



DESCRIPCIÓN BREVE

El parámetro estadístico Entropía de Permutación (EP) determina la complejidad en series temporales, basándose en la comparación de sus valores vecinos. Presenta grandes ventajas frente a otros parámetros: rapidez, robustez, simplicidad de cálculo e invarianza con respecto a transformaciones no lineales. Este trabajo, estudia diferentes técnicas para implementar la EP en el reconocimiento de texturas en imágenes.

Adriana Pilar Antonelli

Directores:

Dra. Ing. Virginia Ballarin

Dr. Ing. Gustavo Meschino

ENTROPÍA DE PERMUTACIÓN PARA LA CARACTERIZACIÓN DE TEXTURAS EN IMÁGENES DIGITALES

Proyecto Final - Ingeniería Electrónica





RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-
NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Contenido

1. Introducción.....	3
1.1 Presentación del proyecto.....	3
1.2 Objetivos.....	3
1.3 Estructura del Proyecto.....	4
2. Estado del Arte.....	5
2.1 Métodos tradicionales de análisis de texturas en imágenes.....	7
2.1.1 Métodos estadísticos.....	7
2.1.2 Métodos estructurales.....	11
2.1.3 Métodos basados en tratamiento de la señal.....	12
2.1.4 Métodos basados en modelos.....	16
2.1.5 Comparación entre métodos.....	18
2.2 Entropía de Permutación para series temporales.....	19
2.2.1 Conceptos previos.....	19
2.2.2 Método de cálculo y características.....	21
2.2.3 Ejemplo de cálculo de EP unidimensional.....	24
2.2.4 Variaciones del algoritmo EP.....	26
2.2.5 Aplicaciones actuales de entropía de permutación en señales temporales.....	30
2.3 Entropía de Permutación adaptada a imágenes.....	38
2.3.1 Segmentación de texturas no supervisada utilizando entropía de permutación (Li et al. 2006).....	38
2.3.2 Plano Complejidad-Entropía causal como una medida de complejidad para patrones bidimensionales (Ribeiro et al. 2012).....	40
2.3.3 Discriminación de texturas en imágenes con el plano Complejidad-Entropía causales multiescala bidimensional (Zunino & Ribeiro 2016).....	44
2.4 Conclusiones del capítulo.....	47
3. Métodos.....	49
3.1 Tratamiento de los datos para el cálculo de EP en una imagen.....	50
3.2 Propuesta 1: Algoritmos mediante análisis de la imagen dispuesta en un vector unidimensional.....	57
3.2.1 Entropía de permutación (EP).....	59
3.2.2 EP ponderada (EPP).....	59
3.2.3 EP de imagen umbralada.....	60
3.2.4 EP con umbral de comparación.....	60
3.3 Propuesta 2: Algoritmos mediante análisis de la imagen por el pixel y su vecindad.....	63
3.3.1 EP de imágenes filtradas.....	64
3.3.2 EP de valores medios.....	68

3.3.3 EP de matriz de entropías	69
3.3.4 EP con vecindad de Quincunx	70
4. Materiales.....	70
4.1 Imágenes sintéticas	71
4.2 Imágenes de prueba Reales.....	74
4.3 Mamografías.....	75
5. Resultados y discusión	80
5.1 Imágenes artificiales textura Lisa y Rugosa	80
5.1.1 Algoritmos mediante análisis de la imagen dispuesta en un vector unidimensional	81
5.1.2 Algoritmos mediante análisis de la imagen por el pixel y su vecindad.....	83
5.2 Imágenes artificiales texturas Gruesas y Finas	99
5.2.1 Ensayos con algoritmos de propuesta 1	100
5.2.2 Ensayos con algoritmos de propuesta 2	102
5.3 Imágenes reales	106
5.4 Mamografías (Caracterización de la densidad mamográfica)	113
6. Conclusiones.....	133
7. Anexo	135
8. Bibliografía.....	148

1. Introducción

1.1 Presentación del proyecto

El parámetro estadístico Entropía de Permutación (EP) es un indicador de la complejidad en señales temporales caóticas. En este proyecto se proponen distintos métodos para representar mediante EP la textura de una imagen.

La caracterización de imágenes mediante el análisis de su contenido, es un método transversal a todas aquellas disciplinas donde el Procesamiento Digital de Imágenes incumbe, particularmente en análisis de imágenes médicas, reconocimiento remoto como imágenes aéreas, o en inspección industrial. Parámetros de textura, forma y color, conforman un conjunto de características visuales propias de cada imagen.

La textura de una superficie hace referencia a la distribución de los valores de intensidad en el espacio que alberga la imagen de la misma, y puede ser índice de propiedades como fineza, rugosidad, suavidad, granulación, aleatoriedad, etc.

No hay una clasificación estándar para los métodos y técnicas en análisis de textura en imágenes, sin embargo una clasificación destacada es separar los métodos como: 1) Métodos estadísticos, 2) Métodos estructurales o sintácticos, 3) Métodos basados en Transformadas, y 4) Métodos basados en Modelos.

Muchos de los métodos y técnicas de análisis de textura tienen un desarrollo algorítmico y matemático complejo, sensibles al ruido y a transformaciones no lineales. Por esta razón, se consideró importante la implementación de EP para el análisis de texturas en imágenes, ya que presenta grandes ventajas frente a otros métodos estadísticos, como la simplicidad del algoritmo, el cálculo extremadamente rápido, la robustez frente al ruido dinámico y observacional, y la invarianza con respecto a las transformaciones monótonas no lineales.

1.2 Objetivos

El objetivo general del proyecto, es obtener el parámetro EP para imágenes de superficies, y adoptarlo como una característica de textura. Además, el algoritmo desarrollado para dicha función debe aprovechar las ventajas que presenta EP intrínsecamente: simplicidad, rapidez, robustez e invarianza a transformaciones no lineales.

Se eligieron distintas imágenes de superficies, cada una con una rugosidad que se diferencia visualmente de las otras. Las mismas poseen irregularidades de contraste, iluminación, saturación, etc, al momento de la adquisición, lo que es considerado como ruido. Para cumplir con el objetivo general, los objetivos específicos se centraron en los siguientes aspectos:

1. Demostrar la robustez y la simplicidad del método, por lo que no se realizó un pretratamiento de las imágenes para eliminar los efectos del ruido.
2. Realizar algoritmos con bajo costo computacional y sin complejidad de cálculo, para optimizar el tiempo de procesamiento y aprovechar la ventaja del cálculo extremadamente rápido de EP frente a otros métodos.

3. Realizar pruebas en las mismas imágenes sesgadas, aumentadas y transformadas, para demostrar que el método funciona para sistemas ergódicos, y es invariante a distintas transformaciones.
4. Utilizar las imágenes de superficies en tonos de gris, ya que se considera que la textura es un parámetro que depende de la distribución espacial de los tonos en una imagen, por lo que el color no es importante.
5. Los algoritmos propuestos se diferenciaron según dos propuestas: 1) analizar la imagen como una cadena secuencial de píxeles, es decir, los valores de los tonos de gris se disponen en un vector unidimensional 2) considerar que la imagen está constituida por un píxel y su entorno cercano, por lo tanto se toman distintas vecindades como entorno del píxel a analizar.
6. Seleccionar el/los algoritmo/s que brinden mejores resultados y realizar con ellos una aplicación móvil con el objetivo de probar la aplicabilidad del algoritmo en tiempo real.

1.3 Estructura del Proyecto

El proyecto se divide en cinco partes principales: Introducción, Estado del Arte, Métodos, Resultados y Propuesta de Aplicación. A continuación se resume lo desarrollado en cada una de las secciones.

En la Introducción se analizó la importancia del proyecto debido a sus potenciales alcances, conociendo las ventajas que Entropía de Permutación presenta en la actualidad, frente a otros parámetros indicadores de complejidad para series temporales.

En el Estado del Arte se realizó la investigación y recopilación de proyectos referidos a Análisis de Texturas en imágenes en general y en Entropía de Permutación en particular.

Dentro del capítulo Métodos, se explica y fundamenta cada uno de los algoritmos desarrollados mediante la simulación con la herramienta de software matemático Matlab 7.12.0 (R2011a). Además, se explican las consideraciones a tener en cuenta al adaptar para imágenes, un método desarrollado originalmente para series temporales.

En la sección Resultados, se muestra lo obtenido con las simulaciones expuestas en el capítulo anterior. Se aplica cada uno de los algoritmos a siete imágenes de superficies, cuyas rugosidades son visualmente diferentes. Se analizan los resultados obtenidos para cada una de ellas, comparando así la eficiencia entre los métodos propuestos.

Por último se muestran los resultados obtenidos mediante una Aplicación Móvil sencilla, realizada con el fin de demostrar las ventajas en la aplicabilidad EP, en comparación con la de otros métodos estadísticos caracterizadores de rugosidad conocidos.

2. Estado del Arte

En este trabajo, el capítulo “Estado del Arte” es crucial para comprender el motivo por el cual este proyecto fue desarrollado. Es aquí donde se justifica el porqué de la elección del tema abordado, y la importancia de la investigación del mismo.

Uno de los motivos, es la conocida importancia del análisis de texturas en imágenes, debido a su variada utilización en aplicaciones industriales, médicas, climáticas, satelital, etc., para realizar tareas como reconocimiento remoto con imágenes satelitales, segmentación de regiones en diagnóstico por imágenes en medicina, inspección de superficies en la industria, visión robótica, solo para mencionar algunos ejemplos.

Otro motivo, quizás el más importante, es el que justifica la adaptación de Entropía de Permutación en imágenes, ya que los trabajos hasta el momento conocidos para el cálculo de esta complejidad en un sistema, son comúnmente aplicados a sistemas en una dimensión, existiendo un vacío bibliográfico muy grande en la aplicación de este concepto para sistemas de dos o más dimensiones, como es el caso de las imágenes.

Por lo dicho anteriormente, es que se considera que este proyecto no es solo un aporte para obtener nuevos descriptores de texturas en aplicaciones electrónicas, sino también, son datos experimentales útiles en el área de la física, para verificar de manera experimental la caracterización de la complejidad a través de EP en sistemas bidimensionales.

Se explican los conceptos utilizados de Procesamiento Digital de Imágenes y se exponen los trabajos más importantes desarrollados en los temas involucrados en este proyecto: “Análisis de textura en imágenes” y “Entropía de Permutación como cuantificador de Complejidad”. Ambos temas convergen y son eje fundamental en este proyecto.

El Análisis de Texturas es un campo de investigación muy importante en el procesamiento de imágenes, debido a que puede ser usada de varias maneras. Sólo en el trabajo de (Gil Rodríguez 2008), donde se presenta el estado del arte del análisis de texturas a la fecha de edición, se reconocen 4 diferentes usos de la textura en imágenes: 1) segmentación de regiones y objetos característicos, 2) clasificación y reconocimiento de materiales en una superficie, 3) cálculo de la forma de una superficie, 4) mediante gradientes de textura, obtener información tridimensional de la imagen, como profundidad, buzamiento y curvatura de una superficie.

Debido a todas estas funciones, y principalmente a que la segmentación de texturas es un paso inicial crucial en la recuperación de imágenes y procesamiento de imágenes basado en el contenido, es que la búsqueda de nuevos y mejores métodos para su análisis está en vigencia permanente.

Como aporte al conocimiento y búsqueda de nuevos métodos, en este trabajo se propone adaptar el algoritmo de cálculo de EP para su implementación en imágenes, ya que este nuevo parámetro estadístico para series temporales, se calcula en base a la comparación con sus valores vecinos, definición que aplica como descripción de la característica de textura de una imagen (Li et al. 2006). Además EP es muy utilizado para el cálculo de complejidad en series temporales caóticas, presentando muchas ventajas frente a los otros métodos estadísticos conocidos, como la simplicidad del algoritmo, extremada rapidez de cálculo, robustez frente al ruido y además es invariante ante transformaciones no lineales.

Para abordar el estado del arte de Análisis de Texturas, se dividieron los métodos según la manera de procesar la imagen para obtener una característica de textura. Entonces, debido al tipo de procesamiento para obtener los descriptores de la textura, se discriminó entre 4 tipos de métodos.

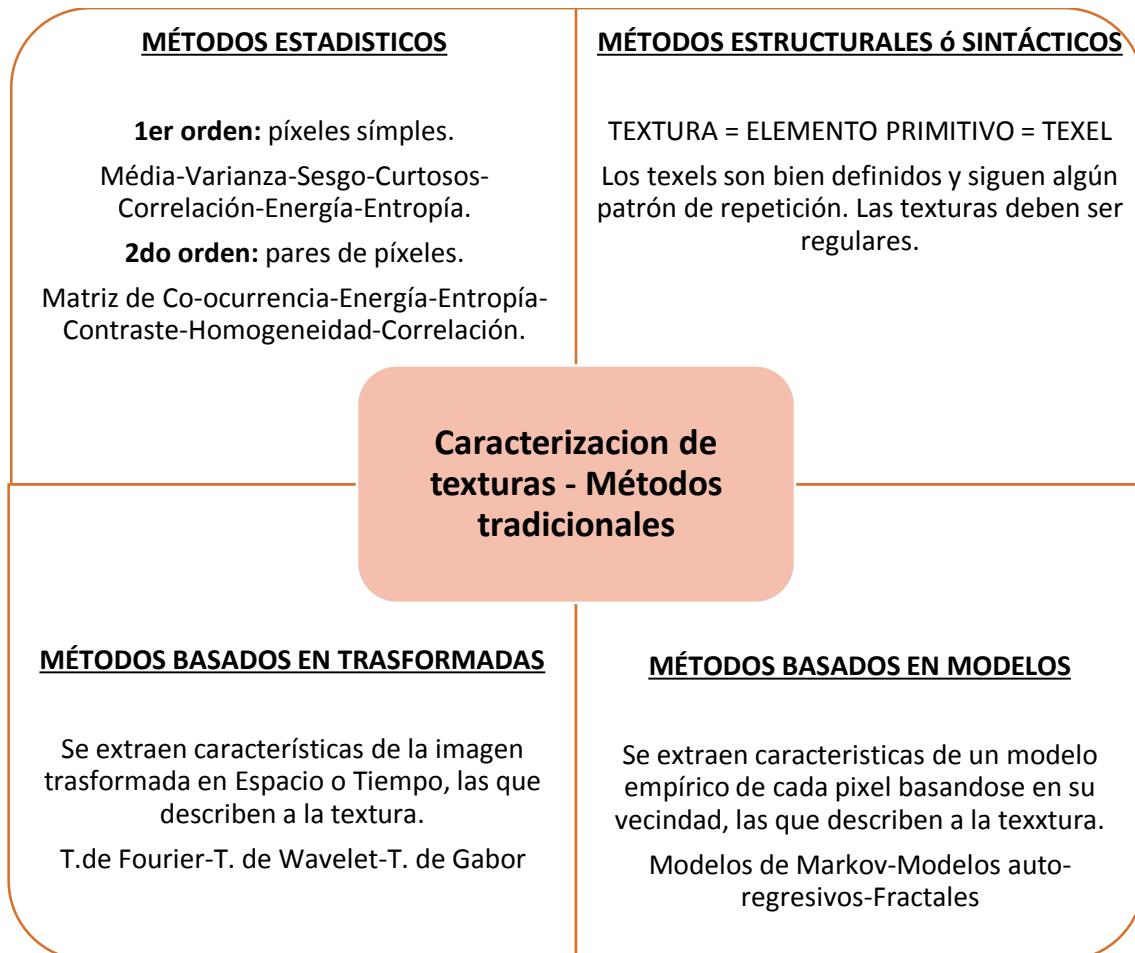


Ilustración 1: Métodos tradicionales para reconocimiento de texturas. Características y ejemplos

Por último se presentan los trabajos donde el parámetro EP es aplicado. Para ello se divide el estado del arte desde dos enfoques, uno es el tradicional, donde se aplica el algoritmo original desarrollado por sus creadores (Bandt & Pompe 2002) a series temporales. El otro enfoque es la adaptación de ese algoritmo para datos distribuidos en dos o más dimensiones, este último incluye la aplicabilidad a imágenes, siendo el tema desarrollado en este proyecto.

- 1) **EP en series temporales:** desde el primer trabajo donde Bandt y Pompe (2002) introducen este parámetro para series temporales, hasta las últimas aplicaciones de EP en 2016 para señales cardiacas y electroencefalogramas.
- 2) **EP adaptada a imágenes:** se muestran las primeras adaptaciones para el cálculo de complejidad en datos distribuidos en dos dimensiones o más, hasta los últimos trabajos donde es aplicado el descriptor Entropía de Permutación en imágenes para reconocimiento de texturas.

2.1 Métodos tradicionales de análisis de texturas en imágenes

Aquí se explican los métodos más conocidos para obtener descriptores de texturas en imágenes. Se explican más profundamente los Métodos Estadísticos, ya que pertenecen a la familia del parámetro aquí analizado (Entropía de Permutación), y dentro de los que a estos concierne, al cálculo de entropía y energía de la imagen.

2.1.1 Métodos estadísticos

Como se presentó en la introducción del capítulo Estado de Arte, se distinguen dos tipos de métodos estadísticos, los de primer y segundo orden. Los métodos se explicaran y se ejemplificaran sus usos en orden respectivo.

Una imagen en niveles de gris, se representa como una función espacial en dos dimensiones ($I(x, y)$), cuyo contenido indica la intensidad del tono de gris en cada coordenada espacial. También puede ser pensada como una matriz de dos dimensiones cuyo tamaño depende de la dimensión de la imagen y contenido representa la intensidad de gris del pixel.

Las medidas estadísticas de primer orden involucran pixeles simples, y son calculadas sobre el histograma de la imagen, cuya normalización proporciona la función densidad de probabilidad de la textura, a través del cual se pueden obtener rasgos locales y globales de la imagen, con el costo de perdida de información espacial de los datos, ya que dentro del histograma solo se considera la cantidad de repeticiones de los datos. Son métodos de algoritmos sencillos y de cálculo rápido, utilizando poca memoria para su obtención.

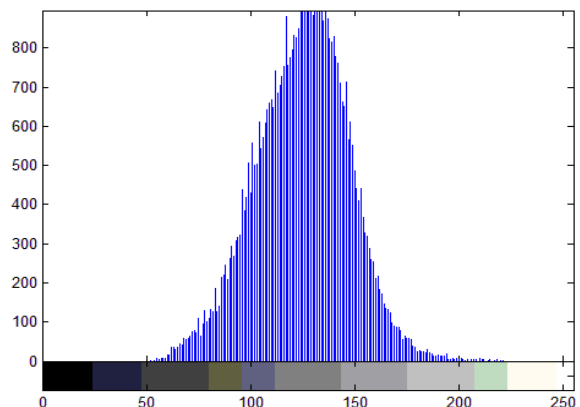
Es necesario definir algunos conceptos utilizados para el cálculo de estas medidas estadísticas, como: histograma, probabilidad del pixel, momento y momento central:

- El histograma de 1er orden ($h(i)$) es la cantidad de pixeles con intensidad i presentes en la imagen $I(x, y)$. Como ejemplo, se calcula el histograma de una de las imágenes utilizadas en este trabajo, de tamaño 255 x 255 pixeles, distribuidos en 255 niveles de gris.

Imagen $I(x, y)$



Histograma $h(i)$



- Obtenido el histograma se calcula la probabilidad de aparición del pixel $p(i)$ como la relación entre el número de pixeles con intensidad i y la cantidad total de pixeles en la imagen. De manera general para una imagen de tamaño $m \times n$, la probabilidad se calcula como:

$$p(i) = \frac{h(i)}{m \times n}$$

- Los momentos estadísticos de 1er orden m_k , se definen como la esperanza de la variable aleatoria i . Para el caso ejemplificado, cuya imagen presenta de 0 a 255 niveles de gris, se define como:

$$m_k = E[i^k] = \sum_{i=0}^{255} i^k p(i)$$

- El momento centra μ_k , se define para aquellas variables aleatorias que posean valor medio. Se define como la esperanza de la variable menos su valor media. Para la imagen ejemplo representada por 255 niveles de gris, el momento central se define como:

$$\mu_k = E[(i - m_1)^k] = \sum_{i=0}^{255} (i - m_1)^k p(i)$$

En la Tabla 1 se muestran parámetros estadísticos de primer orden, a partir de los cuales una textura puede ser caracterizada mediante la información obtenida del histograma normalizado de una imagen. Dicha información describe una característica particular de la textura, como suavidad, rugosidad, regularidad, granularidad, etc.

Estadísticos de 1er orden	Característica de la imagen	Expresión matemática discreta
Momento de orden uno (k=1)	Nivel medio de intensidad	$m_1 = E[i]$
Momento central de orden dos (k=2)	Varianza de la intensidad (ancho del histograma)	$\mu_2 = E[(i - m_1)^2] = Var(i) = \sigma^2$
Momento central de orden tres (k=3)	Coefficiente de Asimetría (asimetría respecto de la media, skewness)	$\mu_3, \gamma_1 = \frac{\mu_3}{\mu_2^{3/2}} = \frac{\mu_3}{\sigma^3}$
Momento central de orden cuatro (k=4)	Coefficiente de Kurtosis (aplanamiento respecto a una distribución normal, sharpness)	$\mu_4, \gamma_2 = \frac{\mu_4}{\mu_2^2} = \frac{\mu_4}{\sigma^4}$

Entropía	Uniformidad del histograma (mayor aleatoriedad de la textura, mayor es la entropía)	$H = -E[\log_2 p(i)]$ $= -\sum_{i=0}^{255} p(i) \log_2 p(i)$
----------	---	--

Tabla 1: estadísticos de 1er orden - caracterizadores de texturas en imágenes

Ventajas y desventajas: Si bien son algoritmos sencillos y de cálculo rápido, utilizando poca memoria para su obtención, la desventaja principal de los métodos estadísticos de 1er orden, es la falta de sensibilidad ante las permutaciones de los píxeles, debido a la pérdida de la información espacial de los mismos, por lo que dos imágenes diferentes pueden tener el mismo histograma. Los métodos estadísticos de 2do orden, vienen a dar solución a este inconveniente, ya que se basan en la observación de a pares de valores de gris, donde se establece una orientación y distancia entre el “dipolo” de píxeles formado, teniendo en cuenta la posición relativa de los mismos dentro de la textura, se los puede definir como histogramas bidireccionales.

La mayor fiabilidad y robustez de los estadísticos de 2do orden, conlleva un mayor costo computacional. Sin embargo, métodos descriptores de texturas como la matriz de co-ocurrencia o GLCM (*grey level cooccurrence matrix*), es el estadístico de 2do orden más utilizado en el procesamiento digital de imágenes, adecuado para texturas con origen aleatorio, las que no reportan un patrón determinado.

Este método permite extraer una gran cantidad de información de textura de imagen, por la gran variedad de descriptores que es posible obtener de esta matriz, lo que hace posible caracterizar con un conjunto de valores cuantificables cada imagen analizada. Debido a la importancia de este descriptor, se realiza la explicación del método de forma detallada, lo que con posterioridad se consultará al momento de comparar éste método con el propuesto en este proyecto.

La matriz co-ocurrencia se define en base a los parámetros distancia (d) y orientación (\emptyset), dada entre los pares de píxeles dentro de la textura de la imagen dada. El parámetro d indica la cantidad de píxeles que conforman la vecindad del píxel central, dando dimensión a la ventana móvil con la que se recorre la imagen. Mientras que \emptyset es la dirección específica entre los píxeles vecinos (0° 45° 90° 135°) se puede expresar como:

$$A_d^\emptyset(i_1, i_2) = p(I(x, y) = i_1, I(x \pm d f(\emptyset), y \pm d g(\emptyset)) = i_2)$$

Ecuación 1: matriz A de co-ocurrencia

El cálculo de probabilidad p se realiza sobre la vecindad definida de tamaño d , entrega un resultado único, el cual reemplaza el valor del píxel central de la ventana móvil. La matriz calculada indica la relación entre los G distintos niveles de gris presentes en la imagen.

Como último paso se normaliza la tabla de co-ocurrencia A de tamaño $G \times G$ por cada dirección y tamaño de ventana, resultando una matriz de probabilidades, sobre la cual se calculan índices de: Homogeneidad, Contraste, Disimilaridad, Media, Desviación Estándar, Correlación, Entropía y Energía. Estos índices, no serán calculados sobre la imagen original, sino sobre una nueva imagen cuyos valores indican la relación entre píxeles.

En la Tabla 2 se describen algunos de los descriptores estadísticos de 2do orden calculados sobre la matriz de co-ocurrencia más importantes.

Estadísticos de 2do orden	Característica de la imagen	Expresión matemática discreta
Momento diferencial inverso (<i>Inverse Difference Moment</i>)	Homogeneidad de la imagen. Valores de la diagonal de la matriz co-ocurrencia altos. Cuanto mayor es el contraste, más alto es este valor.	$IDF = \sum_{i,j=0}^{N-1} \frac{A_{i,j}}{1 + (i - j)^2}$
Contraste	Es opuesto a la homogeneidad. Valor alto los valores alejados a la diagonal principal son altos.	$CON = \sum_{i,j=0}^{N-1} (A_{i,j} \times (i - j)^2)$
Segundo momento angular (<i>Angular Second Moment</i>). Energía	Indica la energía de la matriz de co-ocurrencia, la que es una tabla de probabilidades, por lo que la ecuación de energía maximiza los valores grandes y minimiza los pequeños. Es mayor cuanto más suave es la textura.	$ASM = \sum_{i,j=0}^{N-1} A_{i,j}^2$
Entropía	Toma valores bajos cuanto más suave es la textura. Es mayor cuanto más aleatoria es la matriz co-ocurrencia.	$H = - \sum_{i,j=0}^{N-1} A_{i,j} \log_2 A_{i,j}$

Tabla 2: estadísticos de 2do orden - caracterizadores de texturas en imágenes-matriz de co-ocurrencia

Se desarrollaron los métodos más utilizados en la actualidad para caracterizar texturas, dentro del análisis estadístico de las imágenes, donde la más importante es la matriz de co-ocurrencia. Sin embargo, existen otros análisis estadísticos sin perder información espacial de la imagen, que son utilizados, pero con menor impacto de desarrollo, lo que no serán tratados en este proyecto.

Otro método muy conocido para obtener descriptores de textura, y que guarde la información espacial de los píxeles en la imagen, es la función de autocorrelación. A diferencia del cálculo de caracterizadores con la matriz de co-ocurrencia, éste parámetro se obtiene mediante un procesamiento directo sobre la imagen.

La función autocorrelación de una imagen, puede usarse para evaluar tanto la regularidad como la rugosidad de la textura. Sea una imagen $I(x, y)$, de tamaño $m \times n$ con G niveles de gris, la función autocorrelación $\rho(x, y)$ se calcula como:

$$\rho(x, y) = \frac{(m \times n) \sum_{l=0}^{n-x-1} \sum_{k=0}^{m-y-1} I(l, k)I(l + x, k + y)}{(n - x)(m - y) \sum_{l=0}^{n-1} \sum_{k=0}^{m-1} I^2(l, k)}$$

2.1.2 Métodos estructurales

La definición de textura no presenta un único concepto, ya que depende del área de análisis de la textura es la interpretación de la misma. La literatura relativa al procesamiento de imágenes posee un gran número de definiciones de este concepto, dependiendo del autor y del propósito del trabajo donde el análisis de una textura será aplicado.

Cuando en la definición de textura se presenta el concepto de “elemento de textura” o “primitiva” o “texel”, el método utilizado se clasifica bajo el título de modelo estructural o geométrico, donde el análisis depende principalmente de las propiedades geométricas de estos elementos de textura.

Una primitiva es un conjunto conectado de celdas de resolución caracterizadas por una lista de atributos (forma, tamaño, nivel de gris o color, etc.). La primitiva más simple es el pixel cuyo tono de gris es su atributo único, siendo inútil trabajar sólo con él para determinar elemento de textura.

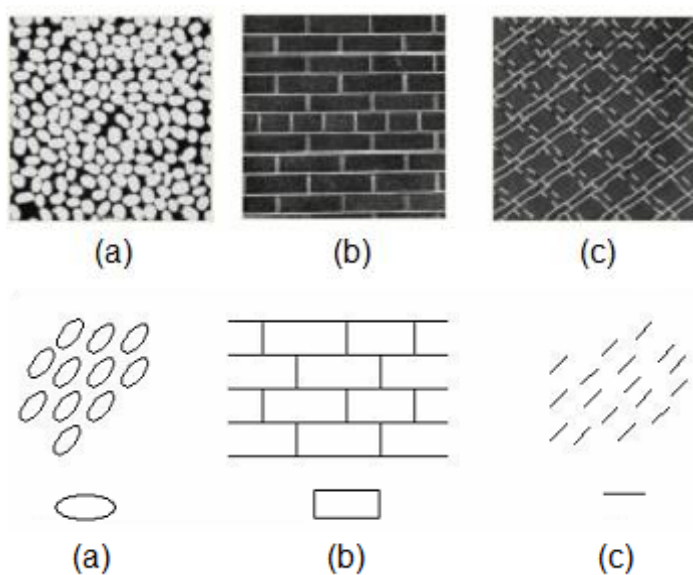


Ilustración 2: ejemplo de texturas y sus primitivas

El método se funda en que una textura se genera por el desplazamiento de estos elementos, bien definidos, distribuidos por la textura con algún patrón de repetición. El análisis de textura estructural consiste en dos pasos esenciales 1) identificación de elementos de textura (primitivas) y 2) extraer una regla de colocación de las primitivas extraídas. Para explicar los conceptos anteriores, se asume que la imagen ya se encuentra segmentada, y con los elementos de texturas identificados, desde este punto de partida se analiza la textura desde dos posibles enfoques, en uno se computan las propiedades estadísticas generales de cada elemento de textura extraído y los utiliza como descriptores de la textura general, en el otro se extrae la regla

de ubicación que describa la textura, aproximación que puede conllevar métodos geométricos (*extracción de tokens con partición de Voronoi*) o sintácticos (*gramáticas de cadenas, gramáticas de grafos y agrupamiento de primitivas*). Estos dos procedimientos, geométrico y sintáctico, se explicaran en este proyecto solo de manera conceptual, ya que la aplicabilidad de este método en general, no se corresponde con las imágenes de superficies aquí analizadas.

El llamado mosaico de Voronoi fue propuesto para la extracción de elementos de textura en métodos geométricos, debido a que entre las propiedades deseables que este contiene, se encuentra la definición de vecindad espacial local y porque las distribuciones espaciales locales de los “tokens” (primitivas) se reflejan en las formas de los polígonos de Voronoi. Debido a estas propiedades fue que Tuceryan y Jain (Tuceryan y Jain, 1993) propusieron la extracción de tokens utilizando las propiedades del recubrimiento de un plano con patrones que no dejen regiones descubiertas (mosaico de Voronoi).

Para aplicar los métodos geométricos a imágenes en niveles de gris, se necesita primero que todo, extraer los tokens de la imagen texturada. A partir de los tokens resultantes se construye el mosaico de Voronoi y en cada celda de Voronoi se calculan los rasgos de la textura. Entonces, los tokens que posean rasgos similares se agrupan para construir regiones de textura uniformes, así se forma una textura.

En los métodos sintácticos utilizados son descriptos a continuación, se aprecia que la obtención de texturas, requiere de la determinación de las primitivas, la descripción de las mismas y la descripción de sus interrelaciones.

- Gramáticas de cadenas: generan textura empezando con un símbolo inicial, seguidos de reglas de transformación de formas.
- Gramáticas de grafos: construye un grafo plano de diseño de textura primitivo, para lo cual se deben conocer las clases primitivas y sus relaciones espaciales, permitiendo construir una estructura de grafos de textura.
- Agrupamiento de primitivas: primitivas de textura de bajo nivel forman algún patrón específico, el que puede ser considerado como una primitiva en un nivel de descripción alto.

Ventajas y desventajas: Estas técnicas resultan aplicables con éxito en el análisis de imágenes que contiene texturas muy regulares, por lo que estos algoritmos son de alcance limitado, y necesitan espacio en memoria donde almacenar posibles elementos estructurales. Sin embargo presenta la ventaja de donar resultados rápidos si la textura está formada por un elemento reconocible por el sistema.

2.1.3 Métodos basados en tratamiento de la señal

Todos los métodos utilizados para caracterizar la textura de una imagen, analizados en este capítulo, operan sobre la señal definida en el dominio espacial. Sin embargo, trabajar en otro dominio puede resultar más conveniente para facilitar la extracción de determinadas características de la imagen.

Estos métodos dependen del tipo de transformación aplicada, ya sean a través de Momentos, Transformada de Fourier, Modelos de Gabor, Wavelet.

Se realiza una introducción de los métodos más utilizados:

- Transformada de Fourier:

Está especialmente indicado para describir la direccionalidad de patrones bidimensionales periódicos de una imagen, ya que estos patrones de textura son fácilmente distinguibles como concentraciones altas de energía en el espectro.

Por ejemplo, propiedades como la rugosidad o suavidad de la textura, son proporcionales al periodo espacial de la imagen, una textura rugosa tendrá la mayor parte de su contenido espacial en altas frecuencias, mientras que las texturas suaves lo concentraran en las bajas frecuencias espaciales.

La transformada de Fourier de una función discreta de dos variables, como lo es una imagen digitalizada $I(x, y)$ de tamaño $m \times n$, está dada por:

$$F(u, v) = \frac{1}{m * n} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} I(x, y) e^{-j2\pi(\frac{ux}{m} + \frac{vy}{n})}$$

Ecuación 3: Transformada Discreta de Fourier

Donde $u = 0, 1, 2, \dots, m - 1$; $v = 0, 1, 2, \dots, n - 1$

Para la descripción de textura, son consideradas tres características del espectro de Fourier:

- 1) picos prominentes del espectro que dan la dirección principal de los patrones de textura.
- 2) localización de dichos picos en el plano de la frecuencia que da el período espacial fundamental de los patrones.
- 3) la eliminación de los componentes periódicos durante el filtrado deja elementos no periódicos en la imagen, que se pueden describir mediante técnicas estadísticas.

En la Ilustración 3 se muestra un ejemplo de una imagen digital y su espectro de potencia en dos dimensiones. La imagen presenta patrones direccionales en su textura, los que se reflejan en la concentración direccional de energía en el espectro. Por este motivo, el espectro es dividido en regiones para el análisis de texturas en el dominio de la frecuencia.

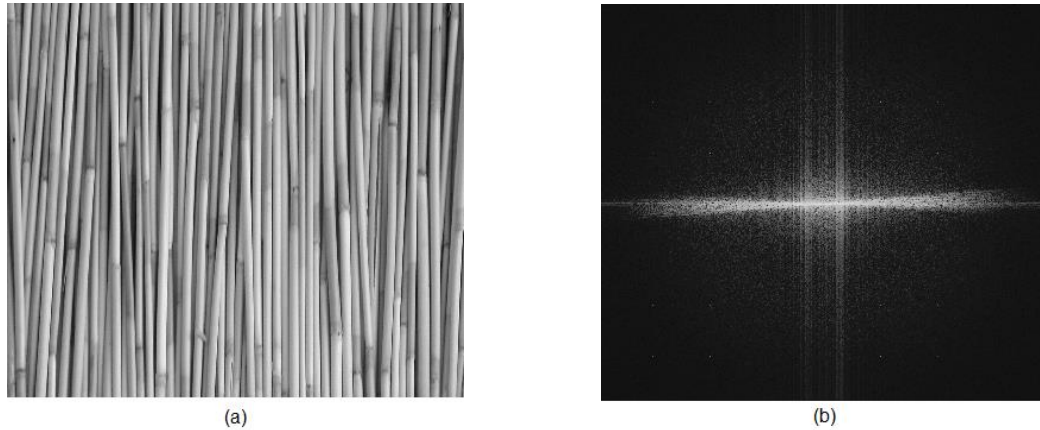


Ilustración 3: a) imagen original de textura con patrones direccionales, b) espectro de potencia visualizado como imagen. Se observan las concentraciones de energía en distintas direcciones.

La imagen es dividida en regiones para obtener su Transformada de Fourier, la que a su vez se divide en *bines* (regiones de interés en el espectro de frecuencia), para analizar la textura de la imagen en el dominio de la frecuencia, estas regiones pueden ser radiales para hacer una división en bandas o angulares. Los valores de entropía son calculados sobre cada uno de estos *bines*.

Para el análisis mediante métodos frecuenciales, se suele utilizar el concepto de entropía, siendo uno de los conceptos más importantes por los que se desarrolla este trabajo. En el caso de análisis en el espectro de una textura, la entropía representa la probabilidad de ocurrencia de distintas frecuencias, utilizando la densidad espectral de potencia de la imagen, como:

$$h = - \sum_{u,v} [p(u,v) \log(p(u,v))]$$

Ecuación 4: entropía del espectro de la imagen

Donde u, v , pertenecen a una región R determinada del espectro y $p(u, v) = \frac{|F(u,v)|^2}{\sum_{u,v}|F(u,v)|^2}$, es la energía normalizada de la región.

Ventajas y desventajas: El mayor problema de utilizar métodos espectrales para caracterizar texturas, se debe a la dependencia del espectro frecuencial a la imagen global. Si existe una variación espacial del patrón de textura en la imagen, no se reflejará en el espectro. Una mejora de este procedimiento es utilizar métodos multi-resolución (Modelos de Gabor y Wavelet).

- **Métodos multi-resolución:** Modelos de Gabor

La transformada de Fourier proporciona un análisis de la frecuencia de la imagen global. Hay veces que es más apropiado localizar el análisis en el dominio espacial. Con estos modelos el análisis espectral también considera el espacio de escala y una cierta

orientación privilegiada. Ambos tipos de métodos son capaces de caracterizar de manera eficiente texturas a diferentes escalas, combinando características espaciales y frecuenciales de la imagen de textura.

Las transformaciones Gabor son particularmente útiles para describir texturas con características específicas de orientación o de altas frecuencias. Estos filtros son una herramienta muy poderosa cuando las texturas tienen una orientación predeterminada, pero no en caso contrario, ya que no son invariantes a la rotación.

La máscara de convolución que utiliza este método, se construye con una función gaussiana modulada con un armónico de una determinada orientación \emptyset y frecuencia f :

$$G_{\emptyset,f}(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \cdot e^{2\pi jf(x \cos \emptyset + y \sin \emptyset)}$$

Ecuación 5: filtro de Gabor-máscara de convolución

Donde, la escala del patrón de la textura se define por la relación entre la varianza σ y la frecuencia f como: $\sigma = 3 \frac{\sqrt{2 \ln 2}}{2\pi} f$. Normalmente, se emplean cuatro orientaciones $\emptyset = \left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\right\}$ y tres diferentes frecuencias $f = \{0.2, 0.35, 0.5\}$, resultando un conjunto de doce máscaras de convolución. Estas máscaras son convolucionadas con la imagen original, cuyos resultados de alto contenido espectral en una dirección indican la presencia del patrón de textura (Platero Dieñas 2009).

Ventajas y desventajas: El principal problema de los filtros de Gabor es que no son ortogonales, lo que provoca que la correlación entre texturas distintas tenga un valor significativo. Además, para poder caracterizar una textura a diferentes escalas con estos filtros, es necesario modificar los parámetros de los filtros para cada escala. Estos problemas son solucionados por la transformada wavelet, que proporciona una herramienta precisa y unificada para el análisis y caracterización de una señal a diferentes escalas (Marfil et al. 2004).

- **Métodos multi-resolución: Wavelets**

El método de Ondículas o Wavelets, tiene como característica principal mantener la estructura textural de las imágenes analizadas a la vez que se llevan a cabo compresiones sobre las mismas, lo que permite variar la resolución espacial y representar texturas de diversa naturaleza (Fernández Sarría 2007). Existen algoritmos eficientes para el cálculo de la transformada de Wavelet como el denominado FWT (transformada rápida de Wavelet), y otra herramienta más reciente y flexible denominada Esquema Lifting.

La transformada discreta wavelet descompone una imagen en un conjunto de imágenes ortonormales más pequeñas. Mientras el histograma de los niveles de gris de la imagen

original puede tener cualquier forma, el correspondiente a las transformadas wavelet es generalmente monomodal o simétrico. Este resultado simplifica el análisis estadístico de las propiedades de una imagen (Cova & Cavallero 2006), como la característica de la textura.

El análisis con Ondículas, es una herramienta que permite revelar aspectos de los datos que otras técnicas de análisis pierden, tales como tendencias y discontinuidades. Además facilita la visión diferente de los datos de aquellas presentadas por las técnicas tradicionales. Este análisis puede ser utilizado para la compresión o eliminación del ruido en la señal sin degradación apreciable en la misma. Desde su existencia como técnica de procesamiento de señales, el análisis mediante Wavelet ha probado su utilidad en numerosos campos de aplicación (Passoni 2005).

El análisis de multiresolución permite estudiar una imagen en distintas resoluciones y extraer características propias de cada resolución. Al realizar el análisis de la imagen a través de la transformada wavelet, la definición de la transformada de wavelet depende directamente de una constante de dilatación (la que definiremos con la letra α), por lo que el análisis de *multiresolución* se encuentra en forma implícita en su cálculo.

Para poder caracterizar una textura consideramos sus frecuencias espaciales. Si deseamos detectar o contornear zonas de diferentes tipos de texturas debemos realizarlo con valores de $\alpha < 1$ lo más pequeño posible (α está limitado por la resolución de la imagen), para buscar las altas frecuencias en la imagen y si deseamos localizar zonas de bajas frecuencias debemos aumentar el valor de α .

La técnica para detectar diferentes texturas es la siguiente,

- 1) realizamos la transformada wavelet en una resolución determinada, según que frecuencias estemos buscando asociadas con esas texturas en la imagen.
- 2) aplicamos un filtro pasa altos los coeficientes obtenidos y a los restantes los multiplicamos por un factor α , para que al realizar la transformada wavelet inversa resalte los detalles en forma más pronunciada.
- 3) restamos la imagen original de la filtrada, de este modo obtenemos puntos en la imagen que nos da la separación de las diferentes texturas o zonas de igual frecuencia.

Ventajas y desventajas: Wavelet es uno de los métodos más utilizados actualmente por variedad de proyectos, principalmente debido a la robustez que presenta frente a los cambios de escala como principal ventaja. Sin embargo presentan un elevado coste computacional debido a que para realizarlo se requieren varios niveles de descomposición.

2.1.4 Métodos basados en modelos

Estos métodos se basan en la construcción de un modelo empírico de cada píxel de una imagen, basado en una suma ponderada de las intensidades de los píxeles que conforman la vecindad del mismo (Alonso 2011). Los parámetros estimados a partir de los modelos obtenidos, se utilizan para describir y sintetizar la textura de la imagen. Dentro de este grupo, están los

modelos de campo aleatorio como los Modelos Aleatorios de Markov (MRF), y los métodos basados en Geometría Fractal. Ambos se explicaran solo de manera conceptual en este proyecto.

- Modelos aleatorios de Markov:

Los métodos que utilizan los campos aleatorios de Markov gaussianos caracterizan la relación estadística entre un píxel y sus vecinos. Es decir, este método se basa en la hipótesis de que la intensidad de un píxel depende de las intensidades de los píxeles vecinos. El modelo estocástico resultante consta de un número de parámetros igual al tamaño de la máscara de la vecindad. Los parámetros se pueden estimar mediante un algoritmo de mínimos cuadrados sobre cada posición de la máscara en la imagen.

La máscara se forma tomando una serie de píxeles vecinos, siendo las más sencillas la de primer orden, formada por los cuatro píxeles directamente conectados a él, y la de segundo orden, formada por los ocho píxeles que lo rodean. Una máscara bastante habitual es la máscara simétrica.

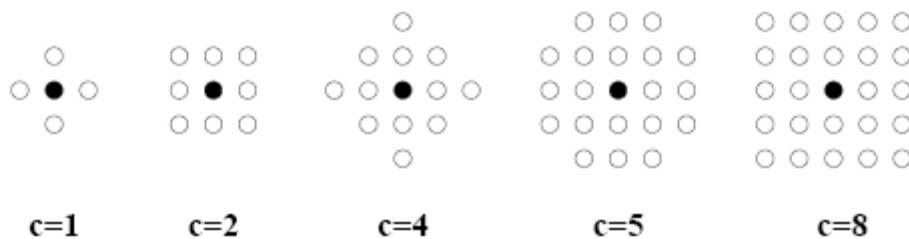


Ilustración 4: máscaras simétricas de distintos tamaños (c) que definen la vecindad del píxel

- Fractales:

La mayoría de los métodos fractales usados para analizar características en imágenes de tejido se basan en la estimación de la dimensión Fractal relacionada con el exponente de Hurst que caracteriza la rugosidad (coarseness) de la superficie global (Pentland, 1984). Existe una correlación entre la dimensión fractal de una textura y su rugosidad, la que se demostró en el trabajo desarrollado por Pentland.

La descripción fractal de texturas se basa en gran medida en la determinación de la dimensión fractal (D). Dicha dimensión fractal es independiente de la escala, esta propiedad se la conoce como auto-similaridad, la cual es muy conveniente para el análisis de texturas en imágenes cuyo escalado puede variar mucho dependiendo del método de captura.

“Un conjunto acotado A es auto-similar, si es la unión de N_r copias no solapadas de un conjunto similar a A , pero escalado por un factor r ”. A partir de esta definición la dimensión fractal se define como:

$$D = \frac{\log N_r}{\log r^{-1}}$$

Ecuación 6: Dimensión Fractal

La dimensión fractal la podemos aproximar si estimamos N_r para varios valores de r , con estos datos se determina la pendiente del ajuste lineal por mínimos cuadrados de $\log N_r$ versus $\log r^{-1}$.

El valor de D caracteriza la rugosidad de la superficie de la imagen tratada, de modo que cuanto mayor sea este valor, mayor será la rugosidad que presenta la superficie.

2.1.5 Comparación entre métodos

La extracción de características de textura se basa en modelos matemáticos que pueden involucrar cualquiera de los métodos descritos anteriormente entre muchos otros. La elección del método a aplicar, dependerá de la naturaleza de la imagen, de las características que se consideran importantes obtener y del análisis de las ventajas y desventajas que cada procedimiento presenta. A continuación se presenta un breve resumen de comparaciones entre métodos caracterizadores de texturas realizado por varios autores, y referenciados en fuentes varias como, estados de arte reconocidos, tesis doctorales, publicaciones en papel, publicaciones en la WEB y en los proceedings de revistas de impacto.

Haciendo referencia al primer método descrito en este trabajo, los autores Tuceyran y Jain (1998) concluyen que los rasgos de coocurrencia están mejor preparados para el análisis de textura que para la segmentación de la imagen. También, ellos concluyen que los rasgos de autocorrelación pueden usarse para evaluar la cantidad de regularidad así como la finura o grosor de la textura presente (Gil Rodríguez 2008).

Por otro lado los métodos basados en el uso de la Transformada de Fourier y el estudio del espectro de frecuencias de la señal, carece de una falta de correspondencia espacial con la periodicidad que se puede extraer a partir del análisis de las frecuencias existentes (Melchor 2016). Por este motivo variantes a esta como la STFT (Short Time Fourier Transform) son las actualmente más empleadas. En estos casos la necesidad de establecer un valor de "ancho de ventana" para fijar espacialmente los resultados de la transformada es un elemento demasiado restrictivo, lo que se considera una desventaja en su implementación.

Métodos basados en el dominio espacio-frecuencial, ofrecen una mejor precisión respecto de los estadísticos y los frecuenciales implementados de manera independiente, junto con una mayor inmunidad al ruido. Así en algunas propuestas importantes como el estándar MPEG-7 (Sookocheff, 2004) se propone su uso en clasificación utilizando una medida de *structuredness*, la que hace referencia a la periodicidad en texturas y está siendo implementada con wavelets (de Gabor mayormente) u otras junto a esquemas multiresolución en el caso de JPEG2000 (Melchor 2016).

Según Gil, J.L. (1996) el modelo fractal ofrece medidas que cuantifican la irregularidad en imágenes muy complejas y de pobre resolución.

El modelo estructural basado en patrones de primitivas y reglas de localización, parece estar en desuso por la complejidad que presenta en la selección de los primeros y en la definición de las reglas. Sin embargo, este enfoque estructural posee cualidades muy favorables para las texturas de comportamiento uniforme y regular (sintetizadas por el hombre) lo que no favorece la discriminación de las texturas creadas por la naturaleza (Gil Rodríguez 2008).

Existen otros métodos para caracterizar texturas en imágenes, como por ejemplo otras definiciones de momentos estadísticos basados en cálculo de características sobre dichas texturas, como las de Wold (Liu y Picard, 1994) o Tamura (Lin et ál., 2001), donde se habla de propiedades como coarseness, contrast, direccionalidad, regularity, line-likeness o roughness. Además la textura como forma de análisis de su contenido ha dado pie a la aparición de algunos “operadores” para cuantificarla, como los algoritmos de esqueletización (JAI) o el MAT (Medial Axes Transform) (Melchor 2016).

En su Reporte Técnico de Reconocimiento de Patrones, Gil Rodríguez (2008) describe el estado actual de la representación y análisis de textura en imágenes, donde concluye que se aprecia que continúa la tendencia de crear nuevos algoritmos para el cálculo de nuevos descriptores. Ha cobrado mucha fuerza el Modelo de Multiresolución sobre todo con el impacto que han tenido los filtros de Gabor y los paquetes de Wavelet en los estudios multiescala de la textura.

Aquí es donde se encuentra el desarrollo de este trabajo, en busca de un nuevo descriptor de texturas que venza las complejidades que otros presentan y con resultados aceptables. Para determinarlo no solo es necesaria la creación de algoritmos eficientes, sino también el ensayo de los mismos sobre casos reales.

2.2 Entropía de Permutación para series temporales

Por definición, *“La Entropía de Permutación (EP) es una medida de complejidad adecuada para series de tiempo caóticas, en particular en presencia de ruido dinámico y observacional”* (Bandt & Pompe 2002). Para poder entender esta definición, se desarrollan en este capítulo algunos conceptos previos referidos a señales como series temporales desde la mirada de la Teoría de la Información. Además, se explica detalladamente el algoritmo original para el cálculo de Entropía de Permutación como un parámetro de complejidad para series temporales caóticas, a partir del cual, se desprenden muchas otras variaciones del algoritmo para series temporales, las cuales también son explicadas. Muchas de estas variantes fueron utilizadas en los Métodos de Adaptación de EP para Imágenes propuestos en este proyecto. Por último se presentan todos los trabajos donde se utiliza el parámetro EP como caracterizador en series temporales.

2.2.1 Conceptos previos

Muchas de las señales reales son el resultado de dinámicas muy complejas o de varias dinámicas acopladas, características del sistema que los originan (Mateos 2016). Esta situación se puede encontrar en series temporales originadas en sistemas biológicos, en las Ciencias Sociales, Meteorología, Finanzas, en la mayoría de los eventos físicos, etc. Dichos sistemas representados en una serie temporal, pueden comportarse de diferentes maneras: regular, caótica o aleatoria.

Por definición, “Un Sistema Caótico es un sistemas complejo y dinámico muy sensibles a las variaciones en las condiciones iniciales”. Pequeñas variaciones en dichas condiciones iniciales pueden implicar grandes diferencias en el comportamiento futuro, imposibilitando la predicción a largo plazo. Esto sucede aunque estos sistemas son en rigor deterministas, lo que indica que su comportamiento puede ser completamente determinado conociendo sus condiciones iniciales.

Uno de los retos para el análisis de señales complejas, es extraer de ellas información significativa y relevante, que nos permita comprender su conducta. En los últimos años los métodos de estudio de las series temporales se han visto enriquecidos con conceptos y técnicas provistas por la teoría de la información y por herramientas provenientes de distintos campos de la física, tales como la teoría de los sistemas caóticos, la fractalidad y las medidas de complejidad.

Por definición, “La complejidad es la medida del nivel de dificultad requerido para expresar o predecir las propiedades de un sistema”. Los alcances de la medida de complejidad son variados, se desarrollaron diversas medidas de la Complejidad para comparar series temporales y distinguir entre tres tipos de comportamientos de un sistema, *regular, caótico y aleatorio*.

Las formas de caracterizar la complejidad en la ciencia son variadas y generalmente están ligadas al concepto de sistema. Los estudios del concepto de complejidad han tenido gran impacto en varias áreas del conocimiento como por ejemplo, las neurociencias, la biología y los sistemas dinámicos no lineales. Entre otros, se ha informado de que la Complejidad de las señales del corazón y del cerebro puede distinguir entre sujetos sanos y enfermos y en ocasiones incluso predecir un ataque al corazón o un ataque epiléptico.

En las últimas décadas ha habido un crecimiento notable en la definición de distintas medidas de complejidad. Basta mencionar la complejidad algorítmica de Kolmogorov, la complejidad de Lempel-Ziv, la Entropía de Permutación de Bandt y Pompe, entre otros. Todos ellos se definen para sistemas dinámicos y presumiblemente ergódicos.

Debido a su posterior mención en este trabajo, se realiza una introducción a las medidas de complejidad mencionadas, desde el punto de vista de la Teoría de la Información:

- **Por definición,** “la complejidad algorítmica de Kolmogorov es una medida de la cantidad de recursos computacionales necesarios para describir una cierta cantidad de información” (Mateos 2016). En contraste con la noción de tasa de entropía de los procesos estocásticos introducida por Shannon varios años antes, Kolmogorov propuso una medida de complejidad que no se basa en la noción de la probabilidad de un evento, ni en la necesidad de asociar una distribución de probabilidad de un objeto finito (cadena), la cual es necesaria para una definición rigurosa de la tasa de entropía de los procesos estocásticos.

Es bastante lógico considerar la longitud más corta para la descripción de un objeto como una medida para su complejidad. Formalmente la complejidad de Kolmogorov $K(s)$ de una cadena S , es la longitud del programa binario más corto (*prog*):

$$K(s) = \min_{prog} \{largo(prog)\}$$

Ecuación 7: Complejidad de Kolmogorov

El problema de encontrar la complejidad de Kolmogorov de una cadena puede ser reducido al problema de encontrar un esquema de compresión sin pérdida de información de la cadena original.

- **Por definición,** “la complejidad Lempel-Ziv (LZC) es una técnica de análisis no lineal que evalúa la aleatoriedad de secuencias finitas en función del número de patrones distintos existentes en una señal y la codificación simbólica” (Picó et al. 2014). Esta medida relaciona el número de patrones distintos que se encuentran a lo largo de una secuencia. Lempel y Ziv inspirados en el concepto de complejidad de Kolmogorov, se interesaron en la “historia óptima de una secuencia”, es decir, la mínima producción necesaria para generar una secuencia, llamada complejidad Lempel-Ziv ($C_{LZ}(S_{0:T-1})$).

Una clase importante de técnicas para la compresión de cadenas ha sido desarrollada por estos autores. Los algoritmos de compresión que han propuesto hacen uso de diccionarios adaptativos. Esta idea se remonta a la época del telégrafo, era más práctico generar un libro con los códigos de las frases con mayor frecuencia de uso, que codificar letra por letra cada vez que se enviaba un mensaje. Por lo tanto, era posible que el remitente comprimiera un mensaje a transmitir y el receptor lo descomprimiera de nuevo sin ninguna pérdida de información. Lempel y Ziv tomaron esta idea y la mejoraron generando un diccionario autoadaptado.

