



**EDUCACIÓN**

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

# Tecnológico Nacional de México

Centro Nacional de Investigación  
y Desarrollo Tecnológico

## Tesis de Maestría

Diseño e Implementación de un Modelo  
Autoadaptable de Segmentación de Imágenes  
Digitales por Discontinuidades

presentada por

**Ing. José Manuel Pérez Tovar**

como requisito para la obtención del grado de  
**Maestro en Ciencias de la Computación**

Director de tesis  
**Dr. Raúl Pinto Elías**

Cuernavaca, Morelos, México. Julio de 2020.

# Aceptación del documento de tesis



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



Centro Nacional de Investigación y Desarrollo Tecnológico  
Departamento de Ciencias Computacionales

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

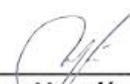
Cuernavaca, Mor., 13/julio/2020

OFICIO No. DCC/103/2020  
Asunto: Aceptación de documento de tesis  
CENIDET-AC-004-M14-OFCIO

**C. DR. GERARDO VICENTE GUERRERO RAMÍREZ**  
**SUBDIRECTOR ACADÉMICO**  
**PRESENTE**

Por este conducto, los integrantes de Comité Tutorial del **C. Ing. José Manuel Pérez Tovar**, con número de control M18CE011, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado **"Diseño e implementación de un modelo auto-adaptable de segmentación de imágenes digitales por discontinuidades"** y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

  
\_\_\_\_\_  
**Dr. Paul Pinto Elías**  
**Doctor en Ciencias en la Especialidad de**  
**Ingeniería Eléctrica**  
**3890453**  
**Director de tesis**

  
\_\_\_\_\_  
**Dra. Andrea Magadán Salazar**  
**Doctora en Ciencias Computacionales**  
**10654097**  
**Revisor 1**

  
\_\_\_\_\_  
**Dr. Dante Mújica Vargas**  
**Doctor en Comunicaciones y Electrónica**  
**09131756**  
**Revisor 2**

C.c.p. Depto. Servicios Escolares  
Expediente / Estudiante  
JGGS/mz



Interior Internado Palmira S/N, Col. Palmira, C. P. 62490Cuernavaca, Morelos.  
Tel. (01) 777 3 62 77 70, ext. 3202, e-mail: dcc@cenidet.edu.mx  
www.tecnm.mx | www.cenidet.tecnm.mx



# Autorización de impresión de tesis



Centro Nacional de Investigación y Desarrollo Tecnológico

"2020. Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Morelos 16/julio/2020

OFICIO No. SAC/ 221/2020

Asunto: Autorización de impresión de tesis

**JOSÉ MANUEL PÉREZ TOVAR**  
**CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS**  
**DE LA COMPUTACIÓN**  
**PRESENTE**

Por este conducto tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado *"Diseño e implementación de un modelo auto-adaptable de segmentación de imágenes digitales por discontinuidades"*, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

**ATENTAMENTE**

*Excelencia en Educación Tecnológica*  
*"Conocimiento y tecnología al servicio de México"*



**CENTRO NACIONAL  
 DE INVESTIGACIÓN  
 Y DESARROLLO  
 TECNOLÓGICO  
 SUBDIRECCIÓN  
 ACADÉMICA**

**DR. GERARDO VICENTE GUERRERO RAMÍREZ**  
**SUBDIRECTOR ACADÉMICO**

C.c.p. M.E. Guadalupe Carrido Rivera, Jefa del Departamento de Servicios Escolares  
 Expediente  
 CVGR/CHG



Interior Internado Palmira S/N. Col. Palmira, C. P. 62490, Cuernavaca, Morelos.  
 Tel. (01) 777 3 62 77 70, ext. 4106, e-mail: dir\_cenidet@tecnm.mx

www.tecnm.mx | www.cenidet.edu.mx



## Agradecimientos

Un agradecimiento muy especial al Dr. Raúl Pinto Elías, director de esta tesis, por su orientación, su confianza, su paciencia, sus consejos y soporte diligente en cada una de las asesorías que dieron dirección y forma al desarrollo de este trabajo.

A mis revisores, la Dra. Andrea Magadán Salazar por sus consejos y amables sugerencias durante el desarrollo de la tesis. Al Dr. Dante Mújica Vargas por su soporte, sus recomendaciones y la agudeza de sus observaciones. Al Dr. Manuel Mejía Lavallo por sus observaciones, comentarios y sugerencias en los temas de métricas y descriptores MPEG-7.

Al CONACYT por la beca otorgada y al TECNM por facilitar sus instalaciones para la realización de mis estudios.

Al CENIDET por la oportunidad de integrarme al programa de maestría en ciencias computacionales y permitirme lograr una meta postergada por más de dos décadas. La mejor parte de asistir al CENIDET no fueron solamente las clases que se tomaban o la investigación que se realizaba, sino aspectos adicionales del entorno: convivir con los compañeros y profesores, ir a seminarios, conocer gente, unirse a organizaciones y asistir a clases y talleres para aprender lo que no se sabía.

A mis compañeros de generación: Arthur, Oliver y Tonny por su apoyo y sobre todo por los agradables e inolvidables momentos en la convivencia cotidiana durante el tiempo que permanecemos juntos en el programa de maestría.

Y finalmente, a todo el grupo de ingenieros por su respaldo y soporte incondicional.

¡Gracias, totales!

# Dedicatorias

Esta tesis está dedicada a todos aquellos que siempre han confiado en mí, que me han dado palabras de aliento y consejo cuando han sido necesarias, y que han compartido mis alegrías como propias.

## Resumen

En el presente trabajo de investigación se propuso el diseño y la implementación de un modelo autoadaptable de segmentación de imágenes digitales por discontinuidades.

La segmentación es un tema complejo y abierto a la investigación, su objetivo es subdividir una imagen en sus partes u objetos constituyentes de interés para alguna aplicación particular. En un sistema clásico de visión artificial, la segmentación es una etapa medular que definirá el éxito o fracaso de las etapas de análisis posteriores; estos sistemas generalmente se diseñan para un determinado tipo de imágenes y son de configuración fija; el modelo propuesto buscaba proporcionar cierto grado de flexibilidad y adaptabilidad a la segmentación, con el objeto de segmentar distintos tipos de imágenes.

Para lograr tal objetivo, las imágenes se caracterizaron a través de la extracción de sus *descriptores visuales* de color, textura y forma del estándar MPEG-7 (se les denominará *descriptores visuales MPEG-7* o simplemente *descriptores visuales* en el resto de este documento), de tal manera que mediante esta información fuera posible su identificación y reconocimiento. Los *descriptores visuales MPEG-7* generalmente se utilizan en sistemas de recuperación de imágenes con base en contenido CBIR (*Content-Based Image Retrieval*). En este caso, a través de una combinación de los *descriptores visuales*, se creó un *vector de características* como identificador de cada imagen en un conjunto predefinido.

El modelo funciona aplicando una serie de combinaciones de técnicas de preprocesamiento y de segmentación a un conjunto de imágenes, el resultado obtenido es comparado con su imagen de bordes de referencia (*ground truth*), si el grado de similitud entre ellos es igual o mayor a un valor o umbral mínimo predefinido, entonces la secuencia de las técnicas aplicadas se considera exitosa y se utiliza para “etiquetar” al *vector de características* de la imagen procesada.

Los *descriptores visuales MPEG-7* asociados con las secuencias de técnicas aplicadas exitosamente, se emplean para entrenar algoritmos de aprendizaje automático supervisado y crear modelos de predicción. Estos modelos reciben los *descriptores visuales* de una nueva imagen de entrada y generan una recomendación de la secuencia de técnicas de preprocesamiento y segmentación que deberán ser aplicadas para segmentar la nueva imagen.

Se realizó una implementación del modelo y los resultados obtenidos en la experimentación mostraron que es posible utilizar descriptores de imagen MPEG-7 y las técnicas de aprendizaje automático para generar modelos de clasificación que permitan predecir o recomendar las técnicas de segmentación adecuadas para procesar nuevas imágenes.

# Abstract

This project presents the design and implementation of a digital image segmentation model based on discontinuities.

Segmentation is a complex subject and remain open to research, its objective is to subdivide an image into its constituent parts or objects of interest for any particular application. In a classic machine vision system, segmentation is a core stage, that will define the success or failure of subsequent analysis stages. These systems are normally designed for a certain type of images and are of fixed configuration; The proposed model sought to provide a certain degree of flexibility and adaptability to segmentation, in order to segment different types of images.

To achieve this goal, the images were characterized through the extraction of their visual descriptors of color, texture and shape from the MPEG-7 standard (they will be called MPEG-7 visual descriptors or simply visual descriptors in the rest of this document), in such a way that through this information its identification and recognition were possible. *MPEG-7 visual descriptors* are generally used in content-based image retrieval (CBIR) image retrieval systems. In this case, through a combination of the visual descriptors, a vector of characteristics was created as the identifier of each image in a predefined set.

A combination series of preprocessing and segmentation techniques are applied to a group of images by the model, then, every processed image is compared with its own *ground truth*, if the degree of similarity between them is equal to or greater than a predefined value, then, the combination number of the applied techniques is used to "label" the vector of characteristics of the processed image.

Descriptors tagged with technique combinations are used to train supervised machine learning algorithms to create prediction models, which, by applying the descriptors of a new input image, generate a recommendation to combining appropriate preprocessing and segmentation techniques for processing.

An implementation of the model was carried out and the results obtained from the experimentation showed that it is possible to use image descriptors and machine learning techniques to generate classification models that allow recommending the appropriate segmentation techniques to process new images.

# Tabla de contenido

<b>Aceptación del documento de tesis .....</b>	<b>I</b>
<b>Autorización de impresión de tesis .....</b>	<b>II</b>
<b>Carta de cesión de derechos .....</b>	<b>III</b>
<b>Agradecimientos.....</b>	<b>IV</b>
<b>Dedicatorias .....</b>	<b>V</b>
<b>Resumen .....</b>	<b>VI</b>
<b>Abstract.....</b>	<b>VII</b>
<b>Tabla de contenido .....</b>	<b>VIII</b>
<b>Índice de tablas.....</b>	<b>X</b>
<b>Índice de figuras .....</b>	<b>XII</b>
<b>Glosario .....</b>	<b>XV</b>
<b>Acrónimos.....</b>	<b>XVIII</b>
<b>Notación .....</b>	<b>XIX</b>
<b>Organización de la tesis.....</b>	<b>XXI</b>
<b>Introducción.....</b>	<b>22</b>
1.1 Motivación .....	22
1.2 Segmentación por discontinuidades .....	23
1.3 Planteamiento del problema .....	27
1.4 Preguntas de investigación.....	28
1.5 Objetivos .....	29
1.5.1 <i>Objetivo general</i> .....	29
1.5.2 <i>Objetivos específicos</i> .....	29
1.6 Justificación y beneficios .....	30
1.7 Alcances y limitaciones.....	30
1.8 Marco conceptual .....	31
1.8.1 <i>Preprocesamiento de la imagen</i> .....	31
1.8.2 <i>Segmentación por discontinuidades</i> .....	33
1.8.3 <i>Descriptor</i> .....	36
1.8.4 <i>Métricas</i> .....	49
1.8.5 <i>Aprendizaje automático</i> .....	51

---

<b>Revisión de la literatura .....</b>	<b>58</b>
2.1 Antecedentes.....	58
2.2 Estado del arte .....	65
2.2.1 Segmentación adaptativa.....	65
2.2.2 Preprocesamiento y detección de bordes .....	69
2.2.3 Descriptores.....	71
2.2.4 Métricas .....	75
2.2.5 Discusión del estado del arte .....	82
2.3 Propuesta de solución .....	87
2.4 Método de solución.....	88
<b>Diseño e implementación del modelo .....</b>	<b>89</b>
3.1 Modelado de la solución .....	89
3.1.1 Modelo de aprendizaje.....	90
3.1.2 Modelo de utilización.....	96
3.2 Implementación del modelo .....	97
3.2.1 Técnicas de preprocesamiento y detección de bordes.....	99
3.2.2 Descriptores.....	99
3.2.3 Métricas de similitud.....	101
3.2.4 Técnicas de aprendizaje supervisado .....	105
3.2.5 Calibración del modelo implementado.....	106
3.2.6 Conclusiones .....	111
<b>Experimentación y resultados .....</b>	<b>115</b>
4.1 Detalles y objetivos de la experimentación .....	115
4.2 Experimentación .....	115
4.3 Discusión.....	153
<b>Conclusiones y trabajo futuro .....</b>	<b>157</b>
5.1 Objetivos logrados .....	157
5.2 Productos y aportaciones .....	159
5.3 Conclusiones generales.....	161
5.4 Trabajos futuros.....	162
<b>Referencias .....</b>	<b>164</b>
<b>ANEXOS .....</b>	<b>169</b>
<b>Anexo A.....</b>	<b>169</b>
<b>Anexo B.....</b>	<b>172</b>
<b>Anexo C.....</b>	<b>177</b>

# Índice de tablas

Tabla 1.1 Resultados obtenidos en tres diferentes tratamientos de imágenes. ....	31
Tabla 1.2 Descriptor EHD de 80 componentes ( <b>80-tupla</b> ).....	38
Tabla 1.3 Semántica de los contenedores de histograma de borde local EHD.....	44
Tabla 3.1 Técnicas de preprocesamiento y segmentación implementadas en el sistema. ....	99
Tabla 3.2 Fórmulas de cálculo de distancias de los diferentes descriptores MPEG-7. ....	102
Tabla 3.3 Matriz de confusión, coincidencia de píxeles entre la imagen segmentada y su <i>ground truth</i> .....	104
Tabla 3.4 Longitudes de los descriptores MPEG-7 evaluados.....	112
Tabla 4.1 Combinación de técnicas representadas por su número de clase. ....	117
Tabla 4.2 Experimentos realizados para evaluar el modelo implementado.....	118
Tabla 4.3 Comparativa del desempeño de modelo respecto a cantidad de técnicas aplicadas con un umbral de la métrica = 0.8. ....	121
Tabla 4.4 Clase predominante y su secuencia de técnicas equivalente de cada apartado correspondiente con un umbral de métrica CFHu = 0.8. ....	121
Tabla 4.5 Comparativo de clasificaciones bajo las métricas CFHU y <i>accuracy</i> . ....	124
Tabla 4.6 Promedio de las métricas evaluadas sobre el conjunto Corel200. ....	124
Tabla 4.7 Comparativo de desempeño del modelo respecto al uso o no de la base de conocimiento del sistema. ....	126
Tabla 4.8 Comparativa de desempeño al variar el valor del umbral de la métrica <i>accuracy</i> . ....	128
Tabla 4.9 Comparativa de las clasificaciones del subconjunto de prueba obtenidas con cuatro clasificadores.....	132
Tabla 4.10 Comparación entre las segmentaciones producidas con técnicas recomendadas por los clasificadores KNN y SMV con <i>kernel</i> lineal, polinomial y de base radial para la imagen 11.png. ....	134
Tabla 4.11 Recomendación de técnicas de segmentación para la imagen 11.png con sus respectivos valores de la métrica de similitud. ....	135
Tabla 4.12 Secuencias de segmentación para otras imágenes procesadas.....	135
Tabla 4.13 Superposición de la segmentación sobre la imagen original de entrada al modelo. ....	136

---

Tabla 4.14 Secuencia de técnicas recomendadas por los clasificadores sobre las imágenes de prueba.....	138
Tabla 4.15 Segmentaciones producidas con las técnicas recomendadas por los clasificadores KNN y SMV con <i>kernel</i> lineal, polinomial y de base radial sobre la imagen 00311.png.....	139
Tabla 4.16 Segmentaciones producidas con las mismas técnicas recomendadas por los clasificadores KNN y SMV con <i>kernel</i> polinomial y de base radial sobre las imágenes de prueba.....	140
Tabla 4.17 Valores promedio de las características de las imágenes obtenidas en el entrenamiento con el conjunto BSD300. ....	142
Tabla 4.18 Imágenes 225.png y 229.png con valores métricos extremos, imágenes 150.png y 7.png en la frontera de transición del valor mínimo de la métrica (0.8). ....	148
Tabla 4.19 Comparativa de características entre las imágenes 225.png, 150.png, 7.png, 229.png.....	152
Tabla 4.20 Respuestas a las preguntas de investigación planteadas inicialmente. ....	154
Tabla 5.1 Objetivos logrados.....	157

# Índice de figuras

Figura 1.1 Sistema de visión artificial clásico (Moeslund, 2012).....	22
Figura 1.2 Segmentación por discontinuidades, a) imagen original, b) imagen segmentada con bordes o fronteras detectadas. Imágenes tomadas del conjunto BSD300 (Martin, Fowlkes, Tal, & Malik, 2001).....	23
Figura 1.3 <i>Pipelines</i> de segmentación para tres distintas clases de imágenes. ....	24
Figura 1.4 Problema a resolver. ....	27
Figura 1.5 Aplicación de varios filtros de suavizado lineales y no lineales sobre una imagen a) Imagen original b) Filtro de media 5x5 c) Filtro Gaussiano ( $\sigma = 2.5$ ) d) Filtro de mediana 5x5 (Young, Gerbrands, & J. van Vliet, 2007). ....	32
Figura 1.6 Aplicación de técnicas de realzado a) Imagen original b) Histograma extendido c) Histograma ecualizado (Young, Gerbrands, & J. van Vliet, 2007).....	32
Figura 1.7 Subprocesos de la segmentación. ....	33
Figura 1.8 Ejemplos de los tres operadores a) Imagen original b) Gradiente c) Laplaciano d) Gradiente morfológico (Young, Gerbrands, & J. van Vliet, 2007).....	35
Figura 1.9 Bordes detectados y etiquetados. Imagen tomada de (Mallick, 2015). ....	35
Figura 1.10 Detección de características sobresalientes en una imagen. ....	37
Figura 1.11 Conjunto de <i>descriptores visuales</i> MPEG7. ....	39
Figura 1.12 Canales utilizados en el cálculo de HTD (Choi, Won, Ro, & Manjunath, 2003). ....	42
Figura 1.13 EHD considera 5 tipos de bordes (Won, Park, & Jeon, 2002). ....	43
Figura 1.14 Definición de sub-imagen y bloque-imagen.....	44
Figura 1.15 Comparación de similitud por traslape de imagen segmentada sobre el <i>ground truth</i> (Yeghiazaryana & Voiculescu, 2019).....	49
Figura 1.16 Cálculo de la distancia Euclídea <i>dij</i> . ....	51
Figura 1.17 El aprendizaje como proceso de convertir la experiencia en conocimiento. ....	52
Figura 1.18 La hipótesis <i>y</i> separa los datos etiquetados como positivos (+) y negativos (–). ....	53
Figura 1.19 a) Conjunto de datos etiquetados con tres clases b) Frontera de separación de las clases después del aprendizaje (Chao, 2011).....	55
Figura 1.20 a) Conjunto de muestras sin etiquetas. (b) Después de aplicar el algoritmo de agrupamiento, se descubren tres grupos, delimitados por los contornos punteados (Chao, 2011).....	56

Figura 1.21 a) Conjunto de datos etiquetados (círculos, estrellas y cruces) y no etiquetados (rombos), b) Clasificación después del proceso de aprendizaje (Chao, 2011). .....	57
Figura 2.1 Modelo conceptual de la arquitectura propuesta (Cervantes, 2006). .....	61
Figura 2.2 Diagrama de flujo del algoritmo propuesto (Meiling Gong, 2017). .....	69
Figura 2.3 Características locales ilustrativas de la imagen (a) imagen de entrada, (b) esquinas, (c) bordes y (d) regiones (Salahat & Qasaimeh, 2017). .....	72
Figura 2.4 CLD: detección de colores representativos en una imagen particionada en bloques de 8x8. a) Imagen original b) Imagen particionada por colores (Rayar, 2017). .....	74
Figura 2.5 EHD: Cálculo de histograma para cada celda de 4x4 en la imagen, a) 5 tipos de bordes considerados b) Histograma de bordes de la celda de imagen (Won, Park, & Jeon, 2002) .....	74
Figura 2.6 Clasificación de los métodos de discrepancia (Surya & Satheesh, 2016). .....	79
Figura 2.7 Diferentes métodos de medidas de bondad (Surya & Satheesh, 2016). .....	80
Figura 2.8 Propuesta de solución. ....	87
Figura 3.1 Modelo adaptativo propuesto para la segmentación de imágenes por discontinuidades. ....	89
Figura 3.2 Caso 1, modelo con una base de conocimiento vacía. ....	91
Figura 3.3 Caso 2, utilización incremental del conocimiento que se va adquiriendo durante el entrenamiento. ....	94
Figura 3.4 Utilización del modelo con el conocimiento adquirido para segmentar nuevos casos. ....	96
Figura 3.5 Implementación del modelo en C++. ....	98
Figura 3.6 Utilización de los descriptores MPEG-7 en la generación del modelo de predicción. ....	100
Figura 3.7 Utilización de los descriptores MPEG-7 en la segmentación de nuevas imágenes. ....	101
Figura 3.8 Aplicación <i>XnConvert</i> con la configuración de detección de bordes. ....	107
Figura 3.9 Prueba funcional del modelo implementado, las cuatro métricas coinciden con el máximo valor de similitud cuando la segmentación obtenida es igual al <i>ground truth</i> . ....	109
Figura 3.10 Imagen de salida de la segmentación realizada por el sistema sobre la imagen apple-1.png. ....	110
Figura 3.11 a) Imagen segmentada superpuesta a la original apple-1.png b) Paleta de colores dominantes de la imagen original. ....	111
Figura 3.12 Productos generados a la salida del sistema implementado. ....	113

---

Figura 3.13 Sección de configuración del sistema desarrollado. ....	114
Figura 4.1 a) Imagen original 13.png b) <i>Ground truth</i> gt_13.png. ....	115
Figura 4.2: La imagen 100.png fue la primera en ser segmentada por el sistema durante el entrenamiento. ....	131
Figura 4.3 Proporciones de la distribución de colores dominantes en la imagen de entrada 100.png. ....	132
Figura 4.4 Archivos y directorios generados por el sistema en las fases de entrenamiento y utilización. ....	137
Figura 4.5 Imagen 229.png segmentada comparada con el <i>ground truth</i> de la imagen original de entrada. ....	143
Figura 4.6 Proporciones de la distribución de colores dominantes en la imagen de entrada 229.png. ....	144
Figura 4.7 Superposición de la segmentación en la imagen original 229.png. ....	144
Figura 4.8 Distribución promedio de bordes en la imagen 229.png, descriptor EHD. ....	145
Figura 4.9 Imagen 225.png segmentada comparada con el <i>ground truth</i> de la imagen original de entrada y su diferencia. ....	145
Figura 4.10 Proporciones de la distribución de colores dominantes en la imagen de entrada 225.png. ....	146
Figura 4.11 Superposición de la segmentación en la imagen original 225.png. ....	146
Figura 4.12 Distribución promedio de bordes en la imagen 225.png, descriptor EHD .....	147
Figura 4.13 Segmentación de la imagen 150.png. ....	148
Figura 4.14 Proporciones de la distribución de colores dominantes en la imagen de entrada 150.png. ....	149
Figura 4.15 Segmentación de la imagen 7.png. ....	149
Figura 4.16 Proporciones de la distribución de colores dominantes en la imagen de entrada 7.png. ....	150
Figura 4.17 Distribución promedio de bordes en la imagen 150.png. ....	151
Figura 4.18 Distribución promedio de bordes en la imagen 7.png, descriptor EHD. ....	151
Figura 5.1 Opción de generar el <i>ground truth</i> de los nuevos casos para reentrenar el modelo. ....	162

## Glosario

**Autoadaptable:** La capacidad que posee un sistema o aplicación de ajustarse por sí mismo a los cambios del entorno con el que interactúan, sin tener que ser reconfigurado o reprogramado. Las herramientas de aprendizaje automático son programas cuyo comportamiento se adapta a sus datos de entrada (Shalev-Shwartz & Ben-David, 2014).

**Aprendizaje automático:** El aprendizaje automático (*Machine learning*) se refiere a la detección automatizada de patrones significativos en los datos, esto es, convirtiendo los datos en información relevante (Shalev-Shwartz & Ben-David, 2014).

**Descriptor:** Permite describir un objeto en una imagen al extraer características propias para reconocerlo. Se puede diferenciar entre descriptores de bordes y de regiones (Pajares & de la Cruz, 2008).

**Descriptores Visuales MPEG-7:** El MPEG-7 es un estándar para describir las características del contenido multimedia. El objetivo principal del estándar visual MPEG-7 es proporcionar descripciones estandarizadas de imágenes transmitidas o almacenadas o bits de encabezado estandarizados de video (*descriptores visuales* de bajo nivel) que ayudan a los usuarios o aplicaciones a identificar, categorizar o filtrar imágenes o videos. Estos descriptores de bajo nivel se pueden usar para comparar, filtrar o explorar imágenes o videos basándose únicamente en descripciones visuales no textuales del contenido, o en combinación con consultas comunes basadas en texto. Se utilizan de manera diferente para diferentes dominios de usuario y diferentes entornos de aplicación (Manjunath, Salembier, & Sikora, 2003).

**Detección de características:** La detección de características es el proceso de detectar estructuras significativas en una imagen (color, textura, formas, contornos, entre otras). La salida de un detector de características suele ser una cantidad de ubicaciones específicas en la imagen, llamadas puntos de características o puntos característicos (Mukherjee, Jonathan Wu, & Wang, 2015).

**Ground truth:** En el aprendizaje automático, el término "*ground truth*" se refiere al conjunto de datos de entrenamiento utilizado con las técnicas de aprendizaje automático para probar el funcionamiento de modelos de procesamiento de imagen, Son la respuesta esperada (verdad conocida) a la salida del proceso, es decir la realidad que se quiere predecir con el modelo dado. En el contexto de visión artificial, los datos *ground truth* son las imágenes de referencia o resultado esperado en un proceso de segmentación (Krig, 2014).

**Heurística:** Se conoce como heurística al conjunto de técnicas o métodos para resolver un problema. La palabra heurística es de origen griego *εὕρισκειν* que significa "hallar, inventar".

**Imagen digital:** Se refiere a una función bidimensional de la luz y la intensidad, a la que se denota por  $f(x,y)$ , donde el valor o amplitud de  $f$  en las coordenadas espaciales  $(x,y)$  determina la intensidad (iluminación) de la imagen en un punto (Gonzalez & Woods, 2008).

**Modelo:** Abstracción de un objeto de estudio, cuyo objetivo es comprenderlo antes de implementarlo y dado que los modelos omiten detalles no esenciales es más sencillo manipularlos que la entidad original (Rumbaugh, Blaha, Premerlani, Eddy, & Lorensen, 1996).

**MPEG:** Es el grupo de expertos en imágenes en movimiento (*Moving Picture Experts Group*), establecido en 1988, está dedicado al desarrollo de estándares de compresión audiovisual digital. Estándares que han transformado la forma en que el contenido audiovisual es producido por múltiples industrias, entregado a través de todo tipo de canales de distribución y consumido por una variedad de dispositivos (Manjunath, Salembier, & Sikora, 2003). Es un grupo de trabajo de la organización de normas ISO/IEC (Organización Internacional de Normas/Comité Electrotécnico Internacional).

**Segmentación:** El propósito [de este proceso] es dividir una imagen en varias regiones con base en ciertas características o propiedades de similitud (nivel de gris, espectro en frecuencia, texturas, etc.) o resaltar las diferencias obvias representadas en diferentes regiones, por disimilitud o discontinuidad (Meiling Gong, 2017).

# Acrónimos

**CBIR:** *Content-Based Image Retrieval* (Recuperación de imágenes con base en contenido).

**CFHu:** Comparación de similitud de formas con base en descriptores de Hu.

**CLD:** *Color Layout Descriptor* (Descriptor de diseño de color), estándar MPEG-7.

**CSD:** *Color Structure Descriptor* (Descriptor de estructura de color), estándar MPEG-7.

**CShD:** *Contour-Based Shape Descriptor* (Descriptor de forma con base en el contorno), estándar MPEG-7.

**CSV:** Archivo de texto delimitado por comas.

**DCD:** *Dominant Color Descriptor* (Descriptor de color dominante), estándar MPEG-7.

**EHD:** *Edge Histogram Descriptor* (Descriptor de histograma de borde), estándar MPEG-7.

**HSV:** *Hue-Saturation-Value* (Espacio de color Tono-Saturación-Valor).

**HTD:** *Homogeneous Texture Descriptor* (Descriptor de textura homogénea), estándar MPEG-7.

**IA:** Inteligencia Artificial.

**IBL:** *Instance Based Learning* (Aprendizaje con base en instancias).

**K-NN:** *K-Nearest Neighborhood* (*K*-Vecinos más cercanos).

**MPEG-7:** *Moving Picture Experts Group – 7* (Grupo de expertos en imágenes en movimiento – estándar 7). Estándar internacional ISO / IEC oficialmente conocido como la Interfaz de Descripción de Contenido Multimedia.

**ROI:** *Region of interest* (Región de interés).

**RSD:** *Region-Based Shape Descriptor* (Descriptor de forma con base en región), estándar MPEG-7.

**SCD:** *Scalable Color Descriptor* (Descriptor de color escalable), estándar MPEG-7.

**SVM:** *Support Vector Machine* (Máquina de soporte vectorial).

# Notación

Símbolo	Explicación
$X, Y, Z$	Los caracteres en mayúscula y negrita representan conjuntos arbitrarios.
$x, y, z$	Los caracteres en minúscula, cursiva y negrita representan elementos de conjuntos arbitrarios.
$X, Y, Z$	Los caracteres en negrita y mayúsculas se utilizan para representar conjuntos de puntos.
$x, y, z$	Los caracteres en negrita y minúscula se utilizan para representar puntos, es decir, elementos de conjuntos de puntos.
$\mathbb{X}, \mathbb{Y}, \mathbb{Z}$	Los caracteres en doble trazo y mayúsculas se utilizan para representar conjuntos de números.
$\mathbb{N}$	El conjunto de los números naturales .
$\mathbb{Z}$	El conjunto de los números enteros.
$\mathbb{R}$	El conjunto de los números reales.
$\sigma$	Desviación estándar.
$\sigma^2$	Varianza.
$\nabla$	Operador gradiente.
$\nabla^2$	Operador laplaciano.
$\bar{x}$	Media o promedio estadístico.
$\ominus$	Operador de erosionado entre imágenes.
$(a_1, a_2, a_3, \dots, a_n)$	Secuencia o lista ordenada de $n$ elementos llamada <i>n-tupla</i> .
$: \text{ ó }  $	Tal que.
$\in$	Pertenece o es elemento de.
$\exists$	Existe.
$\subset$	Es un subconjunto de.
$\forall$	Para todo.
$f: \mathbb{X} \rightarrow \mathbb{Y}$	$f$ es una función del conjunto $\mathbb{X}$ aplicado sobre el conjunto $\mathbb{Y}$ .
$\mathbb{X} \times \mathbb{Y}$	Producto cartesiano $\mathbb{X} \times \mathbb{Y} = \{(x, y): x \in \mathbb{X}, y \in \mathbb{Y}\}$ .
$(\dots)$ ó $[ \dots ]$	Indica que el texto citado de alguna referencia ha sido fragmentado, es decir, que se ha omitido material intencionalmente que, a criterio del investigador, no es relevante para la construcción del argumento.

[idea propia] Representa una intromisión que el investigador inserta en la idea original de un autor citado.

# Organización de la tesis

La tesis está organizada de la siguiente manera:

En el Capítulo 1, se presenta la introducción, se describe el problema planteado, las preguntas de investigación, los objetivos a lograr, la justificación y los beneficios esperados, los alcances y las limitaciones de este trabajo de investigación y el marco conceptual.

En el Capítulo 2, se presenta la revisión de la literatura, estado del arte y trabajos relacionados, la propuesta de solución y el método empleado.

En el Capítulo 3, se describe el diseño del modelo propuesto y su implementación.

En el Capítulo 4, se presentan los experimentos, resultados obtenidos y su discusión.

En el Capítulo 5, se concluye el trabajo de investigación realizado, productos y aportaciones generados, objetivos logrados y trabajos futuros.

Al final del documento se proporcionan las referencias citadas en él y se tiene un apartado de anexos con información adicional para los capítulos que así lo requerían.

# Capítulo 1

## Introducción

### 1.1 Motivación

La segmentación es una etapa medular de un sistema de visión artificial clásico, esta etapa definirá el éxito o fracaso de las etapas de análisis posteriores Figura 1.1 (Moeslund, 2012).

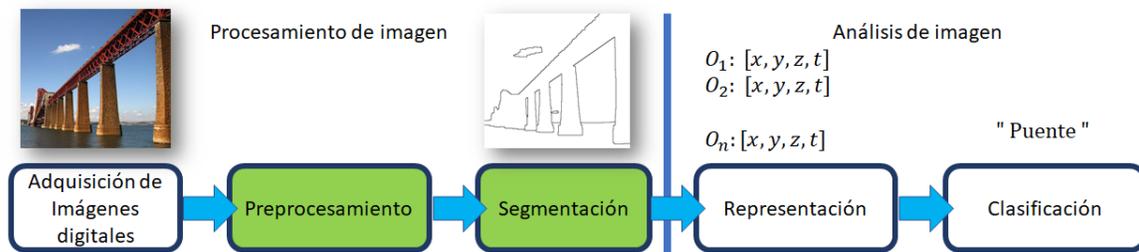


Figura 1.1 Sistema de visión artificial clásico (Moeslund, 2012).

El objetivo de la segmentación es subdividir una imagen en sus partes u objetos constituyentes. El nivel al que se lleva a cabo esta subdivisión depende del problema a resolver. Esto es, la segmentación debería detenerse cuando los objetos de interés de un problema en particular hayan sido aislados (Gonzalez & Woods, 2008).

Es un tema complejo y abierto a la investigación, además de ser una de las tareas más difíciles del procesamiento en un sistema de visión artificial clásico.

En general, la segmentación es pieza clave hacia tareas de alto nivel tales como la comprensión de imágenes, y sirve en una variedad de aplicaciones, incluyendo reconocimiento de objetos, análisis de escenas o indexado de imágenes y video.

