



**UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA**

**CLASIFICACIÓN DE PATRONES COMPLEJOS DE TEXTURA-
COLOR MEDIANTE EXTRACCIÓN DE CARACTERÍSTICAS
GLOBALES Y LOCALES, UN CLASIFICADOR SVM, Y POST-
PROCESAMIENTO**

**TESIS PARA OPTAR AL GRADO DE DOCTOR EN INGENIERÍA
ELÉCTRICA**

CARLOS FELIPE NAVARRO CLAVERÍA

**PROFESOR GUÍA:
Dr. CLAUDIO PÉREZ FLORES**

**MIEMBROS DE LA COMISIÓN:
Dr. DOMINGO MERY QUIROZ
Dr. GONZALO RUZ HEREDIA
Dr. JORGE SILVA SÁNCHEZ**

**SANTIAGO DE CHILE
2020**

RESUMEN DE LA TESIS PARA OPTAR AL GRADO

DE: Doctor en Ingeniería Eléctrica

POR: Carlos Felipe Navarro Clavería

FECHA: 22 de septiembre de 2020

PROFESOR GUÍA: Claudio Pérez Flores

CLASIFICACIÓN DE PATRONES COMPLEJOS DE TEXTURA-COLOR MEDIANTE EXTRACCIÓN DE CARACTERÍSTICAS GLOBALES Y LOCALES, UN CLASIFICADOR SVM, Y POST-PROCESAMIENTO

Resumen:

Distintas aplicaciones en análisis de imágenes requieren la clasificación precisa de patrones complejos que incluyen color y textura. En esta tesis se propone un nuevo método para la clasificación de patrones mediante información de color y textura. El método propuesto incluye la división de cada imagen en particiones globales y locales, la extracción de características de textura y color de las particiones usando estadísticos de Haralick y el método binario de preservación del momento de cuaterniones (*BQMP*), una etapa de clasificación usando máquinas de soporte vectorial (*SVM*) y una etapa final de post-procesamiento empleando un conjunto combinado (*Bagging*). Una de las principales contribuciones de este método es la partición de imágenes, que permite la representación de imágenes en características globales y locales. Esta partición captura la mayor parte de la información presente en la imagen para la clasificación de texturas con color, mejorando los resultados de clasificación. El método propuesto se probó en bases de datos ampliamente utilizadas: Brodatz, VisTex, Outex y KTH-TIPS2b, obteniendo tasas de clasificación correctas de 99,88%, 100%, 98,97% y 95,75%, respectivamente. Estos resultados fueron comparados con los mejores resultados publicados previamente en las mismas bases de datos, encontrando mejoras significativas en todos los casos.

**EXECUTIVE SUMMARY OF THE THESIS TO OPT TO
THE DEGREE OF:** Doctor in Electrical Engineering
BY: Carlos Felipe Navarro Clavería
DATE: September 22nd, 2020
ADVISOR: Claudio Pérez Flores

**Color–Texture Pattern Classification Using Global–Local Feature
Extraction, an SVM Classifier, with Bagging Ensemble Post-Processing**

Abstract:

Many applications in image analysis require the accurate classification of complex patterns including both color and texture. A new method for pattern classification using both color and texture information is proposed in this paper. The proposed method includes the division of each image into global and local samples, texture and color feature extraction from samples using a Haralick statistics and binary quaternion-moment-preserving method, a classification stage using support vector machine, and a final stage of post-processing employing a bagging ensemble. One of the main contributions of this method is the image partition, allowing image representation into global and local features. This partition captures most of the information present in the image for colored texture classification allowing improved results. The proposed method was tested on databases extensively used in color–texture classification: the Brodatz, VisTex, Outex, and KTH-TIPS2b databases, yielding correct classification rates of 99.88%, 100%, 98.97%, and 95.75%, respectively. We compared our results to the best previously published results on the same databases finding significant improvements in all cases.

TABLA DE CONTENIDO

AGRADECIMIENTOS	V
TABLA DE CONTENIDO	VI
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	IX
ABREVIACIONES Y ACRÓNIMOS	X
CAPÍTULO 1 INTRODUCCIÓN	1
1.1 MOTIVACIÓN	1
1.2 HIPÓTESIS	3
1.3 OBJETIVO GENERAL	4
1.4 OBJETIVOS ESPECÍFICOS	4
1.5 ESTRUCTURA DEL DOCUMENTO	4
CAPÍTULO 2 ESTADO DEL ARTE	5
2.1 EXTRACCIÓN DE CARACTERÍSTICAS	5
2.1.1 MÉTODO DE BINARIZACIÓN USANDO PRESERVACIÓN DE MOMENTOS POR CUATERNIONES (BQMP)	6
2.1.1.1 ALGEBRA DE CUATERNIONES	6
2.1.1.2 MOMENTOS Y VALORES REPRESENTANTES DE CUATERNIONES	7
2.1.2 ESTADÍSTICOS DE HARALICK	9
2.2 CLASIFICADORES	12
2.2.1 MÁQUINAS DE SOPORTE VECTORIAL (SVM)	12
2.2.2 K VECINOS MÁS CERCANOS (KNN)	16
2.3 MÉTODOS DE ENSAMBLAJE	16
2.4 BASES DE DATOS	18
CAPÍTULO 3 METODOLOGÍA PROPUESTA	21
3.1 METODOLOGÍA	21
3.1.1 EXTRACCIÓN DE CARACTERÍSTICAS EN CUADRANTES	22
3.1.2 CLASIFICACIÓN Y POST-PROCESAMIENTO	26
3.2 EXPERIMENTOS	28
3.2.1 BASE BRODATZ	28
3.2.2 BASE VISTEX	28
3.2.3 BASE OUTEX	29
3.2.4 BASE KTH-TIPS2B	29
3.2.5 CONTRIBUCIÓN COLOR-TEXTURA Y CONJUNTO.	29

CAPÍTULO 4 RESULTADOS	30
4.1 BASE BRODATZ	30
4.2 BASE VISTEX	32
4.3 BASE OUTEX	33
4.4 BASE KTH-TIPS2B	34
4.5 CONTRIBUCIÓN COLOR-TEXTURA Y CONJUNTO	35
4.6 TIEMPO DE PROCESAMIENTO	35
4.7 COMPARACIÓN CON TÉCNICAS DE <i>DEEP LEARNING</i>	36
CAPÍTULO 5 DISCUSIÓN Y CONCLUSIONES	38
5.1 DISCUSIÓN	38
5.2 CONCLUSIONES	39
5.3 TRABAJO FUTURO	40
BIBLIOGRAFÍA	41

Índice de Figuras

Figura 1: Ampliación del espacio RGB de 3 dimensiones al espacio de cuaterniones.	7
Figura 2: Separación del espacio en el método BQMP.	8
Figura 3: Ejemplo de la primera iteración del método BQMP en el caso de 4 colores.	9
Figura 4: Ejemplo de la segunda iteración del método BQMP en el caso de 4 colores.	9
Figura 5: Plano separador de dos clases utilizando SVM.	13
Figura 6: Ejemplo de un problema separable en dos dimensiones.	13
Figura 7: Pasos básicos de KNN.	16
Figura 8: Diagrama de bloques de un método de ensamblaje.	17
Figura 9: Pseudocódigo del método Bagging.	17
Figura 10: Ejemplo de texturas de color diferentes de la base Brodatz Colored Texture.	18
Figura 11: Ejemplo de texturas de color diferentes de la base Vistex.	19
Figura 12: Ejemplo de texturas de color diferentes de la base Outex.	19
Figura 13: Ejemplo de texturas de color diferentes de la base KTH-TIPS2b.	20
Figura 14: Diagrama de Bloques del método propuesto.	21
Figura 15: Separación de base de entrenamiento y prueba para distintas bases de datos	22
Figura 16: El proceso de subdivisión de imágenes en imágenes locales.	23
Figura 17: Ejemplo de escalas espaciales: cuatro locales y una global.	23
Figura 18: Vector de características extraído de una imagen de la base de datos Brodatz.	24
Figura 19: Vector de características extraído de una imagen de la base de datos Vistex	26
Figura 20: Ejemplo del resultado del post-procesamiento	27

Índice de tablas

Tabla 1: Resultados de la clasificación de los experimentos en la base de datos Brodatz para el conjunto de 10 imágenes	30
Tabla 2: Resultados de la clasificación de los experimentos en la base de datos Brodatz para el conjunto de 30 imágenes	31
Tabla 3: Resultados de la clasificación de los experimentos en la base de datos Brodatz para el conjunto de 40 imágenes	31
Tabla 4: Resultados de la clasificación de los experimentos en la base de datos Brodatz para el conjunto de 111 imágenes.....	31
Tabla 5: Los mejores resultados de la clasificación utilizando el método global local BQMP-Haralick para la base de datos Brodatz en los cuatro conjuntos de 10, 30, 40 y 111 imágenes	32
Tabla 6: Resultados de clasificación del experimento en la base de datos Vistex.....	32
Tabla 7: Mejores resultados de la clasificación global local Haralick-BQMP para la base de datos Vistex	33
Tabla 8: Resultados de clasificación del experimento en la base de datos Outex.	33
Tabla 9: Mejores resultados de la clasificación Global Local Haralick-BQMP para la base de datos Outex	34
Tabla 10: Resultados de clasificación en la base de datos KTH-TIPS2b.....	34
Tabla 11: Mejores resultados de la clasificación Global-Local Haralick-BQMP para la base de datos KTH-TIPS2b	35
Tabla 12: Mejores resultados de la clasificación utilizando el método desarrollado en todas las bases de datos con la contribución de cada parte del modelo (color y textura).	35
Tabla 13: Tiempo computacional del método propuesto para extracción de características (FE), clasificación usando SVM, tiempo de postprocesado (PP) y tiempo total.....	35
Tabla 14: Comparación de métodos basados en Deep Learning vs mejores resultados de la clasificación en todas las bases de datos.	37

Abreviaciones y Acrónimos

3D-ASDH: Suma y diferencia adaptativa tridimensional. [7]

Bagging: Bootstrap aggregation

BQMP: Binary Quaternion Moment Preserving. [6]

CCM: Matrices de ocurrencia cromática. [4]

CFA: Matriz de filtro de color. [4]

DITC: Agrupamiento teórico de información divisiva. [8]

FE: Feature Extraction

KNN: K-Nearest Neighbor.

KTH: Instituto Real de Tecnología. Suecia.

KTH-TIPS: Textures under varying Illumination, Pose and Scale, creado por KTH.

LBP: Local Binary Pattern.

Outex: Base de Datos de la Universidad de Oulu. Finlandia.

PP: Post-procesamiento.

SVM: Support Vector Machines. [40]

Vistex: Vision Texture. Massachusetts Institute of Technology.

Capítulo 1 Introducción

1.1 Motivación

El problema de clasificación por textura es considerado un tópico importante en visión en máquinas (*Machine Vision*), debido a la gran variedad de aplicaciones posibles, incluyendo la inspección no destructiva de anomalías en distintas superficies tales como madera, acero, cerámica, fruta y superficies de aviones, entre muchas otras. [1–6]. El clasificar por medio de texturas sigue siendo un desafío, dado que la textura de los objetos varía significativamente según el ángulo de visión, las condiciones de iluminación, el cambio de escala y la rotación [1,4,7,8]. El reconocimiento de objetos basándose en la apariencia es un problema especial dentro del procesamiento de imágenes y es un importante campo de desarrollo para varias aplicaciones de visión industrial [1,4,7,8].

La extracción de características de color, textura y forma de imágenes se ha utilizado con éxito para clasificar patrones, reduciendo la dimensionalidad y la complejidad computacional del problema [3,9–16]. Determinar las características apropiadas para cada problema es un reto recurrente que aún no ha sido totalmente resuelto por la comunidad de visión computacional [1,3,16]. La extracción y selección de características permite la representación de la información presente en la imagen y limita el número de características permitiendo así un análisis posterior en un tiempo razonable. La extracción de características se ha utilizado en una amplia gama de aplicaciones, como la biometría [12,14,15], clasificación de tela, superficies, paisajes, madera y rocas minerales [16,17], detección de saliencia [18], sustracción de fondo [19], entre otras. Durante los últimos 40 años, mientras se ha desarrollado un número considerable de métodos para la clasificación por textura en escala de grises [3,5] también ha habido un interés creciente en las texturas con color [1,2,9,10,13,20,21]. La integración adaptativa de los atributos de color y textura en el desarrollo de descriptores de imágenes complejos ha sido un área de investigación intensiva en visión computacional [21]. La mayoría de estas investigaciones se centran en el proceso de integración en aplicaciones de segmentación de imágenes digitales [20,22] o aplicando algoritmos de aprendizaje automático a un conjunto de descriptores obtenidos de la imagen [23]. Algoritmos de aprendizaje profundo (*Deep learning*) se han aplicado con éxito al reconocimiento de objetos o escenas [24], a clasificación de escenas [25]. También ha sido investigado el uso de redes neuronales profundas (*Deep Neural Networks*) para el análisis de texturas [26].

Los enfoques para integrar textura y color pueden clasificarse en tres grupos: el enfoque paralelo, el enfoque secuencial y el enfoque integrador [11,27]. El enfoque paralelo considera la textura y el color como fenómenos separados. El análisis de color se basa en la distribución de color en una imagen, independientemente de la relación espacial entre las intensidades de los píxeles. El análisis de textura se basa en la variación relativa de la intensidad de los vecinos, independientemente del color de los píxeles. En [2], los autores convierten las imágenes RGB (*Red, Green, Blue*) originales en otros espacios de color: HSI (*Hue, Saturation, Lightness*), YIQ (*Luminance In phase Quadrature*) y los creados por la Comisión Internacional de Iluminación: CIE-XYZ (*Comission Internationale de*

l'Éclairage Tristimulus values), y CIE-LAB (*Comission Internationale de l'Éclairage Lightness-Green-Blue*), y luego extraen las características de textura y color por separado. De una manera similar, tal como fue reportado en [13], las imágenes se transforman en los espacios de color HSV (*Hue, Saturation, Value*) y YCbCr (*Luminance, Blue-luma, Red-luma*), obteniendo características de tipo wavelet y con ello estadísticos de primer orden en cada canal. La elección del espacio de color con mejor rendimiento ha sido una pregunta abierta en los últimos años, ya que el uso de un espacio en lugar de otro puede aportar mejoras considerables en ciertas aplicaciones [28]. La representación de colores utilizando cuaterniones en segmentación ha mostrado ser efectiva en [29] y el método de extracción de características de color llamado *Binary Quaternion Moment Preserving (BQMP)*, es un método de binarización de imágenes que utiliza cuaterniones (números complejos de cuatro dimensiones), posee el potencial de ser una herramienta poderosa en análisis de imágenes por color [6]. Dicho método se explicará con mayores detalles en la sección 2.1.1.

En el enfoque secuencial, el primer paso es aplicar un método de indexación de imágenes en color. Como resultado, las imágenes indexadas se procesan obteniendo texturas en escala de grises. La matriz de co-ocurrencia se ha utilizado ampliamente ya que representa la probabilidad de ocurrencia del mismo par de píxeles de color a una distancia dada [9]. Otro ejemplo de este enfoque se basa en descriptores de textura utilizando tres diferentes métodos de indexación de color, y tres diferentes características de textura [11]. Esto resulta en nueve clasificadores independientes que se combinan a través de varios esquemas.

Los modelos de integración se basan en la interdependencia de textura y color. Estos modelos se pueden dividir en bandas simples, si los datos de cada canal se consideran por separado, o una banda múltiple, si dos o más canales se consideran conectados. La ventaja del enfoque de una sola banda es la fácil adaptación de modelos clásicos basados en un dominio de escala de grises, como filtros de *Gabor* [15,30–32], patrones binarios locales (*LBP*) o sus variantes [5,7,8,33–38], campos *Galois* [39], y estadísticos de *Haralick* [3]. En [2] el principal objetivo es determinar la contribución de la información de color para el rendimiento general de la clasificación utilizando filtros de *Gabor* y medidas de co-ocurrencia, obteniéndose resultados casi un 10% mejores que los obtenidos con sólo imágenes en escala de grises. En [4] los resultados informados utilizando matrices de co-ocurrencia alcanzaron 94,41% y 99,07% en las bases de datos Outex y Vistex, respectivamente. Diferentes clasificadores, como los *k*-vecinos más cercanos, redes neuronales y máquinas vectoriales de soporte (*SVM*) [40], se han utilizado para combinar características. Este último ha demostrado ser más eficiente cuando se realiza la selección de características [41,42] o se utiliza *clustering* [22].

