B_bin_207
15	LBP_B_bin_252	LBP_A_bin_241	LBP_B_bin_252	LBP_A_bin_255	LBP_A_bin_241

La selección final se determinó considerando la frecuencia de aparición de las variables a lo largo de los *folds*, utilizando además un ranking promedio como criterio de desempate. La Tabla 5-3 presenta las características elegidas de manera definitiva junto con sus respectivos resultados. Se observa que la mayoría de las variables provienen del **lado A** de la imagen y están principalmente asociadas a información derivada de las subbandas de la transformada DWT. Esto no significa que

el lado A tenga mejores patrones de textura que el lado B. La mayor presencia de características del lado A se debe a que, durante el proceso de captura de la imagen, la espuma se analiza de frente, lo que hace que este lado tenga una superficie visible más grande. Esta mayor área permite capturar más información de textura, la cual es aprovechada por la DWT y contribuye al proceso de clasificación.

Tabla 5-3.: Características finales seleccionadas según su frecuencia y ranking promedio en los *folds*.

Nº	Características	Descripción	Frecuencia	Ranking (Promedio)
1	A_HH_entropia	Entropía Subbanda HH (Imagen A)	5	1.0
2	A_LL_energia	Energía Subbanda LL (Imagen A)	5	1.0
3	B_LL_entropia	Entropía Subbanda LL (Imagen B)	5	1.0
4	LBP_A_bin_112	Histograma LBP bin 112 (Imagen A)	5	1.0
5	LBP_A_bin_241	Histograma LBP bin 241 (Imagen A)	5	1.0
6	LBP_A_bin_28	Histograma LBP bin 28 (Imagen A)	4	2.4
7	A_LH_media	Media Subbanda LH (Imagen A)	4	4.2
8	A_LL_entropia	Entropía Subbanda LL (Imagen A)	4	4.4
9	B_LL_varianza	Varianza Subbanda LL (Imagen B)	3	10.8
10	A_LH_entropia	Entropía Subbanda LH (Imagen A)	2	4.0
11	A_LL_varianza	Varianza Subbanda LL (Imagen A)	2	4.2
12	B_HL_media	Media Subbanda HL (Imagen B)	2	6.4
13	A_HL_energia	Energía Subbanda HL (Imagen A)	2	7.8
14	B_LL_energia	Energía Subbanda LL (Imagen B)	2	8.0
15	LBP_A_bin_4	Histograma LBP bin 4 (Imagen A)	2	8.8

5.1.2 Selección de características e hiperparámetro C - RLM

En la Tabla 5-4 se presentan los resultados detallados de la validación cruzada por *fold*, considerando tres configuraciones del número de características (10, 15 y 20). Para cada caso se reportan los valores del hiperparámetro C , accuracy (exactitud), sensibilidad, especificidad y F1-score. Los resultados correspondientes a todas las combinaciones posibles entre el número de características seleccionadas (10, 15 y 20), los valores del hiperparámetro C se presentan en el Anexo C.

En cuanto al rendimiento, en la tabla se puede observar que, para 10 características, el modelo exhibió una marcada inestabilidad al presentar distintos grados de sobreajuste en los cinco *folds*. Los *folds* 1, 2 y 4 (todos con $C = 10$) mostraron un sobreajuste leve, ya que sus métricas de validación externa fueron significativamente más bajas que las internas. En el *fold* 1, la exactitud (accuracy)

cayó de 0,8765 a 0,7377, el F1-score de 0,8758 a 0,7380, y la diferencia también se mantuvo en la Sensibilidad (0,8765 $\rightarrow$ 0,7385) y la Especificidad (0,9588 $\rightarrow$ 0,9126). En contraste, el *fold* 5 (con $C = 1$) presentó un sobreajuste moderado, donde las métricas de validación interna fueron mucho más altas que las externas (Accuracy 0,8607 $\rightarrow$ 0,6833 y F1-score 0,8598 $\rightarrow$ 0,6824). Este comportamiento indica que el modelo, con 10 características, no logra una generalización consistente.

Tabla 5-4.: Resultados de la selección de características e hiperparámetro evaluados según los macro-promedios de las métricas en validaciones internas (Int.) y externas (Ext.).

Macro-Promedio			Accuracy		Sensibilidad		Especificidad		F1-score	
Nº	Fold	C	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.
10	1	10	0.8765	0.7377	0.8765	0.7385	0.9588	0.9126	0.8758	0.7380
10	2	10	0.8642	0.7869	0.8646	0.7833	0.9547	0.9292	0.8646	0.7830
10	3	10	0.8807	0.8852	0.8805	0.8833	0.9602	0.9617	0.8803	0.8815
10	4	10	0.8724	0.8033	0.8723	0.8063	0.9575	0.9342	0.8718	0.8001
10	5	1	0.8607	0.6833	0.8610	0.6835	0.9535	0.8945	0.8598	0.6824
15	1	10	0.9259	0.7869	0.9260	0.7875	0.9753	0.9291	0.9256	0.7871
15	2	10	0.9465	0.8361	0.9465	0.8333	0.9822	0.9455	0.9465	0.8348
15	3	10	0.9095	0.9180	0.9099	0.9177	0.9699	0.9728	0.9092	0.9182
15	4	10	0.9177	0.9344	0.9177	0.9354	0.9726	0.9781	0.9182	0.9349
15	5	10	0.9220	0.9167	0.9228	0.9153	0.9741	0.9721	0.9215	0.9162
20	1	1	0.9342	0.7705	0.9342	0.7698	0.9781	0.9238	0.9340	0.7699
20	2	10	0.9753	0.8689	0.9753	0.8667	0.9918	0.9564	0.9753	0.8654
20	3	10	0.9465	0.9016	0.9465	0.9021	0.9822	0.9670	0.9463	0.9029
20	4	10	0.9794	0.8361	0.9794	0.8385	0.9931	0.9450	0.9794	0.8358
20	5	1	0.9180	0.7667	0.9182	0.7649	0.9727	0.9221	0.9181	0.7624

Para la configuración con 15 características, se observó una estabilidad considerablemente mayor en el modelo en comparación con otras configuraciones. Solo los *folds* 1 y 2, ambos con $C = 10$, presentaron un leve sobreajuste. Este sobreajuste se manifestó en caídas notables en las métricas externas respecto a las internas: en el *fold* 1, Accuracy (0,9259 $\rightarrow$ 0,7869), Sensibilidad (0,9260 $\rightarrow$ 0,7875), Especificidad (0,9753 $\rightarrow$ 0,9291) y F1-score (0,9256 $\rightarrow$ 0,7871). Sin embargo, los *folds* 3, 4 y 5 mostraron una excelente capacidad de generalización, donde las cuatro métricas alcanzaron valores muy similares entre la validación interna y la externa, lo que indica un

modelo bien ajustado. La alta estabilidad y el rendimiento consistente de estos tres *folds* sugieren que 15 características ofrecen el mejor equilibrio entre la complejidad del modelo y la capacidad de generalización.

Al utilizar 20 características, el modelo exhibió la mayor inestabilidad y los problemas de ajuste más severos en las cuatro métricas. El *fold* 1 (con $C = 10$) presentó un sobreajuste fuerte, evidenciado por caídas pronunciadas en todas las métricas de validación externa: Accuracy ($0,9342 \rightarrow 0,7705$), Sensibilidad ($0,9342 \rightarrow 0,7698$), Especificidad ($0,9781 \rightarrow 0,9238$) y F1-score ($0,9340 \rightarrow 0,7699$). Esta diferencia marcada entre desempeño interno y externo indica una clara pérdida de capacidad de generalización.

De forma similar, los *folds* 2, 3 y 4 también mostraron sobreajuste leve, ya que sus métricas internas fueron sistemáticamente superiores a las externas, aunque el rendimiento en validación externa se mantuvo en un rango aceptable (con Accuracy entre 0,83 y 0,90). Por otro lado, el *fold* 5 (con $C = 1$) presentó un sobreajuste moderado, pues las métricas internas fueron considerablemente más altas que las externas en todos los indicadores (por ejemplo, F1-score $0,9181 \rightarrow 0,7624$), lo que confirma que el modelo se ajustó en exceso a los datos de entrenamiento en detrimento de su capacidad de generalización.

A partir de este análisis, la configuración con 15 características y $C = 10$ fue seleccionada como la más adecuada, ya que logró el mejor compromiso entre desempeño en validación y control del sobreajuste. Este hallazgo sugiere que un número intermedio de variables aporta la cantidad de información suficiente para la clasificación, evitando tanto la pérdida de señal (subajuste con pocas características) como la redundancia y ruido excesivo (sobreajuste con muchas características).

Selección de las 15 mejores características

A partir del análisis realizado en los distintos *folds* de validación cruzada, se construyó un ranking de importancia de las variables. Para seleccionar las 15 mejores características, se tuvo en cuenta no solo el peso de los coeficientes del modelo, sino también su frecuencia de aparición y la estabilidad de su relevancia a lo largo de los *folds*, siguiendo los criterios metodológicos descritos en la Sección anterior.

La Tabla 5-5 presenta el conjunto final de 15 características seleccionadas mediante este procedi-

miento. Dichas variables incluyen descriptores derivados de la transformada wavelet y de histogramas LBP, lo que evidencia que la combinación de información de textura global y patrones locales resulta adecuada para la tarea de clasificación.