Los algoritmos de segmentación de imágenes a menudo se seleccionan heurísticamente, y se ajustan estrechamente mediante un experto en procesamiento de imágenes, con respecto a las necesidades de la aplicación. Estos algoritmos generalmente se basan en dos propiedades básicas (Gonzalez & Woods, 2008), éstas propiedades son:

1. **Por similitud:** se divide la imagen buscando zonas o regiones que tengan valores similares de nivel de intensidad de pixel, de espectro en frecuencia, de textura, entre otros.
2. **Por discontinuidades o disimilitud:** se divide la imagen observando los cambios bruscos del nivel de intensidad entre los píxeles, para localizar las fronteras o bordes que delimitan los objetos en la imagen.

## 1.2 Segmentación por discontinuidades

La segmentación por discontinuidades consiste en localizar las fronteras (*boundaries*) que delimitan los objetos de interés en una imagen, a través de la detección de sus bordes o contornos (*edges, borders o contours detection*), en la Figura 1.2 se observa la imagen original 4.png del conjunto BSD300 con su respectiva imagen de bordes (Martin, Fowlkes, Tal, & Malik, 2001).

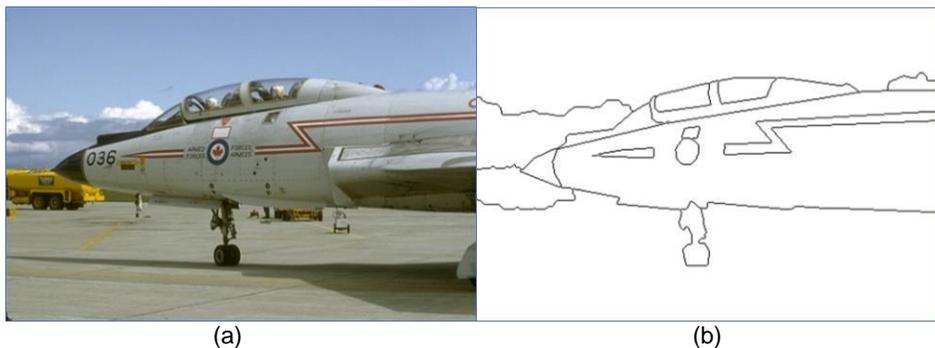


Figura 1.2 Segmentación por discontinuidades, a) imagen original, b) imagen segmentada con bordes o fronteras detectadas. Imágenes tomadas del conjunto BSD300 (Martin, Fowlkes, Tal, & Malik, 2001).

Esta investigación se enfocó en la segmentación de imágenes por discontinuidades.

Un sistema de visión artificial clásico (Figura 1.1), generalmente se diseña para procesar un determinado tipo (clase o categoría) de imágenes y es de configuración fija para un tipo particular de imágenes. Sería deseable que este procesamiento tuviera cierto grado de flexibilidad y adaptabilidad, con el objeto de segmentar distintas clases de imágenes.

En la Figura 1.3 se muestran tres diferentes *líneas de procesamiento (pipelines)* para segmentar tres distintas clases de imágenes. Cada *pipeline* está configurado o codificado para una clase particular de imagen. Si se intercambia el tipo de imagen de entrada entre los tres *pipelines*, no hay garantía de obtener una segmentación adecuada.

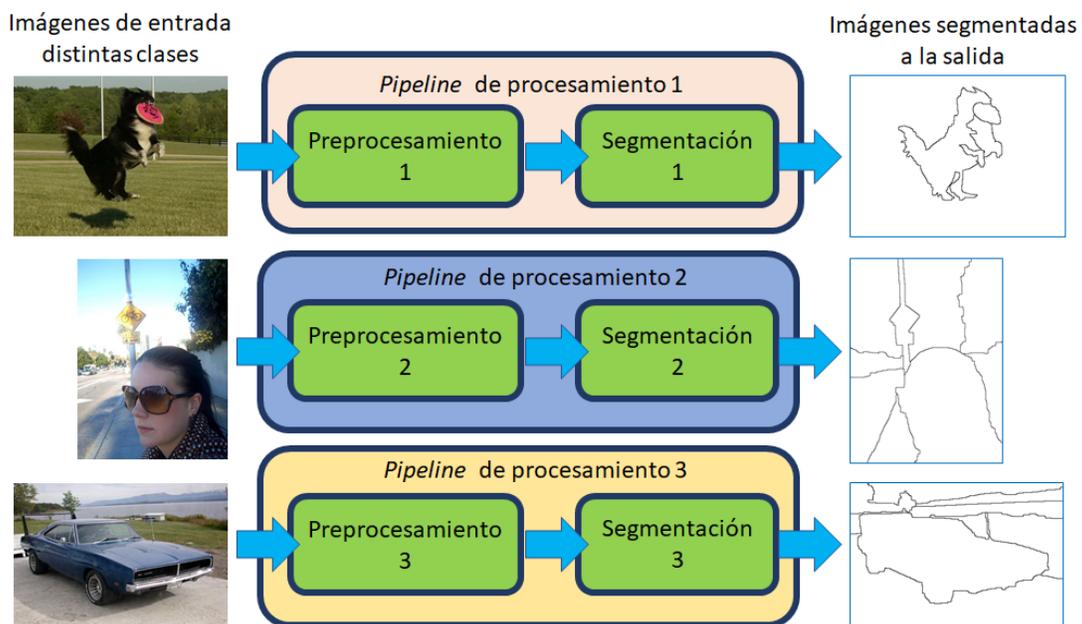


Figura 1.3 *Pipelines* de segmentación para tres distintas clases de imágenes.

Si bien, la segmentación de imagen por discontinuidades tiene como objetivo localizar los límites de los objetos de acuerdo con la interpretación visual humana, es inherentemente un “problema mal planteado” (*ill-posed problem*) ya que existen múltiples segmentaciones válidas para la misma imagen de entrada, dependiendo

de las interpretaciones humanas (*ground truth*) realizadas sobre ésta (Unnikrishnan, Pantofaru, & Hebert, 2007), (Peng, Zhang, Mou, & Yang, 2017).

En el campo de la investigación del procesamiento de imágenes, la mayoría de los problemas se definen como mal planteados, en el sentido de que no es posible proporcionar una solución única.

Un problema está bien planteado si:

1. Tiene una solución (Existencia).
2. La solución es única (Unicidad).
3. La solución depende continuamente de los datos del problema (datos de entrada).

Un problema que no cumple con las condiciones anteriores se define como un problema mal planteado (*ill-posed problem*) (Hadamard, 1923).

En (Béréziat & Herlin, 2009) se explica que, bajo la definición anterior, la estimación del flujo óptico, el registro de imágenes, la coincidencia por similitud de curvas o superficies, el seguimiento de múltiples objetos, la segmentación, la restauración, la deconvolución, la eliminación de ruido y la forma del sombreado en imágenes, son problemas mal planteados.

Como un problema de procesamiento de imágenes generalmente se modela mediante un sistema de ecuaciones a resolver, el llamado modelo de imagen, este tipo de mal planteamiento significa que el modelo de imagen no es invertible. Una segunda causa del mal planteamiento ocurre cuando el cálculo de las características de la imagen puede obtenerse mediante diferentes algoritmos. Por ejemplo, determinar el gradiente de la imagen requiere aproximar un operador diferencial continuo, por uno discreto, entre varias formulaciones de diferencias finitas posibles; cada uno brinda un resultado diferente.

---

Una estrategia común para resolver problemas mal planteados es proporcionar el modelo de imagen con información adicional o establecer algunas restricciones.

Se pueden considerar dos opciones:

- 1) Proporcionar información explícita: se utilizan imágenes adicionales para ampliar el conjunto de datos de entrada. Sin embargo, esto generalmente no es posible porque otras adquisiciones que tienen las propiedades solicitadas no están disponibles.
- 2) Proporcionar información implícita: se pueden utilizar hipótesis sobre las propiedades de la imagen o restricciones sobre la solución. Una restricción habitual es restringir la dimensión del espacio de soluciones admisibles.

### 1.3 Planteamiento del problema

El problema que se presenta es cómo diseñar un modelo que determine las técnicas adecuadas para segmentar imágenes de entrada de distintas categorías, a partir de casos de ejemplo Figura 1.4.

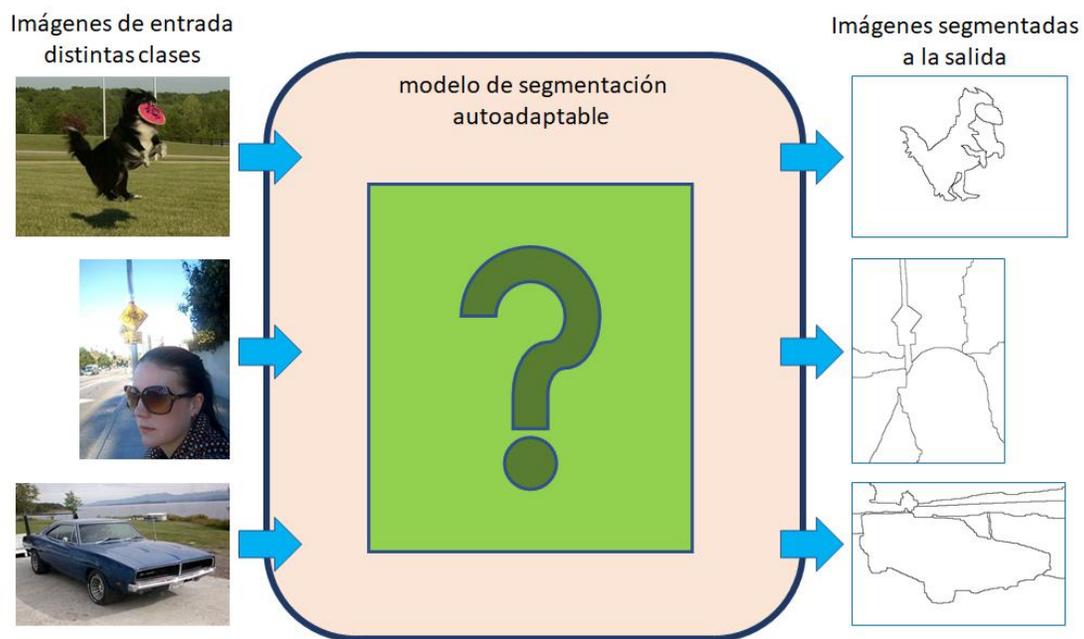


Figura 1.4 Problema a resolver.

## 1.4 Preguntas de investigación

Del problema planteado surgieron varias interrogantes:

1. ¿Será posible establecer un modelo con un mecanismo de aprendizaje automático que le permita adaptarse a los diferentes tipos de imágenes de entrada, partiendo de un conjunto predefinido de técnicas de preprocesamiento y segmentación?
2. ¿Será posible diseñar un modelo que pueda aprender a segmentar distintas clases de imágenes, a partir de procesar imágenes de ejemplo mediante la aplicación de un conjunto predefinido de técnicas de preprocesamiento y segmentación, de forma tal, que le permita adaptarse a los diferentes tipos de imágenes de entrada?
3. ¿Existirá alguna relación o patrón característico entre las imágenes segmentadas adecuadamente y sus correspondientes técnicas aplicadas para lograrlo?
4. ¿Existirán técnicas de preprocesamiento y segmentación comunes entre imágenes de la misma categoría?
5. ¿Existirán características que describan a las imágenes y que den algún indicio sobre las técnicas que se les debería aplicar para segmentarlas adecuadamente?
6. ¿Podría el modelo informar la forma en que realizó la segmentación de las imágenes en casos de éxito y reportar las características de las imágenes donde no lo logró?
7. ¿Sería posible utilizar el modelo para segmentar nuevas imágenes identificadas a través de algunas de sus características intrínsecas obtenidas mediante sus descriptores?

## 1.5 Objetivos

### 1.5.1 Objetivo general

Diseñar e implementar un modelo que aprenda cómo ajustar un sistema de visión artificial clásico, para resolver un problema específico, partiendo de un conjunto de primitivas o técnicas aplicadas a un conjunto de imágenes.

### 1.5.2 Objetivos específicos

1. Definir el concepto del modelo y diseñar la arquitectura correspondiente para su implementación.
2. Seleccionar las técnicas de aprendizaje supervisado apropiadas para el modelo.
3. Seleccionar técnicas de segmentación por discontinuidades adecuadas.
4. Seleccionar las herramientas de desarrollo necesarias para la implementación.
5. Desarrollar las interfaces supervisor [observador durante el entrenamiento] y ejecutor [observador durante la utilización] que expliquen cómo se logra la segmentación de las imágenes.
6. Definir la base de conocimiento del sistema, que debe contener el conocimiento adquirido durante el entrenamiento.
7. Integrar en el modelo las técnicas de aprendizaje automático y las de procesamiento de imagen (técnicas de preprocesamiento y de segmentación).
8. Probar el modelo con el conocimiento adquirido, aplicado sobre nuevos casos.

## 1.6 Justificación y beneficios

Justificación:

- Contar con sistema de visión artificial que se ajuste y adapte a nuevos casos sin requerir la recodificación o reconfiguración.

Beneficios:

- Tener un modelo adaptable que se pueda ampliar si se le agregan nuevas técnicas de preprocesamiento, de segmentación y de aprendizaje automático, y se le entrena con diferentes tipos de imágenes ejemplo.
- El contar con un modelo adaptable facilitaría la evaluación y el desarrollo de sistemas de visión artificial.

## 1.7 Alcances y limitaciones

Alcances:

1. El modelo entregará una imagen segmentada a la salida.
2. El modelo segmentará imágenes dentro del espacio de soluciones permitido por su base de conocimiento (adquirida durante el entrenamiento).
3. La segmentación se realizará a nivel de discontinuidades.
4. El modelo utilizará aprendizaje supervisado.
5. El conjunto de primitivas estará integrado por:
  - a. Técnicas de preprocesamiento (al menos 10).
  - b. Técnicas de segmentación (al menos 5).
  - c. Técnicas de descripción de imagen (al menos 10 descriptores).
  - d. Técnicas de aprendizaje automático (al menos 5).

Limitaciones:

1. No se persigue que el sistema opere en tiempo real.
2. No se considera como parámetro de calidad el tiempo de respuesta.
3. El modelo no reconoce objetos ni instancias.
4. Se utilizarán bancos de imágenes con fondos contrastados o con algún tipo de gradiente (que contengan imagen de referencia o *ground truth*).

## 1.8 Marco conceptual

Derivado del problema planteado y de las preguntas de investigación, existió la necesidad de investigar sobre temas particulares tales como: técnicas de preprocesamiento, técnicas de segmentación, técnicas de aprendizaje automático, técnicas de descripción de características de imagen, técnicas de adquisición de conocimiento y métricas de similitud. Así, en esta sección, se presentan los conceptos fundamentales necesarios para la comprensión del trabajo desarrollado.

En el tratamiento de imágenes es importante destacar los resultados que se obtienen en tres distintas categorías, Tabla 1.1 (Young, Gerbrands, & J. van Vliet, 2007).

**Tabla 1.1 Resultados obtenidos en tres diferentes tratamientos de imágenes.**

<b>Tratamiento de imagen</b>	<b>Resultado obtenido</b>
<b>1. Procesamiento de imagen</b>	Imagen de entrada → Imagen de salida
<b>2. Análisis de imagen</b>	Imagen de entrada → Medidas de salida
<b>3. Comprensión de imagen</b>	Imagen de entrada → Descripción de alto nivel de salida

Fuente (Young, Gerbrands, & J. van Vliet, 2007). Elaboración propia.

Esta investigación se basó en técnicas de procesamiento de imagen (preprocesamiento y segmentación) y técnicas de análisis de imagen (descriptores y métricas) en suma con las técnicas de aprendizaje automático.

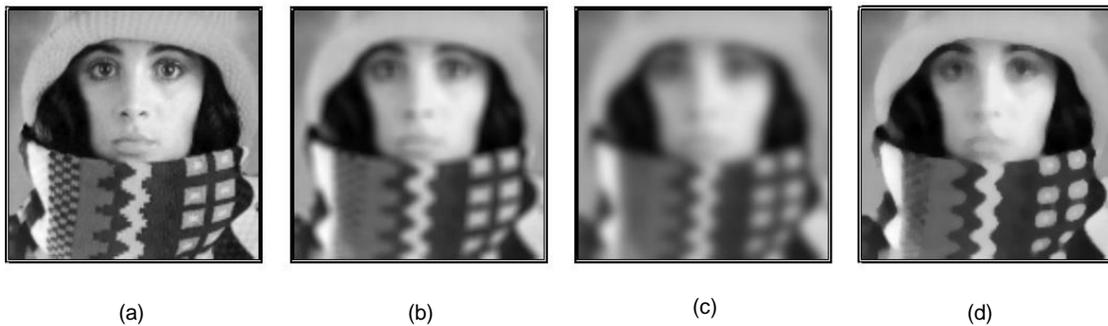
### 1.8.1 Preprocesamiento de la imagen

A veces la calidad de la imagen no es lo suficientemente buena, de forma que no es posible extraer la información adecuadamente, ello implica la necesidad de utilizar ciertas técnicas o filtros de mejora de la calidad de la imagen original, técnicas conocidas como de preprocesamiento.

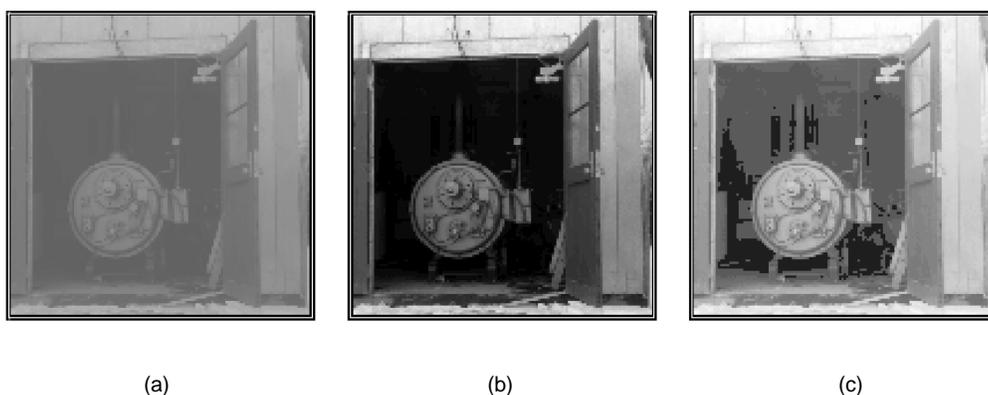
Se consideraron dos procesos dentro de esta etapa:

- Suavizado
- Realzado

El suavizado se encamina hacia la supresión del ruido introducido durante la captura de la imagen, Figura 1.5, mientras que el realzado se orienta a eliminar falsos reflejos y sombras que dificultan también la extracción de la información para el procesamiento de la imagen, Figura 1.6. Si la calidad de la imagen capturada es suficientemente buena para la aplicación en cuestión, el preprocesamiento puede obviarse (Pajares & de la Cruz, 2008).



**Figura 1.5** Aplicación de varios filtros de suavizado lineales y no lineales sobre una imagen a) Imagen original b) Filtro de media 5x5 c) Filtro Gaussiano ( $\sigma = 2.5$ ) d) Filtro de mediana 5x5 (Young, Gerbrands, & J. van Vliet, 2007).



**Figura 1.6** Aplicación de técnicas de realzado a) Imagen original b) Histograma extendido c) Histograma ecualizado (Young, Gerbrands, & J. van Vliet, 2007).

### 1.8.2 Segmentación por discontinuidades

La segmentación es el proceso mediante el cual se extrae cierta información subyacente en las imágenes para su posterior uso. La segmentación por discontinuidades se basa en la detección de bordes o fronteras entre los diferentes objetos dentro de una imagen (Pajares & de la Cruz, 2008).

La segmentación por discontinuidades comprende una serie de subprocesos a saber:

- Detección de bordes.
- Cierre de contornos.
- Etiquetado.

En la Figura 1.7 se muestran los bloques del proceso de la segmentación.

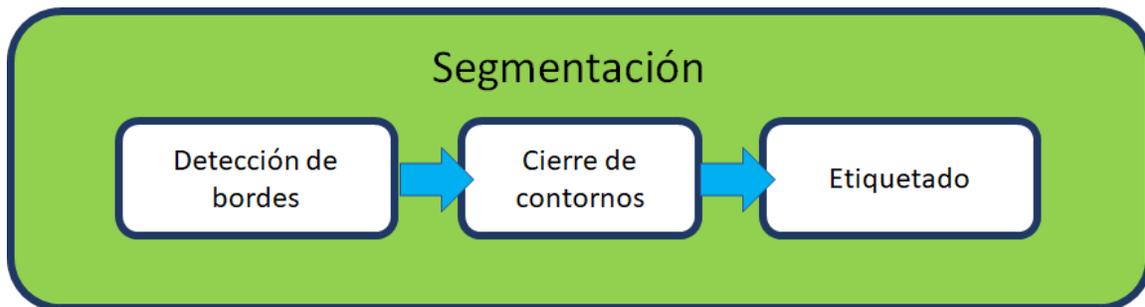


Figura 1.7 Subprocesos de la segmentación.

La detección de bordes implica una manipulación de la imagen original, que supone en definitiva una transformación de esta, de forma que los valores de los píxeles originales son modificados mediante ciertas funciones de transformación u operadores.

Para detectar los cambios abruptos en los niveles de intensidad de los píxeles de la imagen y localizar los bordes (discontinuidades), se emplean los siguientes tipos de operadores:

- Operadores de primera derivada (Gradiente)
- Operadores de segunda derivada (Laplaciano)
- Operadores morfológicos (Gradiente morfológico)

Gradiente:

Sea la función imagen definida por  $I(x, y)$ . Los bordes de  $I$  se obtienen de la magnitud del vector gradiente  $|\nabla I(x, y)|$ , Ec.( 1.1 ) y el ángulo  $\phi(x, y)$ , Ec.( 1.2 ), que determina su dirección.

$$G(x, y) = |\nabla I(x, y)| = \sqrt{G_x^2 + G_y^2} \quad (1.1)$$

$$\phi(x, y) = \tan^{-1} \frac{G_y}{G_x} \quad (1.2)$$

Laplaciano:

El laplaciano de una función imagen  $u(x, y)$  se define como el operador de segunda derivada, Ec. ( 1.3 ).

$$G(x, y) = |\nabla^2 I(x, y)| \quad (1.3)$$

Gradiente morfológico:

El término morfología comúnmente se refiere a una rama de la biología que trata con la forma y estructura de animales y plantas. Se utiliza el mismo término en el contexto de la morfología matemática, como herramienta para extraer elementos que son útiles en la representación y descripción de la forma de una región, tales como: bordes, esqueletos y convexidades.

En el caso particular de la extracción de borde de la imagen se define lo siguiente: Sea  $A$  el conjunto de píxeles una imagen  $I(x, y)$  y sea  $\beta(A)$  los bordes de ese conjunto, éstos pueden ser obtenidos aplicando la operación morfológica erosionado, aplicando sobre  $A$  un elemento estructurante  $B$  y realizando la diferencia entre  $A$  y su erosionado, Ec. ( 1.4 ) (Gonzalez & Woods, 2008).

$$\beta(A) = A - (A \ominus B) \tag{1.4}$$

En la Figura 1.8 se muestran ejemplos de los operadores indicados anteriormente.

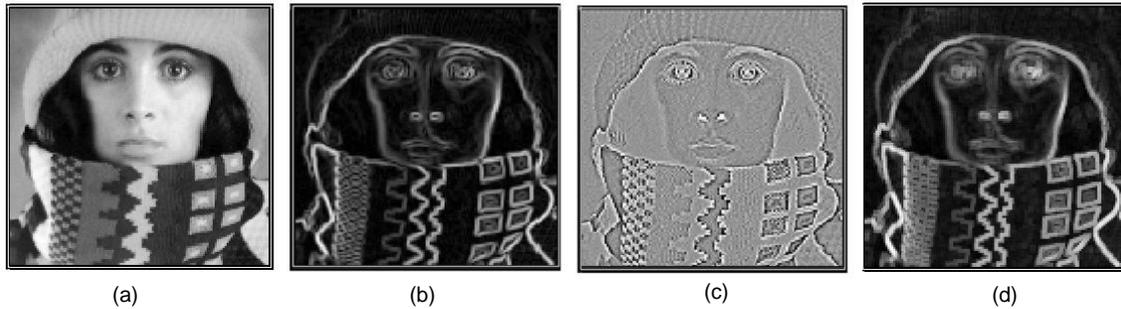


Figura 1.8 Ejemplos de los tres operadores a) Imagen original b) Gradiente c) Laplaciano d) Gradiente morfológico (Young, Gerbrands, & J. van Vliet, 2007).

Una vez detectados los bordes, es conveniente cerrar los contornos obtenidos y finalmente, los bordes cerrados son identificados o etiquetados como objetos de interés, como se observa en la Figura 1.9.

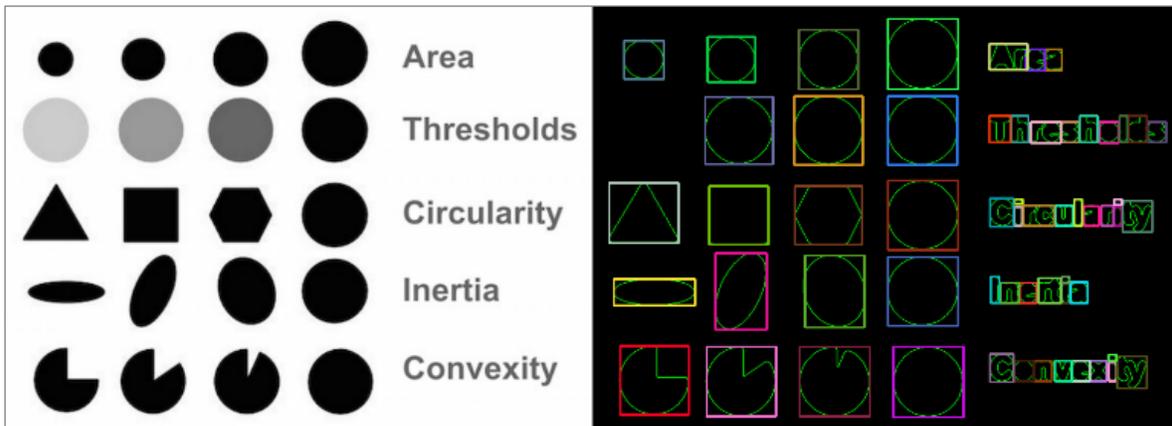


Figura 1.9 Bordes detectados y etiquetados. Imagen tomada de (Mallick, 2015).

La segmentación por discontinuidades, debido a su naturaleza intrínseca tiene algunas limitaciones en su operación, es sensible al bajo contraste de las imágenes (se dificulta la detección de las diferencias de intensidad), al exceso de texturas o con demasiado ruido; bajo estas circunstancias la técnica no es funcional.

### 1.8.3 Descriptores

La descripción de una imagen consiste en extraer de ésta propiedades o atributos para su uso en las aplicaciones particulares. Básicamente, se trata de reconocer las diferentes estructuras de la imagen para su identificación de forma inequívoca (Pajares & de la Cruz, 2008).

Los objetos o estructuras, en una imagen, son representados como una colección de píxeles. Así, para propósitos de reconocimiento, es necesario describir las propiedades de grupos de píxeles.

La descripción es a menudo un conjunto de números, *los descriptores de los objetos*. A partir de éstos, se pueden comparar y reconocer los objetos, buscando simplemente hacer coincidir los descriptores de los objetos en una imagen conocida con los descriptores de imágenes desconocidas.

Sin embargo, para que sean útiles en el reconocimiento, los descriptores deberían tener cuatro importantes propiedades:

1. Definir un conjunto completo, esto es, dos objetos deben tener la misma descripción sí y solo sí, tienen la misma forma.
2. Deberían ser congruentes, tal que se debería poder reconocer objetos similares cuando éstos tengan descriptores similares.
3. Es conveniente que tengan propiedades de invarianza. Por ejemplo, invarianza a la rotación, sería útil para reconocer objetos independientemente de su orientación.

4. Deberían ser un conjunto compacto, un descriptor debería representar la esencia de un objeto de manera eficiente, esto es, sólo debería contener la información que hace único al objeto o lo diferencia de otros objetos.

Desafortunadamente, **no hay conjuntos completos y compactos de descriptores** para caracterizar objetos en general. Como tal, el proceso de reconocimiento está fuertemente relacionado a cada aplicación en particular, con un tipo específico de objetos (Nixon & Aguado, 2008).

A continuación, se establece:

Sea una  $n$ -*tupla* definida como una secuencia o lista ordenada de  $n$  elementos, siendo  $n$  un número natural y  $a$  tomando valores enteros o reales, Ec. ( 1.5 ).

$$(a_1, a_2, a_3, \dots, a_n) \quad n \in \mathbb{N} \quad (1.5)$$

Los descriptores son tuplas de  $n$  componentes que representan o describen características sobresalientes en una imagen, un descriptor también se conoce como *vector de características* o *patrón característico* Figura 1.10.

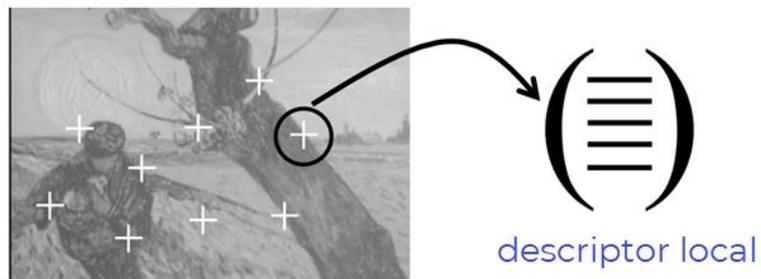


Figura 1.10 Detección de características sobresalientes en una imagen.

En la Tabla 1.2 se muestra un ejemplo con el descriptor de histograma de borde (EHD) del estándar MPEG-7 (R & Kannan, 2009), es un vector con 80 componentes de números enteros ( $80$ -*tupla* ).

Tabla 1.2 Descriptor EHD de 80 componentes (80-tupla).

<i>n</i>	Descriptor
80	[0 0 0 0 0 0 0 0 0 0 1 0 5 0 4 4 2 5 4 6 0 0 0 0 1 1 5 0 5 3 1 5 1 5 1 1 1 2 3 4 1 3 3 3 6 1 5 3 4 4 0 3 1 2 4 1 1 2 2 4 2 1 3 4 6 2 4 3 2 5 4 4 3 3 3 5 2 5 2]

Fuente: descriptor extraído con la biblioteca *MPEG7FexLib* (Bastan, Cam, Gudukbay, & Ulusoy, 2010). Elaboración propia.

### Descriptores visuales MPEG-7

El objetivo principal del estándar visual MPEG-7 es proporcionar descripciones estandarizadas de imágenes transmitidas o almacenadas o bits de encabezado de video (*descriptores visuales* de bajo nivel) que ayudan a los usuarios o aplicaciones a identificar, categorizar o filtrar imágenes o videos.

Estos descriptores de bajo nivel se pueden utilizar para comparar, filtrar o explorar imágenes o videos basándose únicamente en descripciones visuales no textuales del contenido, o en combinación con consultas comunes basadas en texto. Se utilizan de manera diferente para diferentes dominios de usuario y diferentes entornos de aplicación.

Los descriptores MPEG-7 se pueden clasificar en términos generales en *descriptores visuales* generales y *descriptores visuales* específicos del dominio. Los primeros incluyen características de color, textura, forma y movimiento, mientras que los segundos dependen de la aplicación (Manjunath, Salembier, & Sikora, 2003).

En la Figura 1.11 se observan las categorías de los *descriptores visuales* generales de color (Ohm, y otros, 2003), textura (Choi, Won, Ro, & Manjunath, 2003) y forma (Miroslaw Bober, 2003).

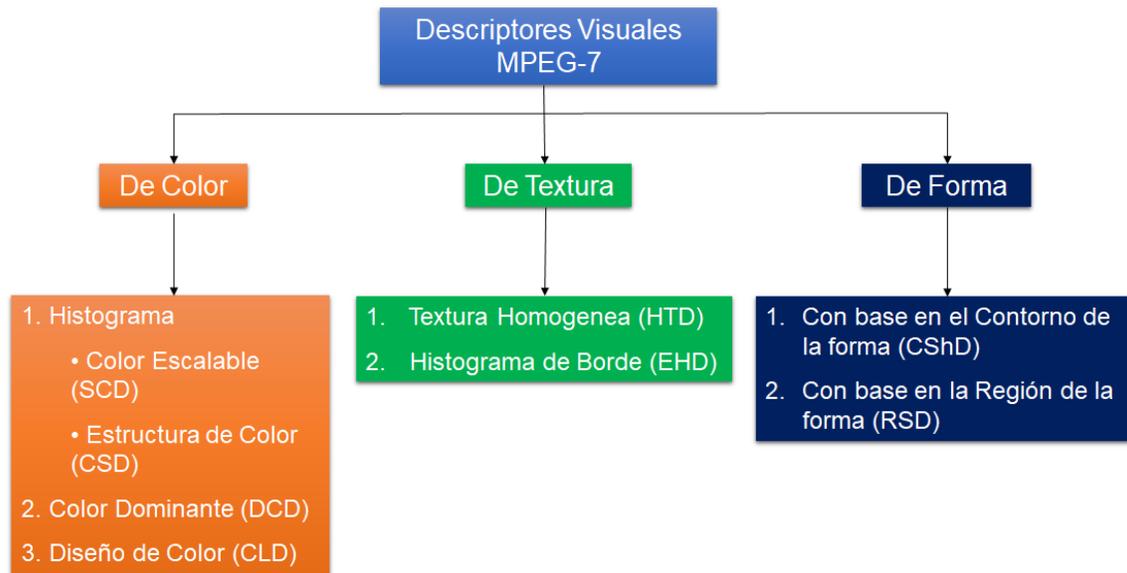


Figura 1.11 Conjunto de *descriptores visuales* MPEG7.

Se resumen brevemente los *descriptores visuales* definidos como parte del estándar MPEG-7 (Manjunath, Salembier, & Sikora, 2003):

- El descriptor de color dominante (DCD) permite la especificación de un pequeño número de valores de color dominantes, así como sus propiedades estadísticas, como la distribución y la varianza. Su propósito es proporcionar una representación efectiva, compacta e intuitiva de los colores presentes en la imagen o una región de ésta. Sus principales aplicaciones son la recuperación por similitud en bases de datos de imágenes y la exploración de bases de datos de imágenes con base en un color único o en varios.
  - Definición del descriptor DCD:

$$DCD = \{(c_i, p_i, v_i), s\}, \quad (i = 1, 2, \dots, N).$$

Donde  $N$  es el número de colores dominantes. Cada valor de color dominante  $c_i$  es un vector de valores de componentes del espacio de color correspondiente (por ejemplo, un vector tridimensional en el espacio de color RGB). El porcentaje  $p_i$  es la fracción de píxeles en la imagen o región de la imagen correspondiente al color

$c_i$ . La variación de color opcional  $v_i$  describe la variación de los valores de color de los píxeles en un grupo alrededor del color representativo correspondiente. La coherencia espacial  $s$  es un número único que representa la homogeneidad espacial general de los colores dominantes en la imagen. El número de colores dominantes  $N$  puede variar de una imagen a otra, y los autores encontraron que un máximo de **ocho colores dominantes es suficiente** para representar una imagen o una región de interés de la imagen (ROI).