Se considera una secuencia de tamaño finito de símbolos $S_{0:T-1} = S_0 \dots S_{T-1}$, pertenecientes a un alfabeto A de tamaño finito. Basándose en la idea de complejidad de Kolmogorov para la generación de la secuencia, y particularizando a la idea de un programa construido con solo dos operaciones (“reproducción” y “producción”), Lempel y Ziv avanzaron en un esquema de compresión.

Hasta aquí se ha realizado un análisis de la secuencia puramente determinista, sin embargo la complejidad $C_{LZ}(S_{0:T-1})$ incluye la noción de información en un sentido estadístico, siempre que el tamaño de la secuencia sea lo suficientemente grande en relación al tamaño total del alfabeto.

Para un proceso aleatorio estacionario y ergódico, correctamente normalizado, la complejidad de Lempel-Ziv asociada con este proceso tiende a la tasa de entropía del mismo:

$$\lim_{T \rightarrow \infty} C_{LZ}(S_{0:T-1}) \frac{\log T}{T} = \lim_{T \rightarrow \infty} \frac{H(S_{0:T-1})}{T}$$

Donde, $H(S_{0:T-1})$ es la entropía conjunta de los símbolos S , y $\lim_{T \rightarrow \infty} \frac{H(S_{0:T-1})}{T}$ es la tasa de entropía por símbolo del proceso.

2.2.2 Método de cálculo y características

El contenido de información de un sistema se evalúa típicamente a través de una función distribución de probabilidad (*fdp*) de un sistema observable, cuya evolución puede ser observada a través del tiempo, correspondiendo en una serie temporal $x(t)$.

La entropía de información, puede considerarse como una medida de la incertidumbre asociada al proceso descrito por una *fdp*. Este método tradicional presenta algunos inconvenientes como:

1. Las medidas de entropía clásica como la Entropía de Shannon, descuidan las relaciones temporales existentes entre los valores de $x(t)$, por lo que los posibles patrones temporales existentes en el proceso no se tienen en cuenta. Este inconveniente produce que series con los mismos valores en distintos órdenes posean la misma entropía. Como consecuencia las relaciones de orden y las escalas de tiempo de la dinámica se pierden, como por ejemplo el caso habitual del uso de histogramas.
2. Estas medidas tradicionales de entropía suponen algún conocimiento previo sobre el sistema, por lo que la distribución de probabilidad asociada a la serie temporal analizada debe ser conocida. La representación más adecuada de la *fdp* representa un problema fundamental, el que conlleva la aplicación de distintos procedimientos (frecuencia de conteo, estadística de amplitudes, dinámica simbólica binaria, análisis de Fourier o wavelet). La aplicabilidad de estos procedimientos depende en concreto de características de los datos tales como: estacionariedad, longitud de la serie temporal, variación de los parámetros, nivel de ruido, etc. Se busca una manera más general e independiente del sistema para definir una *fdp*.
3. Los métodos clásicos para la determinación de entropía de un sistema, entregan buenos resultados frente a sistemas lineales y poco describen a los regímenes caóticos no lineales.

En el trabajo "*Permutation Entropy: A Natural Complexity Measure for Time Series*" (Bandt & Pompe 2002), se abordan estos inconvenientes, otorgando una solución mediante la introducción de un método simple y robusto que considera la causalidad mediante la comparación de los valores vecinos en una serie de tiempo. La secuencia apropiada de símbolos surge naturalmente de la serie de tiempo sin ningún conocimiento previo asumido del sistema. Las "particiones" (bines del histograma) son naturalmente generadas por la comparación del orden de los valores relativos entre los vecinos, en lugar de un prorrateo de amplitudes de acuerdo a diferentes niveles. Basado en este análisis simbólico, se construye la Entropía de Permutación (*EP*). Su uso ha sido una clara mejora en la calidad de la información de los cuantificadores basados en la teoría de la información.

Considerando la serie de tiempo: $\{x_t\}_{t=1,\dots,T}$, estudiamos todas las $n!$ permutaciones π de orden n que se consideran como posibles tipos de órdenes de n números diferentes, siendo x_t tomada de n vectores de n elementos. En la práctica se recomienda que la dimensión del vector de embedding se encuentre entre 3 y 7 y que el retardo sea 1, valores empíricos recomendados (Bandt & Pompe 2002). Se consideran un factor de retardo τ , cuyo valor indica la cantidad de valores de la serie temporal que son salteados para formar cada uno de los vectores de embedding. Cada uno de estos vectores representa un patrón (o forma). Existen $n!$ posibles patrones. En la sección "*2.2.3 Ejemplo de cálculo de EP unidimensional*", se observa cómo los patrones van apareciendo naturalmente en una señal. Para una secuencia lo suficientemente grande en comparación con $n!$, es posible calcular las frecuencias de ocurrencia de cualquiera de los $n!$ posibles vectores de permutación. A partir de estas frecuencias, se puede estimar la entropía de Shannon asociada con las distribuciones de probabilidad de los vectores de permutación. Algunos de los posibles patrones pueden no ocurrir. A estos se les llama "patrones prohibidos". Existen muchos estudios en los cuales la aparición o no de dichos patrones dan

información sobre la dinámica del sistema. Entonces, para cada π determinamos la frecuencia relativa (# significa el número), esta frecuencia determina un histograma de permutación $p(\pi)$:

$$p(\pi) = \frac{\#\{t | t \leq T - n, (x_{t+1}, \dots, x_{t+n}) \text{ sea tipo } \pi\}}{T - n + 1}$$

Ecuación 8: frecuencia relativa de los $d!$ patrones de orden. Se usa retardo recomendado $\tau=1$

La EP de orden $n \geq 2$ se define con Shannon:

$$H_n = EP = - \sum p(\pi) \log p(\pi)$$

Ecuación 9: Entropía de Permutación. Entropía de Shannon de la frecuencia relativa de los patrones de orden

Esta es la información contenida en la comparación de los n valores consecutivos de la serie temporal $x(t)$. Luego, es claro que $0 \leq H_n \leq \log n!$ donde el límite inferior se alcanza para una secuencia de aumento o disminución de la secuencia de valores, y el límite superior para un sistema completamente al azar (secuencia i.i.d.) donde todas las $n!$ posibles permutaciones aparecen con la misma probabilidad.

También se puede definir la entropía de permutación por símbolo de orden n como: $h_n = \frac{H(n)}{n-1}$ con $n \geq 2$, ya que las comparaciones comienzan a partir de la comparación de al menos dos valores de la secuencia temporal.

Para enunciar algunas de las características y ventajas de este método, se menciona que en contraste con todos los parámetros de complejidad conocidos, un pequeño ruido no cambia esencialmente la complejidad de una señal caótica. EP se puede calcular para series arbitrarias de tiempo del mundo real y además, puesto que el método es muy rápido y robusto, es preferible a otros, en aquellos casos donde los conjuntos de datos son grandes y no hay tiempo para el pre-procesamiento y ajuste fino de parámetros. En el intervalo de varios miles de valores de parámetros, EP es muy similar a exponentes de Lyapunov, mejorando los resultados de este en la presencia de ruido de observación y dinámica.

Distintas características de la serie temporal surgen de estos cálculos, entre ellas se pueden destacar:

- 1) Si los patrones de orden de todos los vectores de embedding es el mismo, la EP será cero. Caso donde los valores del histograma se concentra en un solo bin.
- 2) Si los patrones de orden de todos los vectores de embedding tienen una distribución uniforme, la EP será máxima, igual a uno. Caso donde el histograma es uniforme, con la misma cantidad de valores para cada bin.
- 3) El valor de $0 \leq EP \leq \log_2 n!$ indica la cantidad de información de la serie analizada, y representa la cantidad de bits con la que se puede representar el máximo orden en bits.

- 4) El valor de EP normalizado, siempre estará entre 0 y 1, nos indica el nivel de energía de la serie.

2.2.3 Ejemplo de cálculo de EP unidimensional

Para comprender el método, se expone un ejemplo donde se calcula la EP de la serie temporal:

$$x(t) = [4; 7; 9; 10; 6; 11; 3]$$

En primera instancia se procesan los datos de aun un conjunto de n valores distribuidos en un vector de embedding, cuyos valores se asignan mediante un muestreo de a valores de la serie temporal. En este caso se elige $n = 3$ y $\tau = 1$. Los vectores de embedding (V_e) que se desprenden de esta serie temporal y de los parámetros elegidos resultan:

$x(t)$	4	7	9	10	6	11	3
V_{e1}	4	7	9				
V_{e2}			7	9	10		
V_{e3}			9 10 6				
V_{e4}			10 6 11				
V_{e5}			6 11 3				

Ilustración 5: Vectores de embedding de la serie temporal $x(t)$ con $n=3$ y $\tau=1$

Cada uno de estos cinco vectores V_e presenta un patrón de orden (PO) que depende de la relación entre sus tres elementos. Al ser un ejemplo de solo tres valores por vector, podemos asignar a cada uno de los elementos la característica de Menor, Medio o Mayor dentro de la comparación con sus vecinos. En este ejemplo existen $3!$ patrones sin repetición posibles, cada uno de los cuales guarda una partición dentro del histograma de permutación.

V_{e1}	4	7	9	PO_6	Menor	Medio	Mayor
V_{e2}	7	9	10	PO_6	Menor	Medio	Mayor
V_{e3}	9	10	6	PO_3	Medio	Mayor	Menor
V_{e4}	10	6	11	PO_4	Medio	Menor	Mayor
V_{e5}	6	11	3	PO_3	Medio	Mayor	Menor

Ilustración 6: Patrón de Orden de cada vector de embedding

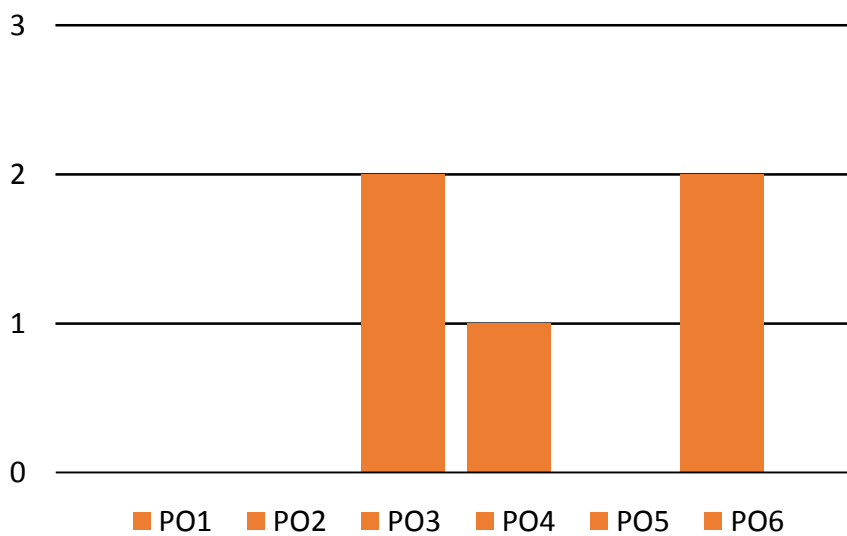


Ilustración 7: histograma de permutación

El histograma de permutación $p(\pi)$ resultante de este ejemplo se puede ver en la Ilustración 7. La EP se calcula a través de la entropía de Shannon del histograma de permutación normalizado, luego se obtiene la energía de orden de la serie según EP normalizado:

$$p(\pi)_{\text{normalizado}} = \frac{[PO_1 \ PO_2 \ PO_3 \ PO_4 \ PO_5 \ PO_6]}{\text{Longitud serie} - n - 1} = \frac{[0 \ 0 \ 2 \ 1 \ 0 \ 2]}{5} = [0, 0, 0.4, 0.2, 0, 0.4]$$

Ecuación 10: Normalización del histograma según el total de vectores de embedding extraídos de la serie temporal

$$EP = H_{(3)} = -[0.4 * \log_2(0.4) + 0.2 * \log_2(0.2) + 0.4 * \log_2(0.4)] = 1,522$$

Ecuación 11: Aplicación de entropía de Shannon. Entropía de Permutación

$$\text{Energía} = \frac{H(n)}{\log_2(n!)} = \frac{1,522}{\log_2(3!)} = 0,589$$

Ecuación 12: Energía de orden de la serie

De estos resultados se deduce que la cantidad de bits en la que se puede representar la información es 2 y que la serie presenta un orden dentro de todo alto, ya que el valor de energía resulto alejado de 1. Esto último se observa en el orden resultante de los valores de los vectores de embedding, ya que de cinco vectores totales, solo aparecen tres patrones de orden diferentes de seis posibles.

2.2.4 Variaciones del algoritmo EP

Según el campo de aplicación, el cálculo de *EP* sobre una serie de datos puede requerir modificaciones en su modo de cálculo, sin perder el concepto del parámetro como un indicador del orden relativo a sus vecinos de la serie analizada. Así se obtienen resultados acordes a los que se pretende deducir con el descriptor. Además, *EP* posee algunas limitaciones como la incapacidad de diferenciar entre patrones de orden generados por distintos valores de componentes. En esa sección se muestran las variaciones encontradas en distintas bibliografías y aplicaciones que buscan una interpretación del concepto desarrollado por Bondt & Pompe que se adapte a sus necesidades dependiendo de la naturaleza y características de las series analizadas.

Entropía de permutación ponderada: En el cálculo por definición de *EP* no se conserva ninguna información además de la estructura de orden al extraer sólo patrones ordinales. Esto puede conducir a inexactitudes en su cálculo, ya que la información de amplitud de los datos analizados podría perderse cuando se extrae sólo la estructura de orden de la series de tiempo. Existen patrones de orden cuya información de amplitud es diferente, pero sin embargo guardan el mismo valor de *EP*. Para el ejemplo de vectores de embedding de tres elementos ($n = 3$), se muestran posibles situaciones donde se evidencia la pérdida de información de amplitud relativa entre los valores de la serie.

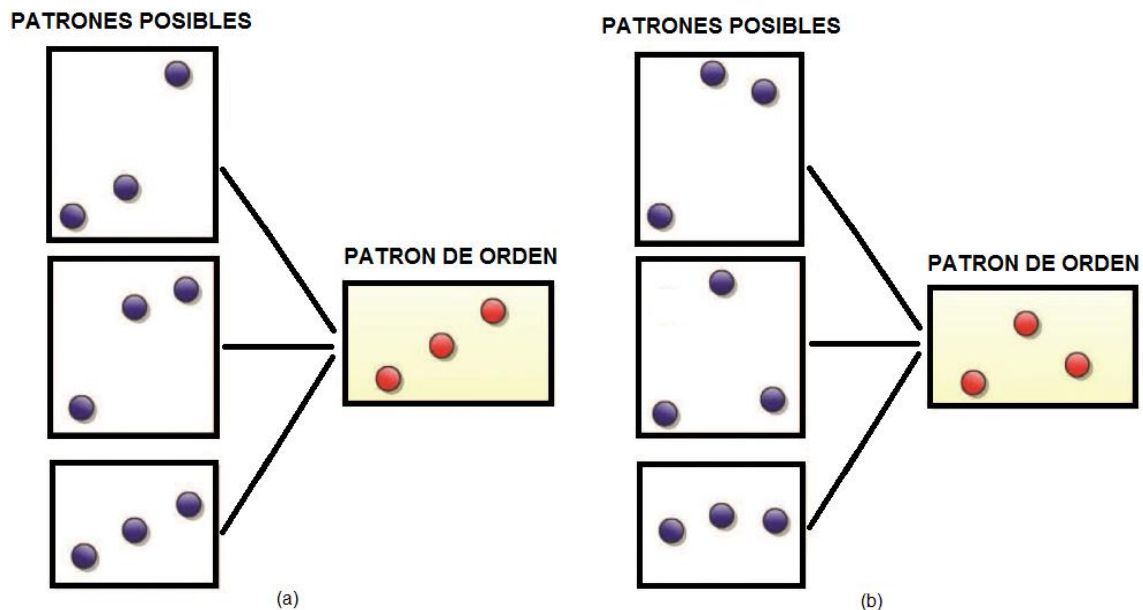


Ilustración 8: Pérdida de información de amplitud en el cálculo de EP. Ejemplo para $n=3$. En la imagen (a) se ven 3 posibles distribución de valores para un único patrón de orden: [Menor Medio Mayor]. En la imagen (b) se muestra otro ejemplo de posibles valores relativos entre los datos del vector de embedding, donde el patrón de orden resultante solo es uno: [Menor Mayor Medio], sin importar las diferencias relativas entre los datos de la serie.

Debido a estas limitaciones, se propuso una variación simple de EP , donde se tiene en cuenta que los patrones de orden pueden contener valores de amplitud muy dispares, por lo que sus varianzas serán grandes. Dicho método se denomina Entropía de Permutación Ponderada (EPP) (Fadlallah et al. 2013) y procede asignando pesos para cada vector de embedding extraído de la serie, con el que luego se calcula la frecuencia relativa asociada a cada parámetro de orden. Se muestra la metodología de cálculo según la siguiente serie de pasos:

Paso 1: Se calcula la varianza o la energía de cada vector de embedding tomado de la serie, expresado como la media aritmética \bar{X}_j^n , entre los valores del vector:

$$\bar{X}_j^n = \frac{1}{n} \sum_{k=1}^n x_{j+k-1}$$

Ecuación 13: Media aritmética de los datos de cada vector embedding

Paso 2: Se calculan los valores de peso de cada vector w_j , basados en las varianzas obtenida en la Ecuación 13.

$$w_j = \frac{1}{n} \sum_{k=1}^n (x_{j+k-1} - \bar{x}_j^n)^2$$

Ecuación 14: Pesos de cada frecuencia relativa

Paso 3: Se calcula la frecuencia relativa con la adición de la información de amplitud entregada por los pesos w_j :

$$p_w(\pi) = \frac{\#\{w_j | w_j \leq T - n, (x_{t+1}, \dots, x_{t+n}) \text{ sea tipo } \pi\}}{\sum w_j}$$

Ecuación 15: Frecuencia relativa ponderada

Paso 4: Utilizando la frecuencia relativa ponderada, se extiende el concepto de EP manteniendo la expresión de entropía de Shannon de la Ecuación 9, con la adición de información de amplitud:

$$H_{w(n)} = EPP = - \sum p_w(\pi) \log p_w(\pi)$$

Ecuación 16: Entropía de Permutación Ponderada.

Entropía de permutación/Entropía de permutación ponderada multiescala: Tanto EP como WEP , se limitan a evaluar los valores de entropía para sólo una escala temporal, la asociada con el muestreo original de la señal a analizar. Esto puede limitar la capacidad de estos parámetros para inspeccionar la dinámica que reside en escalas temporales más largas. Para cubrir esta limitación, se propone la entropía de permutación multiescala (EPM) y la entropía de permutación ponderada multiescala ($EPPM$), en las que se calculan las entropías respectivas en una gama de escalas para evaluar la complejidad de una serie de tiempo.

La evaluación multiescalar tiene ventajas como conocer la dinámica a lo largo de más de una escala temporal, ventaja importante para por ejemplo sistemas biológicos (Azami & Escudero

2016). Además presenta mejor respuesta ante el ruido. Este método utiliza un procedimiento similar al submuestreo de la serie, por consecuencia debe ser utilizado en el campo de series de tiempo extensas, ya que no provee un análisis confiable para series de tiempo cortas.

El procedimiento de multiescalar, se realiza sobre la señal al aplicarle el proceso denominado “coarse-graining” (granulado grueso), el que consiste en producir múltiples y sucesivas versiones de la serie, haciendo el promedio de los datos que aparecen dentro de una ventana de tiempo de longitud τ que se llama factor de escala y que no se superpone a medida que recorre la serie temporal analizada. De aquí resulta una serie de tiempo de granulado grueso $y_j^{(\tau)}$ para cada valor de τ , la que se define en la Ecuación 17, considerando que la serie de tiempo analizada es $\{x_t\}_{t=1,\dots,T}$.

$$y_j^{(\tau)} = \frac{1}{\tau} \sum_{i=(j+1)\tau+1}^{j\cdot\tau} x_i \quad ; \quad 1 \leq j \leq \frac{T}{\tau}$$

Ecuación 17: Serie de tiempo de granulado grueso dependiente del factor de escala τ

Considerando a $y_j^{(\tau)}$ la nueva serie temporal a analizar y se aplica sobre la misma los algoritmos de cálculo de EP y EPP descritos anteriormente para obtener los descriptores multiescales EPM y $EPPM$.

Entropía de permutación compuesta: La EP tiene dos parámetros predefinidos, el orden n y el retraso o retardo τ . Se considera que este último, proporciona el EP con diferentes características de frecuencia. EPC fue desarrollado en “Permutation entropy of the electroencephalogram: a measure of anaesthetic drug effect” (Olofsen et al. 2008) para aprovechar las ventajas de combinar las características de dos bandas de frecuencia, mediante la utilización de diferentes valores de retraso. Además, se agrega un nivel umbral th para evitar las pequeñas fluctuaciones en los valores de la serie debido a diferentes tipos de ruido en la medición. Si la diferencia entre dos puntos de una secuencia es menor que el umbral, entonces la secuencia se contabiliza como un nuevo patrón de orden $(PO_{n!+1})$, resultando un total de $n! + 1$ posibles patrones de orden en vez de $n!$. Por lo tanto, EPC es una familia de EP que se define por tres parámetros: n, τ, th .

$$EPC_n = \frac{(\sum p_i \times \log p_i)_{\tau=1, th<0.5} + (\sum p_i \times \log p_i)_{\tau=2, th<0.5}}{\log(n! + 1)^2}$$

Ecuación 18: Cálculo de CEPI para dos bandas de frecuencias, $t=1$ y $t=2$

Entropía de permutación de Renyi: Hasta el momento se ha presentado la entropía de permutación basada en la entropía de Shannon, ya que éste es el ejemplo más paradigmático utilizado para cuantificar desorden e incertidumbre en sistemas dinámicos. Sin embargo, hay otros tipos de medidas de EP , cada una de las cuales tiene sus particularidades. Una de ellas es

la Entropía de Permutación de Renyi (EPR) de orden α , la que se basa en la entropía de Renyi aplicada a la distribución de probabilidad de permutación o sea la frecuencia relativa de los parámetros de orden ($p(\pi)$).

$$EPR_n = \frac{1}{1 - \alpha} \log \sum_{i=1}^{n!} p(\pi)_i^\alpha$$

Ecuación 19: Entropía de Permutación de Renyi de orden α

El orden α ($\alpha \geq 0$ y $\alpha \neq 1$) es el parámetro de sesgo, para acontecimientos raros $\alpha < 1$, mientras que para acontecimientos destacados $\alpha > 1$. Para la particularidad de $\alpha \rightarrow 1$ se recupera la entropía de Shannon. En el límite de $\alpha \rightarrow \infty$, se puede demostrar que EPR es solo función de la probabilidad más alta, y converge a la Entropía de Permutación mínima (EPR_{min}).

$$EPR_{min} = -\log(\max_{i=1, \dots, n!} p(\pi)_i)$$

Ecuación 20: Entropía de Permutación Mínima - Permutation min-entropy

El análisis mediante EPR permite una mejor caracterización del proceso que el caso particular de análisis con Shannon. Es de variada aplicación de EPR_{min} en el ámbito financiero, para la aplicación de métodos econofísicos en finanzas, al examinar series temporales de datos financieros como eficiencia informacional y dependencia a largo plazo (Zunino & et al. 2016). Este parámetro ha sido experimentado en otras áreas recientemente, como en el monitoreo de señales biomédicas (Liang et al. 2015).

Entropía de permutación de Tsallis: Éste es otro tipo de medida de EP como la vista recientemente basada en la entropía de Renyi. La entropía de Tsallis normalmente también se conoce como q -entropía, ya que se encuentra caracterizada por el índice q . Es la forma matemáticamente más apropiada de calcular la entropía de sistemas que presentan correlaciones de largo alcance, con memoria o con propiedades fractales (Zunino et al. 2008). La Entropía de Permutación de Tsallis (EPT) se basa en el cálculo de la entropía de Tsallis aplicada a la frecuencia relativa de los parámetros de orden ($p(\pi)$) obtenida mediante el método de Bandt & Pompe.

$$EPT_n = \frac{1}{q - 1} \sum_{i=1}^{n!} [p(\pi)_i - (p(\pi)_i)^q]$$

Ecuación 21: Entropía de Permutación de Tsallis de orden q

El orden q ($q \geq 0$ y $q \neq 1$) es el parámetro de sesgo, para acontecimientos raros $q < 1$, mientras que para acontecimientos destacados $q > 1$. Para la particularidad de $q \rightarrow 1$ se recupera la entropía de Shannon. Los resultados obtenidos dependen del valor del índice entrópico q , y se espera que para cada caso específico el valor de q utilizado sea el apropiado, lo que se considera un parámetro abierto.

En el trabajo *“Fractional Brownian motion, fractional Gaussian noise, and Tsallis permutation entropy”* (Zunino et al. 2008), sus autores demuestran que *EPT* se comporta como una poderosa herramienta para caracterizar procesos estocásticos fractales, permitiendo una mejor discriminación de los procesos que la contraparte de Shannon, mientras se utilicen rangos adecuados de sesgo (q).

2.2.5 Aplicaciones actuales de entropía de permutación en señales temporales

Se exponen principalmente datos de la revisión bibliográfica *“Permutation Entropy and Its Main Biomedical and Econophysics Applications”* (Zanin et al. 2012), donde se analizan los fundamentos teóricos de la entropía de permutación, y sus principales aplicaciones para:

- Distinguir el ruido del caos
- Determinar la Complejidad estadística y el plano de complejidad-entropía
- Identificación de las escalas de tiempo
- Dependencias entre series de tiempo

Además se realiza una búsqueda de los trabajos recientes más influyentes y novedosos. Para demostrar el amplio espectro de aplicación del parámetro EP y su eclosión en los últimos tiempos, a continuación se muestran algunas de las aplicaciones de EP al análisis de Mercados Económicos, al conocimiento de Sistemas Biomédicos, y a la Actividad Industrial. Existe una extensa cantidad de aplicaciones dirigida al ámbito también a otras áreas, como al de la Física (Física óptica, Geofísica, Biofísica, Econofísica, etc), siendo este el ámbito de procedencia del descriptor.

ÁREA DE INTERÉS	APLICACIONES	TRABAJOS REALIZADOS
Biomedicina	Análisis de Epilepsia y Alzheimer	<p>-<i>“A hybrid automated detection of epileptic seizures in EEG records”</i> (Tawfik et al. 2016). El sistema propuesto utiliza medidas basadas en entropía de permutación ponderada para los segmentos EEG durante la convulsión epiléptica. Los resultados mostraron un desempeño sobresaliente y revelaron resultados prometedores en términos de discriminación de segmentos de convulsiones y segmentos libres de ataques. También manifiesta una alta robustez contra las fuentes de ruido.</p> <p>-<i>“Permutation entropy of scalp EEG: A tool to investigate epilepsies Suggestions from absence epilepsies ”</i> (Ferlazzo et al. 2014). Se evaluó la entropía de permutación (EP)</p>

	<p>extraída de diferentes electrodos en pacientes con ausencias típicas (AT) y sujetos sanos. En los pacientes con AT se observó un comportamiento recurrente de la topografía del EP, con electrodos de regiones anteriores asociados a mayor aleatoriedad y electrodos de regiones posteriores asociados a menor aleatoriedad. EP parece una herramienta útil para revelar anomalías de la actividad cerebral eléctrica no revelada por el EEG convencional.</p> <p>-“Multivariate Multi-Scale Permutation Entropy for Complexity Analysis of Alzheimer’s Disease EEG” (Morabito et al. 2012). Se propone una metodología multi-escala multivariable original para evaluar la complejidad de las señales fisiológicas. La medida de complejidad básica se realiza mediante la utilización de la entropía de permutación, Dado que las series de tiempo derivadas de sistemas biológicos muestran estructuras en múltiples escalas espacio-temporales, la técnica propuesta puede ser útil para otros tipos de análisis de señales biomédicas. En este trabajo, se verifica la posibilidad de distinguir entre los estados cerebrales relacionados con pacientes con enfermedad de Alzheimer y sujetos con alteraciones cognitivas leves de ancianos sanos normales en una base de datos experimental real, aunque bastante limitada.</p> <p>-“Fast and Robust Detection of Epilepsy in Noisy BEG Signals Using Permutation Entropy ” (Veisi et al. 2007). En este trabajo se investiga la capacidad de PE para caracterizar las señales EEG normales y epilépticas. La clasificación se realiza mediante análisis discriminante. También se estudia el efecto del ruido gaussiano aditivo en el comportamiento de discriminación y se sugieren algunos parámetros derivados de PE para mejorar la precisión de clasificación cuando la señal está contaminada con ruido. Los resultados indican que las medidas propuestas pueden distinguir señales EEG normales y epilépticas con una precisión de más del 97% para el EEG limpio y más del 85% para las señales EEG con alto contenido de ruido.</p>
<p>Estudio de Efecto de la Anestesia</p>	<p>-“Depth of anaesthesia assessment based on adult electroencephalograph beta frequency band” (Li & Wen 2016). Este artículo presenta un nuevo método para aplicar las características de tiempo de electroencefalografía (EEG) beta bandas de frecuencia para evaluar la profundidad de la anestesia (DoA). Se calculan la movilidad (M), la entropía de permutación (EP) y la complejidad Lempel-Ziv (LCZ) de las bandas de frecuencias beta (21,5-30 Hz), se seleccionan para derivar un nuevo índice. El nuevo índice se evalúa y compara con el bispectral medido (BIS). Los resultados muestran que existe una estrecha correlación entre el índice propuesto y el BIS durante diferentes estados</p>

anestésicos. El nuevo índice también muestra una respuesta de 25-264 segundos antes que el BIS durante el período transitorio de los estados anestésicos. Además, el índice propuesto es capaz de evaluar continuamente el DoA cuando la calidad de la señal es pobre y el BIS no tiene salidas válidas.

-*"Multiscale permutation entropy analysis of EEG recordings during sevoflurane anesthesia"* (Li et al. 2010) En este trabajo, se propuso una nueva medida que combinaba la información de EP de múltiples escalas, llamada CMSPE (entropía de permutación multi-escala compuesta), para cuantificar el efecto del fármaco anestésico en las grabaciones de EEG durante la anestesia con sevoflurano. Se encontró que el PE crudo de una sola escala era ciego a transiciones sutiles entre la luz y la anestesia profunda, mientras que el índice CMSEP rastreó estos cambios con precisión. Alrededor del tiempo de pérdida de conciencia, el CMSPE respondió significativamente más rápidamente que el EP crudo.

-*"Permutation entropy of the electroencephalogram: a measure of anaesthetic drug effect"* (Olofsen et al. 2008). Sería útil disponer de un índice electroencefalográfico de código abierto (EEG) del efecto anestésico del ácido g-amino-butírico (GABA), que responda rápidamente a los cambios en el patrón EEG y pueda vincularse a los mecanismos neurofisiológicos y neurofarmacológicos subyacentes que controlan el estado consciente. Se evaluó la respuesta del PE a diversas formas de onda de tipo EEG generadas por ordenador. Se desarrolló un índice de EP compuesto (CPEI), que era la suma de dos EP simples e incluía un pequeño umbral de medición-ruido. Se concluyó que EP del EEG muestra una promesa como una simple medida del efecto analgésico del GABAérgico.

Estudios en
Neurociencia
Cognitiva

-*" A new methodology for automated diagnosis of mild cognitive impairment (MCI) using magnetoencephalography (MEG)"* (Amezquita-Sanchez et al. 2016). El deterioro cognitivo leve (MCI) es un trastorno cognitivo caracterizado por una pérdida de memoria, mayor de lo esperado por la edad. Se presenta una nueva metodología para identificar a los pacientes MCI durante una tarea de memoria de trabajo utilizando señales MEG. Se utiliza la descomposición completa del modo empírico conjunto (CEEMD) para descomponer la señal MEG en un conjunto de subbandas. La metodología utiliza una medida de dinámica no lineal basada en el análisis de entropía de permutación (EP) para analizar las subbandas y detectar características que se utilizarán para la detección de MCI. Se demostró que el método CEEMD integrado con PE

análisis proporciona una herramienta eficaz para distinguir entre normal y MCI con una alta precisión del 98,4%.

-*“Weighted-Permutation Entropy as Complexity Measure for Electroencephalographic Time Series of Different Physiological States ”* (Vuong et al. 2015). Este artículo presenta un enfoque mediante la incorporación de WPE (entropía de permutación ponderada) en el análisis de diferentes estados fisiológicos a saber EEG serie temporal. Se incluyeron tres estados fisiológicos del EEG, ojo cerrado (EC), ojo abierto (EO) y tarea extraña visual (VOT) para examinar la capacidad de WPE para identificar y discriminar diferentes estados fisiológicos. La clasificación por WPE ha logrado los resultados con una precisión del 87% entre los estados EC y EO, y 83% entre EO y VOT, respectivamente, usando análisis de discriminación lineal. Los resultados mostraron el potencial de WPE para ser una característica prometedora para el análisis no lineal en diferentes estados fisiológicos del cerebro.

-*“Permutation Entropy: a new feature for Brain-Computer Interfaces”* (Nicolaou & Georgiou 2010). Aquí, el PE se utiliza para caracterizar las señales de electroencefalogramas de 3 sujetos realizando 4 tareas de imágenes motoras, que luego se clasifican utilizando una máquina vectorial de apoyo. A pesar de que es posible obtener una precisión del 100% en el ensayo único, esto depende mucho del sujeto.

Análisis del Ritmo Cardíaco

-*“ Automated detection and localization of myocardial infarction using electrocardiogram: a comparative study of different leads”* (Acharya et al. 2016). Este trabajo propone un nuevo método de detección y localización automatizadas de infarto de miocardio (MI) mediante el análisis de señales ECG. La señal de ECG obtenida a partir de 12 derivaciones de ECG se somete a transformada de onda corta (DWT) hasta cuatro niveles de descomposición. Se extraen 12 características no lineales de la DWT, una de ellas es EP. El método propuesto ha logrado la mayor exactitud media de 98,80%, sensibilidad del 99,45% y especificidad del 96,27% en la clasificación normal y MI.

-*“Application of the Permutation Entropy over the Heart Rate Variability for the Improvement of Electrocardiogram-based Sleep Breathing Pause Detection”* (Ravelo et al. 2015). Se analiza la entropía de permutación (EP) obtenida a partir de la variabilidad de la frecuencia cardíaca (VFC) en un modelo estadístico. También se integran otras técnicas de extracción de características. El objetivo del modelo es la detección de apnea obstructiva del sueño (SAOS). Para probar los modelos se usan 70 grabaciones de ECG. Las

		<p>pruebas muestran que la contribución de la entropía de permutación aumenta el rendimiento de los clasificadores, lo que implica que la complejidad de las series de tiempo desempeña un papel importante en la detección de pausas respiratorias.</p> <p>-<i>“Classifying cardiac biosignals using ordinal pattern statistics and symbolic dynamics”</i> (Parlitz et al. 2012). El desempeño de la clasificación de (bio) señales depende fuertemente de la elección de características adecuadas (también llamadas parámetros o biomarcadores). En este artículo se evalúa el poder discriminativo de las estadísticas de los patrones ordinal (EP) y la dinámica simbólica en comparación con los parámetros establecidos de la variabilidad de la frecuencia cardíaca aplicados a los intervalos beat-to-beat. Estas comparaciones muestran que los patrones ordinal muestreados con un retraso de tiempo adicional son características prometedoras para una clasificación eficiente.</p> <p>-<i>“Permutation entropy improves fetal behavioural state classification based on heart rate analysis from biomagnetic recordings in near term fetuses ”</i> (Frank et al. 2006). La relevancia de la complejidad de las fluctuaciones de la frecuencia cardíaca fetal con respecto a la clasificación de los estados del comportamiento fetal no ha sido aclarada satisfactoriamente hasta ahora. Debido a los estados de comportamiento cortos, la entropía de permutación proporciona una estimación ventajosa de la complejidad que conduce a la entropía de Kullback-Leibler (KLE).</p>
<p>Mercados económicos</p>	<p>Eficiencia del Mercado financiero</p>	<p>-<i>“Monitoring the informational efficiency of European corporate bond markets with dynamical permutation min-entropy”</i> (Zunino & et al. 2016). En este trabajo se ha implementado la min-entropía de permutación para desvelar la presencia de estructuras temporales en los valores diarios de los índices de bonos corporativos europeos desde abril de 2001 hasta agosto de 2015. Más precisamente, la evolución de la eficiencia informativa de los precios de quince índices sectoriales se han estudiado mediante la estimación de esta herramienta simbólica derivada de la teoría de la información sobre una ventana de tiempo de deslizamiento. Este análisis dinámico permite obtener conclusiones relevantes sobre el efecto que la crisis crediticia de 2008 ha tenido en los distintos sectores europeos de bonos corporativos.</p> <p>-<i>“On the efficiency of sovereign bond markets”</i> (Zunino et al. 2012). En este trabajo se estudia la eficiencia del mercado soberano de treinta índices de bonos de países desarrollados y emergentes, utilizando una herramienta</p>

		<p>estadística innovadora en la literatura financiera: el plano de causalidad complejidad-entropía. Este espacio de representación nos permite establecer una clasificación de eficiencia de diferentes mercados y distinguir diferentes dinámicas del mercado de bonos. Consideramos que la clasificación derivada del plano de causalidad complejidad-entropía es consistente con las calificaciones asignadas por las principales compañías de calificación a los instrumentos soberanos. Además, encontramos una correlación entre la entropía de permutación, el desarrollo económico y el tamaño del mercado que podría ser de interés para los responsables políticos y los inversores.</p> <p><i>-“A New Approach to Measure Volatility in Energy Markets”</i> (Ruiz et al. 2012). Este trabajo sugiere utilizar la entropía de permutación, la entropía topológica y la entropía de permutación modificada como alternativas para medir la volatilidad en los mercados de energía. Los datos simulados muestran que estas medidas son más apropiadas para cuantificar la incertidumbre asociada a una serie temporal que las basadas en la desviación estándar u otras medidas de dispersión.</p> <p><i>-“Forbidden patterns, permutation entropy and stock market inefficiency”</i> (Zunino et al. 2009). En este trabajo introducimos dos nuevos cuantificadores para la ineficiencia del mercado de valores: el número de patrones prohibidos y la entropía de permutación. Hay evidencia sólida de que el grado de ineficiencia del mercado está positivamente correlacionado con el número de patrones prohibidos y negativamente correlacionado con la entropía de permutación. Nuestros resultados empíricos sugieren que estas dos herramientas físicas son útiles para discriminar la etapa de desarrollo del mercado de valores y pueden ser fácilmente implementadas.</p>
<p>Industria</p>	<p>Diagnóstico de fallas</p>	<p><i>-“Improved LMD, Permutation Entropy and Optimized K-Means to Fault Diagnosis for Roller Bearings”</i> (Shi et al. 2016). En este artículo se presenta un nuevo método de extracción y reconocimiento de características de fallas de señales de vibración de cojinetes, basado en la descomposición de media local mejorada (LMD), la entropía de permutación (EP) y el algoritmo optimizado de clustering de K-means. Los resultados del experimento muestran que el método propuesto es efectivamente la extracción de fallas y el reconocimiento de rodamientos de rodillos.</p> <p><i>-“A novel bearing fault diagnosis model integrated permutation entropy, ensemble empirical mode decomposition and optimized SVM”</i> (Zhang et al. 2015). En este trabajo se presenta un nuevo modelo híbrido para la</p>

	<p>detección de fallas y clasificación de cojinetes de motores. En el modelo propuesto, la entropía de permutación (PE) de la señal de vibración se calcula para detectar los fallos del cojinete. El modelo propuesto se evalúa completamente mediante experimentos y estudios comparativos. Los resultados demuestran su efectividad y robustez para la detección y clasificación de fallas.</p> <p><i>-"Self-adaptive bearing fault diagnosis based on permutation entropy and manifold-based dynamic time warping"</i> (Tian et al. 2015). Para hacer más sistemático y efectivo el diagnóstico de fallas de los cojinetes con una mejor operatividad y capacidad en tiempo real, este estudio propone un enfoque que utiliza la entropía de permutación y la deformación dinámica del tiempo basada en múltiples colectores, que puede reflejar la complejidad de los datos con buena robustez y capacidad de cómputo rápido, actúa como característica de fallo. Los resultados experimentales demuestran que el enfoque propuesto puede diagnosticar eficazmente las fallas de los cojinetes en condiciones de trabajo variables y diferentes niveles de gravedad de falla.</p>
<p>Reconocimiento de estructuras</p>	<p><i>-"State recognition of viscoelastic sandwich structures based on permutation entropy and generalized Chebyshev support vector machine"</i> (Qu et al. 2015). Las estructuras sándwich viscoelásticas son ampliamente utilizadas en equipos mecánicos. El reconocimiento del estado de estructuras sándwich viscoelásticas es muy necesario para supervisar estados de salud estructurales y mantener el equipo funcionando de manera fiable. Teniendo en cuenta la alta no linealidad de las características dinámicas y la fuerte no estacionariedad de las señales de respuesta de vibración, se propone un nuevo método basado en la entropía de permutación (PE). Los resultados de las pruebas muestran que el método propuesto es efectivo para el reconocimiento del estado de estructuras sándwich viscoelásticas, que tiene una capacidad de generalización más fuerte y puede alcanzar una precisión de reconocimiento más alta que la de la SVM de onda.</p> <p><i>-"Evaluation of speckle-interferometry descriptors to measuring drying-of-coatings"</i> (Blotta et al. 2011). En el proceso industrial de pintura, la pintura-secado es una etapa importante debido a su alto impacto en el resultado final. Su estudio es de relevancia para mejorar las propiedades del revestimiento resultante. La entropía de permutación y los patrones binarios locales evidenciaron curvas de secado similares, mostrando un punto de inflexión notable, coincidente con el paso en el área de secado constante a un estado posterior, definido por una</p>

	<p>difusión más lenta del disolvente a través de la capa seca de la superficie.</p>
<p>Análisis de Estabilidad dinámica en flujos</p>	<p>-<i>“Nonlinear multi-scale dynamic stability of oil–gas–water three-phase flow in vertical upward pipe Lian-Xin”</i> (Zhuang et al. 2016). La caracterización de la estabilidad y la no linealidad subyacente del flujo trifásico aceite-agua-gas es un desafiante problema de importancia significativa. Realizamos experimentos y medimos las señales de fluctuación de un sensor de conductancia de campo eléctrico giratorio con ocho electrodos. Se emplea el plano de causalidad de entropía de complejidad ponderada a escala múltiple (MS-WCECP) para explorar las características no lineales de cinco estructuras de flujo trifásico de aceite-agua-gas típicas. Los resultados demuestran que el plano de causalidad de la entropía de la complejidad ponderada en múltiples escalas puede representar efectivamente las transiciones de las estructuras de flujo trifásico y servir como una herramienta útil para probar la dinámica no lineal de los flujos trifásicos.</p> <p>-<i>“The experimental signals analysis for bubbly oil-in-water flow using multi-scale weighted-permutation entropy”</i> (Chen et al. 2015). Combinamos el método de múltiples escalas (MS) y la entropía de permutación ponderada (WPE) para analizar series de tiempo caótico, ruidoso y fractal, y encontramos que MSWPE puede distinguir diferentes series de tiempo no lineales y mostrar una mayor robustez en presencia de mayores niveles de ruido, una tarea que la entropía de permutación de múltiples escalas (MSPE) no puede realizar. A continuación, aplicamos MSWPE para analizar las señales de los experimentos de flujo bifásico de aceite en agua verticales hacia arriba. Nuestros resultados sugieren que la tasa de cambio de MSWPE permite caracterizar la transición de los patrones de flujo y la entropía de permutación ponderada multi-escala permite indicar la discrepancia de la complejidad del flujo en dos fases de aceite en agua.</p>

Tabla 3: algunos de los trabajos más influyentes o recientes que utilizan el parámetro Entropía de Permutación como descriptor.

2.3 Entropía de Permutación adaptada a imágenes

Como medida de complejidad para series de tiempo, la Entropía de Permutación, se basa en la comparación entre los valores vecinos de una muestra consecutiva de los datos. Esta definición permite contrastar fácilmente el concepto de este descriptor, con una de las varias interpretaciones de la textura de una imagen: *“Arreglo de píxeles cuya relación es la variación espacial de los tonos de grises”* (Pinto 2006). En esta interpretación de textura, se vislumbra claramente el vínculo entre el concepto de textura y la relación entre píxeles vecinos.

El desarrollo de medidas de complejidad para datos en dos o más dimensiones, ha sido objetivo de estudio de muchos años. La adaptación a dos o más dimensiones de muchos de estos descriptores de complejidad de una señal unidimensional, también ha sido de interés, debido a su aplicabilidad en áreas de la física, biología, comunicaciones, etc.

Dentro de las medidas de complejidad de estructuras de patrones para sistemas dinámicos, muchas de las preguntas ya respondidas en sistemas de una dimensión, se encuentran aún abiertas en sistemas de dimensiones más altas, como por ejemplo en el campo de las comunicaciones: *¿Cómo es compartida, almacenada y transmitida la información a través de una grilla de dos dimensiones para producir un set de configuraciones dado?*, o en el ámbito de la física y la biología: *¿Cómo podemos distinguir cuantitativamente entre diferentes tipos de patrones de orden en dos dimensiones?* Algunos trabajos se han basado en responder estas inquietudes y extender la definición de descriptores de complejidad como la entropía, como caracterizador de sistemas en más de una dimensión (Feldman & Crutchfield 2003).

Es el caso de este trabajo, donde se toma el concepto de Entropía de Permutación para series temporales, para ampliar su definición a sistemas de dos dimensiones como son las imágenes. Los trabajos realizados anteriormente a en esta línea de investigación son escasos, y datan de unos pocos años atrás, siendo los más influyentes del año 2012 y el primero en aparecer del año 2006. Estos, fueron algunos de los motivos por los que se decidió abordar este proyecto, la escasa cantidad de trabajos desarrollados al momento de elegir el tema, y por ende la flexibilidad en el desarrollo de los métodos sugeridos para la adaptación del parámetro entropía de permutación a imágenes como caracterizador de texturas.

En esta parte del capítulo Estado del Arte, los proyectos revisados se exponen de una forma distinta a la realizada hasta el momento. Debido a que la cantidad de trabajos hallados fue escasa, se detalla sobre cada uno de ellos por considerarlos relevantes en la posterior explicación y justificación de los métodos elaborados en este proyecto, para utilizar a *EP* como parámetro caracterizador de imágenes de superficies según la rugosidad de sus texturas.

2.3.1 Segmentación de texturas no supervisada utilizando entropía de permutación (Li et al. 2006)

Objetivo del trabajo: utilizar la nueva característica de medida de la complejidad combinada con la media de escala de grises y la desviación de la escala de grises, construyendo el vector de características multidimensionales, al cual se le aplica el algoritmo fuzzy c-means como clasificador para agrupar los vectores de características, y obtener la segmentación de texturas

en una imagen gris. El objetivo es la obtención de EP como descriptor de textura de la imagen. El trabajo justifica su investigación en base a la aseveración de EP como descriptor de textura, y a la complejidad de cálculo y alto costo computacional de los métodos tradicionales para tal fin.

Método utilizado: se resume este trabajo en tres partes para explicar el método adoptado.

1. Se utiliza una variación de EP original, ya que por definición el algoritmo de EP parte del análisis de una señal de valores continuos, de modo que valores consecutivos repetidos son muy raros, por lo que no se tienen en cuenta a las permutaciones con repetición como posible patrón de orden, al afirmar que un vector de embedding con todos o algunos de sus valores iguales resulta improbable. Dicha afirmación resulta desacertada para el caso de señales de valores discretos, como lo son las imágenes digitales. En el trabajo se propone un algoritmo mejorado de EP, que considera los valores consecutivos iguales en las series temporales, para lo que utiliza permutaciones Cayley (Mor & Fraenkel 1984). El desarrollo del algoritmo será el mismo que para EP, sólo que ahora hay más combinaciones posibles al incluir los casos donde haya repetición. Hay q tipos de secuencias posibles, la definición se muestra en la Ecuación 22, donde m es la cantidad de permutaciones posibles, n el largo del vector embedding, $i_0 = 0, i_j > 0, j = 1, 2, \dots, m$ y $\sum_{j=0}^m i_j = n$.

$$q = n! + \sum_{m=1}^{n-1} \left[m! \prod_{l=1}^m \left(n - \sum_{k=0}^{l-1} i_k \right) \right]$$

Ecuación 22: Cantidad de permutaciones posibles con repetición

El valor entregado por q es la nueva cantidad de permutaciones posibles. Se utiliza luego como $\log_2 q$, para la normalización de la entropía de permutación hallada.

2. Los patrones de orden se forman teniendo en cuenta el carácter de dirección de textura. Se recorre la imagen con una ventana de tamaño 4x4 y se seleccionan cuatro direcciones básicas 0°, 45°, 90° y 135° para la extracción de características de entropía de permutación, para analizarlas sintéticamente. Las q distintas clases $d_i(\theta, l)$ con $i = 1, 2 \dots q$, se dividen dependiendo del ángulo de dirección θ entre los píxeles y la distancia l entre los píxeles. Para cada una de las cuatro direcciones se calcula la entropía de permutación $H(\theta, l)$, calculado como:

$$H(\theta, l) = - \sum_{i=1}^q p[d_i(\theta, l)]. \log[p(d_i(\theta, l))]$$

Ecuación 23: entropía de Shannon para cada una las cuatro direcciones

Finalmente, resulta un vector de características de entropía de permutación para las 4-direcciones, en el que cada valor varía entre 0 y $\log(q)$.

3. Por último, se extraen características del nivel de gris entre vecinos, por medio de métodos tradicionales de máscaras, obteniendo la *media del nivel de gris* y la *desviación de niveles de gris*. Se combinan estas dos características junto con el vector de características de entropía de permutación en las cuatro direcciones $H(\theta, l)$, para concluir con un vector de características de textura de la imagen de seis dimensiones. Con la finalidad de segmentación de este trabajo, se aplica el algoritmo de clustering no supervisado fuzzy c-means una vez obtenido el vector de características.

Resultados: el método puede capturar regiones texturizadas de imágenes arbitrarias. Los resultados del experimento muestran que el método tiene un buen desempeño en la segmentación de imágenes con texturas distribuidas uniformemente. En la premisa de mantener una alta precisión en la segmentación de textura, el método no sólo puede reducir la complejidad computacional, sino también es robusto y tiene la capacidad de resistir ruidos. Teniendo en cuenta la combinación de las otras características, los resultados de segmentación para la imagen de textura estructural se puede mejorar.

Observación: analizado el método desarrollado en este trabajo para la implementación de EP en imágenes, se concluye que la complejidad en el tratamiento de la señal para obtener las características de entropía de permutación, no acompañan las virtudes intrínsecas de sencillez y rapidez de cálculo del algoritmo original de EP, además de no demostrarse la invarianza a transformaciones no lineales. La única de las ventajas que se observa explotada en este trabajo, es la robustez del parámetro EP, por considerar que suple la falta de tal ventaja de las características de niveles de gris media y desviación.

2.3.2 Plano Complejidad-Entropía causal como una medida de complejidad para patrones bidimensionales (Ribeiro et al. 2012)

Objetivo del trabajo: reducir el vacío existente en definiciones de medidas de complejidad para dos o más dimensiones, aplicando el concepto de entropía de permutación combinado con un índice entrópico relativo. Se construye un procedimiento numérico que se puede implementar fácilmente para evaluar la complejidad de dos o más patrones dimensionales. Se trabaja con este método en diferentes escenarios, donde se toman en cuenta experimentos numéricos y datos empíricos. Específicamente, se aplica el método a: i) paisajes fractales generados numéricamente donde se compara las medidas desarrolladas con el exponente de Hurst; ii) texturas de cristal líquido donde las transiciones de fase nematic-isotropic-nematic fueron identificadas apropiadamente; iii) 12 texturas características de cristales líquidos donde los diferentes valores muestran que el método puede distinguir diferentes fases; iv) y superficies de Ising donde el método aquí propuesto identifica la temperatura crítica, además de demostrar ser estable.

Método utilizado: En este trabajo se toman los conceptos planteados por Bandt y Pompe para justificar su desarrollo. Uno de los planteos de los autores de EP es que la mayoría de las medidas de complejidad dependen de algoritmos o recetas específicas para procesar los datos que también pueden depender de los parámetros de ajuste. Como consecuencia directa, existen enormes dificultades para reproducir resultados previos sin el conocimiento de los detalles de los métodos. Este trabajo propone implementar el *plano de complejidad-entropía causal*, construido por primera vez en el trabajo “A Statistical Measure of Complexity” (Lopez-Ruiz et al. 1995), extendido para patrones en dimensiones superiores. Específicamente se aplica este enfoque en diferentes escenarios relacionados con estructuras bidimensionales como imágenes.

A diferencia del trabajo expuesto anteriormente, donde se realiza una variación en el algoritmo de cálculo de EP debido a la utilización de datos discretos, en este trabajo se calcula EP sin variaciones y se hace foco en la representación de los pixeles (datos bidimensionales) en un vector unidimensional. El método adoptado se explica en dos partes.

1. En el caso de una imagen, el sistema está representado por un arreglo bidimensional $I(x, y)$ de tamaño $d_x \times d_y$. Se crea un método para representar los datos en un vector unidimensional. En analogía con el denominado vector de embedding definido en el algoritmo de EP, se define una matriz de tamaño $n_x \times n_y$ con $n_x, n_y > 1$ a través de la cual se recorre la imagen y se evalúan las permutaciones existentes. Este nuevo sistema tiene $(n_x \times n_y)!$ estados posibles, ya que no se considera que existan valores repetidos, por usar el método original de EP. Con la frecuencia de aparición de estos estados, se calcula la probabilidad de distribución de orden de la imagen $p(\pi)$. Para facilitar la comprensión de cómo se forman los estados posibles, se ilustra el procedimiento en un ejemplo, adaptado para una pequeña matriz que representa a la imagen $I(x, y)$ de tamaño 3×3 , la que será recorrida con una matriz de embedding de orden 2×2 .