Existen otros métodos que alcanzaron buenos resultados en el análisis de textura de color en bases de datos que están disponibles públicamente. En [1], se utiliza una arquitectura multiescalar llamada *Multi-Scale Supervised Self-Organizing Orientational-invariant Neural (Multi-Scale SOON)*, la que alcanza un 91,03% en la base Brodatz, que contiene 111 texturas diferentes. Estos resultados se comparan con dos estudios previos en la misma base de datos, reportado en las referencias [43,44], y alcanzan tasas de clasificación de 89,71% y 88,15%, respectivamente. Otro enfoque utilizó toda la información proveniente de los sensores que se utilizan para crear la imagen [4].

Este método mejoró los resultados a 98,61% y 99,07%, en la misma base de datos, pero requirió un cambio no trivial en la arquitectura de la recolección de datos. En [8] se propone un descriptor de textura basado en un esquema de codificación de patrones locales usando histogramas de suma máxima y diferencia locales. Los resultados experimentales se probaron en las bases de datos Outex y KTH-TIPS2b que alcanzaron el 95,8% y el 91,3%, respectivamente. En las referencias [32,33,36,39], los métodos no fueron probados en las bases de datos completas. En [26,45,46], los métodos utilizaron una métrica diferente para calcular la clasificación.

En esta tesis se propone un nuevo método para la clasificación de patrones complejos de texturas con colores. Las imágenes se dividen en muestras globales y locales donde se extraen las características para proporcionar una nueva representación en las características globales y locales. Se describe la extracción de características de diferentes muestras de la imagen usando cuadrantes. Se presenta la extracción de las características en cada uno de los cuadrantes de imágenes, obteniendo características de color y textura en diferentes escalas espaciales: las escalas globales usando la imagen completa y las escalas locales usando cuadrantes.

Esta representación logra capturar la mayor parte de la información presente en la imagen para mejorar la clasificación de textura con color. Luego, se informa de una clasificación de la máquina de soporte vectorial realizada, y finalmente se presenta la etapa de post-procesamiento que utiliza *bagging*. El método propuesto fue probado en cuatro bases de datos distintas: Colored Brodatz Texture [20], VisTex [4], Outex [47], y KTH-TIPS2b [48]. La subdivisión de la partición de entrenamiento de la base de datos, en sub-imágenes mientras se extrae información de cada sub-imagen de diferentes tamaños, también es nueva. Los resultados se compararon con los métodos del estado del arte con resultados ya publicados en las mismas bases de datos. Este método y los resultados obtenidos fueron publicados en [49].

1.2 Hipótesis

Esta tesis busca mejorar los resultados de clasificación de texturas coloreadas utilizando un método de generación de información mediante el uso de particiones de imágenes aleatorias, no necesariamente disjuntas, con distintas escalas locales y globales, realizando sobre éstas la extracción de características de texturas y color, además de una etapa de post-procesamiento. Las hipótesis son:

1. La partición de las texturas con color en escalas locales y globales para la extracción de características de color y textura mejora el rendimiento en la clasificación de estas texturas.
2. El uso de una etapa de post-procesamiento permite también mejorar los resultados de clasificación de texturas coloreadas

1.3 Objetivo General

Desarrollar un nuevo método que mejore la clasificación de texturas con color utilizando un método de partición de imágenes que aproveche la sinergia entre las características globales y locales.

1.4 Objetivos Específicos

1. Diseñar e implementar un método para la clasificación de texturas con color basado en características extraídas a nivel local y global en la imagen.
2. Evaluar si el uso de características de color y textura extraídas de características globales y locales permite mejorar el desempeño en clasificación de las texturas con color.
3. Implementar y evaluar el uso de un método de post procesamiento basado en *bagging* y un clasificador de máquinas de soporte vectorial para mejorar los resultados de clasificación de patrones de texturas con color.
4. Validar el método desarrollado comparando con las bases de datos de textura con color disponibles a nivel internacional.

1.5 Estructura del documento

La tesis está organizada de la siguiente manera. En el Capítulo 1 se da una breve introducción y la motivación de este trabajo, además de la hipótesis y los objetivos. En el Capítulo 2 es presentado el estado del arte del problema y las bases de datos utilizadas. La metodología diseñada para resolver el problema es explicada en el Capítulo 3. El Capítulo 4 presenta los experimentos realizados y resultados obtenidos en las cuatro bases de datos. Finalmente, el Capítulo 5, se discute las ventajas del método, presentadas como análisis y conclusiones.

Capítulo 2 Estado del arte

2.1 Extracción de características

Los monitores de video análogos muestran imágenes en color modulando la intensidad de los tres colores primarios (rojo, verde y azul) en cada píxel de la imagen. Habitualmente en una imagen digital a color, cada color primario es cuantizado con 8 bits de resolución. Así, cada color en el espacio RGB es representado como una combinación de los tres colores base: rojo (“*Red*” en inglés), verde (“*Green*” en inglés) y azul (“*Blue*” en inglés).

Generalmente la cantidad de información presente en las tres matrices es enorme para cualquier procesamiento que utilice una secuencia de imágenes en tiempo real. Por ejemplo, el utilizar imágenes de 1250 x 750 píxeles a 30 cuadros por segundo supone un manejo de 84.375.000 valores de 8 bits en cada segundo, lo que se traduce en un costo computacional muy alto. Para reducir dicha información, en vez de utilizar todos los píxeles presentes en una imagen, se utilizan valores que representen de manera eficiente dicha información. A estos valores se les llama características y se utilizan en el proceso de clasificación [3,7,16,31,50].

El proceso de extraer estas características por diferentes métodos será explicado en este capítulo. En una sección aparte se describe el funcionamiento de los clasificadores que se utilizan en base a las características extraídas y finalmente se detallan los diferentes métodos de selección de características.

Existen distintos métodos de extracción de características de color. El método de binarización utilizando preservación de momentos por cuaterniones (*BQMP*) [6] resume las características de color en un número determinado de representantes y su frecuencia de aparición, sin importar la textura de la imagen. Este método permite representar las imágenes obteniendo características para clasificación [16,51,52].

Asimismo, existen diversos métodos de extracción de características de textura. Las características definidas por Haralick [3], que incluyen estadísticos como correlación, entropía y segundo momento angular, entre otros, son capaces de describir una imagen usando la información presente en la variación relativa de la intensidad de los píxeles vecinos, independientemente de su color. Se ha realizado una selección de dichas características [53] utilizando matrices de co-ocurrencia, método que logra importantes mejoras en tiempo y detecciones. Ambos métodos (*BQMP* y Haralick) serán detallados y explicados a continuación, ya que se utilizan como base para el nuevo método que se propone en esta tesis.

2.1.1 Método de Binarización usando Preservación de Momentos por Cuaterniones (BQMP)

Una técnica de uso de umbrales llamada *Binary Quaternion Moment Preserving* (BQMP) fue desarrollada por Pei y Cheng [6]. Este método es utilizado como una herramienta para la extracción de características de color utilizando cuaterniones. Los cuaterniones son una expansión de los números complejos a cuatro dimensiones, y en este caso, cada píxel en el espacio RGB se representa como un cuaternión. Por medio de momentos de primer, segundo y tercer orden, es posible calcular dos cuaterniones llamados representantes a partir de una imagen dada. Cada representante simboliza una parte distinta de todos los píxeles de una imagen.

Utilizando los representantes se crea un hiper-plano que divide el espacio en dos, logrando una binarización de toda la imagen. Este método se puede realizar recursivamente n veces sobre cada fracción de la imagen binarizada, obteniéndose 2^n representantes de una imagen, resumiendo de esta manera la información presente en una imagen a un vector de largo dado de 5×2^n , donde los primeros 4 valores son el cuaternion representante y el quinto valor es el número de pixeles representados por dicho representante. Los fundamentos de esta descripción operativa se encuentran en el capítulo 2.1.1.2

2.1.1.1 Álgebra de Cuaterniones

Los cuaterniones son una expansión de los números complejos de dos dimensiones (real e imaginaria) a cuatro dimensiones (una real y tres imaginarias). Cada cuaternión se puede definir de la siguiente forma:

$$\hat{q} = q_0 + q_1 \cdot i + q_2 \cdot j + q_3 \cdot k, \quad (1)$$

donde i , j y k son las unidades que permiten extender el sistema de los números reales a cuaterniones. Estas unidades poseen las siguientes reglas:

$$i^2 = j^2 = k^2 = i \cdot j \cdot k = -1, \quad (2)$$

Tal como en los números complejos, el conjugado $\hat{q}^*$ de un cuaternión $\hat{q}$ se puede definir como:

$$\hat{q}^* = q_0 - q_1 i - q_2 j - q_3 k, \quad (3)$$

Finalmente, cada valor de color representado por el trío $\{R, G, B\}$ puede ser expresado como un cuaternión en donde $q_1 = R$, $q_2 = G$, $q_3 = B$ y $q_0 = 0$. Este hecho permite que sea posible utilizar los cuaterniones para el análisis de imágenes, aumentando de tres a cuatro su dimensión. Esto es mostrado en la siguiente figura.

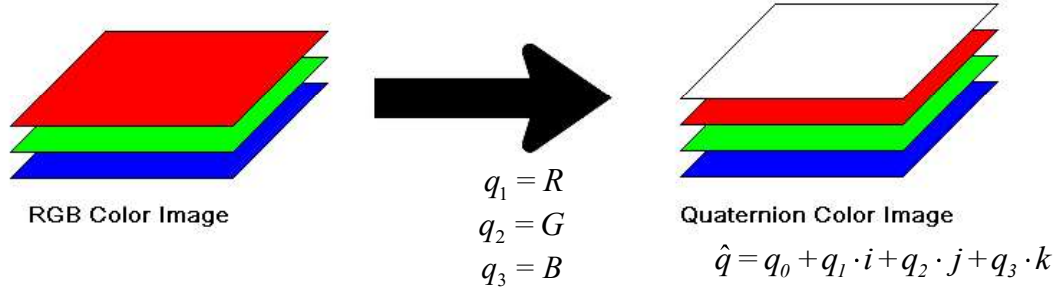


Figura 1: Ampliación del espacio RGB de 3 dimensiones al espacio de cuaterniones de 4 dimensiones. La dimensión extra posee solo valores cero en un comienzo.

Al realizar operaciones tales como multiplicaciones de cuaterniones, necesarias para realizar la binarización, los valores guardados en q_0 pueden dejar de ser cero. A modo de ejemplo, sean dos cuaterniones $a = a_1 + a_2i + a_3j + a_4k$ y $b = b_1 + b_2i + b_3j + b_4k$, su producto será igual a $ab = (a_1b_1 - a_2b_2 - a_3b_3 - a_4b_4) + (a_1b_2 + a_2b_1 + a_3b_4 - a_4b_3)i + (a_1b_3 - a_2b_4 + a_3b_1 + a_4b_2)j + (a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1)k$. Por lo tanto, incluso si a_1 y b_1 son iguales a cero, la parte real de la multiplicación no necesariamente lo será.

2.1.1.2 Momentos y Valores Representantes de Cuaterniones

Para lograr la binarización de la imagen, se definen los momentos de orden 1, 2 y 3, cuyas ecuaciones son las siguientes:

$$\hat{m}_1 = E(\hat{q}), \quad (4.a)$$

$$\hat{m}_2 = E(\hat{q} \cdot \hat{q}^*), \quad (4.b)$$

$$\hat{m}_3 = E(\hat{q} \cdot \hat{q}^* \cdot \hat{q}) \quad (4.c)$$

en donde el operador $E()$ representa la esperanza. Desarrollando las ecuaciones anteriores se obtiene que:

$$\hat{m}_1 = E(\hat{q}) = E(q_0) + E(q_1) \cdot i + E(q_2) \cdot j + E(q_3) \cdot k, \quad (5.a)$$

$$\hat{m}_2 = E(\hat{q} \cdot \hat{q}^*) = E(q_0^2 + q_1^2 + q_2^2 + q_3^2), \quad (5.b)$$

$$\hat{m}_3 = E(\hat{q} \cdot \hat{q}^* \cdot \hat{q}) = m_{30} + m_{31}i + m_{32}j + m_{33}k \quad (5.c)$$

donde

$$m_{30} = E(q_0^3 + q_0q_1^2 + q_0q_2^2 + q_0q_3^2),$$

$$m_{31} = E(q_0^2q_1 + q_1^3 + q_1q_2^2 + q_1q_3^2),$$

$$m_{32} = E(q_0^2q_2 + q_1^2q_2 + q_2^3 + q_2q_3^2) \quad y$$

$$m_{33} = E(q_0^2q_3 + q_1^2q_3 + q_2^2q_3 + q_3^3).$$

En la práctica, para calcular los momentos de un conjunto de datos Q , la esperanza $E(\cdot)$ es reemplazada por el valor promedio, ya que se puede suponer que la cantidad de datos es suficiente para hacer esa aproximación a un sistema ergódico, es decir el promedio de un espacio muestral es el mismo promedio entre muestras.

Se utilizan los momentos para obtener las ecuaciones de conservación de momento:

$$p_0 z_0 + p_1 z_1 = \hat{m}_1, \quad (6.a)$$

$$p_0 \hat{z}_0 \hat{z}_0^* + p_1 \hat{z}_1 \hat{z}_1^* = \hat{m}_2, \quad (6.b)$$

$$p_0 \hat{z}_0 \hat{z}_0^* \hat{z}_0 + p_1 \hat{z}_1 \hat{z}_1^* \hat{z}_1 = \hat{m}_3, \quad (6.c)$$

donde z_0 y z_1 son los representantes de dos porciones de imagen y p_0 y p_1 denotan las fracciones de los números que pertenecen a uno u otro grupo de representados. Es posible obtener los valores de los representantes utilizando el método de Prony [6], por medio de un polinomio auxiliar, cuyas raíces son los representantes.

Se selecciona un hiper-plano perpendicular al segmento $\overline{z_0 z_1}$ y equidistante a ambos valores representantes. Este separa el espacio en dos y cada mitad es representada por z_0 y z_1 respectivamente. Esto está explicado gráficamente en la Figura 2.

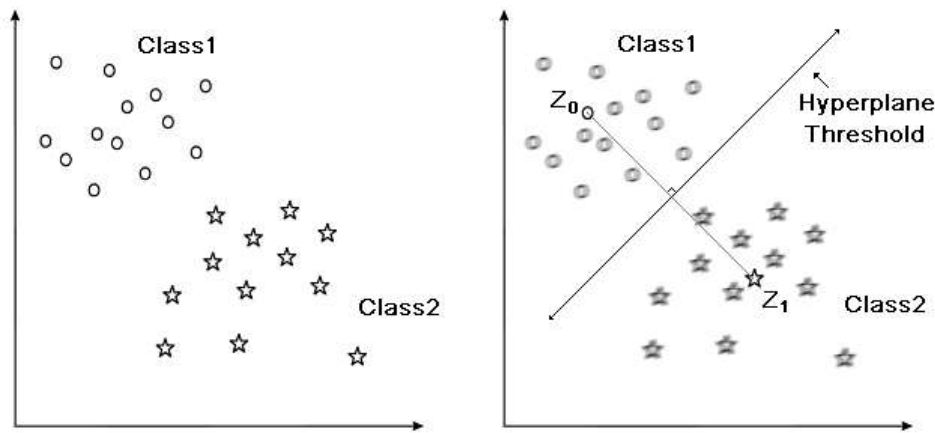


Figura 2: Separación del espacio en el método BQMP.

En el caso de querer más colores representantes de una imagen, el proceso se puede repetir recursivamente n veces, obteniendo 2^n representantes de la imagen y una estructura de árbol binario, agregando una rama por cada binarización. La Figura 3 muestra el método BQMP empleado una vez para el caso de 4 colores diferentes. Los números muestran el código de color (q_0, q_1, q_2, q_3) y el número de píxeles que cada representante simboliza (histograma). La Figura 4 muestra la segunda iteración para el mismo caso. El vector de características que se utiliza en la clasificación se forma concatenando las características de color e histogramas, ya normalizadas.

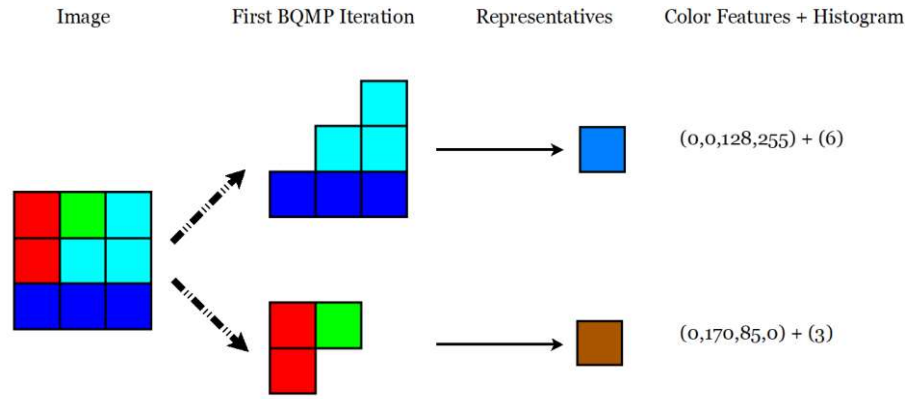


Figura 3: Ejemplo de la primera iteración del método BQMP en el caso de 4 colores. Los números muestran el color codificado en (q_0, q_1, q_2, q_3) y el número de píxeles representados por cada color (histograma).