El valor de **coherencia espacial**  $s$  se cuantifica de manera no uniforme a 5 bits, donde 31 significa mayor confianza y 1 significa que no hay confianza. El valor cero se usa para casos en los que no se calcula.

- El descriptor de color escalable (SCD) se puede interpretar como un esquema de codificación con base en la transformada de *Haar* aplicada a través de valores de un histograma de color en el espacio de color *HSV (Hue-Saturation-Value)*. Los valores del histograma se extraen, se normalizan y se asignan de forma no lineal a una representación entera de cuatro bits, lo que da mayor importancia a los valores pequeños. La transformada de *Haar* se aplica a los valores enteros de cuatro bits en los contenedores de histograma. La unidad básica de la transformación consiste en una operación de suma y una operación de diferencia, que se relacionan con filtros primarios de paso bajo y paso alto. Sumar pares de contenedores adyacentes es equivalente al cálculo de un histograma con la mitad del número de contenedores.
- El descriptor de estructura de color (CSD) representa una imagen tanto por la distribución de color de la imagen (similar a un histograma de color) como por la estructura espacial local del color. La información adicional sobre la

estructura del color hace que el descriptor sea sensible a ciertas características de imagen para las cuales el histograma de color es ciego.

- El descriptor de diseño de color (CLD) es una representación de color muy compacta e invariable en resolución para la recuperación de imágenes a alta velocidad. Está diseñado para representar eficientemente la distribución espacial de colores, captura el diseño espacial de los colores representativos en una cuadrícula superpuesta a una región o imagen completa. Esta característica se puede utilizar para una amplia variedad de recuperación con base en similitudes, filtrado de contenido y visualización. Es especialmente útil para aplicaciones de recuperación con base en estructuras espaciales, por ejemplo, recuperación con base en bocetos e identificación de segmentos de video. CLD proporciona información sobre la distribución espacial del color dentro de las imágenes. Después de dividir una imagen en 64 bloques, se extrae el descriptor de cada uno de los bloques en función de la transformada discreta del coseno DCT (*Discrete Cosine Transform*), en una matriz 2D de colores representativos locales en el espacio de color YCbCr; este espacio de color es heredado de los estándares MPEG precedentes, (MPEG-1/2/4). Se define mediante una transformación lineal del espacio de color RGB Ecs. ( 1.6 ) , ( 1.7 ) , ( 1.8 ).

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \quad (1.6)$$

$$Cb = -0.169 * R - 0.331 * G + 0.500 * B \quad (1.7)$$

$$Cr = 0.500 * R - 0.419 * G - 0.081 * B \quad (1.8)$$

Así el descriptor es representado por sus componentes  $Y$  (luminancia)  $Cr$  (diferencia de color rojo) y  $Cb$  (diferencia de color azul)  $CLD = \{DY, DCr, DCb\}$ .

- El descriptor de textura homogéneo (HTD) caracteriza la textura de la región de imagen utilizando la energía media y la desviación estándar de energía de un conjunto de 30 canales de frecuencia modelados por las funciones de Gabor. El plano de frecuencia 2D se divide en 30 canales como se muestra en la Figura 1.12.

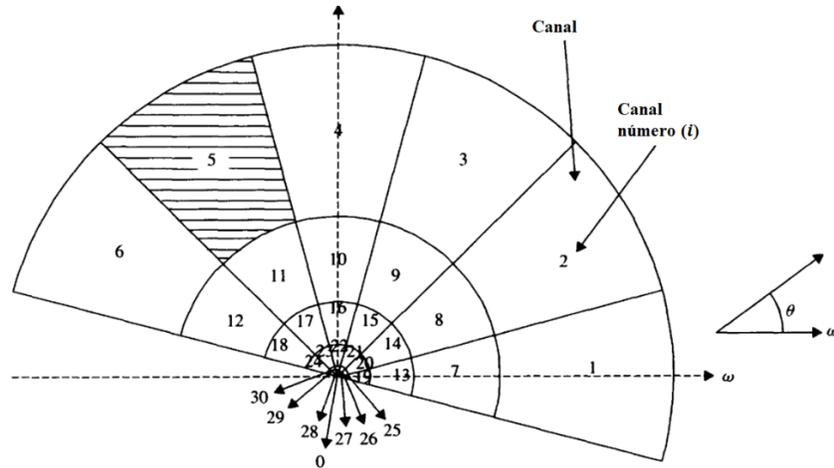


Figura 1.12 Canales utilizados en el cálculo de HTD (Choi, Won, Ro, & Manjunath, 2003).

La energía media y su desviación se calculan en cada uno de estos 30 canales de frecuencia. Se observa que la partición del plano de frecuencia es uniforme a lo largo de la dirección angular (igual tamaño de paso de 30 grados) pero no uniforme a lo largo de la dirección radial. Más específicamente, la división a lo largo de la dirección radial está en una escala de octava. (...). Hay alguna evidencia de que el procesamiento visual temprano en la corteza visual humana se puede modelar bien utilizando el diseño de frecuencia similar a la Figura 1.12.

La sintaxis de HTD es la siguiente:

$$HTD = [f_{DC}, f_{SD}, e_1, e_2, \dots, e_{30}, d_1, d_2, \dots, d_{30}]$$

Dónde  $f_{DC}$  y  $f_{SD}$  son la desviación media y estándar de la imagen, respectivamente, y  $e_i$  y  $d_i$  son la desviación de energía media, escalada y cuantificada de forma no lineal del  $i$ -ésimo canal correspondiente en la Figura 1.12.

- El descriptor de histograma de borde (EHD) representa la distribución local de bordes en la imagen. Específicamente, dividiendo el espacio de la imagen en  $4 \times 4$  sub-imágenes, la distribución de borde local para cada sub-imagen se puede representar mediante un histograma. Para generar el histograma, los bordes en

las sub-imágenes se clasifican en cinco tipos: vertical (a), horizontal (b),  $45^\circ$  diagonal (c),  $135^\circ$  diagonal (d) y bordes no direccionales (e) Figura 1.13.

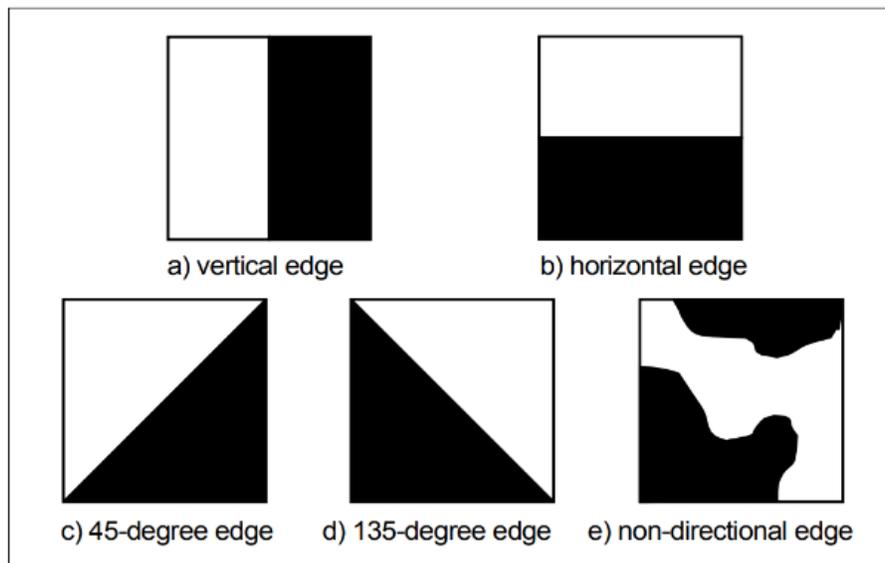


Figura 1.13 EHD considera 5 tipos de bordes (Won, Park, & Jeon, 2002).

Como se muestra en la Figura 1.14, la sub-imagen se define al dividir el espacio de la imagen en bloques  $4 \times 4$  no superpuestos. Por lo tanto, la partición de la imagen siempre produce 16 sub-imágenes del mismo tamaño, independientemente del tamaño de la imagen original (invariante a escala).

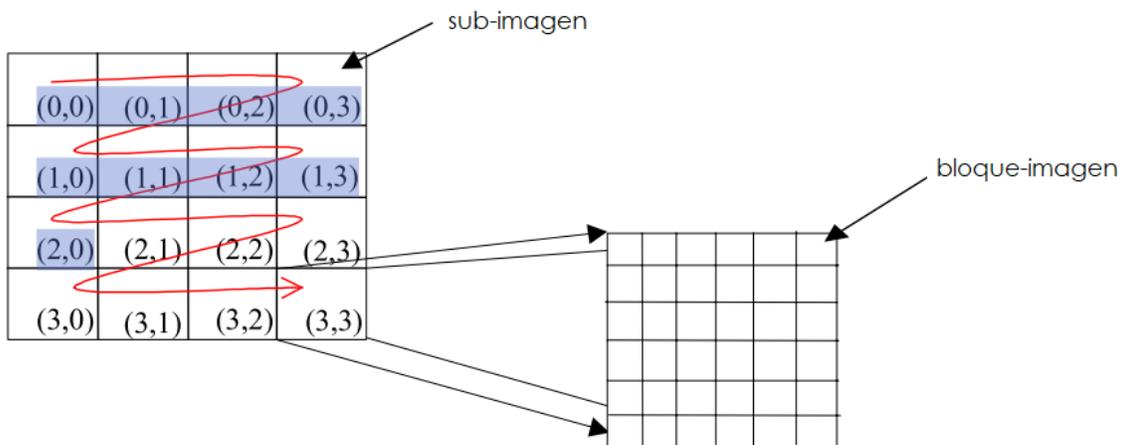


Figura 1.14 Definición de sub-imagen y bloque-imagen.

Dado que hay 16 sub-imágenes, se requiere un total de  $5 \times 16 = 80$  contenedores o *bins* de histograma. La Tabla 1.3 resume la semántica de los 80 contenedores.

Tabla 1.3 Semántica de los contenedores de histograma de borde local EHD.

Contenedor de histograma	Significado
<b>Borde_Local [0]</b>	Borde vertical sub-imagen (0,0)
<b>Borde_Local [1]</b>	Borde horizontal sub-imagen (0,0)
<b>Borde_Local [2]</b>	Borde 45 grados sub-imagen (0,0)
<b>Borde_Local [3]</b>	Borde 135 grados sub-imagen (0,0)
<b>Borde_Local [4]</b>	Borde no direccional sub-imagen (0,0)
<b>Borde_Local [5]</b>	Borde vertical sub-imagen (0,1)
⋮	⋮
<b>Borde_Local [74]</b>	Borde vertical sub-imagen (3,2)
<b>Borde_Local [75]</b>	Borde horizontal sub-imagen (3,3)
<b>Borde_Local [76]</b>	Borde 45 grados sub-imagen (3,3)
<b>Borde_Local [77]</b>	Borde 135 grados sub-imagen (3,3)
<b>Borde_Local [78]</b>	Borde no direccional sub-imagen (3,3)
<b>Borde_Local [79]</b>	Borde vertical sub-imagen (3,3)

Fuente: (Choi, Won, Ro, & Manjunath, 2003). Elaboración propia.

Por lo tanto, cada grupo de cinco valores del vector de características EHD representan el histograma de bordes de una sub-imagen. A continuación, se encuentra un ejemplo donde se muestran alternados en negritas y letra normal los grupos correspondientes de los histogramas locales de una imagen.

Descriptor de histograma de borde EHD:

**[0 0 0 0 0 0 0 0 0 0 1 0 5 0 4 4 2 5 4 6 0 0 0 0 0 1 1 5 0 5 3 1 5 1 5 1 1 1 1 2 3 4 1 3 3**  
**3 6 1 5 3 4 4 0 3 1 2 4 1 1 2 2 4 2 1 3 4 6 2 4 3 2 5 4 4 3 3 3 5 2 5 2].**

El EHD también se puede usar para recuperar o identificar imágenes a partir de un boceto o dibujo de ejemplo.

- El descriptor de forma con base en región (RSD), tiene en cuenta todos los píxeles que constituyen la forma [de los objetos en la imagen], es decir, los píxeles del límite y del interior. Por lo tanto, es aplicable a objetos que consisten en una sola región conectada o múltiples regiones, posiblemente con agujeros.
- El descriptor de forma con base en contorno (CShD) se basa en la representación de la curvatura de escala-espacio (*Curvature Scale-Space - CSS*) del contorno de los objetos presentes en la imagen. El descriptor describe de manera eficiente los objetos para los cuales las características de forma están contenidas en su contorno. Algunas características importantes de este descriptor son:
  - Emula bien la percepción de similitud de forma del sistema visual humano.

- Es robusto para deformaciones no rígidas significativas (por ejemplo, el contorno de una persona corriendo o escarabajo andante) y para transformaciones de perspectiva.
- Es compacto y el tamaño del descriptor se ajusta a la complejidad del contorno (14 bytes en promedio).

Una vez que los *descriptores visuales MPEG-7* se han extraído de las imágenes, éstos se pueden emplear para la búsqueda y recuperación en bancos de imágenes, mediante la comparación entre las características de la imagen de consulta y las contenidas en el banco, utilizando para ello las medidas de similitud correspondientes. Estos descriptores generalmente se utilizan en sistemas de recuperación de imágenes con base en contenido o sistemas CBIR (*Content-Based Image Retrieval*).

**Descriptores estadísticos** (Pajares & de la Cruz, 2008).

Estos descriptores son derivados del histograma de imagen. El histograma es una función discreta que representa el número de píxeles en la imagen en función de los niveles de intensidad  $g$ . La probabilidad  $P(g)$  de ocurrencia de un determinado nivel  $g$  se define en la Ec. ( 1.9 ).

$$P(g) = \frac{N(g)}{M} \quad (1.9)$$

$M$  es el número de píxeles de la imagen y  $N(g)$  es el número de píxeles en el nivel de intensidad  $g$ . Como en cualquier distribución de probabilidad, todos los valores de  $P(g)$  son menores o iguales que 1 y la suma de todos los valores  $P(g)$  es 1.

**Media**

Es el valor medio de los niveles de intensidad  $g$  y proporciona información sobre el “brillo” general de la imagen Ec. ( 1.10 )

$$\bar{g} = \sum_{g=0}^{L-1} gP(g) = \sum_i \sum_j \frac{I(i,j)}{M} \quad (1.10)$$

$L$  es el número total de niveles de gris ( $g$ ), así para una imagen con valores de gris entre 0 y 255;  $L$  sería 256. Una imagen brillante tendrá una media alta y viceversa.

**Varianza y desviación estándar**

Mide la dispersión de los niveles de intensidad alrededor de la media, la varianza se obtiene mediante la Ec. ( 1.11 ). La raíz cuadrada de la varianza  $\sigma^2$  define la desviación estándar  $\sigma$ .

$$\sigma^2 = \sum_{g=0}^{L-1} (\bar{g} - g)^2 P(g) \quad (1.11)$$

Una varianza o desviación estándar alta corresponde a una imagen con “contraste” alto y, al contrario, una varianza baja define una imagen con bajo contraste.

**Descriptor de suavidad relativa R**

La varianza  $\sigma^2$  se puede utilizar para obtener el descriptor de suavidad relativa “R”, el valor es 0 para áreas con intensidad constante ( $\sigma^2 = 0$ ) y es 1 para valores grandes de  $\sigma^2$ , áreas con varios niveles de intensidad, Ec. ( 1.12 ).

$$R = 1 - \frac{1}{1 + \sigma^2} = \begin{cases} 0 & \text{si } \sigma^2 = 0, \text{ Bajo contraste} \\ 1 & \text{si } \sigma^2 \rightarrow \infty, \text{ Alto contraste} \end{cases} \quad (1.12)$$

### Entropía

La entropía Ec. ( 1.13 ) es una forma de expresar el número de estados de un sistema. Un sistema con muchos estados tiene alta entropía, y un sistema con pocos estados tiene una baja entropía. Si se considera al número de niveles de gris ( $g$ ) de una imagen como el número de estados o información, entonces la entropía se puede considerar como un promedio de la cantidad de información contenida en la imagen, esto es, la entropía proporciona información sobre la distribución de los niveles de gris.

$$e = - \sum_{g=0}^{L-1} P(g) \log_2 [P(g)] \quad (1.13)$$

Cuanto mayor el número de niveles de gris mayor es la entropía, cuando se tiene un solo nivel de gris su valor es 0. Una imagen con mucho ruido o texturas complejas producirán un valor de entropía alto mientras que una imagen sin texturas y más uniforme tendrá una entropía baja.

### Momentos invariantes de Hu

La geometría de una región plana se basa en el tamaño, la posición, la orientación y la forma. Todas estas medidas están relacionadas en los parámetros denominados momentos.

Mediante los siete momentos invariantes de Hu (Hu, 1962), se consigue una descripción independiente a las posiciones, tamaños y ángulos del objeto.

Cuando se desea obtener un descriptor invariante respecto del tamaño y de la posición se puede utilizar los momentos de Hu centrales normalizados de orden dos y tres. Estos descriptores permiten realizar reconocimiento de formas y es posible determinar la similitud entre dos imágenes comparando sus correspondientes descriptores de Hu.

### 1.8.4 Métricas

¿Qué es medir? Medir es comparar una cantidad dada con respecto a otra, llamada unidad, tomada como patrón.

Para evaluar la eficacia de muchos procesos, se puede considerar tanto una evaluación objetiva como subjetiva. La métrica objetiva permite evaluar los resultados mediante una métrica analítica, mientras que los métodos subjetivos a menudo producen resultados imprevisibles (Pajares & de la Cruz, 2008).

Cuando se utiliza la representación de la imagen con base en su distribución espacial de píxeles, las medidas de similitud utilizan como patrón de referencia el *ground truth* de la imagen a segmentar, en la Figura 1.15 se observa una comparación de imágenes con una métrica de traslape. Superponiendo la imagen segmentada sobre el *ground truth*, una métrica basada en traslape indicaría el máximo de coincidencia si todos los puntos del contorno de las formas coinciden (Yeghiazaryana & Voiculescu, 2019).

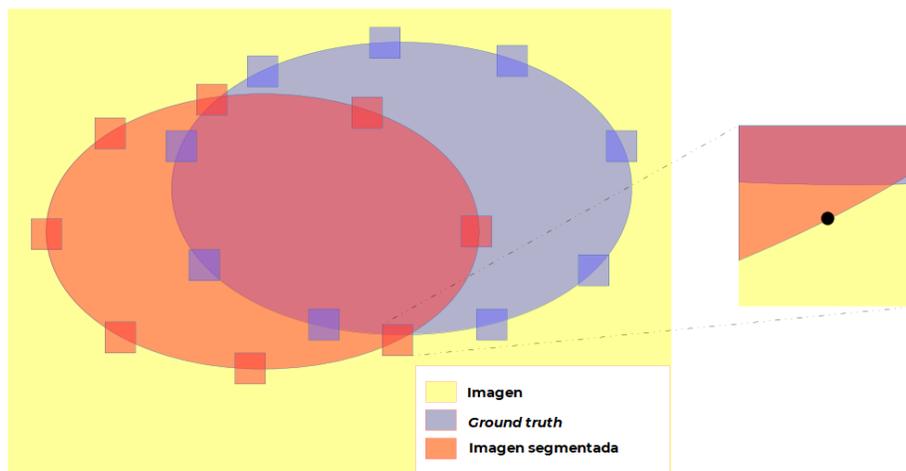


Figura 1.15 Comparación de similitud por traslape de imagen segmentada sobre el *ground truth* (Yeghiazaryana & Voiculescu, 2019).

Por otra parte, si la imagen está representada por sus atributos o características, la similitud entre imágenes se determina al comparar sus descriptores correspondientes, componente a componente. Para tal efecto se utilizan las medidas de distancia o disimilaridad.

Sea una matriz de datos  $X_{n \times p} = (x_{ic})$ , donde  $i$  representa a una característica y  $c$  una variable, una matriz de disimilaridades o distancias es una matriz  $D_{n \times n}$  tal que cada elemento es una disimilaridad  $d_{ij}$  que suele cumplir los tres primeros axiomas de una métrica, Ecs. ( 1.14), ( 1.15 )y ( 1.16 ):

Sea  $\{ d_{ii}, d_{ij}, d_{ji}, d_{ih}, d_{hj} \}$  un conjunto de distancias, entonces:

$$d_{ii} = 0 \quad (1.14)$$

$$d_{ij} \geq 0 \quad (1.15)$$

$$d_{ij} = d_{ji} \quad (1.16)$$

$$d_{ij} \leq d_{ih} + d_{hj} \quad (1.17)$$

Si se cumplieran los cuatro los axiomas anteriores la disimilaridad sería de hecho una medida de distancia y, en caso de cumplirse para todos los pares, se trataría de una matriz de distancias.

El tipo de disimilaridades o distancias que se pueden utilizar es muy variado y depende del tipo de variables y datos considerados. En las Ecs. ( 1.18 ) y ( 1.19 ) se muestran dos de las más utilizadas.

$$\text{Distancia Euclídea: } d_{ij} = \sqrt{\sum_{c=1}^p (x_{ic} - x_{jc})^2} \quad (1.18)$$

$$\text{Distancia Manhattan o City Block: } d_{ij} = \sum_{c=1}^p |x_{ic} - x_{jc}| \quad (1.19)$$

Por ejemplo, utilizando la Ec. ( 1.18 ) para calcular la distancia euclídea  $d_{ij}$  entre el punto  $i = (5,5)$  y  $j = (1,1)$  en la Figura 1.16, se obtiene el resultado  $d_{ij} = 5.65$ , Ec. ( 1.20 ).

$$d_{ij} = \sqrt{(5 - 1)^2 + (5 - 1)^2} = 5.65 \quad ( 1.20 )$$

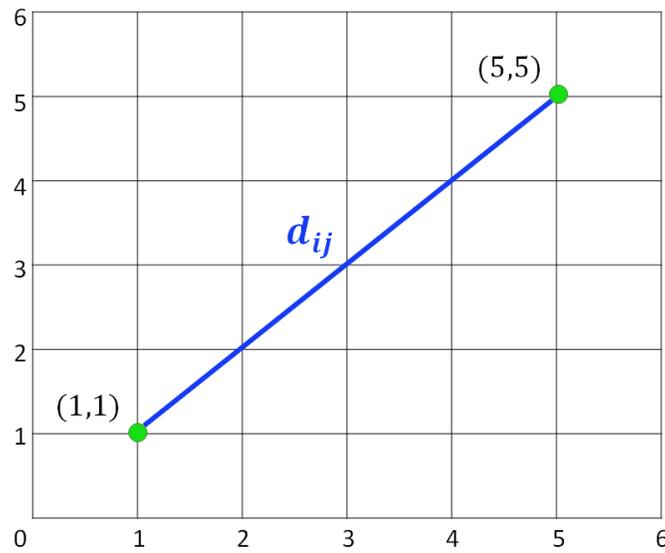


Figura 1.16 Cálculo de la distancia Euclídea  $d_{ij}$ .

### 1.8.5 Aprendizaje automático

El término aprendizaje automático se refiere a la detección automatizada de patrones significativos en los datos.

El objetivo del aprendizaje automático (*Machine Learning*), es crear programas de computadora (algoritmos) que puedan “aprender” de los datos disponibles para ellos. El aprendizaje automático procesa los datos convirtiéndolos en información [útil en la toma de decisiones] (Harrington, 2012).

En términos generales, el aprendizaje es el proceso de convertir la “experiencia” en conocimiento, Figura 1.17.



Figura 1.17 El aprendizaje como proceso de convertir la experiencia en conocimiento.

La entrada a un programa o algoritmo de aprendizaje son los datos de entrenamiento, los cuales representan la experiencia y, el resultado producido es cierto conocimiento, que generalmente toma la forma de otro programa de computadora (modelo) que puede realizar alguna tarea en particular.

Al buscar una comprensión formal de este concepto en (Shalev-Shwartz & Ben-David, 2014), se tiene que ser más explícito sobre lo que se quiere decir con cada uno de los términos involucrados:

1. ¿Cuáles son los datos de entrenamiento a los que accederán los programas?
2. ¿Cómo se puede automatizar el proceso de aprendizaje?
3. ¿Cómo se puede evaluar el éxito de dicho proceso, es decir, la calidad del resultado de un programa de aprendizaje?

De manera formal (Shalev-Shwartz & Ben-David, 2014) :

Sea  $\mathbb{X}$  un conjunto formado por instancias o tuplas  $x_i$  y sea  $\mathbb{Y}$  un conjunto de “etiquetas”  $y_j$ . También, sea  $S$  una secuencia de muestras o “datos de entrenamiento”, Ec. ( 1.21 ).

$$S = \{(x_1, y_1), \dots, (x_n, y_m)\} \quad S \subset \mathbb{X} \times \mathbb{Y} \quad (1.21)$$

$$\mathbb{X} = \{x_1, \dots, x_n\}; \quad \mathbb{Y} = \{y_1, \dots, y_m\};$$

Sea la función  $h$  una explicación de cómo  $\mathbb{X}$  se relaciona con  $\mathbb{Y}$ , esta función es la “hipótesis” o “modelo”, Ec. ( 1.22 ).

$$h: \mathbb{X} \rightarrow \mathbb{Y} \quad (1.22)$$

Entonces el aprendizaje se define como:

$$\exists h: \mathbb{X} \rightarrow \mathbb{Y}, \quad h(x_i) = y_j, \quad \forall (x_i, y_j) \in S \quad (1.23)$$

Es decir, existe una función  $h$  que asigna al vector  $x_i$  su etiqueta correspondiente  $y_j$ . Una característica importante que se desea en  $h$  es que la función aprendida, Ec. ( 1.23 ), sea robusta a la generalidad, es decir, el error de predicción de  $h$  sea mínimo.

Se requiere que la salida de un algoritmo de aprendizaje  $A$  genere una “regla de predicción” y esto se logra a través de la hipótesis  $h$ . A la función  $h$  también se le denomina “predictor” o “clasificador”.

En el ejemplo de la Figura 1.18, una posible hipótesis  $y = mx + b$  modela la frontera de separación de las muestras positivas y negativas.

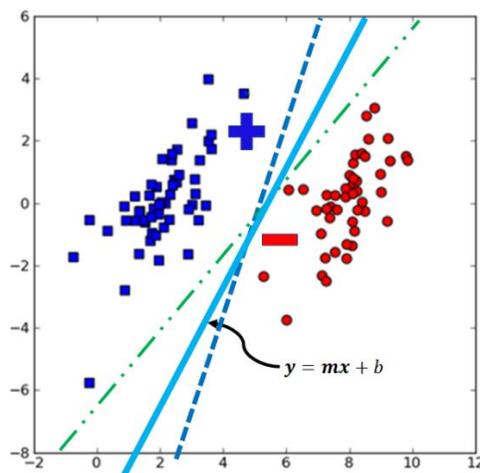


Figura 1.18 La hipótesis  $y$  separa los datos etiquetados como positivos (+) y negativos (-).

Los valores  $m$  y  $b$  son los parámetros que el modelo debe determinar o aprender a partir de las muestras de entrada ( $S$ ).

El problema del aprendizaje es que solo se conocen los ejemplos o muestras de entrada ( $S$ ), y se intenta encontrar una función  $h$  que permita realizar predicciones a partir de éstos, lo cual no siempre es posible por la misma naturaleza de los datos de muestra.

En general, hay tres tipos de aprendizaje automático con base en el tipo de problema y el conjunto de datos muestra de entrada: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semi-supervisado.

### **Aprendizaje supervisado**

El conjunto de entrenamiento dado para el aprendizaje supervisado es un conjunto de muestras  $S$  (datos etiquetados) como los definidos anteriormente. El aprendizaje supervisado trata de encontrar las relaciones entre el conjunto de características y el conjunto de etiquetas, esas relaciones son el conocimiento y propiedades que se puede aprender del conjunto de datos etiquetados.

Si a cada vector característico  $x$  le corresponde una etiqueta  $y \in \mathbb{Y}$  el problema de aprendizaje se trata como **clasificación** (dominio discreto).

Por otro lado, si a cada vector de características  $x$  le corresponde un valor real  $y \in \mathbb{R}$ , el problema de aprendizaje se define como un problema de **regresión** (dominio continuo).

El conocimiento extraído del aprendizaje supervisado a menudo se utiliza para la predicción y el reconocimiento.

En la Figura 1.19, se observa un conjunto de datos etiquetados, formando tres distintos grupos o clases (a).

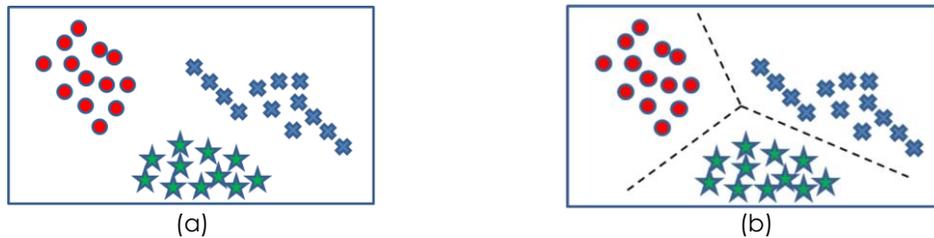


Figura 1.19 a) Conjunto de datos etiquetados con tres clases b) Frontera de separación de las clases después del aprendizaje (Chao, 2011).

Un algoritmo de aprendizaje supervisado procesa estos datos para aprender de su estructura y posteriormente define una frontera de separación de las clases (b), el modelo así generado, será capaz de clasificar nuevos objetos pertenecientes a cualquiera de las tres clases (Chao, 2011).

### **Aprendizaje no supervisado**

El conjunto de entrenamiento para aprendizaje no supervisado (también llamado agrupamiento o *clustering*) es un conjunto de datos muestra sin etiquetar.

El aprendizaje sin supervisión tiene como objetivo el agrupamiento, estimación de la densidad de probabilidad, encontrar la asociación entre características y reducción de dimensionalidad.

En general, un algoritmo no supervisado puede aprender simultáneamente más de una de las propiedades enumeradas anteriormente y los resultados del aprendizaje no supervisado podrían usarse para aplicarse algoritmos de aprendizaje supervisado.

En la Figura 1.20, se observa un conjunto de datos no etiquetados (a), los cuales al ser procesados por un algoritmo de aprendizaje no supervisado intenta descubrir su estructura y agrupar los objetos con características similares (b), cada elemento se etiqueta con base al grupo al que pertenece (Chao, 2011).

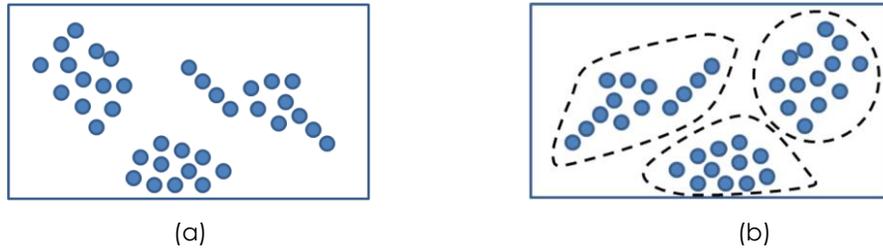


Figura 1.20 a) Conjunto de muestras sin etiquetas. (b) Después de aplicar el algoritmo de agrupamiento, se descubren tres grupos, delimitados por los contornos punteados (Chao, 2011).

### Aprendizaje semi-supervisado

En el aprendizaje semi-supervisado se emplea una combinación de datos etiquetado y no etiquetados, posiblemente se conoce parte de la información y no toda (tal vez se conozca el número de clases). Por lo regular se emplean pocos datos etiquetados y muchos datos no etiquetados como parte del conjunto de entrenamiento  $S$ .

Los algoritmos semi-supervisados tratan de explorar la estructura que contienen los datos no etiquetados con el objetivo de generar modelos predictivos que funcionen mejor que los que sólo utilizan datos etiquetados. Es una solución intermedia entre el aprendizaje supervisado y no supervisado.

En la Figura 1.21, se observa en (a) un conjunto de datos etiquetados y no etiquetados, en (b) se encuentran agrupados los objetos no etiquetados con base en el conocimiento de los objetos que sí están etiquetados y de características similares (Chao, 2011).

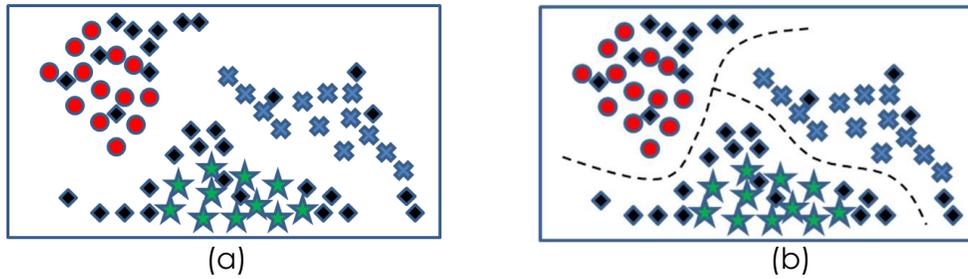


Figura 1.21 a) Conjunto de datos etiquetados (círculos, estrellas y cruces) y no etiquetados (rombos),  
 b) Clasificación después del proceso de aprendizaje (Chao, 2011).

Este trabajo se centró en aprendizaje supervisado aplicado para clasificación.

## Capítulo 2

### Revisión de la literatura

En esta sección se describen documentos científicos relacionados a los temas de: segmentación adaptativa, preprocesamiento y detección de bordes, descriptores y métricas.

#### 2.1 Antecedentes

Se revisaron los trabajos de investigación realizados con anterioridad en el CENIDET sobre los temas relacionados a: segmentación adaptativa, estructuración del conocimiento y descriptores de imagen.