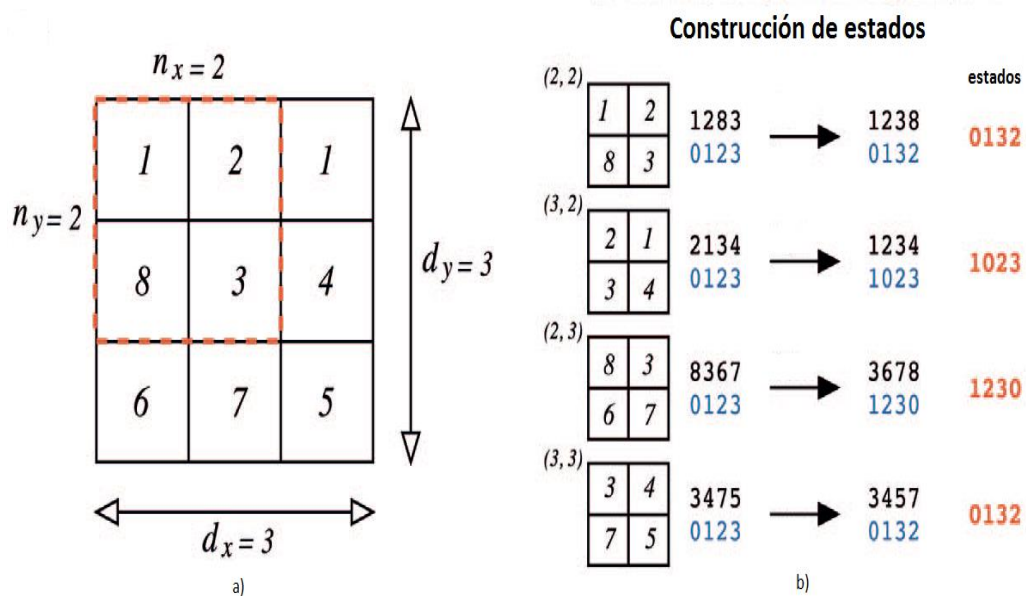


Ilustración 9: las imágenes muestran la construcción de los estados. a) Matriz de 3×3 que representa a la imagen. Es recorrida de manera solapada por una matriz de embedding de 2×2 . b) De cada matriz de embedding se obtiene un vector alineando las filas, una seguida a de la otra. Se asigna un valor de orden a

cada elemento del vector y por último se ordenan los elementos de menor a mayor. El orden en que resultaron modificados los elementos originales es considerado un patrón que indica el orden (estado).

El procedimiento de orden de los elementos de la matriz a un vector, no es único como en el caso unidimensional, ya que los elementos de la matriz de embedding pudieron haber sido también ordenados columna tras columna, o de cualquier otra manera. Sin embargo, estas otras definiciones de orden, solo cambian el “nombre” de los estados, pero no su frecuencia relativa de aparición. Por lo tanto, no hay pérdida de generalización al suponer una receta de orden distinta para definir $p(\pi)$.

2. Con $p(\pi)$ se calcular EP de la forma original, y además se obtiene otro parámetro de desequilibrio C , que cuantifica el grado de estructuras correlacionales que proporcionan información adicional importante que no puede ser obtenida solo por EP . El desequilibrio C se calcula como:

$$C = Q \times EP$$

Ecuación 24 : Desequilibrio que aporta información de los datos que no puede ser obtenida con EP

Donde Q es una métrica entrópica relativa entre la probabilidad empírica ordinaria $p(\pi)$ y el estado equiprobable $p_e = \frac{1}{(d_x d_y)!}$. La cantidad Q es conocida como un tipo de desequilibrio y se define en términos de la divergencia de Jansen-Shannon como: $Q = \frac{S\left[\frac{p(\pi)+p_e}{2}\right] - \frac{S[p(\pi)] + S[p_e]}{2}}{Q_{\max}}$, donde S indica la aplicación de la entropía de Shannon y Q_{\max} se calcula cuando uno de los componentes de $p(\pi)$ es igual a uno u todos los demás desaparecen.

Debido a que C proporciona información importante y adicional a EP y además existe un rango de valores posibles de C para un EP dado, se propone emplear un diagrama de C vs. EP como herramienta diagnóstica, construyendo el plano de complejidad causal-entropía causal, propuesta por primera vez en el trabajo “*Distinguishing Noise from Chaos*” (Rosso & Larrondo 2007). El objetivo de este trabajo tiene como eje calcular el diagrama C vs. EP para medir la complejidad de un sistema bidimensional y distinguir entre diferentes patrones.

Mediante el método resumido anteriormente, en este trabajo se propone aplicar una generalización del plano C vs. EP causales, a patrones de dimensiones superiores, en superficies fractales y texturas de cristal líquido para comprobar la diferenciación de rugosidades entre distintas imágenes, superficies de Ising con el objetivo de identificar transiciones de fase.

En cada caso de aplicación, se compararon diferentes índices según el plano C vs. EP causales. Por ejemplo en el caso de superficies fractales, se obtiene la dependencia del plano complejidad-entropía causales según el exponente de Hurst (h), para distintas dimensiones de la matriz de embedding. Para el caso de imágenes de texturas de cristal líquido y superficies de Ising, se analiza la dependencia del plano C vs. EP causales para diferentes temperaturas T .

Resultados: Se demostró que los índices EP y C se desempeñan muy bien para distinguir entre las diferentes rugosidades de las superficies fractales. Los parámetros identificaron adecuadamente las transiciones de fase de un cristal líquido liotrópico y ordenaron diferentes texturas según su complejidad. Por último, en relación con las superficies de Ising, los índices no sólo identificaron la temperatura crítica, sino que también demostraron ser estables después de 10^4 pasos de Monte Carlo. Los resultados se ilustran en las siguientes ilustraciones:

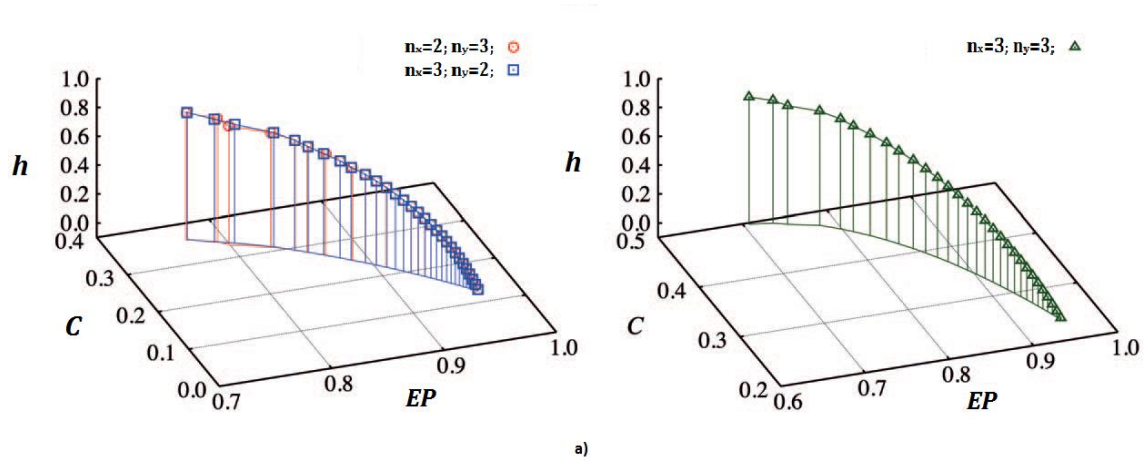


Ilustración 10: Dependencia del coeficiente de Hurts en superficies fractales según el plano C vs. EP para distintas dimensiones de la matriz de embedding.

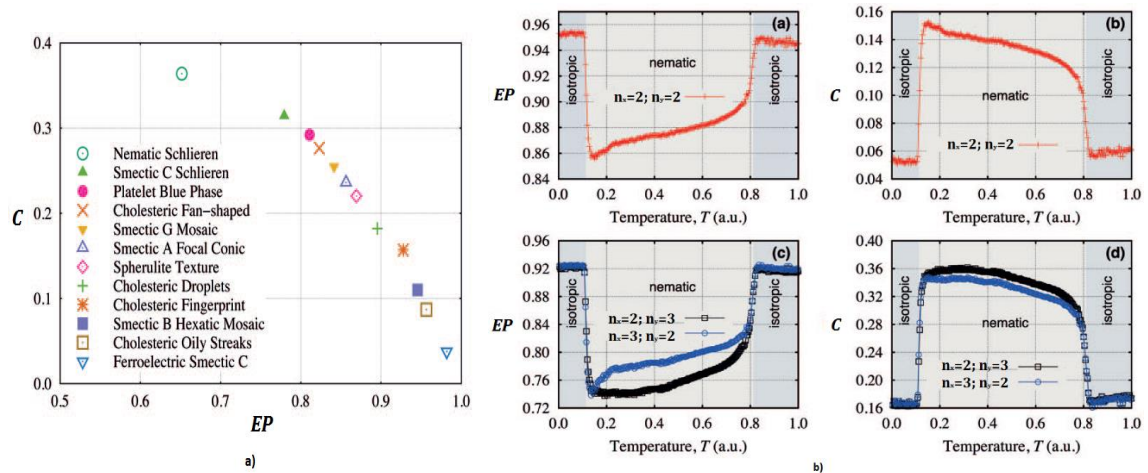


Ilustración 11: se ven los resultados obtenidos para texturas de cristales líquidos. a) Relación entre C y EP para distintas texturas de cristal líquido. b) Comportamiento de EP y C dependiendo de la temperatura en un cristal líquido liotrópico.

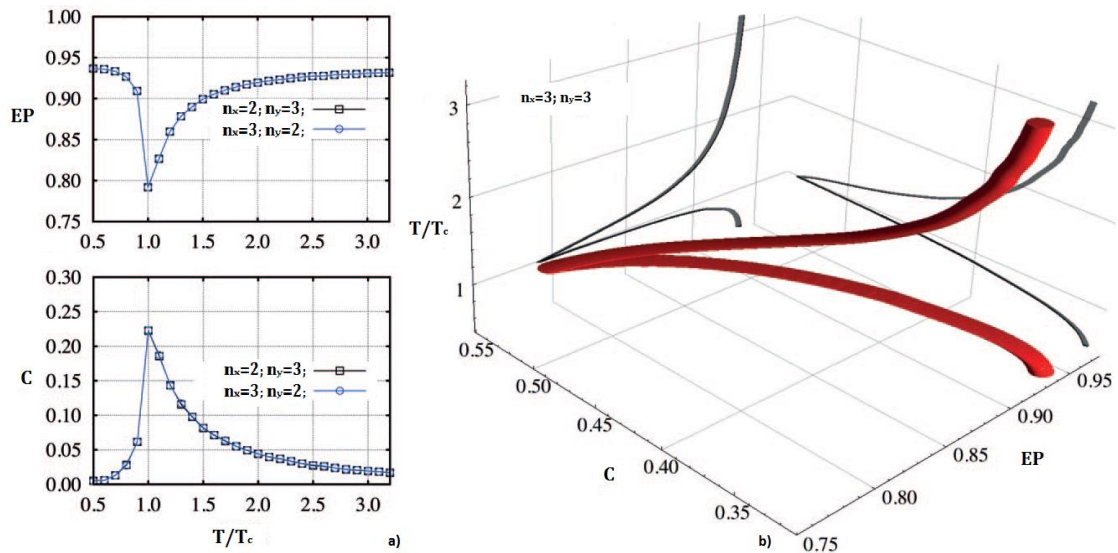


Ilustración 12: Dependencia de EP y C según la temperatura en superficies Ising. a) EP vs T y C vs T. b) Visualización 3D del modelo Ising para transición de fase.

El método también tiene una evaluación numérica muy rápida y sencilla. Teniendo en cuenta todos estos hallazgos, los autores se mostraron muy optimistas, al visualizar su método como una herramienta útil para reducir la brecha entre las medidas de complejidad unidimensional y las de mayor dimensión.

Observación: El método adoptado en este trabajo, aprovecha muchas de las ventajas brindadas por el parámetro EP, ya que realiza un pretratamiento simple de los datos para determinar los estados presentes y no utiliza una variación del algoritmo original de cálculo de EP, acciones que incrementarían el tiempo de ejecución del algoritmo. Sin embargo, el método desarrollado para determinar los estados presentes en la imagen, no tiene en cuenta la disposición espacial del patrón en la imagen, siendo la ubicación espacial una propiedad importante en los datos distribuidos en el espacio bidimensional. El trabajo demuestra la invarianza a transformaciones no lineales del método, otro característico del parámetro EP, al realizar los cálculos con matrices de embedding con rotación de sus dimensiones en ordenadas y abscisas. Se obtienen mismos resultados para $n_x \rightarrow n_y$ y $n_y \rightarrow n_x$.

2.3.3 Discriminación de texturas en imágenes con el plano Complejidad-Entropía causales multiescala bidimensional (Zunino & Ribeiro 2016)

Objetivo del trabajo: explorar aún más la utilidad del plano bidimensional complejidad-entropía causales como un descriptor de imagen de textura, originado en el trabajo anteriormente expuesto, ya que se considera que el mismo presentó resultados prometedores en la detección de patrones en dos dimensiones. Aquí se introduce una generalización multiescalar para distinguir entre distintas características de rugosidad en imágenes de superficies a pequeñas y grandes escalas espaciales. Inicialmente se consideran estructuras bidimensionales generadas numéricamente para ilustrar conceptos básicos en un marco controlado. Luego, se estudian situaciones más realistas. Se busca obtener características intrínsecas a la imagen mediante un

método de aproximación simbólica multiescalar, derivado de técnicas de teoría de la información.

Método utilizado: Se utilizan los conceptos del plano complejidad causal-entropía causal, para la distribución de probabilidad de patrones de orden de permutación en imágenes, definidos en el trabajo anterior. Estos cuantificadores se conocen habitualmente como entropía de permutación y complejidad estadística de permutación y son los que caracterizan la diversidad y la estructura correlacional, de los ordenamientos espaciales presentes en la imagen. El plano de causalidad complejidad-entropía se define como el espacio de representación obtenido mediante el trazado de la complejidad estadística de permutación (ordenadas) frente a la entropía de permutación (abscisas) para un sistema dado.

Como se ha mencionado anteriormente, los retardos determinan la separación espacial entre símbolos, lo que se corresponde a múltiplos de la escala de resolución espacial de la imagen analizada. Por consiguiente, se pueden obtener diferentes escalas de resolución espacial cambiando los retardos en la reconstrucción simbólica de los patrones de orden. En este trabajo se propone generalizar la estimación de ambos cuantificadores simbólicos (entropía de permutación y complejidad estadística) a diferentes retrasos τ_x y τ_y para obtener los vectores de embedding. Este plano de complejidad causal-entropía causal multiescala, parece ser particularmente adecuado para caracterizar las correlaciones espaciales de imágenes a diferentes escalas de resolución.

Resultados: Los resultados obtenidos nos permiten confirmar que las correlaciones espaciales intrínsecas de las imágenes se definen con éxito mediante la implementación del enfoque multiescalar propuesto. En consecuencia, concluimos que la herramienta propuesta es versátil y práctica para identificar, caracterizar y discriminar las texturas de las imágenes.

Se exponen los ensayos realizados sobre distintas imágenes en este trabajo, para aseverar las conclusiones sobre los resultados obtenidos anteriormente expuestas. Para la primera aplicación se genera una imagen periódica en 256 niveles de gris, a partir de la cual se obtienen otras imágenes aleatorizando gradualmente los niveles de gris de cada pixel con probabilidad de cambio $p \in \{1, 2, 3, 4, 5, 10, 15, \dots, 95, 100\}$. Así se obtienen distintas muestras que describen el ruido periódico. Se calcula la complejidad y la entropía causal de permutación sobre cada una de las imágenes obtenidas, para distintos órdenes de matriz de embedding, y con sus resoluciones espaciales originales ($\tau_x = 1$ y $\tau_y = 1$). Los cálculos se realizaron 10 veces para cada imagen tomando como resultado la media y la desviación estándar de los cuantificadores ordinales, ilustrados a continuación.

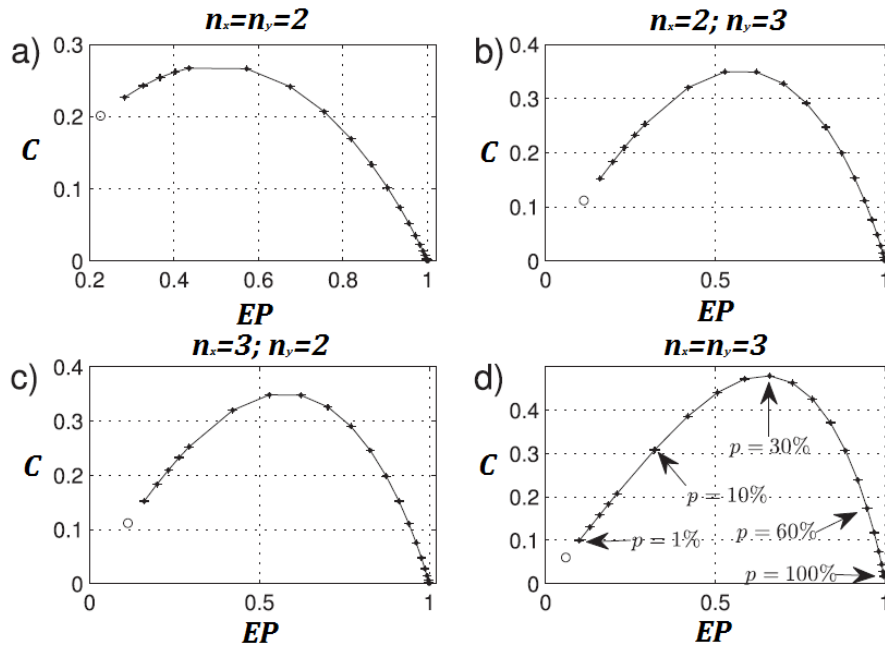


Ilustración 13: Cada punto en los gráficos representa a cada una de las imágenes aleatorizadas, representando imágenes ruidosas. En b) y c) se demuestra que la curva es invariante a la rotación del orden por la matriz de embedding $n_x \times n_y$ o $n_y \times n_x$, reflejando la simetría de la imagen. Patrones bidimensionales regulares tienen valores más bajos tanto para la entropía como para la complejidad, mientras que aquellas muestras aleatorias más "desordenadas" tienen entropías cerca de uno y complejidad cercana a cero. Se alcanza un máximo de la complejidad normalizada de Jensen-Shannon en la frontera entre orden y desorden según se espera para $p=30\%$.

Otra aplicación realizada en este trabajo, es el análisis multiescalar para discriminar rugosidades a distintas escalas. Para ello se genera una superficie fractal simulada, la que luego es segmentada para obtener imágenes de mayor escala, manteniendo la dimensión de la imagen original. Luego se calculan los parámetros causales C y EP para matrices de embedding de tamaño 2×2 , para distintos valores de retardo $\tau_x = \tau_y = \tau$ ($1 \leq \tau \leq 200$). Se obtuvieron valores de EP cercanos a 1 y de C cercanos a 0 para los casos de τ bajos. Esto confirma la ausencia de correlación espacial aleatoria para bajas escalas. Se comprueba que los cuantificadores C y EP causales son capaces de discriminar entre diferentes rugosidades según la resolución espacial considerada.

Como tercera aplicación se trabaja en el plano $C-EP$ con la base de datos normalizada de texturas Brodatz. Esta base de datos es una mejora respecto a la original, al eliminar los efectos de fondo en imágenes en escala de grises, lo que hace más difícil discriminar entre texturas utilizando solo descriptores estadísticos de primer orden. Se observa la ubicación de estas imágenes en el plano $C-EP$ causal para $n_x = n_y = 3$ y $\tau_x = \tau_y = \tau = 1$, y se concluye que ambos cuantificadores deben ser estimados para discriminar entre las diferentes imágenes de textura. Imágenes que tienen aproximadamente el mismo valor EP estimado, poseen un valor de C diferente, permitiendo discriminar entre cada una de ellas. Evidentemente, debido a su heterogeneidad intrínseca, las imágenes de textura de la base de datos Brodatz requieren un análisis multiescalar con el fin de lograr una identificación más completa de sus características a diferentes escalas. A fines ilustrativos, se realiza el análisis multiescalar para una imagen periódica de perteneciente al banco de datos analizado. Se observa que los valores estimados para C y EP son fuertemente dependientes de la escala de la imagen y además, reflejan claramente la periodicidad de la misma. Los resultados se ilustran a continuación.

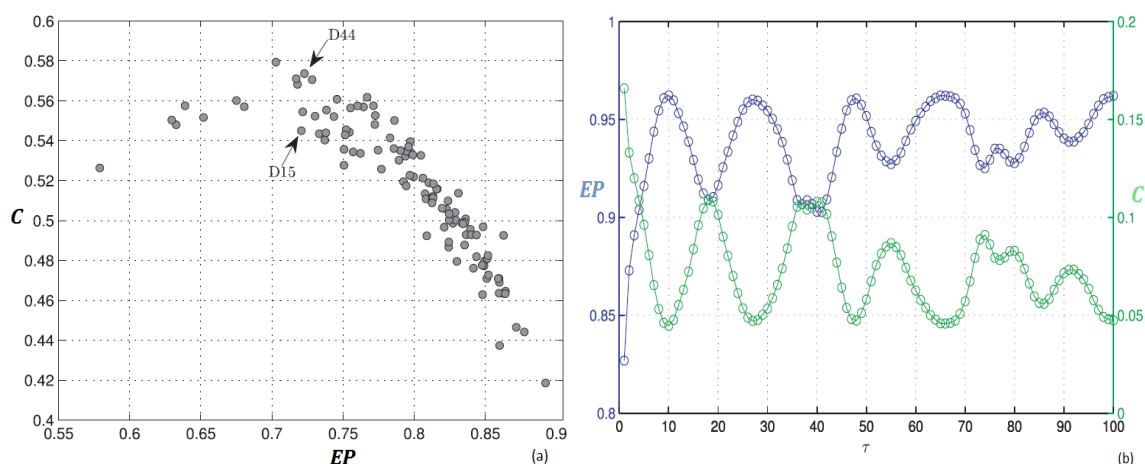


Ilustración 14: Aplicación para texturas de banco de Brodatz Normalizado. a) Ubicación de las 112 imágenes del banco analizado en el plano C-EP. Los parámetros se obtienen mediante $n_x = n_y = 3$ y $\tau_x = \tau_y = \tau = 1$. Se observa que la imagen D44 tiene la misma EP que la D15, sin embargo se diferencian en el plano debido a la diferencia en el valor de C. b) Análisis multiescalar ($1 \leq \tau \leq 100$), de la imagen de la textura periódica D101 de Brodatz Normalizado. Se realiza con $n_x = n_y = 2$. Se observa la alta dependencia de la escala y la periodicidad de los descriptores C y EP.

Por último, se realizan aplicaciones de experimentación para probar el desempeño del enfoque multiescalar en un contexto práctico, utilizando imágenes reales. Se seleccionaron dos imágenes de alta resolución del planeta Marte del banco de Archivo de Imágenes Planetarias de la NASA. Se observa que para los muestreos bajos las imágenes tienen una textura similar según los valores estimados de C y EP. Sin embargo la rugosidad puede ser bien diferenciada a medida que las escalas espaciales crecen.

Observación: este trabajo es una extensión del anterior (Ribeiro et al. 2012), con el agregado del análisis multiescala que realiza un gran aporte al análisis de texturas en imágenes, ya que se considera que la caracterización de una imagen a distintas escalas es una problemática común a todas las áreas donde el procesamiento de imágenes tenga incumbencia. Los métodos utilizados no han cambiado con respecto al trabajo anterior y ninguna nueva propuesta ha surgido en este sentido.

2.4 Conclusiones del capítulo

Durante las últimas dos décadas se introdujeron varios enfoques para una distinguir cuantitativamente patrones de orden en señales bidimensionales, como las imágenes. En particular, se ha demostrado que las técnicas para detectar características fractales y multifractales son útiles en estos casos. La geometría fractal ha sido introducida para imitar patrones texturados naturales, y pueden describir adecuadamente imágenes mediante análisis de escalado y multiescalado.

Aplicaciones recientes, confirman que las técnicas fractales son herramientas muy valiosas. Algunos ejemplos son la identificación de regiones lesionadas por enfermedades en tejidos

musculares, estimación del exponente de Hurst en imágenes de satélite para medir cambios en la superficie de la Tierra y determinación de propiedades de escala en imágenes cifradas. A pesar de todos estos esfuerzos significativos, el desarrollo de una metodología robusta para detectar y cuantificar las estructuras espaciales en las imágenes sigue representando un problema abierto y sutil. En esta dirección, la investigación y desarrollo de nuevas técnicas cada vez más eficientes se observa como un esfuerzo necesario en el ámbito del análisis de las imágenes.

3. Métodos

Como se vio el capítulo “Estado del Arte”, los trabajos desarrollados hasta el momento para la adaptación del parámetro EP en imágenes, trasladan la disposición espacial de los píxeles a un vector para luego calcular la EP como una serie de datos, de la misma manera que se realiza en series temporales. Además, no solo se calcula este parámetro para la caracterización de la textura de la imagen, sino que se realiza un plano complejidad-entropía de permutación para diferenciar entre rugosidades. En este trabajo se prueban varios métodos para utilizar el parámetro EP como un descriptor de rugosidad en imágenes de texturas, los que difieren en la manera en que los datos son agrupados para su posterior análisis.

En una serie de datos temporal, la información del momento en que sucede el acontecimiento está representado por la posición donde se guarda el dato dentro de un vector, por lo tanto analizar los datos de manera secuencial es correcto para no perder información temporal. Lo anterior no sucede en los casos de las imágenes, donde un análisis lineal de los datos, provoca una pérdida de información de la ubicación de los píxeles dentro de la imagen. El mayor desafío en la creación de una adaptación del parámetro EP en espacios bidimensionales, es guardar la información que contienen los datos en relación a su entorno cercano, sin perder la información que referencia al espacio que ocupa dicho dato. Bajo estas consideraciones se crean métodos que consideren a los píxeles no como elementos singulares, sino como parte de un entorno perteneciente a un espacio específico de la imagen.

Se realizan los métodos según dos propuestas, 1) *Analizar las imágenes como una serie de píxeles dispuestos en un vector unidimensional*, 2) *Analizar las imágenes como una serie de píxeles junto con su entorno cercano*. Varios algoritmos sobre cada una de las propuestas mencionadas son desarrollados, con el objetivo de seleccionar los métodos que mejor se adapten a la caracterización de una imagen según su nivel de rugosidad mediante el parámetro Entropía de Permutación. Se puede realizar una analogía entre los métodos tradicionales y los aquí propuestos, ya que los estadísticos de 1er orden se calculan para píxeles simples como el caso de la Propuesta 1 y los estadísticos de 2do orden se realizan con pares de píxeles tal como es el caso de la Propuesta 2.

Además a diferencia de los trabajos realizados hasta el momento en EP adaptada a imágenes, se recorre la imagen manteniendo la coherencia en el análisis de orden entre los píxeles vecinos. Es decir, la secuencia de datos extraídos de la imagen para su posterior análisis, mantienen una relación de vecindad en la imagen original. Antes de exponer los algoritmos realizados, se muestra las distintas maneras de recorrer una imagen dependiendo del método y de la propuesta de cada algoritmo particular.

Los algoritmos propuestos en este trabajo se desarrollaron según consideraciones y fundamentos que serán expuestos en cada caso particular, en el apartado 3.2 y 3.3 donde se muestran todos los algoritmos desarrollados para las Propuestas 1 y 2 respectivamente.

En los siguientes subcapítulos, se detallan todas las apreciaciones vistas en esta introducción al capítulo dedicado a los Métodos.

3.1 Tratamiento de los datos para el cálculo de EP en una imagen

La Teoría de Señales y Sistemas da un marco de trabajo para el procesamiento digital de las imágenes. Se pueden extender los conocimientos dados sobre señales unidimensionales y sistemas discretos, a las imágenes digitales. Las imágenes digitales son consideradas como señales discretas bidimensionales, las cuales son procesadas por sistemas discretos o filtros. Para entender este concepto se desarrolla el tratamiento de los datos según la representación espacial de una imagen y su representación frecuencial, siempre que sea posible.

Las imágenes digitales son obtenidas por el proceso de muestreo y cuantificación de las señales de vídeo adquirida a través de sensores especializados (cámaras u otro tipo de adquisición). La cuantificación se refiere a la cantidad de niveles posibles que puede guardar un pixel, los que se representan generalmente por potencias de 2 para facilitar el almacenamiento en el computador. Cuando el pixel sólo está representado por la información de la cantidad de luminancia, se define que la imagen está en niveles de gris, siendo estas las que se utilizan en este trabajo.

El muestreo en dos dimensiones consiste en la discretización del escenario en el plano sensor del tipo adquirente mediante dos frecuencias de pulsación, una para el eje horizontal y otra para el eje vertical. La imagen es convertida en una matriz de valores discretos de $M \times N$ píxeles. Es así que el muestreo está directamente relacionado con la resolución espacial de la imagen, cuanto mayores sean las frecuencias de muestreo mayor será la cantidad de píxeles con los que se represente a la imagen digital.

En la Ilustración 15 se muestra el proceso de discretización de la imagen continua $F(x,y)$ captada por los sensores del adquirente, donde $S(x,y)$ es el tren de impulsos bidimensional que se muestra en la Ilustración 16.

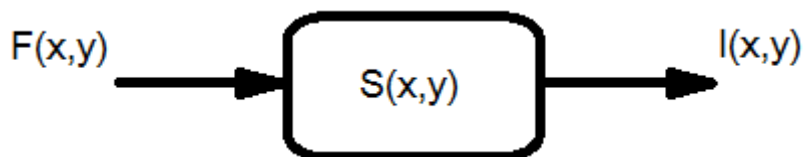


Ilustración 15: Representación del proceso de discretización de la imagen $F(x,y)$ continua en el espacio

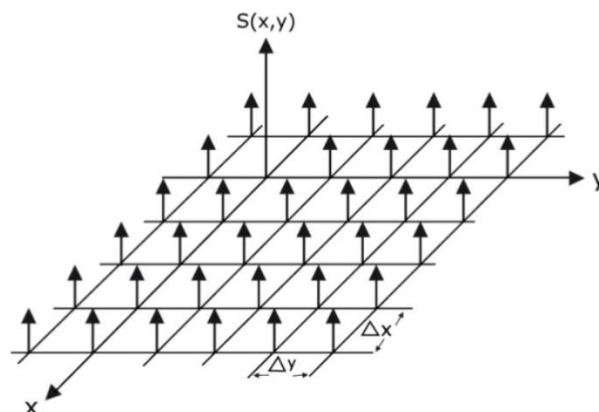


Ilustración 16: Representación de la señal de muestreo bidimensional en el espacio.

Matemáticamente la salida del sistema se puede calcular como indica la Ecuación 25 mediante la convolución de la imagen continua de entrada con el tren de impulsos bidimensional, por ser un sistema lineal e invariable en el tiempo.

$$I(x, y) = F(x, y) * S(x, y)$$

Ecuación 25: Respuesta de un sistema LTI

El operador asterisco (*), representa la convolución discreta 2D que se presenta como una combinación lineal, donde el valor de cada píxel de salida está definido por la suma de la multiplicación de sus vecinos por los coeficientes de las máscara de convolución que define a la respuesta al impulso de la señal de muestreo.

Desde el punto de vista frecuencia, el teorema de Shannon afirma que la frecuencia de muestreo debe ser al menos el doble que al ancho de banda de la señal a muestrear. En una imagen capturada, sin cambios abruptos de intensidad, las componentes de alta frecuencia de su transformada de Fourier tienden a ser nulos. Por el contrario, si la imagen contiene muchos pequeños detalles y una gran cantidad de bordes de los objetos proyectados, el espectro es elevado en alta frecuencia. Considerando que la imagen tiene un espectro de ancho de banda limitado, W , la frecuencia de muestreo debe de ser al menos del doble, $2W$. En la cuantificación de la imagen, la unidad de frecuencia espacial es un ciclo por píxel. Por tanto, desde el punto de vista del teorema, la mayor frecuencia espacial representada es la de $\frac{1}{2}$ ciclo por píxel. El armónico de mayor frecuencia requiere de dos píxeles, realizando una transición de blanco-negro (valor frecuencial más alto). Por esta razón, se determina que el detalle más pequeño debe tener un entorno de 2×2 píxeles.

La adaptación del parámetro EP desde un descriptor de orden en series temporales a un descriptor de texturas en imágenes, origina el replanteo en la lectura de los datos que aporta una imagen $I(x,y)$, como señal discreta bidimensional. De dicha adaptación surgen distintas propuestas para recorrer los datos distribuidos en el espacio de la imagen, para su posterior procesamiento en los algoritmos desarrollados. Es decir, en este trabajo utilizan distintas técnicas para extraer los píxeles que serán utilizados como los datos de análisis de los distintos algoritmos.

La principal causa para determinar la manera de leer los píxeles de la imagen, surge de la necesidad de mantener la condición de vecindad entre los píxeles de los bordes de la imagen, y así mantener la coherencia de vecindad entre los elementos extraídos para el cálculo de EP como descriptor de la rugosidad de la imagen.

La siguiente ilustración muestra la manera en que se recorren las imágenes tratadas para el caso de la Propuesta 1. Como ejemplo se muestra una imagen pequeña de tamaño 4×4 , cuya matriz de tonos de gris es la siguiente:

69	87	123	246
54	80	103	229
50	71	100	201
42	64	99	189

Ilustración 17: sentido en que se recorren los pixeles de la imagen tratada. Propuesta 1

El sentido de las flechas indica el orden en que se presentan los datos en un vector unidimensional, sobre el que luego se aplican los algoritmos desarrollados en la Propuesta 1:

69	87	123	246	229	103	80	54	50	71	100	201	189	99	64	42
----	----	-----	-----	-----	-----	----	----	----	----	-----	-----	-----	----	----	----

Ilustración 18: vector unidimensional sobre el que se calcula EP. Propuesta 1

Este vector de datos es el objeto de análisis para los algoritmos que calculan en valor de EP mediante la Propuesta 1. La manera en que se extraen los datos de este vector para su posterior tratamiento, es de manera secuencial y solapada depende del tamaño definido para el vector de embedding, como se explica en el capítulo “Estado del Arte” en la sección dedicada al método Entropía de Permutación para series temporales.

Para el caso de la Propuesta 2 donde se considera un pixel y su entorno, el análisis de los datos se realiza recorriendo la imagen de la misma manera que la vista en la Ilustración 17 para la Propuesta 1, solo que en vez de tratarse de pixeles individuales, ahora se trata de una submatriz de un tamaño determinado para representar la vecindad del pixel. Las vecindades usadas para representar el entorno del pixel son: 1) Matrices cuadradas tomando al pixel como elemento central y su vecindad como el resto de datos que quepan en la matriz, o 2) Matrices cuadradas cuya vecindad se define según el criterio de Quincunx, donde el pixel central posee una vecindad definida por un criterio de distancia a los elementos cercanos. La dimensión de la vecindad se toma en función del valor de embedding presente en los parámetros de entrada de cada una de las funciones desarrollados.

Para mostrar un ejemplo particular, se utiliza la misma imagen de la Ilustración 17, con un tamaño de vecindad de 3×3 . Los datos se tomaran de manera solapada, tal y como se realiza en la mayoría de los algoritmos desarrollados para la Propuesta 2.

En la Ilustración 19 se muestra el orden en que se extraen los datos para el cálculo de EP según la Propuesta 2. El objetivo es el mismo que para el caso de la Propuesta 1, adoptar un recorrido de la imagen donde se mantenga la relación de vecindad entre los datos extraídos.

1 69	2 87	123	246
54	80	103 4	229 3
50	71	100	201
42	64	99	189

Ilustración 19: modo de recorrer una imagen de 5x5 con una vecindad de 3x3, para extraer los datos de manera solapada. Propuesta 2. El orden en que se extraen los datos en los algoritmos de la Propuesta 2 se indica con los números de cada submatriz: 1, 2, 3, 4.

En la Ilustración 20 se muestra de manera secuencial el orden en que se extraen los datos de la imagen del ejemplo, para ser procesados por los algoritmos de la Propuesta 2:

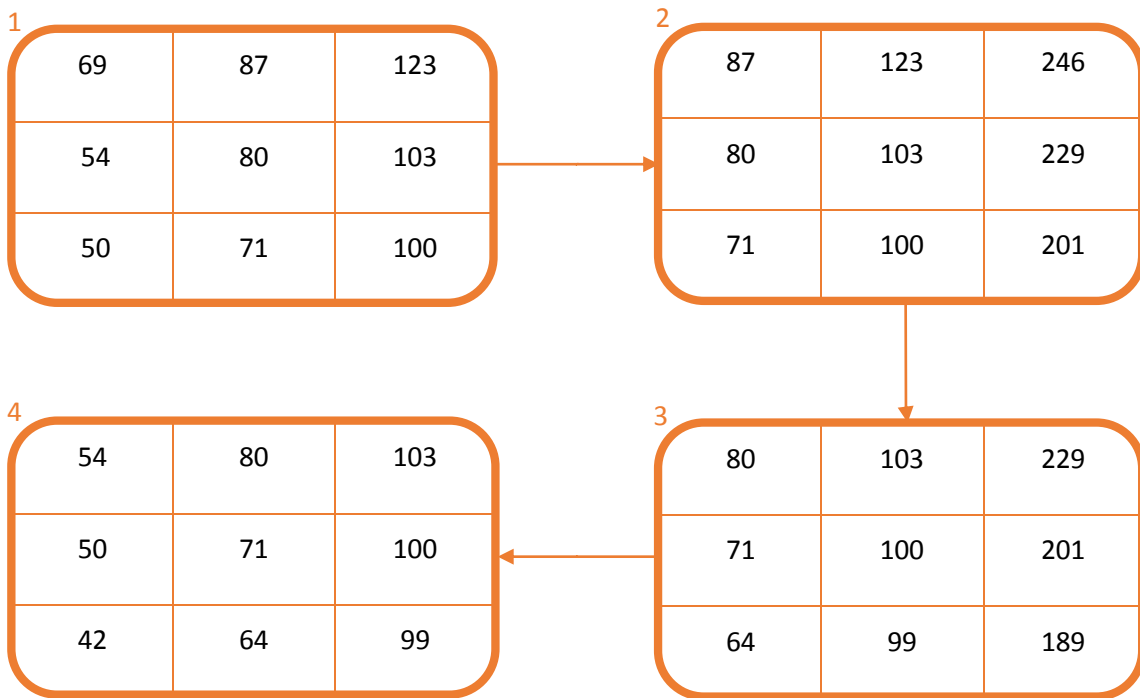


Ilustración 20: orden de análisis de las submatrices extraídas de manera solapada desde la imagen original para el cálculo de EP con algoritmos de Propuesta 2.

En el caso anterior, la vecindad del pixel central está determinada por todos los elementos pertenecientes a la submatriz extraída. Para el caso de analizar en entorno del pixel según la vecindad de Quincunx, los elementos pertenecientes al entorno cercano del pixel son determinados por la distancia a la que se encuentran respecto del pixel central. La manera de recorrer la imagen es la misma que la de la Ilustración 19, y las submatrices extraídas de la imagen original son las mismas que las de la Ilustración 20. Sin embargo en los algoritmos que utilicen a la vecindad de Quincunx como criterio de entorno cercano al pixel central, los pixeles

analizados serán solamente aquellos que se encuentren a una distancia menor o igual a $d = n - 2$ del pixel central, siendo n el valor de embedding, ya que las denominadas matrices de Quincunx son submatrices de una matriz cuadrada.

El esquema de muestreo óptimo en el espacio de imagen bidimensional es el retículo hexagonal. En esta área, la matriz de Quincunx presenta ventajas particulares como un sesgo casi isotrópico y muy pequeño en las diferentes direcciones, para situaciones tales como re-escalado de la imagen. Para ejemplificar el modo de utilización de la misma, se muestra una matriz de Quincunx de tamaño 4, que representa la máscara centrada en cada pixel de la imagen con la que se determina la vecindad correspondiente. Los valores del parámetro d , indican la distancia entre la ubicación de cada pixel y el pixel central.

0	0	0	0	1 (d=4)	0	0	0	0
0	0	0	1 (d=3,16)	1 (d=3)	1 (d=3,16)	0	0	0
0	0	1 (d=2,83)	1 (d=2,24)	1 (d=2)	1 (d=2,24)	1 (d=2,83)	0	0
0	1 (d=3,16)	1 (d=2,24)	1 (d=1,41)	1 (d=1)	1 (d=1,41)	1 (d=2,24)	1 (d=3,16)	0
1 (d=4)	1 (d=3)	1 (d=2)	1 (d=1)	1 (d=0)	1 (d=1)	1 (d=2)	1 (d=3)	1 (d=4)
0	1 (d=3,16)	1 (d=2,24)	1 (d=1,41)	1 (d=1)	1 (d=1,41)	1 (d=2,24)	1 (d=3,16)	0
0	0	1 (d=2,83)	1 (d=2,24)	1 (d=2)	1 (d=2,24)	1 (d=2,83)	0	0
0	0	0	1 (d=3,16)	1 (d=3)	1 (d=3,16)	0	0	0
0	0	0	0	1 (d=4)	0	0	0	0

Ilustración 21: Matriz de Quincunx de tamaño 4. La máscara se central en cada pixel de la imagen, definido por $d=0$. Los pixeles de la submatrix extraída de la imagen original que conforman la vecindad de Quincunx son los ubicados en las posiciones correspondientes a los 1's de la máscara. La máxima distancia entre las posiciones de los pixeles de la vecindad de Quincunx y el pixel central es $d=4$.

A modo de ejemplo aplicamos el criterio de Quincunx con la imagen de la Ilustración 17. Se elige un valor de embedding de $n = 3$, por lo tanto la distancia máxima de la vecindad del elemento central será $d = 1$, y el tamaño de las submatrices cuadradas extraídas de manera solapada de la matriz original será de: $\text{tamaño submatriz} = (d * 2) + 1 = 3$. La manera de recorrer la imagen, es igual que el caso anterior, como se ve en la Ilustración 19. En la Ilustración 22 y la Ilustración 23, se muestra la matriz de Quincunx para este ejemplo y las submatrices resultantes luego de la aplicación de la máscara respectivamente.

0	1 (d=1)	0
1 (d=1)	1 (d=0)	1 (d=1)
0	1 (d=1)	0

Ilustración 22: Matriz de Quincunx de tamaño 1. Máscara aplicada sobre cada pixel de la imagen analizada.

Se aplica esta máscara de 1's y 0's sobre las submatrices cuadradas extraídas para cada pixel de manera solapada como lo indica la Ilustración 19. Los datos resultantes son los utilizados para el cálculo del valor de Entropía de Permutación.

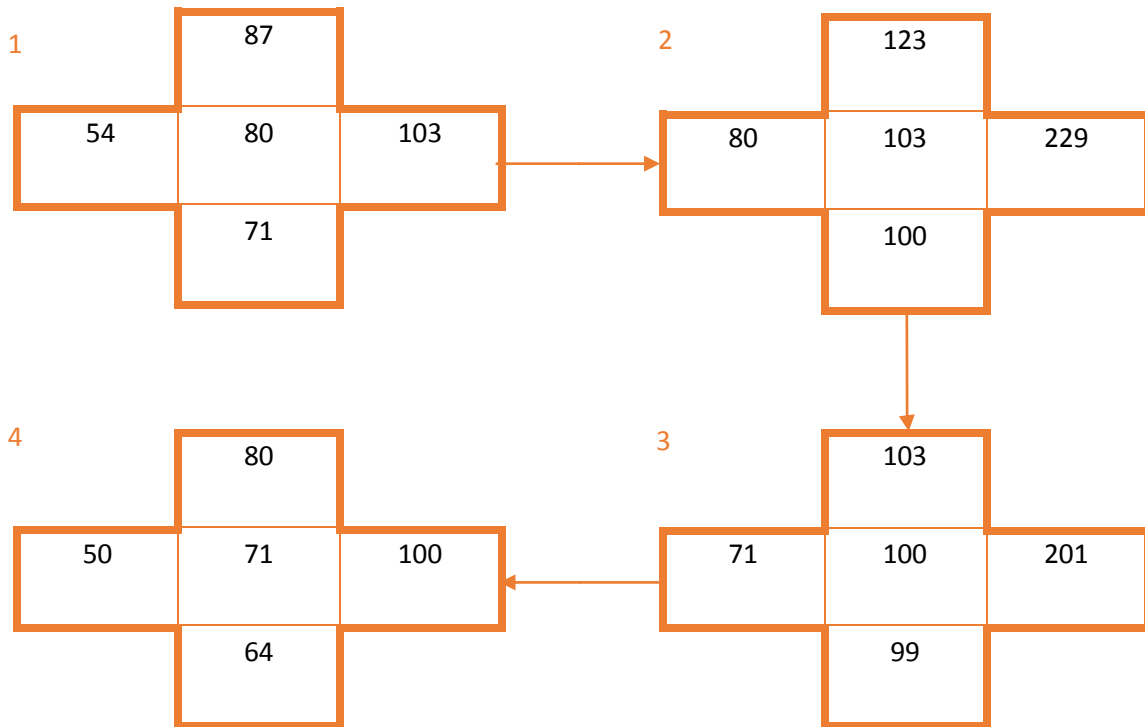


Ilustración 23 : orden de análisis de las submatrices extraídas de manera solapada desde la imagen original según el criterio de Quincunx, para el cálculo de EP con algoritmos de Propuesta 2.

Por último se muestra otra manera de recorrer una imagen considerando el pixel y su entorno, donde el recorrido será con el mismo sentido que el visto anteriormente solo que de manera no solapada. Esto es requerido por algunos algoritmos de la Propuesta 2. El sentido para mantener la coherencia entre los vecinos es el mismo que el visto hasta ahora. Para ejemplificar se utiliza la misma imagen de los ejemplos anteriores, con un tamaño de vecindad de 2×2 .

1	69	87	2	123	246
	54	80		103	229
	50	71	4	100	201
	42	64		99	189

Ilustración 24: modo de recorrido de una imagen de 5x5 mediante una vecindad de 2x2 para extraer submatrices no solapadas. El orden en que se extraen los datos en los algoritmos de la Propuesta 2 se indica con los números de cada submatriz: 1, 2, 3, 4.

En la Ilustración 25 se muestra de manera secuencial el orden en que se extraen los datos de la imagen del ejemplo, para adoptar un recorrido donde se mantenga la relación de vecindad entre los datos extraídos.

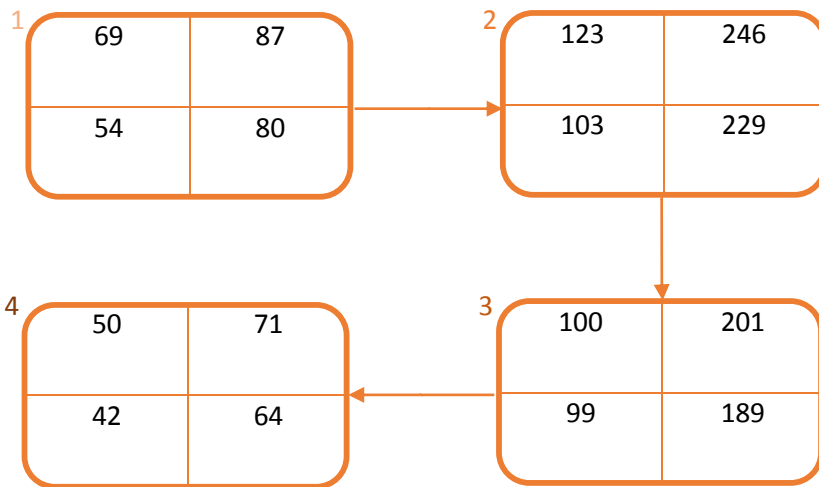


Ilustración 25: orden de análisis de las submatrices extraídas de manera no solapada desde la imagen original para el cálculo de EP con algoritmos de Propuesta 2.

Aquí se han visto cada una de las maneras que se puede tratar una imagen en este trabajo. Su implementación a un método o a otro dependerá del algoritmo desarrollado y de los objetivos del mismo. A continuación se describen estos algoritmos, indicando para cada uno de ellos cual es la manera de recorrer la imagen. Todas las funciones de la Propuesta 1 y 2, poseen los mismos parámetros de entrada:

- imagenGris
- transparencia

- *embedding*

La *imagenGris* es propiamente los datos a analizar, el parámetro *transparencia* es una matriz de 0's y 1's del mismo tamaño que la imagen, cuyos valores 1 indican la región donde se encuentra la textura de interés y por último el valor *embedding* es un número entero entre 3 y 7.

3.2 Propuesta 1: Algoritmos mediante análisis de la imagen dispuesta en un vector unidimensional

Los algoritmos desarrollados mediante esta propuesta rerorren los datos (píxeles de la imagen) como se muestra anteriormente en la Ilustración 17, para mantener una coherencia de en el análisis de datos cercanos, resultando un vector unidimensional al igual que en su trabajo original Bandt & Pompe operaban con las series temporales para el cálculo de EP. Dicho vector es el objeto de análisis para los algoritmos propuestos en esta sección.

Luego se muestran los detalles de los algoritmos diseñados mediante el software matemático Matlab versión 7.12.0 (R2011a), recordando conceptos vistos en el Capítulo 2, especialmente los referidos al procedimiento algorítmico para la obtención del histograma de patrones de orden, requerido para el posterior cálculo de EP. Mientras que el código correspondiente a cada una de las funciones donde se desarrollan dichos algoritmos se muestran en el Anexo I de este trabajo.

La secuencia del proceso general que se realiza en los métodos de la Propuesta 1, es para todos los casos como se indica en la Ilustración 26. El histograma de patrones de orden resultante depende del valor de *embedding* y del algoritmo aplicado para su obtención. Dichos algoritmos se explican a continuación, indicando cuáles fueron los objetivos para sus desarrollos. El tratamiento de la información de la imagen, depende del método aplicado para la formación del histograma de patrones de orden que tiene el algoritmo, por tal motivo se especifica la manera de analizar los datos en la explicación de cada uno de los métodos de esta propuesta.

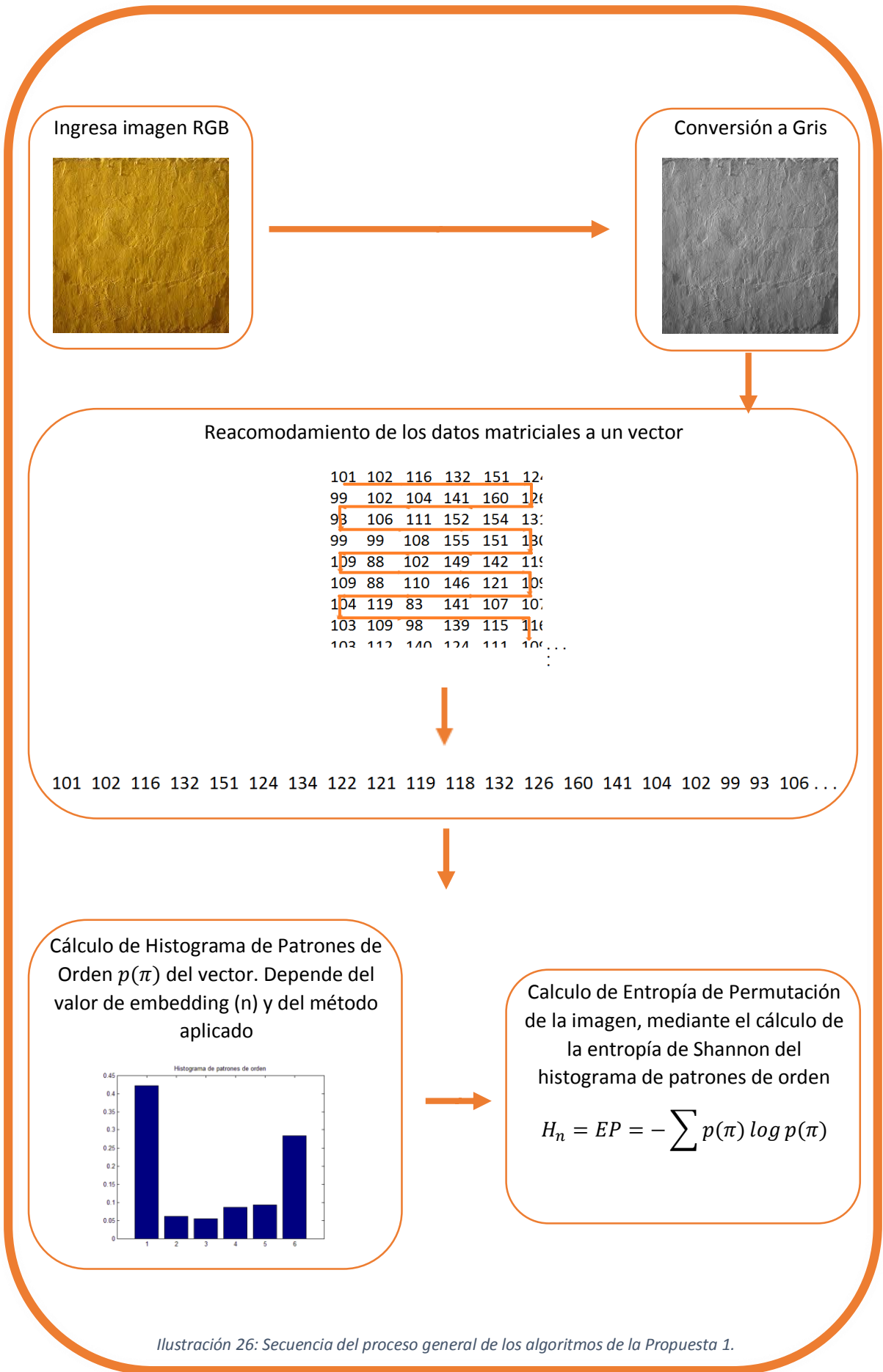


Ilustración 26: Secuencia del proceso general de los algoritmos de la Propuesta 1.

3.2.1 Entropía de permutación (EP)

Objetivo del algoritmo: adquirir el valor del parámetro EP de la misma forma que para series temporales. Este valor será referencia para comparar con los resultados de los métodos que siguen, ya que se considera que este algoritmo es la manera elemental de obtener un valor para EP. El código desarrollado para este algoritmo se encuentra en el Anexo de este trabajo, bajo la función nombrada “*entroperm.m*”.

Tratamiento de los datos: El proceso de los datos será como se indica en la Ilustración 26, aquí se explica cómo se analizan los datos en este método, para obtener el histograma de patrones de orden, también llamado histograma de permutación. Se toman los valores del vector secuencialmente y de manera solapada. Los datos son analizados mediante subvectores (llamados vector de embedding), los que poseen una cantidad de elementos entre 3 y 7, determinada por el valor de embedding (n) elegido. Se determina a qué patrón de orden corresponde dicho vector de embedding. Existen $n!$ patrones de orden posibles, los que conforman los bins del histograma. Se suma el valor de 1 al bin cuyo vector de embedding pertenece. El procedimiento se muestra ejemplificado en el capítulo “Estado del Arte” bajo el título 2.2.3 Ejemplo de cálculo de EP unidimensional.