Tabla 5-5.: Características finales seleccionadas según su frecuencia y ranking promedio en los *folds* (modelo de regresión logística multinomial).

N°	Característica	Descripción	Frecuencia	Ranking (Promedio)
1	A_LL_energia	Energía subbanda LL (imagen A)	5	4.5681
2	B_LL_entropia	Entropía subbanda LL (imagen B)	5	4.5144
3	A_LH_media	Media subbanda LH (imagen A)	5	4.1610
4	A_LL_entropia	Entropía subbanda LL (imagen A)	5	3.9088
5	A_LL_varianza	Varianza subbanda LL (imagen A)	5	3.7651
6	LBP_A_bin_112	Histograma LBP bin 112 (imagen A)	4	5.5421
7	LBP_A_bin_28	Histograma LBP bin 28 (imagen A)	4	4.1437
8	LBP_A_bin_193	Histograma LBP bin 193 (imagen A)	4	3.3523
9	B_LL_energia	Energía subbanda LL (imagen B)	4	2.9898
10	LBP_A_bin_125	Histograma LBP bin 125 (imagen A)	3	3.6441
11	LBP_B_bin_207	Histograma LBP bin 207 (imagen B)	3	3.1921
12	LBP_B_bin_249	Histograma LBP bin 249 (imagen B)	3	2.9891
13	B_HL_media	Media subbanda HL (imagen B)	3	2.4777
14	A_HH_entropia	Entropía subbanda HH (imagen A)	2	4.9587
15	A_HL_energia	Energía subbanda HL (imagen A)	2	4.7749

5.2 Resultado del entrenamiento final

Se comparan los resultados de los modelos finales, los cuales fueron entrenados empleando la totalidad del conjunto de entrenamiento (304 pares de imágenes) y posteriormente evaluados sobre el conjunto de prueba (76 pares de imágenes). Cada modelo utilizó los hiperparámetros y condiciones previamente optimizados, con el fin de asegurar el mejor rendimiento en la etapa de evaluación final.

En la clasificación final, el modelo de Máquinas de Soporte Vectorial (SVM) se entrenó con un kernel radial, un valor de penalización $C = 10$ y el subconjunto de 15 características definidas en la Tabla 5-3. Por su parte, el modelo de Regresión Logística Multinomial se entrenó con una

regularización L2, un valor de penalización $C = 10$ y las 15 características seleccionadas que se presentan en la Tabla 5-5.

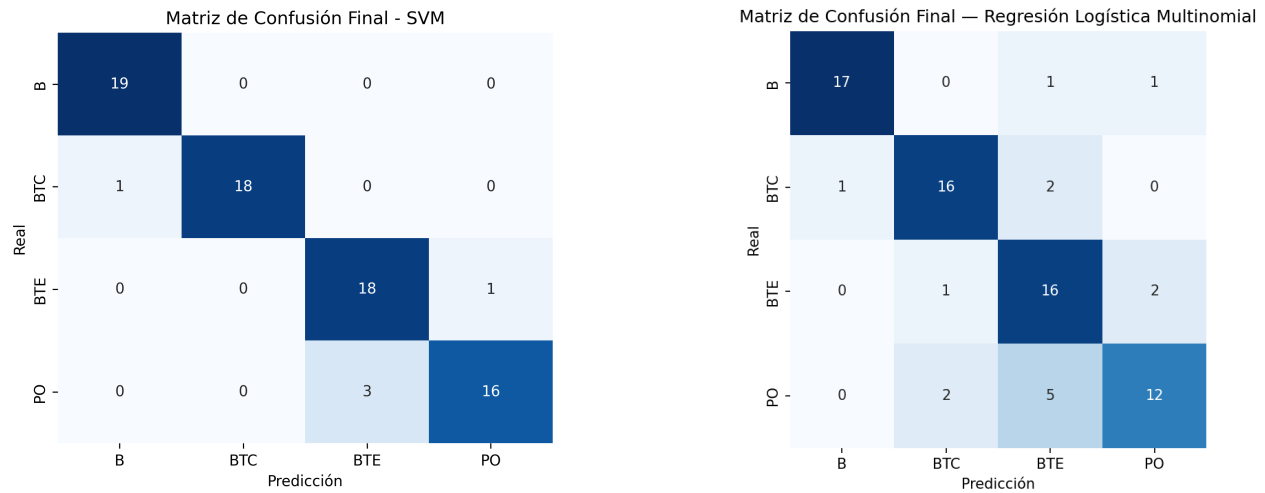


Figura 5-2.: Comparación de matrices de confusión de los modelos finales

En la Figura 5-2 se aprecia que, de acuerdo con las matrices de confusión, el modelo de máquina de soporte vectorial (SVM) presentó un menor número de errores de clasificación en el conjunto de prueba. En la matriz del SVM se identifican 5 observaciones mal clasificadas en total, distribuidas así: 1 error en la clase **BTC**, 1 error en la clase **BTE** y 3 errores en la clase **PO**. La clase **B** fue clasificada correctamente en todos los casos

En contraste, el modelo de regresión logística multinomial (RLM) registró 15 errores en total, distribuidos de la siguiente forma: clase **B**: 2 errores (una observación predicha como **BTE** y una como **PO**); clase **BTC**: 3 errores (una observación predicha como **B** y dos como **BTE**); clase **BTE**: 3 errores (una observación predicha como **BTC** y dos como **PO**); y clase **PO**: 7 errores (dos observaciones predichas como **BTC** y cinco como **BTE**). Es decir, mientras el SVM concentra pocos errores y en menos clases, el RLM comete más equivocaciones y las reparte en todas las categorías, especialmente en la clase **PO**.

Tabla 5-6.: Métricas finales de los modelos SVM y Regresión Logística Multinomial. Las celdas en verde indican las métricas con mejor desempeño, en rojo las de menor rendimiento y en gris aquellas con valores muy similares.

Modelo	Clase	Accuracy	Sensibilidad	Especificidad	F1-Score
Maquina de Soporte Vectorial	B	0.9868	1.0000	0.9824	0.9743
	BTC	0.9868	0.9473	1.0000	0.9729
	BTE	0.9473	1.0000	0.9473	0.9000
	PO	0.9473	0.8421	0.9824	0.8888
Macro-promedio		0.9671	0.9342	0.9780	0.9340
Logístico Multinomial	B	0.9605	0.8947	0.9825	0.9189
	BTC	0.9211	0.8421	0.9474	0.8421
	BTE	0.8553	0.8421	0.8596	0.7442
	PO	0.8684	0.6316	0.9474	0.7059
Macro-promedio		0.9263	0.8026	0.9342	0.8028

Asimismo, en la Tabla 5-6, que resume las métricas macro e individuales por clase, se observó lo siguiente:

- **Exactitud (accuracy):** El SVM alcanza una exactitud macro de 0.9671, mientras que el RLM llega a 0.9263. Esto significa que, considerando todas las clases con el mismo peso, el SVM comete menos errores globales. Esta diferencia es coherente con lo que se observó en las matrices de confusión, en donde el SVM solo presentó 5 clasificaciones incorrectas, mientras que el RLM registró 15.
- **Sensibilidad (recall):** El SVM identifica una mayor proporción de las observaciones que realmente pertenecen a cada clase, en contraste, RLM al presentar más desviaciones hacia clases cercanas (PO predicha como BTE o BTE predicha como PO), reduce su capacidad de detección. ($SVM = 0,9342 \rightarrow RLM = 0,8026$)
- **Especificidad:** El SVM muestra una mayor capacidad para no confundir instancias de otras clases con la clase analizada, alcanzando una especificidad de 0.9780, superior a la obtenida por la RLM (0.9342). Esto se refleja en que el SVM solo cometió 5 clasificaciones erróneas hacia la clase objetivo, mientras que la RLM incurrió en 15. En otras palabras, el SVM no

solo acierta más, sino que también se equivoca menos al clasificar como pertenecientes a la clase analizada ejemplos que en realidad provienen de otras categorías.

- **F1-Score:** Dado que esta métrica representa la media armónica entre la Precisión y la Sensibilidad, el valor superior de SVM:0,9340 frente al 0,8028 de la RLM indica un desempeño más consistente y equilibrado en la clasificación. La RLM, al registrar una mayor incidencia de Falsos Positivos y Falsos Negativos, experimenta una penalización significativa que afecta simultáneamente su Precisión y Sensibilidad, lo que resulta en un F1-Score global notablemente inferior.