#### **Sintonización de una Red Totalmente Conectada para Segmentación de Dos Clases de Objetos en Imágenes (Suárez, 2018).**

En esta tesis se propuso la implementación de una Red Neuronal Convolutiva con base en técnicas de aprendizaje profundo para segmentar imágenes a color sin ruido.

Se utilizaron los modelos de redes neuronales convolucionales: FCN-Alexnet y FCN-8s para realizar los experimentos, dando mejores resultados el FCN-8s, tanto cualitativa como cuantitativamente. Para evaluar la calidad de la segmentación se emplearon las métricas: Índice Probabilístico Rand (PRI), Variación de la Información (VI) y Error de Consistencia Global (GCE).

Se utilizó el banco de imágenes BSD500 del Grupo de Visión por Computadora (*Computer Vision Group, CVG*) de la Universidad de Berkeley como base para sus experimentos de segmentación.

Una de las aportaciones de esta tesis fue ofrecer un antecedente para aquellos trabajos que requieran entrenar una Red Neuronal Convolutiva a partir de un repositorio.

Se concluyó que las Redes Neuronales Convolutivas necesitan un repositorio con una cantidad considerable de imágenes por clase para ser capaces de aprender.

Se obtuvo el mejor resultado con la Red Neuronal Convolutiva FCN-8s, tomando las clases ave y persona del repositorio BSDS500, para entrenar, validar y probar el modelo mencionado.

### **Metodología para la estructuración y uso de conocimiento en segmentación de imágenes digitales (Cervantes, 2006)**

En esta tesis del CENIDET, se realizó una investigación para desarrollar una metodología para la segmentación de imágenes, donde el conocimiento relacionado se encontraba separado de los algoritmos necesarios para el procesamiento de la imagen.

Los sistemas de procesamiento digital de imágenes son desarrollados a la medida, por lo que dependen fuertemente del dominio del problema. Los problemas que normalmente se presentan al desarrollar los sistemas de procesamiento digital de imagen de la forma expuesta anteriormente son:

- a) Problemas para reutilizar el conocimiento.

- b) Se dificulta la reutilización de los algoritmos de procesamiento digital de imágenes.
- c) Dificultad para modificar el funcionamiento del sistema.

Debido a lo anterior, en este trabajo se propuso establecer una estructuración que proporciona una separación entre el conocimiento y los algoritmos de procesamiento digital de imágenes, de tal forma que se facilita la reutilización y el mantenimiento de este tipo de sistemas.

Con el fin de separar el conocimiento, se tomó como base la estructura de los sistemas basados en conocimiento, por lo que cabe resaltar que el desempeño del sistema resultante fue totalmente dependiente de la calidad del conocimiento que se tenía para resolver problemas. El modelo trabajó con el conocimiento explícito que proporcionaba el usuario. Por esto, el desarrollo se basó en la estructura que tienen los sistemas basados en conocimiento.

Con base en el estudio realizado durante la investigación, se eligieron dos formalismos: Los marcos y las reglas. Por lo que para representar el conocimiento se diseñó e implementó el lenguaje LBFR (Lenguaje con Base en Frames y Reglas).

En la Figura 2.1 se muestra el modelo conceptual de la arquitectura propuesta, donde se observa cómo se estableció una separación física entre la base de conocimiento y los algoritmos de segmentación.

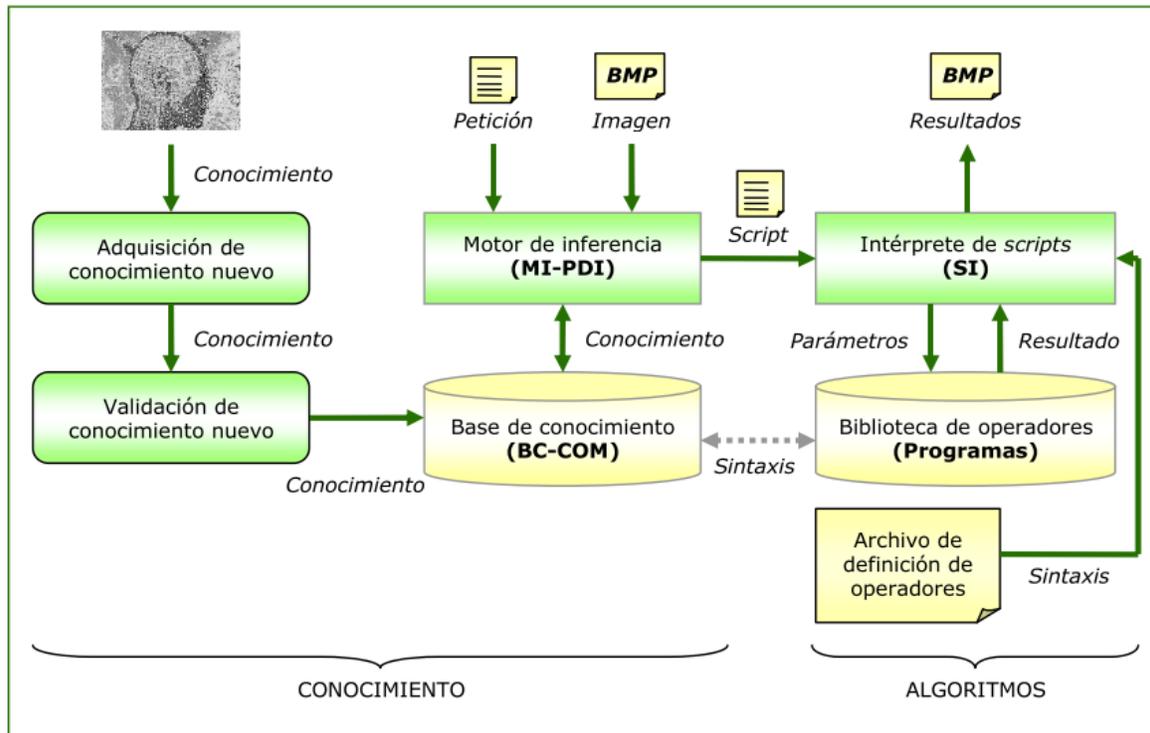


Figura 2.1 Modelo conceptual de la arquitectura propuesta (Cervantes, 2006).

Los algoritmos base para la detección de bordes fueron: Canny, Frei Chen, Hough, CHT para detección de círculos, EHT para la detección de elipses.

Finalmente, con el desarrollo de este trabajo se concluyó que el conocimiento *a priori* ayuda a mejorar el desempeño de los algoritmos de segmentación en un sistema de visión artificial, pero también es cierto que sólo mediante la separación de los algoritmos y el conocimiento, es posible reutilizar los sistemas de segmentación de forma eficaz, sin necesidad de interactuar con el código de los algoritmos.

### Indexado y Recuperación de Imágenes por Contenido (Troncoso, 2007)

En esta investigación se trabajó con contenido (imágenes) que no se puede describir por su contorno, ya que no tenía una forma definida, sino que sus

características principales estaban determinadas por su textura. Por lo que se seleccionaron técnicas de extracción de características para descriptores de región y descriptores de textura.

Se propuso un sistema de recuperación de imágenes por contenido o CBIR (*Content-Based Image Retrieval*). El sistema se soportó en la utilización de técnicas de: procesamiento digital de imágenes, de selección de variables, de reconocimiento de patrones y de indexado en bases de imágenes.

Se presentaron las razones por las cuales, la tarea de desarrollar sistemas CBIR no es simple. El sistema de recuperación propuesto implicó definir cómo realizar la descripción del contenido de las imágenes, la selección de descriptores que proporcione información suficiente para representar el contenido, definición de métricas de comparación entre el contenido de dos imágenes, incluyendo la flexibilidad para agregar elementos a la base de imágenes y cómo mantener una buena organización de la colección.

### **Recuperación automatizada de imágenes mediante la Implementación de descriptores del estándar MPEG-7 (Pérez, 2014)**

En este trabajo se desarrolló un sistema CBIR (*Content-Based Image Retrieval*) empleando *descriptores visuales* del estándar MPEG-7. En esta investigación se seleccionaron dos descriptores del estándar, uno de color (*Color Layout Descriptor*) y otro de textura (*Edge Histogram Descriptor*).

Para cada una de las pruebas realizadas se utilizaron tres conjuntos de imágenes especiales (*datasets*) utilizados para pruebas y mencionados en la literatura, *Common Color Dataset* (CCD), *Corel\_1k* y *Uncompressed Colour Image Dataset* (UCID); se utilizó adicionalmente uno más con imágenes capturadas en el Centro

Nacional de Investigación y Desarrollo Tecnológico (CENIDET) de México, nombrado *Dataset* en CENIDET (DEC). Los resultados de las recuperaciones de imágenes fueron medidos con la métrica ANMRR (*Averaged Normalized Modified Retrieval Rank*) tal índice fue desarrollado por el estándar MPEG-7.

En las pruebas del sistema se hicieron experimentos para la detección de parámetros óptimos de tales descriptores; en el caso de *Edge Histogram Descriptor* (EHD) se determinó un umbral óptimo que mejora en 4.57 % al propuesto en MPEG-7; en el caso del *Color Layout Descriptor* (CLD), los experimentos arrojaron una combinación de coeficientes que superan al MPEG-7 en 13.21 % en cuanto a calidad de recuperación de imágenes, empleando apenas una tercera parte más de coeficientes con relación a los sugeridos por el estándar.

También, en el presente trabajo se propuso el uso de cinco nuevas medidas de distancias con el fin de mejorar la recuperación de las imágenes empleando el descriptor CLD, y cuyos resultados son comparables con las medidas propuestas por el estándar, pero son más sencillas de implementar.

Se comparó el rendimiento del sistema CBIR de (Troncoso, 2007) contra la implementación basada en MPEG-7 y se obtuvieron resultados bastante parecidos al usar el descriptor CLD, con la diferencia de que el CBIR basado en MPEG-7 es ajustado a un estándar internacional.

Para el desarrollo de todos los experimentos anteriores se desarrolló un software para recuperación de imágenes usando los descriptores seleccionados para esta investigación llamado RIPE (Recuperación de Imágenes Por Ejemplo) y que tiene la ventaja de estar diseñado para aceptar en el futuro más descriptores y, al ser un sistema abierto, permite a futuros investigadores experimentar de manera más

---

sencilla y directa con nuevas ideas de descriptores o métricas de evaluación relacionadas con la recuperación de imágenes por contenido.

En trabajos revisados, se encontró una propuesta de segmentación adaptativa de imágenes para dos clases (Suárez, 2018), si bien el tipo de segmentación empleado se basa en regiones de imágenes a color, proporciona un buen ejemplo de la adaptabilidad de un modelo, en este caso, soportado por redes neuronales convolucionales. En (Cervantes, 2006) se encontró una serie de aportaciones en conceptos e ideas sobre la estructuración del conocimiento separado de las técnicas de procesamiento de imagen. De (Troncoso, 2007) y (Pérez, 2014) se obtuvo el aporte de caracterización de imágenes utilizando descriptores, entre ellos los *descriptores visuales* del estándar MPEG7. Estas aportaciones fueron útiles en la conceptualización de la propuesta de solución del problema planteado.

## **2.2 Estado del arte**

Se investigó en el estado del arte, buscando propuestas de solución expuestas hasta el momento, para resolver el problema planteado o parte de este, así como las estrategias utilizadas. Se encontraron algunas alternativas, las cuales se describen a continuación, agrupadas por temas relacionados a: segmentación adaptativa, preprocesamiento y detección de bordes, descriptores y métricas.

### **2.2.1 Segmentación adaptativa**

Se han propuesto algunas técnicas para resolver el problema de segmentación de manera adaptativa, las cuales difieren en el método de solución y el área de aplicación.

#### **Segmentación de objetos estructurados en una imagen (Ajeshwari, 2015)**

La detección de objetos estructurados en el primer plano de las imágenes es una tarea esencial en muchas aplicaciones de procesamiento de imágenes. Este artículo presenta un enfoque de fusión y crecimiento de regiones para la detección automática de los objetos de primer plano en la imagen (segmentación).

El enfoque propuesto identifica los objetos en la imagen, basándose en las propiedades generales de los objetos sin depender del conocimiento previo sobre objetos específicos. La información de contraste de la región se utiliza para separar las regiones de los objetos estructurados de las regiones de fondo.

Las leyes de organización perceptiva se utilizan en el proceso de fusión de regiones para agrupar las distintas regiones, es decir, partes del objeto. El sistema es adaptable al contenido de la imagen. Los resultados de los experimentos muestran que el esquema propuesto puede extraer de manera eficiente los bordes de los objetos para separarlos del fondo.

### **Segmentación adaptativa con base en una métrica de calidad aprendida (Frosio & Ratner, 2015)**

Los autores presentaron un modelo para la evaluación de la calidad de la segmentación de una imagen en color.

Los parámetros del modelo se aprendieron de un conjunto de imágenes de ejemplo. Para este objetivo, primero segmentaron un conjunto de imágenes usando un algoritmo tradicional de corte de grafo, para diferentes valores del parámetro de escala. Un observador humano clasificó estas imágenes en tres clases: sub segmentada, bien segmentada y sobre segmentada. Esta clasificación se empleó para aprender los parámetros del modelo de calidad de segmentación. Esto se utilizó para optimizar automáticamente el parámetro de escala del algoritmo de segmentación de corte de grafo, incluso a una escala local.

El modelo propuesto proporciona una heurística razonable que puede usarse para mejorar y automatizar efectivamente los algoritmos de segmentación existentes. Los resultados experimentales mostraron una calidad de segmentación mejorada para el algoritmo adaptativo con base en su modelo de calidad de segmentación.

### **Un método de detección de bordes basado en el algoritmo adaptativo de Canny y el umbral de segmentación iterativa (Qiang, Guoying, Jingqi, & Hongmei, 2016)**

Debido a las características de alta precisión de posicionamiento, detección precisa y resistencia al ruido, el algoritmo Canny ha sido ampliamente utilizado en la detección de grietas en superficies. En la tecnología de detección de grietas, el algoritmo tradicional de Canny utiliza un coeficiente de escala espacial fija ( $\sigma$  – *varianza*) del filtro de Gauss y valores empíricos de los umbrales alto y bajo, y tiene una auto adaptabilidad poco funcional porque no puede ajustar los parámetros de

acuerdo con la imagen real. En este artículo se propone un método basado en un algoritmo adaptativo de Canny y un algoritmo de segmentación de umbral iterativo para detectar grietas en superficies.

En primer lugar, en el proceso de suavizado de la imagen, el algoritmo adaptativo de Canny calcula automáticamente la diferencia entre el valor de gris del pixel actual y el valor de gris promedio de la imagen en la ventana del filtro, y la diferencia se establece como el coeficiente de escala espacial para el filtro de Gauss del pixel actual.

En segundo lugar, el método Otsu se aplica para calcular el histograma del gradiente de la imagen para obtener umbrales altos y bajos y esos nuevos valores se aplican para detectar el borde de la grieta.

Finalmente, con base en el algoritmo de segmentación de umbral iterativo, la imagen con los bordes de la grieta se binarizan y los puntos de ruptura en la grieta y las cavidades dentro de la misma se eliminan por dilatación morfológica. Los resultados experimentales muestran que el nuevo método de detección de grietas puede retener los bordes y obtener un mejor efecto sobre la cancelación de ruido. También puede reducir las detecciones omitidas o falsas, con lo anterior se pueden lograr mejores resultados en la detección de grietas.

### **Algoritmo de segmentación de imagen adaptativo bajo la restricción de la probabilidad posterior de borde (Meiling Gong, 2017)**

Los algoritmos actuales de segmentación tienen problemas tales como: pobre desempeño contra el ruido, sobre segmentación y resultados imprecisos. Para resolver estos problemas los autores propusieron un algoritmo adaptativo de segmentación de imágenes bajo la restricción de la probabilidad posterior de borde.

Este algoritmo primero resuelve el problema de la sobre segmentación mejorando el algoritmo *watershed* (de cuenca o de línea divisoria) existente. Después el algoritmo automáticamente decide si adopta la segmentación de umbral de borde resultante del algoritmo *watershed*, con base en el modelo propuesto.

El método que se propuso fue para resolver problemas como: bajo rendimiento anti-ruido, baja precisión de segmentación y sobre segmentación presente en los algoritmos de segmentación actuales.

En primer lugar, un intervalo de umbral fue introducido con el fin de superar el problema de la sobre segmentación del algoritmo *watershed*. Entonces, se estableció un modelo de probabilidad posterior de borde (EPPM) cuya idea vino del Modelo Oculto de *Markov* (HMM) para probar el resultado de la segmentación de algoritmo *watershed* mejorado.

Si el resultado se ajusta al requisito que se estableció, se podrían generar resultados de segmentación directamente. De lo contrario, todavía se necesita una segmentación de umbral más precisa basado en la información de borde que proviene del resultado del algoritmo *watershed* mejorado. El diagrama de flujo del algoritmo propuesto se muestra en la Figura 2.2 .

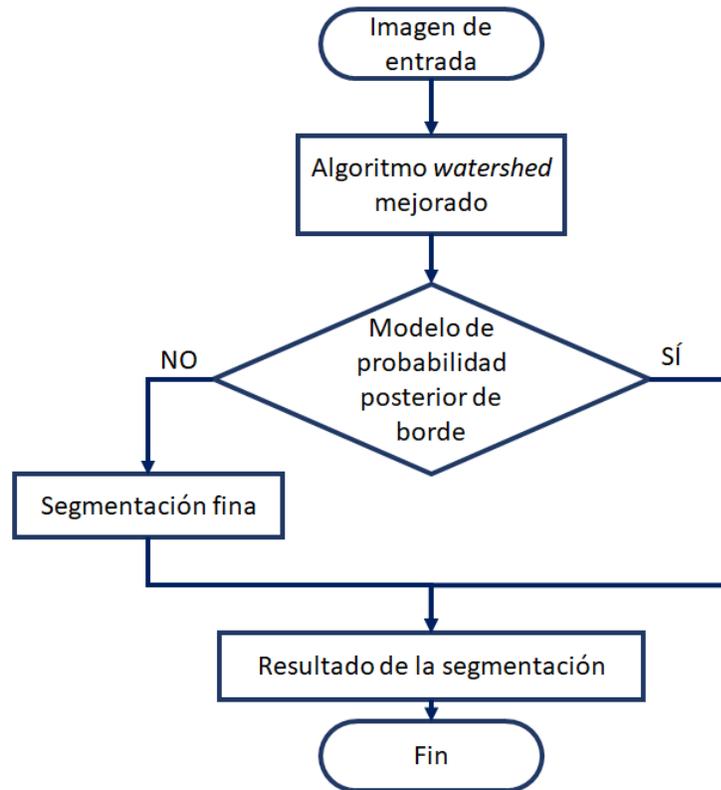


Figura 2.2 Diagrama de flujo del algoritmo propuesto (Meiling Gong, 2017).

### 2.2.2 Preprocesamiento y detección de bordes

#### Una mejora adaptativa de imagen mediante el filtrado de *Wiener* con compresión y segmentación (Raajan, Muthuselvi, & Agnes, 2015)

Este trabajo propuso un método de segmentación de imágenes y técnicas de eliminación de ruido con el filtro *Wiener*, también conocido como filtro lineal de preservación de bordes.

En primer lugar, la imagen se filtró con *Wiener* (preprocesamiento), el siguiente paso fue comprimir la imagen filtrada usando *wavelets-Haar*. La compresión de imagen basada en *wavelets* utilizó una técnica de umbral para eliminar todos los valores insignificantes, es decir suprimir atributos redundantes de la imagen.

Posteriormente, se aplicó el gradiente morfológico a la imagen comprimida, se utilizó para superar el problema de segmentación excesiva. El gradiente morfológico se aplicó a la imagen en escala de grises.

Por último, se aplicó el algoritmo de segmentación *watershed*, este algoritmo requiere algunos puntos de inicio o marcadores para segmentar la imagen. El procedimiento para obtener los marcadores de los objetos de interés (los cuales deben ser bloques de píxeles conectados dentro de cada uno de los objetos) se realizó aplicando técnicas morfológicas de apertura y cierre por reconstrucción.

Estas operaciones crean máximos dentro de cada objeto que se pueden ubicar usando la apertura de imagen. Se calcula un máximo de la región a partir de la imagen reconstruida para obtener los marcadores de los objetos.

Los resultados muestran que el método propuesto reduce efectivamente el efecto de sobre segmentación y logra resultados de segmentación más precisos que los métodos existentes.

### **Segmentación de objetos de múltiples formas superpuestos en una imagen (Kumar Abhinav, 2017)**

En este trabajo, se propuso un nuevo algoritmo de segmentación para imágenes que contienen objetos convexos presentes en múltiples formas con un alto grado de superposición.

El algoritmo propuesto se realiza en dos pasos, primero se identifican los contornos visibles, se segmentan utilizando puntos cóncavos y finalmente se agrupan los segmentos que pertenecen al mismo objeto.

El siguiente paso es asignar una identidad de forma a estos segmentos de contorno agrupados. Para imágenes que contienen objetos en formas múltiples, se comienza primero identificando las clases de formas de los contornos y luego asignando una entidad de forma a estas clases.

Se proporciona una experimentación completa del algoritmo en dos conjuntos de datos de imágenes de cristales. Un conjunto de datos se compone de imágenes que contienen objetos en múltiples formas que se superponen entre sí y el otro conjunto de datos contiene imágenes estándar con objetos presentes en una sola forma.

### **Transformada difusa ( $F$ – *transform*) (Perfilieva, 2015)**

En ese artículo se proponen algunas aplicaciones de la transformada difusa ( $F$ ) para la compresión, reconstrucción y fusión de imágenes. Por otra parte, la transformada difusa grado 1 ( $F^1$  – *transform*) se propone para implementar un detector de bordes, aplicando un procesamiento similar al algoritmo de *Canny* (filtrado, detección de bordes, adelgazado de bordes y cerrado de contornos), donde el filtrado (suavizado) y la detección de bordes se sustituyen por la aplicación de la transformada  $F^1$ .

### **2.2.3 Descriptores**

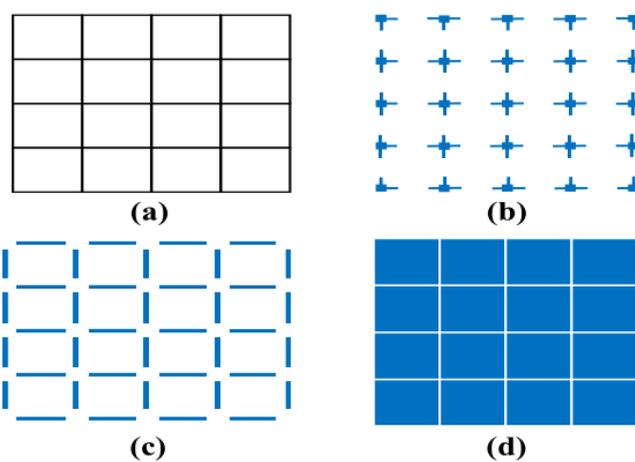
#### **Avances recientes en algoritmos de extracción y descripción de características: un estudio exhaustivo (Salahat & Qasaimeh, 2017)**

Este estudio proporciona una visión general del estado del arte y los avances recientes en la detección de características y algoritmos de descripción. Específicamente, comienza con una visión general de los conceptos fundamentales. Luego compara, informa y analiza su rendimiento y capacidades. El algoritmo de las regiones extremas de máxima estabilidad (MSER, por sus siglas en inglés) y los algoritmos de transformación de la característica invariable de la escala (SIFT, por

sus siglas en inglés), que son dos de los mejores de su tipo, se seleccionaron para informar sus derivados algorítmicos recientes.

Las características locales de la imagen (conocidas como puntos de interés, puntos clave o características sobresalientes) se pueden definir como un patrón específico que se distingue de sus píxeles inmediatamente cercanos, que generalmente se asocia con una o más de las propiedades de la imagen.

Tales propiedades incluyen bordes, esquinas y regiones, entre otros. En la Figura 2.3 se presenta un resumen de dichas características locales.



**Figura 2.3** Características locales ilustrativas de la imagen (a) imagen de entrada, (b) esquinas, (c) bordes y (d) regiones (Salahat & Qasaimah, 2017).

De hecho, estas características locales representan puntos de anclaje esenciales que pueden resumir el contenido del cuadro de imagen (con la ayuda de descriptores de características sobre los puntos de interés) mientras se busca en una imagen o en un video.

Estas características locales se convierten luego en descriptores numéricos, que representan un resumen único y compacto de estas características locales.

**ImageNet Descriptores visuales MPEG-7 (Rayar, 2017)**

En este informe técnico se presenta la aplicación de algunos de los *descriptores visuales MPEG-7* sobre imágenes contenidas en la base de datos ImageNet. Estos descriptores se han hecho públicos en un esfuerzo dirigido hacia la investigación abierta.

ImageNet es una base de datos ontológica de imágenes a gran escala. Los investigadores de este proyecto, a saber, Pr. Li Fei-Fei y sus colegas de la Universidad de Princeton, comenzaron sus esfuerzos en 2009. Este recurso puede ser útil en aplicaciones de reconocimiento visual tales como: Reconocimiento de objetos, clasificación de imágenes o localización de objetos.

En este trabajo, el objetivo es proporcionar a la comunidad científica dos *descriptores visuales MPEG-7* para el conjunto de imágenes ImageNet. Se seleccionaron dos de las cuatro categorías del MPEG 7, descriptores de color y descriptores de textura. El conjunto de imágenes proporcionado con sus respectivos descriptores suma un total de 21,841.

El descriptor de color o CLD (*Colour Layout Descriptor*) está diseñado para capturar la distribución espacial de colores dominantes en una imagen. Los colores se expresan en el espacio de color YCbCr, Figura 2.4. Las principales ventajas de CLD son: primero que es un descriptor muy compacto, por lo que se ajusta perfectamente a aplicaciones de navegación rápida y de búsqueda. En segundo lugar, es invariante a la resolución.

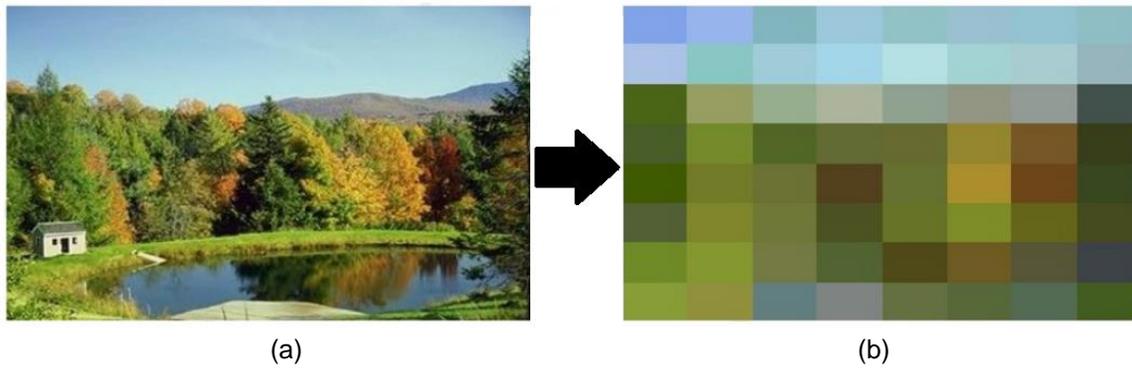


Figura 2.4 CLD: detección de colores representativos en una imagen particionada en bloques de 8x8. a) Imagen original b) Imagen particionada por colores (Rayar, 2017).

El descriptor de textura EHD (*Edge Histogram Descriptor*) captura la distribución espacial de cinco tipos de bordes en una imagen. En la Figura 2.5 se muestran los cinco bordes considerados (a) y la extracción del histograma en una celda de imagen (b).

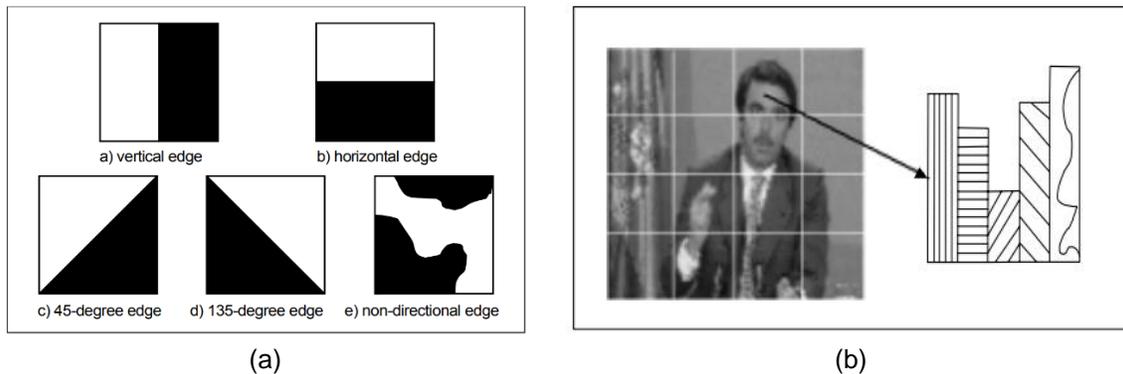


Figura 2.5 EHD: Cálculo de histograma para cada celda de 4x4 en la imagen, a) 5 tipos de bordes considerados b) Histograma de bordes de la celda de imagen (Won, Park, & Jeon, 2002)

**Segmentación de imagen por combinación de propiedades globales y locales (Wang, 2017)**

En este artículo, se propone segmentar la imagen combinando sus propiedades globales y locales.

Las propiedades globales de la imagen se caracterizan por los valores medios de diferentes clases de píxeles y el límite continuo del objeto o región.

Las propiedades locales se caracterizan por las interacciones de los píxeles vecinos y el borde de la imagen.

El enfoque propuesto consta de cuatro partes básicas correspondientes a la propiedad global o local de la imagen, respectivamente.

1. La distribución de la diferencia de pendiente que se utiliza para calcular los valores medios globales de diferentes clases de píxeles.
2. Minimización de energía para eliminar la homogeneidad basada en la distribución de Gibbs que cumple con las interacciones locales de los píxeles vecinos.
3. El operador *Canny* que se utiliza para detectar el borde local del objeto o la región.
4. La curva *spline* polinomial que se utiliza para suavizar el límite del objeto o la región.

Estas cuatro partes básicas se aplican una por una y cada una de ellas es indispensable para lograr una alta precisión. Se utilizó una gran variedad de imágenes para validar el enfoque propuesto y los resultados fueron favorables.

#### **2.2.4 Métricas**

##### **De los contornos al *ground truth*: cómo evaluar los detectores de bordes mediante el filtrado (Hasan Abdulrahman, 2017)**

La detección de bordes sigue siendo una etapa crucial en numerosas aplicaciones de procesamiento de imágenes. Por lo tanto, una técnica de detección de bordes debe evaluarse antes de usarla en una tarea de visión por computadora.

Como las evaluaciones de disimilitud dependen en gran medida de un mapa de borde de *ground truth*, un dato inexacto en términos de localización podría beneficiar a detectores de borde imprecisos y/o favorecer una medida de evaluación de disimilitud inadecuada. Por lo tanto, en este trabajo, se muestra cómo etiquetar los datos *ground truth* de forma semiautomática. Además, varias evaluaciones de detección de bordes con base en imagen de referencia se detallan y se aplican a una evaluación objetiva.

Por lo tanto, cada medida se compara variando el umbral de los bordes delgados. De hecho, teóricamente, la puntuación mínima de la medida corresponde al mejor mapa de borde, en comparación con el *ground truth*. Finalmente, los experimentos en varias imágenes utilizando seis detectores de borde muestran que la nueva base de datos *ground truth* permite una comparación objetiva de numerosas medidas de disimilitud.

Se proporciona una comparativa entre varias medidas de similitud aplicadas a algunos filtros de detección de bordes como *Sobel* y *Canny*.

Los experimentos muestran la importancia de la elección del mapa *ground truth*: un mapa de contorno *ground truth* inexacto en términos de localización penaliza a detectores de bordes precisos y/o da ventajas a los algoritmos de menor precisión.

### **Medición de similitud de imagen con base en el contexto de forma (Qian, 2015)**

Las métricas de similitud con base en la intensidad de la imagen son comunes, pero tienen un rendimiento limitado dentro de un tipo de distorsión. Las métricas con base en geometría miden la similitud con mayor precisión y en todos los tipos de distorsión, pero incurren en un mayor costo computacional. De acuerdo con la percepción humana, el sistema visual humano es muy sensible a variaciones de

forma de algunas regiones visuales sobresalientes cuando se comparan dos imágenes.

Para medir la similitud entre dos imágenes, se debería prestar más atención a la preservación de la forma de las regiones visuales sobresalientes.

Por lo tanto, se puede tener en cuenta a las estructuras geométricas de los objetos al medir la similitud de la imagen de la misma manera que en las métricas de similitud con base en la geometría, pero no se tiene la necesidad de establecer las correspondencias entre pares de píxeles, ya que la medición se centra en las regiones visuales sobresalientes.

Como resultado favorable, se espera que el menor costo computacional y la mejor precisión se logren. A partir de la idea anterior, este artículo propone un enfoque novedoso para medir la similitud de imagen con base en el contexto de la forma.

El contexto de forma es un descriptor más robusto y compacto, pero altamente discriminante.

Se emplea el contexto de la forma para describir la forma de la región visual sobresaliente. Se detectan las deformaciones de la forma de las regiones sobresalientes entre dos imágenes a través de la estimación de las distancias del contexto de la forma y, en consecuencia, se calculan los valores de similitud de la imagen.

Los resultados experimentales muestran que los rangos de similitud alcanzados para las imágenes reales coinciden estrechamente con las elecciones humanas subjetivas y son consistentes.

### **Evaluación de la segmentación de imágenes utilizando métodos objetivos (Surya & Satheesh, 2016)**

En este documento, se analizaron distintas métricas para la evaluación de la segmentación obtenida al aplicar diferentes algoritmos, con base en métodos objetivos.

El método de evaluación tradicionalmente subjetivo se utiliza para determinar la precisión del rendimiento de la segmentación. Como este método de evaluación es cualitativo y sesgado, se exige un método de evaluación cuantitativo. Esto se hace utilizando el método objetivo de evaluación en el que se usan los métodos de discrepancia y de bondad.

Los métodos de discrepancia se usan en general sobre imágenes con referencia predefinidas, es decir que se tiene la imagen *ground truth* correspondiente para la comparación.

Los métodos de discrepancia se clasifican en términos generales en tres grupos : según la medida de similitud, la medida de distancia y la medida estándar como se muestra en la Figura 2.6. Se considera que este método produce un resultado de evaluación con mayor precisión.