3.2.2 EP ponderada (EPP)

Objetivo del algoritmo: se propone adquirir el valor del parámetro EP ponderada (EPP) de la misma forma que para series temporales. El parámetro también es conocido como “*Weighted-permutation entropy (WPE)*”. Este será otro de los valores referencia para comparar con los resultados de los métodos que siguen. Se considera que para el caso de las imágenes, EPP presenta una ventaja frente al valor del EP básico, ya que se tiene en cuenta que los patrones de orden pueden contener valores de amplitud muy dispares, debido a que cada vector de embedding analizado afecta al histograma de permutación según un peso dado por la distancia existente en la amplitud relativa entre los elementos del vector, y como es sabido no es lo mismo la variación de la intensidad de los píxeles en una parte lisa de la imagen que cuando se presenta un borde, representando este último una cantidad de energía mayor. Por lo tanto debe sumar más energía al histograma de permutación una sección de la imagen con un borde que aquella que no lo tenga. El código desarrollado para este algoritmo se encuentra en el Anexo de este trabajo, bajo la función nombrada “*entropermPond.m*”.

Tratamiento de los datos: La secuencia del proceso de los datos será como se indica en la Ilustración 26, aquí se explica cómo se analizan los datos para obtener el histograma de patrones de orden, también llamado histograma de permutación. Se toman los valores del vector secuencialmente y de manera solapada. Los vectores de embedding poseen una cantidad de elementos entre 3 y 7, determinada por el valor de embedding (n) elegido. Se determina a qué patrón de orden corresponde dicho vector de embedding, y luego se calcula el peso (p) dado por la diferencia cuadrática media entre cada elemento y el valor medio de las amplitudes del vector. Existen $n!$ patrones de orden posibles, los que conforman los bins del histograma. Se suma el valor de p al bin cuyo vector de embedding pertenece, resultando luego de analizar la

totalidad de los datos el histograma de permutación final. El algoritmo detallado se encuentra explicado en el capítulo “Estado del Arte” en la sección 2.2.4 Variaciones del algoritmo EP.

3.2.3 EP de imagen umbralada

Objetivo del algoritmo: normalmente, el valor de EP se calcula comparando los datos de una serie, pero sin considerar la distancia relativa entre los elementos de la secuencia. A partir de esta definición se pensó una manera de clasificar los datos en clases, dependiendo del rango de valores al que pertenezcan, a través de la comparación con un valor umbral. De esta manera, valores cercanos pertenecerán a la misma clase, para que al momento de compararlos se los considere iguales y no distintos sólo por unos pocos tonos de gris de diferencia. Esto se logra mediante una previa clusterización de los píxeles de la imagen, reemplazando el valor del píxel por el número de clase a la que pertenece. El código desarrollado para este algoritmo se encuentra en el Anexo de este trabajo, bajo la función nombrada “*epUmbralTonos.m*”.

Tratamiento de los datos: A diferencia de los métodos anteriores, no se cambia la manera de obtener el histograma, sino que se realiza un pretratamiento de los datos antes del cálculo del histograma de patrones de orden. Por lo tanto la secuencia del proceso de los datos será como se indica en la Ilustración 26, agregando la clusterización de los elementos del vector de datos a analizar. Para ello se utilizan tantas clases como es el valor de embedding (n). Por ejemplo si $n = 3$, entonces existirán 3 clases posibles a las que puede pertenecer el píxel. Para el caso de utilizar una resolución de 256 niveles de gris, los rangos de separación entre las clases son:

- Clase 1: niveles de gris entre 0 y 84.
- Clase 2: niveles de gris entre 85 y 169.
- Clase 3: niveles de gris entre 170 y 255.

Luego de este proceso, se aplican los algoritmos EP y de EPP al nuevo vector clasificado, surgiendo nuevos parámetros llamados entropía de permutación umbral de tonos y entropía de permutación umbral de tonos ponderada.

3.2.4 EP con umbral de comparación

Objetivo del algoritmo: Conceptualmente, cuanto mayor sea la diferencia entre un píxel y sus vecinos, más rugosa se verá la textura de una imagen. A raíz de este concepto, es importante crear métodos que continúen con propuesta desarrollada en “*Entropía de permutación ponderada*”, donde se busca obtener un histograma de patrones de orden dependiente de la distancia entre el valor de los píxeles. Aplicando esta propuesta, se crea este algoritmo, donde la decisión de si un elemento del vector de embedding es mayor o menor que otro, dependerá de que estén distanciados a un valor umbral de diferencia. Dicho umbral depende del valor de embedding (n) adoptado. Se llegó a un valor empírico dado por:

$$Umbral = Parte\ entera\ de\ \left(\frac{256}{2n}\right)$$

Ecuación 26: definición del Umbral. Este valor determina la distancia relativa entre datos vecinos para considerarlos pertenecientes a distintas clases.

A modo de ejemplo para $n = 3$, resulta un $Umbral = 42$, lo que implica que un pixel cualquiera sólo será diferente de aquellos que disten a 42 tonos de grises de él. Este método salva el problema del algoritmo anterior (entropía de permutación con umbralización de la imagen), el que considera la misma entropía entre los tonos de gris del final de un rango con los primeros y los últimos del rango siguiente, para ejemplificar con la propuesta de un embedding de 3, un pixel de valor 84 comparado con un vecino de 85, pondera la misma cantidad de energía que con un pixel vecino de un valor de 169, ya que la cantidad de energía estará dada por el salto de Clase 1 a Clase 2, y no por la distancia relativa entre los pixeles como se realizar en el método que aquí se expone. El código desarrollado para este algoritmo se encuentra en el Anexo de este trabajo, bajo la función nombrada “*epUmbral.m*”.

Tratamiento de los datos: La secuencia del proceso de la información será como se indica en la Ilustración 26. Para obtener el histograma de patrones de orden, se procede como con todos los algoritmos anteriores, se elige un valor n de embedding y se extraen los valores del vector de datos secuencialmente y de manera solapada, generando los vectores de embedding que poseen una cantidad de elementos entre 3 y 7, determinada por el valor de n . La diferencia con otros métodos, radica en el proceso que determina el patrón de orden al que pertenece el vector de embedding, aquí se comparan los elementos vecinos del vector y se determina que son diferentes solo si distan en un valor mayor o igual al del Umbral. Esta nueva manera de comparar los datos se aplica a los algoritmos EP y de EPP, surgiendo dos nuevos parámetros llamados entropía de permutación umbral y entropía de permutación umbral ponderada.

Para resumir, se presenta la Tabla 4 donde se indica en pasos el funcionamiento algorítmico de cada una de las funciones realizadas para la Propuesta 1, junto a sus parámetros de salida.

Función	Pasos del algoritmo	Parámetros de Salida
<i>entroperm.m</i>	<ol style="list-style-type: none"> 1. Ingresa Imagen RGB. 2. Se convierte a gris. 3. Los datos en formato Uint8 distribuidos en una matriz se reacomodan en un vector unidimensional. 4. Se recorre el vector de datos de manera secuencial y solapada mediante el subvector de embedding de tamaño n (vector de embedding). 5. Se determina el patrón de orden correspondiente a cada subvector. 6. Se suma el valor de 1 en el bin del histograma de permutación representante del patrón de orden del vector de embedding analizado. 7. Una vez que se analiza la totalidad de los datos, se normaliza el histograma. 8. Se calcula la entropía de 	- EP

	Shannon del histograma de permutación resultante, la que se nombra “Entropía de Permutación” (EP).	
<i>entropermPond.m</i>	<ol style="list-style-type: none"> 1. Ingresar Imagen RGB. 2. Se convierte a gris. 3. Los datos en formato Uint8 distribuidos en una matriz se reacomodan en un vector unidimensional. 4. Se recorre el vector de datos de manera secuencial y solapada mediante el subvector de embedding de tamaño n (vector de embedding). 5. Se determina el patrón de orden correspondiente a cada subvector. 6. Se calcula el peso (w) del vector de embedding como el valor medio cuadrático de las distancias entre los valores del pixel y la media aritmética del subvector. 7. Se suma el valor de w en el bin del histograma de permutación representante del patrón de orden calculado en el paso 6. 8. Una vez que se analiza la totalidad de los datos, se normaliza el histograma. 9. Se calcula la entropía de Shannon del histograma de permutación resultante, la que se nombra “Entropía de Permutación Ponderada” (EPP). 	- EPP
<i>epUmbralTono.m</i>	<ol style="list-style-type: none"> 1. Ingresar Imagen RGB. 2. Se convierte a gris. 3. Los datos en formato Uint8 distribuidos en una matriz se reacomodan en un vector unidimensional. 4. Se clasifican los pixeles en tantas clases como es el número de embedding (n), para ello se reemplaza el valor del pixel por la clase a la que pertenece. 5. Se recorre el vector de datos de manera secuencial y solapada mediante el subvector de embedding de tamaño n (vector de embedding). 6. Se determina el patrón de orden correspondiente a cada subvector. 7. Se suma el valor de 1 en el bin del histograma de permutación representante del patrón de orden del vector de embedding analizado. 8. Se calcula el peso (w) del vector de embedding como el valor medio cuadrático de las distancias entre los valores del pixel y la media aritmética del subvector. 9. Se suma el valor de w en el bin de otro histograma de permutación representante del patrón de orden ponderado. 10. Una vez que se analiza la totalidad de los datos, se normalizan los histogramas. 11. Se calcula la entropía de Shannon de los histogramas de permutación resultantes de manera independiente. Los resultados se nombran “Entropía de Permutación Umbral de Tonos” (EPT) y “Entropía de Permutación Umbral de Tonos Ponderada” (EPTp). 	- EPT - EPTp
<i>epUmbral.m</i>	<ol style="list-style-type: none"> 1. Ingresar Imagen RGB. 2. Se convierte a gris. 3. Los datos en formato Uint8 distribuidos en una matriz se reacomodan en un vector unidimensional. 4. Se calcula el valor de umbral dependiendo del valor de embedding (n) adoptado. 5. Se recorre el vector de datos de manera secuencial y solapada mediante el subvector de 	- EPU - EPU _p

	<p>embedding de tamaño n (vector de embedding). 6. Se utiliza el valor Umbral para determinar el patrón de orden correspondiente a cada subvector. 7. Se suma el valor de 1 en el bin del histograma de permutación representante del patrón de orden del vector de embedding analizado. 8. Se calcula el peso (w) del vector de embedding como el valor medio cuadrático de las distancias entre los valores del pixel y la media aritmética del subvector. 9. Se suma el valor de w en el bin de otro histograma de permutación representante del patrón de orden ponderado. 10. Una vez que se analiza la totalidad de los datos, se normalizan los histogramas. 11. Se calcula la entropía de Shannon de los histogramas de permutación resultantes de manera independiente. Los resultados se nombran “Entropía de Permutación Umbral” (EPU) y “Entropía de Permutación Umbral Ponderada” (EPU_p).</p>	
--	--	--

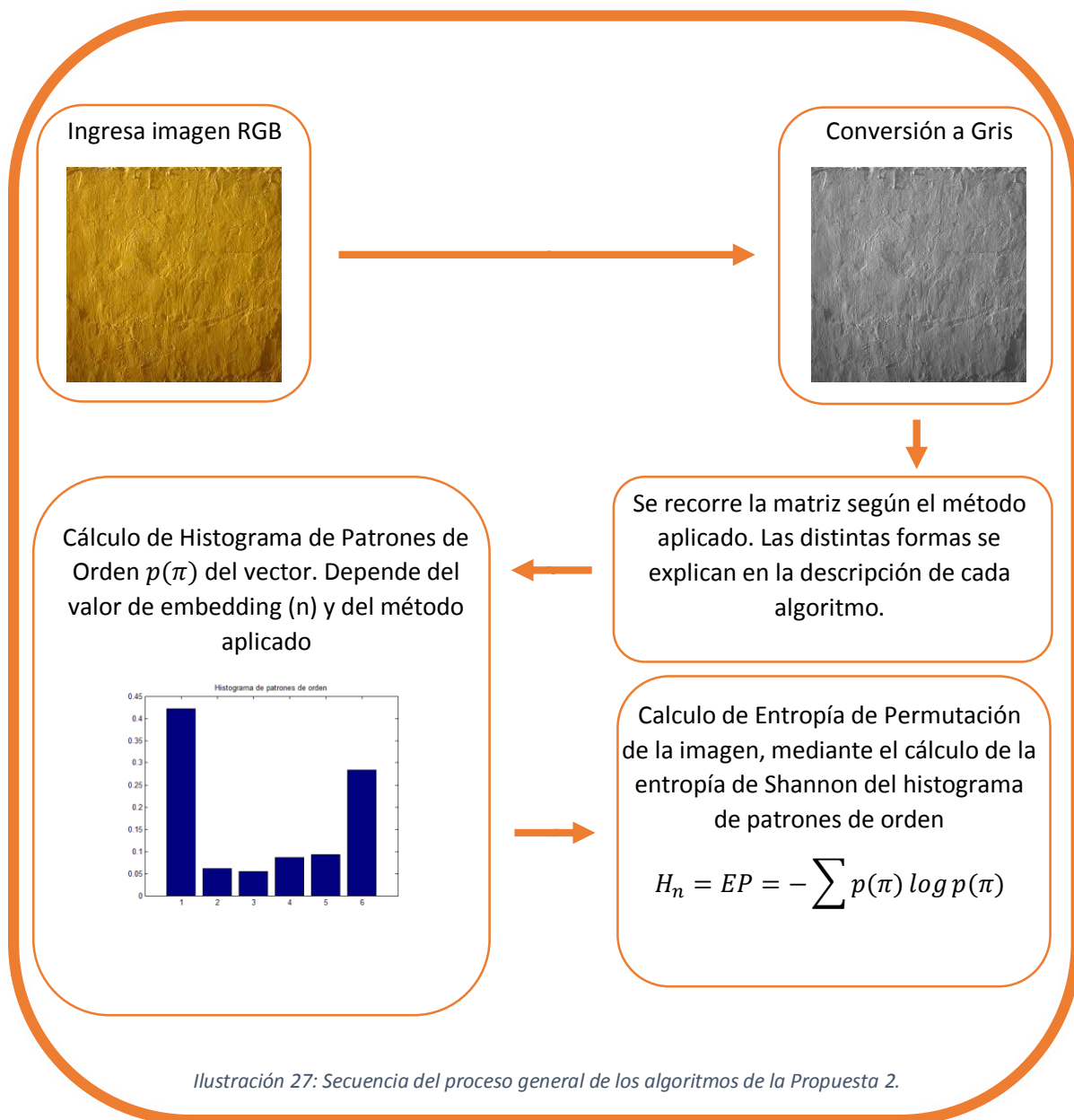
Tabla 4: Nombre de la función desarrollada en Matlab, parámetros de salida y descripción breve del procedimiento para cada uno de los algoritmos de la Propuesta 1.

3.3 Propuesta 2: Algoritmos mediante análisis de la imagen por el pixel y su vecindad

En los algoritmos de la Propuesta 2 se analizan los píxeles junto con su entorno cercano, para determinar el valor de energía de orden de la imagen. En este sentido se pueden crear una gran variedad de propuestas, ya que sólo se debe pensar en métodos que extraigan características de orden que representen al pixel y su vecindad, con las que luego se pueda determinar un grado de orden presente en ese extracto de la imagen. Muchos algoritmos fueron creados y probados en la práctica de este trabajo bajo esta propuesta. Algunos fueron descartados inmediatamente luego de ensayados, debido a su complejidad de cálculo o tiempo irrisoria de obtención de los resultados.

El modo de analizar los datos de la imagen en esta propuesta, se realiza mediante submatrices que recorren la imagen secuencialmente de manera solapada o no, como se muestra en la Ilustración 19 e Ilustración 24 respectivamente. Los anteriores son los ejemplos generales de recorrido de la imagen, sin embargo hay algunos métodos que hacen uso de otras formas, las que se explican luego dentro de cada caso particular.

El acondicionamiento de los parámetros de entrada en el inicio de los algoritmos, y el tratamiento del histograma de permutación al final de los mismos, es igual en ambas propuestas. Sin embargo, cada algoritmo trata de manera distinta a la imagen al momento de determinar los patrones de orden que la misma posee. Estas diferencias se muestran detalladamente en la explicación de cada uno de los métodos realizados para la Propuesta 2, mientras que la secuencia general del proceso se muestra en la siguiente ilustración:



3.3.1 EP de imágenes filtradas

Objetivo del algoritmo: Este es una primera aproximación a la Propuesta 2, donde se considera que la información en una imagen debe tratarse según el píxel y su entorno. Aquí se busca minimizar las distancias relativas entre los píxeles cercanos para enfocarse en el análisis del ordenamiento que presentan respecto de sus vecinos, es decir, se busca tener elementos que guarden la información de espacio y representen a una porción de la imagen, y no al píxel individual. Para ello se aplican distintos filtros espaciales, los que se caracterizan por aplicarse directamente sobre los píxeles de la imagen y no sobre la transformada de Fourier de la misma, los que realizan una operación de convulsión sobre el píxel y su vecindad, sustituyendo el valor

de cada píxel por la suma de los productos entre la máscara de convulsión que representa al tipo de filtro y los píxeles situados en un entorno cercano. El resultado es una nueva matriz de píxeles filtrados, donde ahora cada elemento representa la relación entre el píxel de la imagen original y su vecindad. Los filtros aplicados son pasa-bajos, de manera que cada nuevo valor de píxel represente el promedio del entorno. El código desarrollado para este algoritmo se encuentra en el Anexo de este trabajo, bajo la función nombrada “epFiltros.m”.

Tratamiento de los datos: Para la aplicación de los filtros se usará la relación fundamental de los sistemas discretos. No es necesario actuar sobre el espectro de la propia imagen para filtrarla, sólo se requiere aplicar una máscara de convolución cuya respuesta frecuencial corresponda a la cancelación del espectro deseado. Normalmente, el filtro discreto está constituido por una secuencia de ponderación finita FIR, del tipo 3x3 o de 5x5. En nuestro caso, se define el tamaño del filtro bidimensional mediante el valor del *embedding*. A los coeficientes de estos filtros se les llama máscaras de convolución. El proceso de convolución se presenta como una combinación lineal de los píxeles del entorno ponderados por los elementos de la secuencia de ponderación o máscara de convolución, de forma parecida al caso unidimensional. Entonces la imagen en tonos de gris es convolucionada por máscaras predeterminadas en Matlab, que representan filtros de Media aritmética y Gauss. Si bien estos filtros son muy usados para eliminar el ruido aleatorio presente en las altas frecuencias de la imagen, en este caso se utiliza para obtener una representación del entorno el píxel que guarde la información de la posición en la que ello ocurre. Las características en el espacio y en frecuencia de los dos filtros se muestran en la Tabla 5.

Filtro	Representación espacial (mascara)	Representación frecuencial
Media aritmética	$mascara = \frac{1}{n \times n} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \dots \\ 1 & 1 & 1 \end{bmatrix}_{n \times n}$	
Gauss	$h_g(x, y) = e^{-\frac{(x^2+y^2)}{2\sigma^2}}$ $mascara = \frac{h_g(x, y)}{\sum_n \sum_n h_g(x, y)} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \dots \\ 1 & 1 & 1 \end{bmatrix}_{n \times n}$	

Tabla 5: Características de los filtros convolucionales aplicados en el dominio espacial de la imagen.

Se eligen filtros predeterminados de Matlab, todos ellos se representan por una máscara cuadrada en el espacio cuyo tamaño es el valor de *embedding*. El filtro de media tiene una representación en el espacio de un escalón bidimensional, por lo tanto su respuesta frecuencial presenta muchos lóbulos secundarios de gran amplitud. Además es un filtro de fase no lineal, lo que genera distorsiones en la fase de la imagen.

Luego se implementa un filtro gaussiano, cuya representación en el espacio es una campana de gauss de desviación estándar σ . El valor máximo aparece en el pixel central y disminuye hacia los extremos tanto más rápido cuanto menor sea el parámetro σ . En frecuencia los lóbulos secundarios tienen baja amplitud y el lóbulo principal tiene un ancho que depende del valor de σ . Las respuestas en frecuencia de estas máscaras son filtros pasos cuyos anchos de banda dependen de sus varianzas y son de fase lineal. A medida de que la varianza sea más elevada significará que tiene en cuenta a vecinos más alejados, por el contrario, si la varianza es más pequeña indicará que sólo ponderará con los más próximos. La varianza funciona como grado de ponderación en la vecindad. Ésta es la interpretación en el dominio espacial. Desde el punto de vista frecuencial, la varianza determina el ancho de banda del filtro paso bajo que supone la máscara de Gauss. A mayor varianza menor ancho de banda.

En el proceso de discretización de la máscara Gaussiana aparece el problema del número de coeficientes finitos a emular a la función continua. Para evitar la formación de lóbulos secundarios en la respuesta frecuencial de la máscara, se toma como regla que el tamaño del de la ventana debe ser al menos de $ventana \geq 3c$, donde c es el tamaño del lóbulo principal, que depende de la varianza de la manera $c = 2\sqrt{2\sigma^2}$. En el caso de nuestro algoritmo el parámetro fijo es el tamaño de ventana, el que es igual al valor de *embedding* (n). Por lo tanto el valor de la desviación estándar utilizado para minimizar los lóbulos secundarios será:

$$\sigma \leq \frac{n\sqrt{2}}{12}$$

Ecuación 27: desviación estándar para filtro de Gauss sin lóbulos secundarios

Por último, se utiliza un tercer filtro, llamado filtro de mediana, el que no está representado por una máscara de convolución, sino que reemplaza cada píxel por el valor mediano entre todos los elementos de la vecindad. Tiene la ventaja de que el valor final del píxel es un valor real presente en la imagen y no un promedio, de este modo se reduce el efecto borroso que realizan los filtros de media. Para entender mejor el procedimiento se explica el siguiente ejemplo, en una vecindad de 3x3:

4	56	71
28	130	156
200	42	23

Ilustración 28: Segmento de la imagen. Píxel central (130) acompañado de su vecindad.

Los píxeles de la porción de 3x3 de la imagen original, son ordenados de menor a mayor y el píxel central es reemplazado por el valor que se encuentre en el centro del ordenamiento:

4	23	28	42	56	71	130	156	200
---	----	----	----	----	----	-----	-----	-----

Ilustración 29: Orden por intensidad de los elementos de la vecindad. El valor ubicado en la posición central (56) es el representante de vecindad de 3x3.

En el ejemplo representado en la Ilustración 28 y la Ilustración 29, se reemplaza el píxel central (130), por el valor ubicado en la posición central del vector que contiene los elementos ordenados (56). Este procedimiento es aplicado sobre todos los píxeles de la imagen. Este filtrado es una operación no lineal que se utiliza a menudo en el procesamiento de imágenes para reducir el ruido de "sal y pimienta". Un filtro mediano es más eficaz que la convolución cuando el objetivo es simultáneamente reducir el ruido y conservar los bordes. En este trabajo no se utiliza con tal sentido, ya que como se ha hecho mención no se realiza un acondicionamiento de la imagen previo al procesamiento para el cálculo de EP, para demostrar la robustez del método y no demorar el proceso de cálculo aprovechando la cualidad de rapidez de cálculo de EP. El sentido de la utilización de dicho filtro es representar a cada píxel por un valor que indique la relación que guarda con su entorno, de esta manera se obtiene una nueva matriz cuyos elementos representan la vecindad de cada posición. Esta nueva matriz lleva el nombre de "imMediana".

Así, a través de la aplicación de cada filtro con la imagen en el espacio, se obtienen 3 imágenes diferentes derivadas de la imagen original ("ImMediana", "ImAritmetica" e "ImGauss"), sobre las cuales se aplican cada uno de los métodos de la Propuesta 1, por lo tanto a partir de aquí, el tratamiento de los datos para obtener el valor de entropía de permutación en sus diferentes variantes para cada una de las 3 imágenes filtradas, será igual al explicado en los algoritmos de la Propuesta 1 en la Tabla 4.

Se hace uso de la Tabla 6 para resumir las características de los algoritmos y el modo de aplicación del filtro sobre la imagen original.

Tipo de Filtro	Parámetros de uso del filtro	Resultado de aplicación del filtro
<i>Mediana</i>	<ul style="list-style-type: none"> - Imagen en tonos de gris. - Tamaño de máscara: matriz cuadrada de dimensión de embedding (n). 	Sustituye el valor de cada píxel por la mediana de los valores que engloba una ventana de selección dada por el valor de embedding.
<i>Media aritmética</i>	<ul style="list-style-type: none"> - Imagen en tonos de gris. - Tamaño de máscara: matriz cuadrada de dimensión de embedding (n). 	Sustituye el valor del píxel por la mediana aritmética (promedio) de los valores que engloba una ventana de selección dada por el valor de embedding.

<p><i>Gaussiano</i></p>	<ul style="list-style-type: none"> - Imagen en tonos de gris. - Tamaño de máscara: matriz cuadrada de dimensión de <i>embedding</i> (n). - Desviación estándar del filtro gaussiano: $\sigma \leq \frac{n\sqrt{2}}{12}$ 	<p>Implementa una máscara de convolución sobre cada pixel de la imagen, que imita una campana de Gauss cuyos parámetros se definen por la varianza y el valor de embedding.</p>
-------------------------	--	---

Tabla 6: Características de los filtros aplicados sobre la imagen a analizar para el método "epFiltros.m"

Se obtienen nuevas imágenes mediante el filtrado de la imagen original. Cada pixel revela una relación entre los datos de una vecindad de tamaño del *embedding* de la imagen original. Ésta, es la manera del método de tratar al pixel junto a su entorno cercano, para cumplir con las premisas de la Propuesta 2.

Por último, cada una de las nuevas imágenes es el objeto de análisis para los algoritmos de la Propuesta 1. En Tabla 4, se observa que para cada imagen analizada, existen 6 parámetros de salida (EP, EPP, EPT, EPTp, EPU, EPUp), que describen distintas versiones de la entropía de permutación. Como consecuencia, esta función brinda 6 algoritmos nuevos por cada una de las 3 imágenes de entrada, dando un total de 18 métodos distintos.

3.3.2 EP de valores medios

Objetivo del algoritmo: se calcula el valor medio de una submatriz dentro de la imagen original y se almacena ese valor en una ubicación dentro de una nueva matriz, que guarda la información de la ubicación de la submatriz en la imagen original. Se calculan todos los algoritmos de la Propuesta.1 sobre esta nueva matriz de valores medios. Este método se realizó con el objetivo de relajar la propuesta de filtrar la imagen con un filtro de media sobre cada uno de los píxeles de la imagen. Aquí se aplica una ventana que calcula el valor medio de una región cuadrada de tamaño n de la imagen, y luego se pasa a la región siguiente, sin centrar la ventana particularmente en cada uno de los píxeles. El código desarrollado para este algoritmo se encuentra en el Anexo de este trabajo, bajo la función nombrada "epMedia.m".

Tratamiento de los datos: El recorrido de la imagen se realiza de manera secuencial y no solapada, como se muestra en el ejemplo de la Ilustración 24. Las submatrices cuadradas son del tamaño del valor de embedding (n), el valor medio entre sus elementos se almacena en una nueva matriz llamada "matrizPromedios" cuyo tamaño resulta ser n veces menor al de imagenGris. La posición que adquieren estos valores medios dentro de la matriz matrizPromedios, guarda relación con la ubicación de la submatriz en la imagen original. Una vez recorrida la totalidad de imagenGris, se aplican los algoritmos de Entropía de Permutación, EP ponderada y EP umbral para la matriz matrizPromedios. A diferencia del método anterior no se somete a la información bajo el análisis de EP umbral tonos, ya que se considera que la

información fue lo suficientemente umbralizada como para agregar otro filtrado de los datos. En este caso, para una imagen de entrada habrá 4 valores de salida, por cada valor de embedding.

3.3.3 EP de matriz de entropías

Objetivo del algoritmo: Para guardar la información de los datos en relación a su entorno cercano, sin perder la información referente al espacio que ocupa dicho dato dentro de la imagen, este método crea nuevas matrices llamadas “*matrizEP*”, “*matrizEPP*” y “*matrizEPU*” cuyos valores, representan respectivamente la EP, EPP y EPU entre los pixeles de la vecindad de cada valor. Para explicar mejor el método, se detalla el procedimiento para obtener el valor de entropía sobre *matrizEP*: se seleccionan pequeñas muestras de *imagenGris*, para calcular la EP que poseen y almacenar ese valor en la ubicación correspondiente al pixel central de dicha submatriz de la imagen original, se obtiene una matriz de entropías de permutación por sector de la imagen (*matrizEP*), de esta manera se mantiene la información de ubicación de la entropía en la imagen. Luego se calcula el descriptor EP de *matrizEP* de la misma manera que se realiza en la propuesta 1, por lo tanto el orden o desorden de los datos se analiza sobre *matrizEP*, considerando que existe más energía en la imagen, cuanto mayor es la diferencia entre las energías de los distintos sectores de la misma. Se aplican los algoritmos EP y EPP desarrollados en la Propuesta 1 directamente sobre las matrices de entropías de permutación obtenidas (*matrizEP*, *matrizEPP* y *matrizEPU*), de esta manera se obtiene una combinación de 6 algoritmos nuevos. No se aplican los algoritmos restantes de la propuesta 1 (EPT, EPTp, EPU y EPUp) ya que los mismos fueron realizados exclusivamente para imágenes en tonos de gris, o sea para matrices cuyos elementos son valores enteros entre 0 y 255 o más, y no para fueron pensados para matrices de entropías, cuyos valores varían entre 0 y 1. El código desarrollado para este algoritmo se encuentra en el Anexo de este trabajo, bajo la función nombrada “*epEspacial.m*”.

Tratamiento de los datos: Se recorre la imagen de manera secuencial y solapada, como se muestra en el ejemplo de la Ilustración 19: modo de recorrer una imagen de 5x5 con una vecindad de 3x3, para extraer los datos de manera solapada. Propuesta 2. El orden en que se extraen los datos en los algoritmos de la Propuesta 2 se indica con los números de cada submatriz: 1, 2, 3, 4.. Las submatrices cuadradas, representa un pequeño entorno de la imagen original de tamaño del valor de embedding (n) el que varía entre 3 y 7. Se aplican 3 de los algoritmos de la Propuesta 1 sobre estas submatrices, con el menor valor de embedding posible según Bandt & Pompe, ya que el entorno de pixeles analizado es muy pequeño como máximo de 7x7. Los resultados son almacenados en 3 matrices diferentes, dependiendo del método aplicado. Por ejemplo, si se calcula EP de cada submatriz, los resultados son almacenados en una matriz que llamaremos “*matrizEP*”, si se obtiene el valor de EPP de cada submatriz los resultados se guardan en una matriz llamada “*matrizEPP*” y si se extrae de cada submatriz el parámetro EPU los resultados se ubican en otra matriz denominada “*matrizEPU*”. De esta manera se obtuvieron 3 matrices que guardan en sus datos la cantidad de energía de permutación en las diferentes regiones de la imagen original. Los resultados de este método son

los valores de EP, EPP y EPU de cada una de las 3 matrices formadas, dando un total de 9 parámetros de salida.

3.3.4 EP con vecindad de Quincunx

Objetivo del algoritmo: Como se hizo mención, el desafío de este trabajo se centra en hallar la manera óptima de recorrer los datos que una imagen brinda, para obtener información del orden de sus píxeles. Por lo tanto, experimentar con distintos criterios de vecindad para definir el entorno de un píxel no es trivial, sino necesario para abordar a mejores conclusiones. En este caso se propuso recorrer la imagen según el criterio de Quincunx. Se utiliza la vecindad hexagonal de Quincunx para definir la región donde buscar elementos representativos cercanos a cada píxel. Para ello se crea un vector de embedding de 3 elementos: 1) valor más alejado al píxel central, 2) el propio píxel central, y 3) el valor más cercano al píxel central. En base a estos vectores se calculan los valores de EP, EPP y EPU. El código desarrollado para este algoritmo se encuentra en el Anexo de este trabajo, bajo la función nombrada “*epQuincunx.m*”.

Tratamiento de los datos: Se recorre cada píxel a través de las submatrices cuadradas de tamaño $(n \times 2) + 1$, donde n es el valor de embedding de entrada al algoritmo. Dicho píxel se define como el elemento central de la submatriz, de cada una de ellas se seleccionan 3 valores representativos, los que deben pertenecer a la vecindad de Quincunx correspondiente, como se describe en la Ilustración 23. Los valores representativos elegidos de cada submatriz dependen del valor del píxel central, ya que se comparan todos los elementos dentro de la vecindad de quincunx con el píxel central y se toma el vecino más cercano (vecinoMin) y el más alejado (vecinoMax) al elemento central. Con estos dos elementos y el valor del píxel central se arma un vector de embedding en el siguiente orden: [vecinoMin píxelCentral vecinoMax], con el que se determina el patrón de orden correspondiente según el algoritmo de entropía de permutación ponderada (EPP). Este es el primer parámetro de salida de la función. Luego a modo de ensayo se utiliza el mismo vector de embedding para aplicar los algoritmos de cálculo de EP y EPU, siendo estos los otros parámetros de la salida de la función “*epQuincunx.m*”. Se debe notar que como en la mayoría de los algoritmos de la Propuesta 2, existen dos valores de embedding, uno es parámetro de entrada a la función y es utilizado para definir la dimensión de las submatrices con que se recorre la imagen; el otro valor de embedding es el que se utiliza para calcular el patrón de orden que representa cada una de esas submatrices, y así obtener los parámetros de salida EPP, EP y EPU en este caso. Este valor de embedding debe ser 3 en este algoritmo, ya que el vector de los valores representativos de la vecindad de Quincunx siempre tiene 3 elementos.

4. Materiales

La performance de estos algoritmos, es evaluada a través de imágenes de texturas sintéticas creadas para analizar características específicas de los algoritmos, un banco de imágenes reales de texturas elegidas al azar, para realiza pruebas sobre casos reales, y luego para un banco de datos del área de la medicina para evaluar su rendimiento práctico. El objeto de estudio práctico en este trabajo, fue el de la caracterización de la densidad mamaria mediante la aplicación de

los algoritmos desarrollados para EP en mamografías, obtenidas del banco de datos “*Digital Database for Screening Mammography*” (DDSM) (Heath et al. 2001). Se estudian sus resultados de los algoritmos aplicados sobre la región de interés de mamografías clasificadas según el método de clasificación BI-RADS, para definir posibles aplicaciones en la caracterización de la densidad mamaria. Los descriptores obtenidos fueron utilizados luego como datos de entrenamiento supervisado de una red neuronal multicapa, cuyos resultados permitieron cuantificar la performance del método para caracterizar el nivel de densidad mamaria.

En primera instancia se crearon dos imágenes sintéticas (lisa y rugosa) con el fin de realizar una primera selección de los algoritmos que mejor cuantifican el valor de Entropía de Permutación esperado en una imagen. Es decir, se crearon imágenes particulares, las cuales poseen un valor esperado de EP debido a sus características y la definición de entropía de permutación como un descriptor del nivel de orden de una serie, lo que traducido a imágenes implica un descriptor del nivel de rugosidad en texturas. Se considerara que las imágenes que representen una textura de rugosidad baja son lo más cercano a las superficies lisas, lo que implica que los niveles de gris son similares entre vecinos cercanos, esperando lo contrario en aquellas de rugosidad pronunciada.

Además se hacen pruebas con las mismas imágenes pero con ruido superpuesto o en su versión escalada, para comprobar la robustez de los métodos frente al ruido, y su invarianza a transformaciones.

Luego de evaluar los resultados de los métodos sometidos al análisis de imágenes sintéticas, sobre las que se conocía el resultado teórico esperado para EP, se procede a probar los algoritmos con 8 imágenes de superficies reales de distintas rugosidades. De este procedimiento se analizaron los resultados y se concluyó la batería de algoritmos de mejor performance, las que se consideran aplicables al caso práctico de caracterización de densidad mamaria en mamografías.

4.1 Imágenes sintéticas

Para analizar la performance de cada algoritmo, se crearon imágenes de los dos casos extremos, una totalmente lisa que representa una textura suave, otra totalmente rugosa, una que representa una textura gruesa y otra que representa una textura fina. Las imágenes lisa y rugosa ideales, son utilizadas con el fin de medir la performance de los algoritmos desarrollado, ya que de estas dos imágenes presentan valores de EP conocidos. Sin embargo las imágenes de textura gruesa y fina serán utilizadas para conocer el comportamiento del descriptor frente a estas variaciones en la textura.

Estas imágenes sintéticas son creadas con la cantidad de pixeles suficiente, para obtener al menos 10 muestras de cada uno de los bins pertenecientes al histograma de patrones de orden. La cantidad de ordenes posibles (bins), depende del valor de embedding adoptado, siendo *Cantidad de Bines* = $n!$. En este trabajo se realizan pruebas cuyo máximo valor de embedding es 7, por lo que se crean imágenes sintéticas con una cantidad mínima de 5040 datos, por lo tanto las matrices deben tener un tamaño mínimo de 71×71 para los algoritmos que calculen EP pixel a pixel, y multiplicar estas dimensiones por 7 para aquellos algoritmos que utilicen como datos a una submatriz cuadrada del tamaño del embedding, resultando una

imagen de 497×497 . Para salvar todas estas consideraciones se escogen matrices cuadradas de tamaño 500×500 para todos los casos.

- **Lisa:** todos los tonos de gris de igual valor, obteniendo un histograma de patrones de orden concentrado en un sólo bin. Se espera $EP \approx 0$.



Ilustración 30: Imagen sintética Lisa, representada por una matriz de tamaño 500x500, con todos sus píxeles del mismo valor.

- **Rugosa:** los tonos de gris tienen con distribución aleatoria, resultando un histograma de patrones de orden uniforme. Se espera $EP \approx 1$.

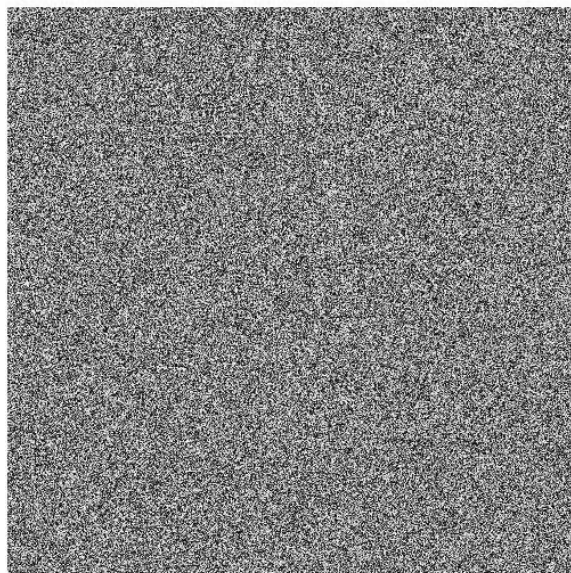
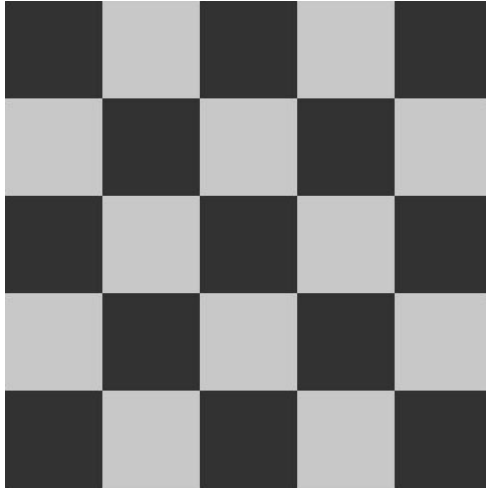


Ilustración 31: Imagen sintética Rugosa representada por una matriz de tamaño 500x500, cuyos píxeles tienen un valor aleatorio.

- **Gruesa:** se crean dos texturas gruesas sintéticas con diferente tamaño de patrón de textura.

Textura Gruesa 1
(Construida con matrices de 100x100)



Textura Gruesa 2
(Construida con matrices de 50x50)

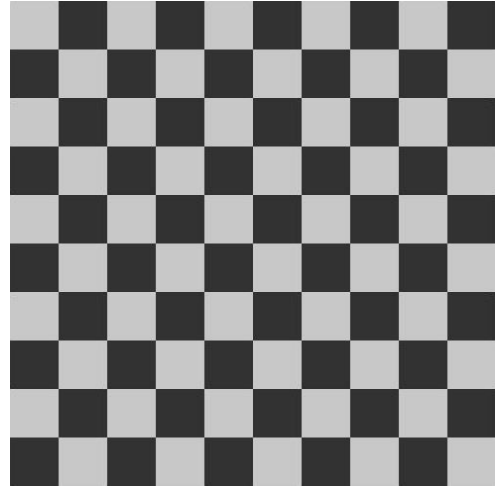
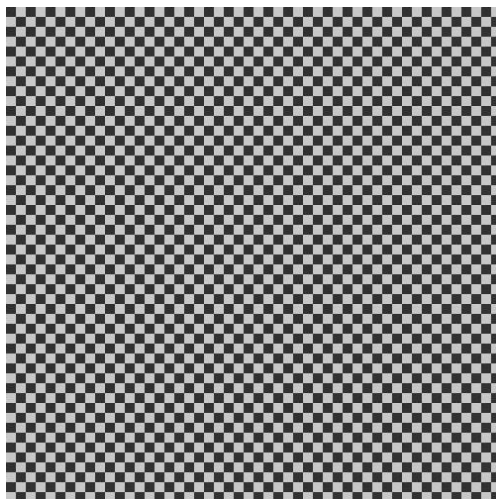


Ilustración 32: Texturas gruesas sintéticas

- **Fina:** se crean dos texturas finas sintéticas, que se diferencian por el tamaño de patrón de textura.

Textura Fina 1
(Construidas con matrices de 10x10)



Textura Fina 2
(Construidas con matrices de 5x5)

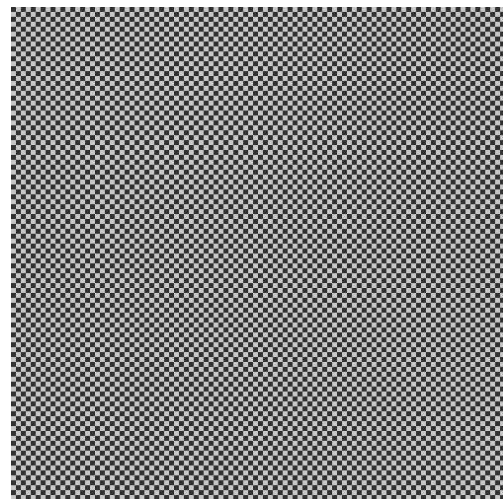


Ilustración 33: Texturas finas sintéticas

Se ensayaron todos los algoritmos desarrollados con las imágenes lisa y rugosa, aquellos descriptores que no presentan los valores el EP esperados fueron descartados para el resto de los ensayos, ya que se considera que aquí están los dos casos más representativos de lo que se

busca diferenciar con los métodos propuestos, una textura totalmente lisa y otra totalmente rugosa. Luego, se ensayan todos los algoritmos sobre las imágenes de textura gruesa y fina. Con los resultados se trazan gráficos que muestran las variaciones del valor de EP según el tamaño de la textura.

4.2 Imágenes de prueba Reales

Como una primera aproximación a la aplicación de los algoritmos en prácticas reales, se escogieron imágenes de texturas variadas, para analizar cómo responden los algoritmos de las distintas propuestas para categorizar y discriminan entre cada una de estas texturas, según el valor de la entropía de permutación de cada una de ellas.

Se escogieron 8 imágenes de texturas diferentes las que muestran en la Ilustración 34, entre las que existen lisas y rugosas. Se buscan imágenes que representen texturas con distintos grados de rugosidad, para comparar los resultados de los algoritmos con la percepción visual según las variaciones o irregularidades en una superficie continua.

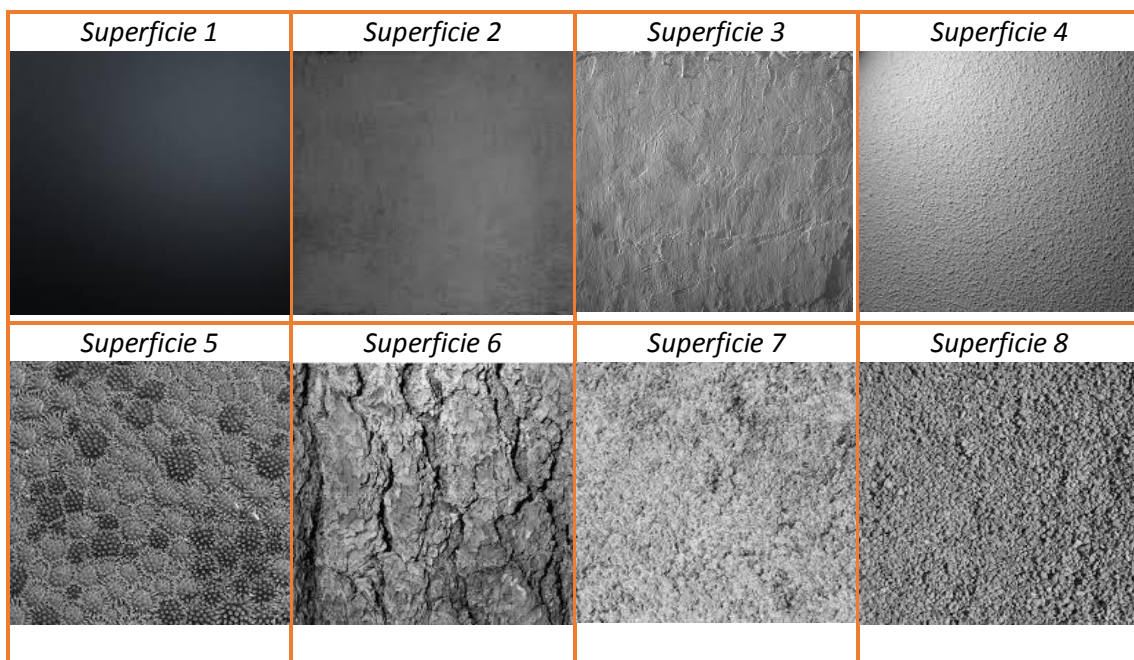


Ilustración 34: Imágenes de prueba para medir el rendimiento de los algoritmos. Ordenadas de más a menos rugosas según inspección visual.

Las imágenes se muestran ordenadas visualmente según su rugosidad, la que se considera una característica táctil de la imagen, y que puede variar entre superficies totalmente lisas (texturas suaves) y superficies totalmente rugosas (texturas ásperas). En este banco de imágenes, “Superficie 1” es considerada la más lisa, mientras que “Superficie 8” es la más rugosa. El objetivo de este orden, es comparar la clasificación visual intrínseca a la naturaleza de la superficie, con los resultados obtenidos en los ensayos de los algoritmos. De esta manera se pueden ordenar las texturas de menor a mayor rugosidad, según los valores de entropía de permutación

resultantes de los métodos desarrollados, donde el valor 0 de entropía representa a las texturas totalmente lisas y el 1 a las totalmente rugosas.

4.3 Mamografías

De todos los cánceres conocidos, el cáncer de mama es el que representa la primera causa de muerte por cáncer en mujeres argentinas, representado en un 20,3% de los casos totales de fallecimiento a causa de cualquier tipo de cáncer. Además, la República Argentina tiene la segunda tasa de mortalidad por cáncer de mama en el continente americano (21,8% por cada 100.000 mujeres) según datos oficiales del Ministerio de Salud (Anon n.d.). En una comparación global, Argentina se ubica dentro de los países con mayores tasas de incidencia y mortalidad por tumores malignos de mama según la Organización Mundial de la Salud (Antoni et al. 2016). El cáncer de mama es una enfermedad difícilmente prevenible y posee múltiples factores de riesgo: edad, sobrepeso, sedentarismo, antecedentes familiares, densidad mamaria, exposición a tratamientos hormonales. La modificación de estos factores requiere de un largo proceso y sus efectos tardarán varias décadas en evidenciarse. Por lo tanto, los esfuerzos para lograr el control de la enfermedad, en un término más corto, deben centrarse en la detección precoz y la implementación de tratamientos pertinentes (Viniestra et al. 2010).

La densidad mamaria es la proporción de tejido fibroglandular con relación a la cantidad de tejido adiposo en la composición de una mama. Muchos estudios indican que se presenta como un indicador importante ante la potencialidad del desarrollo de un cáncer de mama (Yaghjian et al. 2015) (Paulina Neira 2013). Este predictor, posee la ventaja de ser detectado sobre mamografías, siendo éstas el recurso primario de control epidemiológico del cáncer de mama. Además, dentro de la legislación de muchos países se utiliza como disparador único e independiente para la realización de monitoreo e intervenciones más complejas, como ecografías o resonancias magnéticas, en pacientes que presentan un grado alto de densidad, ya que el problema en la densidad mamográfica no es solo el aumento del riesgo de cáncer de mama, sino también una marcada disminución de la sensibilidad de la mamografía frente a la detección por simple inspección de nódulos. Estos problemas fueron y continúan siendo investigados en una sustancial cantidad de trabajos, donde se evidencia que las mujeres con tejido denso en el 75% o más, tienen un riesgo de cáncer de mama cuatro a seis veces mayor que el riesgo entre las mujeres con poco o ningún tejido denso (Boyd et al. 2007), y que la sensibilidad de las mamografías disminuye linealmente según el aumento de densidad (Aibar et al. 2011).

Existen diferentes métodos para medir el grado de densidad mamaria. El más frecuente en su uso es la clasificación BI-RADS (Ruj 2016), donde la densidad presenta cuatro diferentes niveles que se diferencian en el porcentaje de tejido fibroglandular que constituye la mama. Estos niveles son indicados subjetivamente por un experto, mediante inspección visual de la mamografía. Diversos estudios indican que la clasificación de densidad realizada por expertos no evidencia concordancia con la obtenida por algunos conocidos sistemas de medición automática (Ruiz et al. 2014).

Se considera de interés obtener una medición de densidad mamaria cada vez más precisa, buscando la posibilidad de utilizar este predictor de manera individual y hallar métodos e indicadores adecuados que permitan cuantificar de manera certera el tipo de tejido en una

mamografía. Aunque aún no se ha creado ningún programa informático que cumpla con todas las características ideales para la evaluación de las imágenes, es muy recomendable la utilización de alguno que brinde datos mínimos, sin distraer al radiólogo en la evaluación de mamografía.

El atlas del sistema de informes y registro de datos de estudios por imágenes de la mama (BI-RADS®) del Colegio Estadounidense de Radiología (ACR, American College of Radiology) es producto del trabajo conjunto de los integrantes de varios comités. Este sistema propone un método normalizado para informar los estudios por imágenes de la mama, específicamente en mamografías, ecografías y resonancias magnéticas. Sin embargo, es tarea del médico solicitante y el médico radiólogo tomar la decisión definitiva en cuanto a los procedimientos de diagnóstico y el tratamiento, a la luz de todas las circunstancias que presente cada estudio en particular. La redacción de un informe de mamografía se realiza según el vocabulario definido por BI-RADS. En la Tabla 7 se resumen las características que se enumeran en un informe acorde al sistema adoptado por BI-RADS.

Incumbencia	Característica
<i>Tejido mamario</i>	Composición de la mama
<i>Hallazgos</i>	A. Nódulos
	B. Calcificaciones
	C. Distorsión de la arquitectura
	D. Asimetrías
	E. Ganglio linfático inflamado
	F. Lesión cutánea
	G. Conducto dilatado solitario
	H. Hallazgos asociados
	I. Ubicación de la lesión

Tabla 7: Resumen del vocabulario para las características presentes en el sistema de informes BI-RADS

Cada una de las características presentes en las mamografías, debe ser descrita en el informe según la terminología específica determinada por el sistema de informes de BI-RADS. Este trabajo solo considera la característica presente del tejido mamario, por lo que se considera de importancia nombrar la terminología que indica el vocabulario utilizado por BI-RADS en esta incumbencia, descrito en la siguiente tabla:

Incumbencia	Característica	Terminología
<i>Tejido mamario</i>	<i>Composición de la mama</i>	<p>A. Las mamas están compuestas por tejido adiposo casi en su totalidad.</p> <p>B. Se observan sectores dispersos de densidad fibroglandular.</p> <p>C. Las mamas son heterogéneamente densas, que puede ocultar algunos nódulos pequeños.</p> <p>D. Las mamas son muy densas, lo que disminuye la sensibilidad mamográfica.</p>

Tabla 8: Terminología para describir el tejido mamario en el sistema de informes BI-RADS.

La terminología para la descripción del tipo de composición del tejido mamario no hace referencia a hallazgos mamográficos aislados, sino al grado de atenuación radiográfica que presenta el tejido mamario, perteneciendo las zonas oscurecidas de mamografía al tejido adiposo, mientras que el tejido fibroglandular se observa en placas blanquecinas. La sensibilidad mamográfica para detectar las lesiones no calcificadas disminuye conforme aumenta la categoría de la densidad, es decir, cuanto más densa es la mama, más grande es la lesión que puede quedar oculta. Las imágenes mostradas en la Ilustración 35, son ejemplos de la clasificación según la terminología descrita en la Tabla 8.

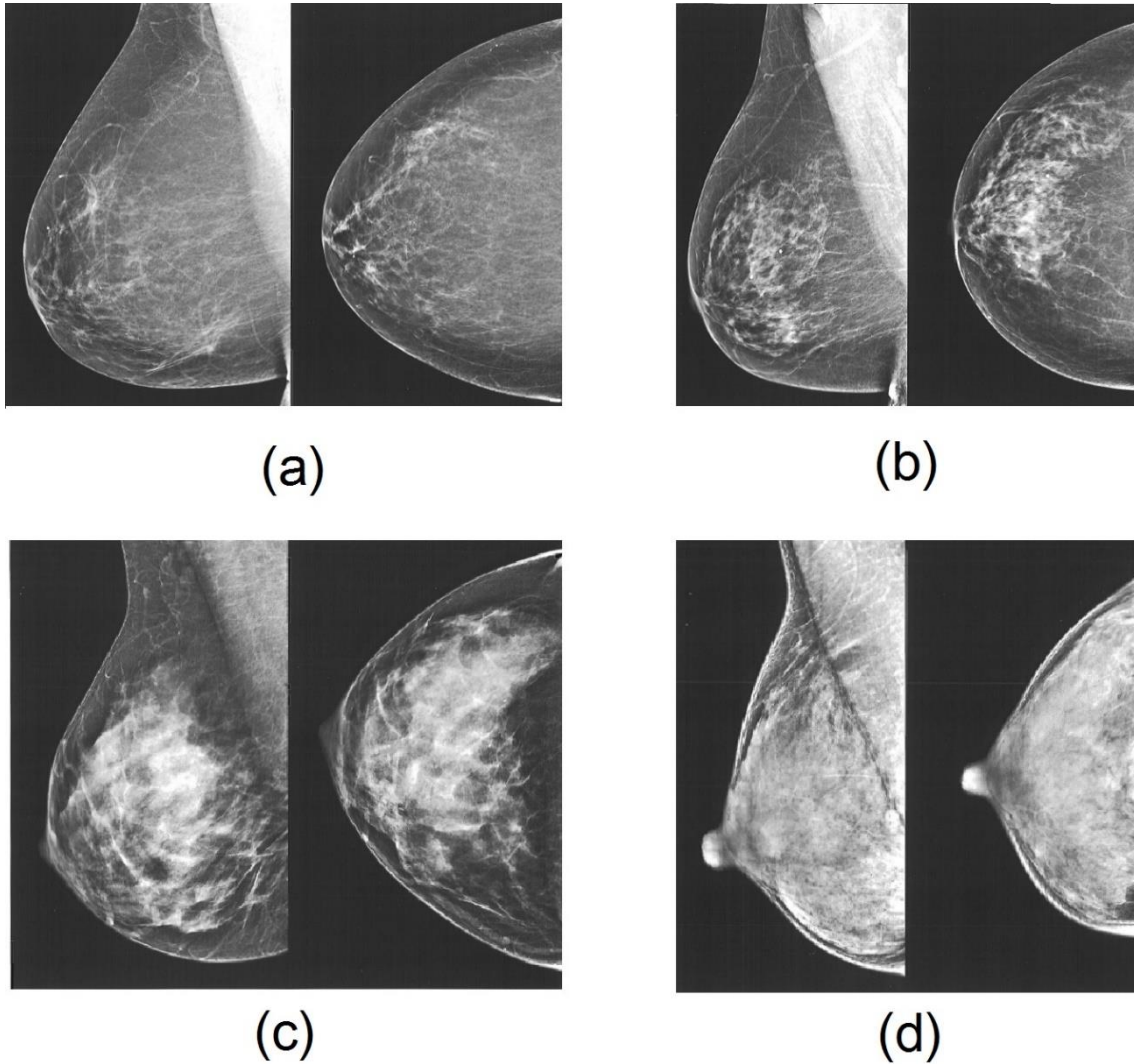


Ilustración 35: Categorización de la densidad mamográfica según el sistema de informes BI-RADS. a) contenido adiposo casi total, b) sectores dispersos de contenido fibroglandular. c) densidad heterogénea, d) densidad alta, insensibilidad del estudio mamográfico. Imágenes obtenidas de la 5ta edición del Atlas BI-RADS.