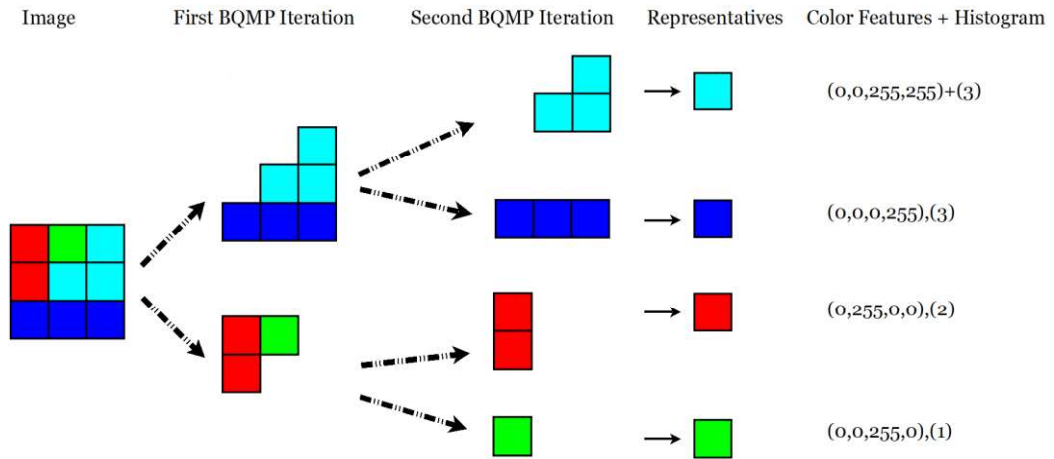


Figura 4: Ejemplo de la segunda iteración del método BQMP en el caso de 4 colores. Los números muestran el color codificado en (q_0, q_1, q_2, q_3) y el número de píxeles representados por cada color (histograma).

2.1.2 Estadísticos de Haralick

Las características de textura de Haralick son segundo momento angular, contraste, correlación, varianza, momento de diferencia inversa, suma de promedios, suma de varianzas, suma de entropías, entropía, varianza de diferencia, entropía de diferencia, medida de información de correlación y el coeficiente de correlación máxima [3]. Estas medidas caracterizan la complejidad y la naturaleza de las transiciones de tono en cada canal de la imagen. La práctica habitual es usar las primeras 13 características de Haralick con una matriz de coincidencia (*co-occurrence matrix*) [4], pero en este trabajo, fueron extraídas las 13 características de Haralick directamente de las imágenes pues proporcionan información espacial de diferentes regiones dentro de cada imagen. Las características de Haralick usadas para extraer las características de textura son las ecuaciones (7) a (19) y las ecuaciones (20) a (27) explican definiciones y notación utilizada.

Las características de textura utilizadas son las siguientes:

- Segundo Momento Angular:

$$f_1 = \sum_i \sum_j p(i,j)^2. \quad (7)$$

- Contraste:

$$f_2 = \sum_{n=0}^{N-1} n^2 \sum_{i=1}^N \sum_{j=1}^N p(i,j), \quad (8)$$

donde $|i - j| = n$.

- Correlación:

$$f_3 = \frac{\sum_i \sum_j (ij) p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}, \quad (9)$$

donde μ_x, μ_y, σ_x y σ_y son las medias y desviaciones estándar de p_x y p_y .

- Varianza:

$$f_4 = \sum_i \sum_j (i - \mu)^2 p(i,j). \quad (10)$$

- Momento de Diferencia Inversa:

$$f_5 = \sum_i \sum_j \frac{p(i,j)}{1 + (i - j)^2}. \quad (11)$$

- Suma de Promedios:

$$f_6 = \sum_{i=2}^{2N} i p_{x+y}(i). \quad (12)$$

- Suma de Varianzas:

$$f_7 = \sum_{i=2}^{2N} (i - f_6)^2 p_{x+y}(i). \quad (13)$$

- Suma de Entropías:

$$f_8 = - \sum_{i=2}^{2N} p_{x+y}(i) \log(p_{x+y}(i)). \quad (14)$$

- Entropía:

$$f_9 = - \sum_i \sum_j p(i, j) \log(p(i, j)). \quad (15)$$

- Varianza de Diferencia:

$$f_{10} = Var(p_{x-y}). \quad (16)$$

- Entropía de Diferencia:

$$f_{11} = - \sum_{i=0}^{N-1} p_{x-y}(i) \log(p_{x-y}(i)). \quad (17)$$

- Medida de Información de Correlación:

$$f_{12} = \frac{f_9 - HXY1}{\max(HX, HY)}, \quad (18)$$

$$f_{13} = 1 - \exp(-2(HXY2 - f_9))^{\frac{1}{2}}, \quad (19)$$

donde HX y HY son las entropías de p_x y p_y , y

$$HXY = - \sum_i \sum_j p(i, j) \log(p(i, j)), \quad (20)$$

$$HXY1 = - \sum_i \sum_j p(i, j) \log(p_x(i)p_y(j)), \text{ and} \quad (21)$$

$$HXY2 = - \sum_i \sum_j p_x(i)p_y(j) \log(p_x(i)p_y(j)). \quad (22)$$

En las ecuaciones (7) - (19), para calcular las características se utilizó la siguiente notación:

$$p(i, j), \quad (23)$$

Es el (i,j)-ésimo pixel de una matriz imagen en escala de grises.

$$p_x(i), \quad (24)$$

Es la i-ésima entrada en la matriz imagen, obtenida al sumar las filas de p.

$$p_y(j), \quad (25)$$

Es la j-esima entrada en la matriz imagen, obtenida al sumar las columnas de p.

$$P_{x+y}(k) = \sum_{i=1}^n \sum_{j=1}^n p(i, j), \quad (26)$$

donde $i + j = k$ y $k = 2, 3, \dots, 2N$.

$$P_{x-y}(k) = \sum_{i=1}^n \sum_{j=1}^n p(i, j), \quad (27)$$

donde $i - j = k$ y $k = 0, 1, \dots, N-1$.

El Coeficiente de Correlación Máxima (la característica 14) no es usado puesto que se necesitan calcular todos los vectores propios de una matriz, lo que es computacionalmente más pesado que el resto de las características.

2.2 Clasificadores

La clasificación es el problema de identificar a cuál categoría pertenece una nueva muestra, sobre la base de un conjunto de entrenamiento de datos cuya pertenencia a una categoría se conoce. Un algoritmo que implementa clasificación, especialmente en una aplicación concreta, es conocido como un clasificador [40].

En la terminología de aprendizaje computacional, la clasificación puede ser considerada una instancia de aprendizaje supervisado, es decir, el aprendizaje se da en un conjunto de entrenamiento cuyas observaciones correctamente identificadas están disponibles. Las observaciones son a menudo conocidas como casos, las variables explicativas se denominan características (agrupadas en un vector de características), y las posibles categorías son las clases [40].

Los clasificadores más utilizados en el área de textura y color son las máquinas de soporte vectorial [40–42], aunque también se utilizan redes neuronales [24,26] y k-vecinos más cercanos [9]. En este trabajo se utilizarán máquinas de soporte vectorial como clasificadores, pues ya se ha obtenido buenos resultados en trabajos previos [10,42,48,52].

2.2.1 Máquinas de Soporte Vectorial (SVM)

Las Máquinas de Soporte Vectorial (*SVM*) se han vuelto muy populares por su gran potencial de clasificación [41]. La *SVM* transforma los vectores de entrada a un espacio de mayor dimensión a través de una transformación no lineal. Dado un espacio de características con muestras de dos clases distintas, el objetivo del SVM es encontrar un plano de ecuación:

$$w^T x + b = 0 \quad (28)$$

que permita separarlas [40], como se observa en la Figura 5.

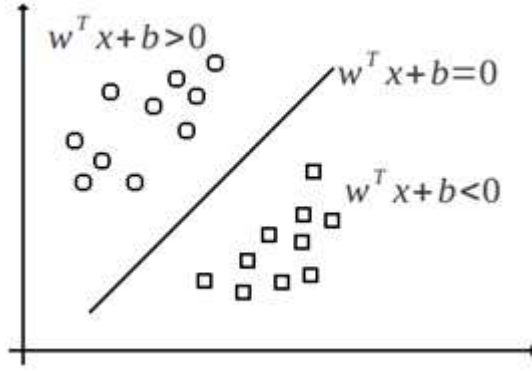


Figura 5: Plano separador de dos clases utilizando SVM.

El objetivo de SVM es que este plano se obtenga de tal forma que su distancia con cada una de las clases sea máxima. Para ello se definen los vectores de soporte (*Support Vectors*), que son aquellos que caracterizan el límite de la clase y que se obtienen de los ejemplos más cercanos al plano separador [40]. El problema entonces se transforma en maximizar la distancia o margen ρ entre los vectores de soporte de las distintas clases, como se muestra en la Figura 6.

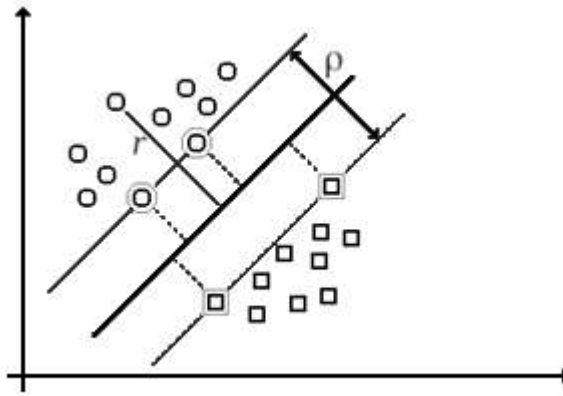


Figura 6: Ejemplo de un problema separable en dos dimensiones. Se denominan vectores de soporte a los puntos que conforman las dos líneas paralelas al hiperplano, siendo la distancia ρ entre ellas la mayor posible.

Si se define la notación $y_i = -1$ si está debajo del plano e $y_i = 1$ si está sobre el plano, las muestras x_i de las clases cumplen la siguiente ecuación:

$$y_i(w^T x_i + b) \geq 1, \quad (29)$$

En particular para los vectores de soporte x_s , la ecuación anterior es una igualdad. Como la distancia de un punto en el espacio, a un plano corresponde a la proyección, se puede expresar el margen como:

$$\rho = \frac{y_s(w^T x_s + b)}{\|w\|} = \frac{1}{\|w\|}, \quad (30)$$

por lo que maximizar el margen es equivalente al siguiente problema de optimización:

$$\underset{w,b}{Min} \frac{1}{2} \|w\|^2, \quad (31)$$

$$\text{sujeto a:} \quad y_i(w^T x_i + b) \geq 1, i = 1 \dots N.$$

Con N el número de muestras utilizadas para entrenar el clasificador. Este problema suele escribirse en su forma dual y resolverse mediante multiplicadores de Lagrange, con lo que finalmente se expresa de la siguiente forma:

$$\underset{\alpha_i}{Max} -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j x_i^T x_j + \sum_{i=1}^N \alpha_i, \quad (32)$$

$$\text{sujeto a:} \quad \sum_{i=1}^N y_i \alpha_i = 0 \\ \alpha_i \geq 0, i = 1 \dots N.$$

Donde los α_i corresponden a los multiplicadores de Lagrange. Resuelto el problema se obtiene el plano separador, con sus parámetros dados por:

$$w = \sum_{i=1}^N \alpha_i y_i x_i, \quad b = y_k - w^T x_k. \quad (33)$$

Para algún $\alpha_k > 0$ y x_k un vector de soporte. Obtenido lo anterior, la función de clasificación viene dada por:

$$y_i = \text{signo}(w^T x_i + b). \quad (34)$$

En el caso que las clases no se encuentren linealmente separadas en el espacio de características, la SVM permite introducir un parámetro C , correspondiente al peso o castigo de clasificar erróneamente las muestras. Así el problema de optimización consiste en un balance entre separar clases y minimizar errores, dado por la siguiente función:

$$\underset{w,b}{Min} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \varepsilon_i \quad (35)$$

sujeto a: $y_i(w^T x_i + b) \geq 1 - \varepsilon_i, \quad \varepsilon_i \geq 0, \quad i = 1 \dots N$, donde ε_i son las distancias de cada muestra al plano donde se encontrarían bien clasificadas. Mientras más grande es C , más importante es no cometer errores frente a encontrar un plano equidistante al momento de minimizar.

El mayor potencial de un clasificador SVM radica en su capacidad de aumentar la dimensionalidad del espacio de características para transformar un problema que no es linealmente separable, a uno que sí lo es [40]. El aumento de dimensionalidad se realiza considerando nuevas componentes que sean funciones de las características de la muestra.

Es decir, si x_i es una muestra, entonces $\varphi(x_i)$ es la muestra en el espacio aumentado a través de la función $\varphi(\cdot)$. El nuevo problema dual de optimización a resolver por el método de Lagrange y considerando la posibilidad de clasificación incorrecta, se escribe entonces:

$$\text{Max}_{\alpha_i} - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j \phi(x_i)^T \phi(x_j) + \sum_{i=1}^N \alpha_i \quad (36)$$

$$\text{sujeto a:} \quad \sum_{i=1}^N y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C, i = 1 \dots N.$$

Un problema es que la función que genera un espacio donde las muestras son linealmente separables puede no ser computable, es decir, no tiene presentación algorítmica, con lo que no se puede utilizar directamente la función $\varphi(x_i)$ [40]. Este problema se soluciona utilizando la función kernel, el que tiene la siguiente propiedad:

$$K(x, y) = \varphi(x)^T \varphi(y). \quad (37)$$

La función kernel de los vectores x e y es equivalente al producto de cada uno transformado a un espacio de mayor dimensión:

$$\text{Max}_{\alpha_i} - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j K(x_i, x_j) + \sum_{i=1}^N \alpha_i, \quad (38)$$

$$\text{sujeto a:} \quad \sum_{i=1}^N y_i \alpha_i = 0 \\ 0 \leq \alpha_i \leq C, i = 1 \dots N.$$

Algunos de los kernels más utilizados en SVM [40] son:

- Polinomial:

$$K(x, y) = (x \cdot y + 1)^d \quad (39)$$

- Gaussiano:

$$K(x, y) = \exp\left(-\frac{((x - y))^2}{2\sigma^2}\right) \quad (40)$$

- Tangente hiperbólica:

$$K(x, y) = \tanh(kx \cdot y + c) \quad (41)$$

En este trabajo se utilizará el kernel gaussiano, pues si el kernel utilizado es una función de base radial de Gauss, el espacio de características correspondiente es un espacio de Hilbert de dimensión infinita, donde puede ser aplicado el discriminante lineal [40]. En su implementación se variarán dos parámetros: C (relacionado con la función de costos, introducido en la ecuación 35) y σ (relacionado con el kernel) para determinar el mejor modelo.

2.2.2 *K* vecinos más cercanos (KNN)

Si bien el clasificador utilizado en este trabajo son Máquinas de Soporte Vectorial, a fin de compararnos de manera fidedigna, también fue implementado el algoritmo KNN de k vecinos más cercanos como clasificador. La idea tras el algoritmo es bastante sencilla: el algoritmo clasifica cada dato nuevo en el grupo que corresponda, según tenga un número k de vecinos definido anteriormente, más cerca de un grupo o de otro. Es decir, calcula la distancia del elemento nuevo a cada uno de los existentes, y ordena dichas distancias de menor a mayor para ir seleccionando el grupo al que pertenece. Un ejemplo puede verse en la Figura 7. Este método en general posee peores resultados que SVM [44].