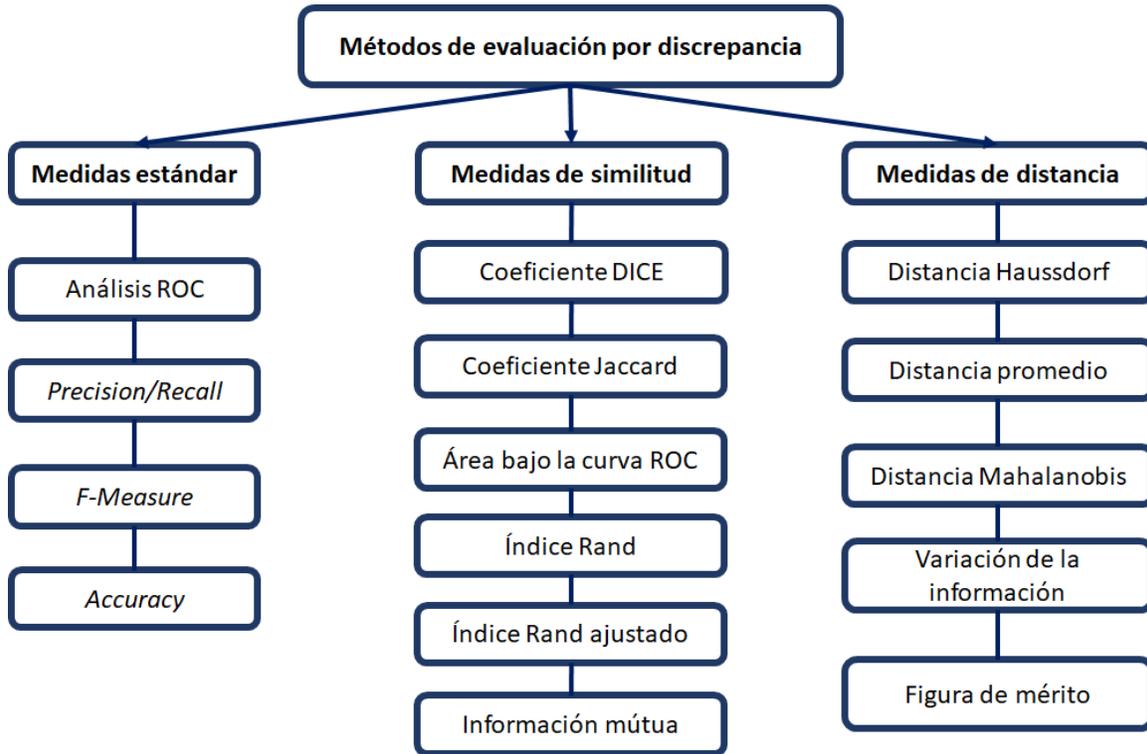


Figura 2.6 Clasificación de los métodos de discrepancia (Surya & Satheesh, 2016).

Los métodos de bondad se proponen para evaluar la segmentación cuando no se tiene ninguna imagen *ground truth* disponible para comparación. Estos métodos son útiles para situaciones en las que no es posible recopilar las imágenes *ground truth*.

Las propiedades de las imágenes como: forma, región, color, textura, varianza, uniformidad y entropía se utilizan como factores clave para analizar el rendimiento de los métodos de segmentación. Sobre la base de estas propiedades, las medidas de bondad se clasifican en diferentes métodos como se observa en la Figura 2.7.

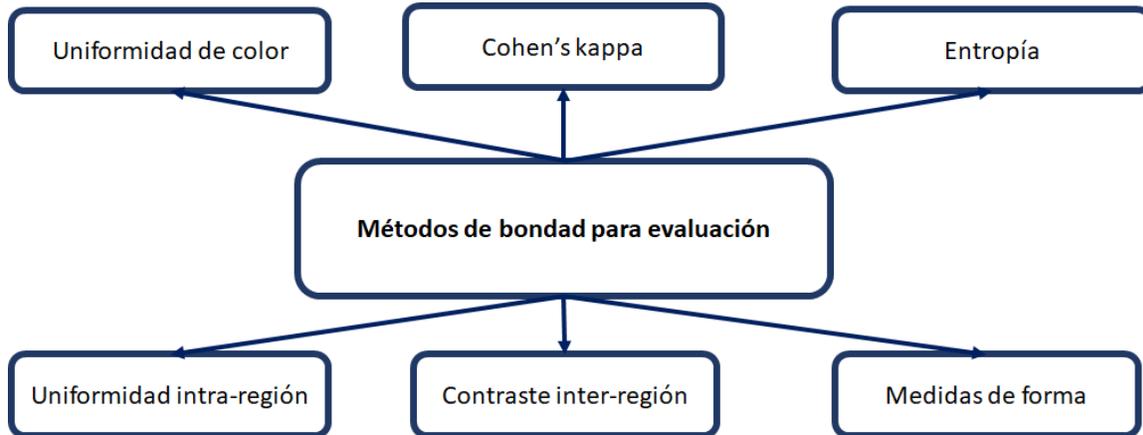


Figura 2.7 Diferentes métodos de medidas de bondad (Surya & Satheesh, 2016).

Estos métodos de evaluación objetiva son necesarios para validar los métodos de segmentación que se han incrementado rápidamente en los últimos años.

Se realizó un estudio y experimentación detallada de los diferentes métodos de evaluación a través de diferentes métodos de segmentación. Se utilizaron técnicas con base en bordes: Sobel (Sobel & Feldman, 1968), Canny (Canny, 1986) y Susan (Smith & Brady, 1997); técnicas con base en regiones: crecimiento de regiones y de umbral; por otra parte, se propuso una técnica híbrida que combina el gradiente de color y con un algoritmo de crecimiento de regiones.

El resultado experimental mostró que el método de segmentación híbrido funciona mejor que otros métodos existentes y también destacó la importancia del método de evaluación de la calidad de la imagen para identificar una mejor técnica de segmentación de acuerdo con el tipo de imágenes a procesar.

Una familia de métricas de superposición de bordes para la evaluación de la segmentación de imágenes médicas (Yeghiazaryana & Voiculescu, 2019)

Todos los algoritmos de segmentación de imágenes médicas deben validarse y compararse, sin embargo, ningún marco de evaluación es ampliamente aceptado dentro de la comunidad de imágenes. Ninguna de las métricas de evaluación que son populares en la literatura son consistentes en la forma en que clasifican los resultados de segmentación: tienden a ser sensibles a uno u otro tipo de error de segmentación (tamaño, ubicación, forma) pero ninguna métrica individual cubre todos los tipos de error.

Se presentó en este artículo una nueva familia de métricas, con características híbridas. Estas métricas cuantifican la similitud o diferencia de las regiones segmentadas al considerar su superposición promedio en vecindades de puntos de tamaño fijo en los límites de esas regiones. Estas métricas son más sensibles a las combinaciones de tipos de error de segmentación que otras métricas existentes en la literatura. Compararon el rendimiento métrico en colecciones de resultados de segmentación obtenidos de datos sintéticos 2D cuidadosamente compilados e imágenes médicas 3D.

Mostró que sus métricas:

1. Penalizan los errores con éxito, especialmente aquellos alrededor de los límites de la región.
2. Proporcionan una puntuación baja de similitud cuando las métricas existentes no están de acuerdo, evitando así puntuaciones demasiado infladas.
3. Puntuación de resultados de segmentación en un rango más amplio de valores. Analizaron una métrica representativa de esta familia y el efecto de su parámetro libre sobre la sensibilidad al error y el tiempo de ejecución.

Los resultados mostraron cómo, en contraste con las métricas existentes, SBD (*Symmetric Boundary Dice*) reacciona a una gama más amplia de tipos de error. Penaliza más los errores, produciendo una mayor variedad de puntajes de similitud (menor para más errores). SBD no incurre en la inflación del puntaje, especialmente en los casos en que las métricas existentes no están de acuerdo (una métrica da una puntuación alta y otra da una baja al mismo resultado de segmentación).

Los experimentos sobre imágenes fractales mostraron que SBD es apto para evaluar resultados de segmentación que son propensos a errores en la vecindad de los límites de la región. Otros miembros de la familia de superposición de bordes tienen propiedades similares a SBD.

En el futuro, se considerarán más clasificaciones de segmentación producidas con otra evaluación métrica; se comprobará si las clasificaciones SBD (*Symmetric Boundary Dice*) se correlacionan mejor que otras métricas con las clasificaciones producidas por humanos.

Para que la familia de métricas propuesta sea adoptada para uso general, deberán demostrar su efectividad al realizar estudios sistemáticos de grandes conjuntos de datos médicos disponibles públicamente. Mientras tanto, propusieron SBD como la mejor y más fácil de usar de las 15 métricas de la familia, la facilidad de su implementación, rapidez de ejecución y cumple con todos los criterios definidos para una buena métrica.

### **2.2.5 Discusión del estado del arte**

Los trabajos revisados se centraron en propuestas de segmentación autónoma o adaptativa, técnicas de preprocesamiento y detección de bordes, detectores de características de imagen (descriptores) y métricas para evaluar la calidad de la segmentación.

En artículos encontrados el enfoque está centrado en desarrollar técnicas de segmentación para un tipo específico de imágenes y no de manera general.

Se encontraron distintas propuestas de segmentación adaptativa, en (Ajeshwari, 2015) se propuso utilizar la técnica de fusión y crecimiento de regiones para segmentar las imágenes utilizando la información de contraste de la región de los objetos de interés para separarlos de las regiones del fondo. En (Frosio & Ratner, 2015) se propone un modelo para segmentar una imagen en color con la técnica de corte de grafo el cual es optimizado automáticamente una vez que el modelo ha aprendido de imágenes de ejemplo previamente clasificadas.

En (Qiang, Guoying, Jingqi, & Hongmei, 2016) se presenta un método de segmentación con base en una versión adaptativa del algoritmo de Canny (detección de bordes) y un algoritmo de segmentación de umbral iterativo (método Otsu) para detectar grietas en superficies. Por otra parte (Meiling Gong, 2017) propone un modelo adaptativo de segmentación por regiones, introduciendo una mejora al algoritmo *watershed* para resolver el problema de la sobre segmentación y posteriormente mediante un modelo probabilístico, evalúan si el resultado obtenido con *watershed* es aceptable o requiere segmentación adicional.

Respecto a métodos de preprocesamiento y segmentación en (Raajan, Muthuselvi, & Agnes, 2015), se propone utilizar un filtro de *Wiener* para la eliminación de ruido en la imagen, en (Kumar Abhinav, 2017) se propone un nuevo algoritmo de segmentación para imágenes que contienen objetos convexos presentes en múltiples formas con un alto grado de superposición. En (Perfilieva, 2015) se propone utilizar la transformada difusa grado 1 para filtrar una imagen y detectar sus bordes a través de una aproximación al operador gradiente.

Por otra parte, en referencia a detectores de características se encontró en (Salahat & Qasaimeh, 2017) un estudio sobre algoritmos de detección y extracción de características. Destacan la importancia de las características locales para representar puntos de anclaje los cuales al ser convertidos en descriptores numéricos pueden resumir el contenido del cuadro de imagen para su búsqueda y recuperación. En (Rayar, 2017) se presenta la aplicación de dos *descriptores visuales* del estándar MPEG-7, los cuales son utilizados para describir el contenido de las imágenes para su búsqueda y recuperación. Los descriptores permiten obtener características sobresalientes de la imagen de manera global o local.

Finalmente, un elemento importante es el criterio de calidad para evaluar la segmentación obtenida de las imágenes respecto a su *ground truth*. Esto se realiza a través de métricas, la selección de la métrica apropiada depende del criterio de calidad establecido. Algunas métricas proponen medidas de similitud con base en el contexto de la forma y no solo en la ubicación espacial de los píxeles (Qian, 2015); otras evalúan la calidad de la segmentación obtenida utilizando referencias *ground truth* simples. En (Surya & Satheesh, 2016) se presenta una evaluación de distintas métricas, basados en métodos objetivos, presenta de manera general medidas para casos cuando se tiene el *ground truth* y, cuando no, sugiere algunas medidas aplicables.

En (Yeghiazaryana & Voiculescu, 2019) se propone una familia de métricas de superposición de bordes para la evaluación de la segmentaciones por discontinuidades, con base en los resultados obtenidos mostraron cómo, en contraste con las métricas existentes, su métrica SBD (*Symmetric Boundary Dice*) reacciona a una gama más amplia de tipos de error de la segmentación, proponen un algoritmo para SBD y sugieren utilizar esta métrica por sus características de evaluación y facilidad de implementación.

Como las evaluaciones de disimilitud dependen en gran medida de un mapa de borde *ground truth*, un dato inexacto en términos de localización de los píxeles podría beneficiar a detectores de borde imprecisos y/o favorecer una medida de evaluación de disimilitud inadecuada, en (Hasan Abdulrahman, 2017) se propone un método para etiquetar y generar imágenes *ground truth* de forma semiautomática, con el objetivo de generar conjuntos de imágenes adecuados a las métricas con base en la ubicación espacial de píxeles.

Las propuestas que abordan el problema de la segmentación adaptativa lo hacen optimizando o modificando alguna técnica de segmentación en particular (Canny, *watershed*, corte de grafo, fusión y crecimiento de regiones, entre otros), lo cual limita la solución al enfoque dado por el algoritmo empleado. Algunas propuestas utilizan características de contraste o color de la imagen en su solución, pero pocas explotan las características sobresalientes de la imagen (color, textura, forma) que pueden aportar información que ayude a determinar la técnica de segmentación apropiada a cada tipo de imagen.

Otros trabajos relacionados sugieren qué técnicas particulares de preprocesamiento o de detección de bordes aplicar en el procesamiento de las imágenes, por ejemplo: filtro de *Wiener* o transformada difusa grado 1.

Por otra parte, existen artículos que destacan la utilidad de las características sobresalientes de la imagen, extraídas y representadas a través de descriptores y empleadas en aplicaciones de búsqueda y recuperación. En sistemas de recuperación de imágenes basadas en contenido (CBIR, por sus siglas en inglés). Estas características se podrían aprovechar en otras aplicaciones del procesamiento de la imagen.

---

Un tema importante son las métricas dedicadas a valorar la calidad de la segmentación respecto a imágenes de referencia (*ground truth*). Posiblemente derivado de la naturaleza del problema que se intenta resolver, algunas medidas son tan rígidas que no reflejan la percepción humana, otras tratan de reflejarla de diversas maneras y ofrecen mayor flexibilidad lo que significa un margen más amplio para evaluar algoritmos de segmentación.

## 2.3 Propuesta de solución

Se propuso diseñar un modelo de segmentación con base en técnicas de visión artificial y de aprendizaje supervisado. Con el objetivo de que el modelo aprendiera a segmentar imágenes mediante la aplicación de las técnicas de preprocesamiento y de segmentación sobre un conjunto de imágenes de ejemplo (imagen original y *ground truth*), en la Figura 2.8 se muestra el concepto general de la propuesta de solución.

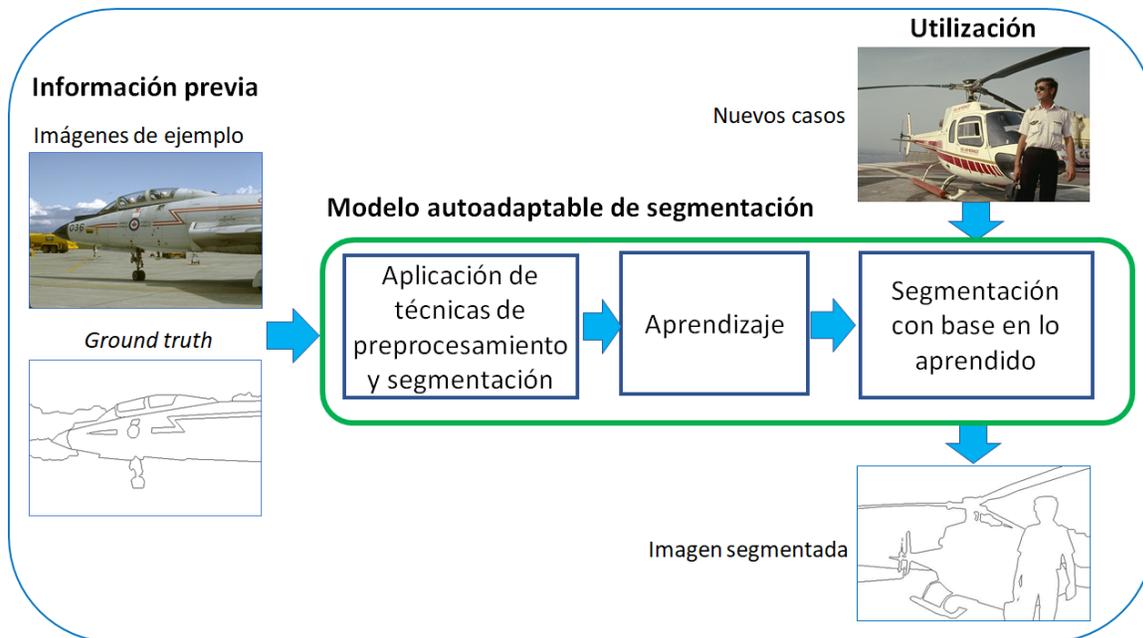


Figura 2.8 Propuesta de solución.

Una vez entrenado el modelo (proceso de aprendizaje) estaría en posibilidades de predecir o recomendar las técnicas apropiadas para segmentar nuevos casos.

Para poder aplicar las técnicas de aprendizaje supervisado, fue necesario encontrar una representación adecuada de las imágenes, para tal efecto, se propuso utilizar el concepto de descriptores y mediante éstos, extraer de las imágenes,

características sobresalientes que las identificaran. Los descriptores se agrupan en forma de vectores de características.

## **2.4 Método de solución**

Para lograr los objetivos establecidos en la presente investigación se han seguido los pasos siguientes:

1. Diseñar la arquitectura del modelo, definir su estructura y operación.
2. Seleccionar las técnicas de preprocesamiento y segmentación a utilizar en la implementación del modelo.
3. Seleccionar los descriptores de características para identificar a las imágenes.
4. Definir los algoritmos de aprendizaje para implementar el modelo.
5. Seleccionar los conjuntos de imágenes para la realización de los experimentos del modelo implementado.
6. Entrenar y evaluar la funcionalidad del modelo. Analizar, clasificar e interpretar los resultados obtenidos.
7. Conclusiones y discusión de resultados.

## Capítulo 3

# Diseño e implementación del modelo

### 3.1 Modelado de la solución

En la Figura 3.1, se presenta el concepto general del modelo de solución propuesto, en él se emplean dos representaciones de las imágenes, la primera con base en la distribución espacial de sus píxeles, la cual se utilizó durante la aplicación de las técnicas de preprocesamiento y segmentación.

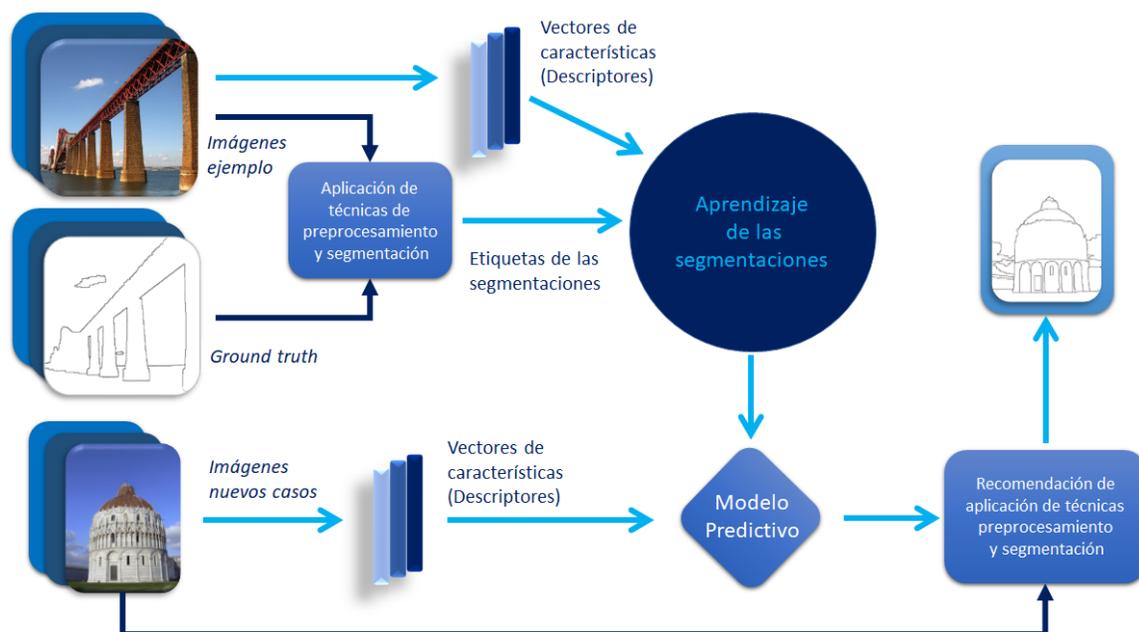


Figura 3.1 Modelo adaptativo propuesto para la segmentación de imágenes por discontinuidades.

La segunda representación es a través de sus características (descriptores), ésta última se empleó para extraer los vectores de características de las imágenes que permitieron la aplicación de las técnicas de aprendizaje automático para su

identificación y reconocimiento y, con ello, generar un modelo de predicción capaz de recomendar las técnicas adecuadas para aplicarlas a la segmentación de nuevas imágenes de características similares.

El modelo propuesto se integra de dos partes, la primera es un modelo de aprendizaje y la segunda un modelo de utilización.

### 3.1.1 Modelo de aprendizaje

En la Figura 3.2 se define el diseño del modelo de aprendizaje, en él aparecen a detalle los bloques de técnicas de preprocesamiento, de segmentación, descriptores de imagen, técnicas de aprendizaje automático (clasificadores) y elemento supervisor, todo lo anterior establecido en los objetivos de la tesis.

Una pieza clave del modelo propuesto, son los descriptores, dado que ellos funcionan como la llave para identificar y reconocer a las imágenes a través de sus vectores de características.

El modelo, durante la etapa de aprendizaje, evalúa las diferentes combinaciones de técnicas de procesamiento y de segmentación que generen imágenes segmentadas con la máxima similitud comparadas con su *ground truth*.

Cada combinación de técnicas tiene asociado un valor entero que la identifica y que en lo sucesivo se denominará “etiqueta” o “clase”. En el caso particular que ninguna combinación de técnicas produzca una segmentación que supere el umbral mínimo de similitud establecido, se asignará “cero” al valor de la clase (clase = 0 o clase cero).

El modelo asocia cada etiqueta o clase que segmentó exitosamente con el vector de características (descriptor) de la imagen correspondiente. De tal manera que,

para procesar nuevos casos, se utilizarán los vectores característicos de las nuevas imágenes, para buscar y recomendar las técnicas apropiadas para su segmentación. Así los descriptores representan el conjunto  $\mathbb{X}$  y las clases representan al conjunto  $\mathbb{Y}$  relacionadas a través de la hipótesis  $h$  en la Ec.( 1.23 ).

Un elemento importante en la adquisición de conocimiento en el modelo es el “supervisor”, éste observará y registrará las acciones realizadas sobre las imágenes para encontrar la solución adecuada para su segmentación. La información que el supervisor recopile se integrará a la “base de conocimiento” del modelo y será utilizada por el elemento “ejecutor” que permitirá la segmentación de nuevos casos y permitirá explicar los resultados obtenidos.

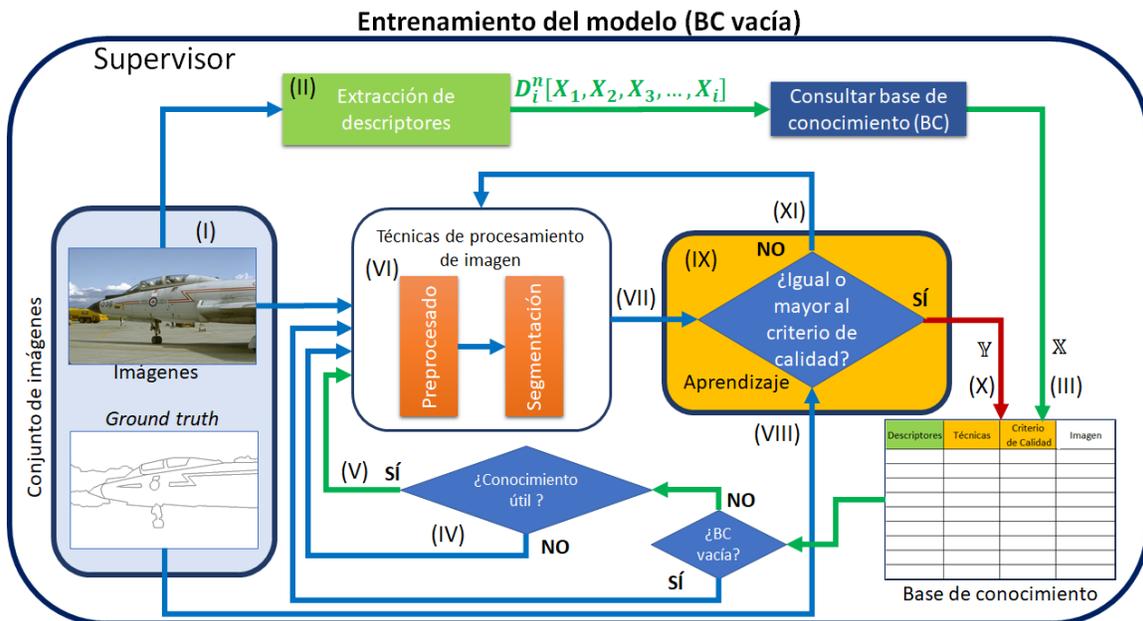


Figura 3.2 Caso 1, modelo con una base de conocimiento vacía.

### Operación del modelo de aprendizaje

Se presentan dos casos posibles:

1. Cuando el sistema no tiene conocimiento, se parte de una base de conocimiento vacía, sólo se tiene el conjunto de imágenes de entrada con su respectivo *ground truth* (modelo en su estado inicial).
2. Cuando el sistema tiene algún conocimiento producto del aprendizaje y con imágenes de entrada con su respectivo *ground truth* (modelo en entrenamiento).

**Primer caso:** Partiendo del modelo con cero conocimiento (base de conocimiento vacía) y tomando como referencia la Figura 3.2, la operación del modelo es la siguiente:

1. A la primera imagen de entrada (I) se le aplican las técnicas de extracción de descriptores de imagen (II).
2. Con base en los descriptores (II) se consulta en la base de conocimiento, BC, (III) la cual está vacía (como se espera en el caso inicial), entonces se procede a aplicar una de las combinaciones de las técnicas de preprocesamiento y segmentación (VI) para procesarla.
3. Se obtiene una imagen segmentada (VII) después de aplicarle las técnicas de procesamiento (VI).
4. Se compara la imagen obtenida en la segmentación (VII) con el *ground truth* de la imagen de entrada (VIII) y mediante un criterio de calidad de la

segmentación se clasifica el resultado, es decir, se aprende de la experiencia (IX).

5. Si el resultado es exitoso (X), es decir, se iguala o supera el criterio de calidad entonces:

a. Se asocian los descriptores (II) con las técnicas aplicadas (VI) a la imagen (I) y esta relación se guarda junto con el valor de su criterio de calidad en la base de conocimiento (III).

b. Si el valor del criterio de calidad es máximo (100%) se termina la operación con esa imagen y se continua con la siguiente. Si no, se intenta con otra conjunto de técnicas (VI) y se itera hasta encontrar una segmentación exitosa (VII) o agotar la combinación de técnicas (VI).

6. En caso de no ser exitoso el resultado (XI)

a. Se intenta con otra combinación de técnicas en (VI) y se itera hasta encontrar un segmentación exitosa (VII) o agotar la combinación de técnicas (VI).



- produce una segmentación exitosa, se aplicarán las combinaciones restantes (VI).
3. Se obtiene una imagen segmentada (VII) al aplicarle las técnicas de procesamiento (VI).
  4. Se compara cada imagen segmentada (VII) obtenida en el punto 3 con el *ground truth* de la imagen de entrada (VIII) y, mediante un criterio de calidad de la segmentación se clasifica el resultado, se aprende de la experiencia (IX).
    - a. Se registra la relación de descriptores (II) con las técnicas aplicadas (VI) a la imagen (I) y se guarda junto con el valor de su criterio de calidad en la base de conocimiento (III). Si el valor del criterio de calidad es máximo (100%) se termina la operación con esa imagen y se continua con la siguiente.
    - b. En caso de no ser exitoso el resultado (XI) se intenta con otra conjunto de técnicas (VI) y se itera hasta encontrar una segmentación exitosa (VII) o agotar la combinación de técnicas (VI).
  5. Se itera el proceso desde el punto 1 hasta terminar de procesar el conjunto de imágenes de entrada.

### 3.1.2 Modelo de utilización

Cuando el sistema ya está entrenado y se procesan nuevas imágenes sin *ground truth*, se espera que el modelo generado segmente las imágenes con características similares a los ejemplos utilizados en el entrenamiento, Figura 3.4.

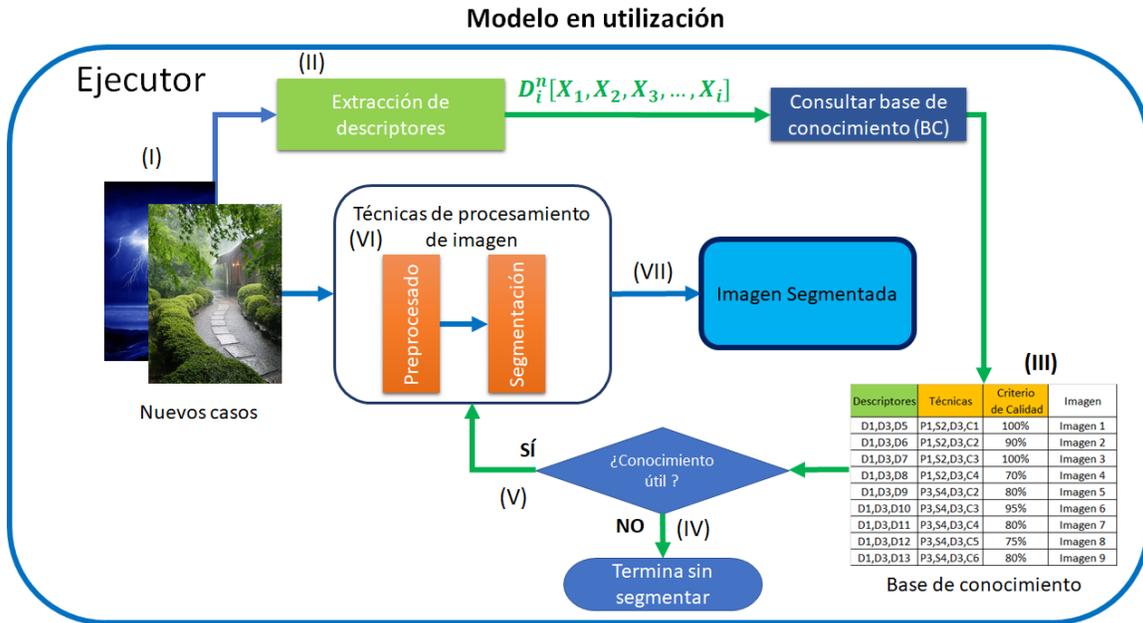


Figura 3.4 Utilización del modelo con el conocimiento adquirido para segmentar nuevos casos.

#### Operación del modelo de utilización

1. A una imagen de entrada (I) se le aplican las técnicas de extracción de descriptores de imagen (II).
2. Con base en los descriptores obtenidos (II) se busca en la base de conocimiento (III) si existe algún patrón similar que permita aplicar algunas técnicas de segmentación (VI) apropiadas a la imagen de entrada.
3. Si la imagen tiene descriptores similares a los registrados en la base de conocimiento del modelo (V), se espera una recomendación de las técnicas

de segmentación adecuadas (VI) indicadas en la base de conocimiento (III) y obtener una segmentación aceptable (VII), en caso contrario no se tendrá el conocimiento suficiente para la segmentación y se terminará la ejecución (IV).

### 3.2 Implementación del modelo

El modelo se implementó en C++ MinGW, bajo el ambiente de desarrollo *Qt Creator 4.9*, utilizando bibliotecas de código abierto como: *OpenCV 4.0* para procesamiento de imagen y aprendizaje automático, la biblioteca *MPEG7FexLib* para extracción de descriptores (Bastan, Cam, Gudukbay, & Ulusoy, 2010), se actualizó esta biblioteca para que funcionara con *OpenCV 4.0*, originalmente está implementada para *OpenCV 2.2*. En la Figura 3.5 se observa la ventana principal del sistema desarrollado.

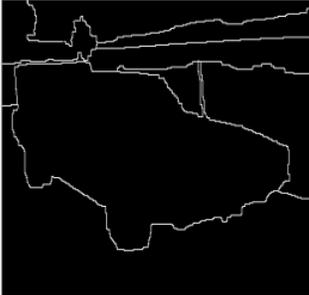
Modelo auto-adaptable de segmentación de imágenes por discontinuidades: proyecto1

Archivo Acciones Analizar Herramientas Ayuda

Imagen de entrada



Imagen ground truth



Detección de bordes



Umbralizado

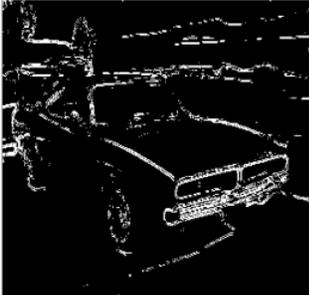


Imagen segmentada

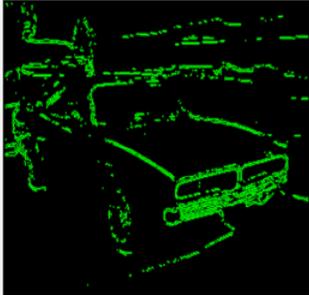


	Imagen	Técnica Preprocesado	Técnica Segmentación	Clase	Métrica Accuracy	Métrica CFHu	Métrica F-Measure	Métrica Dice
1	5.png	NA	NA	0	0.871151	0.793468	0.141477	0.141477

Base de conocimiento
Métricas de base de conocimiento
Clase cero
Nuevas Segmentaciones
Supervisor
Ejecutor
Características
Configuración

	Imagen	Técnica Preprocesado	Técnica Segmentación	Clase	Métrica Accuracy	Métrica CFHu	Métrica F-Measure	Ancho	Alto
1	1.png	FILTRO MEDIANA_K7	GRADIENTE MORFOLOGICO	11	0.952245	0.937638	0.42941	300	400
2	2.png	FILTRO MEDIANA_K7	GRADIENTE MORFOLOGICO	11	0.911489	0.907041	0.147921	300	400
3	3.png	FILTRO MEDIANA_K7	LAPLACIANO	10	0.908629	0.945522	0.133769	400	300
4	4.png	FILTRO MEDIANA_K7	GRADIENTE MORFOLOGICO	11	0.943474	0.895206	0.293012	300	400
5	5.png	NA	NA	0	0.871151	0.793468	0.141477	400	267

Instancias en base de conocimiento : 5

Figura 3.5 Implementación del modelo en C++.