Al analizar el desempeño por clase se puede observar que:

- **Clase B:** Ambos modelos presentan un rendimiento alto, pero el SVM conserva la mejor respuesta. El SVM obtuvo una exactitud de 0.9868 y un F1-Score de 0.9743, mientras que el RLM alcanzó 0.9605 de exactitud y 0.9189 de F1-Score. La diferencia se explica porque el RLM cometió 2 errores en esta clase (una muestra la asignó a BTE y otra a PO), mientras que el SVM no presentó errores en B.
- **Clase BTC:** El SVM logró una exactitud de 0.9868, una sensibilidad de 0.9473, una especificidad de 1.0000 y un F1-Score de 0.9729. El RLM, en cambio, presentó 3 errores en esta clase (1 instancia se clasificó como B y 2 se clasificaron como BTE), lo que ocasionó una disminución de sus métricas hasta 0.9211 de exactitud y 0.8421 de F1-Score. Esto evidencia que el SVM distingue con mayor claridad esta clase frente al resto.
- **Clase BTE:** El SVM volvió a obtener los mejores valores, con 0.9473 de exactitud, 1.0000 de sensibilidad y 0.9000 de F1-Score. El RLM presentó también 3 errores en esta clase (1 instancia fue predicha como BTC y 2 como PO), lo que redujo su exactitud a 0.8553 y su F1-Score a 0.7442. Se observa que el RLM tiende a confundir esta categoría con otras clases de defecto.
- **Clase PO:** Es la clase en la que la diferencia entre los modelos es más marcada. El SVM cometió 3 errores en PO y aun así mantuvo 0.9473 de exactitud, 0.8421 de sensibilidad y 0.8888 de F1-Score. El RLM, por su parte, cometió 7 errores en esta clase (2 observaciones se asignaron a BTC y 5 a BTE), por lo que su exactitud descendió a 0.8684, su sensibilidad a 0.6316 y su F1-Score a 0.7059. Aunque el RLM muestra una especificidad ligeramente mayor en esta clase, dicha ventaja no compensa la pérdida de sensibilidad ni el mayor número de confusiones.

En síntesis, los resultados indican que el modelo **SVM** ofrece un mejor rendimiento general, especialmente en las clases que representan defectos (BTC, BTE y PO). Esto sugiere que dicho modelo logra capturar de manera más efectiva las variaciones texturales presentes en las imágenes (particularmente aquellas relacionadas con bordes y cambios de intensidad detectados mediante las transformadas wavelet y LBP), lo que lo convierte en la opción más adecuada para la clasificación de productos no conformes.

6. Conclusiones

Se concluye que el objetivo de desarrollar un modelo de clasificación supervisada para la detección de defectos en las espumas florales fue plenamente alcanzado. Este trabajo establece una primera aproximación técnica validada que demuestra de manera concluyente la viabilidad de automatizar el control de calidad mediante el análisis de imágenes. El modelo de Máquina de Soporte Vectorial (SVM), configurado con un kernel radial, se determinó como la alternativa más adecuada y sólida, superando significativamente el desempeño de la Regresión Logística Multinomial (RLM).

El diseño experimental para la adquisición de la base de datos fue un factor determinante en la confiabilidad de los resultados. La estandarización rigurosa de las condiciones de captura (iluminación, fondo y disposición de las muestras) aseguró que los modelos de clasificación pudieran centrarse exclusivamente en las variaciones texturales intrínsecas de los defectos, minimizando el ruido externo y validando así el proceso de adquisición de datos. Este rigor metodológico fue complementado por la eficacia de la extracción de características, donde la combinación de las transformadas Wavelet (para texturas globales) y los descriptores LBP (para detalles locales) resultó ser el conjunto de variables óptimo y necesario para representar y diferenciar las cuatro categorías.

La superioridad del SVM no es marginal, sino que se manifiesta de forma consistente en todas las métricas globales de desempeño, lo que confirma su idoneidad para la tarea. El modelo demostró una clara ventaja en Exactitud (0,9671 vs. 0,9263), Sensibilidad (0,9342 vs. 0,8026), Especificidad (0,9780 vs. 0,9342) y F1-Score (0,9340 vs. 0,8028). Esta consistencia en las cuatro métricas es crítica, pues garantiza que el SVM no solo acierta globalmente, sino que mantiene un equilibrio óptimo entre la capacidad de detección de fallas y la minimización de clasificaciones erróneas.

Desde la perspectiva de la aplicación industrial, el valor agregado del modelo SVM radica en su capacidad dual de aprobación y detección. La clase B (Espuma Buena) fue clasificada correctamente en la totalidad de los casos, un hecho que minimiza el riesgo económico de Falsos Positivos

al certificar la calidad de los productos aprobados. Por otro lado, la alta precisión en la identificación de los defectos más complejos (BTC, BTE y PO) asegura que el sistema es eficaz para filtrar el producto no conforme, reduciendo drásticamente la subjetividad y la variabilidad inherente a la inspección visual humana.

Finalmente, este estudio muestra que el análisis de textura constituye un enfoque adecuado para abordar la complejidad de los defectos superficiales en este tipo de material poroso. Los resultados obtenidos indican que, entre los modelos evaluados en este trabajo (SVM y regresión logística multinomial), la solución desarrollada ofrece el mejor desempeño bajo las condiciones experimentales establecidas. No obstante, es importante resaltar que estos resultados se obtuvieron en un entorno controlado; por tanto, sería pertinente realizar pruebas adicionales en condiciones más cercanas al contexto operativo real para confirmar la robustez y estabilidad del sistema antes de su implementación final en la empresa.

7. Recomendaciones

El presente trabajo constituye un primer acercamiento al desarrollo de un sistema automatizado de inspección visual para espumas florales. A partir de los resultados obtenidos y de las limitaciones identificadas, se plantean las siguientes recomendaciones para futuras investigaciones:

- **Diseñar un nuevo experimento bajo condiciones reales de captura:** Las imágenes utilizadas se adquirieron en un entorno controlado de laboratorio; sin embargo, el área de inspección disponible en la empresa presenta condiciones de iluminación y ambientales distintas, que afectan la calidad de las imágenes y, en consecuencia, la precisión del modelo (Coşkun et al., 2022; Gonzalez & Woods, 2018). Por ello, se recomienda realizar un nuevo diseño experimental bajo condiciones reales de campo, que contemple diferentes niveles de iluminación y variabilidad ambiental.
- **Explorar nuevas técnicas de extracción de características:** Si bien la transformada discreta de wavelet (DWT-2D) combinada con los patrones locales binarios (LBP) permitió obtener resultados consistentes, se sugiere evaluar otros descriptores, ya sea como técnicas principales o complementarias. Entre ellos, destacan la detección de bordes mediante el método de Canny, las características geométricas, los momentos invariantes y las matrices de coocurrencia (GLCM) (Coşkun et al., 2022; Nazir et al., 2021). Estas alternativas podrían capturar información adicional sobre textura y forma, mejorando la discriminación entre defectos visualmente similares. No obstante, su implementación debe equilibrarse con el tamaño del conjunto de datos, pues una mayor complejidad de características sin un incremento proporcional de muestras podría inducir sobreajuste y aumentar el costo computacional.
- **Explorar métodos alternativos de reducción de dimensionalidad:** Aunque el método RFE permitió seleccionar un subconjunto representativo de características, se recomienda explorar enfoques complementarios o alternativos, como el Análisis de Componentes Principales (PCA), el Análisis Discriminante Lineal (LDA) o los Autoencoders (Kiraz et al., 2023; Nazir et al., 2021; Yousif et al., 2022). Estas técnicas pueden contribuir a reducir la redundancia

entre variables altamente correlacionadas y mejorar la interpretabilidad de los espacios de características, conservando la información relevante para la clasificación. Su comparación frente al RFE aportaría evidencia sobre la influencia del método de reducción en el desempeño final del modelo.

- **Evaluar arquitecturas de aprendizaje profundo:** Aunque el modelo SVM mostró un desempeño sobresaliente en las métricas de clasificación, su flujo de trabajo requiere un proceso previo de extracción de características que incrementa la complejidad y el tiempo de procesamiento. En este sentido, se recomienda considerar la implementación de redes neuronales convolucionales (CNN) u otros modelos de aprendizaje profundo que utilicen directamente las imágenes RGB como entrada, reduciendo las etapas intermedias y simplificando la aplicación práctica del sistema. No obstante, para obtener un desempeño competitivo con estos modelos sería necesario ampliar y diversificar la base de datos, así como disponer de recursos computacionales adecuados (Münke et al., [2024](#)).

A. Detalles del proceso de captura de imágenes

Equipo Utilizado

- **Cámara:** Canon EOS M50 Mark II, seleccionada por su capacidad para capturar imágenes de alta calidad y detalles precisos.
- **Lente:** Canon EF-M 15-45mm f/3.5-6.3 IS STM, adecuado para fotografiar superficies texturizadas.
- **Accesorios:** Trípode, empleado para garantizar estabilidad y evitar vibraciones que afecten la nitidez de las imágenes.



Figura A-1.: Equipos utilizados en la captura de imagen.

Configuración de la Cámara Los parámetros ajustados en la cámara fueron los siguientes:

- **Apertura:** F16, configurada para obtener una profundidad de campo amplia que mantuviera toda la superficie de las espumas enfocada.
- **Velocidad de obturación:** 0.6 segundos, ajustada para asegurar una exposición adecuada en condiciones de iluminación controlada.