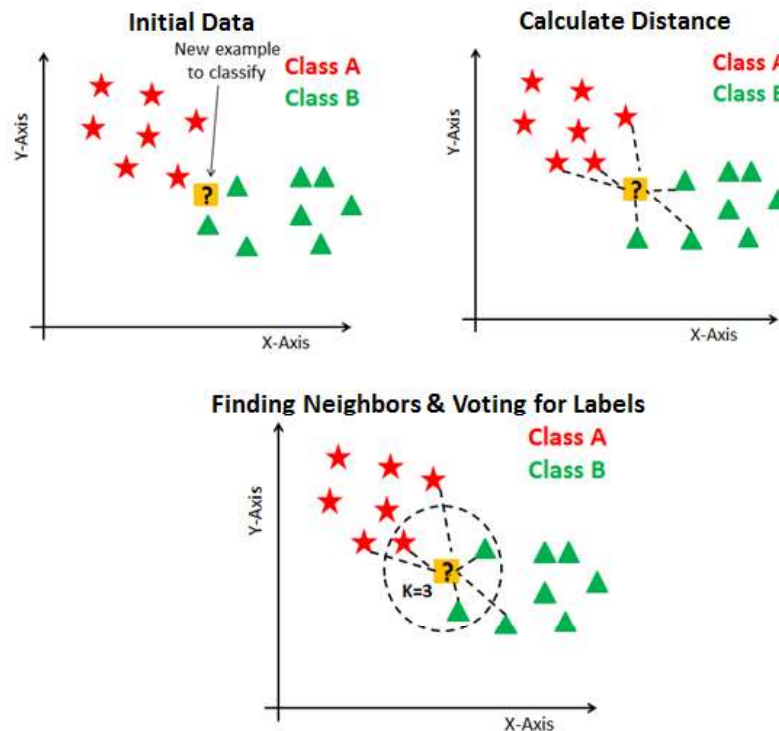


Figura 7: Pasos básicos de KNN.

KNN tiene tres pasos básicos: Calcular la distancia al nuevo ejemplo a clasificar, encontrar los k vecinos más cercanos y luego votar por las clases. El parámetro k es de vital importancia pues no se puede elegir ningún valor aleatorio para k e incluso pequeños cambios en k pueden dar lugar a grandes cambios.

2.3 Métodos de ensamblaje

En general, combinar distintos modelos de clasificación o en regresión, aumenta el desempeño de un predictor [54]. Esta idea de utilizar clasificadores múltiples o métodos de ensamblaje (*ensemble methods*) ha obtenido especial atención en la comunidad de inteligencia computacional y aprendizaje de máquinas durante las últimas décadas [45]. Estos métodos han probado ser efectivos y extremadamente versátiles en un espectro amplio de problemas y aplicaciones de mundo real. En la Figura 8 es posible ver el diagrama de bloques de un método de ensamblaje con K clasificadores débiles que son combinados para obtener un mejor resultado.

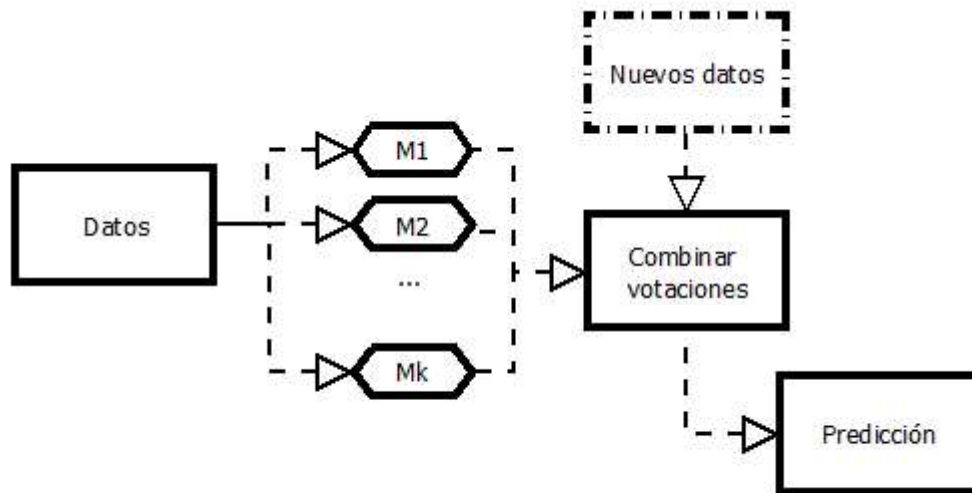


Figura 8: Diagrama de bloques de un método de ensamblaje.

Existen distintos métodos de ensamblaje tales como *bagging* o *bootstrap*. El método *Bagging (bootstrap aggregation)* [54] es una técnica que reduce la varianza y obtiene un mejor resultado en una predicción. Este método puede ser utilizado para clasificación o regresiones. El procedimiento básico de *bagging* está resumido en la siguiente Figura.

Input:

- D , a set of d training tuples;
- k , the number of models in the ensemble;
- a learning scheme (e.g., decision tree algorithm, backpropagation, etc.)

Output: A composite model, M^* .

Method:

- (1) **for** $i = 1$ to k **do** // create k models:
- (2) create bootstrap sample, D_i , by sampling D with replacement;
- (3) use D_i to derive a model, M_i ;
- (4) **endfor**

To use the composite model on a tuple, X :

- (1) **if** classification **then**
- (2) let each of the k models classify X and return the majority vote;
- (3) **if** prediction **then**
- (4) let each of the k models predict a value for X and return the average predicted value;

Figura 9: Pseudocódigo del método Bagging.

2.4 Bases de datos

Es importante validar el método en bases de datos internacionales de textura y color con resultados previamente publicados [55]. Con este propósito se escogen cuatro bases de datos que han sido utilizadas recientemente:

2. Colored Brodatz Texture (CBT) [20],
3. Vistex [4],
4. Outex [47], y
5. KTH-TIPS2b [48].

La base de datos Brodatz Colored Texture (CBT) posee 111 imágenes de un tamaño de 640 x 640 píxeles y cada imagen es una textura diferente. Ejemplos de esta base de datos pueden verse en la Figura 10.



Figura 10: 30 de las 111 texturas de color diferentes de la base Brodatz Colored Texture.

La base de datos VisTex (*Vision Texture*) fue desarrollada por el MIT. Posee 54 imágenes de un tamaño de 512 x 512 píxeles y cada imagen es una textura diferente. Ejemplos de esta base de datos pueden verse en la Figura 11.



Figura 11: 30 de las 54 texturas de color diferentes de la base Vistex.

La base de datos Outex fue desarrollada por la Universidad de Oulu, Finlandia. Posee 68 imágenes de textura en color de 746×538 píxeles. Ejemplos de esta base de datos pueden verse en la Figura 12.



Figura 12: 30 de las 68 texturas de color diferentes de la base Outex.

La base KTH-TIPS2b contiene 1188 imágenes de 200 x 200 píxeles. Esta base de datos posee 4 ejemplos físicos de 11 materiales expuestos a 4 iluminaciones distintas, 3 poses y 9 escalas distintas. Ejemplos de esta base de datos pueden verse en la Figura 13.



Figura 13: 22 de las 1188 texturas de color diferentes de la base KTH-TIPS2b.

Capítulo 3 Metodología propuesta

Se propone desarrollar un nuevo método de extracción de características que permita la detección y diferenciación de patrones complejos de color y textura. El método BQMP se usó previamente en la compresión de datos, la detección de bordes y la agrupación multiclase de datos, pero no en la clasificación [6,47]. El método BQMP reduce una imagen a colores representantes y crea un histograma que indica las partes de la imagen que están representadas por estos colores. Por lo tanto, las características de color de la imagen se representan en este histograma. Las características Haralick [3] a menudo se extraen para caracterizar texturas que miden la distribución en escala de grises, así como para considerar las interacciones espaciales entre píxeles [3,9,23,38]. La creación de un conjunto de entrenamiento es parte de la estrategia para obtener características locales y globales que contienen toda la información, local y global, para lograr una clasificación correcta. Diferentes clasificadores, tales como redes neuronales, k-vecinos más cercanos y máquinas de soporte vectorial [41], son usados para combinar características. En [44], SVM mostró un buen rendimiento en comparación con 16 métodos de clasificación diferentes.

3.1 Metodología

El método propuesto divide las imágenes para obtener características locales y globales. El método consta de cuatro etapas: en primer lugar, las imágenes de la base de datos se dividen en imágenes que se utilizarán en entrenamiento y aquellas que se utilizarán en las pruebas. En la segunda etapa, las características de color y textura se extraen desde las imágenes de entrenamiento en escalas globales y locales. En la tercera etapa, las características de color y textura se fusionan para convertirse en las entradas al clasificador SVM, y la última etapa es una etapa de post-procesamiento que utiliza *bagging* con las imágenes de prueba para la clasificación. Estas etapas se resumen en el diagrama de bloques de la Figura 14.

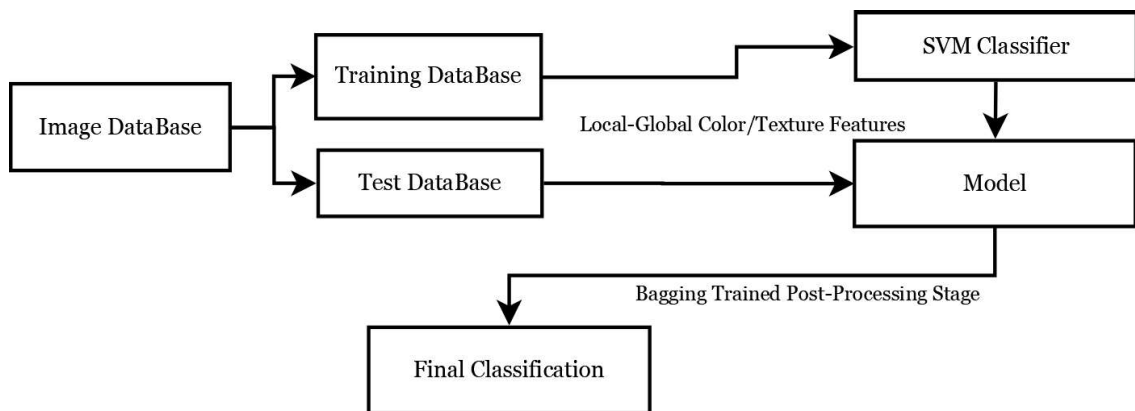


Figura 14: Diagrama de Bloques del método propuesto.

3.1.1 Extracción de Características en Cuadrantes

Para poder comparar el rendimiento de este método con los resultados publicados anteriormente, utilizamos la misma partición, en conjuntos de entrenamiento y prueba, en cada base de datos. En el caso de la base de datos Brodatz, como en la Referencia [1], cada imagen de textura coloreada se dividió en nueve sub-imágenes, utilizando una para entrenamiento y el resto para prueba. Un ejemplo de esta partición se muestra en las Figuras 15a – c. La base de datos Brodatz Coloured Texture (CBT) tiene 111 imágenes de 640×640 píxeles. Cada imagen en la base de datos tiene una textura diferente. En el caso de la base Vistex, el número de imágenes de entrenamiento y test son ocho, tal como en la referencia [4]. En el caso de la base Outex, el número de imágenes de entrenamiento y test son diez tal como en los trabajos de [4,9,47]. La base KTH-TIPS2b ya está particionada en cuatro ejemplos, y se realiza una validación cruzada tal como hizo antes [7,48].

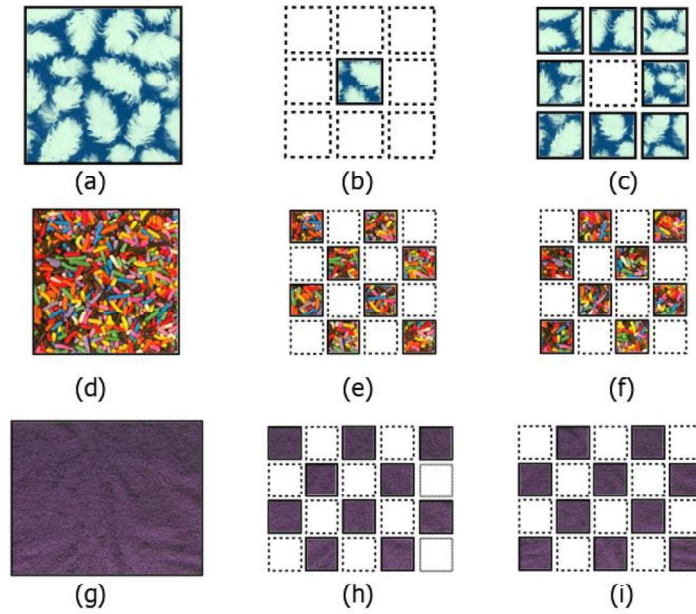


Figura 15: **(a)** Una imagen de la base Brodatz (D88) es usada para crear la imagen de entrenamiento **(b)** y las de prueba **(c)**. **(d)** Una imagen de la base Vistex (Food0007) es usada para crear las imágenes de entrenamiento **(e)** y las de prueba **(f)**. **(g)** Una imagen de la base Outex (Canvas002) es usada para crear las imágenes de entrenamiento **(h)** y las de prueba **(i)**.

En cada caso, se ha subdividido la base de entrenamiento y de prueba usando dos parámetros: n es el número de imágenes por lado, y r es el número de veces que tomamos n^2 imágenes locales de cada muestra. Es posible obtener $r \times n^2$ imágenes locales para extraer características de todas las muestras de la base de datos. La Figura 16 explica el esquema de subdivisión para las imágenes de entrenamiento. Se puede observar el esquema de particionado obtiene características provenientes de diferentes partes de la imagen, a niveles global y local. El método fue diseñado para seguir este enfoque y que no se pierda información relevante de la imagen.

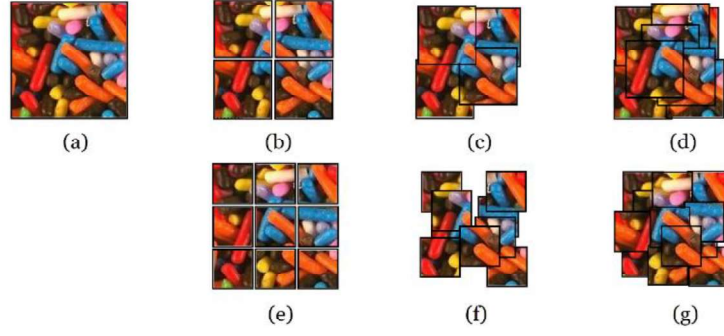


Figura 16: El proceso de subdivisión de imágenes en imágenes locales. (a) Una muestra de la base Vistex (Food0007) (b,e) La imagen es dividida en n^2 imágenes. (c,f) La imagen es dividida en n^2 imágenes aleatorias del mismo tamaño anterior. (d,g) La imagen es dividida en $r \times n^2$ imágenes aleatorias. En este ejemplo se ve la variación de parámetros n y r : en el caso de (b–d) poseen $n = 2$, (e–g) $n = 3$, y (d–g) $r = 2$.

BQMP y Haralick son invariantes ante traslación y rotación, es decir, se obtienen las mismas características si se realiza un intercambio de la posición de dos píxeles en la imagen [3,6,47]. Esto sugiere que hay información espacial presente en la imagen que estas características no extraen. En el método propuesto, se utiliza el esquema de dos escalas, local y global, para agregar información espacial a las características extraídas. Esto se muestra en la Figura 17. Las características BQMP y Haralick se extraen en cada cuadrante. Las imágenes de prueba se pueden subdividir en imágenes locales de las que se extraen las características. Esto permite la creación de una etapa de post-procesamiento en la que se puede realizar un proceso de *bagging*. Este método es invariante a la traslación debido a la aleatoriedad de las posiciones de la imagen local, pero es parcialmente invariante a la rotación porque las características se concatenan en un orden establecido. Sin embargo, las características de color, así como las características de textura de Haralick, son invariables para la rotación. Hay problemas donde la dependencia de la orientación es deseable [50].

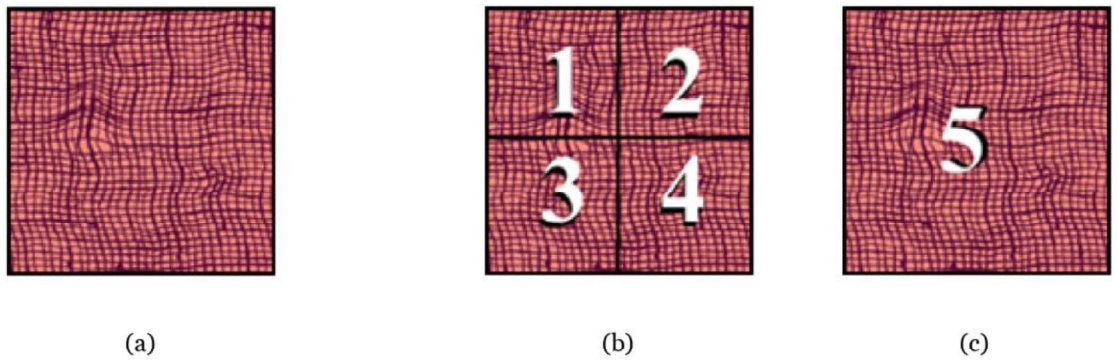


Figura 17: Ejemplo de (a) la imagen original, (b) las cuatro particiones locales y (c) la partición global. Estas cinco imágenes generan dos escalas espaciales: cuatro locales (b) y una global, agregando información espacial a las características extraídas.