### 3.2.1 Técnicas de preprocesamiento y detección de bordes

Las técnicas de preprocesamiento y detección de bordes implementadas en el sistema se muestran en la Tabla 3.1.

Tabla 3.1 Técnicas de preprocesamiento y segmentación implementadas en el sistema.

No	PREPROCESAMIENTO	SEGMENTACIÓN
1	Ecuilización de Histograma	Sobel
2	Filtro Mediana ( <i>Kernel 7x7</i> )	Canny
3	Filtro Bilateral ( <i>Kernel 3x3</i> )	Scharr
4	Filtro Blur Homogéneo ( <i>Kernel 3x3</i> )	Laplaciano
5	Filtro Blur Gausiano ( <i>Kernel 3x3</i> )	Gradiente morfológico
6	Filtro Blur Mediana ( <i>Kernel 3x3</i> )	Prewitt
7	Filtro Blur Mediana ( <i>Kernel 5x5</i> )	Fuzzy_F1
8	Filtro Blur Gausiano ( <i>Kernel 5x5</i> )	
9	Filtro Box ( <i>Kernel 3x3</i> )	
10	Filtro de Wiener ( <i>Kernel 3x3</i> )	

Fuente: elaboración propia.

El detalle de todas las combinaciones posibles de las técnicas de preprocesamiento y segmentación se muestran en el Anexo A.

### 3.2.2 Descriptores

Aunque los descriptores MPEG7 usualmente se utilizan en sistemas de recuperación de imágenes con base en contenido o sistemas CBIR (*Content-Based Image Retrieval*). En la implementación del modelo, se utilizaron los descriptores MPEG7 para caracterizar y reconocer a las imágenes que segmentaron con técnicas similares y aplicar estas técnicas a nuevas imágenes que presentan características similares.

Lo anterior permite realizar búsquedas en la base de conocimiento del sistema implementado y aplicarlos a algoritmos de aprendizaje automático para producir modelos de predicción, Figura 3.6, los cuales permitirían recomendar las técnicas de segmentación adecuadas para procesar nuevas imágenes. Los descriptores MPEG-7 representados por  $(x_1, x_2, \dots, x_n)$  son las características de las imágenes y la etiquetas o clases  $(y_1, y_2, \dots, y_m)$  representan la combinación de técnicas de preprocesamiento y segmentación aplicadas en cada caso.

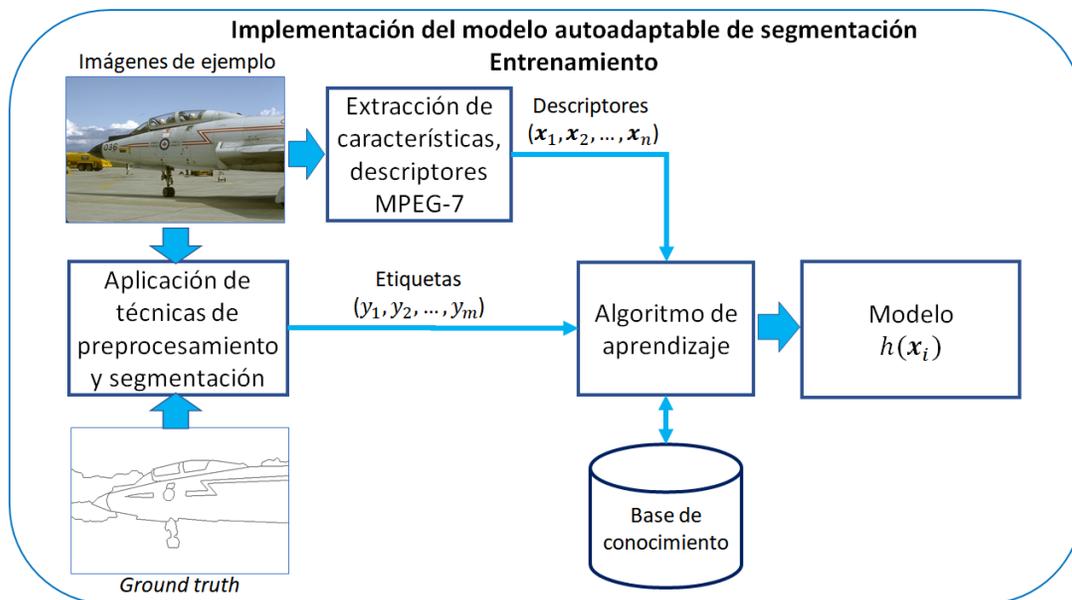


Figura 3.6 Utilización de los descriptores MPEG-7 en la generación del modelo de predicción.

En Figura 3.7, se observa la extracción de los descriptores MPEG-7  $(x_1, x_2, \dots, x_n)$  obtenidos de un conjunto de nuevas imágenes a segmentar, los descriptores son aplicados a un modelo de predicción  $h(x_i)$  que entrega a la salida una recomendación de las técnicas de procesamiento (preprocesamiento y segmentación) para aplicarlas sobre las nuevas imágenes.

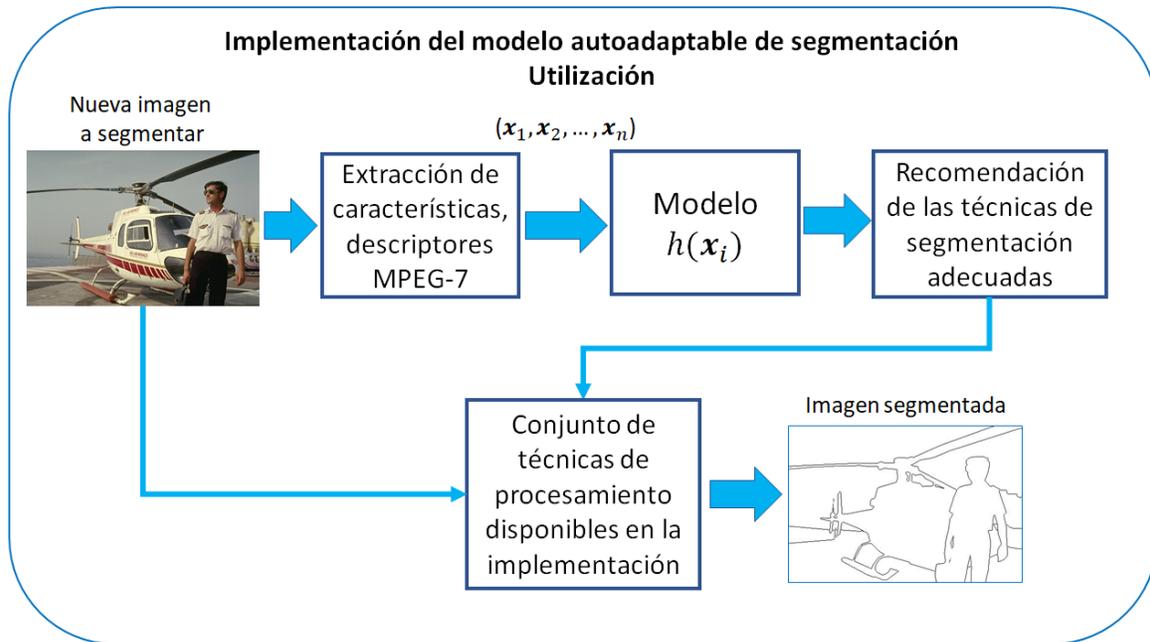


Figura 3.7 Utilización de los descriptores MPEG-7 en la segmentación de nuevas imágenes.

### 3.2.3 Métricas de similitud

#### Similitud de imágenes a través de descriptores

Para determinar la similitud entre dos imágenes mediante sus descriptores MPEG-7 se utilizan sus medidas de distancia, el estándar define las fórmulas para cada distancia (Manjunath, Salembier, & Sikora, 2003), (ISO, 2001), (ISO, 2004).

En la Tabla 3.2 se definen las medidas de distancia para cada descriptor. Los descriptores CSD, SCD, RSD y HTD utilizan la distancia  $L1$  o Mahattan Ec.( 3.1 ), el resto de los descriptores emplean una medida de distancia particular.

Tabla 3.2 Fórmulas de cálculo de distancias de los diferentes descriptores MPEG-7.

Descriptor	Medida de similitud	Núm.
CSD, SCD, RSD, HTD.	$D(h_1, h_2) = \sum_{i=1}^M  h_1(i) - h_2(i) $	(3.1)
EHD	$D(h_1, h_2) = \sum_{i=0}^{79}  h_1(i) - h_2(i)  + 5 \times \sum_{i=0}^4  h_1^g(i) - h_2^g(i) $ $+ \sum_{i=0}^{64}  h_1^s(i) - h_2^s(i) $	(3.2)
CLD	$D(h_1, h_2) = \sqrt{\sum_{i=1}^L w_{yi}(DY_{1i} - DY_{2i})^2}$ $+ \sqrt{\sum_{i=1}^M w_{bi}(DCb_{1i} - DCb_{2i})^2}$ $+ \sqrt{\sum_{i=1}^M w_{ri}(DCr_{1i} - DCr_{2i})^2}$	(3.3)
DCD	$D^2(h_1, h_2) = \sum_{i=1}^{N_1} p_{1i}^2 + \sum_{i=1}^{N_2} p_{2i}^2$ $- \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} 2a_{1i,2j} p_{1i} p_{2j}$	(3.4)

Fuente: (Manjunath, Salembier, & Sikora, 2003). Elaboración propia.

La medida de similitud entre dos descriptores EHD se define en la Ec. (3.2) donde  $h_1(i)$  y  $h_2(i)$  representan los valores del histograma normalizados de la imagen 1 y la imagen 2 que se están comparando, respectivamente. En  $h_1^g(i)$  y  $h_2^g(i)$  se representan los valores normalizados para los histogramas de borde globales de la imagen 1 y la imagen 2, respectivamente, que se obtienen de los histogramas locales correspondientes  $h_1(i)$  y  $h_2(i)$ . De manera similar,  $h_1^s(i)$  y  $h_2^s(i)$  representan los valores del histograma para los histogramas de borde semi global

de la imagen 1 y la imagen 2, respectivamente. Dado que el número de contenedores del histograma global es relativamente menor que el de los histogramas locales y semi globales, se aplica el factor de ponderación 5.

La similitud entre dos descriptores CLD,  $\{DY, DCr, DCb\}$  y  $\{DY', DCr', DCb'\}$ , se obtiene aplicando la Ec. ( 3.3 ) entre sus respectivas componentes. La característica perceptiva del sistema de visión humana podría incluirse para el cálculo de similitud ya que la descripción de la característica está en el dominio de la frecuencia. Las distancias deben ser ponderadas apropiadamente, con pesos mayores ( $w_{yi}$ ,  $w_{bi}$  y  $w_{ri}$ ) dados a los componentes de frecuencia más baja, para que coincidan con la característica.

En el descriptor DCD, ignorando el parámetro de varianza (opcional) y la coherencia espacial, la disimilitud  $D(h_1, h_2)$  entre los dos descriptores se puede calcular como se indica en la Ec.( 3.4 ), donde los subíndices 1 y 2 en todas las variables representan las descripciones  $h_1$  y  $h_2$  respectivamente, y  $a_{k,l}$  es el coeficiente de similitud entre dos colores  $c_k$  y  $c_l$  Ec.( 3.5 ).

$$a_{k,l} = \begin{cases} 1 - d_{k,l}/d_{\max} & d_{k,l} \leq T_d \\ 0 & d_{k,l} > T_d \end{cases} \quad (3.5)$$

donde  $d_{k,l}$  es la distancia euclidiana entre dos colores  $c_k$  y  $c_l$ ,  $T_d$  es la distancia máxima para que dos colores se consideren similares y  $d_{\max} = \alpha T_d$ . En particular, esto significa que dos colores dominantes cualquiera, están separados al menos por  $T_d$ .

Por otra parte, se emplearon los descriptores de Hu para evaluar la similitud entre los contornos obtenidos de la segmentación y el *ground truth* respectivo (métrica de contorno de forma "CFHu").

### Similitud de imágenes a través de mapa de bordes

El principal criterio de similitud utilizado entre la imagen segmentada y su imagen de referencia o *ground truth* fue la métrica *accuracy* Ec. ( 3.6 ), definida a partir de la matriz de confusión Tabla 3.3.

**Tabla 3.3 Matriz de confusión, coincidencia de píxeles entre la imagen segmentada y su *ground truth*.**

Imagen segmentada por el modelo	Resultado	Ground truth	
		Pixel = 1/P	Pixel = 0/N
	Pixel = 1/P	True Positive (TP)	False Positive (FP)
	Pixel = 0/N	False Negative (FN)	True Negative (TN)

Fuente: (Surya & Satheesh, 2016). Elaboración propia.

$$Accuracy = \frac{|TP + TN|}{|P + N|}; P = TP + FN; N = FP + TN \quad (3.6)$$

*Accuracy* es una relación entre la observación predicha correctamente y el total de observaciones; esto es la relación entre el número de píxeles coincidentes entre la segmentación obtenida por el modelo y el *ground truth*, dividido por el número total de píxeles (segmentación y *ground truth*). Como verificadores de la exactitud de la segmentación se evaluaron las métricas: *F-Measure* (*F-score* o  $F_1$ -*score*) Ec. ( 3.7 ) y el coeficiente Dice (DSC) Ec. ( 3.8 ), ambas son medidas de traslape equivalentes y determinan la cantidad de superposición espacial entre el *ground truth* ( $G$ ) y la imagen segmentada ( $M$ ) generada por el modelo. *F-Measure* también deriva de la matriz de confusión.

$$F - Measure = \frac{2 * (Recall * Precision)}{(Recall + Precision)} \quad (3.7)$$

$$Precision = \frac{TP}{TP + FP} = \frac{|M \cap G|}{|M|}$$

$$Recall = \frac{TP}{TP + FN} = \frac{|M \cap G|}{|G|}$$

$$DSC = \frac{2|M \cap G|}{|M| + |G|} \quad (3.8)$$

### 3.2.4 Técnicas de aprendizaje supervisado

Se seleccionaron cinco algoritmos de aprendizaje supervisado: IB1 (aprendizaje con base en instancias) el cual se programó de acuerdo a (Aha, Kibler, & Albert, 1991), KNN (K-vecinos más cercanos), SVM (máquina de soporte vectorial) con *kernel* lineal, SVM con *kernel* polinomial y SVM con *kernel* de base radial; se utilizaron las implementaciones de los algoritmos KNN y SVM contenidas en la biblioteca ML de OpenCV versión 4.0.

Para el motor de clasificación en la etapa de aprendizaje se utilizaron los algoritmos: IB1 y KNN con la base de conocimiento del modelo. Estos funcionaron como agrupadores en la etapa de entrenamiento, dado que se debían asociar las combinaciones de técnicas exitosas (clases) con los descriptores de las imágenes segmentadas que se aproximaban más a su *ground truth* de acuerdo con la métrica de similitud establecida.

Una vez etiquetados los descriptores de cada imagen con su respectiva clase o número de combinación de técnicas (instancias), se realizó el entrenamiento del algoritmo SVM con los tres *kernels* indicados anteriormente para que aprendiera de las instancias generadas con KNN o IB1. El resultado fueron tres modelos de clasificación de SVM. Así, en la etapa de utilización, se emplearon como clasificadores: KNN (con la base de conocimiento del modelo), SVM con *kernel* lineal, SVM con *kernel* polinomial y SVM con *kernel* de base radial.

KNN y SVM clasificaron las imágenes con base en sus descriptores para recomendar la clase o combinación de técnicas de segmentación adecuada según lo aprendido durante el entrenamiento.

### 3.2.5 Calibración del modelo implementado

Una vez implementado el modelo, se realizó la calibración de éste a través una serie de pruebas funcionales para verificar que todos los elementos integrados operaran de manera adecuada.

Para este proceso se utilizó el conjunto de imágenes MPEG7\_CE-Shape-1\_Part\_B, formado por 1400 imágenes en blanco y negro (Latecki, Lakämper, & Eckhardt, 2000), (<http://www.imageprocessingplace.com>, 2019). El *ground truth* de este conjunto se obtuvo mediante la aplicación *XnConvert* ([www.xnview.com](http://www.xnview.com), 2019), debido a que de origen no lo contiene. En la

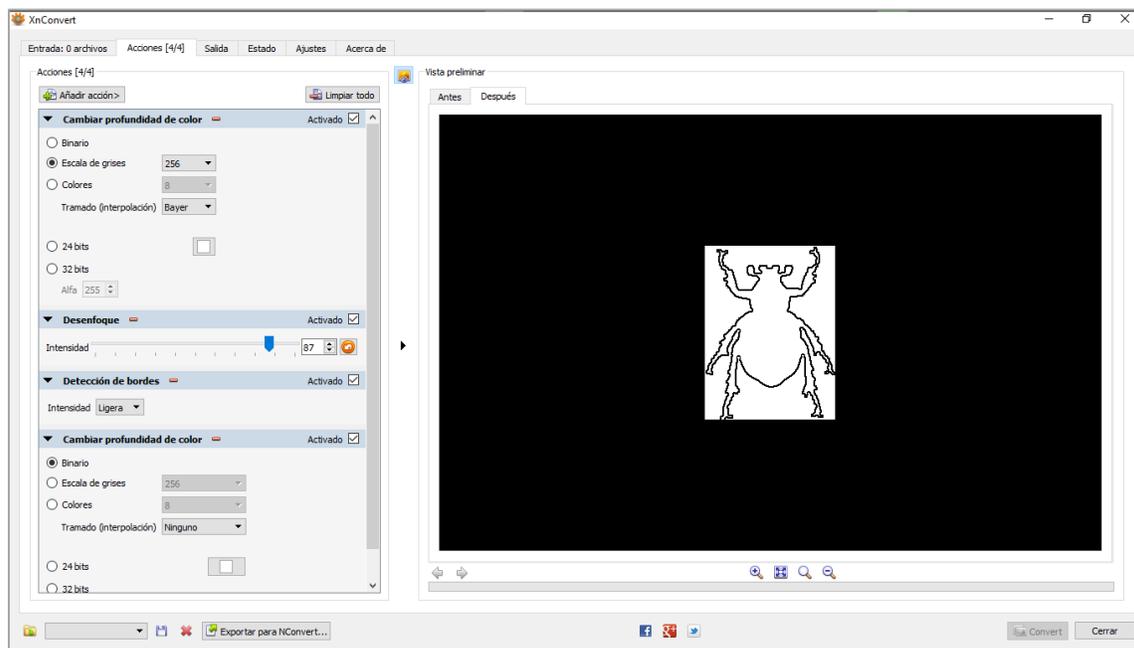


Figura 3.8 se observa la aplicación *XnConvert* durante la extracción de bordes de una imagen, *XnConvert* no especifica el algoritmo empleado en la detección de bordes.

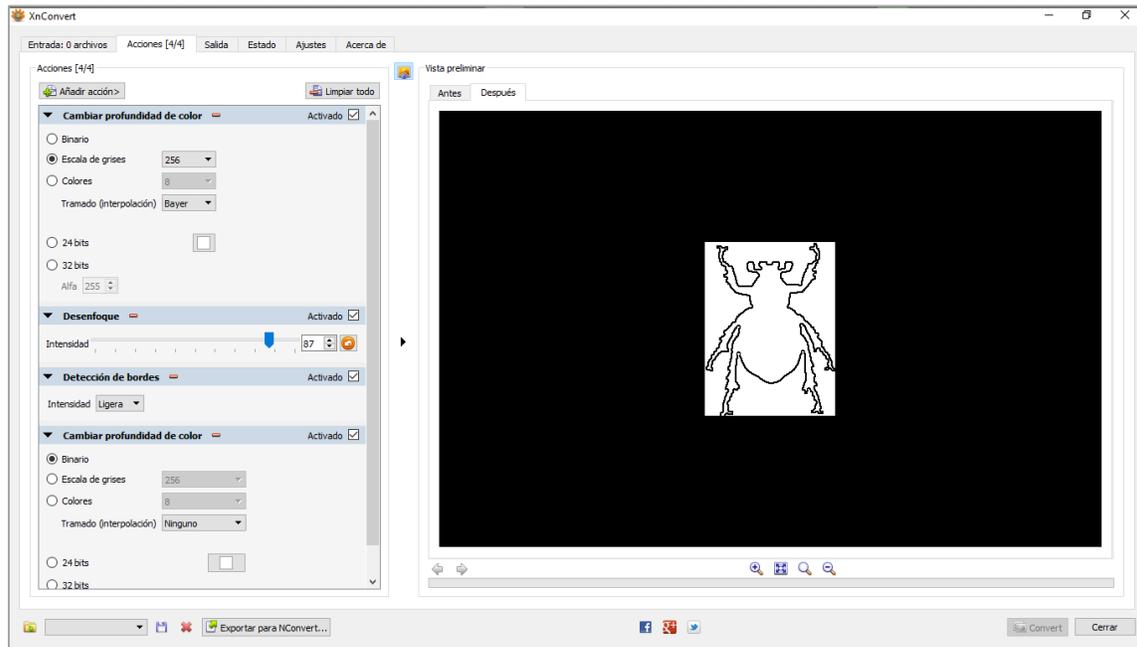


Figura 3.8 Aplicación *XnConvert* con la configuración de detección de bordes.

Del conjunto MPEG7\_CE-Shape-1\_Part\_B, se utilizaron 700 imágenes para validar el funcionamiento del modelo implementado: evaluar la segmentación, las métricas de similitud seleccionadas, evaluar el modelo durante el entrenamiento y en la utilización

Se evaluaron distintas métricas de similitud para comparar la segmentación con el *ground truth* correspondiente a la imagen procesada. Se utilizó la métrica *accuracy* con base en los resultados de la matriz de confusión para cada imagen, CFHu que realiza la comparación a través de las formas de la segmentación y el *ground truth*; *F-Measure* y el coeficiente *Dice* evalúan la exactitud de la segmentación a través de una comparación de traslape, es decir sobre la superposición de la distribución espacial de los píxeles entre la segmentación obtenida por el modelo y el *ground truth*.

En la Figura 3.9 se observa la evaluación de la segmentación generada por el modelo con las métricas de similitud mencionadas, sobre la imagen apple-1.png. Se puede apreciar que para segmentaciones exactas las cuatro medidas coinciden en el máximo valor de similitud (igual a uno).

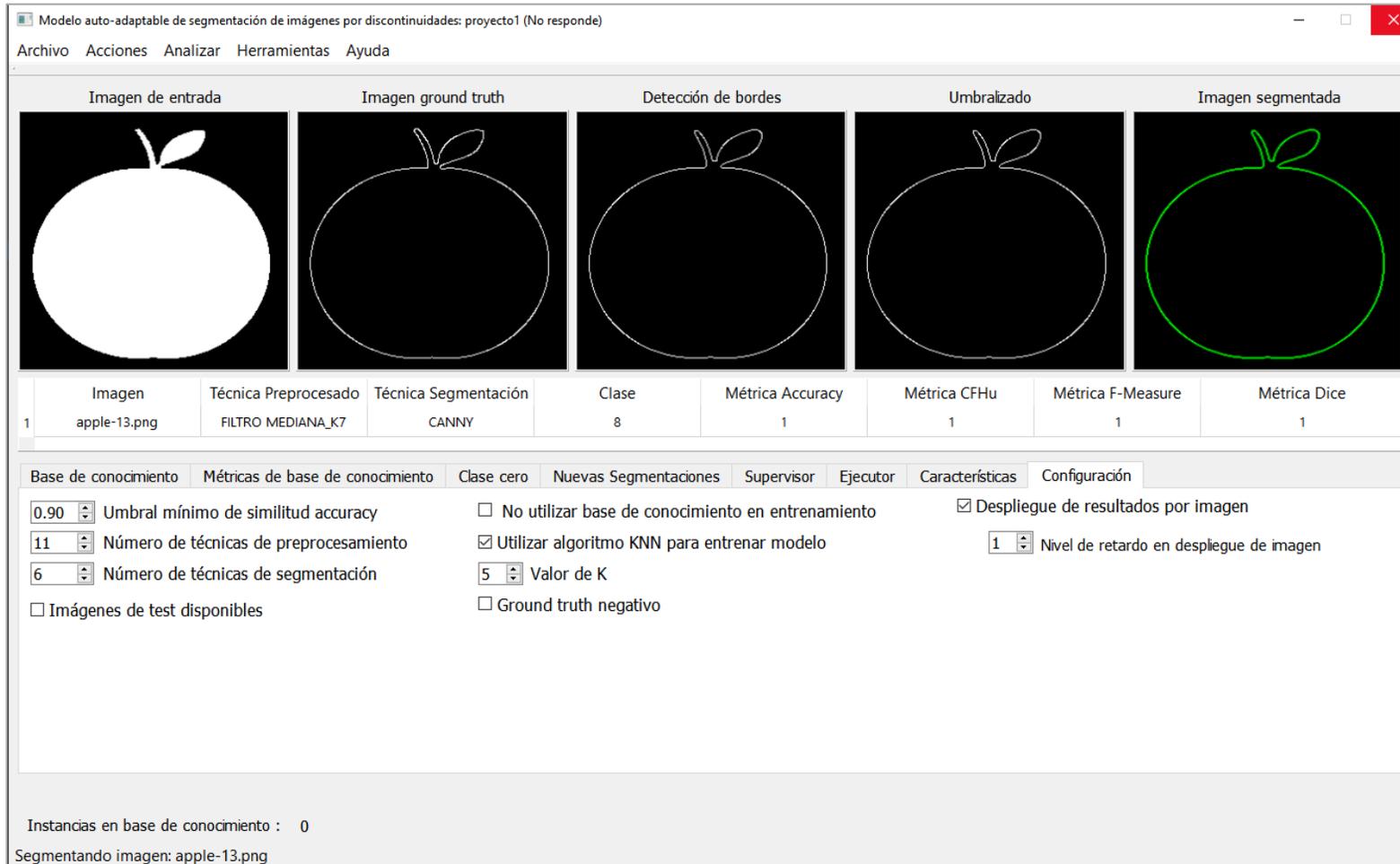


Figura 3.9 Prueba funcional del modelo implementado, las cuatro métricas coinciden con el máximo valor de similitud cuando la segmentación obtenida es igual al *ground truth*.

La secuencia de técnicas que segmentaron la imagen fue: un filtro *Gaussiano* con un kernel de 5x5 (preprocesamiento) y *Canny* (detector de bordes), a esta combinación se le asignó la etiqueta o clase número 44.

En la Figura 3.10 se presenta la imagen comparativa, resultado de la segmentación generada por el sistema. Se observa la imagen de entrada original, su *ground truth* (GT), el resultado de la segmentación (S) en color verde y la diferencia entre las dos últimas (S-GT), en ésta última, los píxeles coincidentes entre la imagen segmentada (en verde) y el *ground truth* (en rojo) aparecen en color amarillo.

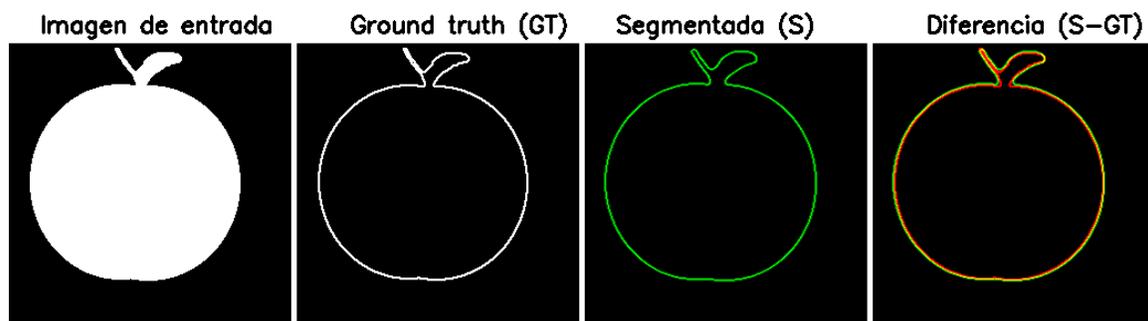


Figura 3.10 Imagen de salida de la segmentación realizada por el sistema sobre la imagen apple-1.png.

En la Figura 3.11 se observa la superposición de la segmentación sobre la imagen original apple-1.png (a) y la paleta con colores dominantes (b) extraídos con el *descriptor visual* DCD, en este caso el mayor porcentaje lo obtuvieron el color blanco y el color negro, como se esperaba.

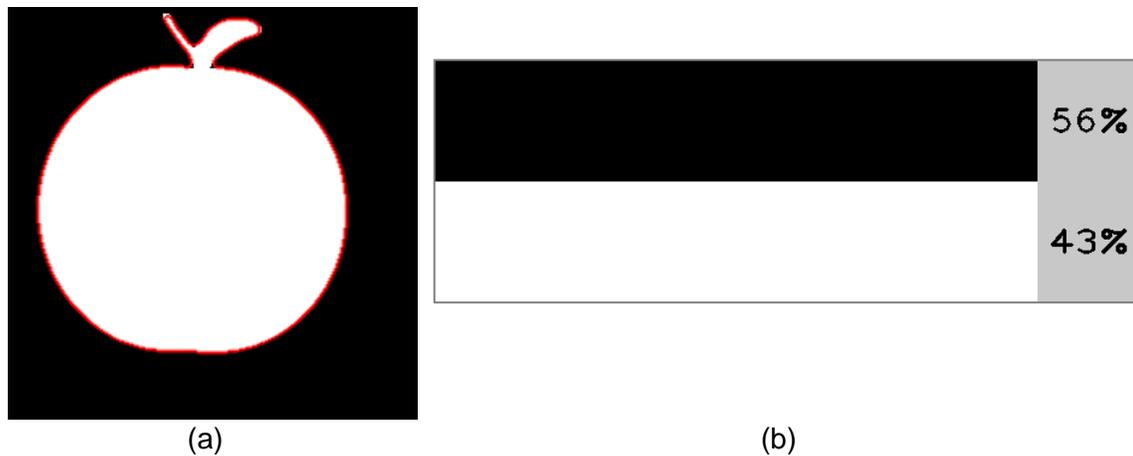


Figura 3.11 a) Imagen segmentada superpuesta a la original apple-1.png b) Paleta de colores dominantes de la imagen original.

Los resultados obtenidos en el proceso de calibración, sobre el conjunto MPEG7\_CE-Shape-1\_Part\_B, por ser imágenes simples en cuanto a color, forma y textura, producen resultados confiables que permitieron validar el funcionamiento del sistema.

### 3.2.6 Conclusiones

Resultado de las pruebas de funcionalidad y evaluación de descriptores se concluyó que el modelo opere con los *descriptores visuales MPEG-7*: CSD, SCD, CLD, EHD, HTD y RSD concatenados para constituir el vector de características y el DCD para caracterización de color y contraste de las imágenes, se descartó el CShD por no aportar información útil al modelo implementado. En la Tabla 3.4 se observa la lista de los descriptores con su longitud equivalente a  $n$  números enteros.

Tabla 3.4 Longitudes de los descriptores MPEG-7 evaluados.

<b>Descriptor</b>	<b>Descripción</b>	<b>n</b>
<b>CSD</b>	<i>Color Structure Descriptor</i>	64
<b>SCD</b>	<i>Scalable Color Descriptor</i>	256
<b>CLD</b>	<i>Color Layout Descriptor</i>	120
<b>EHD</b>	<i>Edge Histogram Descriptor</i>	80
<b>HTD</b>	<i>Homogeneous Texture Descriptor</i>	62
<b>RSD</b>	<i>Region Shape Descriptor</i>	35
<b>CONCATD</b>	<i>Vector característico (concatenación de los anteriores)</i>	617
<b>DCD</b>	<i>Dominant Color Descriptor</i>	58
<b>CShD</b>	<i>Contour Shape Descriptor</i>	4

Fuente: (Manjunath, Salembier, & Sikora, 2003). Elaboración propia.

También se utilizaron los descriptores de Hu para evaluar la característica de similitud de forma mediante la métrica CFHu.

Adicionalmente se integraron los descriptores estadísticos: media; la cual aporta información sobre el “brillo” de la imagen; la varianza y desviación estándar; aportan información del “contraste”, la entropía y el factor de suavidad de textura R (relacionada con la varianza); proporcionan información sobre la presencia de textura en la imagen.

El sistema desarrollado, dado un conjunto de imágenes de entrada (entrenamiento y nuevos casos), produce a la salida imágenes y archivos con información para diversos análisis, Figura 3.12.

## CONJUNTOS DE ENTRENAMIENTO Y PRUEBA

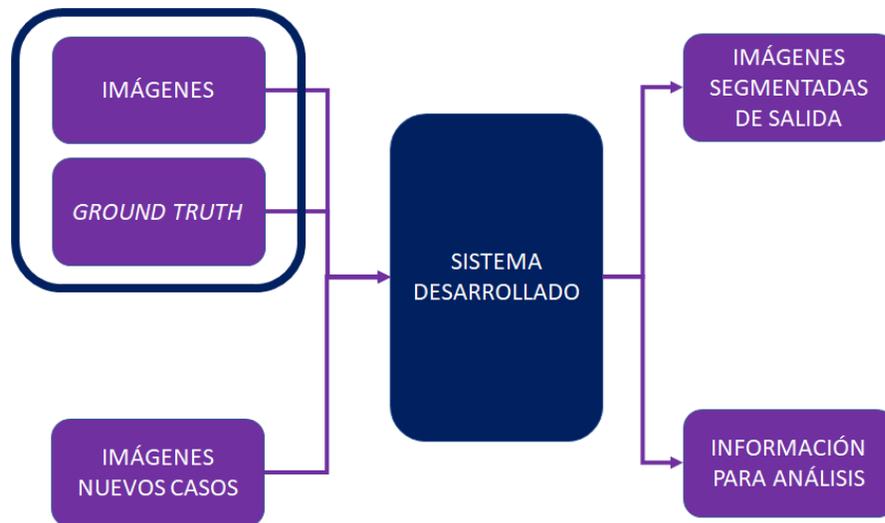


Figura 3.12 Productos generados a la salida del sistema implementado.