- **ISO:** Automático, equilibrando la luz disponible y minimizando el ruido en las imágenes.
- **Balance de blancos:** Automático, adaptado a la iluminación blanca utilizada.
- **Medición:** Evaluativa, para analizar de manera uniforme la exposición de toda la escena.
- **Compensación de exposición:** 0, buscando un resultado neutral.
- **Estilo de imagen:** Neutro, con los siguientes ajustes:
 - Nitidez: 2
 - Contraste: 0
 - Saturación: 0
 - Tono de color: 0

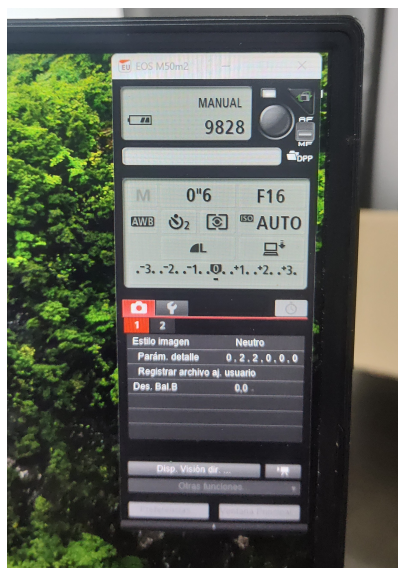


Figura A-2.: Configuración de la cámara

Iluminación Para garantizar una iluminación adecuada y resaltar las texturas y detalles de las espumas, se utilizó un bombillo LED de 7W y 600 lúmenes. Sus características principales son:

- **Temperatura de color:** Luz blanca fría (aproximadamente 6500 K), ideal para resaltar detalles en superficies texturizadas.

- **Consumo energético:** 7W, eficiente y sostenible.
- **Flujo luminoso:** 600 lúmenes, proporcionando una iluminación uniforme y suficiente para las capturas.
- **Tipo de luz:** Direccional y suave, lo que evitó sombras pronunciadas y resaltó las irregularidades en las espumas.

La iluminación fue proyectada desde diferentes ángulos para resaltar las texturas y las imperfecciones presentes en la superficie del material. Este enfoque permitió capturar detalles precisos y minimizar reflejos no deseados.

Composición y Posicionamiento

- **Ángulo de la cámara:** Aproximadamente 45 grados hacia abajo, proporcionando una vista clara y comprensiva de la superficie del material.
- **Fondo:** Blanco, para eliminar distracciones visuales y centrar la atención en las características del objeto fotografiado.
- **Orientación de la espuma:** Las espumas se colocaron de manera estratégica para enfatizar áreas con defectos e irregularidades, facilitando su análisis.

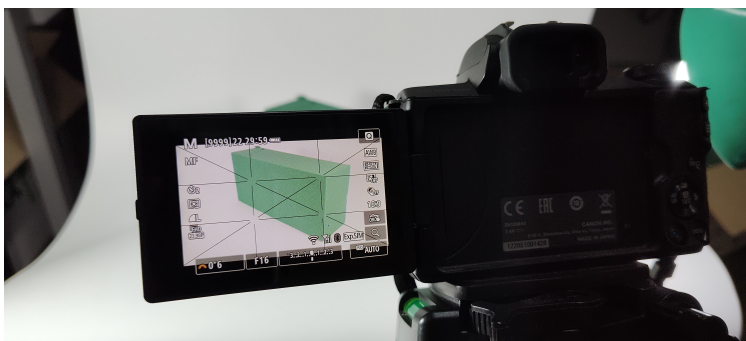


Figura A-3.: Iluminación y posicionamiento

B. Resultados detallados de las combinaciones evaluadas en los folds para SVM

Tabla B-1.: Resultados completos (macro-promedios) para todas las combinaciones de número de características e hiperparámetros en SVM. Validaciones internas (Int.) y externas (Ext.).

Macro-Promedio				Accuracy		Sensibilidad		Especificidad		F1-score	
Nº	Fold	C	Kernel	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.
10	1	0.1	Lineal	0.6666	0.7377	0.6680	0.7354	0.8889	0.9121	0.6493	0.7304
10	1	0.1	Radial	0.7325	0.8033	0.7341	0.8042	0.9109	0.9347	0.7217	0.8051
10	1	0.1	Poly	0.7489	0.7869	0.7500	0.7865	0.9162	0.9289	0.7405	0.7781
10	1	1	Lineal	0.7407	0.7213	0.7415	0.7198	0.9136	0.9070	0.7320	0.7207
10	1	1	Radial	0.8272	0.8361	0.8280	0.8354	0.9423	0.9454	0.8250	0.8345
10	1	1	Poly	0.8230	0.8852	0.8236	0.8854	0.9410	0.9618	0.8185	0.8837
10	1	10	Lineal	0.7366	0.7541	0.7369	0.7542	0.9122	0.9181	0.7224	0.7551
10	1	10	Radial	0.8560	0.8689	0.8563	0.8677	0.9519	0.9563	0.8541	0.8680
10	1	10	Poly	0.8477	0.8525	0.8480	0.8521	0.9492	0.9510	0.8437	0.8525
10	1	100	Lineal	0.6914	0.7049	0.6917	0.7052	0.8971	0.9018	0.6871	0.7062
10	1	100	Radial	0.8765	0.8689	0.8762	0.8688	0.9587	0.9563	0.8765	0.8687
10	1	100	Poly	0.8148	0.8525	0.8147	0.8510	0.9382	0.9508	0.8118	0.8496
10	2	0.1	Lineal	0.7078	0.7705	0.7095	0.7677	0.9027	0.9233	0.6939	0.7665
10	2	0.1	Radial	0.7572	0.7869	0.7585	0.7833	0.9192	0.9286	0.7506	0.7805
10	2	0.1	Poly	0.7572	0.7541	0.7575	0.7510	0.9191	0.9180	0.7455	0.7450
10	2	1	Lineal	0.7572	0.7869	0.7577	0.7833	0.9192	0.9287	0.7541	0.7829
10	2	1	Radial	0.8354	0.8525	0.8361	0.8500	0.9452	0.9506	0.8352	0.8496

Macro-Promedio				Accuracy		Sensibilidad		Especificidad		F1-score	
N°	Fold	C	Kernel	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.
10	2	1	Poly	0.8601	0.8361	0.8605	0.8333	0.9534	0.9452	0.8608	0.8333
10	2	10	Lineal	0.7695	0.8197	0.7700	0.8167	0.9232	0.9397	0.7637	0.8158
10	2	10	Radial	0.8971	0.9016	0.8976	0.9000	0.9657	0.9671	0.8971	0.9005
10	2	10	Poly	0.8765	0.8361	0.8770	0.8333	0.9589	0.9454	0.8763	0.8339
10	2	100	Lineal	0.7654	0.8033	0.7669	0.8000	0.9219	0.9343	0.7611	0.7989
10	2	100	Radial	0.9136	0.9180	0.9141	0.9167	0.9712	0.9726	0.9134	0.9169
10	2	100	Poly	0.8930	0.8689	0.8940	0.8667	0.9643	0.9563	0.8933	0.8672
10	3	0.1	Lineal	0.6543	0.7541	0.6548	0.7552	0.8846	0.9181	0.6313	0.7542
10	3	0.1	Radial	0.7366	0.8197	0.7363	0.8198	0.9120	0.9401	0.7222	0.8204
10	3	0.1	Poly	0.7737	0.8525	0.7722	0.8552	0.9243	0.9511	0.7647	0.8507
10	3	1	Lineal	0.7325	0.7869	0.7319	0.7854	0.9109	0.9287	0.7168	0.7818
10	3	1	Radial	0.8354	0.9016	0.8353	0.9021	0.9451	0.9671	0.8357	0.9030
10	3	1	Poly	0.8148	0.8689	0.8143	0.8687	0.9383	0.9563	0.8110	0.8682
10	3	10	Lineal	0.7366	0.8689	0.7373	0.8677	0.9124	0.9563	0.7322	0.8662
10	3	10	Radial	0.8848	0.9672	0.8851	0.9677	0.9616	0.9891	0.8849	0.9672
10	3	10	Poly	0.8560	0.9180	0.8560	0.9188	0.9521	0.9726	0.8553	0.9193
10	3	100	Lineal	0.7695	0.8689	0.7694	0.8698	0.9234	0.9563	0.7678	0.8682
10	3	100	Radial	0.8765	0.9016	0.8758	0.9021	0.9588	0.9674	0.8752	0.9020
10	3	100	Poly	0.8519	0.9016	0.8520	0.9010	0.9507	0.9673	0.8454	0.9003
10	4	0.1	Lineal	0.6667	0.7049	0.6669	0.7042	0.8888	0.9012	0.6464	0.6650
10	4	0.1	Radial	0.7778	0.7377	0.7768	0.7375	0.9258	0.9122	0.7671	0.7306
10	4	0.1	Poly	0.8107	0.8689	0.8101	0.8688	0.9367	0.9562	0.8046	0.8691
10	4	1	Lineal	0.7407	0.6721	0.7395	0.6729	0.9135	0.8905	0.7257	0.6639
10	4	1	Radial	0.8601	0.8197	0.8593	0.8219	0.9533	0.9396	0.8569	0.8201
10	4	1	Poly	0.8642	0.8852	0.8631	0.8844	0.9545	0.9614	0.8586	0.8861
10	4	10	Lineal	0.7613	0.7869	0.7599	0.7885	0.9203	0.9286	0.7495	0.7902
10	4	10	Radial	0.9136	0.9180	0.9133	0.9198	0.9712	0.9726	0.9135	0.9183
10	4	10	Poly	0.9012	0.9016	0.9010	0.9021	0.9670	0.9669	0.9016	0.9023
10	4	100	Lineal	0.7366	0.7213	0.7369	0.7219	0.9123	0.9068	0.7273	0.7207
10	4	100	Radial	0.8765	0.8197	0.8760	0.8219	0.9589	0.9399	0.8752	0.8217
10	4	100	Poly	0.8601	0.8033	0.8603	0.8042	0.9533	0.9343	0.8601	0.8058