La extracción de características se realiza utilizando escalas locales y globales. El vector de características se crea extrayendo características de cada partición de imagen diferente (local y global). Un ejemplo se muestra en la Figura 18. La imagen original (a) se divide en cinco particiones, (b) cuatro locales y una global (la misma imagen original), como se muestra en la Figura 17. El vector de características se obtiene de cada partición como se muestra en (c). Como BQMP se aplica una vez, obtenemos valores diferentes para dos colores representativos para cada sub-imagen. Como en el ejemplo que se muestra en la Figura 18c, el primer color representativo es marrón con $R1 = 78$, $G1 = 62$ y $B1 = 39$. El otro color es rosa con $R2 = 215$, $G2 = 80$, $B2 = 119$. Mediante binarización, sabemos que el marrón representa el 41% de la imagen y el rosa el 59%; por lo tanto, $H1 = 0.41$ y $H2 = 0.59$.

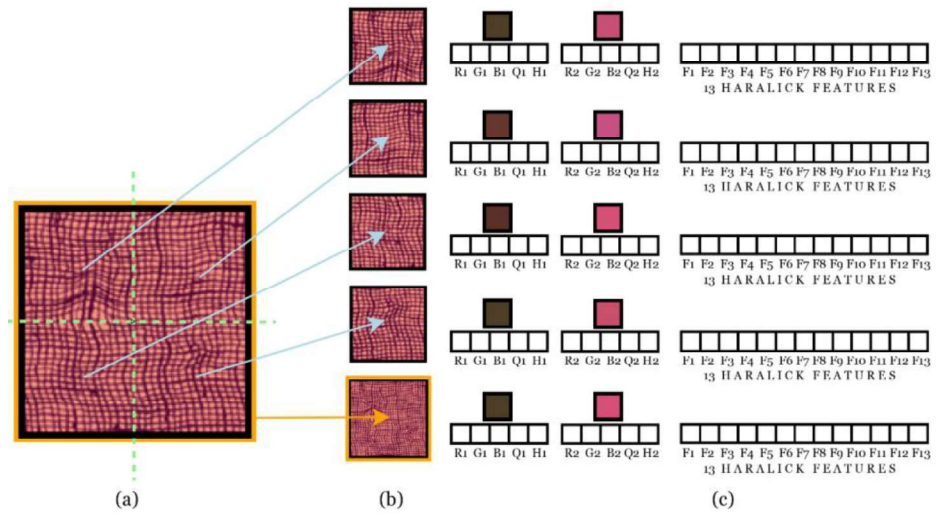


Figura 18: Vector de características extraído de una imagen de la base de datos Brodatz. (a) La imagen original se divide en cinco particiones (b): cuatro particiones locales y una global (la misma imagen, abajo). (c) Los vectores de características obtenidos de cada partición. El vector de características de color y el vector de características de textura se concatenan.

Para lograr la binarización, la información tridimensional RGB se transformó en un cuaternión cuatridimensional. Esos cuaterniones permiten obtener los momentos de orden 1, 2 y 3 que se usaron para obtener los colores representativos ($R1$, $G1$, $B1$ y $R2$, $G2$, $B2$) y los histogramas representantes ($H1$ y $H2$), como en la referencia [6] se describe. Los momentos se calcularon usando la multiplicación de cuaterniones, que es una operación de cuatro dimensiones. Por ejemplo, en el caso de dos cuaterniones a y b , donde $a = a_1 + a_2i + a_3j + a_4k$ y $b = b_1 + b_2i + b_3j + b_4k$, el producto ab será igual a:

$$ab = (a_1b_1 - a_2b_2 - a_3b_3 - a_4b_4) + (a_1b_2 + a_2b_1 + a_3b_4 - a_4b_3)i + (a_1b_3 - a_2b_4 + a_3b_1 + a_4b_2)j + (a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1)k.$$

Por lo tanto, incluso si a_1 y b_1 son iguales a cero, la parte real de la multiplicación no será necesariamente cero. En el caso del ejemplo, esta información adicional es $Q1 = -0.61$ en el primer color y $Q2 = 0.43$ en el segundo color.

Tras lo anterior, las 13 características de Haralick (descritas en la Sección 2.1.2) se extraen en cada sub-imagen y cada canal de color, obteniendo $13 \times 5 \times 3 = 195$ características más para concatenar en el vector final.

En la textura de la Figura 18, se realiza solo una iteración BQMP en una imagen de la base de datos Brodatz, obteniendo solo dos representantes de color. En general, el método BQMP genera 2^n representantes de cada imagen, cuando se usan n iteraciones. En las Figuras 3 y 4, se puede ver un ejemplo de un patrón de color simple muestra los representantes para dos iteraciones, $n = 2$. En los experimentos preliminares, los resultados no mejoraron significativamente para $n \geq 3$, y el tiempo computacional aumentó significativamente.

La extracción de características se realizó en escalas locales y globales, de modo que los colores representantes capturan la diversidad de toda la imagen de manera local y global. En el caso de texturas más complejas, es posible utilizar más iteraciones del método BQMP para obtener colores, histogramas y cuaterniones más representativos. La Figura 19 muestra la extracción de características de una imagen de muestra de la base de datos Vistex (Food0007) utilizando una, dos o tres iteraciones del método.

En ejemplo que se muestra en la Figura 19c, el primer color representativo es azul oscuro con $R1 = 30$, $G1 = 32$ y $B1 = 69$. El otro color es crema con $R2 = 217$, $G2 = 172$ y $B2 = 106$. A través de binarización, sabemos que el azul oscuro representa el 74% de la imagen y la crema el 26%; por lo tanto, $H1 = 0.74$ y $H2 = 0.26$. $Q1$ y $Q2$ son 1.49 y -2.43 , respectivamente.

Las características de Haralick calculadas a partir de las ecuaciones (1) - (13) para la primera sub-imagen son las siguientes:

$$\begin{array}{lll}
 F1 = 1.09 \times 10^{-4}, & F2 = 2.66 \times 10^3, & F3 = 9.90 \times 10^8, \\
 F4 = 4.75 \times 10^3, & F5 = 2.60 \times 10^{-2}, & F6 = 1.24 \times 10^2, \\
 F7 = 1.65 \times 10^4, & F8 = 5.28, & F9 = 9.32, \\
 F10 = 2.43 \times 10^{-5}, & F11 = 4.63, & F12 = -6.37 \times 10^{-2}, \\
 \text{y } F13 = 6.78 \times 10^{-1}.
 \end{array}$$

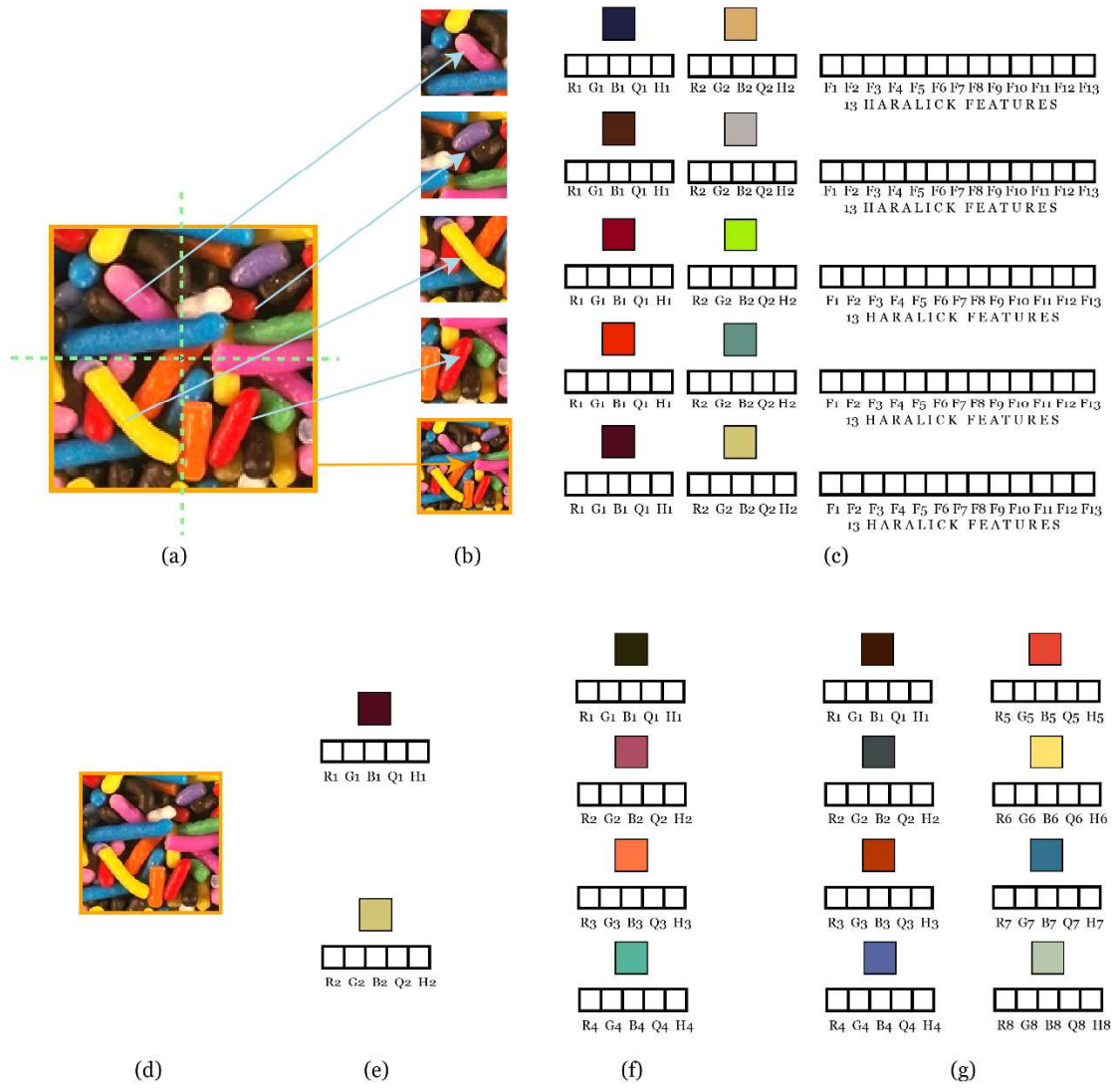


Figura 19: Vector de características extraído de una imagen de la base de datos Vistex (Food0007). (a) La imagen original se divide en cinco particiones (b): cuatro particiones locales y una global (la misma imagen, abajo). (c) Los vectores de características obtenidos de cada partición. Los vectores de características de color y los vectores de características de textura se concatenan. (d – g) La sub-imagen global como un ejemplo real utilizando una iteración (e), dos iteraciones (f) o tres iteraciones (g) del método BQMP.

3.1.2 Clasificación y Post-procesamiento

Después de extraer las características de cada imagen, se utilizó un clasificador SVM para determinar cada clase de textura. El SVM se hizo muy popular dentro de la comunidad de aprendizaje automático debido a su gran potencial de clasificación [41,42]. El SVM asigna vectores de entrada en una transformación no lineal a un espacio de alta dimensión donde se construye un hiperplano de decisión lineal para la separación de clases. Se usó un núcleo gaussiano SVM y se realizó una búsqueda exhaustiva general sobre los parámetros SVM restantes para encontrar la configuración óptima en el conjunto de entrenamiento.

Se utilizó una búsqueda de cuadrícula con validación cruzada para encontrar los mejores parámetros para la cascada SVM multiclase. Utilizamos la mitad del conjunto de entrenamiento para determinar los parámetros SVM y la otra mitad en la validación. Para las pruebas, utilizamos un conjunto diferente como se explica en la Sección 4. En el caso de *bagging*, se tomaron muestras repetidas del conjunto de entrenamiento original para equilibrar las distribuciones de clase para generar nuevos conjuntos de datos equilibrados. Se ajustaron dos parámetros: el número de árboles de decisión que votaron en el conjunto y el parámetro de complejidad relacionado con el tamaño del árbol de decisión. El método fue entrenado para la clasificación de texturas usando los conjuntos de entrenamiento tal como se especifican para cada base de datos.

En general, la combinación de modelos de clasificación múltiple aumenta el rendimiento predictivo [55]. En la etapa posterior al procesamiento, se realizó un modelo predictivo de embolsado compuesto por una combinación ponderada de clasificadores débiles con los resultados del modelo SVM [54].

Bagging es una técnica que utiliza el método de *bootstrap* para reducir la varianza y mejorar la precisión de un predictor [54]. Puede ser utilizado en clasificación y regresión. Se crea un conjunto de *bagging* para la clasificación utilizando árboles profundos (*Deep trees*) como clasificadores débiles (*weak learners*). El predictor de *bagging* fue entrenado con nuevas imágenes tomadas del conjunto de entrenamiento de cada base de datos. Este resultado se asigna como la clasificación final para cada imagen, como se puede ver en la Figura 20.

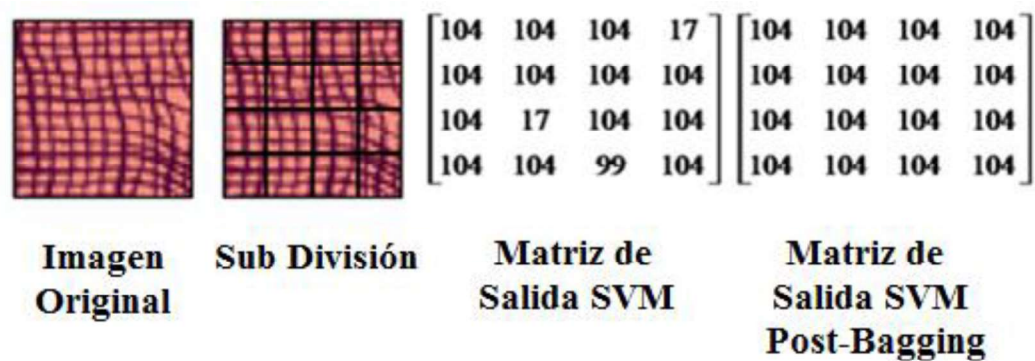


Figura 20: Ejemplo del resultado del post-procesamiento. Una imagen de prueba es subdividida en 16 imágenes. La SVM entrega un resultado por cada imagen, generándose una matriz de salida de SVM. El post-procesamiento entrega una clasificación final para cada imagen.

3.2 Experimentos

Para tener una comparación justa entre las tasas de clasificación obtenidas en este trabajo versus las publicadas anteriormente, utilizamos las mismas bases de datos y particiones utilizadas para el entrenamiento y pruebas que en Diaz-Pernas et al., 2011 [1], Khan et al., 2015 [7], Arvis et al., 2004 [9], Mäenpää et al., 2004 [27], Qazi et al., 2011 [28], Losson et al., 2013 [4], y Couto et al., 2017 [56]. Los conjuntos de entrenamiento y prueba provienen de sub-imágenes separadas, y los métodos nunca usaron la misma sub-imagen para el entrenamiento y las pruebas.

3.2.1 Base Brodatz

La metodología, como en [1], utiliza cuatro conjuntos diferentes de imágenes de la misma base de datos: la primera consta de 10 imágenes, la segunda de 30 imágenes, la tercera de 40 imágenes y la cuarta de todas las 111 imágenes. Los resultados de la clasificación en artículos publicados previamente alcanzaron el 91,03% en la Referencia [1], el 89,71% en la Referencia [43] y el 88,15% en la Referencia [44] para el cuarto conjunto de la base de datos Brodatz. Utilizamos la misma partición utilizada para el entrenamiento y las pruebas que en Diaz-Pernas et al., 2011 [1]. La base de datos de imágenes Brodatz consta de 111 imágenes de tamaño 640×640 píxeles. Particionando cada imagen en nueve sub-imágenes no superpuestas de 213×213 píxeles, se obtienen 999 sub-imágenes de 111 clases de textura. Díaz-Pernas et al., 2011 [1], utilizando la sub-imagen del centro (2×2) como entrenamiento y el resto para evaluar, alcanzaron los mejores resultados de clasificación (ver Figura 14).

En el entrenamiento, cada imagen con un tamaño de 213×213 píxeles se subdivide en un número n de imágenes. Las características se extraen de cada imagen subdividida. El parámetro n cambia de 2 a 7 en los tres primeros experimentos y de 2 a 9 en el último. Una vez que se ha calculado el vector de características, cada vector se asigna a una clase de textura utilizando un SVM como clasificador.

3.2.2 Base VisTex

La metodología utilizó 54 imágenes que se subdividieron en 864 sub-imágenes; 432 fueron utilizados en el entrenamiento y los otros 432 como imágenes de prueba, como en Arvis et al., 2004 [9], Mäenpää et al., 2004 [27], Qazi et al., 2011 [28], Losson et al., 2013 [4], y Couto et al., 2017 [56]. Los resultados publicados anteriormente alcanzaron 98,61% y 99,07% [4] en la misma base de datos usando una matriz de filtros de color (CFA) y matrices de ocurrencia cromática (CCM). Se escogieron las mismas 54 imágenes de textura para poder comparar los resultados obtenidos con los publicados anteriormente.