La quinta y, a la fecha, última edición del Atlas BI-RADS ya no describe las cuatro categorías de densidad en intervalos de cuartiles porcentuales (a incrementos del 25% de tejido denso), sino que deja esta clasificación al análisis subjetivo del radiólogo. Este cambio, obedece al propósito de hacer hincapié en la importancia clínica de utilizar a las descripciones de la densidad como reflejo del efecto de enmascaramiento de lesiones que tiene una mama densa, para que las pacientes sean clasificadas dentro de dos grandes grupos: aquellas que requieren de monitoreo y estudios más complejos y las que no.

Sin embargo, muchas investigaciones realizadas últimamente y otras muchas en curso, indican la relación que puede tener el porcentaje de densidad mamaria como indicador del riesgo de cáncer de mama. En este sentido, BI-RADS indica que los radiólogos seguirán utilizando las categorías de la densidad en los informes de mamografía como lo han venido haciendo a lo largo de los últimos años, pero igualmente se esperan publicaciones de datos más robustas acerca de la densidad mamaria en función del volumen de la mama. Se espera que estos trabajos definan umbrales porcentuales validados (no necesariamente cuartiles) que sean fáciles de determinar y reproducibles con métodos de diagnóstico, para volver a incluir los intervalos porcentuales correspondientes a cada categoría de la densidad.

Frente a la necesidad de descriptores más robustos para continuar con las investigaciones sobre implicancia de la densidad mamaria como indicador de riesgo de cáncer de mamas, ante la nueva ausencia de referencias numéricas y justificadas de la categorización de la densidad mamográfica según las organizaciones de regulación del sistema de informes mamográficos BI-RADS, los descriptores basados en EP se presentan como una alternativa a continuar estudiando para la detección de diferentes densidades en mamografías.

Se planteó como hipótesis que se obtienen resultados de EP similares para las densidades de los extremos (1 y 4 para la terminología antigua, *a* y *d* según la nueva terminología), ya que ambas se caracterizan por tener valores de gris similares entre vecindades, lo que representa un nivel de entropía bajo de la imagen. La misma hipótesis se planteó para las mamografías de densidades medias (2 y 3 para la terminología antigua, *b* y *c* según la nueva terminología), ya que tal como se ve en la Ilustración 35, los niveles de gris se presentan con grandes variaciones entre vecinos cercanos, por lo tanto, un nivel de entropía más alto representaría a estas mamografías. En este trabajo se utilizó la terminología antigua, ya que era la que presentaba el banco de datos utilizado.

Se obtuvieron 168 mamografías del banco de datos DDSM, para casos normales (sin tumores, benignos ni malignos). Se decidió considerar éstas debido a que las imágenes de mamas con algún tipo de tumor contienen marcas manuales para indicar las zonas afectadas, lo que imposibilita considerarlas parte del banco de datos para su procesamiento. Por otro lado, en este trabajo sólo se hace hincapié en un análisis del tejido mamario general, sin particularizar en el tejido nodular.

Las imágenes fueron digitalizadas por diferentes scanners (DBA, Howtek y Lumysis) con diferentes niveles de resolución (42 micrones, 43,5 micrones y 50 micrones). Todas las mamografías poseen su informe pertinente, describiendo el tejido mamario y los hallazgos según un especialista mediante el sistema de informes BI-RADS. La descripción del tejido mamario se realiza según la tipología numérica antigua de BI-RADS, donde el mismo es clasificado según niveles numéricos de densidad, representados según porcentajes del contenido denso durante la lectura mamográfica en: 1. mamas de predominio graso (<25%), 2. patrón de densidades glandulares dispersos (25 a 50%), 3. heterogéneas (51 a 75%) y 4. extremadamente densas (>75%). Las mamografías se seleccionaron según la densidad indicada en los informes realizados por diversos expertos radiólogos, y corresponden a ambas mamas del mismo paciente, cada una adquirida desde las dos proyecciones básicas: cráneo-caudal y oblicuo mediolateral, resultando un total de 4 mamografías por cada paciente. Las imágenes obtenidas desde el banco de datos corresponden a 40 mamografías con Densidad 1, 40 con Densidad 2, 44 con Densidad 3 y 44 con Densidad 4.

Se seleccionó de cada mamografía la región de interés de estudio (tejido glandular), obteniéndose imágenes de tamaños variados de aproximadamente 70x100 píxeles. Estas últimas son las que finalmente se utilizan para su procesamiento.

Mediante pruebas la región de interés, se seleccionaron los mejores descriptores de EP, con el criterio de valor del parámetro n donde la entropía difiriera más marcadamente entre las distintas densidades, o donde se pueda distinguir entre los conjuntos de densidades de los que se esperan valores de EP diferentes. Se pretende que cada una de las mamografías de estudio, se representen por las mejores características brindadas por los algoritmos desarrollados. Así, se puede formar un vector característico para imagen, del que se espera una representación unívoca.

Se aplicaron los algoritmos propuestos sobre las 168 mamografías. Los parámetros de entrada fueron en un principio estudiados heurísticamente: retardo de 1 y embedding (n) entre 3 y 7. Sin embargo, se observó que con ciertos algoritmos se obtuvieron mejores resultados para embeddings mayores a 7, donde el rango dinámico de los resultados aumentaba en proporción con el aumento del tamaño de n utilizado. Por este motivo, se utilizaron para todos los algoritmos un retardo de 1 y valores de embedding entre 3 y 10.

Una vez obtenido el vector característico para cada mamografía, se propone realizar el entrenamiento de una red neuronal multicapa como clasificador, con el fin de evaluar la potencialidad de los descriptores como caracterizadores de la densidad mamográfica, en concordancia con la nomenclatura BI-RADS.

5. Resultados y discusión

Se muestran los resultados de cada uno de los algoritmos ensayados sobre los tres bancos de datos, y se analizan los resultados y los problemas encontrados en cada uno de ellos. Un buen desempeño de los métodos, garantiza resultados fiables en la aplicación de los algoritmos en casos prácticos, por lo tanto el análisis del comportamiento de los algoritmos sobre las imágenes sintéticas artificiales tiene vital importancia, ya que se conocen los resultados esperados, lo que ayuda a realizar un estudio preliminar de selección de los algoritmos más robustos, que serán aplicables luego a casos reales.

En la sección de imágenes artificiales, también se observa el comportamiento de los distintos métodos frente a texturas de distintos tamaños. De esta manera no sólo se analiza la capacidad de los descriptores para diferenciar entre imágenes de diferentes rugosidades, sino también como posibles caracterizadores entre texturas gruesas y finas.

Las pruebas con imágenes reales, se realizan con aquellos algoritmos que tuvieron mejor performance para las imágenes sintéticas lisa y rugosa. Aquí se observa la capacidad de caracterización de rugosidad para distintas imágenes de superficies texturales. Además, se realizan pruebas con las imágenes algunas de las imágenes reales en su versión escalada y otras pruebas con las mismas particionadas.

Por últimos, se muestran los resultados obtenidos para el caso de aplicación del descriptor para caracterizar la densidad mamaria en mamografías. Aquí se puede analizar una primera aproximación del método como posible solución de un caso práctico en el área médica. Para ello se realizaron redes neuronales de distintas arquitecturas, analizando los porcentajes de error de categorización y de generalización de las mismas, lo que traduce la efectividad de los descriptores realizados para solucionar la problemática propuesta.

5.1 Imágenes artificiales. Textura Lisa y Rugosa

En base al análisis de los resultados obtenidos de todos los métodos desarrollados en las propuestas 1 y 2, aplicados a las imágenes lisa y rugosa, con valores de *embedding* entre $3 \leq n \leq 7$. Se seleccionaron aquellos algoritmos cuya performance permita distinguir entre las

superficies ideales rugosa y lisa, para ello los métodos que deben entreguen valores de entropía cercanos a cero (0) para el caso de la imagen sintética lisa, mientras que para la imagen rugosa se espera un valor de entropía uno (1) o cercano. Los algoritmos de performance satisfactoria se aplicaron luego sobre las imágenes de prueba reales.

Los resultados obtenidos son mostrados en figuras y tablas. Las figuras muestran los valores obtenidos para cada algoritmo, dependiente del valor adoptado por el parámetro *embedding*. A través de las figuras se otea de manera rápida la performance de cada método, lo que luego se corrobora con los valores detallados en las tablas, para arribar en la conclusión de si el algoritmo será incluido o no para las aplicaciones prácticas sobre imágenes reales que se ensayan luego.

5.1.1 Algoritmos mediante análisis de la imagen dispuesta en un vector unidimensional

Los métodos desarrollados en esta primera propuesta, son utilizados luego por los algoritmos de la propuesta 2. Se puede decir entonces, que estos son los algoritmos de *base*, y sus resultados condicionan la respuesta de todos los métodos que siguen.

Algoritmos de Propuesta 1							
Imagen	<i>n</i>	<i>EP</i>	<i>EPP</i>	<i>EPT</i>	<i>EPTp</i>	<i>EPU</i>	<i>EUp</i>
<i>Lisa</i>	3	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	4	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	5	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	6	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	7	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Imagen	<i>n</i>	<i>EP</i>	<i>EPP</i>	<i>EPT</i>	<i>EPTp</i>	<i>EPU</i>	<i>EUp</i>
<i>Rugosa</i>	3	0,9999	0,9999	0,9046	0,9546	0,9289	0,9683
	4	0,9999	0,9999	0,9328	0,9475	0,9516	0,9645
	5	0,9999	0,9999	0,9472	0,9522	0,9647	0,9692
	6	0,9997	0,9997	0,9555	0,9575	0,9705	0,9722
	7	0,9988	0,9987	0,9597	0,9603	0,9738	0,9743
Algoritmos elegidos		✓	✓	✓	✓	✓	✓

Tabla 9: Resultado de los algoritmos de la Propuesta 1 para las imágenes lisa y rugosa

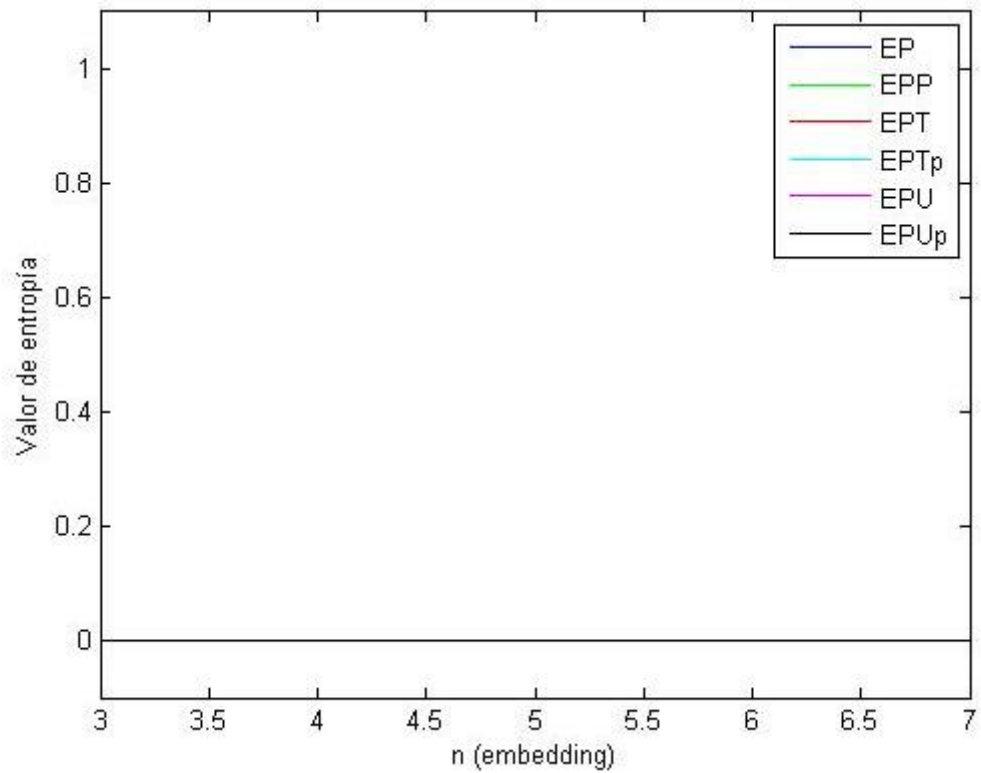


Ilustración 36: Resultado de los algoritmos de la propuesta 1 para la imagen sintética lisa, para valores de embedding entre 3 y 7

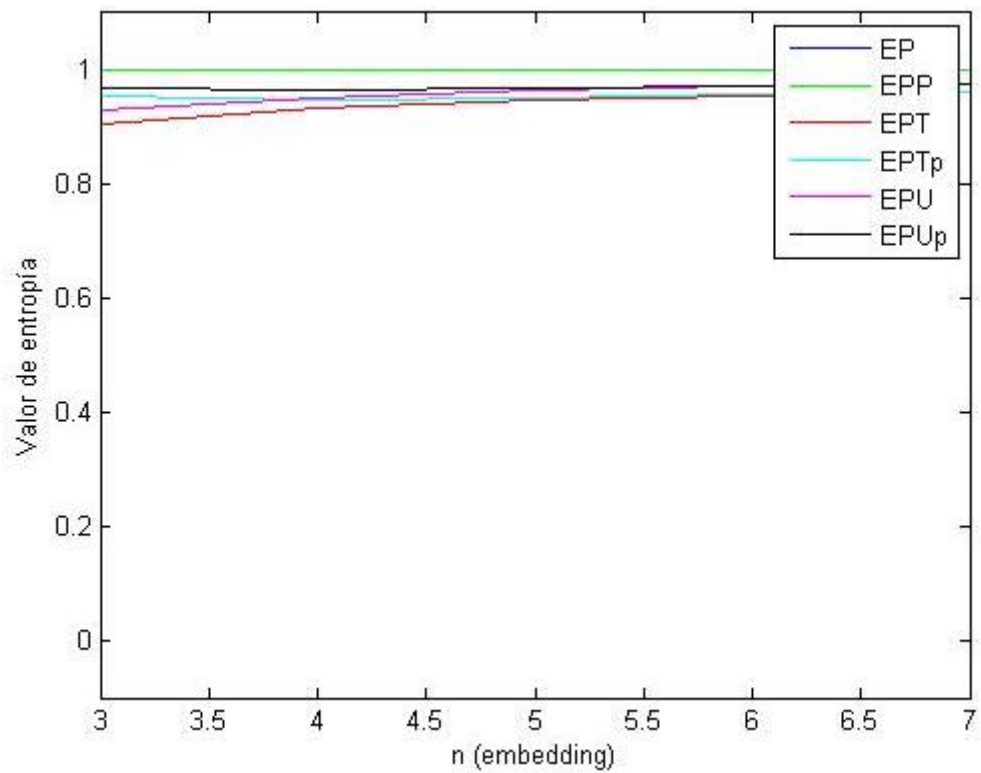


Ilustración 37: Resultado de los algoritmos de la propuesta 1 para la imagen sintética rugosa, para valores de embedding entre 3 y 7

Observación: para las imágenes lisa y rugosa sintéticas, se obtienen los resultados esperados de todos los algoritmos. Estos métodos, en cuya propuesta la imagen se dispone en un vector unidimensional, son utilizados para desarrollar los algoritmos siguientes, con los cuales se calcula el valor de entropía de la imagen, considerando al pixel y su entorno cercano, por lo tanto se utilizaron todos los algoritmos de esta propuesta para los cálculos de la propuesta siguiente, ya que se considera que los resultados obtenidos son satisfactorios.

5.1.2 Algoritmos mediante análisis de la imagen por el pixel y su vecindad

Los resultados se muestran en gráficos para comparar la respuesta de cada método en sus diferentes versiones. Se muestran doce figuras en total, seis de ellas son para comparar los métodos aplicados a la imagen lisa y las seis restantes para la imagen rugosa. Además se exponen los resultados en tablas para observar en detalle los valores obtenidos, y concluir cuáles son los algoritmos que presentan valores de EP coherentes con el tipo de rugosidad.

5.1.2.1 EP con imágenes filtradas

La Tabla 10 junto a la Ilustración 38 y Ilustración 39, muestran los resultados obtenidos para las imágenes lisa y rugosa, habiendo sido estas previamente filtradas por un filtro de mediana, generado con la función “*medfilt2 (imagen, [n , n])*” de Matlab. Este comando genera un filtro del tamaño del valor de *embedding*, el que se aplica sobre cada uno de los píxeles de las imágenes a procesar, reemplazando el valor del pixel por la mediana entre sus *n* píxeles vecinos.

Algoritmos de Propuesta 2. Entropía de permutación con imágenes filtradas (Mediana)							
Imagen	<i>n</i>	EP	EPP	EPT	EPT_p	EPU	EPU_p
<i>Lisa</i>	3	0,0001	0,6040	0,0001	0,6040	0,0001	0,6040
	4	0,0075	0,4200	0,0074	0,4196	0,0075	0,4156
	5	0,0002	0,3800	0,0002	0,3856	0,0002	0,3800
	6	0,0061	0,2714	0,0061	0,2790	0,0061	0,2745
	7	0,0002	0,2800	0,0002	0,2830	0,0002	0,2798
Imagen	<i>n</i>	EP	EPP	EPT	EPT_p	EPU	EPU_p
<i>Rugosa</i>	3	0,9220	0,9337	0,5308	0,8527	0,5036	0,6484
	4	0,9204	0,8976	0,4727	0,6927	0,4463	0,5691
	5	0,8680	0,8484	0,4209	0,6211	0,4462	0,5618
	6	0,8706	0,8265	0,3824	0,5345	0,4159	0,5234
	7	0,8292	0,7881	0,3435	0,4952	0,4008	0,4993
Algoritmos elegidos		✓	X	X	X	X	X

Tabla 10: Resultado de los algoritmos EP, EPP, EPT, EPT_p, EPU y EPU_p para imágenes filtradas con un filtro de Mediana para las imágenes lisa y rugosa

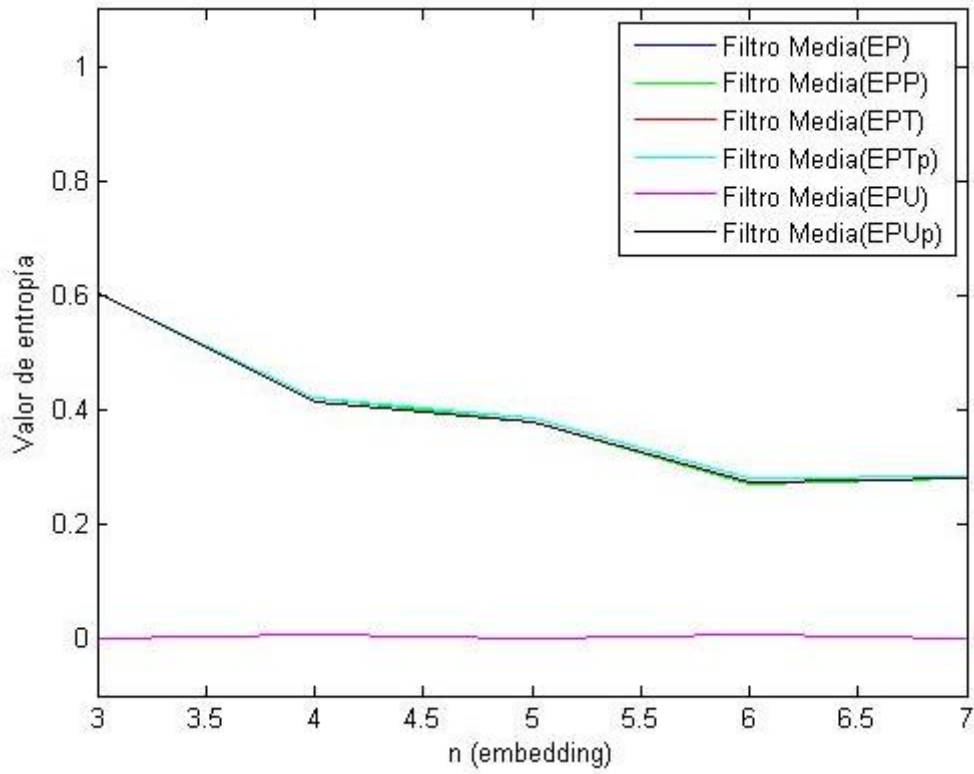


Ilustración 38: Resultados de los algoritmos aplicando pre filtrado de la imagen lisa con filtro de Mediana, para valores de embedding de 3 a 7.

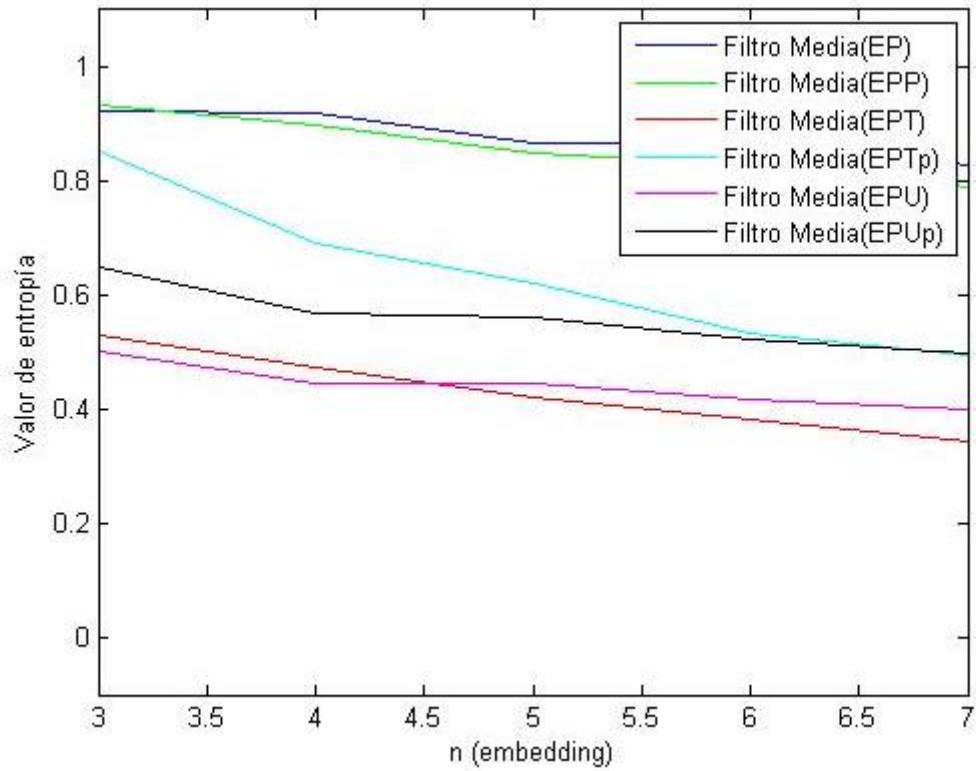


Ilustración 39: Resultados de los algoritmos aplicando pre filtrado de la imagen rugosa con filtro de Mediana, para valores de embedding de 3 a 7.

La Tabla 11 junto a la Ilustración 40 y Ilustración 41, muestran los resultados obtenidos para las imágenes lisa y rugosa filtradas por un filtro de mediana aritmética generado con “*fspecial ('average', n)*” de Matlab y aplicado sobre las imágenes a procesar mediante el comando “*imfilter (imagen , filtro)*”. Al igual que el caso anterior, se genera un filtro de tamaño del valor de *embedding*, que se aplica sobre cada uno de los píxeles de las imágenes a procesar, reemplazando el valor del pixel por la mediana en este caso aritmética entre sus *n* píxeles vecinos.

Algoritmos de Propuesta 2. Entropía de permutación con imágenes filtradas (Media aritmética)							
Imagen	<i>n</i>	<i>EP</i>	<i>EPp</i>	<i>EPT</i>	<i>EPTp</i>	<i>EPU</i>	<i>EPUp</i>
<i>Lisa</i>	3	0,0162	0,5815	0,0160	0,5803	0,0000	0,0021
	4	0,0166	0,5078	0,0074	0,4196	0,0050	0,2000
	5	0,0151	0,3313	0,0120	0,2894	0,0091	0,2519
	6	0,0148	0,3122	0,0116	0,3000	0,0104	0,2700
	7	0,0136	0,2308	0,0135	0,2345	0,0102	0,2012
Imagen	<i>n</i>	<i>EP</i>	<i>EPp</i>	<i>EPT</i>	<i>EPTp</i>	<i>EPU</i>	<i>EPUp</i>
<i>Rugosa</i>	3	0,9670	0,9303	0,2855	0,8337	0,2194	0,3338
	4	0,9373	0,8677	0,4297	0,6420	0,1882	0,3094
	5	0,9132	0,8198	0,2017	0,5655	0,1865	0,3204
	6	0,8913	0,7861	0,3286	0,4752	0,1721	0,3055
	7	0,8721	0,7575	0,1578	0,4391	0,1646	0,2989
Algoritmos elegidos		✓	✗	✗	✗	✗	✗

Tabla 11: Resultado de los algoritmos *EP*, *EPp*, *EPT*, *EPTp*, *EPU* y *EPUp* para imágenes filtradas con un filtro de Mediana Aritmética para las imágenes lisa y rugosa

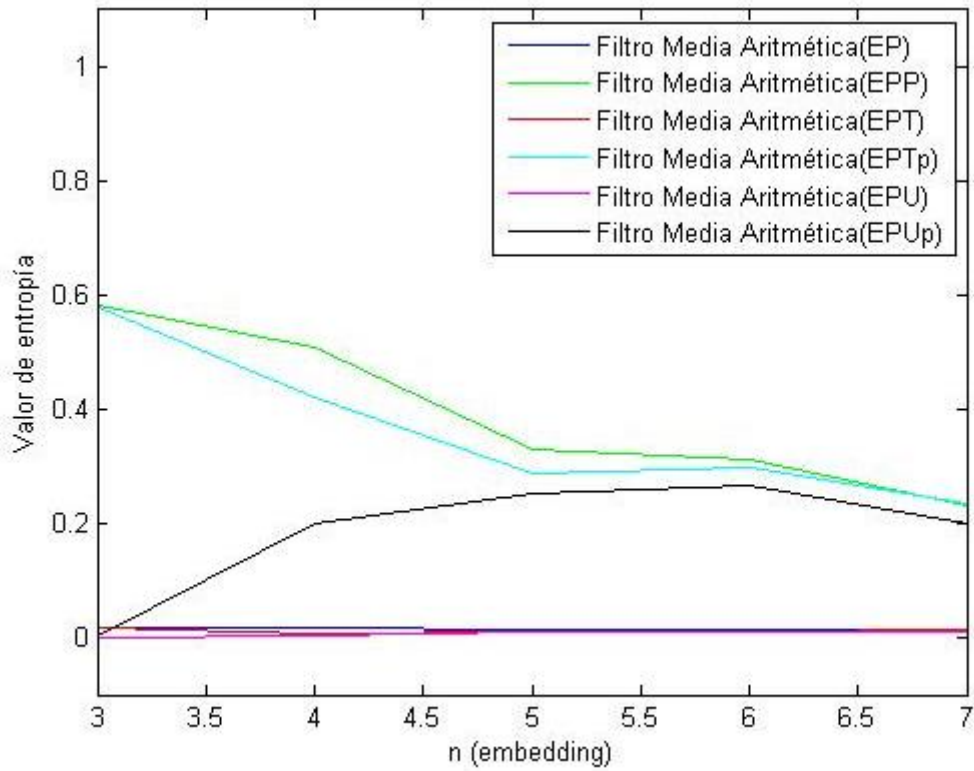


Ilustración 40: Resultados de los algoritmos aplicando pre filtrado de la imagen lisa con filtro de Mediana Aritmética, para valores de embedding de 3 a 7.

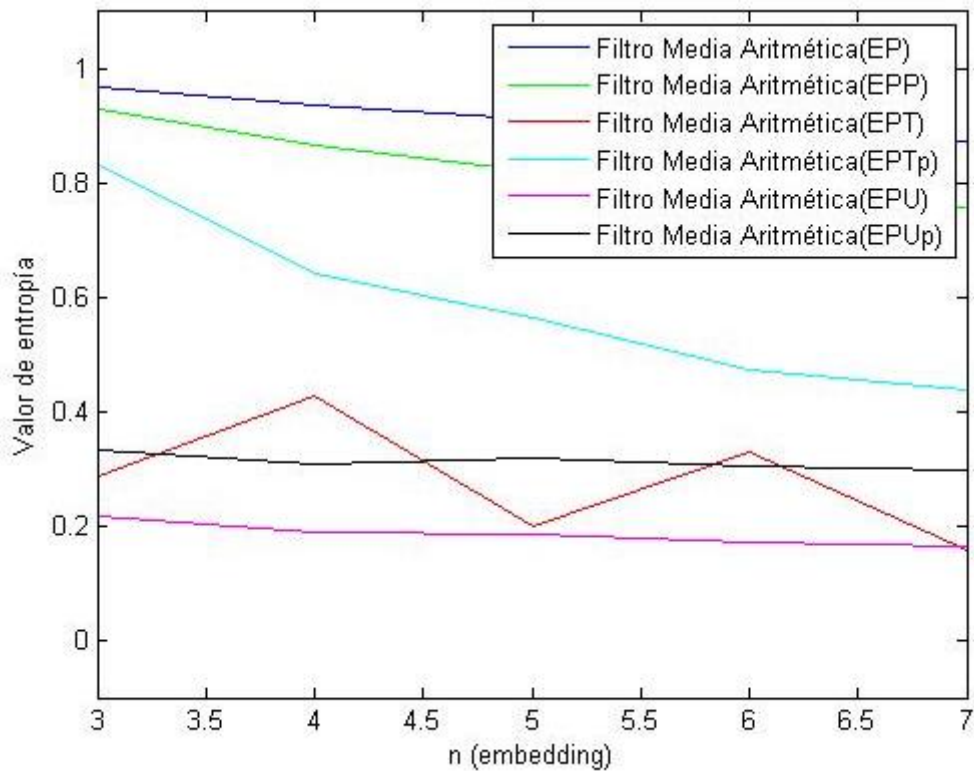


Ilustración 41: Resultados de los algoritmos aplicando pre filtrado de la imagen rugosa con filtro de Mediana Aritmética, para valores de embedding de 3 a 7.

El último filtro aplicado como pretratamiento de las imágenes, es un filtro gaussiano, generado de con Matlab a través de la función “*fspecial ('gaussian', n, desviación)*”. Este filtro de tamaño n cuya desviación estándar se determina para que no posea lóbulos secundarios, se aplica sobre cada uno de los píxeles de la imagen a procesar, y afecta a cada uno de ellos por la amplitud de la campana de gauss centrada en el píxel a analizar. Dicha acción se ejecuta mediante el comando “*imfilter (imagen, filtro)*”. Los resultados de ambas imágenes se exponen en la Tabla 12, en la Ilustración 42 se muestran los distintos valores que adquiere la entropía de la imagen lisa con respecto a la variación del *embedding*, mientras que en la Ilustración 43 se realiza lo mismo con la imagen rugosa.

Algoritmos de Propuesta 2. Entropía de permutación con imágenes filtradas (Gauss)							
Imagen	n	EP	EPP	EPT	EPTp	EPU	EUPp
<i>Lisa</i>	3	0,0161	0,5844	0,0001	0,6040	0,0000	0,0025
	4	0,0166	0,5071	0,0074	0,4196	0,0075	0,3337
	5	0,0151	0,3347	0,0121	0,3332	0,0120	0,3153
	6	0,0148	0,3138	0,0116	0,2998	0,0128	0,3189
	7	0,0136	0,2339	0,0135	0,2345	0,0119	0,2311
Imagen	n	EP	EPP	EPT	EPTp	EPU	EUPp
<i>Rugosa</i>	3	0,9179	0,8342	0,3092	0,8139	0,2569	0,3761
	4	0,8831	0,7658	0,3865	0,5863	0,1843	0,3123
	5	0,8018	0,6480	0,1936	0,5141	0,1865	0,3269
	6	0,8173	0,6664	0,2825	0,4168	0,1630	0,3013
	7	0,7583	0,5956	0,1422	0,3931	0,1557	0,2931
Algoritmos elegidos		✓	✗	✗	✗	✗	✗

Tabla 12: Resultado de los algoritmos EP, EPP, EPT, EPT p , EPU y EUP p para imágenes filtradas con un filtro de Gauss para las imágenes lisa y rugosa

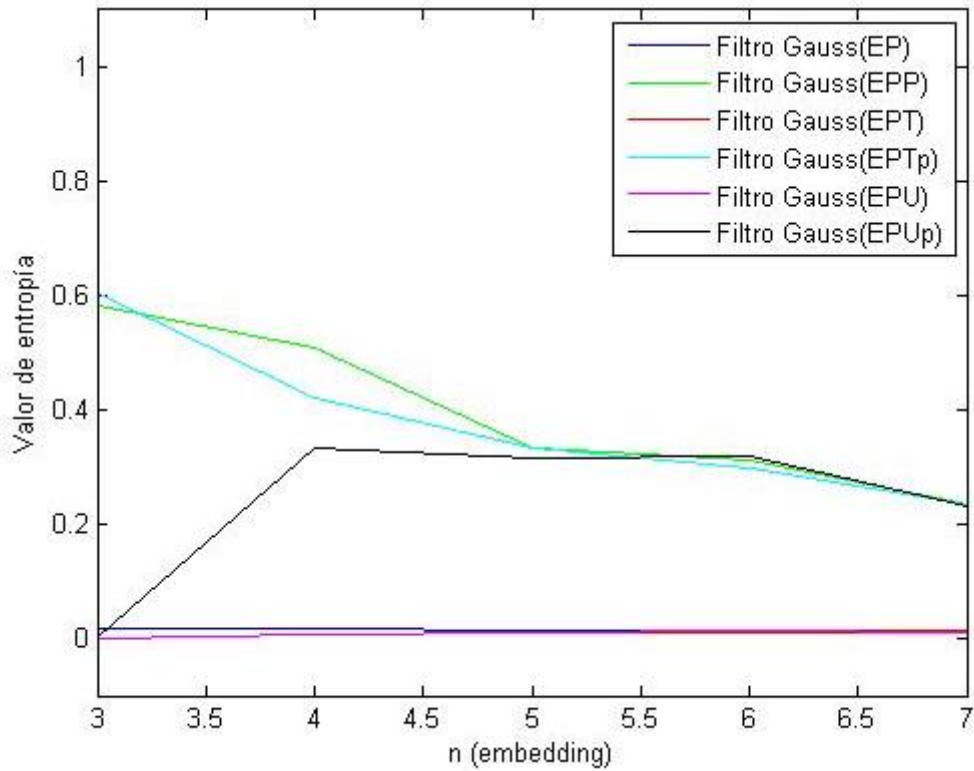


Ilustración 42: Resultados de los algoritmos aplicando pre filtrado de la imagen lisa con filtro de Gauss, para valores de embedding de 3 a 7.

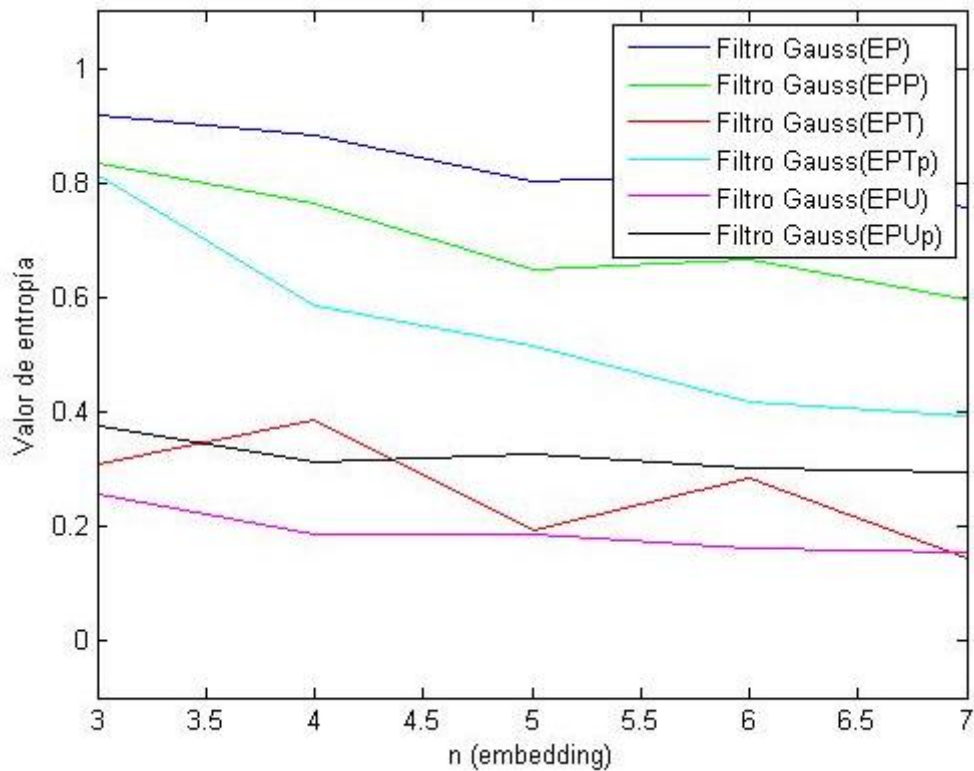


Ilustración 43: Resultados de los algoritmos aplicando pre filtrado de la imagen rugosa con filtro de Gauss, para valores de embedding de 3 a 7.

Observaciones: la aplicación de filtros preestablecidos por Matlab a las imágenes, cambia los valores de los píxeles de las imágenes, provocando una pérdida de la información de la textura relativa. En todos los casos se aplicaron filtros de “suavizado” debido a su característica de pasa bajos. Si bien estos filtros son muy usados para eliminar el ruido aleatorio presente en las altas frecuencias de la imagen, en este caso se utiliza para obtener una representación del entorno el píxel que guarde la información de la posición en la que ello ocurre. En este sentido, se observa que todos los descriptores presentan un valor de entropía menor al caso de sin filtrado de la imagen rugosa, lo que era de esperar por el efector de alisamiento que ejercen los filtros sobre la imagen, esfumando los detalles de la imagen, achicando las variaciones abruptas entre píxeles vecinos. También se observa, que todos los métodos que utilizan la técnica de EPP presentan grandes variaciones en la entropía de la imagen lisa filtrada respecto a la misma sin filtrar. Esto indica que este método puede presentar grandes variaciones frente a pequeños cambios de la imagen. Por último, en todos los casos se mantiene la premisa de que la imagen lisa presenta menor energía que la imagen rugosa.

5.1.2.2 EP de valores medios

La Tabla 13 junto a la Ilustración 44 y Ilustración 45, muestran los resultados obtenidos para las imágenes lisa y rugosa mediante la implementación del algoritmo EP de Valores Medios.

Propuesta 2. Entropía de permutación de valores medios					
Imagen	n	EP	EPP	EPU	EPUp
<i>Lisa</i>	3	0,0004	0,5796	0,0004	0,5808
	4	0,0006	0,3894	0,0006	0,3894
	5	0,0008	0,2867	0,0008	0,2881
	6	0,0010	0,2254	0,0010	0,2254
	7	0,0013	0,1841	0,0013	0,1841
Imagen	n	EP	EPP	EPU	EPUp
<i>Rugosa</i>	3	0,0221	0,9950	0,0081	0,8012
	4	0,0218	0,9530	0,0098	0,6991
	5	0,0218	0,8546	0,0119	0,6046
	6	0,0225	0,7445	0,0131	0,5080
	7	0,0227	0,6238	0,0145	0,4284
Algoritmos elegidos		X	X	X	X

Tabla 13: Resultado de los algoritmos EP, EPP, EPU y EPUp para Matriz de Valores Medios para las imágenes lisa y rugosa

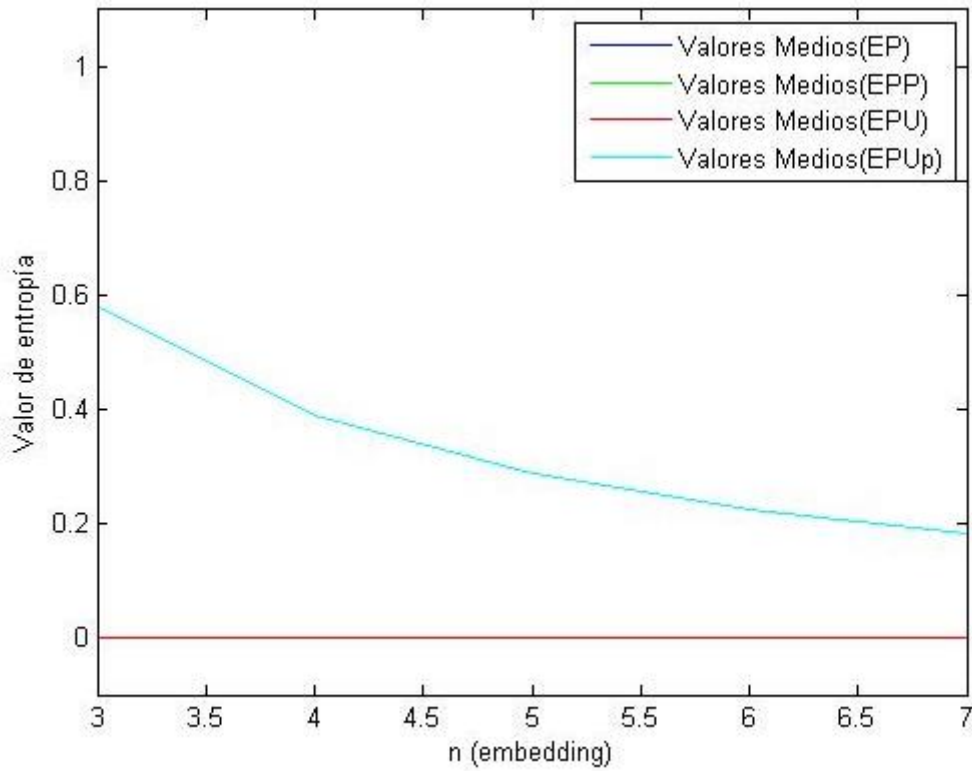


Ilustración 44: Resultados de EP para matriz de Valores Medios de la imagen lisa, para valores de embedding entre 3 y 7

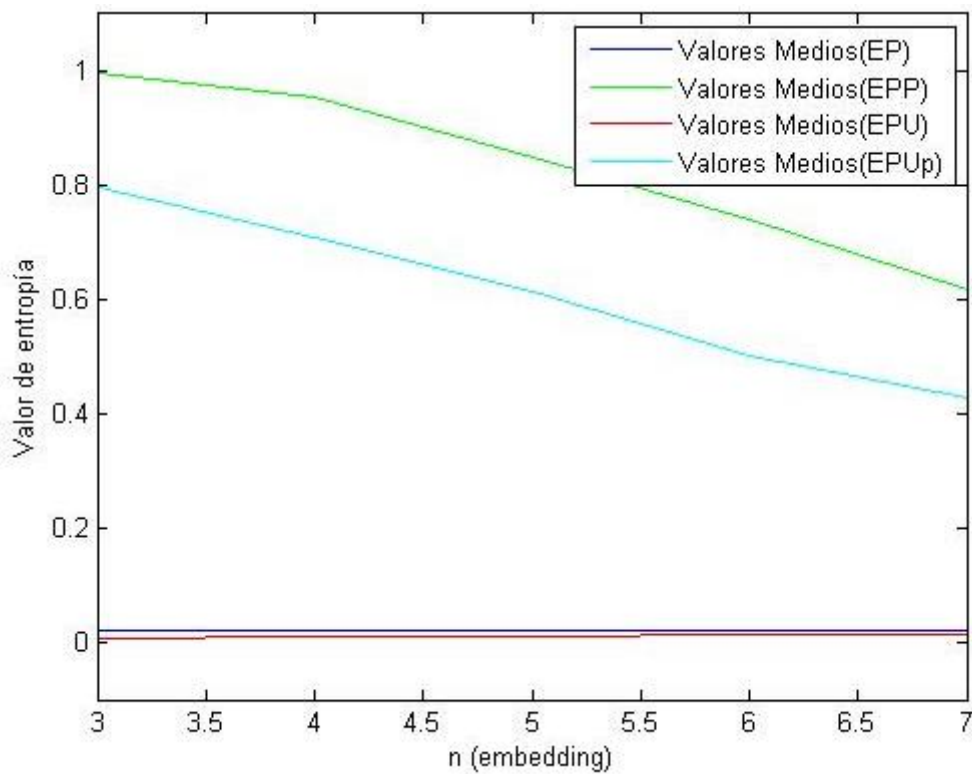


Ilustración 45: Resultados de EP para matriz de Valores Medios de la imagen rugosa, para valores de embedding entre 3 y 7

Observaciones: los resultados de este método no pueden ser adoptados para describir el tipo de rugosidad en texturas. El promedio de las regiones de la imagen, conlleva una baja en ellos

niveles de entropía que en este método se ven muy pronunciados. Los valores de entropía para la imagen lisa son más bajos que para la imagen rugosa como es de esperar, sin embargo, se evidencia la diferencia entre aquellos métodos que utilizan EPP y los que no. En este sentido, se ve que el promediado de la imagen rugosa genera valores de EP y EPU muy bajos debido a haber achicado la diferencia entre elementos cercanos en la matriz de valores medios. Este acercamiento de los valores cercanos no es visible con los descriptores EPP y EPU, ya que el método ponderado analiza la cercanía a la distancia promedio de los valores medios obtenidos para cada sector de la imagen, mientras que EP y EPU tiene en cuenta la intensidad de los valores medios cercanos.

5.1.2.3 EP de matriz de entropías

En la Tabla 14 se muestran los resultados del cálculo de EP y EPP sobre matrices de entropía de permutación para la imagen lisa y rugosa. Los resultados también se representan en gráficos de EP vs n en la Ilustración 46 para la imagen lisa y en la Ilustración 47 para la superficie rugosa.

Propuesta 2. Entropía de permutación para Matrices de EP			
Imagen	n	EP	EPP
<i>Lisa</i>	3	0,0000	0,0000
	4	0,0000	0,0000
	5	0,0000	0,0000
	6	0,0000	0,0000
	7	0,0000	0,0000
Imagen	n	EP	EPP
<i>Rugosa</i>	3	0,9708	0,9915
	4	0,9930	0,9886
	5	0,9907	0,9727
	6	0,9860	0,9571
	7	0,9792	0,9418
Algoritmos elegidos		✓	✓

Tabla 14: Resultado de los algoritmos EP y EPP para MatrizEP para las imágenes lisa y rugosa

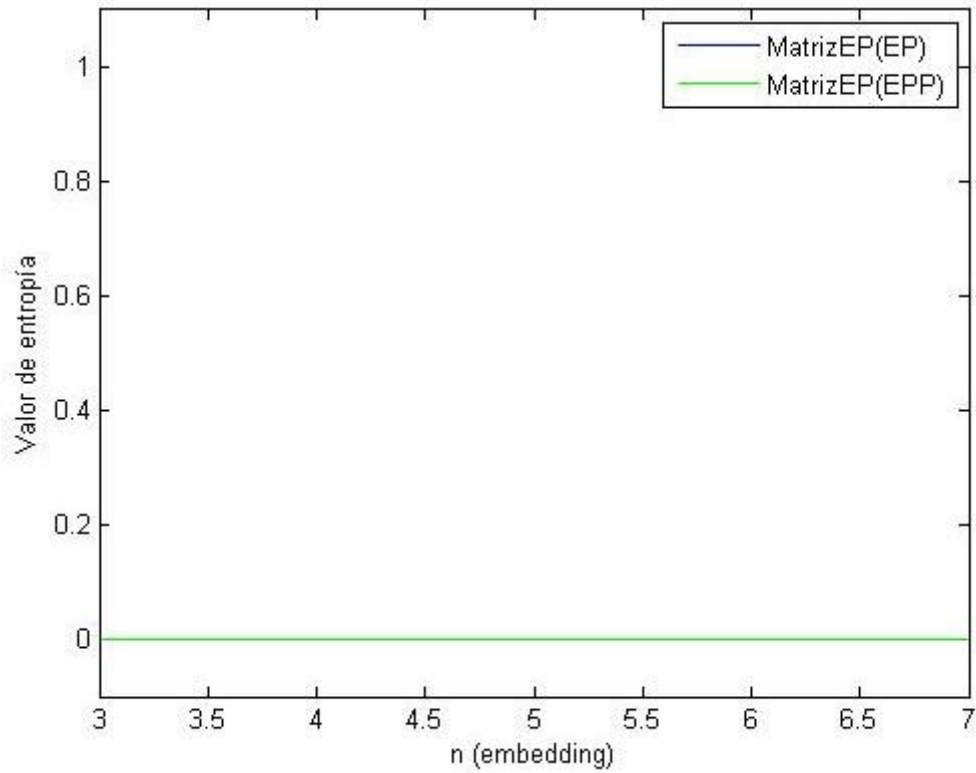


Ilustración 46: Resultados de los algoritmos aplicados sobre la MatrizEP de la imagen lisa, para valores de embedding de 3 a 7.

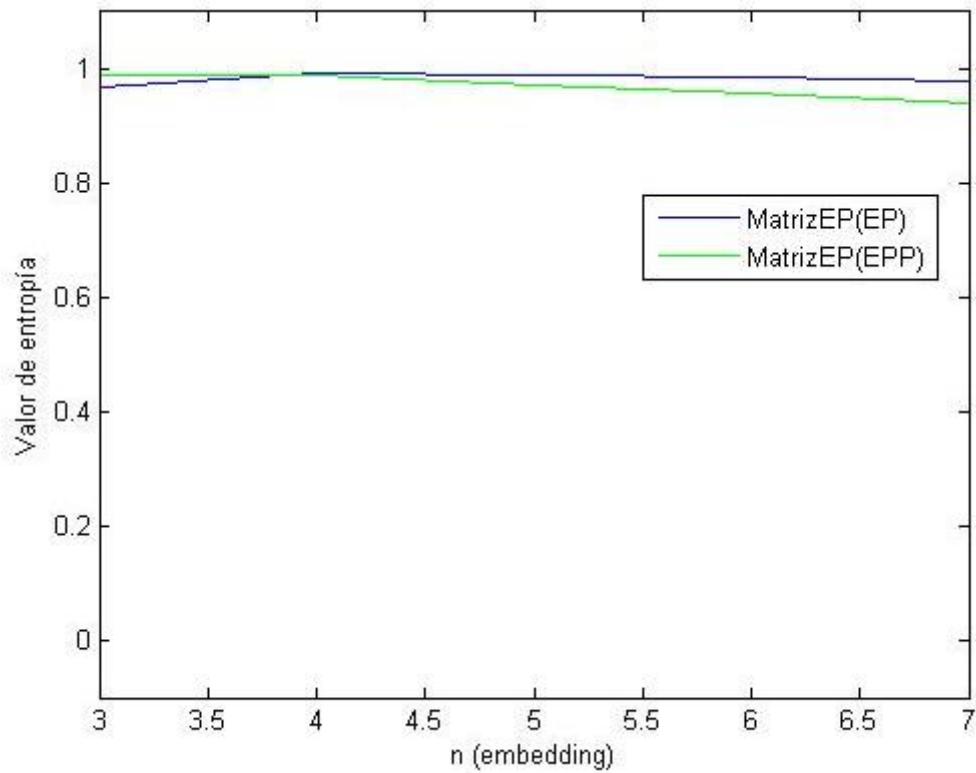


Ilustración 47: Resultados de los algoritmos aplicados sobre la MatrizEP de la imagen rugosa, para valores de embedding de 3 a 7.

En la Tabla 15, se muestran los resultados del cálculo de EP y EPP sobre matrices de entropía de permutación ponderada de las imágenes lisa y rugosa. En la Ilustración 48 y Ilustración 49 se muestran gráficamente los resultados de EP y EPP vs n .

Propuesta 2. Entropía de permutación para Matrices de EPP			
Imagen	n	EP	EPP
<i>Lisa</i>	3	0,0000	0,0000
	4	0,0000	0,0000
	5	0,0000	0,0000
	6	0,0000	0,0000
	7	0,0000	0,0000
Imagen	n	EP	EPP
<i>Rugosa</i>	3	0,9998	0,9995
	4	0,9976	0,9915
	5	0,9932	0,9763
	6	0,9872	0,9610
	7	0,9803	0,9443
Algoritmos elegidos		✓	✓

Tabla 15 : Resultado de los algoritmos EP y EPP para MatrizEPP para las imágenes lisa y rugosa

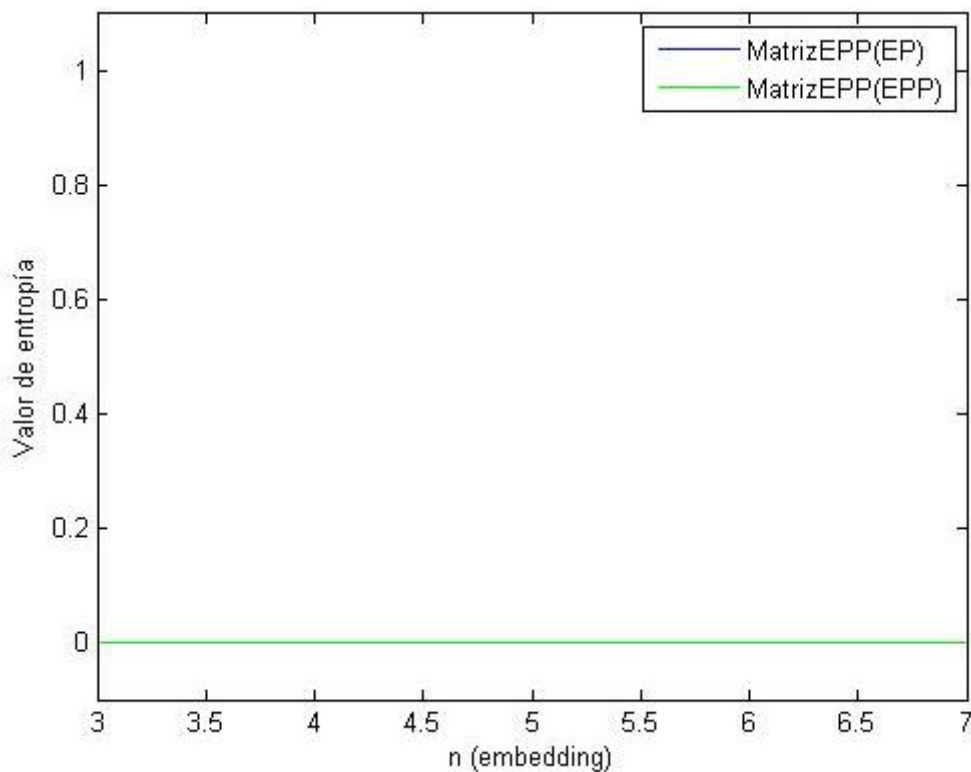


Ilustración 48: Resultados de los algoritmos aplicados sobre la MatrizEPP de la imagen lisa, para valores de embedding de 3 a 7.