Macro-Promedio				Accuracy		Sensibilidad		Especificidad		F1-score	
N°	Fold	C	Kernel	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.
10	5	0.1	Lineal	0.7008	0.6167	0.6996	0.6244	0.9002	0.8734	0.6739	0.5945
10	5	0.1	Radial	0.7660	0.7833	0.7659	0.7871	0.9221	0.9284	0.7486	0.7805
10	5	0.1	Poly	0.7702	0.7833	0.7712	0.7781	0.9235	0.9275	0.7608	0.7724
10	5	1	Lineal	0.7458	0.7333	0.7456	0.7305	0.9152	0.9105	0.7367	0.7303
10	5	1	Radial	0.8277	0.8667	0.8288	0.8685	0.9425	0.9555	0.8263	0.8665
10	5	1	Poly	0.8524	0.8333	0.8538	0.8295	0.9510	0.9442	0.8515	0.8284
10	5	10	Lineal	0.7051	0.8000	0.7048	0.8051	0.9017	0.9341	0.6908	0.7978
10	5	10	Radial	0.9096	0.9333	0.9101	0.9332	0.9699	0.9777	0.9087	0.9345
10	5	10	Poly	0.8604	0.8833	0.8611	0.8841	0.9535	0.9611	0.8598	0.8825
10	5	100	Lineal	0.7134	0.8333	0.7137	0.8363	0.9044	0.9452	0.7042	0.8335
10	5	100	Radial	0.8686	0.9167	0.8698	0.9165	0.9562	0.9721	0.8678	0.9180
10	5	100	Poly	0.8278	0.9167	0.8284	0.9177	0.9425	0.9723	0.8278	0.9164
15	1	0.1	Lineal	0.7037	0.7541	0.7048	0.7531	0.9012	0.9178	0.6921	0.7518
15	1	0.1	Radial	0.7654	0.8033	0.7673	0.8042	0.9219	0.9347	0.7568	0.8047
15	1	0.1	Poly	0.8066	0.8197	0.8073	0.8198	0.9355	0.9397	0.8039	0.8182
15	1	1	Lineal	0.8189	0.7705	0.8198	0.7708	0.9396	0.9234	0.8179	0.7714
15	1	1	Radial	0.8395	0.8197	0.8403	0.8219	0.9464	0.9400	0.8395	0.8197
15	1	1	Poly	0.8807	0.8689	0.8808	0.8698	0.9601	0.9563	0.8792	0.8682
15	1	10	Lineal	0.7942	0.7869	0.7944	0.7844	0.9314	0.9291	0.7885	0.7839
15	1	10	Radial	0.8930	0.8852	0.8933	0.8844	0.9643	0.9620	0.8928	0.8852
15	1	10	Poly	0.8807	0.8852	0.8810	0.8844	0.9601	0.9620	0.8788	0.8851
15	1	100	Lineal	0.7695	0.7377	0.7702	0.7365	0.9232	0.9126	0.7633	0.7358
15	1	100	Radial	0.8848	0.8852	0.8849	0.8865	0.9615	0.9620	0.8840	0.8842
15	1	100	Poly	0.8519	0.8361	0.8526	0.8365	0.9506	0.9455	0.8507	0.8372
15	2	0.1	Lineal	0.6872	0.7869	0.6889	0.7833	0.8959	0.9286	0.6739	0.7821
15	2	0.1	Radial	0.7819	0.7869	0.7845	0.7833	0.9275	0.9286	0.7763	0.7805
15	2	0.1	Poly	0.7737	0.8033	0.7750	0.8000	0.9247	0.9345	0.7640	0.7930
15	2	1	Lineal	0.8148	0.8197	0.8161	0.8167	0.9383	0.9399	0.8141	0.8158
15	2	1	Radial	0.8642	0.9016	0.8645	0.9000	0.9548	0.9671	0.8643	0.9005
15	2	1	Poly	0.8807	0.8689	0.8808	0.8667	0.9602	0.9563	0.8803	0.8655
15	2	10	Lineal	0.8272	0.8033	0.8278	0.8000	0.9423	0.9344	0.8278	0.7986

Macro-Promedio				Accuracy		Sensibilidad		Especificidad		F1-score	
N°	Fold	C	Kernel	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.
15	2	10	Radial	0.9342	0.9016	0.9351	0.9000	0.9781	0.9671	0.9346	0.9005
15	2	10	Poly	0.9053	0.9016	0.9063	0.9000	0.9684	0.9671	0.9062	0.9005
15	2	100	Lineal	0.8189	0.8197	0.8196	0.8177	0.9396	0.9401	0.8183	0.8150
15	2	100	Radial	0.9218	0.9016	0.9224	0.9000	0.9740	0.9671	0.9223	0.9005
15	2	100	Poly	0.9095	0.9016	0.9103	0.9000	0.9698	0.9671	0.9100	0.9004
15	3	0.1	Lineal	0.6914	0.7705	0.6909	0.7719	0.8971	0.9236	0.6718	0.7710
15	3	0.1	Radial	0.7407	0.8852	0.7401	0.8865	0.9135	0.9620	0.7314	0.8861
15	3	0.1	Poly	0.7819	0.8525	0.7813	0.8531	0.9271	0.9510	0.7718	0.8532
15	3	1	Lineal	0.8272	0.8525	0.8270	0.8510	0.9425	0.9505	0.8207	0.8506
15	3	1	Radial	0.8519	0.8689	0.8520	0.8677	0.9506	0.9565	0.8515	0.8660
15	3	1	Poly	0.8642	0.9016	0.8643	0.9021	0.9547	0.9673	0.8632	0.9010
15	3	10	Lineal	0.8189	0.9180	0.8188	0.9167	0.9397	0.9726	0.8158	0.9175
15	3	10	Radial	0.9053	0.9344	0.9056	0.9354	0.9685	0.9783	0.9063	0.9342
15	3	10	Poly	0.9095	0.9344	0.9099	0.9344	0.9699	0.9781	0.9100	0.9343
15	3	100	Lineal	0.7942	0.9180	0.7937	0.9177	0.9315	0.9727	0.7911	0.9173
15	3	100	Radial	0.8807	0.9508	0.8804	0.9521	0.9602	0.9837	0.8805	0.9510
15	3	100	Poly	0.8519	0.9016	0.8518	0.9021	0.9506	0.9673	0.8502	0.9011
15	4	0.1	Lineal	0.7037	0.7377	0.7034	0.7375	0.9012	0.9121	0.6876	0.7175
15	4	0.1	Radial	0.8066	0.7377	0.8060	0.7385	0.9355	0.9122	0.7964	0.7205
15	4	0.1	Poly	0.8148	0.8689	0.8141	0.8698	0.9382	0.9563	0.8084	0.8677
15	4	1	Lineal	0.7572	0.8197	0.7563	0.8208	0.9189	0.9395	0.7487	0.8204
15	4	1	Radial	0.8683	0.9180	0.8677	0.9188	0.9561	0.9726	0.8650	0.9186
15	4	1	Poly	0.8683	0.9180	0.8677	0.9177	0.9560	0.9723	0.8653	0.9198
15	4	10	Lineal	0.7695	0.7869	0.7696	0.7875	0.9231	0.9289	0.7657	0.7855
15	4	10	Radial	0.9177	0.9344	0.9177	0.9354	0.9726	0.9781	0.9182	0.9349
15	4	10	Poly	0.9095	0.8852	0.9097	0.8875	0.9698	0.9616	0.9089	0.8848
15	4	100	Lineal	0.7984	0.7377	0.7984	0.7396	0.9327	0.9122	0.7969	0.7380
15	4	100	Radial	0.8889	0.9344	0.8891	0.9354	0.9629	0.9781	0.8885	0.9359
15	4	100	Poly	0.8930	0.9016	0.8929	0.9021	0.9643	0.9669	0.8924	0.9042
15	5	0.1	Lineal	0.7210	0.7333	0.7204	0.7424	0.9069	0.9121	0.7062	0.7140
15	5	0.1	Radial	0.7782	0.8167	0.7788	0.8228	0.9262	0.9395	0.7696	0.8143