3.2.3 Base Outex

La metodología utilizó 68 imágenes que se subdividieron en 1360 sub-imágenes; 680 fueron utilizados en el entrenamiento y los otros 680 como imágenes de prueba, como en Arvis et al., 2004 [9]. La misma partición fue realizada por Mäenpää et al., 2004 [27], Qazi et al., 2011 [28], Losson et al., 2013 [4], y Couto et al., 2017 [56]. Los resultados publicados anteriormente alcanzaron 94,85% y 94,41% [4] en la misma base de datos usando una matriz de filtros de color (CFA) y matrices de ocurrencia cromática (CCM). Se escogieron las mismas 68 imágenes de textura para poder comparar los resultados obtenidos con los publicados anteriormente.

3.2.4 Base KTH-TIPS2b

La base de datos KTH-Tips2b consta de cuatro conjuntos de 1.188 imágenes. Cada conjunto tiene 11 clases diferentes. La metodología utilizó cuatro conjuntos de 108 imágenes, cada una con 11 imágenes, lo que hace un total de 1188 imágenes. Seguimos el mismo protocolo descrito en Khan et al., 2015 [7], donde el rendimiento de clasificación es el promedio de cuatro ejecuciones de prueba. En cada ejecución, todas las imágenes de una muestra se usaron para entrenamiento, mientras que todas las imágenes de las tres muestras restantes se usan como prueba, de la misma manera en que se hace en Khan et al., 2015 [7]. Los resultados publicados previamente alcanzan 70,6% [7] y 91,3% [8] en la misma base de datos usando los métodos agrupamiento teórico de información divisiva (*DITC*) y suma y diferencia adaptativa tridimensional (*3D-ASDH*), respectivamente.

3.2.5 Contribución color-textura y conjunto.

Tras la extracción de características detallada para cada base de datos, se realiza otro experimento en el que se verá el desempeño del método en tres escenarios: cuando solo se usan características de textura, cuando solo se usan características de color y cuando se utilizan ambos tipos de características, de manera de revisar si existe alguna sinergia al combinar características de color con características de textura.

Capítulo 4 Resultados

4.1 Base Brodatz

La Tabla 1 muestra los resultados de clasificación para la base de datos Brodatz utilizando el método desarrollado. En este experimento se utilizaron las características extraídas en el primer set (10 imágenes), variando el tamaño de las imágenes en el entrenamiento $r \times n^2$. La última columna muestra los resultados de usar la etapa de post-procesamiento aplicada en la columna con el mejor rendimiento. El mejor resultado para el primer experimento en la base de datos Brodatz con 10 imágenes es 100%. En este caso, 27 (3×3^2) imágenes con tamaño 71x71 (213/3) píxeles fueron utilizados para la formación. También se puede observar que el uso de la etapa de post-procesado mejoró los resultados hasta el 100% en todos los casos. Estos resultados son superiores al 98,23%, el mejor resultado publicado anteriormente para este experimento en la base de datos Brodatz [1].

Tabla 1: Resultados de la clasificación de los experimentos en la base de datos Brodatz para el conjunto de 10 imágenes. Los mejores resultados alcanzados con y sin post-procesamiento se resaltan con texto en negrita

	$r \times n^2$ imágenes aleatorias en entrenamiento				$r \times n^2$ imágenes aleatorias en entrenamiento y con post-procesamiento
	r=1	r=2	r=3	r=4	r=3
n=2	98,44%	99,68%	99,68%	99,68%	100%
n=3	99,58%	99,58%	100%	100%	100%
n=4	99,69%	99,84%	99,92%	98,84%	100%
n=5	99,65%	99,70%	99,8%	99,80%	100%
n=6	99,62%	99,65%	99,83%	99,79%	100%
n=7	99,46%	99,59%	99,51%	99,41%	100%

n es el número de imágenes por lado en la etapa de entrenamiento. r es las veces que el método itera al generar imágenes.

Las Tablas 2, 3 y 4 muestran los resultados de la clasificación para el segundo, tercer y cuarto experimentos en la base de datos Brodatz con 30, 40 y 111 imágenes, respectivamente. Como se muestra en la Tabla 2, el mejor resultado alcanzado con 30 imágenes fue de 99,84%. En este caso, 64 (4×4^2) imágenes aleatorias con tamaño 53x53 (213/4) píxeles fueron utilizados para el entrenamiento. También se puede observar que el uso de la etapa de post-procesado mejoró los resultados hasta el 100% en todos los casos. Estos resultados son superiores al mejor resultado, 97,54%, publicado anteriormente para este experimento en la base de datos Brodatz [1].

El mejor resultado alcanzado que utilizó la base de 40 imágenes fue 99,71%. En este caso, 100 (4×5^2) imágenes aleatorias con tamaño 42×42 (213/5) píxeles fueron utilizados para el entrenamiento. También se puede observar que el uso de la etapa de post-procesado mejoró los resultados hasta el 100% en todos los casos. Estos resultados son mejores que el 95,5%, el mejor resultado publicado para este experimento en la base Brodatz [1]. El mejor resultado alcanzado que utilizó la base completa de 111 imágenes fue de 97,63%. En este caso, 196 (4×7^2) imágenes aleatorias con tamaño 30×30 (213/7) píxeles fueron utilizados para el entrenamiento. También se puede observar que el uso de la etapa de post-procesado mejoró los resultados hasta 99,88%. Estos resultados son superiores a los resultados, 98,25% y 99,5% publicados previamente en este experimento usando la base de datos Brodatz [30,56].

Tabla 2: Resultados de la clasificación de los experimentos en la base de datos Brodatz para el conjunto de 30 imágenes. Los mejores resultados alcanzados con y sin post-procesamiento se resaltan con texto en negrita.

	r × n ² imágenes aleatorias en entrenamiento				r × n ² imágenes aleatorias en entrenamiento y con post-procesamiento
	r=1	r=2	r=3	r=4	r=4
n=2	88,02%	95,63%	96,97%	97,71%	100%
n=3	96,52%	98,75%	99,39%	99,21%	100%
n=4	98,72%	99,56%	99,81%	99,84%	100%
n=5	99,13%	99,58%	99,83%	99,73%	100%
n=6	99,22%	99,48%	99,55%	99,56%	100%
n=7	99,15%	99,35%	99,53%	99,60%	100%

n es el número de imágenes por lado en la etapa de entrenamiento. r es las veces que el método itera al generar imágenes.

Tabla 3: Resultados de la clasificación de los experimentos en la base de datos Brodatz para el conjunto de 40 imágenes. Los mejores resultados alcanzados con y sin post-procesamiento se resaltan con texto en negrita.

	r × n ² imágenes aleatorias en entrenamiento				r × n ² imágenes aleatorias en entrenamiento y con post-procesamiento
	r=1	r=2	r=3	r=4	r=4
n=2	88,05%	95,70%	96,09%	97,03%	100 %
n=3	96,81%	98,51%	99,27%	99,38%	100 %
n=4	98,39%	99,34%	99,55%	99,67%	100%
n=5	98,55%	99,56%	99,70%	99,71%	100 %
n=6	98,84%	99,47%	99,58%	99,70%	100 %
n=7	98,99%	99,46%	99,57%	99,61%	100 %

n es el número de imágenes por lado en la etapa de entrenamiento. r es las veces que el método itera al generar imágenes.

Tabla 4: Resultados de la clasificación de los experimentos en la base de datos Brodatz para el conjunto de 111 imágenes. Los mejores resultados alcanzados con y sin post-procesamiento se resaltan con texto en negrita.

	r × n ² imágenes aleatorias en entrenamiento				r × n ² imágenes aleatorias en entrenamiento y con post-procesamiento
	r=1	r=2	r=3	r=4	r=4
n=2	73,79%	82,46%	85,67%	87,56%	97,07%
n=3	87,48%	93,19%	94,65%	94,81%	99,32%
n=4	91,53%	95,10%	96,07%	96,24%	99,66%
n=5	93,82%	95,62%	96,43%	96,71%	99,88%
n=6	94,81%	96,45%	97,13%	97,30%	99,77%
n=7	95,45%	96,89%	97,54%	97,63%	99,77%
n=8	96,28%	92,58%	92,14%	90,05%	99,77%
n=9	84,82%	54,58%	54,83%	80,31%	96,85%

n es el número de imágenes por lado en la etapa de entrenamiento. r es las veces que el método itera al generar imágenes.

Se realiza una búsqueda exhaustiva que varió de n = 2 a n = 9. El tamaño de las imágenes más pequeñas alcanzó un óptimo para n = 7 con un tamaño de imagen de 30 × 30. La Tabla 5 compara los resultados publicados previamente en la literatura y los resultados obtenidos en la base de datos Brodatz para los cuatro experimentos que incluyeron 10, 30, 40 y 111 imágenes, respectivamente.

Es posible ver en la tabla 5 que el método desarrollado, con post-procesado, alcanza los mejores resultados. La mejora más significativa se alcanza en la base Brodatz completa que incluye las 111 imágenes.

Tabla 5: Los mejores resultados de la clasificación utilizando el método global local BQMP- Haralick para la base de datos Brodatz en los cuatro conjuntos de 10, 30, 40 y 111 imágenes en comparación con los estudios publicados anteriormente.

Mejores resultados vs. Número de Imágenes	10	30	40	111
Lazebnick et al., 2005 [43]	-	-	-	88,15%
Mellor et al., 2008 [44]	-	-	-	89,71%
Diaz-Pernas et al., 2011 [1]	98,23%	97,54%	95,5%	91,03%
Couto et al., 2017 [56]	-	-	-	98,25%
Kim et al., 2017 [31]	-	-	-	97,84%
SVM global-local Haralick BQMP ¹	100%	99,84%	99,71%	97,63%
KNN ² global-local Haralick BQMP + post-procesado ¹	100%	99,94%	99,98%	94,46%
SVM global-local Haralick BQMP + post-procesado ¹	100%	100%	100%	99,88%

¹ Los últimos tres métodos son los resultados de esta tesis; ² *k-nearest neighbors*.

4.2 Base VisTex

La Tabla 6 muestra los resultados de clasificación del método desarrollado, aplicado a la base de datos Vistex (54 imágenes). Cada imagen en el conjunto de entrenamiento se dividió al azar, y el número de ventanas por lado varió de dos a cuatro en cada imagen elegida para el entrenamiento, con el número de imágenes aleatorias de $2 \times n^2$ a $10 \times n^2$. La primera columna muestra los mejores resultados alcanzados por el método desarrollado, y la segunda columna muestra los resultados después de la etapa de post-procesado.

La Tabla 7 compara los resultados publicados previamente en la literatura y los resultados obtenidos para la base de datos Vistex con 54 imágenes. Se puede observar en la Tabla 7 que el método con post-procesado alcanzó los resultados más altos con 100%.

Tabla 6: Resultados de clasificación del experimento en la base de datos Vistex para el conjunto de 54 imágenes. Los mejores resultados obtenidos con y sin post-procesamiento se destacan con texto en negrita.

$r \times n^2$ imágenes aleatorias ¹ en entrenamiento						
	Sin post-procesado			Con post-procesado		
	r = 2	r = 3	r = 4	r = 2	r = 3	r = 4
n = 2	92,42%	91,95%	92,18%	99,54%	99,77%	99,54%
n = 3	94,33%	94,34%	93,47%	99,54%	100%	99,77%
n = 4	95,52%	94,83%	94,13%	99,54%	100%	100%
n = 5	95,68%	95,29%	94,64%	99,54%	100%	100%
n = 6	96,13%	95,88%	95,22%	100%	100%	100%
n = 7	96,45%	96,26%	95,36%	100%	100%	100%
n = 8	96,72%	96,42%	96,42%	100%	100%	100%
n = 9	95,67%	97,13%	95,88%	100%	100%	100%
n = 10	96,96%	96,32%	95,62%	100%	100%	100%

¹ n es el número de imágenes por lado en entrenamiento. r es las veces que el método itera al generar imágenes.

Tabla 7: Mejores resultados de la clasificación global local Haralick-BQMP para la base de datos Vistex con 54 imágenes y mejores resultados publicados previamente en la misma base de datos.

Paper	Método	Resultado
Arvis et al., 2004 [9]	<i>Multispectral</i>	97,9%
Mäenpää et al., 2004 [27]	<i>Color histogram $I_1I_2I_3I_3I_3I_3$</i>	100%
Qazi et al., 2011 [28]	<i>Improved hue, luminance, and saturation color space (IHLS) B = 16</i>	100%
Losson et al., 2013 [4]	<i>Color filter array - chromatic co-occurrence matrices</i>	98,61%
Losson et al., 2013 [4]	<i>Chromatic co-occurrence matrices</i>	99,07%
Couto et al., 2017 [56]	<i>Deterministic walks' direction histogram</i>	99,65%
Cernadas et al., 2017 [30]	<i>Parallel vectors</i>	99,5%
Neiva et al., 2018 [23]	<i>Smoothed morphological operators (SMO)</i>	99,54%
Kalakech et al., 2018 [38]	<i>Adapted Laplace score</i>	94,9%
SVM Global-local Haralick BQMP ¹		97,13%
KNN Global-local Haralick BQMP + post-procesado ¹		97,25%
SVM Global-local Haralick BQMP + post-procesado ¹		100%

¹ Los últimos tres métodos son los resultados de esta tesis.

4.3 Base Outex

La Tabla 8 muestra los resultados de clasificación del método aplicado a la base de datos Outex (68 imágenes). Cada imagen en el conjunto de entrenamiento se dividió aleatoriamente, variando el número de ventanas por lado de dos a cuatro en cada imagen de entrenamiento, y el número de imágenes aleatorias de $2 \times n^2$ a $18 \times n^2$. Las primeras tres columnas muestran los mejores resultados alcanzados por el método desarrollado, y las segundas tres columnas muestra los resultados después de la etapa de post-procesamiento. Los mejores resultados se resaltan con texto en negrita.

Tabla 8: Resultados de clasificación del experimento en la base de datos Outex.

$r \times n^2$ imágenes aleatorias ¹ en entrenamiento						
Sin post-procesado			Con post-procesado			
	r = 2	r = 3	r = 4	r = 2	r = 3	r = 4
n = 2	84,83%	83,74%	82,63%	97,05%	97,06%	96,76%
n = 3	86,83%	84,64%	84,13%	97,64%	97,20%	97,64%
n = 4	87,44%	85,54%	84,69%	98,52%	97,35%	97,35%
n = 5	87,43%	86,46%	85,40%	97,50%	97,79%	97,79%
n = 6	88,49%	86,72%	85,72%	97,50%	98,08%	98,08%
n = 7	88,84%	86,37%	86,15%	97,94%	97,50%	98,08%
n = 8	89,19%	87,44%	86,27%	98,23%	97,79%	98,38%
n = 9	89,38%	87,73%	86,51%	98,67%	97,79%	98,23%
n = 10	89,61%	87,96%	86,62%	97,94%	98,23%	97,94%
n = 11	90,12%	88,20%	86,85%	97,94%	98,23%	97,94%
n = 12	90,35%	88,37%	87,04%	98,97%	97,94%	98,52%
n = 13	90,52%	88,50%	87,22%	98,82%	98,38%	97,05%
n = 14	90,58%	88,21%	87,38%	98,67%	97,20%	97,20%
n = 15	90,78%	88,06%	86,53%	98,67%	97,79%	96,32%
n = 16	90,19%	86,92%	86,24%	97,94%	96,62%	96,03%
n = 17	89,98%	86,02%	85,10%	97,50%	96,32%	95,44%
n = 18	90,18%	85,54%	84,16%	97,64%	95,59%	94,85%

¹ n es el número de imágenes por lado en entrenamiento. r es las veces que el método itera al generar imágenes.

La Tabla 9 compara los resultados publicados previamente en la literatura y los resultados para la base de datos Outex con 68 imágenes. Se puede observar que el método con post-procesado alcanzó los resultados más altos con 98,97%.

Tabla 9: Mejores resultados de la clasificación Global Local Haralick-BQMP para la base de datos Outex y mejores resultados publicados previamente en la misma base de datos.

Paper	Método	Resultado
Arvis et al., 2004 [9]	<i>Multispectral</i>	94,9%
Mäenpää et al., 2004 [27]	<i>Color histogram HSV16³</i>	95,4%
Bianconi et al., 2011 [11]	<i>Gabor and chromatic features</i>	90,0%
Qazi et al., 2011 [28]	<i>IHLS color space B = 16</i>	94,5%
Losson et al., 2013 [4]	<i>Color filter array - chromatic co-occurrence matrices</i>	94,41%
Losson et al., 2013 [4]	<i>Chromatic co-occurrence matrices</i>	94,85%
Sandid et al., 2016 [8]	<i>3D adaptive sum and difference histograms</i>	95,8%
Couto et al., 2017 [56]	<i>Deterministic walks direction histogram</i>	97,28%
Neiva et al., 2018 [23]	<i>Smoothed morphological operators (SMO)</i>	86,47%
Cernadas et al., 2017 [30]	<i>Parallel vectors</i>	90,6%
SVM Global-local Haralick BQMP ¹		90,78%
KNN global-local Haralick BQMP con post-procesado ¹		96,72%
SVM Global-local Haralick BQMP con post-procesado ¹		98,97%

¹ Los últimos tres métodos son los resultados de esta tesis.