El sistema desarrollado tiene las siguientes características principales:

1. Genera segmentaciones por discontinuidades de imágenes de entrada a color o monocromáticas. En el caso de las imágenes a color, se selecciona el *canal de mayor varianza* sobre el cual se realiza la segmentación. Este canal seleccionado (imagen) se guarda como archivo de imagen en escala de grises.
2. Genera imágenes comparativas de la segmentación de la imagen de entrada contra su *ground truth* y las diferencias entre ellas.
3. Genera la imagen original con la segmentación superpuesta.
4. Produce archivos “csv” de cada uno de los *descriptores visuales* extraídos de cada imagen y guardados en la base de conocimientos del sistema; útiles en el análisis de patrones y análisis semántico. Posiblemente sean útiles en trabajos futuros.
5. Genera matrices de distancia (en formato “csv”) de cada uno de los *descriptores visuales*, útil en el análisis de patrones de estos.

6. Genera registro de la secuencia de técnicas aplicadas a cada imagen en archivo de texto.
7. Genera una imagen con la paleta de colores dominantes indicada por el *descriptor visual* DCD para cada imagen procesada.
8. Se estableció una sección de configuración donde se define el umbral mínimo de la métrica de similitud, el número de técnicas a aplicar, la opción de utilizar o no el contenido de la base de conocimiento (BC), la selección del clasificador a utilizar durante el entrenamiento (IB1 es el algoritmo por defecto, KNN es seleccionable) y el despliegue en pantalla de los resultados de la segmentación de cada imagen procesada Figura 3.13.

Base de conocimiento	Métricas de base de conocimiento	Clase cero	Nuevas Segmentaciones	Supervisor	Ejecutor	Características	Configuración
0.90	Umbral mínimo de similitud accuracy	<input type="checkbox"/>	No utilizar base de conocimiento en entrenamiento	<input type="checkbox"/>	Despliegue de resultados por imagen		
11	Número de técnicas de preprocesamiento	<input checked="" type="checkbox"/>	Utilizar algoritmo KNN para entrenar modelo	1	Nivel de retardo en despliegue de imagen		
6	Número de técnicas de segmentación	5	Valor de K	<input checked="" type="checkbox"/>	Ground truth negativo		
<input type="checkbox"/>	Imágenes de test disponibles						

Figura 3.13 Sección de configuración del sistema desarrollado.

# Capítulo 4

## Experimentación y resultados

### 4.1 Detalles y objetivos de la experimentación

Mediante un proceso de experimentación se buscaba responder a las preguntas de investigación planteadas en la definición del problema (pág. 28) y lograr los objetivos establecidos.

### 4.2 Experimentación

Para el proceso de experimentación se utilizaron dos conjuntos de imágenes:

1. Corel200 con 200 imágenes (B., L., Mou, & M.H., Submitted, 2015).
2. BSD300 con 300 imágenes (Martin, Fowlkes, Tal, & Malik, 2001).
3. *Motorbikes* con 798 imágenes (Fei-Fei, Fergus, & Perona, 2004).

Los tres conjuntos contienen pares de imágenes: la imagen original y la imagen de referencia o *ground truth*. En la Figura 4.1 se muestra un ejemplo del conjunto Corel200.



Figura 4.1 a) Imagen original 13.png b) *Ground truth* gt\_13.png.

El tercer conjunto contiene un *ground truth* para segmentación por regiones el cual se utilizó para generar el *ground truth* de contornos con la aplicación *XnConvert* ([www.xnview.com](http://www.xnview.com), 2019).

Las dimensiones de las imágenes en el conjunto Corel200 varían entre: 557x379, 214x321, 384x256, 399x557 píxeles. Las dimensiones de las imágenes en el conjunto BSD300 varían entre: 481x321 y 321x481 píxeles. Y el de *Motorbikes* varían entre: 282x184 y 261x163 píxeles, todas las imágenes de los tres conjuntos en formato png.

Sus respectivos *ground truth* tienen las mismas dimensiones y el mismo formato que la imagen original. El nombre es el mismo que la imagen original más el prefijo “gt\_”, así por ejemplo la imagen 13.png, tendrá una imagen *ground truth* “gt\_13.png”.

Los conjuntos se emplearon para evaluar la segmentación producida por el modelo implementado, son imágenes de escenas reales y complejas, en términos de iluminación, textura, color y contraste. El elemento clave fue la extracción de los *descriptores visuales MPEG-7*, utilizados para entrenar el modelo y en la fase de utilización se emplearon para detectar imágenes similares y aplicar las técnicas de segmentación apropiadas para procesar nuevos casos.

Dada la naturaleza del problema que se pretendía resolver es del tipo “mal planteado”, se establecieron algunas condiciones o restricciones (algunos conjuntos pueden tener más de una imagen *ground truth*, esto es, más de una solución a la segmentación), se seleccionó la primer imagen *ground truth* del conjunto y se estableció como 0.8 (80%), el mínimo valor de la métrica de similitud normalizada.

La combinación total de las secuencias de segmentación se obtuvo al multiplicar el número de técnicas de preprocesamiento por el número de técnicas de segmentación seleccionadas.

Esto es, si se tienen 11 técnicas de preprocesamiento y 6 de segmentación el total de combinaciones es 66 (11 x 6). Cada combinación de técnicas se representa por medio de un número de “clase” en el sistema desarrollado, Tabla 4.1.

**Tabla 4.1 Combinación de técnicas representadas por su número de clase.**

<b>COMBINACIÓN DE TÉCNICAS DE PREPROCESAMIENTO Y SEGMENTACIÓN</b>
Clase 0: [0,0] = NA,NA (NO EXISTE UNA COMBINACIÓN ADECUADA PARA SEGMENTAR LA IMAGEN EN LOS TÉRMINOS DE LA MÉTRICA DE SIMILITUD)
Clase 1: [1,1] = ECUALIZACION HISTOGRAMA, SOBEL
Clase 2: [1,2] = ECUALIZACION HISTOGRAMA, CANNY
Clase 3: [1,3] = ECUALIZACION HISTOGRAMA, SCHARR
Clase 4: [1,4] = ECUALIZACION HISTOGRAMA, LAPLACIANO

Fuente: elaboración propia.

A las segmentaciones que superaron el umbral de la métrica de similitud predefinido, se les etiquetó o clasificó con el número de la combinación de técnicas correspondiente.

Las segmentaciones que estuvieron por debajo del umbral se etiquetaron como clase cero (clase = 0), lo que significa que el modelo no logró segmentar esas imágenes con el conocimiento y las técnicas que tenía disponibles en ese instante.

Por lo tanto, el modelo no recomendó ninguna combinación de técnicas de segmentación.

Los experimentos se realizaron de acuerdo con las especificaciones indicadas en la Tabla 4.2.

**Tabla 4.2 Experimentos realizados para evaluar el modelo implementado.**

Descripción de experimentos	Objetivo
<b>Experimento 1</b>	
<p>a) <b>Seleccionar un conjunto de imágenes y aplicarle una combinación de 8 técnicas de preprocesamiento y 4 de segmentación, con un umbral mínimo de la métrica de similitud igual a 0.8.</b></p>	<p>Evaluar el desempeño del modelo al incrementar el número de técnicas de preprocesamiento y segmentación.</p>
<p>b) <b>Con el mismo conjunto de imágenes y mismo valor de métrica, repetir el procedimiento del inciso a), estableciendo 9 técnicas de procesamiento y 5 de segmentación.</b></p>	
<p>c) <b>Con el mismo conjunto de imágenes y mismo valor de métrica, repetir el procedimiento del inciso a), agregar todas las técnicas disponibles, 11 técnicas de procesamiento y 6 de segmentación.</b></p>	
<b>Experimento 2</b>	
<p><b>Seleccionar un conjunto de imágenes y utilizando todas las técnicas de preprocesamiento y segmentación disponibles, fijar el umbral mínimo de la métrica en 0.9, dejar pasar todas las segmentaciones cuya métrica se encuentren por debajo del umbral con la clase correspondiente (no etiquetarlos con clase cero).</b>  <b>Hacer un análisis de las imágenes que dieron un valor cercano a la métrica, abajo del umbral. Los casos de métrica cero serán considerados valores atípicos (<i>outliers</i>).</b></p>	<p>Verificar si la métrica de similitud de las segmentaciones es la adecuada para el modelo y cómo afecta ésta en la selección de la segmentaciones apropiadas.          Reducir el número de elementos en la clase cero.</p>

Fuente: elaboración propia.

Tabla 4.2 Experimentos realizados para evaluar el modelo implementado (continuación).

<b>Experimento 3</b>	
<b>Comparar el desempeño del modelo cuando utiliza su base de conocimiento adquirida (BC) durante su entrenamiento para procesar nuevas imágenes y cuando realiza una aplicación exhaustiva de las combinaciones de técnicas de preprocesamiento y segmentación sin utilizar la base de conocimiento. Seleccionar un conjunto de imágenes para experimentar.</b>	Evaluar sí la utilización de la BC afecta o no el desempeño del modelo en cuanto a la velocidad de aprendizaje.
<b>Experimento 4</b>	
<b>Utilizar todas las técnicas disponibles del modelo para procesar un conjunto de imágenes y evaluar los resultados al variar el umbral de la métrica: 0.8, 0.9 y 0.95. Seleccionar un conjunto de imágenes para experimentar.</b>	Analizar el desempeño del modelo en cuanto a la precisión de la segmentación al disminuir o incrementar el umbral .
<b>Experimento 5</b>	
<b>Seleccionar un conjunto de imágenes, extraer el 10% del total, éstas serán la imágenes de prueba, entrenar el modelo con las imágenes restantes (el 90%).</b>	Evaluar la funcionalidad del modelo en su fase de utilización una vez entrenado.
<b>Experimento 6</b>	
<b>Aplicar el modelo a un conjunto específico de imágenes que requiera la mejor combinación de técnicas para su segmentación.</b>	Evaluar la capacidad del modelo para recomendar técnicas de segmentación para imágenes específicas.
<b>Experimento 7</b>	
<b>Aplicar el modelo a un conjunto de imágenes de diferentes categorías para que recomiende las técnicas para su segmentación o las rechace en caso de no ser posible la segmentación.</b>	Mostrar las características de las imágenes que son segmentadas adecuadamente y las que son rechazadas por el modelo. Explicar el porqué de los resultados.

Fuente: elaboración propia.

Para cada experimento se establecieron las condiciones en las que se realizó, se anotaron los resultados obtenidos y las conclusiones.

### Experimento 1

Se realizó bajo las siguientes condiciones:

Imágenes de entrada: Conjunto Corel200

Número de imágenes = 200

KNN como motor de clasificación en la base de conocimiento,  $K = 5$

Umbral de referencia, métrica CFHu = 0.8

Este experimento tiene tres apartados:

#### Apartado 1:

Número de técnicas de preprocesamiento = 8

Número de técnicas de segmentación = 4

Número de clases  $8 \times 4 = 32$

#### Apartado 2:

Número de técnicas de preprocesamiento = 9

Número de técnicas de segmentación = 5

Número de clases  $9 \times 5 = 45$

#### Apartado 3:

Número de técnicas de preprocesamiento = 11

Número de técnicas de segmentación = 6

Número de clases  $11 \times 6 = 66$

Resultados obtenidos:

Se observó una modesta mejora en el desempeño del modelo cuando se incrementó la cantidad de técnicas de preprocesamiento y segmentación sobre el

conjunto de imágenes seleccionado, Tabla 4.3. Sólo se incrementó en cuatro elementos clasificados arriba del umbral en los apartados 2 y 3 (140 elementos con clase > 0) comparados con el apartado 1 (136 elementos con clase > 0), la proporción en porcentaje se indica entre paréntesis.

**Tabla 4.3 Comparativa del desempeño de modelo respecto a cantidad de técnicas aplicadas con un umbral de la métrica = 0.8.**

Apartado del experimento	Número de imágenes clasificadas		Número de técnicas		
	Clase > 0	Clase 0	Preprocesamiento	Segmentación	Total
1	136 (68%)	64 (32%)	8	4	32
2	140 (70%)	60 (30%)	9	5	45
3	140 (70%)	60 (30%)	11	6	66

Fuente: elaboración propia.

En los tres apartados, se encontró que la secuencia de segmentación predominante fue la misma, como se puede observar en la Tabla 4.4. Esto significó que, con el conocimiento, la métrica de similitud seleccionada y las técnicas disponibles en el modelo, la secuencia encontrada fue la mejor combinación para el conjunto de imágenes procesado.

**Tabla 4.4 Clase predominante y su secuencia de técnicas equivalente de cada apartado correspondiente con un umbral de métrica CFHu = 0.8.**

Apartado del experimento	Clase predominante	Técnica de preprocesamiento	Técnica de segmentación
1	8	FILTRO MEDIANA_K7	LAPLACIANO
2	9	FILTRO MEDIANA_K7	LAPLACIANO
3	10	FILTRO MEDIANA_K7	LAPLACIANO

Fuente: elaboración propia.

Recordando que cada combinación se representa como un número de clase, ésta varía al cambiar el número de combinaciones de técnicas de preprocesamiento y de segmentación, ver Tabla 4.1.

Por otra parte, se obtuvo un número importante de segmentaciones por abajo del umbral 0.8 (~30%) con la métrica CFHu. Esta métrica se seleccionó por no ser tan estricta en cuanto a la ubicación espacial de los píxeles, se soporta en las propiedades geométricas de las formas, es invariante ante rotaciones, y traslaciones. Se esperaba se adecuara como criterio de similitud del modelo, no fue así; por lo que fue necesario evaluar otra métrica, lo cual se realizó en el experimento 2.

## Experimento 2

Se realizó bajo las siguientes condiciones:

Imágenes de entrada: Conjunto Corel200

Número de imágenes = 200

KNN como motor de clasificación en la base de conocimiento,  $K = 5$

Número de técnicas de preprocesamiento = 11

Número de técnicas de segmentación = 6

Número de clases  $11 \times 6 = 66$

Umbral de referencia, métrica *accuracy* = 0.9

Umbral de referencia, métrica CFHu = 0.9

Resultados obtenidos:

En el experimento 1 se observó que la cantidad de segmentaciones clase cero fue elevada; por esta razón se realizó una inspección visual de los casos encontrados por debajo del umbral de la métrica CFHu para determinar si las disimilitudes eran perceptibles, se encontraron segmentaciones aceptables visualmente por debajo del umbral de CFHu y donde la métrica *accuracy* los calificaba por arriba del umbral.

Para verificar lo anterior se experimentó clasificando todas las imágenes, permitiendo el paso a todos los casos bajo el umbral de 0.9 (de estos se llevó un conteo) lo cual significó que todas tendrían una clase distinta a cero asignada al final del procesamiento (excepto los casos donde CFHu fuera igual a cero, condición que no sucedió).

Este proceso se realizó con las métricas CFHu y *accuracy*, utilizadas como criterio de similitud entre las segmentaciones generadas por el modelo y el *ground truth*. Los resultados así obtenidos se analizaron estableciendo un comparativo.

Las clasificaciones resultantes mejoraron con la métrica *accuracy*, los casos clase cero (abajo del umbral 0.9) se redujeron considerablemente como se muestra en la Tabla 4.5.

Tabla 4.5 Comparativo de clasificaciones bajo las métricas CFHU y *accuracy*.

Imágenes clasificadas	Clasificaciones bajo la métrica CFHu	Clasificaciones bajo la métrica <i>accuracy</i>
Umbral de la métrica	0.9	0.9
Abajo del umbral (clase cero)	153 (76.5%)	73 (36.5%)
Mayor o igual al umbral	47 (23.5%)	127 (63.5%)
Tiempo de procesamiento	2 horas	1.6 horas

Fuente: elaboración propia.

De esta experimento se obtuvo que la métrica ***accuracy*** era la adecuada para el modelo. Se evaluaron otras métricas como la SBD (Yeghiazaryana & Voiculescu, 2019) encontrada en el estado del arte. Además de la métrica *accuracy* y CFHu se implementaron SBD, *F-Measure* e coeficiente Dice, pero los resultados se encontraron por debajo de los obtenidos con *accuracy*. En la Tabla 4.6 se muestra el promedio del valor de las métricas CFHu, SBD, *accuracy*, *F-Measure* y coeficiente Dice sobre el conjunto de imágenes Corel200.

Tabla 4.6 Promedio de las métricas evaluadas sobre el conjunto Corel200.

CFHu	SBD	ACCURACY	F-MEASURE	DICE
0.8215	0.1831	<b>0.9029</b>	0.1770	0.1770

Fuente: elaboración propia.

La métrica determinante para comparar la similitud entre las segmentaciones generadas con el modelo y el *ground truth* fue *accuracy*; la métrica CFHU se utilizó

---

como descriptor de forma para valorar la similitud de la geometría de las segmentaciones. Las métricas *F-Measure* y coeficiente Dice (las cuales son equivalentes) se dejaron como referencia en cuanto a la exactitud de la segmentación únicamente en la etapa de entrenamiento (en un trabajo futuro podrían ser útiles); la métrica SBD quedó descartada.

### Experimento 3

Se realizó bajo las siguientes condiciones:

Imágenes de entrada: Conjunto Corel200

Número de imágenes = 200

Número de técnicas de preprocesamiento = 11

Número de técnicas de segmentación = 6

Número de clases  $11 \times 6 = 66$

Umbral de referencia, métrica *accuracy* = 0.9

Resultados obtenidos:

Se comparó el desempeño del modelo utilizando su base de conocimiento adquirida (BC) con sus algoritmos de clasificación (IB1 y KNN) contra la aplicación de las combinaciones de técnicas de preprocesamiento y segmentación sin utilizar la base de conocimiento. En la Tabla 4.7 se presenta el resumen de los resultados obtenidos.

**Tabla 4.7 Comparativo de desempeño del modelo respecto al uso o no de la base de conocimiento del sistema.**

<b>Imágenes clasificadas</b>	<b>No se utilizó la base de conocimiento (BC)</b>	<b>Utilizando la base de conocimiento (BC) con algoritmo IB1</b>	<b>Utilizando la base de conocimiento (BC) con algoritmo KNN, K=5</b>
<b>Umbral de la métrica</b>	0.9	0.9	0.9
<b>Abajo del umbral (clase cero)</b>	73 (36.5%)	73 (36.5%)	15 (7.5%)
<b>Mayor o igual al umbral</b>	127 (63.5%)	127 (63.5%)	185 (92.5%)
<b>Tiempo de procesamiento</b>	2.24 horas	1.47 horas	1.24 horas

Fuente: elaboración propia.

---

Derivado del experimento 3 se encontró que el mejor desempeño se obtiene con la **utilización de la base de conocimiento y el algoritmo KNN (K = 5)** como clasificador, para seleccionar las técnicas de preprocesamiento y segmentación apropiadas para segmentar las imágenes durante el entrenamiento.

### Experimento 4

El experimento se realizó bajo las siguientes condiciones:

Imágenes de entrada: Conjunto Corel200

Número de imágenes = 200

KNN como motor de clasificación en la base de conocimiento,  $K = 5$

Umbral de referencia, métrica *accuracy* = 0.8, 0.9 y 0.95

Número de técnicas de preprocesamiento = 11

Número de técnicas de segmentación = 6

Número de clases  $11 \times 6 = 66$

Resultados obtenidos:

Se evaluó el efecto del valor de la métrica en sobre el rendimiento del modelo, en la Tabla 4.8 se muestra el resumen de resultados.

**Tabla 4.8 Comparativa de desempeño al variar el valor del umbral de la métrica *accuracy*.**

Imágenes clasificadas	Valor de la métrica <i>accuracy</i>		
	0.8	0.9	0.95
<b>Abajo del umbral (clase cero)</b>	3 (1.5%)	15 (7.5%)	97 (48.5)
<b>Mayor o igual al umbral</b>	197 (98.5%)	185 (92.5%)	103 (51.5%)
<b>Tiempo de procesamiento</b>	1.09 horas	1.24 horas	1.58 horas

Fuente: elaboración propia.

La consecuencia de disminuir el umbral de la métrica fue una reducción en el número de etiquetados “clase cero” y un menor tiempo de procesamiento, esto en

---

destruimiento de la calidad de la segmentación; sucedió lo contrario al incrementar el umbral. Lo anterior implicó una relación directa entre el incremento del umbral de la métrica, el aumento del número de etiquetados clase cero y el tiempo de procesamiento.

### Experimento 5

El experimento se realizó bajo las siguientes condiciones:

Imágenes de entrada: Conjunto Corel200

Número de imágenes = 200, 180 de entrenamiento, 20 de prueba.

KNN como motor de clasificación en la base de conocimiento,  $K = 5$

Umbral de referencia, métrica *accuracy* = 0.9

Número de técnicas de preprocesamiento = 11

Número de técnicas de segmentación = 6

Número de clases  $11 \times 6 = 66$

Resultados obtenidos:

Se evaluó el modelo en sus dos fases, primero se entrenó tomando el 90% de las imágenes del conjunto Corel200 (180 del total) y el 10% restante se aplicó en la fase de prueba (20 del total).

Durante la fase de entrenamiento el supervisor del modelo implementado en el sistema registra las secuencias aplicadas a cada imagen hasta encontrar la que mejor se adapte para segmentarla adecuadamente.

Para la primer imagen se aplicaron todas las técnicas hasta encontrar la que maximizó el valor de la métrica partiendo del mínimo establecido (0.9), Figura 4.2.

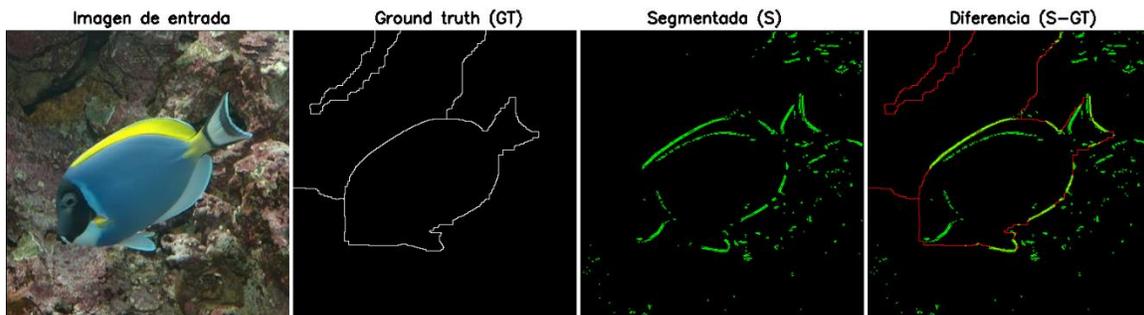


Figura 4.2: La imagen 100.png fue la primera en ser segmentada por el sistema durante el entrenamiento.

El supervisor del sistema registró la aplicación de las 66 combinaciones de técnicas y encontró que la mejor secuencia para segmentar la imagen 100.png fue:

0.953901 (accuracy): [ FILTRO MEDIANA\_K7, LAPLACIANO ]: Clase (10)

El sistema indicó un valor de métrica máximo logrado (0.953901), la secuencia de técnicas aplicadas (FILTRRO MEDIANA\_K7, LAPLACIANO) y la clase asignada (10). En el Anexo B se presenta en detalle el registro de las técnicas aplicadas a cada una de las imágenes del conjunto.

Por otra parte, en la Figura 4.3 se muestra la distribución en porcentaje de los colores dominantes de la imagen 100.png, la información es proporcionada por el *descriptor visual* de color dominante (DCD). Se observó alto contraste en los colores dominantes lo que facilita la segmentación cuando la imagen se transforma a una representación en escala de grises.

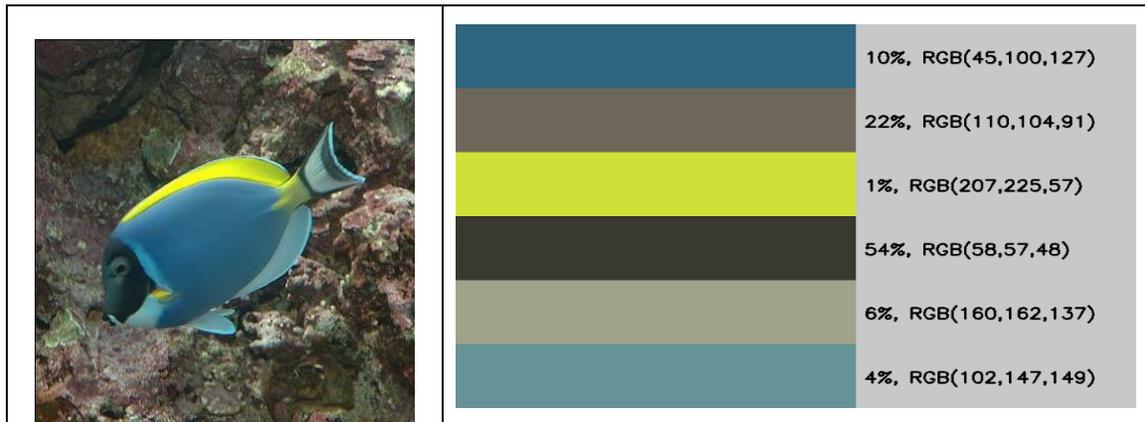


Figura 4.3 Proporciones de la distribución de colores dominantes en la imagen de entrada 100.png.

El proceso anterior se repite con cada imagen durante el proceso de entrenamiento.

Una vez terminado el entrenamiento se aplicaron las imágenes de prueba para evaluar el modelo en fase de utilización.

En la Tabla 4.9 se resumen los resultados de las clasificaciones obtenidas con el modelo entrenado. La secuencia recomendada por los cuatro clasificadores del modelo, para la mayoría de las imágenes fue la combinación:

[FILTRO MEDIANA\_K7, GRADIENTE MORFOLOGICO].

Tabla 4.9 Comparativa de las clasificaciones del subconjunto de prueba obtenidas con cuatro clasificadores.

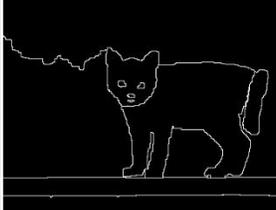
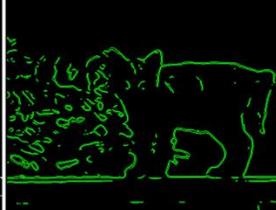
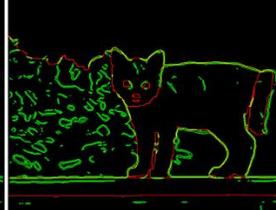
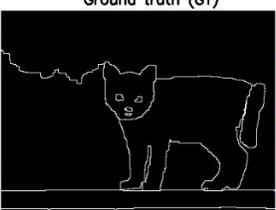
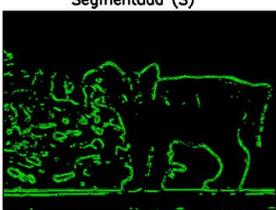
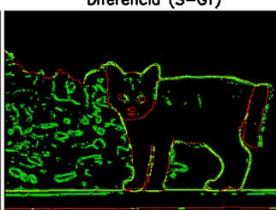
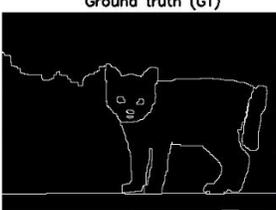
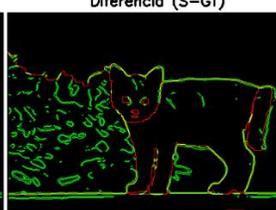
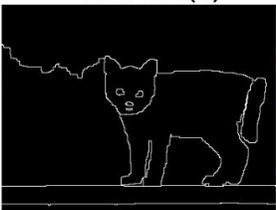
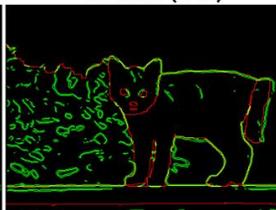
Clasificador	Clasificaciones mayores o igual al umbral 0.8	Secuencia de técnicas predominante recomendada
KNN	13 (65%)	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO
SVM LINEAL	8 (4%)	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO
SVM POLINOMIAL	20 (100%)	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO
SVM RADIAL	20 (100%)	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO

Fuente: elaboración propia.

Los mejores resultados se lograron con el clasificador máquina vector soporte (SVM) de kernel polinomial y radial, el de kernel lineal fue el de menor desempeño seguido por KNN. En la columna de clasificaciones se muestran las cantidades logradas con su porcentaje relativo al total de imágenes (20).

En la Tabla 4.10 se presentan los resultados del procesamiento de la imagen de prueba 11.png, tomada como muestra, se observan: la imagen original, su *ground truth (GT)*, el resultado de la segmentación realizada por el modelo en fase de aplicación (S) y su diferencia contra el *ground truth* (contorno en rojo), los píxeles coincidentes fueron resaltados en contorno amarillo y la segmentación (S) en contorno verde.

Tabla 4.10 Comparación entre las segmentaciones producidas con técnicas recomendadas por los clasificadores KNN y SMV con *kernel* lineal, polinomial y de base radial para la imagen 11.png.

	Imagen de entrada	Ground truth (GT)	Segmentada (S)	Diferencia (S-GT)
KNN				
SVM LINEAL				
SVM POLINOMIAL				
SVM BASE RADIAL				

Fuente: imágenes tomadas del conjunto Corel200 (B., L., Mou, & M.H., Submitted, 2015). Elaboración propia.

Se aplicaron los clasificadores: KNN ( $K = 5$ ), máquina de soporte vectorial (SVM) con *kernel* lineal, polinomial (grado 5) y base radial.

En la Tabla 4.11 se presentan las secuencias de técnicas de segmentación reportadas por el ejecutor del sistema, indicando cuáles son las recomendadas por cada clasificador y aplicadas en cada caso a la imagen 11.png y el valor de la métrica de similitud logrado con su aplicación.

**Tabla 4.11 Recomendación de técnicas de segmentación para la imagen 11.png con sus respectivos valores de la métrica de similitud.**

Clasificador	Secuencia de técnicas recomendada	Métrica accuracy	Métrica CFHu
<b>KNN</b>	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO	0.893572	0.864468
<b>SVM LINEAL</b>	FILTRO MEDIANA_K7, LAPLACIANO	0.872618	0.828671
<b>SVM POLINOMIAL</b>	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO	0.893572	0.864468
<b>SVM RADIAL</b>	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO	0.893572	0.864468

Fuente: elaboración propia.

Las secuencias de técnicas recomendadas por el modelo para otras imágenes procesadas se muestran en la Tabla 4.12.

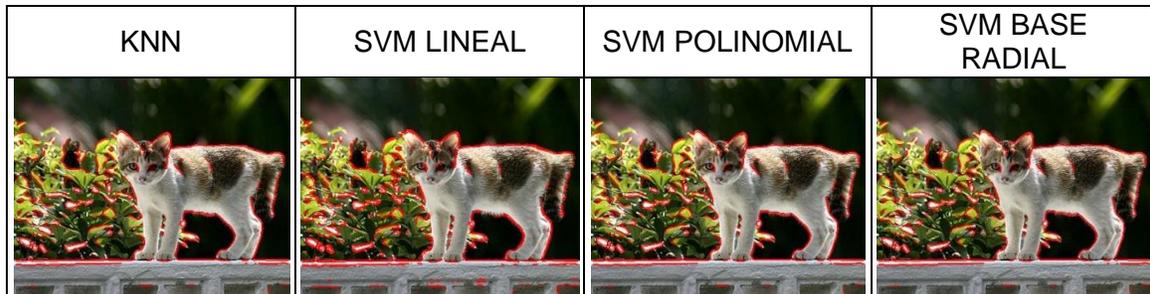
**Tabla 4.12 Secuencias de segmentación para otras imágenes procesadas.**

Imagen	KNN	SVM LINEAL	SVM POLINOMIAL	SVM RADIAL
<b>1.png</b>	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO			
<b>2.png</b>	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO	NO SEGMENTADA	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO
<b>3.png</b>	NO SEGMENTADA	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO
<b>4.png</b>	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO			

Fuente: elaboración propia.

Por otra parte, el sistema generó la superposición de la segmentación sobre la imagen original 11.png, en la Tabla 4.13 se observa una comparativa de las mismas.

**Tabla 4.13 Superposición de la segmentación sobre la imagen original de entrada al modelo.**



Fuente: imágenes tomadas del conjunto Corel200 (B., L., Mou, & M.H., Submitted, 2015). Elaboración propia.

Se concluyó que el sistema funcionaba en sus dos fases (entrenamiento y utilización). El elemento supervisor registra y reporta las acciones del modelo en la fase de entrenamiento, tales como: valores de las métricas, las combinaciones de técnicas aplicadas, modelos generados por los clasificadores, registro de los *descriptores visuales MPEG-7* extraídos de las imágenes, las características de las imágenes segmentadas (por debajo e igual o mayor umbral), directorios con las imágenes generadas durante todo el proceso.

El ejecutor registra y reporta las acciones del modelo en la fase de utilización, como: predicciones de clase de los modelos generados en la fase de entrenamiento (KNN y SVM con kernel lineal, polinomial y base radial, métricas de las imágenes segmentadas, combinación de técnicas de segmentación de acuerdo con la recomendación generada por los clasificadores.

En la Figura 4.4 se muestran los distintos archivos y directorios con imágenes generados durante las fases de entrenamiento y utilización del modelo.