Macro-Promedio				Accuracy		Sensibilidad		Especificidad		F1-score	
N°	Fold	C	Kernel	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.
15	5	0.1	Poly	0.7867	0.8500	0.7885	0.8506	0.9290	0.9501	0.7809	0.8480
15	5	1	Lineal	0.8196	0.7500	0.8202	0.7504	0.9399	0.9161	0.8132	0.7538
15	5	1	Radial	0.8646	0.9000	0.8661	0.9042	0.9550	0.9669	0.8643	0.8988
15	5	1	Poly	0.8729	0.9000	0.8742	0.9009	0.9577	0.9667	0.8714	0.9004
15	5	10	Lineal	0.8443	0.8333	0.8446	0.8351	0.9481	0.9444	0.8415	0.8341
15	5	10	Radial	0.9220	0.9167	0.9228	0.9153	0.9741	0.9721	0.9215	0.9162
15	5	10	Poly	0.8974	0.9000	0.8980	0.8997	0.9658	0.9667	0.8962	0.8997
15	5	100	Lineal	0.8607	0.8667	0.8613	0.8674	0.9536	0.9558	0.8606	0.8666
15	5	100	Radial	0.9137	0.9000	0.9147	0.8997	0.9713	0.9664	0.9128	0.9010
15	5	100	Poly	0.8727	0.9167	0.8734	0.9176	0.9576	0.9721	0.8712	0.9176
20	1	0.1	Lineal	0.7325	0.7869	0.7343	0.7875	0.9108	0.9290	0.7209	0.7867
20	1	0.1	Radial	0.7819	0.8033	0.7837	0.8042	0.9274	0.9347	0.7740	0.8033
20	1	0.1	Poly	0.8066	0.8525	0.8071	0.8531	0.9355	0.9507	0.8025	0.8554
20	1	1	Lineal	0.8477	0.7869	0.8486	0.7833	0.9492	0.9287	0.8458	0.7818
20	1	1	Radial	0.8642	0.8689	0.8645	0.8698	0.9546	0.9565	0.8634	0.8682
20	1	1	Poly	0.8765	0.8525	0.8766	0.8531	0.9587	0.9508	0.8759	0.8516
20	1	10	Lineal	0.7984	0.8361	0.7990	0.8344	0.9327	0.9455	0.7950	0.8347
20	1	10	Radial	0.8807	0.8852	0.8808	0.8854	0.9601	0.9620	0.8808	0.8857
20	1	10	Poly	0.8683	0.8852	0.8692	0.8854	0.9561	0.9618	0.8661	0.8849
20	1	100	Lineal	0.8025	0.8025	0.8025	0.8025	0.8025	0.8025	0.8025	0.7624
20	1	100	Radial	0.8765	0.8765	0.8765	0.8765	0.8765	0.8765	0.8765	0.8859
20	1	100	Poly	0.8724	0.8852	0.8728	0.8854	0.9574	0.9617	0.8728	0.8842
20	2	0.1	Lineal	0.7202	0.7869	0.7216	0.7833	0.9068	0.9287	0.7089	0.7815
20	2	0.1	Radial	0.7572	0.8361	0.7599	0.8333	0.9193	0.9449	0.7491	0.8330
20	2	0.1	Poly	0.7572	0.7869	0.7585	0.7844	0.9192	0.9290	0.7502	0.7807
20	2	1	Lineal	0.8436	0.8689	0.8446	0.8667	0.9479	0.9562	0.8426	0.8661
20	2	1	Radial	0.9012	0.8689	0.9016	0.8677	0.9671	0.9562	0.9017	0.8685
20	2	1	Poly	0.8971	0.8525	0.8976	0.8510	0.9657	0.9508	0.8988	0.8501
20	2	10	Lineal	0.8560	0.8361	0.8565	0.8333	0.9519	0.9457	0.8554	0.8318
20	2	10	Radial	0.9177	0.8852	0.9183	0.8833	0.9726	0.9617	0.9177	0.8832
20	2	10	Poly	0.8930	0.8689	0.8937	0.8667	0.9644	0.9562	0.8951	0.8658

Macro-Promedio				Accuracy		Sensibilidad		Especificidad		F1-score	
N°	Fold	C	Kernel	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.
20	2	100	Lineal	0.8477	0.7869	0.8482	0.7833	0.9492	0.9292	0.8482	0.7795
20	2	100	Radial	0.9218	0.8852	0.9222	0.8833	0.9740	0.9617	0.9218	0.8832
20	2	100	Poly	0.8930	0.8852	0.8938	0.8833	0.9643	0.9616	0.8940	0.8826
20	3	0.1	Lineal	0.7078	0.7541	0.7081	0.7563	0.9026	0.9179	0.6980	0.7543
20	3	0.1	Radial	0.7407	0.9016	0.7401	0.9010	0.9134	0.9674	0.7295	0.9023
20	3	0.1	Poly	0.7572	0.8361	0.7567	0.8385	0.9189	0.9455	0.7471	0.8367
20	3	1	Lineal	0.8107	0.8033	0.8109	0.8010	0.9370	0.9339	0.8043	0.8040
20	3	1	Radial	0.8601	0.8852	0.8599	0.8854	0.9534	0.9620	0.8596	0.8843
20	3	1	Poly	0.8765	0.9016	0.8772	0.9021	0.9589	0.9674	0.8763	0.9009
20	3	10	Lineal	0.7778	0.9180	0.7772	0.9177	0.9260	0.9726	0.7727	0.9175
20	3	10	Radial	0.8971	0.9344	0.8972	0.9344	0.9657	0.9781	0.8979	0.9329
20	3	10	Poly	0.8930	0.9508	0.8931	0.9500	0.9644	0.9836	0.8933	0.9504
20	3	100	Lineal	0.7984	0.8689	0.7976	0.8687	0.9328	0.9560	0.7946	0.8707
20	3	100	Radial	0.8848	0.8852	0.8845	0.8865	0.9616	0.9620	0.8847	0.8844
20	3	100	Poly	0.8889	0.9180	0.8883	0.9177	0.9630	0.9728	0.8880	0.9164
20	4	0.1	Lineal	0.7531	0.7869	0.7526	0.7875	0.9176	0.9284	0.7404	0.7816
20	4	0.1	Radial	0.7984	0.7541	0.7972	0.7552	0.9328	0.9178	0.7884	0.7310
20	4	0.1	Poly	0.8107	0.8852	0.8103	0.8854	0.9368	0.9617	0.8062	0.8834
20	4	1	Lineal	0.7901	0.8525	0.7893	0.8542	0.9300	0.9506	0.7805	0.8522
20	4	1	Radial	0.8642	0.8689	0.8635	0.8729	0.9547	0.9564	0.8619	0.8641
20	4	1	Poly	0.8807	0.9180	0.8800	0.9198	0.9601	0.9726	0.8801	0.9181
20	4	10	Lineal	0.8354	0.8197	0.8351	0.8198	0.9452	0.9396	0.8309	0.8189
20	4	10	Radial	0.9136	0.9016	0.9133	0.9031	0.9712	0.9673	0.9137	0.9019
20	4	10	Poly	0.8971	0.9344	0.8970	0.9344	0.9657	0.9779	0.8976	0.9347
20	4	100	Lineal	0.7901	0.8033	0.7895	0.8042	0.9300	0.9342	0.7840	0.8048
20	4	100	Radial	0.9136	0.9180	0.9135	0.9188	0.9712	0.9727	0.9142	0.9181
20	4	100	Poly	0.9053	0.8852	0.9052	0.8875	0.9684	0.9616	0.9057	0.8855
20	5	0.1	Lineal	0.7254	0.7500	0.7250	0.7580	0.9084	0.9173	0.7127	0.7395
20	5	0.1	Radial	0.7621	0.8500	0.7619	0.8549	0.9208	0.9507	0.7463	0.8448
20	5	0.1	Poly	0.7584	0.9167	0.7601	0.9153	0.9197	0.9722	0.7501	0.9157
20	5	1	Lineal	0.8156	0.8333	0.8163	0.8362	0.9386	0.9447	0.8077	0.8321

Macro-Promedio				Accuracy		Sensibilidad		Especificidad		F1-score	
N°	Fold	C	Kernel	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.
20	5	1	Radial	0.8032	0.8667	0.8046	0.8719	0.9346	0.9559	0.8027	0.8673
20	5	1	Poly	0.8442	0.8833	0.8452	0.8853	0.9481	0.9612	0.8435	0.8838
20	5	10	Lineal	0.7665	0.8667	0.7669	0.8685	0.9223	0.9557	0.7667	0.8655
20	5	10	Radial	0.8769	0.9167	0.8780	0.9153	0.9591	0.9721	0.8768	0.9162
20	5	10	Poly	0.8525	0.9000	0.8526	0.9007	0.9508	0.9667	0.8539	0.8992
20	5	100	Lineal	0.7705	0.8833	0.7708	0.8863	0.9236	0.9611	0.7699	0.8827
20	5	100	Radial	0.8605	0.9167	0.8619	0.9164	0.9536	0.9722	0.8595	0.9164
20	5	100	Poly	0.8524	0.8833	0.8524	0.8851	0.9508	0.9612	0.8538	0.8822

C. Resultados detallados de las combinaciones evaluadas en los folds para RLM

Tabla C-1.: Resultados completos (macro-promedios) para todas las combinaciones de número de características e hiperparámetros. Validaciones internas (Int.) y externas (Ext.).