4.4 Base KTH-TIPS2b

La Tabla 10 muestra los resultados de clasificación del método aplicado a la base de datos KTH-TIPS2b (1188 × 4 imágenes). En cada prueba, todas las imágenes de una muestra se usaron para entrenamiento, mientras que las imágenes de las tres muestras restantes se usaron como un conjunto de prueba. Las primeras dos columnas muestran los mejores resultados alcanzados por el método desarrollado, y las segundas dos columnas muestran los resultados después de la etapa de post-procesamiento.

Tabla 10: Resultados de clasificación en la base de datos KTH-TIPS2b para los 4 conjuntos de 1188 imágenes. Los mejores resultados obtenidos en promedio con y sin post-procesamiento se destacan con texto en negrita.

	r × n² imágenes aleatorias ¹ en entrenamiento			
	Sin post-procesado		Con post-procesado	
	n = 1	n = 2	n = 1	n = 2
S = 1	98,73%	99,16%	99,73%	99,65%
S = 2	92,72%	91,26%	95,79%	95,59%
S = 3	90,74%	88,99%	94,53%	93,71%
S = 4	89,42%	87,88%	92,93%	92,89%
Promedio	92,90%	91,82%	95,75%	95,46%

¹ n es el número de imágenes por lado en la etapa de entrenamiento;

S es el conjunto utilizado para el entrenamiento, utilizando los otros tres conjuntos como prueba.

La Tabla 11 compara los resultados publicados previamente en la literatura y los resultados para la base de datos KTH-TIPS2b con 1188 × 4 imágenes. Se puede observar en la Tabla 11 que el método con post-procesamiento alcanzó los resultados más altos con 95,75%.

Tabla 11: Mejores resultados de la clasificación Global-Local Haralick-BQMP para la base de datos KTH-TIPS2b y mejores resultados publicados previamente.

Paper	Método	Resultado
Khan et al., 2015 [7]	<i>Divisive information theoretic clustering</i>	70,6%
Sandid et al., 2016 [8]	<i>3D-ASDH</i>	91,3%
El Merabet et al., 2018 [34]	<i>Local concave/convex micro-structure pattern</i>	84,44%
El Merabet et al., 2019 [35]	<i>Attractive-and-Repulsive Center-Symmetric LBP</i>	93,61%
Global-local Haralick BQMP ¹		92,90%
KNN global-local Haralick BQMP + post-procesado ¹		92,72%
Global-local Haralick BQMP + post-procesado ¹		95,75%

¹ Los últimos tres métodos son los resultados de esta tesis.

4.5 Contribución color-textura y conjunto

La Tabla 12 compara los resultados usando solo las características de color, solo características de textura y la combinación de ambas, para la clasificación de textura medida en las bases de datos Brodatz, Vistex, Outex y KTH-TIPS2b. Se puede observar que ambos tipos de características, color y textura contribuyen a los resultados generales, con el máximo rendimiento cuando se combinan ambos tipos de características. Al comparar estos resultados con los publicados previamente en las mismas bases de datos, se puede observar que, aunque el método alcanzó el 100% en la base de datos Vistex en la Referencia [28], los resultados obtenidos en Outex fueron solo del 94,5%.

Tabla 12: Mejores resultados de la clasificación utilizando el método desarrollado en todas las bases de datos con la contribución de cada parte del modelo (color y textura).

Base de Datos	Global-Local Haralick BQMP (Todas las características)	Global-Local Haralick BQMP (solo color)	Global-Local Haralick BQMP (solo textura)
Brodatz	99,88%	96,28%	87,84%
Outex	98,97%	91,86%	88,57%
Vistex	100%	95,67%	90,84%
KTH-TIPS2b	95,75%	94,95%	92,59%

4.6 Tiempo de procesamiento

La Tabla 13 muestra el tiempo de cálculo requerido para la extracción de características (FE), el tiempo de clasificación con SVM y el tiempo de post-procesamiento (PP) realizado en la base de datos Vistex. Todas las implementaciones se llevaron a cabo utilizando Python 3 en un Intel (R) Core (TM) i7-7700HM 3.6 GHz, con 64 GB de memoria de acceso aleatorio (RAM).

Tabla 13: Tiempo computacional del método propuesto para extracción de características (FE), clasificación usando SVM, tiempo de postprocesado (PP) y tiempo total. Los experimentos se realizaron en la base de datos Vistex (54 imágenes).

Vistex	Tiempo FE (s)	Tiempo SVM (s)	Tiempo PP (s)	Tiempo total(s)
Entrenamiento	57,05 s	305,19 s	44,96 s	102,86 s
Prueba	55,88 s	41,22 s	5,76 s	407,2 s
Total	112,93 s	346,41 s	50,72 s	510,06 s

4.7 Comparación con técnicas de *Deep Learning*

En contraste con la corriente actual de resolver los problemas de clasificación utilizando técnicas de *Deep Learning* el método propuesto en esta tesis se basa en métodos convencionales de aprendizaje de máquinas (ahora denominado *Shallow Learning*). *Deep Learning* utiliza una red neuronal artificial compuesta por capas de distinto nivel jerárquico que tienen diferentes aprendizajes del más simple al más complejo. En general las características generadas por métodos basados en *Deep Learning* imitan la percepción humana a través de varias operaciones tales como convoluciones y agrupaciones, con ello logran mejores descriptores [25,26,57–60]. Los resultados alcanzados por técnicas de *Deep Learning* en clasificación de patrones general son muy buenos, pero requieren de bases de datos de entrenamiento de gran tamaño [57–59].

A modo de ejemplo, se citan algunas redes neuronales convolucionales (*CNNs*) que son ampliamente conocidas [57–59], que utilizan grandes bases de datos de entrenamiento, y que participaron de las distintas tareas (clasificación, localización y detección de objetos) de los desafíos de la competencia *Large Scale Visual Recognition Challenge (ILSVRC)* entre los años 2012 y 2014. Estas redes fueron entrenadas con Imagenet, una base de datos compuesta por 1,2 millones de imágenes con más de mil clases.

En el caso de *AlexNet* [57] fue utilizada en la tarea de clasificación de imágenes de la competencia *ILSVRC-2012* cuyo modelo logra una tasa de error de 15,3% sobre mil clases y con ello ganando la competencia en sus tareas de clasificación y localización. Otra red es la VGG [58], donde se aumenta la profundidad de la red y se utiliza filtros de convolución pequeños de 3x3. Con ello logra mejoras sobre la base de datos utilizada en [57] con las que obtuvieron el primer y segundo lugar de la competencia de Clasificación y localización de *ILSVRC-2014*, si se ordena por error de localización. Un tercer ejemplo es el caso de GoogLeNet [59] que incluye filtros de distintos tamaños operando al mismo nivel (*inception modules*) los que requieren menos memoria y complejidad computacional, haciéndolo más popular que el resto de los métodos. Esta red obtuvo el primer lugar en la competencia *ILSVRC-2014* tanto en clasificación y detección, si se ordena por error de clasificación.

En [60–65] se presentan varios métodos basados en *Deep Learning* que utilizan las mismas bases de datos que en esta tesis en clasificación de texturas (Brodatz, Outex, Vistex o KTH-TIPS2b). En estos trabajos se utiliza el mismo protocolo de evaluación al utilizado en esta tesis, por lo que es posible comparar los resultados. Lo anterior se resume en la Tabla 14, en donde distintas *CNNs* son utilizadas para resolver problemas de textura y color en las mismas bases de datos utilizadas.

Una posible explicación de estos resultados se presenta en un análisis teórico de la dimensión intrínseca del conjunto de datos de textura que es mayor de lo que puede ser procesado por la red de aprendizaje profundo [26]. Según su hipótesis, las texturas no pueden clasificarse sin características de textura explícitas por redes neuronales profundas debido a su alta dimensionalidad intrínseca. Su trabajo proporciona la validación de que los conjuntos de datos de detección de objetos tienen una menor dimensionalidad intrínseca en comparación con los conjuntos de datos de textura.

Otros trabajos del mismo autor muestran que la clasificación de texturas usando características basadas en texturas se realiza utilizando píxeles sin procesar. También mostraron que las redes más profundas como AlexNet tuvieron el peor desempeño en todos los conjuntos de datos, lo que indica que las redes analizadas tienden a sobreajustarse [62]. Aunque esto también puede deberse a que las bases de datos de textura que se utilizaron en el protocolo no poseen un tamaño suficiente para generar bases de entrenamiento suficientemente pobladas para el tamaño de la red neuronal profunda. Esto es un tema en desarrollo que aún no está definido por completo.

Tabla 14: Comparación de métodos basados en *Deep Learning* vs mejores resultados de la clasificación en todas las bases de datos.

Método (Paper)	Base de Datos (clases)	Resultado
<i>High Dimensional Deep Learning Two Layer Codebook Features.</i> Liu et al., 2017 [60]	<i>Brodatz (111)</i>	92,13%
<i>Deep Neural Network.</i> Gunasekara et al., 2017 [61]	<i>Brodatz (10)</i>	98,70 ± 0,97%
<i>Relational Layers CNN.</i> Loke et al., 2018 [62]	<i>Brodatz (111)</i>	99,00%
<i>Modified CNN + Whale optimization algorithm.</i> Dixit et al., 2019 [63]	<i>Brodatz (111)</i>	97,43%
<i>High Dimensional Deep Learning Two Layer Codebook Features.</i> Liu et al., 2017 [60]	<i>Outex (68)</i>	97,09%
<i>Modified CNN + Whale optimization algorithm.</i> Dixit et al., 2019 [63]	<i>Outex (68)</i>	97,70%
<i>High Dimensional Deep Learning Two Layer Codebook Features.</i> Liu et al., 2017 [60]	<i>Vistex (54)</i>	88,09%
<i>Fisher Vector CNN</i> Cimpoi et al., 2015 [64]	<i>KTH-TIPS2b (11)</i>	81,5%± 2,0%
<i>Neural Network Feature Transformation NFT</i> Song et al., 2016 [65]	<i>KTH-TIPS2b (11)</i>	83,3%± 1,4%
Global-local Haralick BQMP + post-procesado ¹	<i>Brodatz (111)</i>	99,88%
Global-local Haralick BQMP + post-procesado ¹	<i>Outex (68)</i>	98,97%
Global-local Haralick BQMP + post-procesado ¹	<i>Vistex (54)</i>	100%
Global-local Haralick BQMP + post-procesado ¹	<i>KTH-TIPS2b (11)</i>	95,75%

¹ Los últimos cuatro métodos son los resultados de esta tesis.

Capítulo 5 Discusión y Conclusiones

5.1 Discusión

La idea de combinar color y textura se ha propuesto anteriormente, pero el proceso de extracción de características propuesto permite al método preservar la información disponible en la imagen original, obteniendo resultados significativamente mejores que los publicados en la literatura. Un posible inconveniente de los métodos de clasificación de textura anteriores es que se pierde información importante de la imagen original con el método de extracción de características, lo que dificulta su capacidad para mejorar los resultados de clasificación de textura. El proceso de extracción de características que incluye características globales y locales lo que es algo nuevo desde el punto de vista de combinar color con textura. Subdividir la partición de entrenamiento de la base de datos en sub-imágenes e intentar obtener toda la información presente en la imagen usando varios tamaños de imagen o un número diferente de imágenes es algo que no se reportó en publicaciones anteriores.

Las características de color y textura se extraen para clasificar texturas de colores complejos. Sin embargo, el proceso de extracción de características pierde parte de la información presente en la imagen porque la información bidimensional (2D) se transforma en un espacio reducido. Mediante el uso de características globales y locales extraídas de muchas particiones diferentes de la imagen, la información necesaria para la clasificación de texturas coloreadas se conserva mejor. Subdividir los datos de entrenamiento en sub-imágenes (local-global) e intentar obtener toda la información presente utilizando diferentes tamaños de imagen es un nuevo enfoque.

Aunque el método BQMP se propuso hace varios años [6], se usó en la compresión de datos de color, la detección de bordes de color y la agrupación de datos de color en varias clases. La reducción de una imagen en colores representativos y un histograma que indica qué parte de la imagen está representada por estos colores, logra excelentes resultados. Además, las características locales y globales se extraen de cada imagen. Los resultados del método desarrollado se compararon con los de varias otras implementaciones de extracción de características en la base de datos Brodatz con las publicadas en Referencias [1,27,28,31,43,44,56], en la base de datos Vistex con las publicadas en Referencias [4,9,13,23,27,28,30,38,45,46,56], en la base de datos de Outex con los publicados en Referencias [4,8,9,11,27,28,30,38,45,46,56], y en el KTH-TIPS2b con los publicados en Referencias [7,8,31,34,35] (véanse las tablas 2, 4, 6 y 8). El método propuesto generó mejores resultados que los publicados previamente.

Las bases de datos Brodatz, Vistex, Outex y KTH2b-Tips están disponibles para comparar los resultados de diferentes métodos de clasificación de textura. Las pruebas deben realizarse en las mismas condiciones. Se comparan los resultados obtenidos con los de las referencias [1,7] en las mismas condiciones utilizando la misma distribución de entrenamiento / prueba y un SVM como clasificador.

También se compara los resultados obtenidos con los de la Referencia [4] en la que utilizaron un clasificador vecino más cercano (KNN) y, por lo tanto, se implementa el método con KNN en lugar de SVM. Los resultados con KNN se muestran en las Tablas 2, 4, 6 y 8, corroborando que SVM logra mejores resultados. El método propuesto logró mejores resultados que los publicados anteriormente.

5.2 Conclusiones

En este trabajo, es presentado un nuevo método para clasificar patrones complejos de texturas coloreadas. Este método propuesto incluye cuatro pasos principales. En primer lugar, la imagen se divide en imágenes locales y globales. Esta subdivisión de imagen permite la extracción de características en diferentes escalas espaciales, además de agregar información espacial a las características extraídas. Por lo tanto, capturamos características globales y locales de la textura. En segundo lugar, las características de textura y color se extraen de cada imagen dividida utilizando los algoritmos BQMP y Haralick. En tercer lugar, se utiliza una máquina de soporte vectorial para clasificar cada imagen con las características extraídas como entradas. En cuarto lugar, se emplea una etapa de procesamiento posterior utilizando *bagging*.

El método se probó en cuatro bases de datos, *Brodatz*, *VisTex*, *Outex* y *KTH-TIPS2B*, con tasas de clasificación correctas de 97,63%, 97,13%, 90,78% y 92,90% respectivamente. La etapa de post-procesamiento mejoró los resultados a 99,88%, 100%, 98,97% y 95,75%, respectivamente, para las mismas bases de datos. Comparamos los resultados en las mismas bases de datos versus los mejores resultados publicados previamente y encontramos mejoras significativas de 8,85%, 0,93% (a 100%), 4,12% y 4,45%. Finalmente se comparó con trabajos nuevos basados en *Deep Learning*, cuyos resultados no superan los mostrados en esta tesis.

La partición de las bases de datos en imágenes locales y globales entrega información sobre características a diferentes escalas y ubicaciones espaciales dentro de cada imagen lo que es útil en la clasificación utilizando color y textura. Lo anterior y el uso de la etapa de post-procesamiento utilizando un modelo predictivo de *bagging* permite obtener estas mejoras.

Esta investigación entrega una nueva manera de atacar el problema y evidencia sobre la importancia del uso de distintas escalas al obtener las características, a fin de integrar la información de color y textura. Los resultados de esta investigación derivaron en dos publicaciones: un artículo de conferencia [52] y un artículo de revista [49].