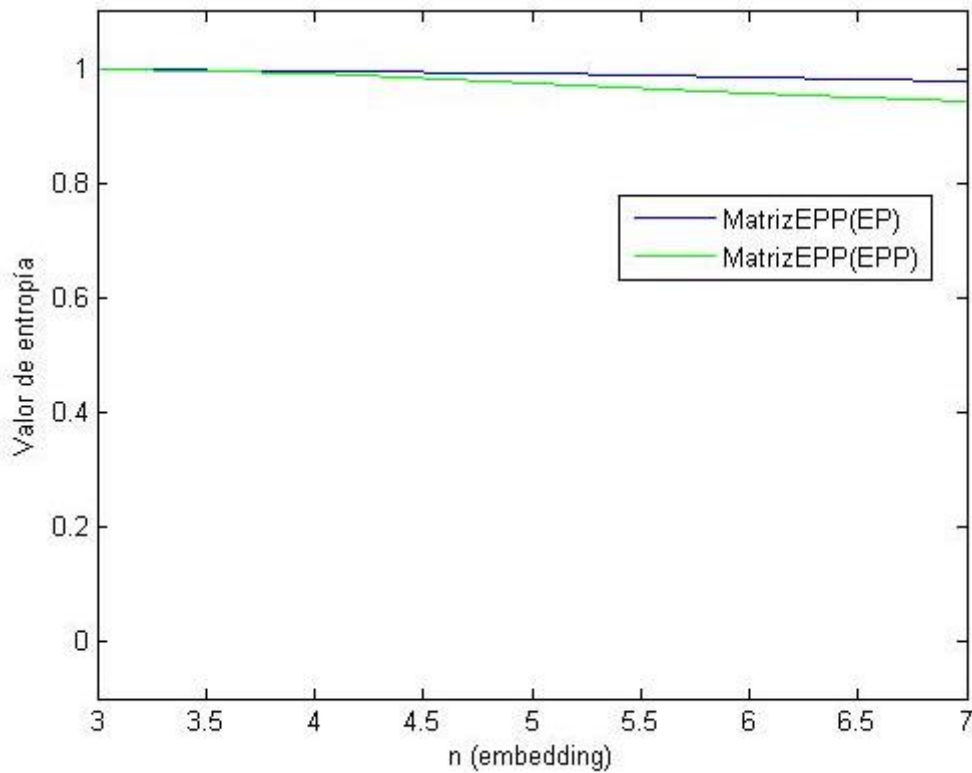


Ilustración 49: Resultados de los algoritmos aplicados sobre la MatrizEPP de la imagen rugosa, para valores de embedding de 3 a 7.

Se muestran los resultados de los algoritmos EP y EPP obtenidos, al aplicarlos sobre matrices de entropía de permutación umbral de las imágenes lisa y rugosa. Los mismos resultados se muestran en la Tabla 16 y de manera gráfica en la Ilustración 50 para la imagen lisa y en Ilustración 51 para la imagen rugosa.

Propuesta 2. Entropía de permutación para Matrices de EPU			
Imagen	<i>n</i>	EP	EPP
<i>Lisa</i>	3	0,0000	0,0000
	4	0,0000	0,0000
	5	0,0000	0,0000
	6	0,0000	0,0000
	7	0,0000	0,0000
Imagen	<i>n</i>	EP	EPP
<i>Rugosa</i>	3	0,9833	0,9873
	4	0,9855	0,9634
	5	0,9733	0,9333
	6	0,9595	0,9083
	7	0,9468	0,8891
Algoritmos elegidos		✓	✓

Tabla 16: Resultado de los algoritmos EP y EPP para MatrizEPU para las imágenes lisa y rugosa

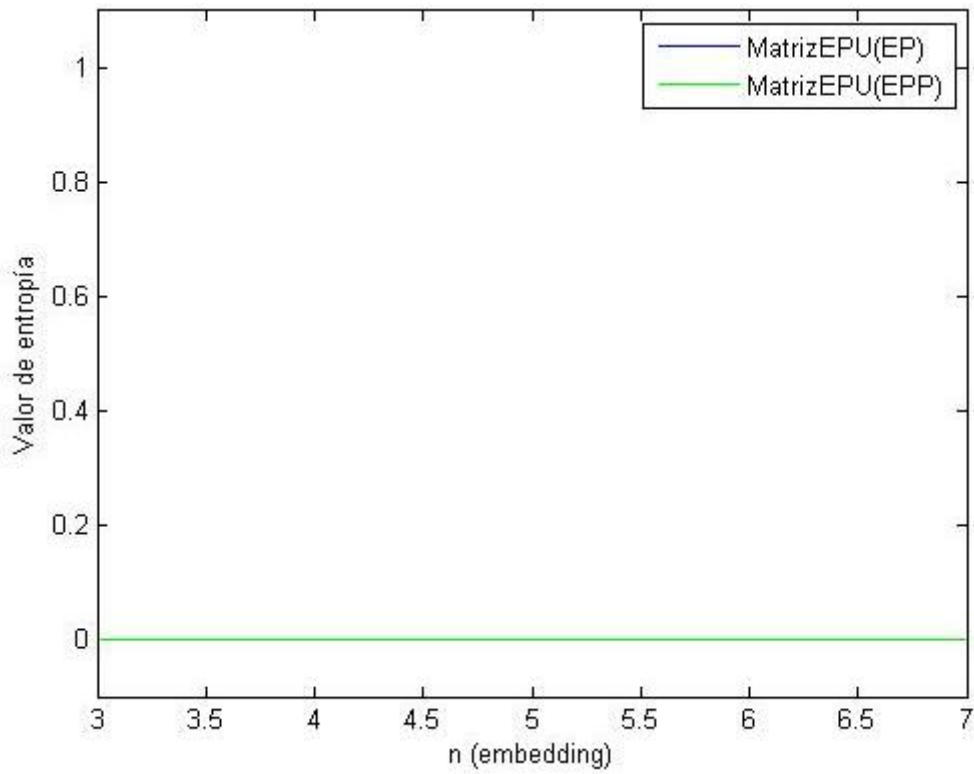


Ilustración 50: Resultados de los algoritmos aplicados sobre la MatrizEPU de la imagen lisa, para valores de embedding de 3 a 7.

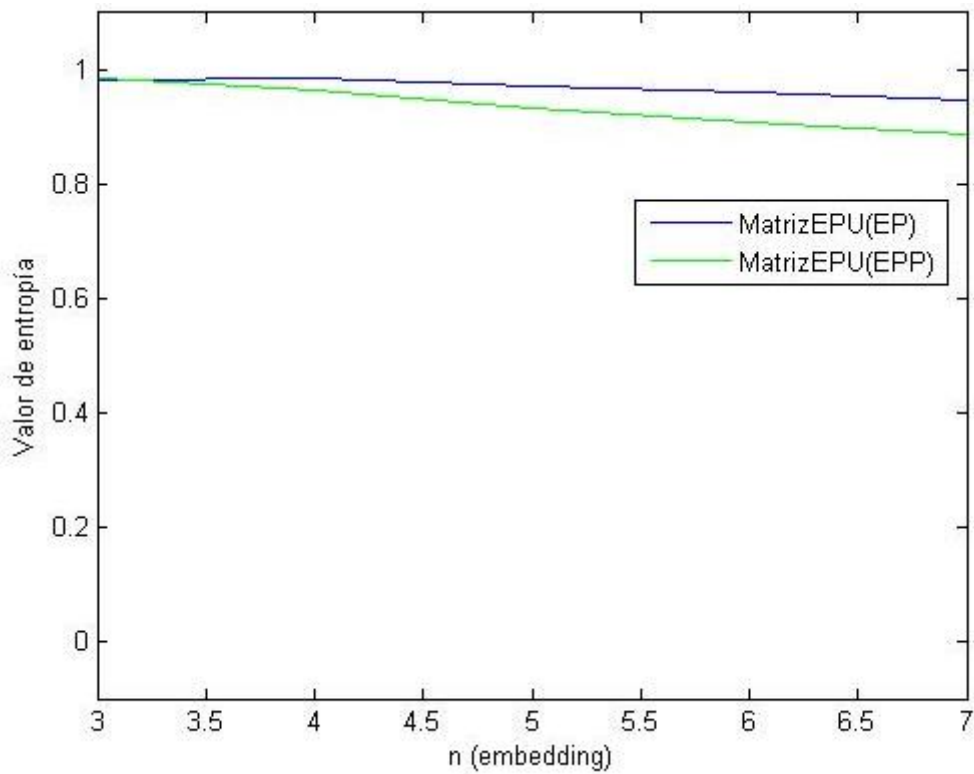


Ilustración 51: Resultados de los algoritmos aplicados sobre la MatrizEPU de la imagen rugosa, para valores de embedding de 3 a 7.

Observaciones: los algoritmos presentan resultados esperados, que logran diferenciar a la imagen totalmente lisa de otra altamente rugosa. Además, los valores de EP y EPP obtenidos fueron muy similares entre sí para todos los valores de *embedding*. Los valores de entropía para la imagen rugosa se diferencian más a medida que aumenta el valor de *n*, debido a que EPP disminuye mediante aumenta el valor de *embedding*.

5.1.2.4 Entropía de permutación con vecindad de Quincunx

El método de obtención de descriptores de EP, EPP y EPU recorriendo la imagen según el criterio de Quincunx, presentó los resultados mostrados en la Tabla 17. Los mismos resultados son graficados en la Ilustración 52 para la imagen lisa y en la Ilustración 53 para la imagen rugosa.

Propuesta 2. Entropía de permutación con vecindad de Quincunx				
Imagen	<i>n</i>	EP	EPP	EPU
<i>Lisa</i>	3	0,0000	0,0000	0,0000
	4	0,0000	0,0000	0,0000
	5	0,0000	0,0000	0,0000
	6	0,0000	0,0000	0,0000
	7	0,0000	0,0000	0,0000
Imagen	<i>n</i>	EP	EPP	EPU
<i>Rugosa</i>	3	0,6890	0,6850	0,0100
	4	0,3930	0,3977	0,0099
	5	0,2487	0,2533	0,0100
	6	0,1748	0,1782	0,0099
	7	0,1311	0,1340	0,0099
Algoritmos elegidos		X	X	X

Tabla 17 : Resultado de los algoritmos EP, EPP y EPU con vecindad de Quincunx para las imágenes lisa y rugosa

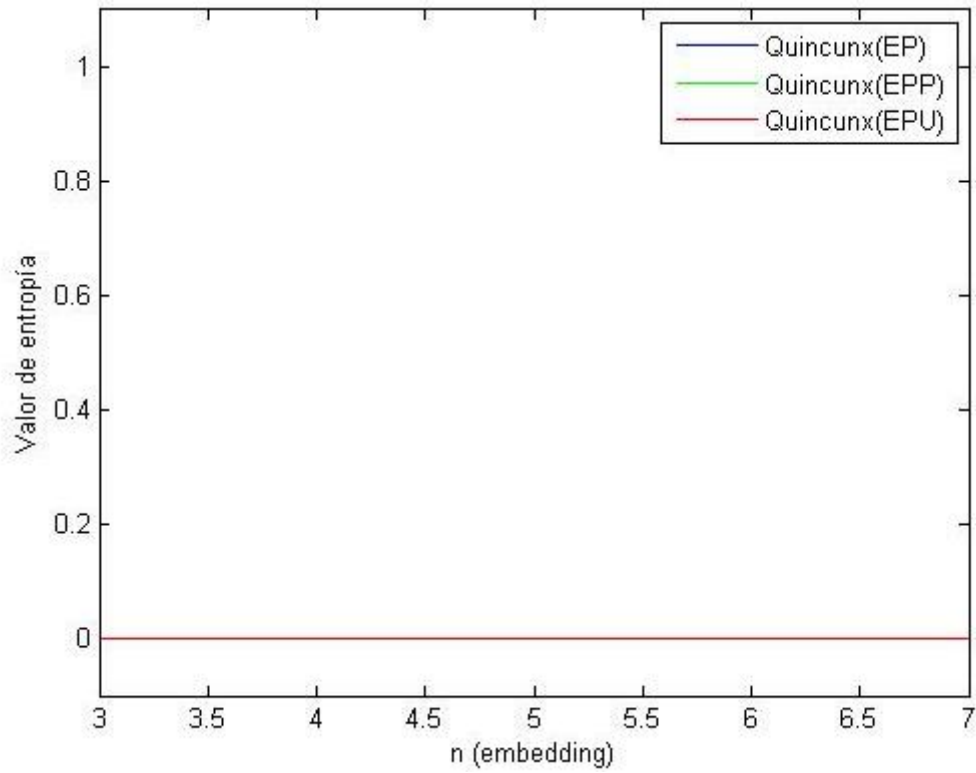


Ilustración 52: Resultados de EP aplicando el criterio de la vecindad de Quincunx para la imagen lisa, con embedding de 3 a 7

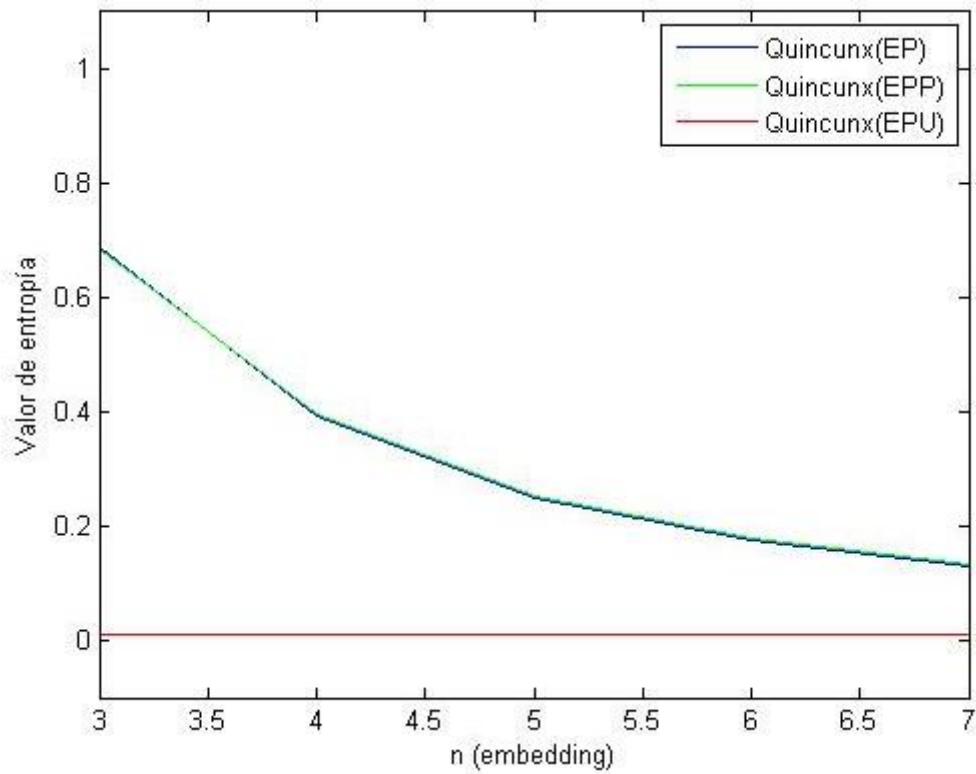


Ilustración 53: Resultados de EP aplicando el criterio de la vecindad de Quincunx para la imagen rugosa, con embedding de 3 a 7

Observaciones: la imagen lisa es caracterizada por los tres algoritmos correctamente. Sin embargo todos los valores obtenidos para la imagen rugosa están muy por debajo del valor de entropía esperado, lo que indica el método promedia mucho los datos, por lo que la imagen rugosa no puede ser caracterizada.

Los métodos seleccionados serán aquellos cuyos valores de entropía resultaron coherentes con los esperados para cada una de las imágenes sintéticas lisa y rugosa, entre 0,8 y 1 para la imagen rugosa y entre 0 y 0,2 para la imagen lisa. Se muestran las características principales de los algoritmos seleccionados en la Tabla 18.

ALGORITMOS SELECCIONADOS	Abreviatura	Propuesta	Mejor valor de n
<i>Entropía de Permutación</i>	EP	1	3-7
<i>EP Ponderada</i>	EPP	1	3-7
<i>EP con umbral de comparación</i>	EPU EPU _p	1	6-7
<i>EP con umbralado de la imagen</i>	EPT EPT _p	1	6-7
<i>EP con imágenes filtradas (con aplicación del algoritmo EP)</i>	EP _m (EP) EP _{ma} (EP) EP _g (EP)	2	3-4
<i>EP de matriz de entropías</i>	MEP _{ep} MEP _p MEP _{Pep} MEP _{PPp} MEP _{Uep} MEP _{Up}	2	4-5 3-4 3-4 3-4 3-4 3-4

Tabla 18: Características de los algoritmos seleccionados según la performance que presentaron para las imágenes lisa y rugosa.

Los 15 algoritmos seleccionados, mostraron mejores resultados para determinados valores de *embedding*. La condición para determinar si un algoritmo brindó mejores resultados para un valor de *n* que para otro, es la cercanía que muestran los resultados a los valores de entropía esperados para la imagen lisa y para la imagen rugosa. De aquí en más se denominarán

descriptores a los obtenidos por los algoritmos respectivos para un determinado tamaño del parámetro *embedding*, su nomenclatura se mostrará como: *NombreDelAlgoritmo(n)*, donde “*NombreDelAlgoritmo*” es la abreviatura del algoritmo y “*n*” el valor de *embedding* usado para como parámetro para el cálculo del descriptor.

Los caso de EP y EPP mantienen constante el valor de la entropía a mediada que *n* aumenta. Los otros métodos desarrollados para la propuesta 1 (EPU, EPU_p, EPT y EPT_p) muestran mejores resultados a medida que el *embedding* aumenta. Sin embargo, los métodos desarrollados en la propuesta 2 presentaron mejores resultados para los valores de *embedding* más bajos.

Durante el ensayo de todos los algoritmos, se contempló el comportamiento de aquellos métodos que en sus cálculos utilizaran EPP. En estos se observó que se producen grandes variaciones frente a pequeños cambios de la imagen. Este desempeño se puede observar especialmente en el filtrado de la imagen lisa en la Tabla 10, Tabla 11, Tabla 12 y Tabla 13, donde mientras todos los otros métodos mantuvieron bajos sus valores de entropía para la imagen lisa filtrada, aquellos que utilizaron el algoritmo EPP aumentaron significativamente en valor de entropía, pasando de 0 a 0,5 aproximadamente, valores que no representan a una señal cuyos datos son muy cercanos entre sí.

Este comportamiento se debe a que el algoritmo de entropía de permutación ponderada, construye el histograma de permutación afectando a los bins por un peso determinado por suma de las distancias de cada uno de los elementos al valor medio de los mismos. De esta manera, no solo se tiene en cuenta el orden que presentan los datos para determinar el orden del vector, sino que además se considera este peso dependiente de las distancias relativas entre las amplitudes de los datos. Así, si un determinado patrón de orden se repite con menos frecuencia en la imagen pero la diferencia entre los píxeles es grande, será ponderado con un peso mayor. Esto genera que lo resultados no sean los esperados para la imagen lisa.

Sin embargo, EPP presenta resultados acordes a los esperados para la imagen lisa, al aplicar los algoritmos de entropía de permutación para matrices de entropías. Se atribuye dicho desempeño a que EPP se aplica sobre matrices de entropías, las cuales tienen valores entre 0 y 1, a diferencia de los otros casos donde EPP era aplicado directamente sobre la matriz de píxeles, cuyos valores varían entre 0 y 255 produciendo en esta última pesos mayores, debido distancias relativas entre los datos más grandes.

5.2 Imágenes artificiales. Texturas Gruesas y Finas

Se analiza el comportamiento de los métodos frente a imágenes de superficies con diferentes tamaños de texturas, identificando las texturas gruesas como aquellas cuyo patrón de repetición abarca un gran porcentaje de la imagen con una baja frecuencia de repetición, mientras que las texturas finas serán aquellas cuyo patrón se repite con mayor frecuencia ocupando cada uno de ellos un porcentaje bajo del espacio en la imagen total. En sentido amplio, la textura puede ser definida por su grosor para una misma imagen en distintas escalas, si dos patrones difieren sólo en escala, el patrón que se ha magnificado es más grueso y con una repetición más baja de sus elementos más gruesa será su textura visual.

En base al análisis de los resultados obtenidos de los ensayos de las propuestas 1 y 2, aplicadas a las imágenes sintéticas lisa y rugosa, se seleccionan aquellos algoritmos cuya performance fue

considerada satisfactoria para discriminar dichas superficies. Los algoritmos seleccionados fueron ensayados sobre las imágenes de texturas gruesas y finas presentadas en la Ilustración 32 y Ilustración 33 respectivamente. Se busca comprender el comportamiento de los descriptores de entropía de permutación para la identificación de diferentes tamaños de texturas. Como se dijo anteriormente, una textura gruesa puede ser la versión escalada de una textura más fina. Las imágenes sintéticas creadas fueron mediante el mismo patrón de forma, repetido con mayor o menor frecuencia y en distintos tamaños, por lo que las texturas pueden representar sus versiones escaladas.

Las imágenes están ordenadas de mayor a menor grosor de la siguiente manera: *Gruesa1*, *Gruesa2*, *Fina1*, *Fina2*. De manera que la textura de la imagen *Gruesa1* representa el mayor grosor y la imagen *Fina2* la del menor.

Los resultados se muestran en gráficos de entropía en función del *embedding*, un gráfico por cada uno de los algoritmos seleccionados. Para facilitar el análisis, se mostrarán todos los gráficos indicando al algoritmo que pertenecen, y luego se discuten los resultados mostrados.

5.2.1 Ensayos con algoritmos de propuesta 1

Las pruebas realizadas con los métodos que procesan píxeles de la imagen mediante vectores unidimensionales, son mostradas en las siguientes figuras, donde las imágenes de la izquierda muestra los métodos EP, EPU y EPT y los de la derecha sus versiones respectivas ponderadas.

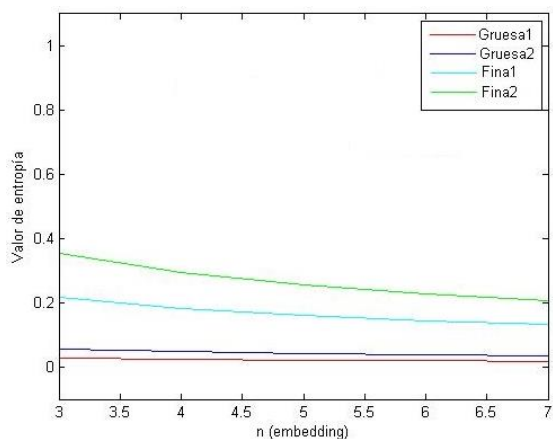


Ilustración 54: Algoritmos EP para imágenes con textura Gruesa y Fina para n de 3 a 7.

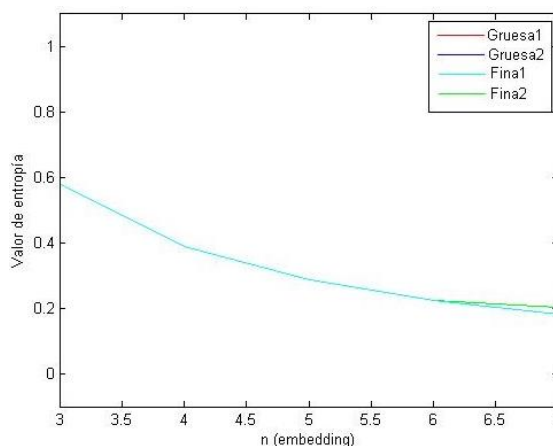


Ilustración 55: Algoritmos EPP para imágenes con textura Gruesa y Fina para n de 3 a 7.

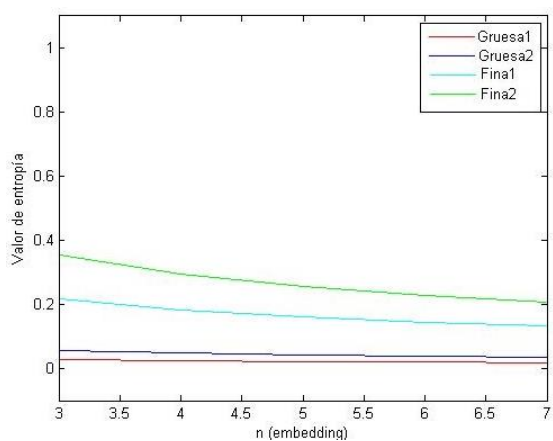


Ilustración 56: Algoritmos EPU para imágenes con textura Gruesa y Fina para n de 3 a 7.

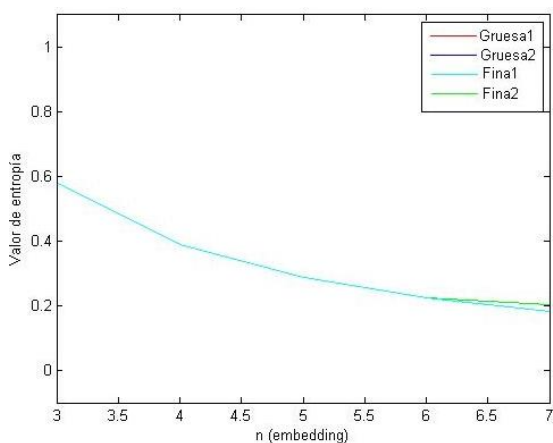


Ilustración 57: Algoritmos EPU_p para imágenes con textura Gruesa y Fina para n de 3 a 7.

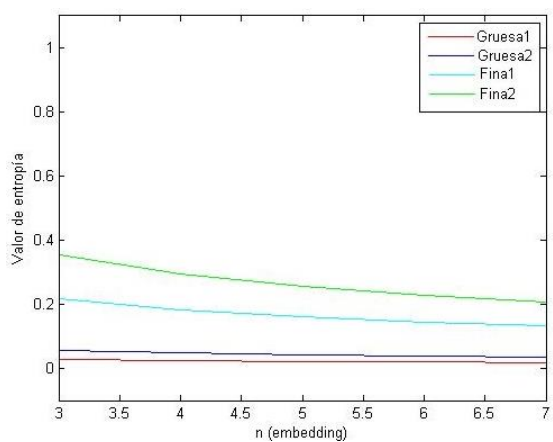


Ilustración 58: Algoritmos EPT para imágenes con textura Gruesa y Fina para n de 3 a 7.

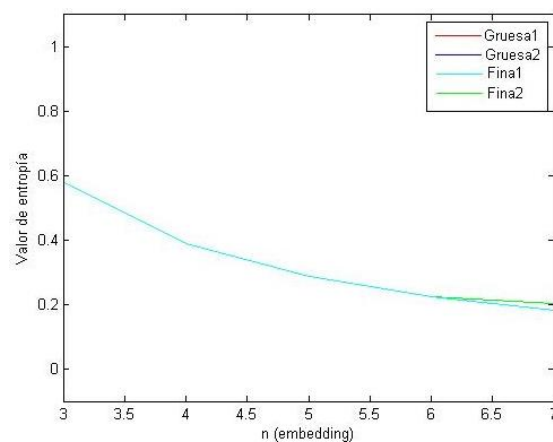


Ilustración 59: Algoritmos EPT_p para imágenes con textura Gruesa y Fina para n de 3 a 7.

Mediante las ilustraciones, se puede concluir que los algoritmos que utilizan el método de ponderación, no presentan cualidades para distinguir entre diferentes tamaños de texturas. El

motivo es el mismo que el indicado anteriormente para el caso de imágenes filtradas, ya que en resumen, este método pondera las repeticiones de un patrón de orden según las distancias relativas entre los píxeles que conforman el vector de *embedding* analizado, que por ser la misma forma textural repite las distancias relativas entre los tonos de gris y la media entre los elementos del vector. Esta es una cualidad importante a analizar con más profundidad, ya que los algoritmos en su versión ponderada, pueden representar a un patrón de repetición textural a diferentes escalas bajo el mismo valor de entropía. Esta hipótesis heurística se comprobará luego con las mismas pruebas pero ante imágenes de superficies texturales reales en sus versiones escaladas.

Las ilustraciones encolumnadas a la izquierda, muestran diferentes niveles de entropía respecto al grosor de la textura. Al no existir una gran cantidad de patrones de órdenes los valores de entropía resultan bajos, entre 0 y 0,4, pero diferenciados para todo valor de n . Las texturas más gruesas presentaron menor entropía que las más finas y este orden se mantuvo para todo valor de n . Por lo tanto, mediante la utilización de estos métodos, se pueden caracterizar texturas más gruesas de aquellas más finas para la misma parametrización de los algoritmos. Al igual que las entropías ponderadas, es necesario un análisis más profundo para la generalización de esta hipótesis, por lo que se analizará luego para imágenes de superficies reales con diferentes niveles de grosor textural.

5.2.2 Ensayos con algoritmos de propuesta 2

En las ilustraciones siguientes, se muestran los resultados obtenidos mediante la aplicación de los métodos cuyo procesamiento de la señal bidimensional guarda información del espacio de los datos. Primero se analizan los casos de EP para imágenes afectadas mediante los filtros de Media, Gauss y Mediana aritmética predeterminados en Matlab. Luego, se muestran resultados obtenidos mediante la aplicación de los algoritmos EP y EPP de la propuesta 1, en las matrices de entropías (*matrizEP*, *matrizEP* y *matrizEPU*).

5.2.2.1 EP de imágenes filtradas para Texturas gruesas y finas

Los resultados de EP aplicados a imágenes filtradas se muestran en las ilustraciones que siguen al texto. Al igual que EP en aplicado a las mismas imágenes pero sin filtrar (ver Ilustración 54), los algoritmos categorizan entre las texturas más gruesas a más finas con resultados de entropía de menor a mayor valor respectivamente. En el caso de EP con imágenes filtradas por filtro de Media no se observan variaciones notables en comparación a no filtrar la imagen. Sin embargo en la aplicación de los filtros de Gauss y Media aritmética, se observa un aumento significativo en el rango dinámico, donde la entropía se sitúa en valores entre 0 y 0,6. De esta manera se pueden utilizar estos algoritmos para distinguir con mayor facilidad entre una textura gruesa y una fina. Además, se ve en la Ilustración 61 y Ilustración 62 que la diferencia entre las dos texturas gruesas se distancian de las dos finas por una mayor cantidad de energía que en el caso de no aplicar los filtros.

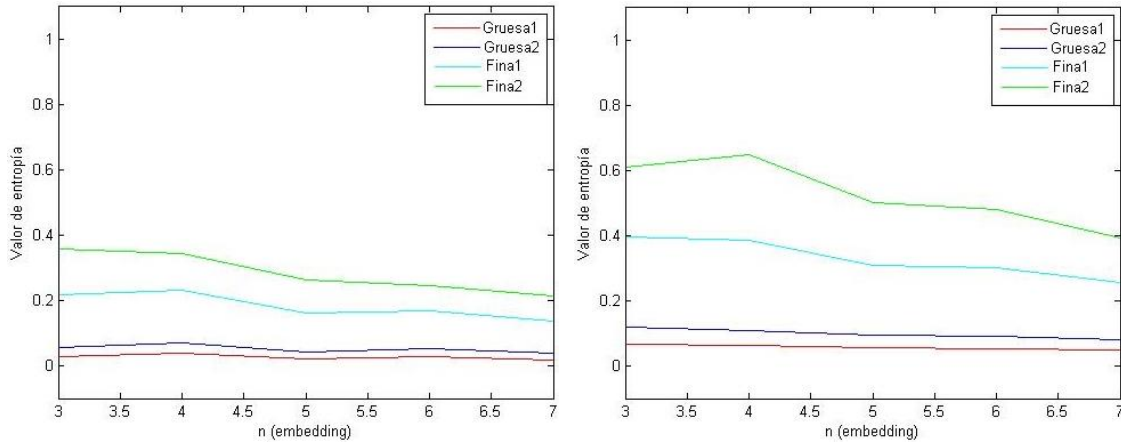


Ilustración 60: EP para imágenes de texturas gruesas y finas filtradas con filtro de Media

Ilustración 61: EP para imágenes de texturas gruesas y finas filtradas con filtro de Gauss

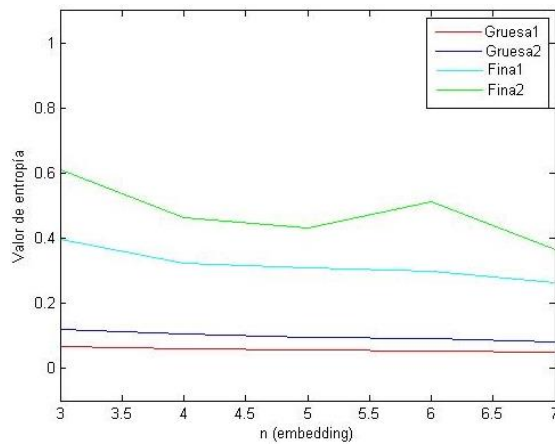


Ilustración 62: EP para imágenes de texturas gruesas y finas filtradas con filtro de Mediana Aritmética

5.2.2.2 EP de matrices de entropías para texturas gruesas y finas

Bajo el texto, se muestran los gráficos para los resultados de entropía en función del tamaño de embedding, para los algoritmos seleccionados en la *sección 5.1 Imágenes artificiales. Textura Lisa* y *Rugosa*. Encolumnados a la izquierda, se encuentran los resultados obtenidos para el ensayo de EP en matrices de entropía de permutación, de permutación ponderada y de permutación umbral, y a la derecha los resultados obtenidos mediante la utilización de EPP para las mismas matrices. Al igual que en el caso de aplicación de los algoritmos desarrollados en la propuesta 1, los métodos que utilizan EPP en sus cálculos, muestran menor capacidad para diferenciar los distintos grosores en la textura de las imágenes. En el caso de los algoritmos que calculan EP de las matrices de entropías, se observa que aumenta aún más el rango dinámico que para las imágenes pre-filtradas analizadas anteriormente. Sin embargo, en este último caso, a medida que el valor de *embedding* aumenta, la textura *Fina2* baja su entropía abruptamente, y se espera que para valores de *embedding* aún mayores, la textura *Fina1* se comporte de igual forma.

Si se realiza una comparación entre los resultados mostrados en los gráficos de la columna izquierda de estos métodos, aquellos de la misma columna de la propuesta 1 y todos los gráficos con las imágenes filtradas, se puede notar que las texturas *Gruesa1* y *Gruesa2* tienen el mismo comportamiento en todos los casos: valor de entropía bajo y constante para todo n . Lo mismo sucede si se comparan los resultados de entropía para estas texturas expuestos en las ilustraciones encolumnadas a la derecha de método pertinente con las de la columna derecha de la propuesta 1, donde se utiliza el método de EPP. Aquí las texturas gruesas presentan un valor de entropía decreciente con el tamaño de *embedding*, tomando valores entre 0,6 y 0,2.

Con lo observado anteriormente, se puede concluir que las imágenes gruesas no presentan grandes alteraciones de EP o EPP, al variar los datos mediante filtros o transformaciones. Esta característica de las texturas gruesas es atribuida que sus patrones de texturas no tienen un tamaño comparable con el valor de *embedding*, por lo tanto las variaciones en los datos procesados no son percibidos en el área del espacio que se analizan los datos, ya que aunque el valor de *embedding* sea el mayor posible, en este caso matrices de 7×7 , las texturas fueron construidas con matrices de 100×100 y de 50×50 que presentan zonas cuadradas de igual tono de gris en la imagen. Por lo tanto las variaciones de los píxeles no serán apreciadas por ventanas de 7×7 . Contrario es el caso de las texturas finas, las que fueron construidas con tamaños de texturas comparables con los valores de *embedding* adoptados. Si se realizan las mismas comparaciones entre los diferentes métodos, pero ahora para las imágenes *Fina1* y *Fina2*, se comprueba lo dicho anteriormente, ya que los valores de entropía para estas texturas varían considerablemente dependiendo el método aplicado y el valor de *embedding*.

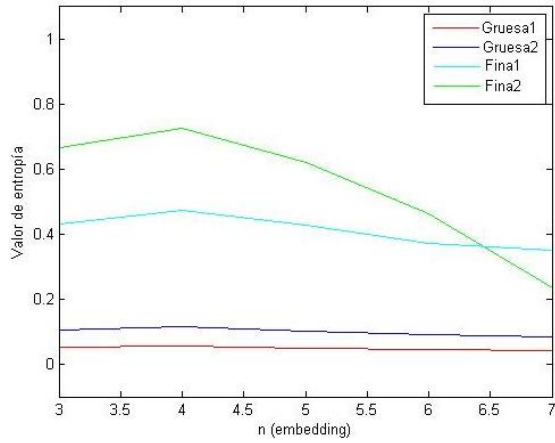


Ilustración 63: Entropía de permutación para matriz de EP

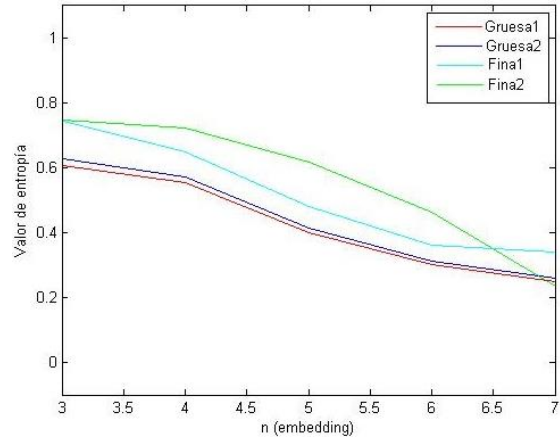


Ilustración 64: Entropía de permutación ponderada para matriz de EP

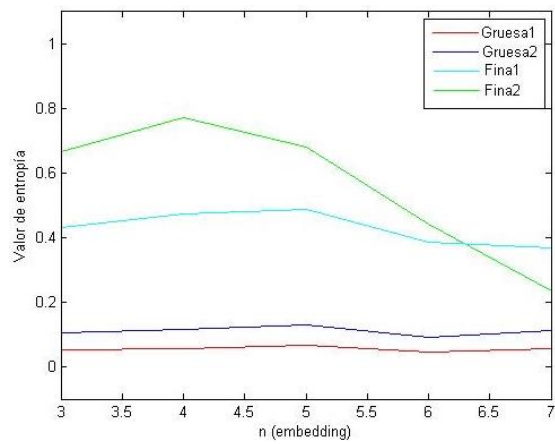


Ilustración 65: Entropía de permutación para matriz de EPP

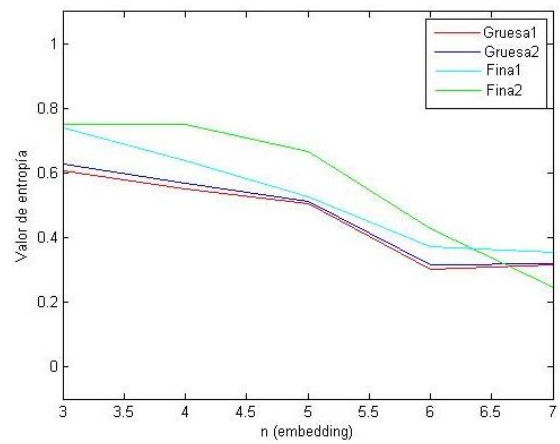


Ilustración 66: Entropía de permutación ponderada para matriz de EPP

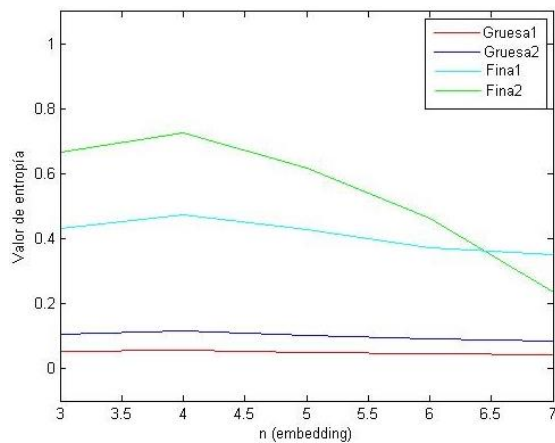


Ilustración 67: Entropía de permutación para matriz de EPU

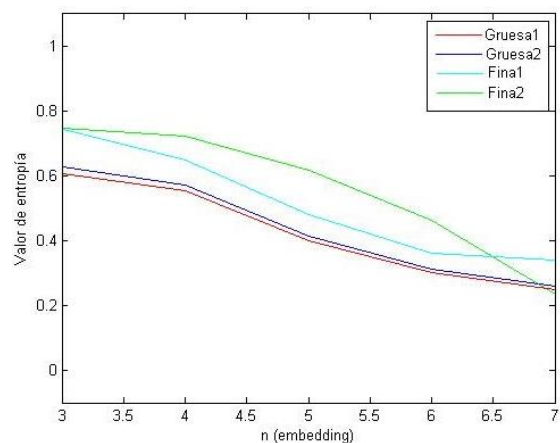


Ilustración 68: Entropía de permutación ponderada para matriz de EPU

5.3 Imágenes reales

Al igual que para imágenes de texturas gruesas y finas, se ensayan los 15 algoritmos seleccionados en la sección 5.1 Imágenes artificiales. Textura Lisa y Rugosa de este trabajo, en las 8 superficies reales mostradas en la Ilustración 34.

Las superficies fueron ordenadas según su nivel de rugosidad. Esta categorización se realiza según la apreciación visual de las imágenes, por lo que pueden existir ambigüedades o diferencias de fiabilidad intra o inter observador. Este es un aspecto a tener en cuenta a la hora de analizar los resultados.

Las imágenes analizadas en esta sección el fin de revelar la capacidad de los métodos para ser aplicados a un caso práctico real. Se espera que los algoritmos puedan diferenciar las superficies más rugosas de las más finas, y que se presenten valores de entropías cercanos entre aquellas imágenes de superficies que poseen aspectos similares en su característica táctil de rugosidad.

5.3.1 Ensayos con algoritmos de propuesta 1

Las pruebas realizadas con los métodos de la propuesta 1, son mostradas en ilustraciones dispuestas luego de este texto. Las imágenes de la izquierda muestran los métodos EP, EPU y EPT y los de la derecha sus versiones respectivas ponderadas.

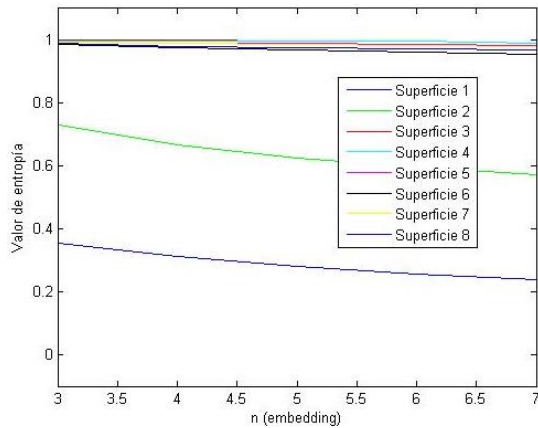


Ilustración 69: Resultados algoritmo EP para las 8 imágenes de superficies reales.

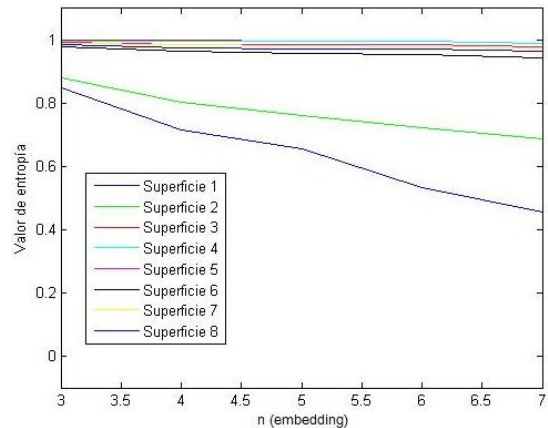


Ilustración 70: Resultados algoritmo EPP para las 8 imágenes de superficies reales.

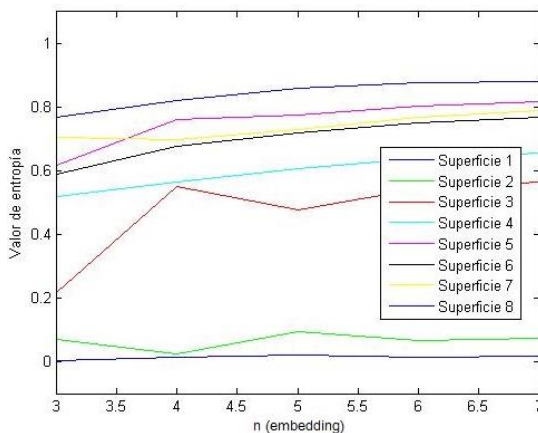


Ilustración 71: Resultados algoritmo EPT para las 8 imágenes de superficies reales.

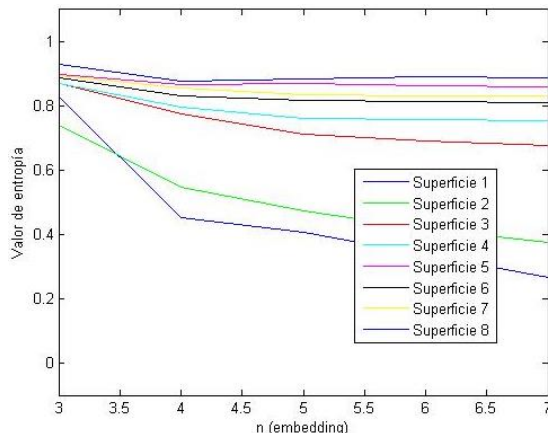


Ilustración 72: Resultados algoritmo EPTp para las 8 imágenes de superficies reales.

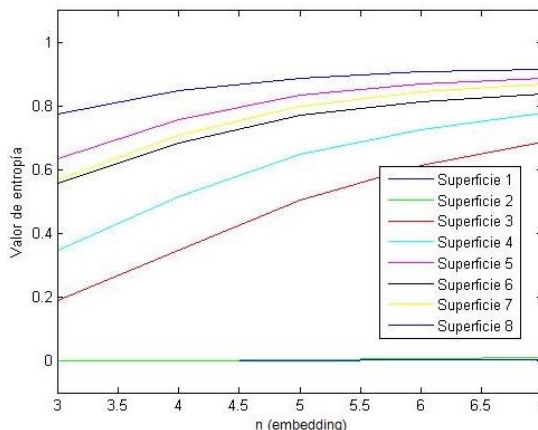


Ilustración 73: Resultados algoritmo EPU para las 8 imágenes de superficies reales.

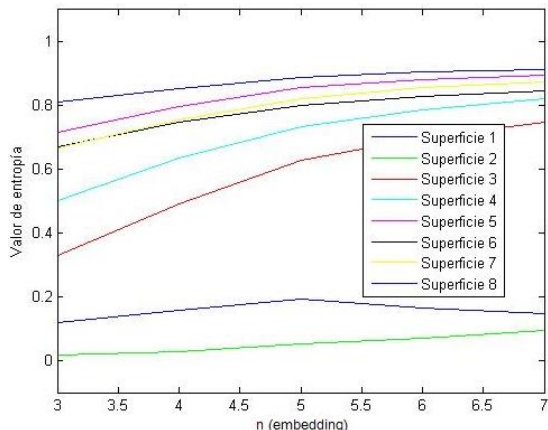


Ilustración 74: Resultados algoritmo EUp para las 8 imágenes de superficies reales.

Al igual que en los casos anteriores, se observa que los algoritmos que utilizan EPP para sus cálculos, muestran valores de entropía más altos que su misma versión pero con la utilización de EP. Para notarlo, basta con comparar las ilustraciones de la izquierda con las de la derecha, y observar que las de la izquierda presentan valores más bajos de entropía para todas las superficies analizadas. Dicho comportamiento se repite en todos los casos por ahora estudiados, resultando que EPP entrega mayores valores de entropía para todos los tipos de superficies,

incluso para imágenes de texturas lisas con pequeñas variaciones relativas entre los valores de sus píxeles.

En la Ilustración 69, Ilustración 70, Ilustración 71, Ilustración 72 que corresponden a los algoritmos EP, EPP, EPT y EPTp respectivamente se nota que las entropías amplían su rango dinámico a medida que aumenta el valor de *embedding*. Esta ventaja para distinguir a de manera unívoca a las distintas rugosidades, se puede extender a tamaños de *embedding* mayores, lo que se tendrá en cuenta para el análisis de otros casos prácticos. Lo contrario se observa para la Ilustración 73 y la Ilustración 74 correspondientes a los métodos EPU y EPU_p respectivamente, viendo que en estos casos, el mayor rango dinámico se obtiene para el valor de *embedding* más bajo ($n=3$) y a medida que n aumenta, el rango dinámico disminuye. Estas características en el comportamiento de los distintos algoritmos, pueden ser relevantes al momento de determinar el conjunto de descriptores óptimos para la caracterización de texturas en un caso práctico.

El orden de rugosidad resultante de la utilización de los algoritmos en comparación con el otorgado por inspección visual, guarda diferencias para las superficies más rugosas y mantiene el orden aplicado para las superficies menos rugosas. Esto último, sólo es diferente en el método EPU_p, el que considera que la *Superficie1* es más rugosa que la *Superficie2*, lo que es atribuido especialmente a los problemas de luminosidad presentes en la imagen de la *Superficie1*. A excepción de EPU_p el orden asignado por los algoritmos fue el mismo. Del nuevo acomodamiento resulta el siguiente orden de más lisas a más rugosas:

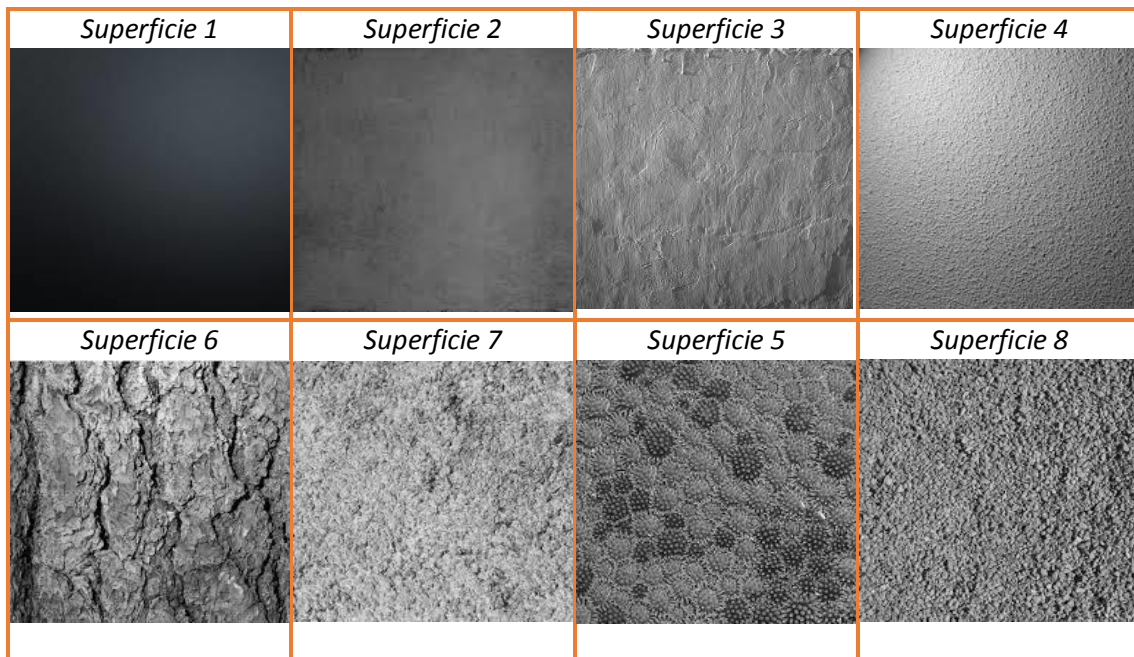


Ilustración 75: Nuevo orden según el nivel de rugosidad determinado por los resultados de la propuesta 1 para las 8 superficies.

Mediante el análisis de este nuevo orden se deducen algunos signos del comportamiento de los algoritmos. Las superficies más lisas, fueron categorizadas de la misma forma que la percepción visual humana. Las diferencias se encontraron en la categorización de las superficies más rugosas. En referencia a este último, se puede decir que la *Superficie5* presenta una mayor cantidad de bordes y saltos de tonos bajos a altos de gris (mayor frecuencia), que la *Superficie6*

y la *Superficie7*, con lo que se puede explicar que presente mayor entropía que las últimas dos. Además, *Superficie5* presenta una gran diferencia frente a las restantes, ya que es una textura rugosa dentro de segmentos, que por sí solos, representan una textura gruesa. Para que esta textura presente las mismas características de adquisición que el resto, habría que considerar la textura rugosa presente en solo uno de los segmentos.

5.3.2 Ensayos con algoritmos de propuesta 2

Se muestran y analizan los resultados obtenidos para los métodos que consideran los píxeles y su entorno cercano, con la intención de guardar información del espacio de los datos. Primero se analizan los casos de EP para imágenes afectadas mediante los filtros de Media, Gauss y Mediana aritmética predeterminados en Matlab. Luego, se muestran resultados obtenidos mediante la aplicación de los algoritmos EP y EPP de la propuesta 1, en las matrices de entropías (*matrizEP*, *matrizEP* y *matrizEPU*).

5.3.2.1 EP para imágenes de superficies reales filtradas

En las ilustraciones que siguen, se muestran los resultados obtenidos por la aplicación de algoritmo EP sobre las 8 superficies previamente filtradas mediante los filtros de Media, Media aritmética y Gauss en la Ilustración 76, Ilustración 77 y Ilustración 78 respectivamente.