Macro-Promedio			Accuracy		Sensibilidad		Especificidad		F1-score	
N°	Fold	C	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.
10	1	0,01	0,6461	0,6721	0,6463	0,6708	0,8819	0,8903	0,6316	0,6662
10	1	0,1	0,7325	0,7377	0,7325	0,7375	0,9108	0,9124	0,7276	0,7404
10	1	1	0,8519	0,7213	0,8518	0,7219	0,9506	0,9071	0,8511	0,7211
10	1	10	0,8765	0,7377	0,8765	0,7385	0,9588	0,9126	0,8758	0,7380
10	1	100	0,8930	0,8033	0,8929	0,8031	0,9643	0,9345	0,8923	0,8053
10	2	0,01	0,6584	0,6557	0,6595	0,6500	0,8862	0,8848	0,6486	0,6379
10	2	0,1	0,7449	0,7049	0,7456	0,7000	0,9149	0,9017	0,7409	0,6933
10	2	1	0,8189	0,7541	0,8195	0,7500	0,9396	0,9181	0,8181	0,7479
10	2	10	0,8642	0,7869	0,8646	0,7833	0,9547	0,9292	0,8646	0,7830
10	2	100	0,8683	0,7869	0,8688	0,7833	0,9561	0,9292	0,8684	0,7830
10	3	0,01	0,6420	0,6885	0,6415	0,6885	0,8807	0,8961	0,6271	0,6830
10	3	0,1	0,7078	0,7705	0,7074	0,7677	0,9026	0,9234	0,7030	0,7629
10	3	1	0,8395	0,8525	0,8394	0,8510	0,9465	0,9508	0,8392	0,8480
10	3	10	0,8807	0,8852	0,8805	0,8833	0,9602	0,9617	0,8803	0,8815
10	3	100	0,9012	0,8689	0,9012	0,8667	0,9671	0,9563	0,9002	0,8618
10	4	0,01	0,6872	0,6393	0,6872	0,6417	0,8957	0,8796	0,6790	0,6198
10	4	0,1	0,7819	0,7049	0,7820	0,7073	0,9273	0,9012	0,7801	0,6930
10	4	1	0,8560	0,7705	0,8559	0,7729	0,9520	0,9232	0,8551	0,7679

Macro-Promedio			Accuracy		Sensibilidad		Especificidad		F1-score	
N°	Fold	C	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.
10	4	10	0,8724	0,8033	0,8723	0,8063	0,9575	0,9342	0,8718	0,8001
10	4	100	0,8765	0,8033	0,8764	0,8052	0,9589	0,9342	0,8757	0,8011
10	5	0,01	0,6680	0,5833	0,6665	0,5899	0,8892	0,8624	0,6513	0,5651
10	5	0,1	0,8033	0,6667	0,8024	0,6701	0,9343	0,8892	0,7989	0,6607
10	5	1	0,8607	0,6833	0,8610	0,6835	0,9535	0,8945	0,8598	0,6824
10	5	10	0,8770	0,6500	0,8774	0,6501	0,9590	0,8834	0,8762	0,6503
10	5	100	0,8689	0,6500	0,8692	0,6501	0,9563	0,8834	0,8681	0,6503
15	1	0,01	0,6667	0,6721	0,6671	0,6698	0,8888	0,8906	0,6525	0,6597
15	1	0,1	0,8189	0,7541	0,8190	0,7531	0,9396	0,9180	0,8178	0,7526
15	1	1	0,9177	0,7705	0,9178	0,7698	0,9726	0,9237	0,9174	0,7702
15	1	10	0,9259	0,7869	0,9260	0,7875	0,9753	0,9291	0,9256	0,7871
15	1	100	0,9383	0,7541	0,9384	0,7563	0,9794	0,9182	0,9380	0,7532
15	2	0,01	0,7202	0,6393	0,7210	0,6333	0,9068	0,8791	0,7065	0,6166
15	2	0,1	0,8683	0,8361	0,8684	0,8333	0,9561	0,9452	0,8684	0,8320
15	2	1	0,9259	0,8852	0,9260	0,8833	0,9753	0,9617	0,9262	0,8823
15	2	10	0,9465	0,8361	0,9465	0,8333	0,9822	0,9455	0,9465	0,8348
15	2	100	0,9506	0,8197	0,9507	0,8167	0,9835	0,9401	0,9508	0,8164
15	3	0,01	0,7202	0,7541	0,7200	0,7521	0,9068	0,9178	0,7058	0,7422
15	3	0,1	0,8313	0,8525	0,8311	0,8500	0,9438	0,9507	0,8300	0,8493
15	3	1	0,8848	0,9016	0,8848	0,9010	0,9616	0,9671	0,8850	0,9007
15	3	10	0,9095	0,9180	0,9094	0,9177	0,9699	0,9728	0,9092	0,9182
15	3	100	0,9259	0,9016	0,9259	0,9010	0,9753	0,9673	0,9253	0,9010
15	4	0,01	0,7366	0,6885	0,7367	0,6906	0,9122	0,8959	0,7276	0,6628
15	4	0,1	0,8313	0,7377	0,8315	0,7406	0,9438	0,9122	0,8291	0,7267
15	4	1	0,9012	0,8361	0,9012	0,8385	0,9671	0,9450	0,9015	0,8377
15	4	10	0,9383	0,8525	0,9383	0,8552	0,9794	0,9506	0,9384	0,8538
15	4	100	0,9671	0,8197	0,9670	0,8208	0,9890	0,9396	0,9669	0,8182
15	5	0,01	0,7459	0,6500	0,7449	0,6555	0,9152	0,8842	0,7378	0,6349
15	5	0,1	0,8361	0,7000	0,8354	0,7036	0,9453	0,9002	0,8354	0,6961
15	5	1	0,9098	0,7500	0,9101	0,7515	0,9699	0,9166	0,9100	0,7484
15	5	10	0,9344	0,8000	0,9344	0,7996	0,9781	0,9333	0,9345	0,7969
15	5	100	0,9590	0,7833	0,9589	0,7829	0,9863	0,9278	0,9592	0,7816

Macro-Promedio			Accuracy		Sensibilidad		Especificidad		F1-score	
N°	Fold	C	Int.	Ext.	Int.	Ext.	Int.	Ext.	Int.	Ext.
20	1	0,01	0,7243	0,7705	0,7247	0,7688	0,9080	0,9234	0,7169	0,7666
20	1	0,1	0,8477	0,7705	0,8478	0,7698	0,9492	0,9237	0,8473	0,7702
20	1	1	0,9342	0,7705	0,9342	0,7698	0,9781	0,9238	0,9340	0,7699
20	1	10	0,9588	0,7541	0,9589	0,7531	0,9863	0,9182	0,9589	0,7536
20	1	100	0,9794	0,7049	0,9795	0,7042	0,9931	0,9019	0,9795	0,7048
20	2	0,01	0,7243	0,6721	0,7251	0,6667	0,9082	0,8900	0,7116	0,6557
20	2	0,1	0,8889	0,8197	0,8890	0,8167	0,9630	0,9396	0,8888	0,8165
20	2	1	0,9712	0,8525	0,9712	0,8500	0,9904	0,9508	0,9712	0,8479
20	2	10	0,9753	0,8689	0,9753	0,8667	0,9918	0,9564	0,9753	0,8654
20	2	100	0,9959	0,8033	0,9959	0,8010	0,9986	0,9345	0,9959	0,7994
20	3	0,01	0,7366	0,7869	0,7365	0,7865	0,9123	0,9287	0,7263	0,7849
20	3	0,1	0,8395	0,8197	0,8395	0,8167	0,9466	0,9395	0,8388	0,8169
20	3	1	0,9177	0,9016	0,9176	0,9010	0,9726	0,9670	0,9177	0,9027
20	3	10	0,9465	0,9016	0,9465	0,9021	0,9822	0,9670	0,9463	0,9029
20	3	100	0,9630	0,9016	0,9629	0,9031	0,9877	0,9671	0,9628	0,9003
20	4	0,01	0,7449	0,6557	0,7448	0,6573	0,9150	0,8850	0,7353	0,6178
20	4	0,1	0,8601	0,7869	0,8602	0,7896	0,9534	0,9285	0,8585	0,7874
20	4	1	0,9424	0,8197	0,9423	0,8219	0,9808	0,9395	0,9425	0,8192
20	4	10	0,9794	0,8361	0,9794	0,8385	0,9931	0,9450	0,9794	0,8358
20	4	100	0,9877	0,7869	0,9876	0,7885	0,9959	0,9286	0,9876	0,7868
20	5	0,01	0,7664	0,6500	0,7654	0,6555	0,9220	0,8843	0,7595	0,6377
20	5	0,1	0,8566	0,7000	0,8561	0,7013	0,9521	0,8998	0,8567	0,6993
20	5	1	0,9180	0,7667	0,9182	0,7649	0,9727	0,9221	0,9181	0,7624
20	5	10	0,9590	0,7333	0,9589	0,7305	0,9863	0,9108	0,9590	0,7309
20	5	100	0,9631	0,7333	0,9630	0,7305	0,9877	0,9107	0,9633	0,7312

Bibliografía

- Azuela, J. H. S., & Morales, R. R. (2011). *Procesamiento y Análisis Digital de Imágenes* (1.^a ed.). RA-MA Editorial y Publicaciones.
- Bhat, N. N., Kumari, K., Dutta, S., Pal, S. K., & Pal, S. (2015). Friction stir weld classification by applying wavelet analysis and support vector machine on weld surface images. *Journal of Manufacturing Processes*, 20, 274-281. <https://doi.org/https://doi.org/10.1016/j.jmapro.2015.07.002>
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Burrus, C. S., Gopinath, R. A., & Guo, H. (1998). *Introduction to Wavelets and Wavelet Transforms: A Primer*. Prentice Hall.
- Contributors, O. (2021a). OpenCV Documentation [Accessed: 2025-07-25]. <https://docs.opencv.org/4.12.0/>
- Contributors, O. (2021b). OpenCV Image Processing: Color Conversions [Accessed: 2025-07-25]. https://docs.opencv.org/4.12.0/de/d25/imgproc_color_conversions.html#autotoc_md143
- Contributors, O. (2021c). OpenCV Image Processing: Filters [Accessed: 2025-07-25]. https://docs.opencv.org/4.12.0/d4/d86/group__imgproc__filter.html
- Contributors, P. (2021). Pillow Documentation [Accessed: 2025-07-25]. <https://pillow.readthedocs.io/en/stable/reference/index.html>