5.3 Trabajo Futuro

Como continuación a este trabajo de tesis, existen diversas líneas de investigación que quedan abiertas en las que es posible seguir trabajando. Entre los posibles trabajos futuros se destacan:

1. La justificación para utilizar SVM se basa en los buenos resultados obtenidos previamente. Sin embargo, las referencias que se entregan para apoyar esta afirmación poseen más de 5 años de antigüedad. Actualmente existen diversos enfoques de *gradient boosting machines*, tales como *XGBoost* [66] o *LightGBM* [67] que lideran problemas de clasificación. Sería de interés probar esas técnicas en este problema.
2. Sería interesante tener un ranking de las características de manera de observar cuales son las que contribuyen en la clasificación y cuáles características se pueden descartar. En la sección 4.5 hay una primera aproximación, pues es posible observar cual es el aporte de las características de color vs las de texturas vs ambas en conjunto. Este análisis se puede realizar utilizando técnicas de selección de características basadas en información mutua, como redundancia mínima y relevancia máxima (mRMR) [68], selección de características de información mutua normalizada (NMIFS) [69], selección de características de información mutua condicional (CMIFS) [70], maximización de información mutua condicional (CMIM) [71] o métodos *leave one out* [72]. Actualmente los métodos de clasificación modernos como *XGBoost* [66] también permiten hacer este análisis basándose en que tan relevantes son las características dentro de bosques aleatorios (*Random Forest*).
3. Dado que *Deep Learning* obtiene buenos resultados y el método desarrollado en esta tesis también, quizás es posible mejorar los resultados proponiendo un enfoque mixto, que incluya características que contienen información complementaria a las características de textura-color o con el uso de particiones locales-globales, por lo que sería interesante explorar un método híbrido combinando ambos enfoques. La gran dificultad de realizar este método mixto es que los métodos basados en *Deep Learning* deben ser entrenados con muchos ejemplos y las bases de datos contienen un número de imágenes por clases que no lo permite. Una manera es realizar aumento de datos (*data augmentation*) incluyendo imágenes con rotaciones, traslaciones y cambios de escala. De no ser esto suficiente, es posible realizar *Fine-tuning* [73,74]: utilizar los pesos de redes convolucionales ya entrenadas para otros problemas. Esto es posible puesto que los modelos logran hacer la transferencia del aprendizaje [73,75] dada la diversidad de las bases de datos.

Se agradece a los miembros de la comisión cuyos comentarios inspiraron todas estas posibles líneas de investigación de manera de expandir este trabajo.

BIBLIOGRAFÍA

1. Díaz-Pernas, F.J.; Antón-Rodríguez, M.; Perozo-Rondón, F.J.; González-Ortega, D. A multi-scale supervised orientational invariant neural architecture for natural texture classification. *Neurocomputing* **2011**, *74*, 3729–3740, doi:10.1016/j.neucom.2011.06.028.
2. Drimbarean, A.; Whelan, P.F. Experiments in colour texture analysis. *Pattern Recognition Letters* **2001**, *22*, 1161–1167.
3. Haralick, R.M.; Shanmugam, K.S.; Dinstein, I. Textural Features for Image Classification. *IEEE Trans. Systems, Man, and Cybernetics* **1973**, *3*, 610–621, doi:10.1109/TSMC.1973.4309314.
4. Losson, O.; Porebski, A.; Vandenbroucke, N.; Macaire, L. Color texture analysis using CFA chromatic co-occurrence matrices. *Computer Vision and Image Understanding* **2013**, *117*, 747–763, doi:10.1016/j.cviu.2013.03.001.
5. Ojala, T.; Pietikainen, M.; Harwood, D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In Proceedings of the 12th International Conference on Pattern Recognition; 1994; Vol. 1, pp. 582–585.
6. Pei, S.C.; Cheng, C.M. Color image processing by using binary quaternion-moment-preserving thresholding technique. *IEEE Trans. on Image Process.* **1999**, *8*, 614–628, doi:10.1109/83.760310.
7. Khan, F.S.; Anwer, R.M.; Weijer, J. van de; Felsberg, M.; Laaksonen, J. Compact color-texture description for texture classification. *Pattern Recognition Letters* **2015**, *51*, 16–22, doi:10.1016/j.patrec.2014.07.020.
8. Sandid, F.; Douik, A. Robust color texture descriptor for material recognition. *Pattern Recognition Letters* **2016**, *80*, 15–23, doi:10.1016/j.patrec.2016.05.010.
9. Arvis, V.; Debain, C.; Berducat, M.; Benassi, A. Generalization of the cooccurrence matrix for colour images: Application to colour texture classification. *Image Analysis & Stereology* **2011**, *23*, 63–72, doi:10.5566/ias.v23.p63-72.
10. Bianconi, F.; González, E.; Fernández, A.; Saetta, S.A. Automatic classification of granite tiles through colour and texture features. *Expert Systems with Applications* **2012**, *39*, 11212–11218, doi:10.1016/j.eswa.2012.03.052.
11. Bianconi, F.; Harvey, R.W.; Southam, P.; Fernández, A. Theoretical and experimental comparison of different approaches for color texture classification. *Journal of Electronic Imaging* **2011**, *20*, 043006.
12. Cament Riveros, L.; Galdames Grunberg, F.; Bowyer, K.; Pérez, C.A. Face recognition under pose variation with local Gabor features enhanced by Active Shape and Statistical Models. *Pattern Recognition* **2015**, *48*, 3371–3384, doi:DOI: 10.1016/j.patcog.2015.05.017.
13. Hiremath, P.S.; Shivashankar, S.; Pujari, J. Wavelet based features for color texture classification with application to cbir. In Proceedings of the International Journal of Computer Science and Network Security; 2006; Vol. 6, pp. 124–133.
14. Perez, C.A.; Aravena, C.M.; Vallejos, J.I.; Estevez, P.A.; Held, C.M. Face and iris localization using templates designed by particle swarm optimization. *Pattern Recognition Letters* **2010**, *31*, 857–868, doi:10.1016/j.patrec.2009.12.029.
15. Perez, C.A.; Cament, L.A.; Castillo, L.E. Methodological improvement on local Gabor face recognition based on feature selection and enhanced Borda count. *Pattern Recognition* **2011**, *44*, 951–963, doi:10.1016/j.patcog.2010.10.017.

16. Perez, C.A.; Saravia, J.A.; Navarro, C.F.; Schulz, D.A.; Aravena, C.M.; Galdames, F.J. Rock lithological classification using multi-scale Gabor features from sub-images, and voting with rock contour information. *International Journal of Mineral Processing* **2015**, *144*, 56–64, doi:DOI: 10.1016/j.minpro.2015.09.015.
17. Perez, C.A.; Estévez, P.A.; Vera, P.A.; Castillo, L.E.; Aravena, C.M.; Schulz, D.A.; Medina, L.E. Ore grade estimation by feature selection and voting using boundary detection in digital image analysis. *International Journal of Mineral Processing* **2011**, *101*, 28–36, doi:10.1016/j.minpro.2011.07.008.
18. Nan, B.; Mu, Z.; Chen, L.; Cheng, J. A Local Texture-Based Superpixel Feature Coding for Saliency Detection Combined with Global Saliency. *Applied Sciences* **2015**, *5*, 1528–1546, doi:10.3390/app5041528.
19. Zeng, D.; Zhu, M.; Zhou, T.; Xu, F.; Yang, H. Robust Background Subtraction via the Local Similarity Statistical Descriptor. *Applied Sciences* **2017**, *7*, 989, doi:10.3390/app7100989.
20. Abdelmounaime, S.; Dong-Chen, H. New Brodatz-Based Image Databases for Grayscale Color and Multiband Texture Analysis. *ISRN Machine Vision* **2013**, *2013*, 1–14, doi:10.1155/2013/876386.
21. Ilea, D.E.; Whelan, P.F. Image segmentation based on the integration of colour–texture descriptors—A review. *Pattern Recognition* **2011**, *44*, 2479–2501, doi:10.1016/j.patcog.2011.03.005.
22. Wang, Y.; Yang, J.; Peng, N. Unsupervised color–texture segmentation based on soft criterion with adaptive mean-shift clustering. *Pattern Recognition Letters* **2006**, *27*, 386–392, doi:10.1016/j.patrec.2005.09.014.
23. Barros Neiva, M.; Vacavant, A.; Bruno, O.M. Improving texture extraction and classification using smoothed morphological operators. *Digital Signal Processing* **2018**, *83*, 24–34.
24. Andrearczyk, V.; Whelan, P.F. Using filter banks in convolutional neural networks for texture classification. *Pattern Recognition Letters* **2016**, *84*, 63–69.
25. Zhu, Q.; Zhong, Y.; Liu, Y.; Zhang, L.; Li, D. A Deep-Local-Global Feature Fusion Framework for High Spatial Resolution Imagery Scene Classification. *Remote Sensing* **2018**, *10*, 568, doi:10.3390/rs10040568.
26. Basu, S.; Mukhopadhyay, S.; Karki, M.; DiBiano, R.; Ganguly, S.; Nemani, R.; Gayaka, S. Deep neural networks for texture classification—A theoretical analysis. *Neural Networks* **2018**, *97*, 173–182.
27. Mäenpää, T.; Pietikäinen, M. Classification with color and texture: jointly or separately? *Pattern Recognition* **2004**, *37*, 1629–1640, doi:10.1016/j.patcog.2003.11.011.
28. Qazi, I.-U.-H.; Alata, O.; Burie, J.-C.; Moussa, A.; Fernandez-Maloigne, C. Choice of a pertinent color space for color texture characterization using parametric spectral analysis. *Pattern Recognition* **2011**, *44*, 16–31, doi:10.1016/j.patcog.2010.07.007.
29. Shi, L.; Funt, B. Quaternion color texture segmentation. *Computer Vision and Image Understanding* **2007**, *107*, 88–96, doi:10.1016/j.cviu.2006.11.014.
30. Cernadas, E.; Fernández-Delgado, M.; González-Rufino, E.; Carrión, P. Influence of normalization and color space to color texture classification. *Pattern Recognition* **2017**, *61*, 120–138, doi:10.1016/j.patcog.2016.07.002.
31. Kim, N.C.; So, H.J. Directional statistical Gabor features for texture classification. *Pattern Recognition Letters* **2018**, *112*, 18–26, doi:10.1016/j.patrec.2018.05.010.
32. Wang, M.; Gao, L.; Huang, X.; Jiang, Y.; Gao, X. A Texture Classification Approach Based on the Integrated Optimization for Parameters and Features of Gabor Filter via Hybrid Ant Lion Optimizer. *Applied Sciences* **2019**, *9*, 2173, doi:10.3390/app9112173.

33. Backes, A.R.; de Mesquita Sá Junior, J.J. LBP maps for improving fractal based texture classification. *Neurocomputing* **2017**, *266*, 1–7, doi:10.1016/j.neucom.2017.05.020.
34. El merabet, Y.; Ruichek, Y. Local Concave-and-Convex Micro-Structure Patterns for texture classification. *Pattern Recognition* **2018**, *76*, 303–322, doi:10.1016/j.patcog.2017.11.005.
35. El merabet, Y.; Ruichek, Y.; El idrissi, A. Attractive-and-repulsive center-symmetric local binary patterns for texture classification. *Engineering Applications of Artificial Intelligence* **2019**, *78*, 158–172, doi:10.1016/j.engappai.2018.11.011.
36. Hiremath, P.S.; Bhusnurmath, R.A. Multiresolution LDBP descriptors for texture classification using anisotropic diffusion with an application to wood texture analysis. *Pattern Recognition Letters* **2017**, *89*, 8–17, doi:10.1016/j.patrec.2017.01.015.
37. Jeena Jacob, I.; Srinivasagan, K.G.; Jayapriya, K. Local Oppugnant Color Texture Pattern for image retrieval system. *Pattern Recognition Letters* **2014**, *42*, 72–78, doi:10.1016/j.patrec.2014.01.017.
38. Kalakech, M.; Porebski, A.; Vandenbroucke, N.; Hamad, D. Unsupervised Local Binary Pattern Histogram Selection Scores for Color Texture Classification. *Journal of Imaging* **2018**, *4*, 112, doi:10.3390/jimaging4100112.
39. Shivashankar, S.; Kudari, M.; Hiremath, P.S. Galois Field-based Approach for Rotation and Scale Invariant Texture Classification. *International Journal of Image, Graphics and Signal Processing (IJIGSP)* **2018**, *10*, 56–64, doi:10.5815/ijigsp.2018.09.07.
40. Cortes, C.; Vapnik, V. Support-Vector Networks. *Machine Learning* **1995**, *20*, 273–297, doi:10.1023/A:1022627411411.
41. Meyer, D.; Leisch, F.; Hornik, K. The support vector machine under test. *Neurocomputing* **2003**, *55*, 169–186, doi:10.1016/S0925-2312(03)00431-4.
42. Shang, C.; Barnes, D. Support vector machine-based classification of rock texture images aided by efficient feature selection. In Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN); 2012; pp. 1–8.
43. Lazebnik, S.; Schmid, C.; Ponce, J. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2005**, *27*, 1265–1278.
44. Mellor, M.; Hong, B.; Brady, M. Locally Rotation, Contrast, and Scale Invariant Descriptors for Texture Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2008**, *30*, 52–61, doi:10.1109/TPAMI.2007.1161.
45. Hayati, S.; Ahmadzadeh, M.R. WIRIF: Wave interference-based rotation invariant feature for texture description. *Signal Processing* **2018**, *151*, 160–171, doi:10.1016/j.sigpro.2018.05.001.
46. Pham, M.-T.; Mercier, G.; Bombrun, L. Color Texture Image Retrieval Based on Local Extrema Features and Riemannian Distance. *Journal of Imaging* **2017**, *3*, 43, doi:10.3390/jimaging3040043.
47. Ojala, T.; Maenpaa, T.; Pietikainen, M.; Viertola, J.; Kyllonen, J.; Huovinen, S. Outex-new framework for empirical evaluation of texture analysis algorithms. In Proceedings of the Object recognition supported by user interaction for service robots; IEEE, 2002; Vol. 1, pp. 701–706.
48. Caputo, B.; Hayman, E.; Mallikarjuna, P. Class-specific material categorisation. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1; IEEE, 2005; Vol. 2, pp. 1597–1604.
49. Navarro, C.F.; Perez, C.A. Color–Texture Pattern Classification Using Global–Local Feature Extraction, an SVM Classifier, with Bagging Ensemble Post-Processing. *Applied Sciences* **2019**, *9*, 3130, doi:10.3390/app9153130.

50. Kashyap, R.L.; Khotanzad, A. A Model-based Method for Rotation Invariant Texture Classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *8*, 472–481, doi:10.1109/TPAMI.1986.4767811.
51. Zhang, C.; Ma, Y. *Ensemble machine learning: methods and applications*; Springer, 2012; ISBN 1-4419-9325-8.
52. Perez, C.A.; Navarro, C.F.; Schulz, D.A.; Saravia, J.A.; Aravena, C.M. Pattern Detection Using a New Haralick Quaternion Color Extraction Model and Support Vector Machine Classifier. *2013 IEEE International Conference on Systems, Man, and Cybernetics* **2013**, 3300–3304, doi:10.1109/SMC.2013.562.
53. Palm, C. Color texture classification by integrative Co-occurrence matrices. *Pattern Recognition* **2004**, *37*, 965–976, doi:10.1016/j.patcog.2003.09.010.
54. Breiman, L. Bagging predictors. *Mach Learn* **1996**, *24*, 123–140, doi:10.1007/BF00058655.
55. Hossain, S.; Serikawa, S. Texture Databases - A Comprehensive Survey. *Pattern Recogn. Lett.* **2013**, *34*, 2007–2022, doi:10.1016/j.patrec.2013.02.009.
56. Couto, L.N.; Backes, A.R.; Barcelos, C.A. Texture characterization via deterministic walks' direction histogram applied to a complex network-based image transformation. *Pattern Recognition Letters* **2017**, *97*, 77–83, doi:10.1016/j.patrec.2017.07.013.
57. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in neural information processing systems; 2012; pp. 1097–1105.
58. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.
59. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition; 2015; pp. 1–9.
60. Liu, P.; Guo, J.-M.; Wu, C.-Y.; Cai, D. Fusion of deep learning and compressed domain features for content-based image retrieval. *IEEE Transactions on Image Processing* **2017**, *26*, 5706–5717.
61. Gunasekara, P.; Wijayakulasooriya, J.V.; Dharmagunawardhana, H.A.C. Image texture analysis using deep neural networks. In Proceedings of the 2017 IEEE International Conference on Industrial and Information Systems (ICIIS); IEEE, 2017; pp. 1–5.
62. Loke, K.S. Texture recognition using a novel input layer for deep convolutional neural network. In Proceedings of the 2018 IEEE 3rd International Conference on Communication and Information Systems (ICCIS); IEEE, 2018; pp. 14–17.
63. Dixit, U.; Mishra, A.; Shukla, A.; Tiwari, R. Texture classification using convolutional neural network optimized with whale optimization algorithm. *SN Applied Sciences* **2019**, *1*, 655.
64. Cimpoi, M.; Maji, S.; Vedaldi, A. Deep filter banks for texture recognition and segmentation. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition; 2015; pp. 3828–3836.
65. Song, Y.; Li, Q.; Feng, D.; Zou, J.J.; Cai, W. Texture image classification with discriminative neural networks. *Computational Visual Media* **2016**, *2*, 367–377.
66. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining; 2016; pp. 785–794.
67. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. In Proceedings of the Advances in neural information processing systems; 2017; pp. 3146–3154.