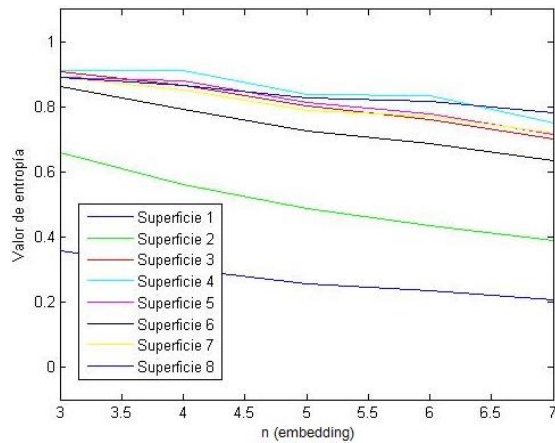


Ilustración 76: Resultados algoritmo EP para las 8 imágenes de superficies reales filtradas con filtro de media.

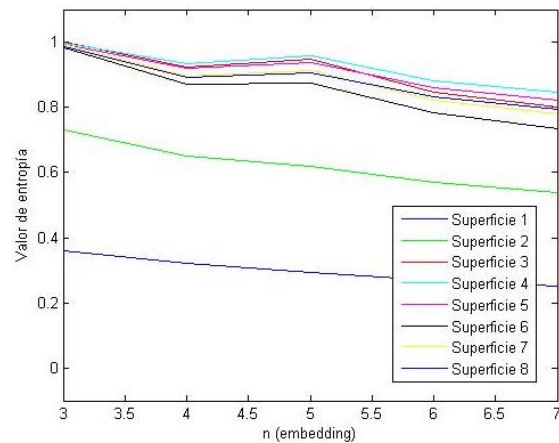


Ilustración 77: Resultados algoritmo EP para las 8 imágenes de superficies reales filtradas con filtro de gauss.

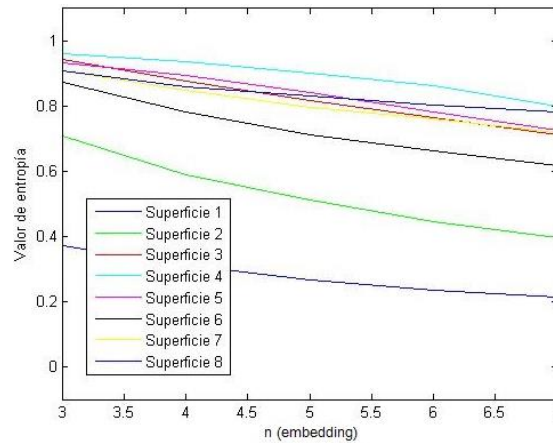


Ilustración 78: Resultados algoritmo EP para las 8 imágenes de superficies reales filtradas con filtro de mediana aritmética.

En comparación con EP en aplicado a las mismas imágenes pero sin filtrar (ver Ilustración 69), las superficies más rugosas presentaron valores de EP más bajos, y a la vez ampliaron su rango dinámico. Lo primero resulta lógico por el efecto intrínseco de suavizado de las imágenes al ser filtradas, los segundo se responde con lo mismo, ya que para imágenes de superficies más suaves (lisas) el algoritmo EP da valores de entropía con mayor dinámica que para el caso de las superficies más rugosas, las cuales presentan un rango muy comprimido en el eje de las ordenadas.

Los algoritmos categorizan entre las texturas más lisas, sin embargo para el resto de las superficies, el orden es muy distinto al previsto, aunque el rango dinámico de los valores de EP es bajo desde la *Superficie3* a la *Superficie8*, las que se distribuyen en un rango de 0,1 en sus valores de entropía, para el mayor valor de *embedding*, donde se presenta la mayor amplitud dinámica. Se concluye que el efecto de filtrado en imágenes de texturas reales puede alterar los resultados esperados, especialmente para los casos de texturas de rugosidad alta, y textura fina, ya que como se vio en la sección 5.2.2.1 EP de imágenes filtradas para Texturas gruesas y finas, la aplicación de filtros a las imágenes antes del cálculo de EP en texturas finas, produce notables alteraciones en los resultados en comparación con los casos sin filtrado previo al procesamiento

de los datos. Por lo que no resulta conveniente aplicar estos métodos para la caracterización de superficies finas.

5.3.2.2 EP de matrices de entropías para imágenes de superficies reales

En la columna izquierda se muestran los gráficos de EP y a la derecha de EPP para matrices de entropías de permutación, de permutación ponderada y de permutación umbral.

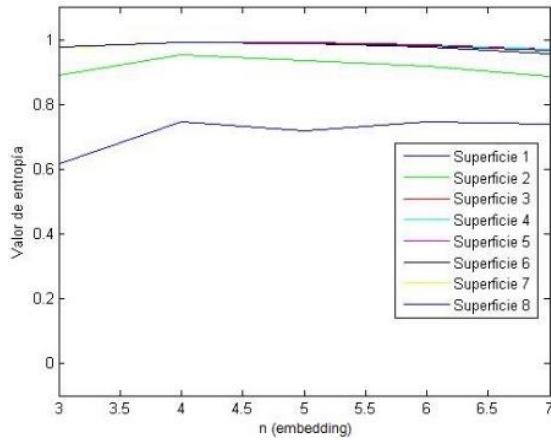


Ilustración 79: Entropía de permutación para matriz de EP

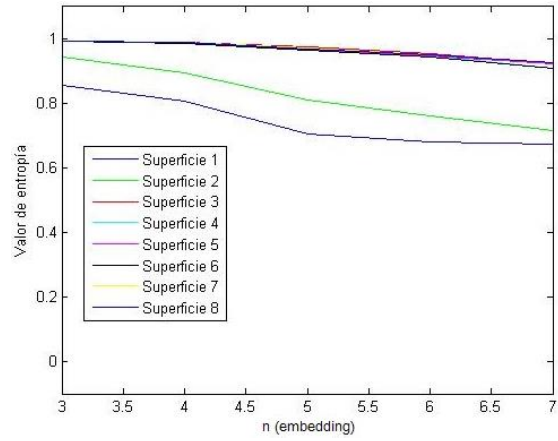


Ilustración 80: Entropía de permutación ponderada para matriz de EP

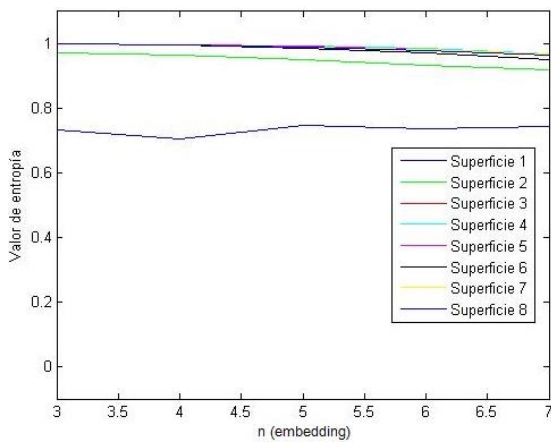


Ilustración 81: Entropía de permutación para matriz de EPP

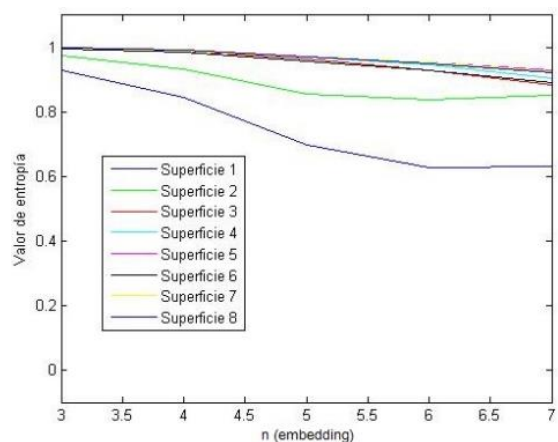


Ilustración 82: Entropía de permutación ponderada para matriz de EPP

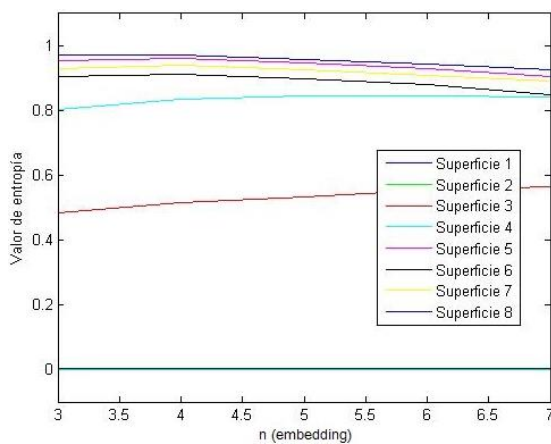


Ilustración 83: Entropía de permutación para matriz de EPU

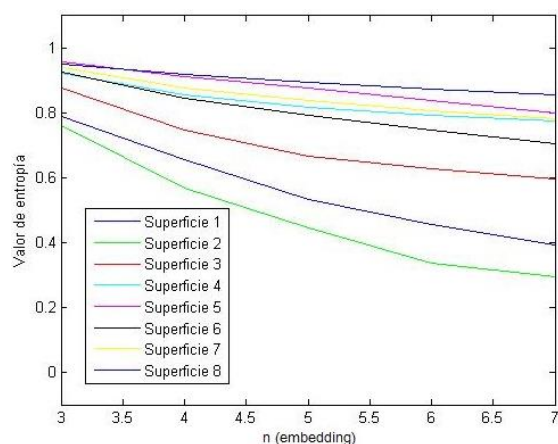


Ilustración 84: Entropía de permutación ponderada para matriz de EPU

Las cuatro primeras ilustraciones, denotan el comportamiento similar entre los algoritmos de esta propuesta y los de EP y EPP de la propuesta 1, donde las superficies más lisas (*Superficie 1* y *Superficie 2*) pudieron ser bien diferenciadas, mientras que las más rugosas presentan descriptores muy cercanos entre sí, dificultando la relación univoca entre el valor de entropía y la superficie a caracterizar.

El ensayo de EP y EPP en matrices de entropía de permutación umbral, muestra rangos dinámicos más amplios que para los casos anteriores, facilitando la identificación de cada superficie en referencia a su valor de entropía. Mientras que EP en matrices de EPU presenta mayor rango dinámico para descriptores con parámetros de *embedding* bajos, EPP en matrices de EPU muestra una tendencia a ampliar su rango dinámico para valores de *embedding* mayores.

A diferencia de los métodos que implementan filtros, de los resultados de estos métodos surge el mismo orden de las superficies según nivel de rugosidad que la obtenida en la propuesta 1 (ver Ilustración 75).

5.4 Mamografías (Caracterización de la densidad mamográfica)

Esta sección se divide en dos grandes partes, en la primera se muestran los resultados obtenidos mediante el ensayo de los algoritmos seleccionados. Mediante el análisis de los resultados se escogen descriptores que serán luego los utilizados para caracterizar a cada una de las mamografías. La segunda parte de esta sección, muestra los resultados obtenidos y la implementación de una red neuronal que utiliza los descriptores seleccionados anteriormente, para clasificar a las mamografías según los cuatro tipos de densidades mamográficas sugeridas por BI-RADS. Aquí se ensayan distintas redes, y se seleccionan las mejores dependiendo de los errores de clasificación y de generalización que surge del entrenamiento y testeo de las mismas.

5.5.1 Ensayo de algoritmos y selección de descriptores

Se aplicaron los 15 algoritmos seleccionados sobre las 168 mamografías. Los parámetros de entrada fueron en un principio estudiados heurísticamente: retardo de 1 y *embedding* entre 3 y 7. Sin embargo, se observó que ciertos algoritmos se obtuvieron una tendencia a incrementar el rango dinámico de los resultados a medida que el valor de *embedding* aumenta. Por este motivo, se utilizaron para todos los algoritmos un retardo de 1 y valores de *embedding* entre 3 y 10.

Para visualizar las similitudes y diferencias entre las imágenes analizadas, se muestra en la Tabla 19 solo la región de las mamografías procesadas por los algoritmos para las cuatro densidades, ya que de la mamografía completa se selecciona un sector rectangular perteneciente al tejido glandular de la mama.

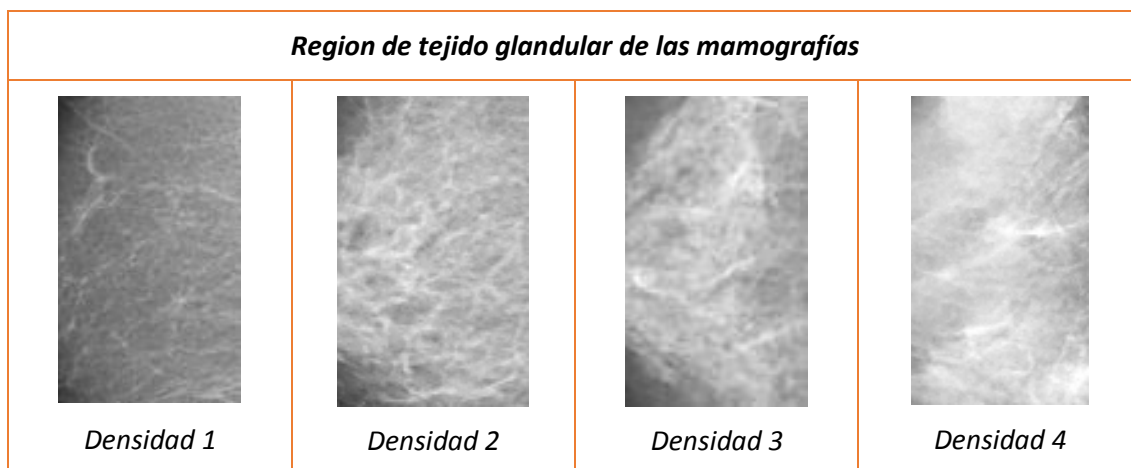


Tabla 19: Tejido glandular de 4 mamografías que representan las distintas densidades. Imágenes reales procesadas por los algoritmos.

En los ensayos realizados con imágenes de texturas reales, los valores de entropía pudieron diferenciarse para cada una de las superficies analizadas, ya que las mismas presentan naturaleza de origen muy variada y rugosidad fácilmente diferenciada por inspección visual. Aquí, se puede adelantar que los valores de entropías serán muy cercanos para las imágenes que representan la región glandular de las mamografías, ya que las mismas muestran características muy similares entre sí. El desafío impuesto a los algoritmos, es presentar diferencias entre los descriptores de entropía para las distintas densidades, o como un desafío menos exigente, que pueda distinguir entre subclases de densidades.

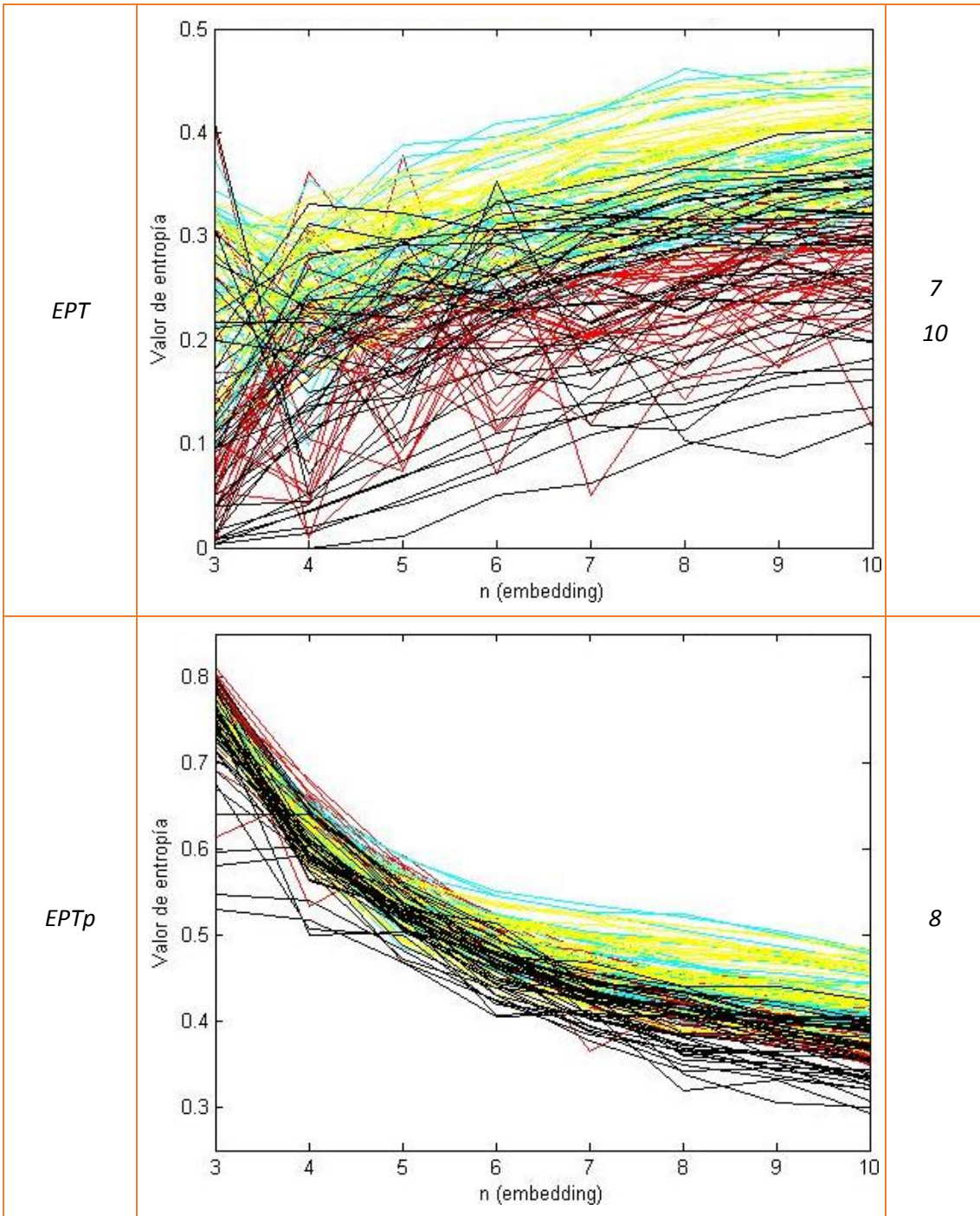
Se planteó como hipótesis que se obtienen resultados de EP similares para las densidades de los extremos (1 y 4), ya que ambas se caracterizan por tener valores de gris similares entre vecindades, lo que representa un nivel de entropía bajo de la imagen. La misma hipótesis se planteó para las mamografías de densidades medias (2 y 3), ya que tal como se ve en las mamografías categorizadas por niveles de densidad según BI-RADS (ver Ilustración 35), los niveles de gris se presentan con grandes variaciones entre vecinos cercanos, por lo tanto, un nivel de entropía más alto representaría a las mamografías de estas densidades.

Luego de ensayar los 15 algoritmos en todas las mamografías, se seleccionaron los descriptores de densidad mamográfica, con el criterio de valor del parámetro n donde la entropía de las distintas densidades difiriera más marcadamente unas de las otras, o donde se pueda distinguir entre los conjuntos de densidades de los que se esperan valores de EP diferentes. Es decir que el valor de n elegido es donde se presente el mayor rango dinámico.

Se describe el procedimiento para la selección del descriptor adecuado en cada algoritmo, para ello se muestran los resultados de todos los algoritmos ensayados en la Tabla 20, Tabla 22 y Tabla 23. Cada una de las tablas contiene la información de los métodos separados en tres grupos: EP de propuesta 1, EP para imágenes filtradas y EP para matrices de entropías. Las ilustraciones de las tablas muestran los valores de entropía en función del *embedding*, representando a la Densidad 1 (D1) en color Rojo, Densidad 2 (D2) en Verde, Densidad 3 (D3) en Amarillo y a la Densidad 4 (D4) en color Negro. Mediante la observación de los colores, se puede determinar el valor de *embedding* óptimo, como aquel donde se presenta mayor rango dinámico en los gráficos, mientras las densidades o subconjuntos de densidades se puedan diferenciar. Al

final de cada tabla se resume el comportamiento de cada algoritmo y se define para cada uno de ellos, el descriptor elegido para caracterizar a las mamografías.

Resultados para algoritmos de Propuesta 1		
<i>Algoritmo</i>	<i>Resultados</i>	<i>n óptimo</i>
<i>EP</i>		7 10
<i>EPP</i>		7 10



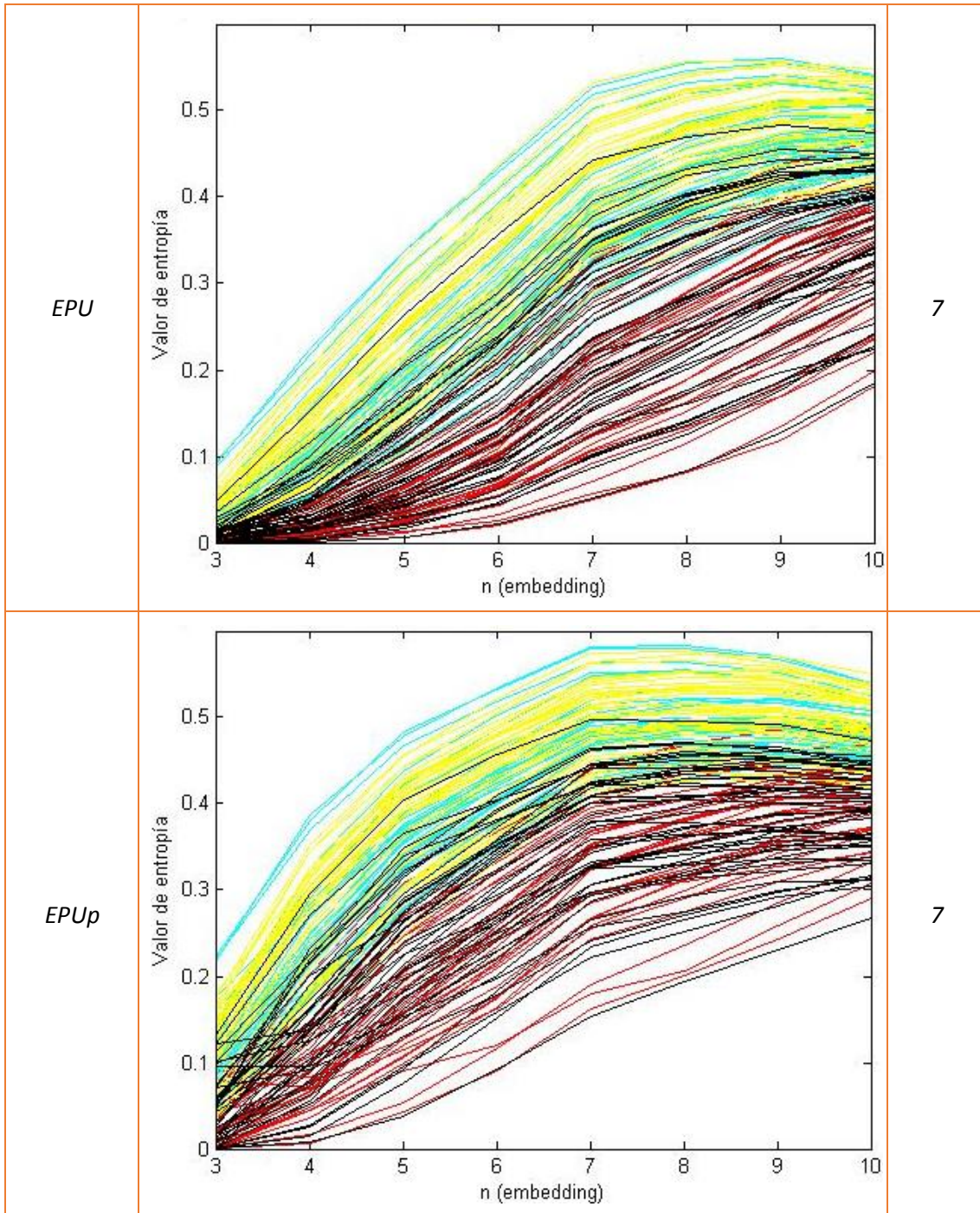


Tabla 20: Resultados de los algoritmos de la propuesta 1 para las 168 mamografías obtenidas del banco de datos BI-RADS. Densidad1-Rojo, Densidad2-Verde, Densidad3-Amarillo, Densidad4-Negro

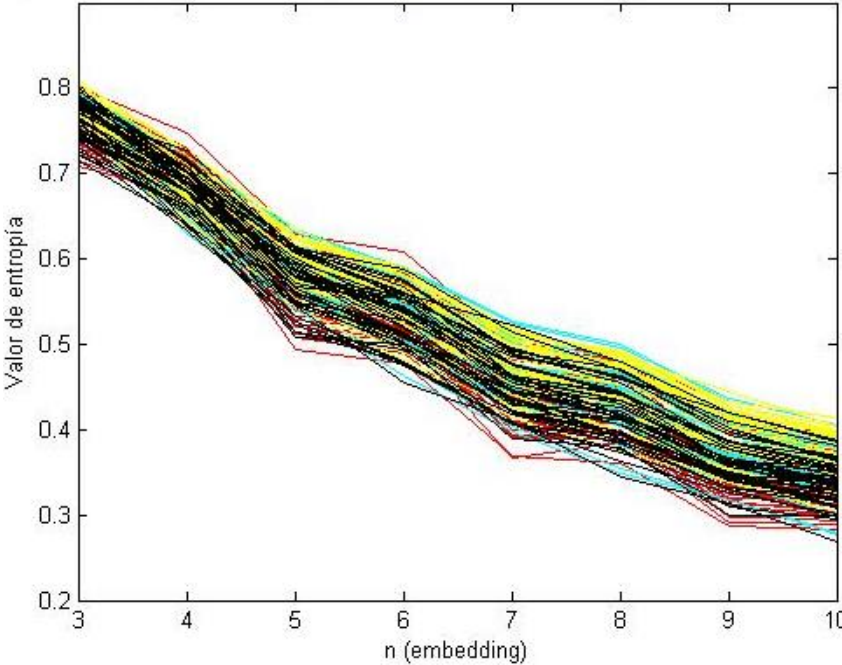
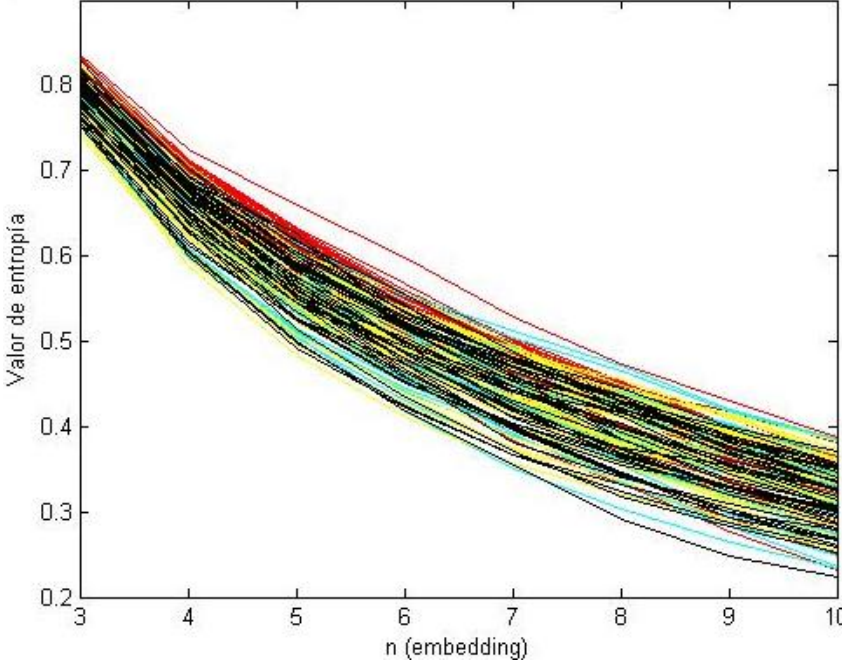
Los algoritmos de la propuesta 1 distinguen entre conjuntos de clases de densidades, pero ninguno las diferencia de manera individual. Este comportamiento fue previsto anteriormente, ya que las imágenes de mamografías procesadas presentan similitudes en sus variaciones de tonos de gris, como se ve en la Tabla 19. El comportamiento de cada algoritmo y la justificación en la elección de los valores de n óptimos, se explican a continuación.

Algoritmo	Detalles de comportamiento (Propuesta 1)	Descriptor
<i>EP</i>	El rango dinámico (RD) de los resultados se mantiene constante para valores de <i>embedding</i> igual o mayores a 7. Además, en $n=7$ la clase D1 se diferencia de todas las restantes, mientras que en $n=10$ la clase D4 se distingue de las restantes. Por estas características, se eligen dichos tamaños de <i>embedding</i> como óptimos, obteniendo dos descriptores.	<i>EP(7)</i> <i>EP(10)</i>
<i>EPP</i>	En este caso RD se mantiene constante hasta $n=7$, valor de <i>embedding</i> donde D1 se diferencia de las restantes densidades. El ED disminuye para valores de n mayores. Sin embargo, es para $n=10$ donde se presenta la mayor separación entre clases, pudiendo diferenciar claramente a D4 de las restantes.	<i>EPP(7)</i> <i>EPP(10)</i>
<i>EPT</i>	El RD se mantiene constante para todo valor de <i>embedding</i> . En $n=7$ y $n=10$ los resultados de EPT para D1 y D4 se separan de los de D2 y D3, se puede ver que las líneas verdes y amarillas se mantienen separadas de las rojas y negras. Los resultados muestran muchas fluctuaciones para distintos valores de <i>embedding</i> , este comportamiento es el esperado debido al procesamiento que realiza el algoritmo con los píxeles.	<i>EPT(7)</i> <i>EPT(10)</i>
<i>EPT_p</i>	Las densidades presentan valores muy similares de entropía para todas las clases hasta $n=6$, valor de <i>embedding</i> a partir del cual D1 y D4 se diferencian de las restantes clases y el RD comienza a aumentar hasta $n=8$ a partir de este valor RD se mantiene constante. Este último es el valor donde las subclases D1-D4 se distinguen más respecto de D2-D3.	<i>EPT_p(8)</i>
<i>EPU</i>	La subclase D1-D4 se diferencia de la subclase D2-D3 para todos los valores de <i>embedding</i> . El RD es variable para todo valor de <i>embedding</i> , presentando su mayor amplitud en $n=7$.	<i>EPU(7)</i>
<i>EPUp</i>	El comportamiento de este algoritmo es muy similar al anterior, RD variable para todos los n , con máxima amplitud en $n=7$. Las subclases de densidades también mantienen sus valores de entropías separados en todos los tamaños de <i>embedding</i> .	<i>EPUp(7)</i>

Tabla 21: Detalles del comportamiento de los algoritmos de la propuesta 1. Descriptores resultantes del análisis de n óptimo.

De la misma manera que para la propuesta 1, se analiza el comportamiento para los algoritmos de la propuesta 2. La Tabla 22 muestra los resultados obtenidos al pre filtrar las imágenes antes de aplicar el método EP. Luego se analiza el comportamiento de cada uno de ellos y se justifica la selección de los descriptores obtenidos de cada algoritmo.

Resultados para algoritmos de Propuesta 2 – EP para imágenes filtradas

Algoritmo	Resultados	n óptimo
<p><i>Filtro Media</i></p>		-
<p><i>Filtro Mediana Aritmética</i></p>		-

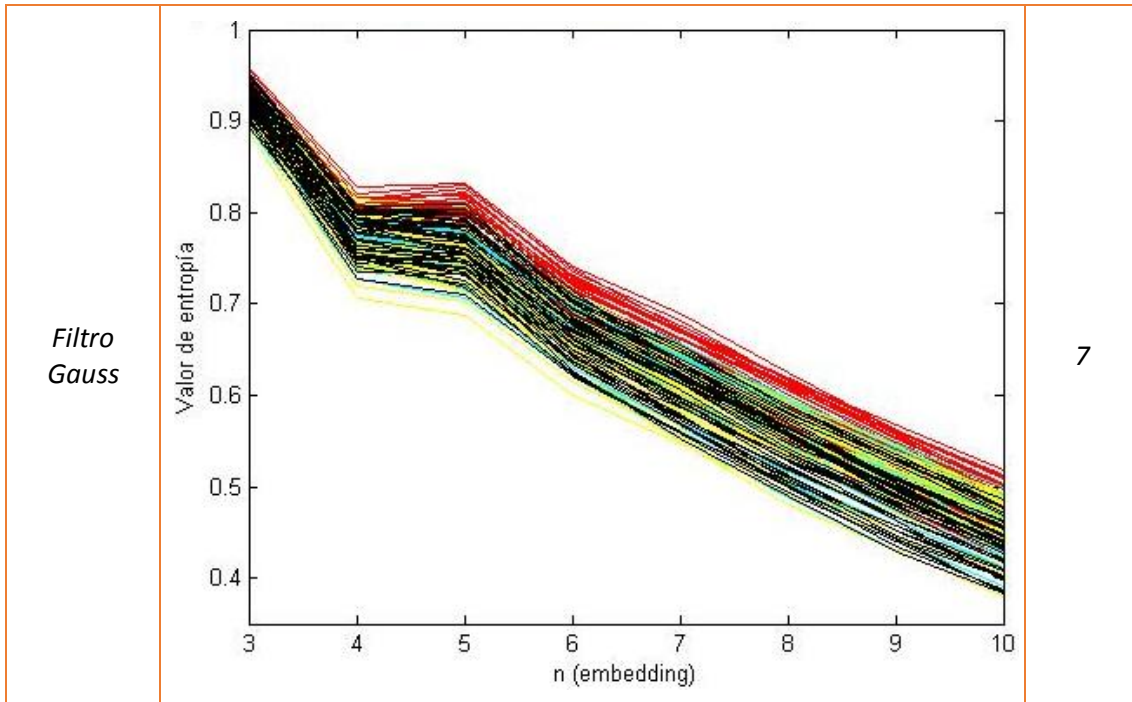
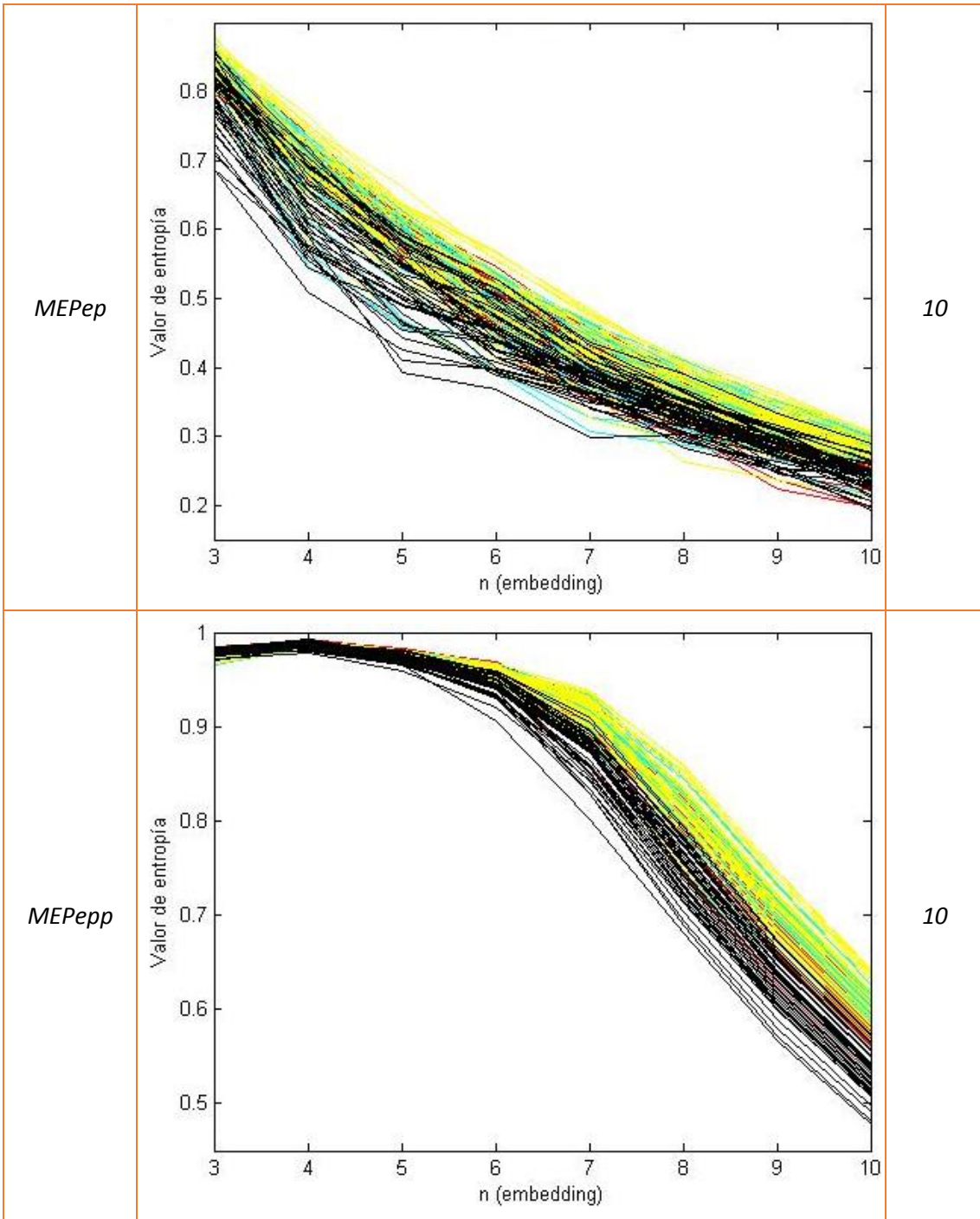


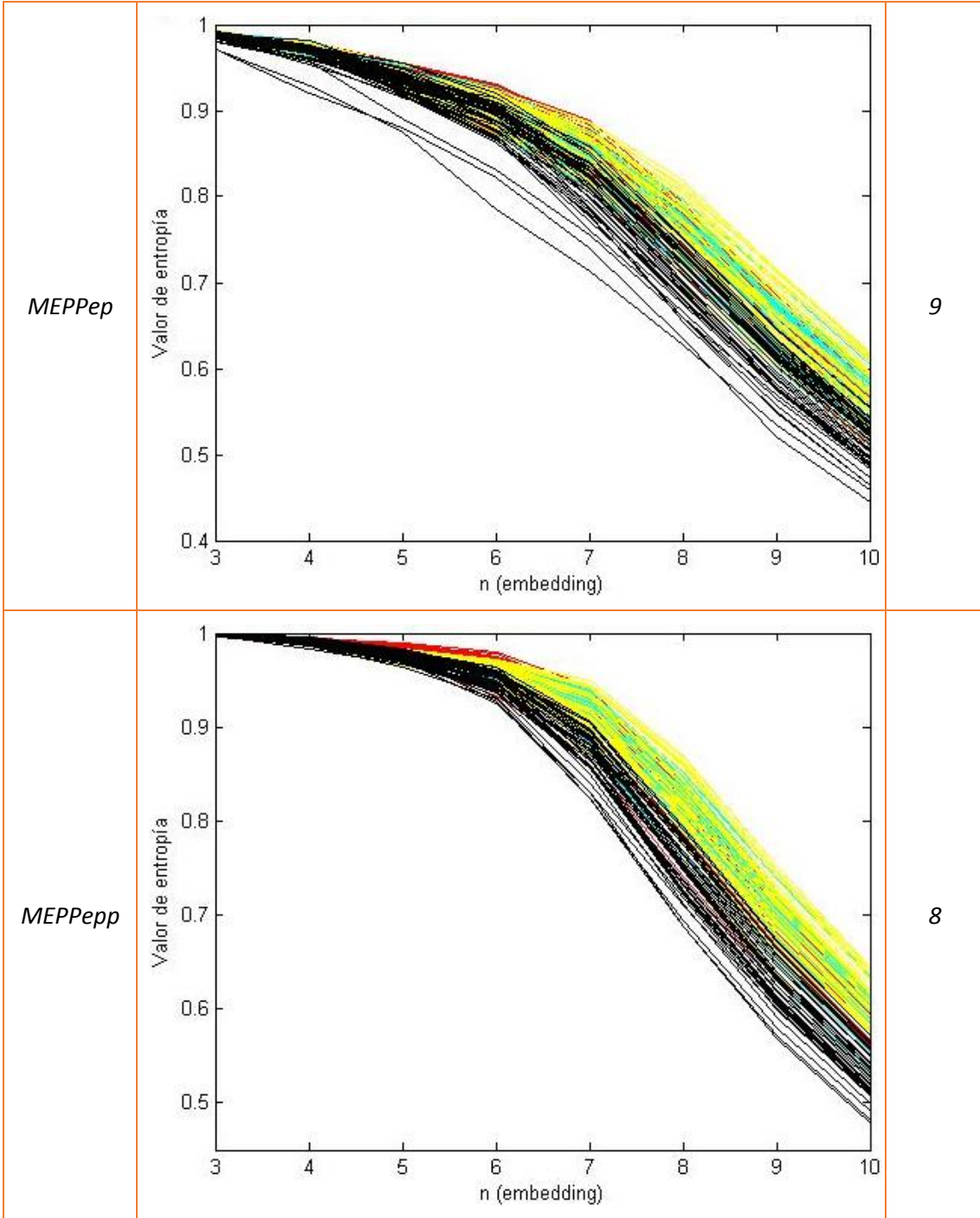
Tabla 22: Resultados del algoritmo EP para imágenes filtradas para las 168 mamografías obtenidas del banco de datos BI-RADS. Densidad1-Rojo, Densidad2-Verde, Densidad3-Amarillo, Densidad4-Negro

La aplicación de filtros de media y mediana aritmética no presenta ventajas para este caso práctico, sino por lo contrario no puede caracterizar las densidades por subclases y mucho menos por las cuatro clases individualmente. Por este motivo no se selecciona ningún descriptor de estos métodos. En el caso de la utilización de un filtro de gauss, se mantiene para todo tamaño de *embedding* la diferenciación de la clase D1 de las restantes. En este algoritmo el RD aumenta hasta $n=7$, valor a partir del cual el RD se mantiene constante y donde D1 presenta entropías separadas de las otras tres clases, por este motivo se concluye como el valor de *embedding* óptimo, resultado el descriptor $EPFg(7)$.

Por último se analiza la aplicación del algoritmo EP para matrices de entropías en la región glandular de las mamografías obtenidas. Los resultados se presentan en la Tabla 23.

Resultados para algoritmos de Propuesta 2 – EP y EPP para Matrices de entropías		
<i>Algoritmo</i>	<i>Resultados</i>	<i>n óptimo</i>





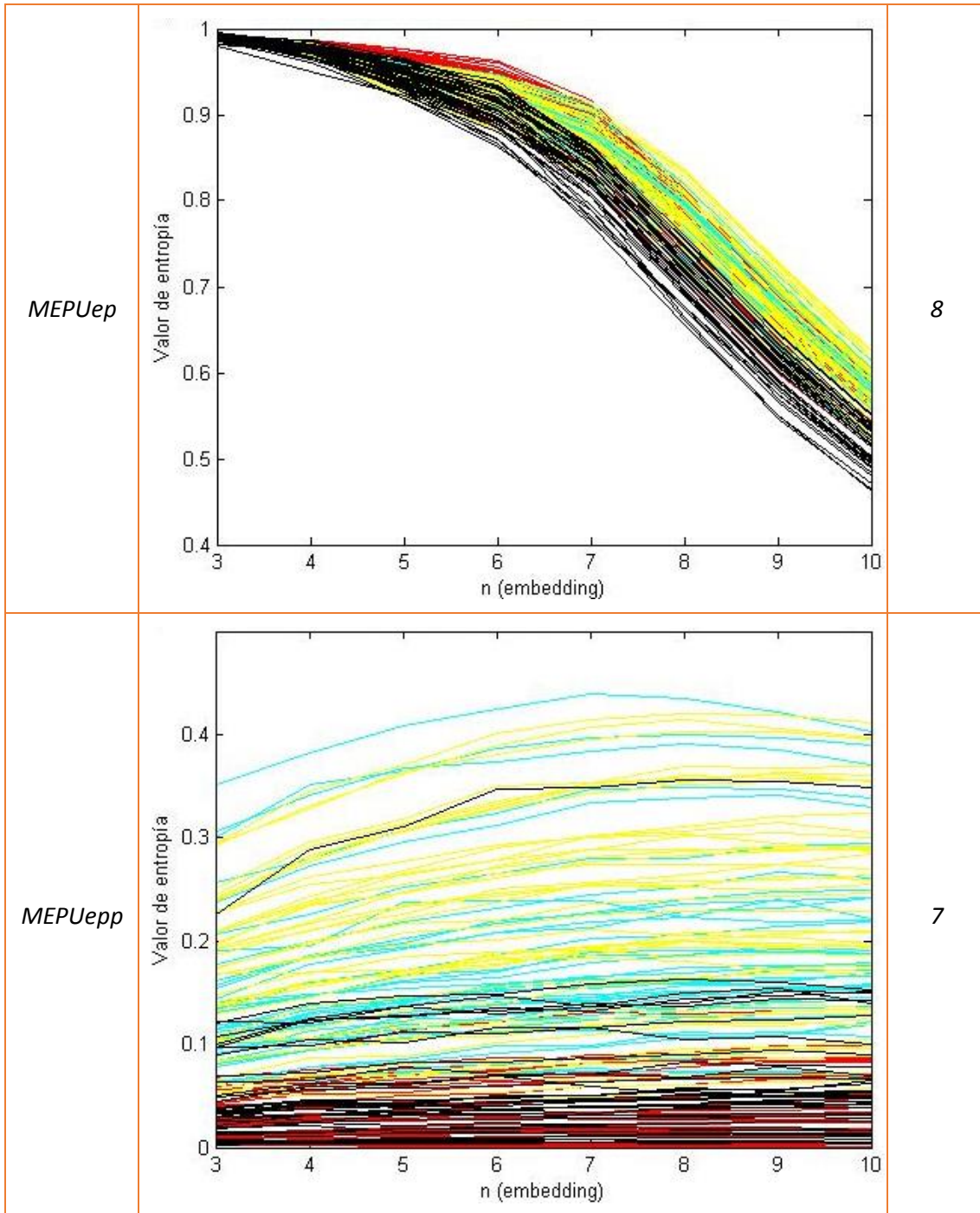


Tabla 23 Resultados del algoritmo EP para matrices de entropías para las 168 mamografías obtenidas del banco de datos BI-RADS. Densidad1-Rojo, Densidad2-Verde, Densidad3-Amarillo, Densidad4-Negro

Al igual que para los métodos anteriores, se realiza la justificación de la elección de los descriptores en base al análisis de los gráficos de cada uno de los algoritmos, para la obtención del n óptimo.

Algoritmo	Detalles de comportamiento (EP para matrices de entropías)	Descriptor
<i>MEPep</i>	El rango dinámico (RD) presenta una leve disminución a medida que el <i>embedding</i> aumenta. Sin embargo, aunque el rango dinámico sea más chico, en $n=10$ la clase D4 se distingue de las tres restantes. Por estas características, se elige este como el tamaño de <i>embedding</i> óptimo.	<i>MEPep(10)</i>
<i>MEPepp</i> <i>MEPPep</i> <i>MEPPepp</i> <i>MEPUep</i>	Los resultados de estos cuatro métodos presentan el mismo comportamiento en los siguientes aspectos: el RD incrementa hasta $n=8$, a partir de ese valor se mantiene constante. Los valores de entropía comienzan en valores aproximados a 0,9 para $n=3$ y terminan alrededor de 0,5 para $n=10$. A partir de $n=7$ las clase D4 pueden diferenciarse de las restantes, presentando esta siempre valores de entropía más bajos. Los descriptores se determinan a partir de la observación detallada de los resultados para <i>embedding</i> de 8, 9 y 10, eligiendo aquel donde la clase D4 esté más diferenciada de las demás. Si a distintos valores de <i>embedding</i> los resultados tienen características similares, se prioriza el caso de <i>embeddings</i> menores, para ahorrar costo computacional en el cálculo del descriptor.	<i>MEPepp(10)</i> <i>MEPPep(9)</i> <i>MEPPepp(8)</i> <i>MEPUep(8)</i>
<i>MEPUepp</i>	El RD presenta leves variaciones para cada n , con máxima amplitud en $n=7$. La subclase D1-D4 se diferencia de la subclase D2-D3 para todos los valores de <i>embedding</i> .	<i>MEPUepp(7)</i>

Tabla 24: Detalles del comportamiento de los algoritmos de EP para matriz de entropías de la propuesta 2. Descriptores resultantes del análisis de n óptimo.

Como se presentó en el análisis de los resultados, algunos algoritmos pueden caracterizar entre la clase D4 y las restantes, y otros entre las subclases D1-D4 y D2-D3, pero ninguno pudo discriminar a D2 y D3 de manera individual.

Para representar a cada una de las mamografías, se seleccionaron 11 descriptores de los 16 obtenidos. Se eligió solo un descriptor de los algoritmos que presentan dos descriptores (EP, EPP) y no se utilizaron los descriptores obtenidos por los métodos EPT y EPFG, ya que el primer caso presenta valores de entropía que varían mucho dependiendo del valor del parámetro de *embedding* utilizado, y en el caso del segundo, los valores de entropía en n óptimo son muy próximo (RD muy chico). Entonces, cada una de las 168 mamografías de estudio, se representan por 11 características diferentes. Así, se puede formar un vector característico de 11 elementos por imagen, del que se espera una representación unívoca.

Una vez obtenido el vector característico para cada mamografía, se propone realizar el entrenamiento de una red neuronal multicapa como clasificador, con el fin de evaluar la potencialidad de los descriptores como caracterizadores de la densidad mamográfica, en concordancia con la nomenclatura BI-RADS.

5.4.2 Clasificador de densidades mamográficas

Se propone una red con 11 entradas, que representan a cada uno de los descriptores elegidos, y 4 salidas, correspondientes a cada una de las clases de densidad (D1, D2, D3 y D4). Los pesos se ajustan durante el entrenamiento según el método del descenso por el gradiente.

El entrenamiento se detiene considerando las capacidades de generalización de la red, según datos de validación. Se mantiene un conjunto de datos de test para monitorear los pasos durante el entrenamiento. Los porcentajes asignados de cada subconjunto de datos, son: 70% de los datos para entrenamiento (118 imágenes), 15% para validación (25 imágenes), 15% para testeo (25 imágenes). La selección de datos para cada subconjunto se realiza de manera aleatoria. Se generan varias redes para clasificar las densidades mamarias, y para todas ellas se utiliza esta distribución de los datos.

La cantidad de capas ocultas y de neuronas por capa oculta se determinó mediante la utilización de algoritmos genéticos, cuya función de costo representa el error de entrenamiento o testeo dependiendo el caso y las variables fueron la cantidad de neuronas por capa oculta. De esta manera, se compararon los resultados de varias redes óptimas propuestas por algoritmos genéticos con distintas condiciones para optimizar la función error.

Una vez diagramada la arquitectura de las redes a utilizar, se realizó una validación cruzada de tipo Hold-out, para una adecuada estimación del error. Se realizan 100 ciclos de entrenamiento con diferentes selecciones aleatorias de datos de test y se calcula la respuesta de la red entrenada ante el ingreso de datos de test. Finalmente, se promedia el resultado de los errores de test para obtener una estimación de error final.

Las arquitecturas utilizadas se justificaron mediante la optimización de la función de costo del algoritmo genético. Para comenzar, se optimizó la función de error de entrenamiento de una red neuronal muy simple, fijando la cantidad de capas solo a una y utilizando como variable la cantidad de neuronas en esa única capa, las que pueden variar de 1 a 100. La red obtenida lleva el nombre de RED 1. La performance de la red se determina mediante los errores de entrenamiento y test. La cantidad de neuronas óptima determinada por el algoritmo genético fue de 72 neuronas en la única capa oculta. Mediante la utilización del método Hold-out, se obtuvo un error de entrenamiento de 2,98%, sin embargo la red no presenta buenas características de generalización, ya que se obtuvo un error de test de 28,42%.

Continuando con redes de arquitecturas simples, se utilizó el algoritmo genético con los parámetros anteriores, pero se cambió la función de costo para optimizar el error de test. En este caso se obtuvo la RED 2, con 20 neuronas para la única capa oculta. Mediante el método de Hold-out se llegó a un error de entrenamiento de 5,58%, este resulta mayor al anterior, sin embargo bajó considerablemente el error de test a 19,36%. Estos datos indican que esta red presenta una mejor performance que la anterior, ya que se acepta el pequeño aumento en el error de entrenamiento frente a la significativa mejora en la capacidad de generalización de la red.

Con la intención de bajar el error de test, se probaron distintas redes utilizando el algoritmo genético configurado para dos variables: 1) cantidad de capas ocultas desde 1 a 5 y 2) cantidad de neuronas por capa oculta desde 0 a 20. Mediante la optimización de la función del error de entrenamiento o de test, no se encontraron arquitecturas de redes que mejoren significativamente el error de test. Este hecho es atribuido al significativo error que surge de la

confusión entre las densidades D2 y D3, ya que las mismas presentan valores de entropía dentro del mismo rango para todos los algoritmos.

En vista de los resultados obtenidos para la clasificación de las cuatro clases D1, D2, D3 y D4, la arquitectura elegida es la de la RED 2: una capa oculta de 20 neuronas. El tiempo de entrenamiento promedio fue de 1 segundo, durante los que realizaron un promedio de 10 iteraciones de entrenamiento y 6 chequeos de validación.

En la Tabla 25 se muestra la matriz confusión resultante del entrenamiento de la RED 2. En la diagonal principal se acumula la cantidad de mamografías clasificadas correctamente por la red, que son aquellas que se clasificaron a la salida de la red con la etiqueta esperada. Los resultados se muestran por cantidad de mamografías y porcentaje que dicha cantidad representa del total del subgrupo al que pertenece.

<i>Mamografía...</i>	<i>Clasificada como D1</i>	<i>Clasificada como D2</i>	<i>Clasificada como D3</i>	<i>Clasificada como D4</i>
<i>Etiquetada como D1</i>	26 92,86%	2 6.67%	0 0.00%	1 3.45%
<i>Etiquetada como D2</i>	1 3,57%	24 80.00%	1 3.03%	0 0.0%
<i>Etiquetada como D3</i>	0 0.00%	4 13.33%	31 93.94%	0 0.0%
<i>Etiquetada como D4</i>	1 3,57%	0 0.00%	1 3.03%	28 96.55%

Tabla 25: Matriz confusión de entrenamiento. RED 2

Se observa en la matriz confusión de entrenamiento, que la clasificación para la clase D2 fue la menos eficiente, con un 80.00% de acierto y con errores dispersos en las clases D1 y D3, ésta última en mayor cantidad. La mayor cantidad de errores se produce en la clasificación de las densidades D2 y D3, lo que es de esperarse por la similitud de dichas clases, concluyendo que ésta es la zona con mayor cantidad de confusiones, resultado esperado debido a la similitud de la entropía de las mamografías con estas densidades. Sin embargo, el entrenamiento resultó con mayor cantidad de aciertos para las clases D1 y D4, densidades extremas.