- contributors, P. (2023). PyWavelets: 2D Discrete Wavelet Transform (DWT) and Inverse DWT (IDWT) [Accessed: 2025-08-07]. <https://pywavelets.readthedocs.io/en/latest/ref/2d-dwt-and-idwt.html>
- contributors, S.-I. (2020). skimage.measure — Measurement functions in skimage [Accessed: 2025-08-07]. <https://scikit-image.org/docs/stable/api/skimage.measure.html#>
- Coşkun, H., Yiğit, T., Üncü, I. S., Ersoy, M., & Topal, A. (2022). An Industrial Application Towards Classification and Optimization of Multi-Class Tile Surface Defects Based on Geometric and Wavelet Features. *Traitement du Signal*, 39(6), 2011-2022. <https://doi.org/10.18280/ts.390602>
- Czimmermann, T., Ciuti, G., Milazzo, M., Chiurazzi, M., Roccella, S., Oddo, C. M., & Dario, P. (2020). Visual-Based Defect Detection and Classification Approaches for Industrial Applications—A SURVEY. *Sensors*, 20(5). <https://www.mdpi.com/1424-8220/20/5/1459>
- Daubechies, I. (1992). *Ten Lectures on Wavelets* (Vol. 61). Society for Industrial; Applied Mathematics.
- Developers, N. (2021). NumPy Documentation [Accessed: 2025-07-25]. <https://numpy.org/devdocs/reference/index.html>
- Developers, S.-I. (s.f.). SVC# [Accessed: 2025-8-8].
- Developers, S.-I. (2021a). Scikit-learn Documentation [Accessed: 2025-07-25]. <https://scikit-learn.org/stable/api/index.html>
- Developers, S.-I. (2021b). sklearn.metrics [Accessed: 2025-08-08]. <https://scikit-learn.org/stable/api/sklearn.metrics.html>
- Developers, S.-I. (2021c). sklearn.model_selection [Accessed: 2025-07-25]. https://scikit-learn.org/stable/api/sklearn.model_selection.html

- Donoho, D. L. (1993). Unconditional Bases are Optimal Bases for Data Compression and for Statistical Estimation. *Applied and Computational Harmonic Analysis*, 1(1), 100-115. <https://doi.org/10.1006/acha.1993.1008>
- Donoho, D. L., Johnstone, I. M., Kerkycharian, G., & Picard, D. (1995). Wavelet Shrinkage: Asymptopia? *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(2), 301-337. <https://doi.org/10.1111/j.2517-6161.1995.tb02032.x>
- Draper, N. R., & Smith, H. (1998). *Applied Regression Analysis*. John Wiley & Sons.
- Ghazali, K. H., Mansor, M. F., Mustafa, M. M., & Hussain, A. (2007). Feature extraction technique using discrete wavelet transform for image classification. *2007 5th Student Conference on Research and Development*, 1-4.
- Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4.^a ed.). Pearson.
- Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3), 389-422.
- Haar, A. (1910). Zur Theorie der orthogonalen Funktionensysteme. *Mathematische Annalen*, 69, 331-371. <https://doi.org/10.1007/BF01456326>
- Hunter, J. D., & Contributors, M. (2021). Matplotlib Documentation [Accessed: 2025-07-25]. <https://matplotlib.org/stable/api/index>
- Ingio, J. A., Nsang, A. S., & Iorliam, A. (2024). Optimizing Rice Production Forecasting Through Integrating Multiple Linear Regression with Recursive Feature Elimination. *Journal of Future Artificial Intelligence and Technologies*, 1(2), 97-107. <https://doi.org/10.62411/faith.2024-17>
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer New York. <https://books.google.com.co/books?id=qcLAAAAQBAJ>

-
- Kiraz, A. H., Djibrillah, F. O., & Yüksel, M. E. (2023). Deep feature extraction, dimensionality reduction, and classification of medical images using combined deep learning architectures, autoencoder, and multiple machine learning models. *TURK. J. OF ELECTR. ENG. COMPUT. SCI.*, 31(6), 1113-1128.
- Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2), 273-324. [https://doi.org/10.1016/S0004-3702\(97\)00043-X](https://doi.org/10.1016/S0004-3702(97)00043-X)
- Lee, G., & et al. (2019). PyWavelets: A Python package for wavelet analysis. *Journal of Open Source Software*, 4(36), 1237. <https://doi.org/10.21105/joss.01237>
- Mohr, F., & van Rijn, J. N. (2024). Learning curves for decision making in supervised machine learning: a survey. *Mach. Learn.*, 113(11-12), 8371-8425.
- Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to Linear Regression Analysis* (5th). John Wiley & Sons.
- Morales, J., & Nocedal, J. (2011). L-BFGS-B: Observación sobre el algoritmo 778: L-BFGS-B, Rutinas FORTRAN para la optimización restringida vinculada a gran escala. *Transacciones ACM sobre software matemático*, 38(1), 1-12.
- Münke, F. R., Schützke, J., Berens, F., & Reischl, M. (2024). A review of adaptable conventional image processing pipelines and deep learning on limited datasets. *Mach. Vis. Appl.*, 35(2).
- Nazir, M. K., Ghani, M. A. N. U., Ashraf, A., Alam, S. M., Farooq, R. U., & Latif, Z. (2021). A Textile Image Classification based on Texture and Shape Features. *2021 International Conference on Innovative Computing (ICIC)*, 1-6. <https://doi.org/10.1109/ICIC53490.2021.9693067>
- Núñez Ramírez, L. M., Galvez Perez, R., & Galvez Perez, H. E. (2010). Análisis Multiresolución. *PESQUIMAT, Revista de la F.C.M. de la Universidad Nacional Mayor de San Marcos*, 13(2), 59-76.

-
- Ojala, T., Pietikäinen, M., & Harwood, D. (1996). A Comparative Study of Texture Measures with Classification Based on Feature Distributions. *Pattern Recognition*, 29(1), 51-59. [https://doi.org/10.1016/0031-3203\(95\)00067-4](https://doi.org/10.1016/0031-3203(95)00067-4)
- Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 971-987. <https://doi.org/10.1109/TPAMI.2002.1017623>
- OpenCV. (s.f.). Tutorial on Color Spaces in OpenCV: HSV Conversion [Accessed: 2025-07-22, OpenCV Color Spaces Tutorial]. https://docs.opencv.org/4.x/df/d9d/tutorial_py_colorspaces.html
- Palomino, N. L. S., Contreras, W., & Ruiz, M. (2010). Procesamiento Digital de textura: Técnicas utilizadas en aplicaciones actuales de CBIR. *Revista de investigación de Sistemas e Informática*, 7(1), 57-64.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
- Prieto, S. O. I. (1999, marzo). *Análisis Multi-Resolución de Imágenes Digitales: Aplicación en Restauración de Imágenes de Satélite* [Tesis de Maestría]. Ensenada, Baja California, México.
- Rainio, O., Teuho, J., & Klén, R. (2024). Author Correction: Evaluation metrics and statistical tests for machine learning. *Sci. Rep.*, 14(1), 15724. <https://doi.org/https://doi.org/10.1038/s41598-024-56706-x>
- Sweldens, W. (1996). Wavelets: What Next? *Proceedings of the IEEE*, 84(4), 680-685. <https://doi.org/10.1109/5.488704>
- Team, T. P. D. (2021). Pandas Documentation [Accessed: 2025-07-25]. <https://pandas.pydata.org/docs/reference/index.html>

The MathWorks, I. (2025). dwt2 [Accessed: 2025-01-19].

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., & the scikit-image contributors. (2014). scikit-image: image processing in Python. *PeerJ*, 2, e453. <https://doi.org/10.7717/peerj.453>

Vaz, P., Braga, A. C., Carvalho, M. S., & Menezes, G. (2021). Predicting Product Quality from Operating Conditions Based on Multinomial Logistic Regression. En O. Gervasi, B. Murgante, S. Misra, C. Garau, I. Blečić, D. Taniar, B. O. Apduhan, A. M. A. C. Rocha, E. Tarantino & C. M. Torre (Eds.), *Computational Science and Its Applications – ICCSA 2021* (pp. 539-551, Vol. 12952). Springer. https://doi.org/10.1007/978-3-030-86973-1_38

Weisberg, S. (2014). *Applied Linear Regression* (4th). John Wiley & Sons.

Yousif, A. S., Omar, Z., Hamoodat, H., & Morad, N. Y. A. (2022). Multi-class tumor diseases classification using discrete wavelet transform and principal component analysis. *2022 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, 303-307.

Zhang, D. (2019). Wavelet transform. En *Texts in Computer Science* (pp. 35-44). Springer International Publishing.