El indicador más importante de la performance de la red, es la capacidad de generalización de la misma, lo que representa la capacidad la red de clasificar datos nuevos. Para cuantificar esta característica, se consulta a la red con imágenes no incluidas en los datos de entrenamiento, una vez que la misma está entrenada, y se analiza la concordancia entre las etiquetas de los

datos de entrada y las salidas. La clasificación que resulta de esta prueba indica la capacidad de generalizar de la RED 2.

La Tabla 26 representa la matriz de confusión de los datos de test, su diagonal principal indica la cantidad de entradas nuevas que fueron clasificadas correctamente, mientras que los porcentajes indicados por fuera de esta diagonal, cuantifican el nivel de confusión de la red ante eventos nuevos.

<i>Mamografía...</i>	<i>Clasificada como D1</i>	<i>Clasificada como D2</i>	<i>Clasificada como D3</i>	<i>Clasificada como D4</i>
<i>Etiquetada como D1</i>	6 85.71%	0 0.00%	0 0.00%	1 14.29%
<i>Etiquetada como D2</i>	0 0.00%	4 80.00%	2 40.00%	0 0.00%
<i>Etiquetada como D3</i>	0 0.00%	1 20.00%	3 60.00%	0 0.00%
<i>Etiquetada como D4</i>	1 14.29%	0 0.00%	0 0.00%	6 85.71%

Tabla 26: Matriz confusión de test. RED 2

Los resultados evidencian que las mamografías categorizadas con densidades del subconjunto D1-D4 o D2-D3 son cuantificadas con valores similares de entropía, debido a la naturaleza de sus rugosidades. Esto se hace visible en la matriz confusión de test, donde imágenes etiquetadas como D3 son clasificadas como D2 en un 20% de los casos y clases etiquetadas como D2 son clasificadas como D3 en un 40% de los casos. Así, se puede asumir que la red obtenida no es un buen generalizador para discriminar entre mamografías que presenten densidades medias D2 o D3.

Observando ambas matrices de confusión, se concluye que la red fue más efectiva para clasificar y generalizar densidades extremas D1 y D4.

Se recuerda que mediante la validación cruzada de la RED2 se obtuvo un error de test de 19,36%. Si bien en los entrenamientos se vieron mejores porcentajes de acierto en la clasificación, con errores de entrenamiento del 0.8%, los errores de test prevalecieron para las clases D2 y D3. Se atribuye la persistencia de este error a distintos factores, uno de ellos es la considerable variabilidad intra e inter observador respecto del cálculo de la densidad mamaria entre dos categorías de densidad adyacentes que se presentan en las imágenes mamográficas estudiadas. Por otra parte, se observa una diferencia muy baja entre la mama más densa de la categoría de

densidad más baja y la mama menos densa de la categoría superior subsiguiente. Estos factores limitan la relevancia clínica que tiene la clasificación de la densidad mamaria de cada mujer, evidenciando la necesidad de indicadores más robustos.

Los errores entre clases adyacentes tienen una fuerte relación con las diferencias de clasificación, presente en el banco de datos debido a la subjetividad de observador experto, además de los errores del método de clasificación. Es decir, aquellas mamografías categorizadas como densidad 2 que fueron clasificadas como de densidad 3 o al revés, contienen en su porcentaje error la suma de los errores producidos en el banco de datos más el error propio del clasificador mediante estos nuevos descriptores de texturas.

En el caso de los errores entre las clases extremas, el error es atribuido directamente al método de clasificación, es decir que las mamografías categorizadas en el banco de datos con densidad 1, que fueron clasificadas por la red neuronal como un caso de densidad 4, el error es atribuido completamente a la sensibilidad de los descriptores de entropía utilizados, que como ya se ha mencionado, presentan valores similares de entropía debido a la distribución homogénea entre los tonos de gris que conforman el tejido glandular.

Los resultados y su análisis permiten concluir que los descriptores considerados en este trabajo son de especial interés para la detección de diferentes densidades en mamografías. Se probaron varios clasificadores, y el error de general de test no pudo ser reducido, por lo tanto se consideró la mejor opción a aquella arquitectura de red neuronal más sencilla. Sin embargo se considera importante continuar el estudio para evaluar su capacidad de discriminación mediante otras técnicas y eventualmente en combinación con descriptores provenientes de otros paradigmas, para lograr una caracterización de las cuatro clases con errores más bajos.

En vista de los errores producidos en la clasificación de nuevos datos para D2 y D3, se propone un nuevo clasificador de tres clases, las que se separan en clases extremas y medias, resultando una categorización de los casos en 3 niveles de riesgo según la densidad mamográfica: 1) nivel bajo, 2) nivel medio, 3) nivel alto. Cada uno de estos niveles incluye a los las categorías BI-RADS siguientes:

<i>Nivel de riesgo de ocultamiento de lesiones</i>	<i>Densidades que incumben</i>	<i>Procedimiento del especialista radiólogo en cada caso</i>
Bajo	D1	La paciente presenta un nivel de densidad mamográfica que permite al profesional radiológico determinar un diagnóstico por simple inspección. En caso de no encontrar anomalías, no es necesaria la realización de otros estudios más complejos, ya que la mamografía es material suficiente para categorizar a la paciente como sana.
Medio	D2 y D3	Esta categoría engloba dos casos con diferente nivel de importancia. El caso de D2, es similar al de D1, ya que en las mamografías que presentan este nivel de densidad, no son necesarios exámenes más complejos si el experto radiólogo no aprecia anomalías. El caso de la clase D3 requiere de mayor

		atención que la anterior, ya que la densidad del tejido mamario puede ocultar lesiones. En este caso se recomiendan estudios más complejos para determinar que la paciente está totalmente sana, sin embargo es decisión del médico solicitante y el médico radiólogo tomar la decisión definitiva en cuanto a los procedimientos de diagnóstico y el tratamiento, a la luz de todas las circunstancias que presente cada estudio en particular.
Alto	D4	Dentro de la legislación de muchos países, este nivel de densidad se utiliza como disparador único e independiente para la realización de monitoreo e intervenciones más complejas, como ecografías o resonancias magnéticas. Por lo que se recomienda al especialista indicar estos procedimientos en el caso de densidades D4, ya que no es posible asegurar la ausencia de anomalías por simple inspección de la mamografía de estas pacientes.

Tabla 27: Conformación de las 3 clases para clasificar entre nivel de densidad baja-media-alta. Cada una de estas 3 clases contiene a una o más clases de las densidades determinadas por el sistema BI-RADS.

Para comprobar la capacidad de caracterización de los 11 descriptores elegidos para representar a cada mamografía, se realiza una red neuronal, la que denominaremos como RED 3, de 11 entradas y 3 salidas (las clases riesgo bajo-medio-alto). Los porcentajes de asignación de los datos para entrenamiento, validación y test son iguales que para el caso anterior. La cantidad de capas ocultas y neuronas por capa se determina mediante la optimización de la función de error de test utilizando un algoritmo genético, cuyas variables fueron la cantidad de capas que pueden variar de 0 a 10, y la cantidad de neuronas con posibilidad de variación entre 0 y 100. Los resultados del algoritmo entregaron una arquitectura de red óptima de 7 capas, cuya cantidad de neuronas por capa oculta es: [39 28 30 17 6 37 13]. Se estimaron los errores mediante el método de hold-out con 100 iteraciones, resultando un error de entrenamiento total de 2,34% mientras que el error de test total estimado fue de 8,14%, se recuerda que los errores totales son el promedio de los errores producidos en la clasificación o testeo de cada clase. El tiempo promedio del entrenamiento es de 1 minuto y la cantidad de iteraciones promedio de 12.

En este caso los errores estimados son mucho menores que para el caso de la categorización de 4 clases. Esto evidencia que la mayor cantidad de errores se producen en la confusión de la RED 2 entre las densidades D2 y D3.

En la Tabla 28 se muestra la matriz confusión resultante del entrenamiento de la RED 3. Al igual que el caso anterior, en la diagonal principal muestra la cantidad de aciertos que se producen durante el entrenamiento de la red, en cantidad de mamografías bien clasificadas y en el porcentaje que las mismas representan respecto de la cantidad total de su clase.

<i>Mamografía...</i>	<i>Clasificada como Riesgo Bajo (D1)</i>	<i>Clasificada como Riesgo Medio (D2 y D3)</i>	<i>Clasificada como Riesgo Alto (D4)</i>
<i>Etiquetada como Riesgo Bajo (D1)</i>	27 96.43%	1 1.56%	0 0.00%
<i>Etiquetada como Riesgo Medio (D2 y D3)</i>	0 0.00%	62 96.88%	0 0.00%
<i>Etiquetada como Riesgo Alto (D4)</i>	1 3.57%	1 1.56%	28 100.00%

Tabla 28: Matriz confusión de entrenamiento. RED 3

El entrenamiento de la RED 3 muestra que la clase *Riesgo Alto* fue la mejor clasificada con un 100% de efectividad. Como ya fue mencionado, categorización de este caso es de especial interés por su importancia clínica, ya que las densidades extremadamente densas requieren de más atención por parte de los médicos responsables de los pacientes que lo presentan. La clasificación de *Riesgo Medio* es alta, presentado errores en sus dos clases adyacentes, lo que se puede atribuir a errores producidos por el profesional radiólogo al momento del etiquetado de las clases y los valores de los descriptores, por ser imágenes cuyas texturas presentan rugosidades cercanas. Si se observan los resultados del entrenamiento de la clase *Riesgo Bajo* en la columna de la izquierda de la matriz de confusión de entrenamiento de la RED 3, se nota que el único error es la confusión entre las clases de riesgos extremos (D1 y D2), lo que es atribuido netamente a los descriptores utilizados, y no a errores de clasificación producidos por los profesionales radiólogos, ya que estas densidades son fácilmente reconocibles por simple inspección de las mamografías, sin embargo los algoritmos mostraron dificultades en diferenciar las densidades extremas entre sí, al igual que las densidades medias entre sí, por presentar niveles de entropía similares.

El entrenamiento de esta red tuvo menos errores que para la RED 2. Por lo tanto se espera que la categorización de mamografías nuevas con esta arquitectura de red presente también menos errores. De esta manera se pueden utilizar los descriptores tal como están para funcionar como generalizadores de 3 clases en vez de 4, y producir mejoras en los algoritmos o la implementación de otro tipo de descriptores, para la categorización de las 4 clases por separado.

El poder de generalización de la red se observa en la respuesta para la clasificación de nuevos casos de la RED 3, representada en la matriz de confusión de test de la Tabla 29.

<i>Mamografía...</i>	<i>Clasificada como Riesgo Bajo (D1)</i>	<i>Clasificada como Riesgo Medio (D2 y D3)</i>	<i>Clasificada como Riesgo Alto (D4)</i>
<i>Etiquetada como Riesgo Bajo (D1)</i>	5 100%	1 8,33%	0 0.00%
<i>Etiquetada como Riesgo Medio (D2 y D3)</i>	0 0.00%	11 91.67%	1 14,29%
<i>Etiquetada como Riesgo Alto (D4)</i>	0 0.00%	0 0.00%	6 85.71%

Tabla 29: Matriz confusión de test. RED 3

La efectividad del 100% en la generalización de las mamografías que no presentan riesgo de ocultamiento de lesiones, permite estimar que la RED 3 se muy útil para descartar a casos que no necesiten inspección de un profesional. Es decir, que se puede proponer una primera selección automática en el banco de datos estudiado por los profesionales, de manera de achicar la cantidad de casos a estudiar por los mismos. Sin embargo, ningún descriptor automático es determinante en un diagnóstico médico, sólo pueden actuar como información que utilice el médico radiólogo para decidir subjetivamente sobre la gravedad del caso.

Para las categorías de riesgo medio y alto, los descriptores presentaron bajos porcentajes de error de generalización. En ambos casos los errores se presentaron entre clases adyacentes, atribuyéndose los mismos a la clasificación realizada por los observadores y a los valores cercanos de las entropías de los descriptores para las distintas clases.

Por último, se ensayó con una nueva red, la que diferencia entre solo dos tipos de clases, casos de *Bajo Riesgo* y de *Alto Riesgo*. De esta manera, se engloban en la primera clase a las densidades de tejido mamario D1 y D2, mientras que la segunda representa en subconjunto de densidades D3 y D4. Estas dos clasificaciones pueden diferenciar a las pacientes en dos grandes grupos, uno es para aquellas que no precisan indicación médica de estudios más complejos, ya que la mamografía es apta para distinguir anomalías o no. El otro caso es para aquellas pacientes cuyo estudio mamográfico no es suficiente para diagnosticar con certeza la presencia o no de anomalías, sino que se requiere para determinarlo exámenes más complejos, seguido de un control más estricto por parte de los profesionales de salud a cargo.

Para este último caso, se entrenó una red de 11 entradas y 2 salidas (*Bajo Riesgo* y *Alto Riesgo*), con los mismos parámetros de distribución de los datos que las redes anteriores. Esta red se nombró RED 4, y la cantidad de capas ocultas y neuronas por capa se determinó por la optimización del error de test de un algoritmo genético. Se observó que aumentar la cantidad de capas ocultas no mejoró la performance de la red, por lo que se utilizó una red de sólo una capa oculta. De este ensayo resultó en 1 segundo con un promedio de 12 iteraciones, una red óptima de 9 neuronas, con errores de entrenamiento y test totales, estimados para 100 iteraciones de 8.00% y 16.00% respectivamente. Este caso mejora la generalización de los casos

a los obtenidos en el caso de la RED 2, siendo el error de test de 19,36%, lo que indica que la RED 4 es mejor generalizadora de casos que la RED 2. Además, presenta beneficios respecto de la RED 3, ya que se distinguen entre los casos de densidad alta y baja, permitiendo que se produzcan menos errores en los casos que requieren de mayor atención (D3 y D4).

Los errores y aciertos durante el entrenamiento se muestran en la matriz de confusión de la Tabla 30. La interpretación de los resultados es la misma que para las matrices de confusión.

<i>Mamografía...</i>	<i>Clasificada como Riesgo Bajo (D1 y D2)</i>	<i>Clasificada como Riesgo Alto (D3 y D4)</i>
<i>Etiquetada como Riesgo Bajo (D1 y D2)</i>	51 89,47%	1 1,59%
<i>Etiquetada como Riesgo Alto (D3 y D4)</i>	6 10,53%	62 98,41%

Tabla 30: Matriz confusión de entrenamiento. RED 4.

Se observa un porcentaje alto de acierto (98,41%), para el caso de las pacientes con riesgo de ocultamiento de lesiones mamarias debido a la alta densidad que presenta el tejido glandular de sus mamas. Esta clasificación de los casos durante el entrenamiento es de gran importancia, alentando el uso de los descriptores para la clasificación en dos clases. Dicha aseveración será determinada en caso de que esta red presente una buena performance para la generalización de datos nuevos, lo que se estudia mediante el análisis de la matriz de confusión de test de la RED 4 presentada en la Tabla 31.

<i>Mamografía...</i>	<i>Clasificada como Riesgo Bajo (D1 y D2)</i>	<i>Clasificada como Riesgo Alto (D3 y D4)</i>
<i>Etiquetada como Riesgo Bajo (D1 y D2)</i>	9 81,82%	2 15,38%
<i>Etiquetada como Riesgo Alto (D3 y D4)</i>	2 18,18%	11 84,62%

Tabla 31: Matriz confusión de test. RED 4.

Para esta red, los errores en todos los casos son atribuidos a las dos causas posibles en este problema: errores de clasificación realizados por el observador al momento de la asignación de etiquetas en el banco de datos, y a los valores cercanos que resultan de los descriptores utilizados.

El error resultante para categorizar datos nuevos fue mayor para la clase *Riesgo Bajo* que para *Riesgo Alto*. Esto indica que la red es mejor generalizadora de los casos de mayor riesgo, lo que resulta conveniente para nuestro objetivo, no tener falsos positivos en los casos de riesgo alto. Sin embargo, el porcentaje de acierto obtenido en la clase *Riesgo Alto* (84,62%), no resulta suficiente como para determinar que con la RED 4 se puede realizar una selección automática de los datos, pero sí, se puede utilizar como soporte de consulta para el profesional al momento de determinar el nivel de densidad pertinente a cada mamografía.

6. Conclusiones

Luego del ensayo de los algoritmos propuestos sobre las imágenes *Lisa* y *Rugosa* para descartar los algoritmos que no presentaron los resultados esperados, se analiza el comportamiento de los métodos frente a imágenes de superficies con diferentes tamaños de texturas, lo que se considera de gran interés, ya que una textura gruesa puede ser la versión escalada de una textura más fina, y las imágenes sintéticas creadas fueron mediante el mismo patrón de forma, repetido con mayor o menor frecuencia y en distintos tamaños, por lo que las texturas pueden representar sus versiones escaladas, lo que permitió obtener la performance de los algoritmos seleccionados frente a este conflicto.

Los resultados fueron alentadores para los métodos que utilizan EPP, ya que para los diferentes grosores de la misma textura el resultado de EPP es el mismo para todos los valores de *embedding*. Mediante los métodos que no utilizan EPP, se obtuvieron mayores valores de entropía para las texturas finas (que representan escalas grandes) y menores para las texturas gruesas (las que representan escalas chicas), por lo que pueden utilizarse estos descriptores como caracterizadores de escalado de una imagen. Dichas observaciones son muy importantes a nivel teórico para ser estudiadas a en trabajos futuros.

Además, se logró caracterizar superficies reales, según la rugosidad de sus texturas, sin perder las ventajas connaturales que la EP presenta para series temporales, principalmente la rapidez de cálculo y la robustez frente al ruido, ya que se pudo obtener un caracterizador de rugosidad en la imagen, sin realizar un pretratamiento de la misma para mejorar todos aquellos problemas que se pudieron haber producido durante la adquisición de la imagen.

Las variantes de cálculo de la EP presentaron ventajas diferentes frente los distintos rangos de rugosidad esperados, por lo que es necesario un análisis previo de la situación de aplicación, para determinar cuál es el o los descriptores a utilizar según el caso.

Otros de los desarrollos importantes realizados en este trabajo, se refiere a los ensayos de estos nuevos descriptores para la caracterización de la densidad mamaria obtenida a partir de mamografías. Considerando la EP y sus variantes, se propusieron 15 algoritmos para calcular descriptores de la rugosidad en cada píxel, formando vectores de intensidades de gris según dos diferentes propuestas de arreglos, de los que se obtuvieron 11 descriptores que caracterizan a cada mamografía.

Los resultados obtenidos son consistentes con los esperados debido a las características de los algoritmos implementados y la información obtenida por el banco de datos. Sin embargo, en lo que compete al interés clínico para la identificación de un grado de densidad mamográfica para cada paciente de estudio, la densidad mayor es la que requiere especial cuidado debido al enmascaramiento que realiza de posibles lesiones. En este sentido, las 3 redes de clasificación propuestas (RED 2, RED 3 y RED 4) brindan los porcentajes más altos de calidad en la clasificación, estimados por validación cruzada.

Este trabajo aporta un paradigma para utilizar el enfoque de Entropía de Permutación para sistemas de dos dimensiones, como lo son las imágenes. Además, marca lineamientos para aplicaciones futuras, donde se utilicen los resultados de estos algoritmos como descriptores de una imagen. Algunos de los campos de investigación y aplicación interesantes son el análisis de imágenes médicas y satelitales o el control de una línea de producción industrial, como también aplicaciones en tiempo real para la detección de características en imágenes de video. Además, al ser una metodología para el procesamiento de series de tiempo basado en análisis ordinales EP es conceptualmente simple, estructuralmente robusto al ruido y computacionalmente muy rápido, lo cual es relevante para diseñar diagnósticos portátiles.

Para el caso práctico ensayado, se concluye que los descriptores de EP utilizados son de gran interés, ya que frente a la necesidad de descriptores más robustos para continuar con las investigaciones sobre implicancia de la densidad mamaria como indicador de riesgo de cáncer de mamas, ante la nueva ausencia de referencias numéricas y justificadas de la categorización de la densidad mamográfica según las organizaciones de regulación del sistema de informes mamográficos BI-RADS, los descriptores basados en EP se presentan como una alternativa a continuar estudiando para la detección de diferentes densidades en mamografías.

Se concluye que los descriptores presentados resultan de interés para la cuantificación de densidad de tejido mamario mediante mamografías, debido a que se calcula en base al examen de control primario de control epidemiológico, y por su potencialidad como indicador de cálculo rápido en eventual combinación con otros descriptores provenientes de otros paradigmas, pero también como posible descriptor individual mediante el análisis de su capacidad de discriminación a través de otras técnicas y clasificadores que no fueron estudiados en este trabajo y que son materia de trabajo futuro inmediato.

Los algoritmos realizados, son un principio de una indeterminada cantidad de variantes para adaptar el concepto de Entropía de Permutación en imágenes. Se esperan mejoras de estos algoritmos en el espacio bidimensional, al mismo tiempo que la creación de otros algoritmos novedosos que guarden las ventajas presentadas por el descriptor en su versión temporal: simple, rápido, robusto e invariante a transformaciones no lineales.

7. Anexo

Algoritmos

Entropía de permutación (EP)

```
function [H] = entroperm( datos,transparencia,embedding
)%datos (rgb2grey,unit8) embedding(integer)

    s = vectorPixel(datos,transparencia);
    d = embedding;
    %Vector v que contiene valores del 0 hasta d-1
    v = factorial(d-1:-1:0);
    %d! permutaciones posibles
    histoperm = zeros(1,factorial(d));

    for i = 1 : (length(s)-d+1)

        r = s(i:d+i-1);
        r2(d) = 0;

        for j = 1 : (d-1)
            a = 0;
            for k = 1 : (d-j)
                if r(j) < r(j+k)
                    a = a + 1;
                end
            end
            r2(j) = a * v(j);
        end

        n = sum(r2);
        %subindice (n+1) para salvar el caso n=0
        histoperm(n+1) = histoperm(n+1) + 1;

    end

    %saco los ceros del histograma, que luego darán NaN en el calculo
del logaritmo
    histoperm(histoperm==0) = [];
    histoperm = histoperm/(length(s)-d+1);
    figure();
    bar(histoperm) ; title('Histograma de patrones de orden')
    %entropia de Shannon
    Hi = - sum(histoperm.*log2(histoperm));
    %factor de normalización
    Hmax = log2(factorial(d));
    H = Hi/Hmax;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Entropía de permutación ponderada (EPP)

```
function [HP] = entropermPond( datos,transparencia,embedding
)%datos (rgb2grey,unit8) embedding(integer)
```

```

s = vectorPixel(datos,transparencia);
d = embedding;
% Vector v que contiene valores desde d-1 hasta 0
v = factorial(d-1:-1:0);
% d! permutaciones posibles
histopermP = zeros(1,factorial(d));
w = zeros(1,length(s)-d+1);

for i = 1 : (length(s)-d+1)

    r = s(i:d+i-1);
% Peso de la diferencia de amplitudes entre los valores del vector
w(i) = (sum((r - mean(r)).^2))/d;
r2(d) = 0;

    for j = 1 : (d-1)
        a = 0;
        for k = 1 : (d-j)
            if r(j) < r(j+k)
                a = a + 1;
            end
        end
        r2(j) = a * v(j);
    end

    n = sum(r2);
% subindice (n+1) para salvar el caso n=0. En la posicion sumo la
% cantidad de peso de ese vector de embedding en vez de sumarle 1
como
% en EP tradicional.
    histopermP(n+1) = histopermP(n+1) + w(i);

end
%saco los ceros del histograma, que luego darán NaN en el calculo
del logaritmo
histopermP(histopermP==0) = [];
histopermP = histopermP / sum(w);
%figure();
%bar(histopermP);title('epp')
% entropia de Shannon
HPi = -sum(histopermP.*log2(histopermP));
% factor de normalización
HPmax = log2(factorial(d));
HP = HPi / HPmax;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Entropía de permutación con umbralado de la imagen (EPT y EPTp)

function [H , HP] = epUmbralTonos( imagen,transparencia,embedding
)%image(rgb2grey,unit8) embedding(integer)

    d = embedding;

```

```

    % vector c que contiene d posibles valores, en vez de contener 255
    posibles valores

```

```

    %%%%%%%%%%%
    c = [];
    index = 0;
    s = vectorPixel(imagen,transparencia);

    for i = 1 : length(s)
        if(s(i) < floor(256/d))
            index = index + 1;
            c(index) = 1;
        end
        if((floor(256/d) <= s(i)) && (s(i) < floor((256/d)*2)))
            index = index+1;
            c(index) = 2;
        end
        if((floor((256/d)*2) <= s(i)) && (s(i) <
floor((256/d)*3)))
            index = index+1;
            c(index) = 3;
        end
        if((floor((256/d)*3) <= s(i)) && (s(i) <
floor((256/d)*4)))
            index = index + 1;
            c(index) = 4;
        end
        if((floor((256/d)*4) <= s(i)) && (s(i) <
floor((256/d)*5)))
            index = index + 1;
            c(index) = 5;
        end
        if((floor((256/d)*5) <= s(i)) && (s(i) <
floor((256/d)*6)))
            index = index + 1;
            c(index) = 6;
        end
        if((floor((256/d)*6) <= s(i)) && (s(i) <
floor((256/d)*7)))
            index = index + 1;
            c(index) = 7;
        end
    end

```

```

    %%%%%%%%%%%

```

```

%% epUmbral usando entropia de permutacion

```

```

v = factorial(d-1:-1:0);
%d! permutaciones posibles
histperm = zeros(1,factorial(d));

for i = 1 : (length(c)-d+1)
    r = c(i:d+i-1);
    r2(d) = 0;

    for j = 1 : (d-1)
        a = 0;
        for k = 1 : (d-j)
            if r(j) < r(j+k)

```

```

        a = a + 1;
    end
end
r2(j) = a * v(j);
end

n = sum(r2);

%subindice (n+1) para salvar el caso n=0
histoperm(n+1) = histoperm(n+1) + 1;
end
%saco los ceros del histograma, que luego darán NaN en el calculo
del logaritmo
histoperm(histoperm==0) = [];
histoperm = histoperm/(length(c)-d+1);
%entropia de Shannon
Hi = -sum(histoperm.*log2(histoperm));
%factor de normalización
Hmax = log2(factorial(d));
H = Hi / Hmax;

%% epUmbral usando entropia de permutacion ponderada

r=[];
r2=[];
histopermP = zeros(1,factorial(d));
w = zeros(1,length(c)-d+1);

for i = 1 : (length(c)-d+1)
    r = c(i:d+i-1);
    r2(d) = 0;

    for j = 1 : (d-1)
        a = 0;
        for k = 1 : (d-j)
            if r(j) < r(j+k)
                a = a + 1;
            end
        end
        r2(j) = a * v(j);
    end

    n = sum(r2);

    % subindice (n+1) para salvar el caso n=0. En la posicion sumo la
    % cantidad de peso de ese vector de embedding en vez de sumarle 1
    como
    % en EP tradicional.
    histopermP(n+1) = histopermP(n+1) + w(i);

end
%saco los ceros del histograma, que luego darán NaN en el calculo
del logaritmo
histopermP(histopermP==0) = [];
histopermP = histopermP/sum(w);
% entropia de Shannon
HPi = -sum(histopermP.*log2(histopermP));
% factor de normalización

```

```

    HPmax = log2(factorial(d));
    HP = HPI/HPmax;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Entropía de permutación umbral (EPU y EPUp)

function [HU,HUP] = epUmbral( datos,transparencia,embedding )%
datos(rgb2grey,unit8) embedding(integer)

    s = vectorPixel(datos,transparencia);
    d = embedding;
    % Vector v que contiene valores del 0 hasta d-1
    v = factorial(d-1:-1:0);
    r=[];
    r2=[];

%% epUmbral usando entropia de permutacion

    % d! permutaciones posibles
    histopermU = zeros(1,factorial(d));
    % El criterio del umbral depende del valor de embedding:
    umbral=256/2d(probando se eligio q sea 2d)
    umbral = floor(256/(2*d));

    for i = 1 : (length(s)-d+1)

        r = s(i:d+i-1);
        r2(d) = 0;

        for j = 1 : (d-1)
            a = 0;
            for k = 1 : (d-j)
                if r(j) < (r(j+k)+umbral)
                    a = a + 1;
                end
            end
            r2(j) = a*v(j);
        end

        n = sum(r2);

        % Subíndice (n+1) para salvar el caso n=0
        histopermU(n+1) = histopermU(n+1) + 1;
    end
    %saco los ceros del histograma, que luego darán NaN en el calculo
del logaritmo
    histopermU(histopermU==0) = [];
    histopermU = histopermU / (length(s)-d+1);
    % entropia de Shannon
    HiU = -sum(histopermU.*log2(histopermU));
    % factor de normalización
    HUmax = log2(factorial(d));
    HU = HiU / HUmax;

```

```

%% epUmbral usando entropia de permutacion ponderada

r=[];
r2=[];
histopermUP = zeros(1,factorial(d));
w = zeros(1,length(s)-d+1);

for i = 1 : (length(s)-d+1)

    r = s(i:d+i-1);
    % Peso de la diferencia de amplitudes entre los valores del vector
    w(i) = (sum((r - mean(r)).^2))/d;
    r2(d) = 0;

    for j = 1 : (d-1)
        a = 0;
        for k = 1 : (d-j)
            if r(j) < (r(j+k)+umbral)
                a = a + 1;
            end
        end
        r2(j) = a*v(j);
    end

    n = sum(r2);
    % subindice (n+1) para salvar el caso n=0. En la posicion sumo la
    % cantidad de peso de ese vector de embedding en vez de sumarle 1
como
    % en EP tradicional.
    histopermUP(n+1) = histopermUP(n+1) + w(i);

end
%saco los ceros del histograma, que luego darán NaN en el calculo
del logaritmo
histopermUP(histopermUP==0) = [];
histopermUP = histopermUP/sum(w);
% entropia de Shannon
HPi = -sum(histopermUP.*log2(histopermUP));
% factor de normalización
HPmax = log2(factorial(d));
HUP = HPi/HPmax;

```

end

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Entropía de permutación de valores medios (EPVM)

```

function [H , HP , hUmbral , hUmbralPond] = epMedia (datos ,
transparencia,embedding)

s = datos;
d = embedding;
matrizVecinos = [];
promedio = zeros(floor(size(s,1)/d),floor(size(s,2)/d));
f = 0;
c = 0;

```

```

%% Creación de nueva matriz de los promedios de la matriz original

for i = 1 : floor(size(s,1)/d)

    f = f + 1;

    for j = 1 : floor(size(s,2)/d)

        %extraigo una submatriz de tamaño dxd de la imagen
original
        matrizVecinos = s(((i-1)*d+1):(i*d), ((j-1)*(d)+1):(j*d));
        % busco si hay transparencias en matrizVecinos en un png
        [hayTransp] = find(transparencia(((i-1)*d+1):(i*d), ((j-
1)*(d)+1):(j*d)) == 0);
        if isempty(hayTransp)
            c = c + 1;
            %calculo el valor medio de los pixeles de submatriz.
El
            %resultado lo guardo en la matriz promedio de un
tamaño d veces menor a la original
            %La posicion del resultado guarda la ubicacion del
pixel central de la submatriz original
            promedio(f,c) = round(mean(mean(matrizVecinos)));
        end
    end
end

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sinTransp = 255*ones(size(promedio,1),size(promedio,2));
H = entroperm( promedio ,sinTransp, d );
HP = entropermPond( promedio ,sinTransp, d );
[hUmbral , hUmbralPond] = epUmbral( promedio ,sinTransp, d );

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Entropía de permutación de matrices de entropías (MEP)

```

function [Hep , HepP , HepU , HPep , HPP , HPU , HUep, HUP , HUU] =
epEspacial( datos , transparencia,embedding )

s = datos;
d = embedding;
matrizEntropy = zeros((size(s,1) - (d - 1)) , (size(s,2) - (d - 1)));
matrizEntropyP = zeros((size(s,1) - (d - 1)) , (size(s,2) - (d - 1)));
matrizEntropyU = zeros((size(s,1) - (d - 1)) , (size(s,2) - (d - 1)));

    for i = 1 : (size(s,1) - (d - 1)) % deajo un borde de d filas para
no tener en cuenta las esquinas.

        if mod(i,2) == 0

            for j = 1 : (size(s,2) - (d - 1)) % deajo un borde de d
columnas por la misma razon.

```

```

        matrizVecinos = s( i:(i+d-1) , j:(j+d-1) );
        sinTransp =
255*ones(size(matrizVecinos,1),size(matrizVecinos,2));

        % busco si hay transparencias en matrizVecinos en un
png. Si
        % no hay transparencia calculo las entropias
        hayTransp = find(transparencia( i:(i+d-1) , j:(j+d-1)
) == 0);
        if isempty(hayTransp)
            matrizEntropy(i,j) =
entroperm(matrizVecinos,sinTransp,3); %Uso embedding de 3 ya que las
matrices matrizVecinos son chicas, como máximo con d=7 matrizVecinos
de 7x7
            matrizEntropyP(i,j) =
entropermPond(matrizVecinos,sinTransp,3);
            matrizEntropyU(i,j) =
epUmbral(matrizVecinos,sinTransp,3);
        end

    end

else

    for j = (size(s,2) - (d - 1)) : -1 : 1% dejo un borde de d
columnas por la misma razon.

        matrizVecinos = s( i:(i+d-1) , j:(j+d-1) );
        sinTransp =
255*ones(size(matrizVecinos,1),size(matrizVecinos,2));
        % busco si hay transparencias en matrizVecinos en un
png. Si
        % no hay transparencia calculo las entropias
        hayTransp = find(transparencia( i:(i+d-1) , j:(j+d-1)
) == 0);
        if isempty(hayTransp)
            matrizEntropy(i,j) =
entroperm(matrizVecinos,sinTransp,3);
            matrizEntropyP(i,j) =
entropermPond(matrizVecinos,sinTransp,3);
            matrizEntropyU(i,j) =
epUmbral(matrizVecinos,sinTransp,3);
        end
    end
end

sinTransp = 255*ones(size(matrizEntropy,1),size(matrizEntropy,2));

Hep = entroperm(matrizEntropy,sinTransp,embedding);
HepP = entropermPond(matrizEntropy,sinTransp,embedding);
HepU = epUmbral(matrizEntropy,sinTransp,embedding);

HPep = entroperm(matrizEntropyP,sinTransp,embedding);
HPP = entropermPond(matrizEntropyP,sinTransp,embedding);
HPU = epUmbral(matrizEntropyP,sinTransp,embedding);

```



```

HUep = entroperm(matrizEntropyU, sinTransp, embedding);
HUP = entropermPond(matrizEntropyU, sinTransp, embedding);
HUU = epUmbral(matrizEntropyU, sinTransp, embedding);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Entropía de permutación con vecindad de Quincunx (EPQ)

function [H,HP,HU] = epQuincunx ( datos,transparencia,embedding )

    s = datos;
    d = embedding-2; % uso d para determinar el tamaño de la matriz de
    vecinos (d*2+1), si embedding=3, la matriz es de dimensión 7x7

    % Vector v que contiene valores del 0 hasta d-1
    v = factorial(embedding-1:-1:0);
    % armo la region de vecinos quincunx de distancia mayor d
    vecinos = strel('diamond',d);
    % quincunx es una matriz cuadrada de tamaño d*2+1, donde la
    region de vecindad de quincunx se determina por los
    valores '1' y la región exterior a la vecindad por '0'
    quincunx = vecinos.getnhood;

    % Cantidad de valores recorridos sin contar los bordes que no se
    analizan
    pixeles = (size(s,1)-(d*2))*(size(s,2)-(d*2));
    cantPixeles = 0;
    w = zeros(1,pixeles);
    umbral = floor(256/(2*d));
    rlong = 3;

    histoperm = zeros(1,factorial(embedding));
    histopermP = zeros(1,factorial(embedding));
    histopermU = zeros(1,factorial(embedding));

    for i = (d+1) : (size(s,1)-d) % dejo un borde de d filas para no
    tener en cuenta las esquinas como pixeles centrales de la vecindad.

        if mod(i,2) == 0

            for j = (d+1) : (size(s,2)-d)

                %% Obtengo el vector de embedding r representativo de la vecindad del
                pixel s(i,j),siendo r=[maximoVecindad minimoVecindad s(i,j)]

                    vecinosQuincunx = s((i-d):(i+d) , (j-d):(j+d));
                    % busco si hay transparencias en un png
                    hayTransp = find(transparencia((i-d):(i+d) , (j-
                    d):(j+d))==0);
                    if isempty(hayTransp)
                        maximo = 0;
                        minimo = 256;
                        cantPixeles = cantPixeles + 1;
                        % búsqueda del maximo y minimo dentro de la
                        vecindad de Quincunx
                        for f = 1 : size(vecinosQuincunx,1)

```

```

        for c = 1 : size(vecinosQuincunx,2)
            if quincunx(f,c)
                % busco maximo dentro de vecindad de
quincunx                if vecinosQuincunx(f,c) > maximo
                            maximo = vecinosQuincunx(f,c);
                end
                % busco minimo dentro de vecindad de
quincunx                if vecinosQuincunx(f,c) < minimo
                            minimo = vecinosQuincunx(f,c);
                end
            end
        end
    end

    end

    end

    r = [minimo s(i,j) maximo];

    % calculo entropias de los elementos representativos en la vecindad de
    quincunx almacenados en r.

    %% entropia de permutacion (entoperm)
    r2(rlong) = 0;

    for l = 1 : (rlong-1)
        a = 0;
        for k = 1 : (rlong-1)
            if r(l) < r(l+k)
                a = a + 1;
            end
        end
        r2(l) = a * v(l);
    end

    n = sum(r2);
    % subindice (n+1) para salvar el caso n=0
    histoperm(n+1) = histoperm(n+1) + 1;

    %% entropia de permutacion Ponderada (entopermPond)
    w(cantPixeles) = (sum((r - mean(r)).^2))/rlong;
    histopermP(n+1) = histopermP(n+1) +

w(cantPixeles);

    %% entropia de permutacion Umbral (epUmbral) debo recalcular
    el valor de r
    r2(rlong) = 0;

```

```

for l = 1 : (rlong-1)
    a = 0;
    for k = 1 : (rlong-1)
        if r(l) < (r(l+k)+umbral)
            a = a + 1;
        end
    end
    r2(l) = a*v(l);
end

n = sum(r2);

% Subindice (n+1) para salvar el caso n=0
histopermU(n+1) = histopermU(n+1) + 1;

end
end

elseif mod(i,2) == 1 % En el caso de que la fila sea
par, recorro las columnas de atras para adelante (coherencia con
vecindad)

for j = (size(s,2)-d) : -1 : (d+1) % empiezo de atras para
adelante si la fila es par.

    %% Obtengo el vector de embedding r representativo de la vecindad
del pixel s(i,j),
    %% siendo r=[maximoVecindad minimoVecindad s(i,j)]

    vecinosQuincunx = s((i-d):(i+d) , (j-d):(j+d));
    % busco si hay transparencias en vecinosQuincunx en un
png
[hayTransp] = find(transparencia((i-d):(i+d) , (j-
d):(j+d))==0);
if isempty(hayTransp)
    maximo = 0;
    minimo = 256;
    cantPixeles = cantPixeles + 1;
    % busqueda del maximo y minimo dentro de la
vecindad de Quincunx
for f = 1 : size(vecinosQuincunx,1)

        for c = 1 : size(vecinosQuincunx,2)

            if quincunx(f,c)

                % busco maximo dentro de vecindad de
quincunx
                if vecinosQuincunx(f,c) > maximo

                    maximo = vecinosQuincunx(f,c);

                end

                % busco minimo dentro de vecindad de
quincunx
                if vecinosQuincunx(f,c) < minimo

                    minimo = vecinosQuincunx(f,c);

```

```

                                end

                                end

                                end

                                end

                                r = [minimo s(i,j) maximo];

% calculo entropias de los elementos representativos en la vecindad de
quincunx almacenados en r.

%% Entropia de permutacion (entroperm)
r2(rlong) = 0;

for l = 1 : (rlong-1)
    a = 0;
    for k = 1 : (rlong-1)
        if r(l) < r(l+k)
            a = a + 1;
        end
    end
    r2(l) = a * v(l);
end

n = sum(r2);
%subindice (n+1) para salvar el caso n=0
histoperm(n+1) = histoperm(n+1) + 1;

%% Entropia de permutacion Ponderada (entropermPond)
w(cantPixeles) = (sum((r - mean(r)).^2))/rlong;
histopermP(n+1)= histopermP(n+1) + w(cantPixeles);

%% Entropia de permutacion Umbral (epUmbral) debo recalcular el valor
de r

r2(rlong) = 0;

for l = 1 : (rlong-1)
    a = 0;
    for k = 1 : (rlong-1)
        if r(l) < (r(l+k)+umbral)
            a = a + 1;
        end
    end
    r2(l) = a*v(l);
end

n = sum(r2);

% Subindice (n+1) para salvar el caso n=0
histopermU(n+1) = histopermU(n+1) + 1;

                                end
                                end
                                end
end

```

```

%% Normalizacion entropia de permutacion
    %saco los ceros del histograma, que luego darán NaN en el calculo
del logaritmo
    histoperm(histoperm==0) = [];
    histoperm = histoperm/cantPixeles;
    %figure();
    %bar(histoperm) ; title('ep')
    %entropia de Shannon
    Hi = - sum(histoperm.*log2(histoperm));
    %factor de normalización
    Hmax = log2(4); %en este caso solo hay 4 posibles combinaciones el
vector r
    H = Hi/Hmax;

%% Normalizacion entropia de permutacion ponderada
    %saco los ceros del histograma, que luego darán NaN en el calculo
del logaritmo
    histopermP(histopermP==0) = [];
    histopermP = histopermP/sum(w);
    %figure();
    %bar(histopermP);title('epp')
    % entropia de Shannon
    HPi = -sum(histopermP.*log2(histopermP));
    % factor de normalización
    HPmax = log2(4); % en este caso solo hay 4 posibles combinaciones
del vector r
    HP = HPi/HPmax;

%% Normalizacion entropia de permuracion umbral
    %saco los ceros del histograma, que luego darán NaN en el calculo
del logaritmo
    histopermU(histopermU==0) = [];
    histopermU = histopermU / cantPixeles;
    % entropia de Shannon
    HiU = -sum(histopermU.*log2(histopermU));
    % factor de normalización
    HUmax = log2(4); % en este caso solo hay 4 posibles combinaciones
del vector r
    HU = HiU / HUmax;

end

```

8. Bibliografía

- Acharya, U.R. et al., 2016. Automated detection and localization of myocardial infarction using electrocardiogram: A comparative study of different leads. *Knowledge-Based Systems*, 99, pp.146–156. Available at: <http://dx.doi.org/10.1016/j.knosys.2016.01.040>.
- Aibar, L. et al., 2011. Clasificación radiológica y manejo de las lesiones mamarias. *Clinica e Investigacion en Ginecologia y Obstetricia*, 38(4), pp.141–149.
- Alonso, Á.R., 2011. *Comportamiento y análisis de descriptores de texturas en imágenes modis*. Facultad de informática, universidad complutense de madrid.
- Amezquita-Sanchez, J.P., Adeli, A. & Adeli, H., 2016. A new methodology for automated diagnosis of mild cognitive impairment (MCI) using magnetoencephalography (MEG). *Behavioural Brain Research*, 305(Mci), pp.174–180. Available at: <http://dx.doi.org/10.1016/j.bbr.2016.02.035>.
- Anon, Ministerio de Salud Presidencia de la Nación-Cáncer de mama. Available at: <http://www.msal.gob.ar/index.php/component/content/article/48/246-cancer-de-mama#panel-7> [Accessed June 1, 2017].
- Antoni, S. et al., 2016. An assessment of GLOBOCAN methods for deriving national estimates of cancer incidence. *Bulletin of the World Health Organization*, 94(3), pp.174–184. Available at: <http://www.who.int/entity/bulletin/volumes/94/3/15-164384.pdf> [Accessed June 11, 2017].
- Azami, H. & Escudero, J., 2016. Improved multiscale permutation entropy for biomedical signal analysis: Interpretation and application to electroencephalogram recordings. *Biomedical Signal Processing and Control*.
- Bandt, C. & Pompe, B., 2002. Permutation entropy: a natural complexity measure for time series. *Physical review letters*, 88(17), p.174102. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/12005759> [Accessed September 7, 2016].
- Blotta, E. et al., 2011. Evaluation of speckle-interferometry descriptors to measuring drying-of-coatings. *Signal Processing*.
- Boyd, N.F. et al., 2007. Mammographic Density and the Risk and Detection of Breast Cancer. *New England Journal of Medicine*, 356(3), pp.227–236. Available at: <http://www.nejm.org/doi/abs/10.1056/NEJMoa062790>.
- Chen, X. et al., 2015. The experimental signals analysis for bubbly oil-in-water flow using multi-scale weighted-permutation entropy. *Physica A: Statistical Mechanics and its Applications*, 417, pp.230–244. Available at: <http://dx.doi.org/10.1016/j.physa.2014.09.058>.
- Cova, W.J.D. & Cavallero, R.A., 2006. Sobre Wavelets e Imágenes. *utn*.
- Fadlallah, B. et al., 2013. Weighted-permutation entropy: A complexity measure for time series incorporating amplitude information. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 87(2), pp.1–7.
- Feldman, D.P. & Crutchfield, J.P., 2003. Structural information in two-dimensional patterns: Entropy convergence and excess entropy. *Physical Review E*, 67(5), p.51104. Available at: <http://arxiv.org/abs/cond-mat/0212078> <http://dx.doi.org/10.1103/PhysRevE.67.051104> <http://link.aps.org/doi/>

10.1103/PhysRevE.67.051104.

- Ferlazzo, E. et al., 2014. Permutation entropy of scalp EEG: A tool to investigate epilepsies. Suggestions from absence epilepsies. *Clinical Neurophysiology*, 125(1), pp.13–20. Available at: <http://dx.doi.org/10.1016/j.clinph.2013.06.023>.
- Fernández Sarría, A., 2007. Estudio de técnicas basadas en la transformada wavelet y optimización.
- Frank, B. et al., 2006. Permutation entropy improves fetal behavioural state classification based on heart rate analysis from biomagnetic recordings in near term fetuses. *Medical and Biological Engineering and Computing*, 44(3), pp.179–187.
- Gil Rodríguez, J.L., 2008. Estado Actual de la Representación y Análisis de Textura en Imágenes . *CENATAV-Serie Azul*, RNPS No. 0(21812).
- Heath, M. et al., 2001. The digital database for screening mammography. *Proceedings of the Fifth International Workshop on Digital Mammography*, pp.212–218. Available at: http://www.nd.edu/~kwb/HeathEtAlIWDM_2000.pdf \npapers2://publication/uuid/37C3C3F3-5994-4145-AB4B-5B1CC5381282.
- Li, D. et al., 2010. Multiscale permutation entropy analysis of EEG recordings during sevoflurane anesthesia. *Journal of neural engineering*, 7(4), p.46010.
- Li, T. & Wen, P., 2016. Depth of anaesthesia assessment based on adult electroencephalograph beta frequency band. *Australasian Physical & Engineering Sciences in Medicine*, 39(3), pp.773–781. Available at: <http://link.springer.com/10.1007/s13246-016-0459-5>.
- Li, Y., Qian, C. & Fan, Y., 2006. Unsupervised texture segmentation using permutation entropy and grey-level feature. *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, 2, pp.9845–9848.
- Liang, Z. et al., 2015. EEG entropy measures in anesthesia. *Frontiers in computational neuroscience*, 9(February), p.16. Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4332344&tool=pmcentrez&rendertype=abstract>.
- Lopez-Ruiz, R., Mancini, H. & Calbet, X., 1995. A Statistical Measure of Complexity. , pp.1–8. Available at: <http://arxiv.org/abs/1009.1498>.
- Marfil, R., Jim, F. & Sandoval, A.B.F., 2004. Analisis de imagenes basado en textura mediante transformada wavelet. , (1), pp.1–4.
- Mateos, D.M., 2016. *Medidas de complejidad y de información como herramientas para el análisis de series temporales*. Facultad de Matemática Astronomía y Física. Universidad Nacional de Córdoba.
- Melchor, M.A., 2016. Análisis y clasificación de imágenes repetitivas mediante técnicas de simetría computacional Tesis doctoral.
- Mor, M. & Fraenkel, A.S., 1984. Cayley permutations. *Discrete Mathematics*, 48(1), pp.101–112.
- Morabito, F.C. et al., 2012. Multivariate multi-scale permutation entropy for complexity analysis of Alzheimer's disease EEG. *Entropy*, 14(7), pp.1186–1202.
- Nicolaou, N. & Georgiou, J., 2010. Permutation entropy: A new feature for brain-computer interfaces. *2010 IEEE Biomedical Circuits and Systems Conference, BioCAS 2010*, pp.49–

- Olofsen, E., Sleight, J.W. & Dahan, A., 2008. Permutation entropy of the electroencephalogram: A measure of anaesthetic drug effect. *British Journal of Anaesthesia*, 101(6), pp.810–821.
- Parlitz, U. et al., 2012. Classifying cardiac biosignals using ordinal pattern statistics and symbolic dynamics. *Computers in Biology and Medicine*, 42(3), pp.319–327. Available at: <http://dx.doi.org/10.1016/j.combiomed.2011.03.017>.
- Passoni, L.I., 2005. Modelos en Bioingeniería: Caracterización de imágenes estáticas y dinámicas. , p.123.
- Paulina Neira, V., 2013. Densidad mamaria y riesgo de cáncer mamario. *Revista Médica Clínica Las Condes*, 24(1), pp.122–130. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S0716864013701378>.
- Picó, A.M., Roldan, E.C. & Frau, D.C., 2014. Complejidad Lempel-Ziv : robustez frente a spikes y pérdida de muestras en registros EEG. , (5), pp.2–5.
- Pinto, A., 2006. *Segmentación de imágenes por textura*. Facultad de Ingeniería, Universidad de Concepción, Santiago de Chile.
- Platero Dieñas, C., 2009. Interpretación automática de las imágenes. *Capítulo 7: Interpretación de las imágenes*, pp.199–221.
- Qu, J. et al., 2015. State recognition of viscoelastic sandwich structures based on permutation entropy and generalized Chebyshev support vector machine. *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, 2015–July, pp.957–962.
- Ravelo, A.G. et al., 2015. Application of the permutation entropy over the heart rate variability for the improvement of electrocardiogram-based sleep breathing pause detection. *Entropy*, 17(3), pp.914–927.
- Ribeiro, H. V et al., 2012. Complexity-entropy causality plane as a complexity measure for two-dimensional patterns. *PLoS ONE*.
- Rosso, O.A. & Larrondo, H.A., 2007. Distinguishing noise from chaos. *Physical Review Letters*, 99(15), pp.1–4.
- Ruiz, M.D.C., Guillamón, A. & Gabaldón, A., 2012. A new approach to measure volatility in energy markets. *Entropy*, 14(1), pp.74–91.
- Ruiz, O.M. et al., 2014. Valoración de las categorías BI-RADS® de la densidad mamaria entre radiólogos y respecto a un método cuantitativo de medición. *Revista de Senología y Patología Mamaria*, 27(4), pp.157–162.
- Ruj, E., 2016. *BI-RADS®. Sistema de informes y registro de datos de estudios por imágenes de la mama* 5ta ed. A. C. of Radiology & J. S.A, eds.,
- Shi, Z., Song, W. & Taheri, S., 2016. Improved LMD, permutation entropy and optimized K-means to fault diagnosis for roller bearings. *Entropy*, 18(3).
- Tawfik, N.S., Youssef, S.M. & Kholief, M., 2016. A hybrid automated detection of epileptic seizures in EEG records. *Computers and Electrical Engineering*, 53, pp.177–190. Available at: <http://dx.doi.org/10.1016/j.compeleceng.2015.09.001>.
- Tian, Y., Wang, Z. & Lu, C., 2015. Self-adaptive bearing fault diagnosis based on permutation entropy and manifold-based dynamic time warping. *Mechanical Systems and Signal Processing*. Available at: <http://dx.doi.org/10.1016/j.ymssp.2016.04.028>.

- Veisi, I., Pariz, N. & Karimpour, A., 2007. Fast and robust detection of epilepsy in noisy EEG signals using permutation entropy. ... *and Bioengineering*, 2007. *BIBE ...*, pp.2007–2010. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4375565.
- Viniegra, M., Paolino, M. & Arrossi, S., 2010. *Cáncer de mama en Argentina: organización, cobertura y calidad de las acciones de prevención y control*,
- Vuong, P.L., Malik, A.S. & Bornot, J., 2015. Weighted-permutation entropy as complexity measure for electroencephalographic time series of different physiological states. *IECBES 2014, Conference Proceedings - 2014 IEEE Conference on Biomedical Engineering and Sciences: "Miri, Where Engineering in Medicine and Biology and Humanity Meet"*, (December), pp.979–984.
- Yaghjian, L. et al., 2015. Mammographic breast density and breast cancer risk: interactions of percent density, absolute dense, and non-dense areas with breast cancer risk factors. *Breast Cancer Research and Treatment*, 150(1), pp.181–189.
- Zanin, M. et al., 2012. Permutation entropy and its main biomedical and econophysics applications: A review. *Entropy*, 14(8), pp.1553–1577.
- Zhang, X. et al., 2015. A novel bearing fault diagnosis model integrated permutation entropy, ensemble empirical mode decomposition and optimized SVM. *Measurement: Journal of the International Measurement Confederation*, 69, pp.164–179. Available at: <http://dx.doi.org/10.1016/j.measurement.2015.03.017>.
- Zhuang, L.X. et al., 2016. Nonlinear multi-scale dynamic stability of oil-gas-water three-phase flow in vertical upward pipe. *Chemical Engineering Journal*, 302, pp.595–608.
- Zunino, L. et al., 2009. Forbidden patterns, permutation entropy and stock market inefficiency. *Physica A: Statistical Mechanics and its Applications*, 388(14), pp.2854–2864. Available at: <http://dx.doi.org/10.1016/j.physa.2009.03.042>.
- Zunino, L. et al., 2008. Fractional Brownian motion, fractional Gaussian noise, and Tsallis permutation entropy. *Physica A: Statistical Mechanics and its Applications*, 387(24), pp.6057–6068.
- Zunino, L. et al., 2012. On the efficiency of sovereign bond markets. *Physica A: Statistical Mechanics and its Applications*, 391(18), pp.4342–4349. Available at: <http://dx.doi.org/10.1016/j.physa.2012.04.009>.
- Zunino, L. & et al., 2016. Monitoring the informational efficiency of European corporate bond markets with dynamical permutation min-entropy. *Physica A: Statistical Mechanics and its Applications*, 456, pp.1–9. Available at: <http://dx.doi.org/10.1016/j.physa.2016.03.007>.
- Zunino, L. & Ribeiro, H. V., 2016. Discriminating image textures with the multiscale two-dimensional complexity-entropy causality plane.