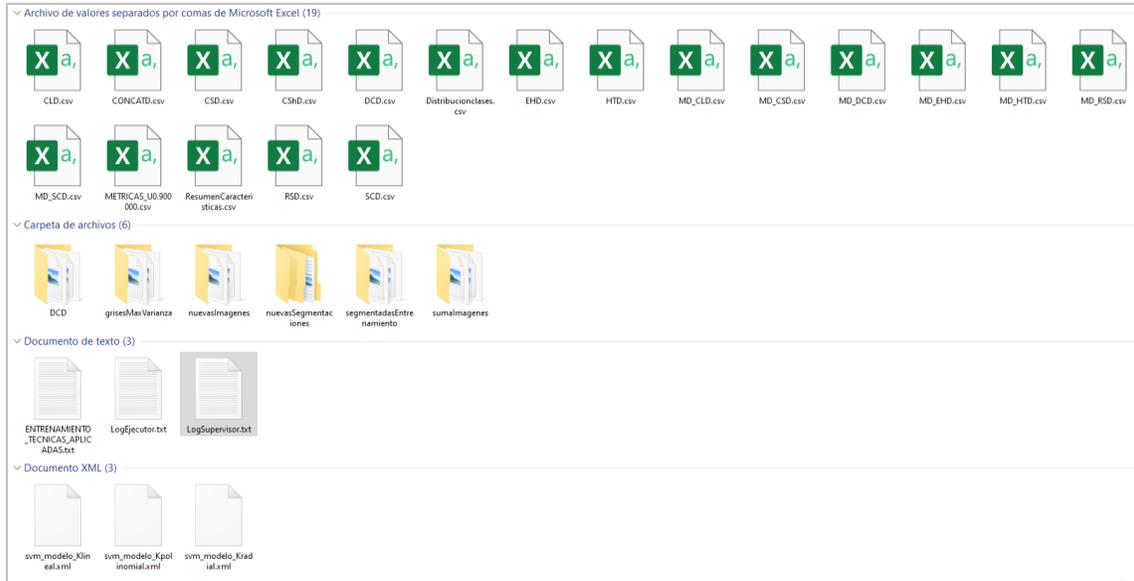


Figura 4.4 Archivos y directorios generados por el sistema en las fases de entrenamiento y utilización.

### Experimento 6

El experimento se realizó bajo las siguientes condiciones:

Imágenes de entrada: Conjunto *Caltech 101: Motorbikes*

Número de imágenes = 330, 300 de entrenamiento, 30 de prueba.

KNN como motor de clasificación en la base de conocimiento,  $K = 5$

Umbral de referencia, métrica *accuracy* = 0.85

Número de técnicas de preprocesamiento = 11

Número de técnicas de segmentación = 6

Número de clases  $11 \times 6 = 66$

Resultados obtenidos:

El conjunto *Motorbikes*<sup>1</sup> contiene imágenes de la categoría motocicletas. El modelo se entrenó con 300 imágenes del conjunto y se realizó la prueba con 30 imágenes adicionales. En la Tabla 4.14 se resumen los resultados obtenidos con los distintos clasificadores después de entrenar el modelo.

**Tabla 4.14** Secuencia de técnicas recomendadas por los clasificadores sobre las imágenes de prueba.

Clasificador	Clasificaciones mayores o igual al umbral 0.8	Secuencia de técnicas recomendadas para la segmentación
<b>KNN</b>	30 (100%)	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO
<b>SVM LINEAL</b>	1 (3.3%)	FILTRO MEDIANA_K7, LAPLACIANO
<b>SVM POLINOMIAL</b>	30 (100%)	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO
<b>SVM RADIAL</b>	30 (100%)	FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO

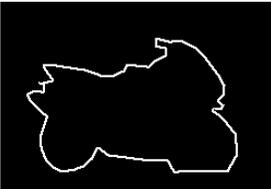
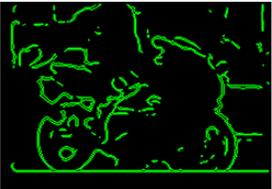
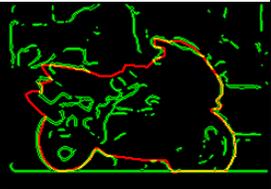
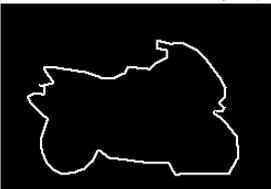
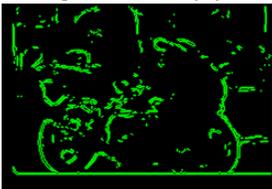
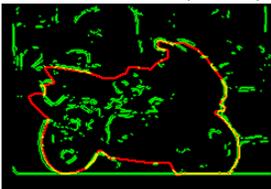
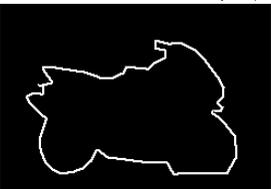
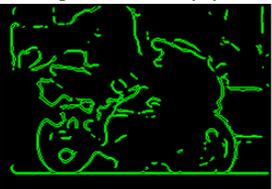
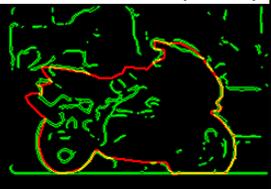
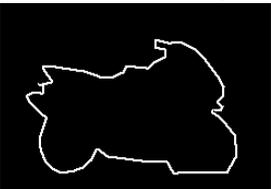
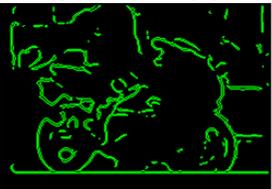
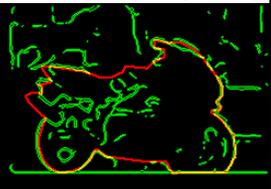
Fuente: elaboración propia.

Los clasificadores KNN, SVM polinomial y base radial coincidieron en la recomendación de la misma secuencia de técnicas de segmentación (clase 11)

<sup>1</sup> Se puede obtener una copia del conjunto *Motorbikes* en <http://www.robots.ox.ac.uk/~vgg/data/bicos/>

para las 30 imágenes de prueba; el clasificador SVM lineal solo recomendó una secuencia diferente (clase 10) para la imagen 00311.png y para las 29 restantes indicó no poder segmentarlas. En la Tabla 4.15 se muestran las segmentaciones de la imagen 00311.png producidas con las secuencias de técnicas recomendadas por cada clasificador.

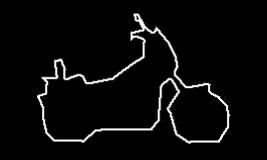
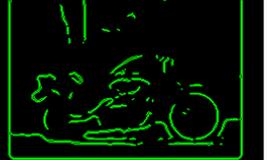
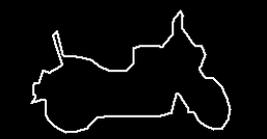
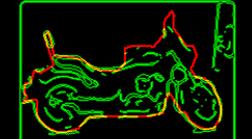
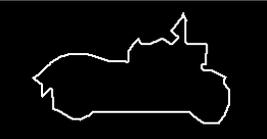
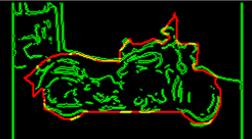
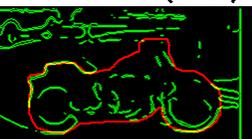
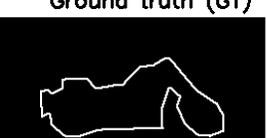
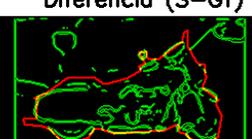
**Tabla 4.15 Segmentaciones producidas con las técnicas recomendadas por los clasificadores KNN y SMV con *kernel* lineal, polinomial y de base radial sobre la imagen 00311.png.**

	Imagen de entrada	Ground truth (GT)	Segmentada (S)	Diferencia (S-GT)
KNN				
SVM LINEAL				
SVM POLINOMIAL				
SVM BASE RADIAL				

Fuente: imágenes tomadas del conjunto *Motorbike* (Fei-Fei, Fergus, & Perona, 2004). Elaboración propia.

En la Tabla 4.17 se presentan otras imágenes de prueba segmentadas por la misma secuencia de técnicas sugeridas por KNN, SVM polinomial y base radial.

Tabla 4.16 Segmentaciones producidas con las mismas técnicas recomendadas por los clasificadores KNN y SMV con *kernel* polinomial y de base radial sobre las imágenes de prueba.

	Imagen de entrada	Ground truth (GT)	Segmentada (S)	Diferencia (S-GT)
00301.png				
00307.png				
00310.png				
00312.png				
00324.png				
00329.png				

Fuente: imágenes tomadas del conjunto *Motorbike* (Fei-Fei, Fergus, & Perona, 2004). Elaboración propia.

### Experimento 7

El experimento se realizó bajo las siguientes condiciones:

Imágenes de entrada: Conjunto BSD300

Número de imágenes = 300

KNN como motor de clasificación en la base de conocimiento,  $K = 5$

Umbral de referencia, métrica *accuracy* = 0.8

Número de técnicas de preprocesamiento = 11

Número de técnicas de segmentación = 6

Número de clases  $11 \times 6 = 66$

Resultados obtenidos:

Se procesó el conjunto BSD300 con el modelo implementado, se presenta la información global de características resumida en la Tabla 4.17. Se muestran los valores promedio de parámetros de brillo, contraste, entropía, colores dominantes y coherencia espacial (de colores dominantes) y valores de métricas de similitud.

Mediante un proceso de análisis de la información obtenida se determinó qué caracteriza a la clasificación de las segmentaciones producidas después del entrenamiento.

Tabla 4.17 Valores promedio de las características de las imágenes obtenidas en el entrenamiento con el conjunto BSD300.

Imágenes clasificadas		Brillo (Media)	Contraste (Desviación estándar)	Entropía	Colores Dominantes (DCD)	Coherencia Espacial (DCD)	CFHu	Accuracy
<b>Mayor o igual al umbral</b>	247 (82.33%)	114.81	58.76	7.2591	6	<b>15.61</b>	0.797972	0.863491
<b>Abajo del umbral (clase cero)</b>	38 (17.66%)	101.33	57.17	7.4276	6	<b>9.65</b>	0.788715	0.762191

Fuente: elaboración propia.

La interpretación de la información general obtenida del sistema muestra que las imágenes que segmentaron con una métrica igual o mayor al umbral tienen un mayor contraste comparadas con las que no lo superaron, esto se determinó por los valores la varianza y los parámetros del *descriptor visual* color dominante (DCD).

El descriptor de color dominante del MPEG-7 (DCD) proporciona el número de colores dominantes y la coherencia espacial promedio de esos colores; se observó mayor coherencia espacial en el grupo de imágenes segmentadas satisfactoriamente (15) comparada con las que no (8). Lo que significa que las imágenes de coherencia más alta tienen mejor contraste y, por lo tanto, objetos distinguibles para su segmentación.

La métrica CFHu, indica que en promedio las segmentaciones obtenidas por el modelo, en términos de forma, son muy similares al *ground truth* correspondiente.

¿Qué características particulares tienen las imágenes segmentadas adecuadamente comparadas con las que no?

En la Figura 4.5 se observa la imagen original 229.png, su *ground truth*, el resultado de la segmentación realizada por el modelo y su comparación contra el *ground truth*. Esta imagen fue segmentada adecuadamente con un métrica de similitud de 0.965629, la cual es la máxima de todo el conjunto de imágenes procesadas.

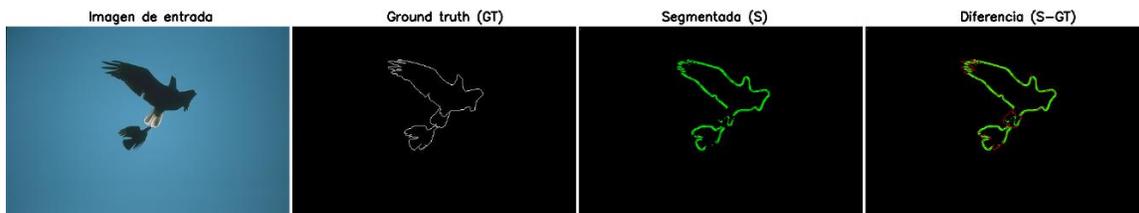


Figura 4.5 Imagen 229.png segmentada comparada con el *ground truth* de la imagen original de entrada.

Para segmentar la imagen 229.png, el sistema generó la recomendación de técnicas siguientes:

Imagen: 229.png

Combinación de técnicas aplicadas = 1

Clase (1): [ FILTRO MEDIANA\_K7, LAPLACIANO ]: 0.965629 (accuracy)

En este caso el uso de la base de conocimiento permitió que el sistema sólo aplicara una combinación y con esta combinación superó el umbral mínimo establecido.

En la Figura 4.6, se muestra la distribución en porcentaje de los colores dominantes de la imagen correspondiente, la información es proporcionada por el contenido semántico del *descriptor visual* de color dominante (DCD). Se observa alto contraste en los dos colores dominantes y una coherencia de color de 30, lo

que significa que los colores están altamente agrupados, esto se verifica al observar la imagen.

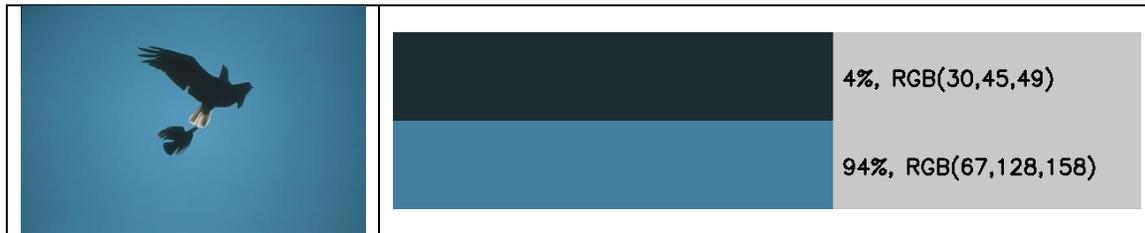


Figura 4.6 Proporciones de la distribución de colores dominantes en la imagen de entrada 229.png.

En la Figura 4.7 se observa la superposición de la imagen original y su correspondiente segmentación.



Figura 4.7 Superposición de la segmentación en la imagen original 229.png.

En la Figura 4.8 se presenta el histograma de la distribución promedio de los cinco tipo de bordes en la imagen 229.png provista por el *descriptor visual* EHD. Se observa que en promedio existen dos tipos de bordes dominantes definidos sobre la imagen, en este caso bordes a 45 grados y bordes no direccionales en una cantidad pequeña, lo que implica que la textura de la imagen no es muy compleja o entramada.

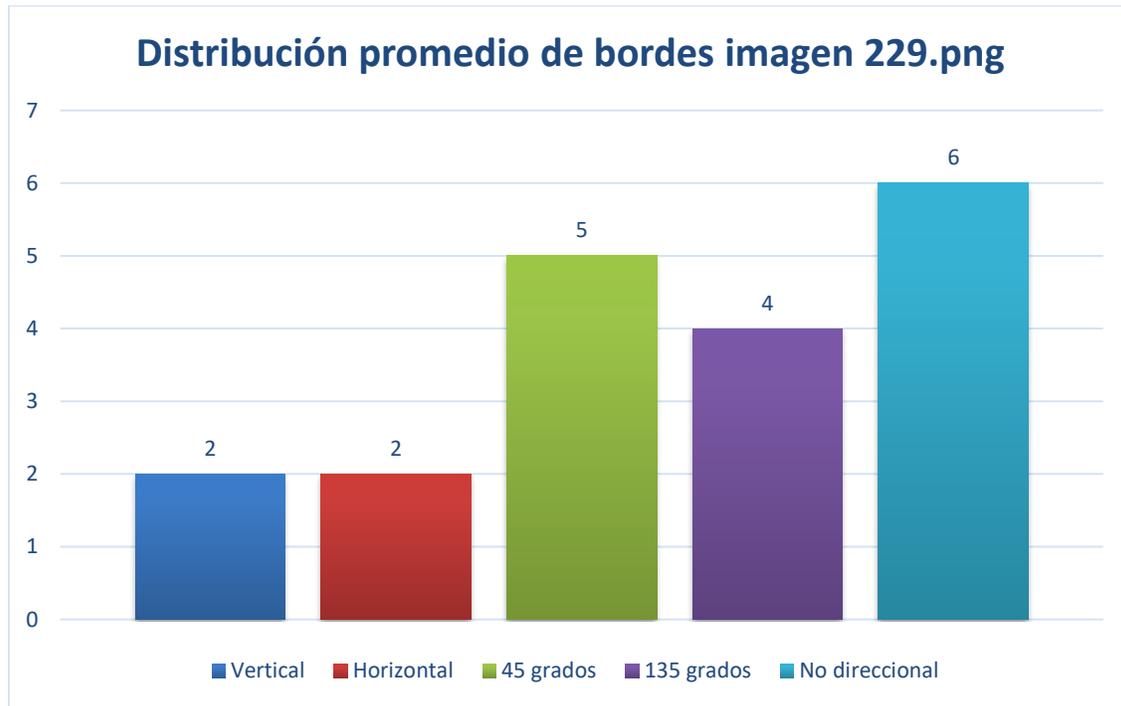


Figura 4.8 Distribución promedio de bordes en la imagen 229.png, descriptor EHD.

Por otra parte, en la Figura 4.9 se observan la imagen original 225.png, su *ground truth*, el resultado de la segmentación realizada por el modelo y su comparación contra el *ground truth*. La imagen original fue segmentada pero no adecuadamente, su métrica de similitud de 0.712437, abajo del umbral, fue la mínima de todo el conjunto BSD300 procesado.

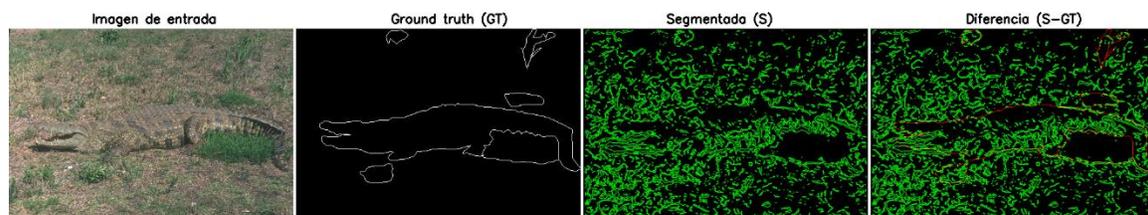


Figura 4.9 Imagen 225.png segmentada comparada con el *ground truth* de la imagen original de entrada y su diferencia.

En la distribución de colores dominantes (3 colores) de la Figura 4.10 se observa que existe poco contraste entre ellos, eso significa que los operadores de

detección de bordes no lograron distinguir bordes claros, además de tener una coherencia de 4, es decir están dispersos los colores, poco agrupamiento en de objetos en la imagen.



Figura 4.10 Proporciones de la distribución de colores dominantes en la imagen de entrada 225.png.

En la Figura 4.11 se observa la dificultad que tuvo el sistema para segmentar la imagen debido al bajo contraste y presencia de texturas.



Figura 4.11 Superposición de la segmentación en la imagen original 225.png.

En la Figura 4.12 se presenta el histograma de la distribución promedio de los cinco tipo de bordes en la imagen 225.png provista por el *descriptor visual* EHD.

Se observó que en promedio existe gran cantidad de bordes en cuatro de los cinco tipos de bordes dominantes sobre la imagen, lo que implicó que la textura de la imagen fuera muy compleja o entramada, como se detectó en la inspección visual de la Figura 4.11.

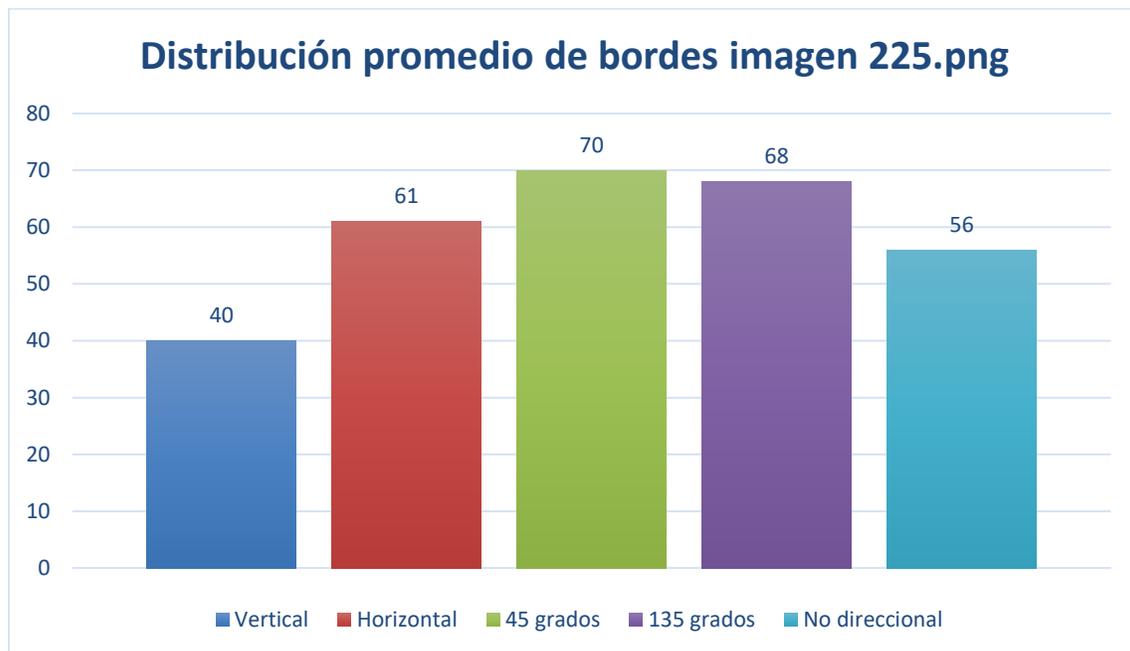


Figura 4.12 Distribución promedio de bordes en la imagen 225.png, descriptor EHD

Los imágenes analizadas anteriormente representaban los casos extremos de los valores máximo y mínimo de la métrica obtenidos al procesar el conjunto de imágenes. ¿Qué sucedería con los dos casos que estuvieran en la frontera de transición de la métrica? En la Tabla 4.18 se indican los nombres de las imágenes en los extremos de valores métricos y las dos de la frontera de transición.

Tabla 4.18 Imágenes 225.png y 229.png con valores métricos extremos, imágenes 150.png y 7.png en la frontera de transición del valor mínimo de la métrica (0.8).

Mínimo valor	Frontera de transición del umbral de la métrica		Máxima valor
0.712437 225.png	(...) (...)	0.797939 150.png	0.802047 7.png (...) 229.png

Fuente: elaboración propia.

Se repitió el mismo análisis sobre el procesamiento de las imágenes 150.png y 7.png. En la Figura 4.13 se observa la segmentación de la imagen 150.png.

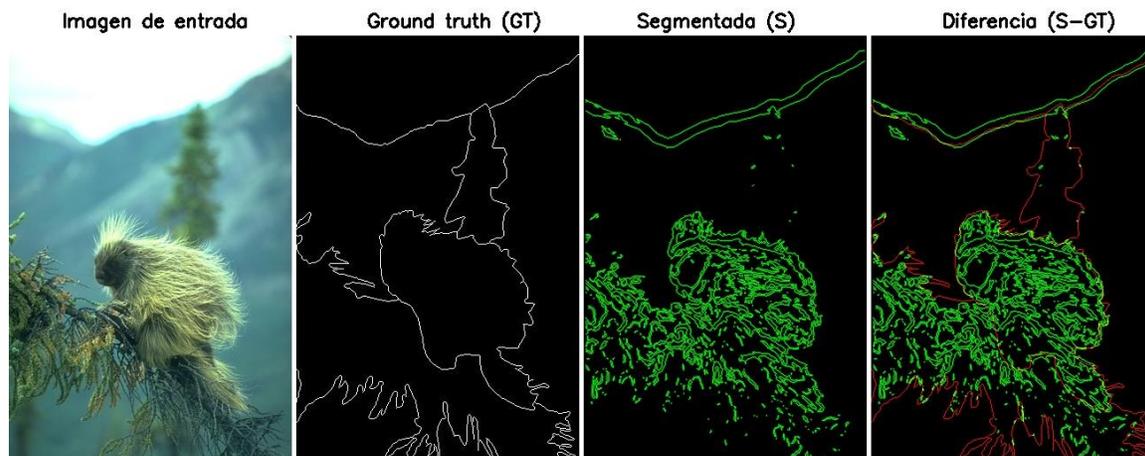


Figura 4.13 Segmentación de la imagen 150.png

En la distribución de colores dominantes (8 colores) de la Figura 4.14 se observa que existe bajo contraste entre ellos, eso significa que los operadores de detección de bordes no lograron distinguir bordes claros, pero con una coherencia de 19, los colores presentaban buen agrupamiento en objetos de la imagen. Es decir, el bajo contraste se compensó con la coherencia alta.



Figura 4.14 Proporciones de la distribución de colores dominantes en la imagen de entrada 150.png.

En la Figura 4.15. se muestra la segmentación del imagen 7.png

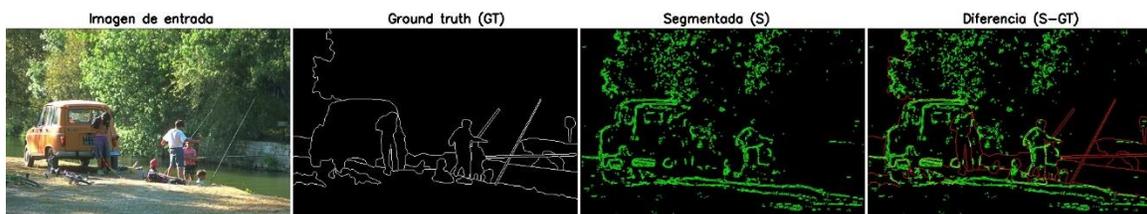


Figura 4.15 Segmentación de la imagen 7.png.

La distribución de colores dominantes (8) de la imagen 7.png se presenta en la Figura 4.16, se observa que existe buen contraste entre ellos, eso facilitó la detección de los bordes, aunque los colores no presentaban buen agrupamiento en objetos de la imagen debido a la baja coherencia (1). Esta se compensó con un contraste aceptable.

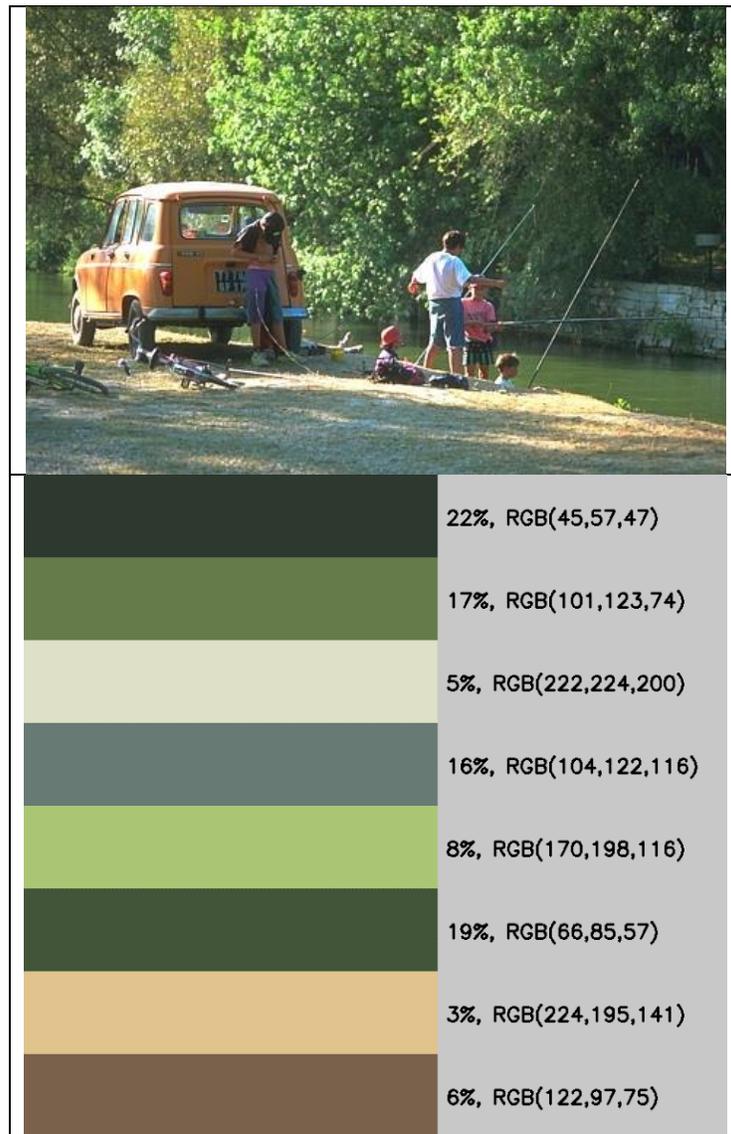


Figura 4.16 Proporciones de la distribución de colores dominantes en la imagen de entrada 7.png.

Por último, la distribución promedio de los bordes proporcionada por el descriptor de textura EHD, presentó una elevada cantidad de bordes (textura) por lo menos en cuatro de los cinco tipos de bordes que detecta el descriptor, EHD, como se observa en la Figura 4.17 y en la Figura 4.18.

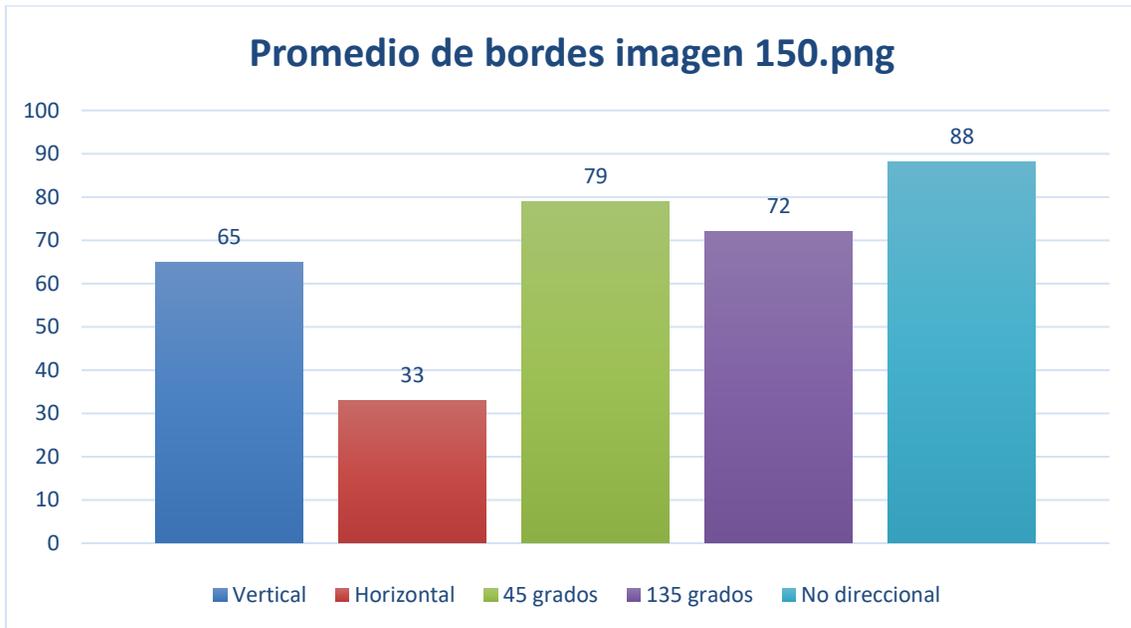


Figura 4.17 Distribución promedio de bordes en la imagen 150.png.

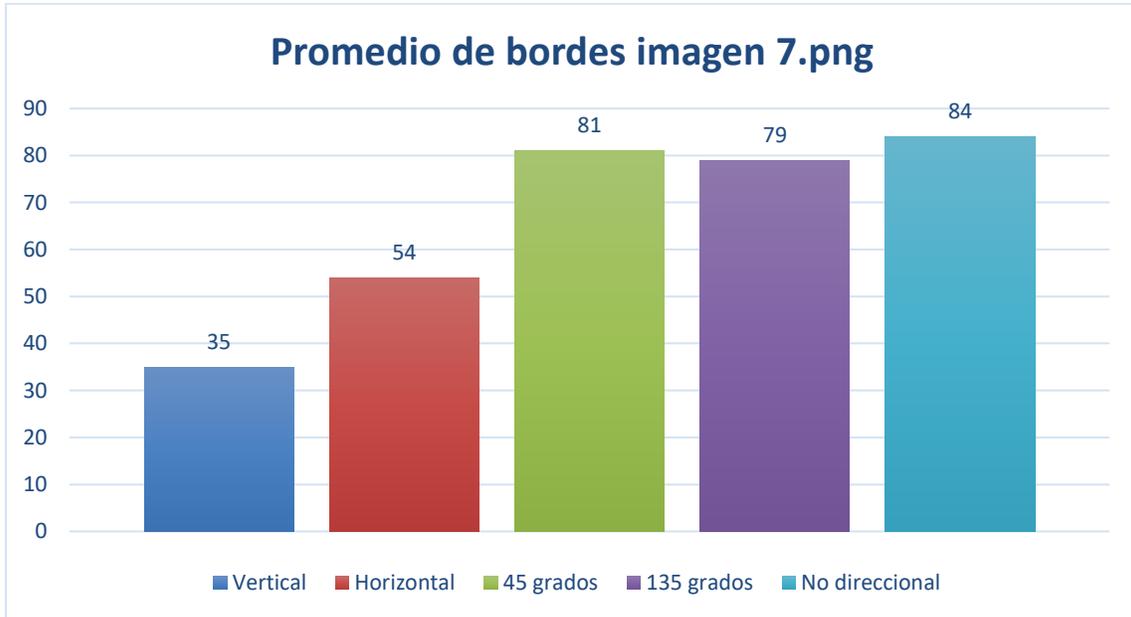


Figura 4.18 Distribución promedio de bordes en la imagen 7.png, descriptor EHD

La Tabla 4.19, es una comparativa resumida de las cuatro imágenes analizadas con base en sus características obtenidas mediante los *descriptores visuales* y estadísticos. Las diferencia entre los valores de los descriptores de las imágenes 225.png y 229.png son perceptibles mientras que la diferencia entre los valores de las imágenes 150.png y 7.png es más sutil, eso se observó por tener valores de la métrica *accuracy* próximos al umbral mínimo (0.8), la diferencia se notó en los valores de contraste y del descriptor de forma CFHu, la segmentación de la imagen 7.png se aproximó más a la forma de su *ground truth* (CFHu = 0.852341) en comparación con la imagen 150.png (CFHu = 0.774283).

**Tabla 4.19 Comparativa de características entre las imágenes 225.png, 150.png, 7.png, 229.png.**

Imágenes	Brillo (Media)	Contraste (Desviación estándar)	Entropía	Colores Dominantes (DCD)	Coherencia Espacial (DCD)	CFHu	Accuracy	Clase asignada
225.png	119	35	7.15	3	4	0.677771	0.712437	0
150.png	117	70	6.90	8	19	0.774283	0.797939	0
7.png	101	57	7.48	8	1	0.852341	0.802047	10
229.png	153	27	6.10	2	30	0.906788	0.965629	10

Fuente: elaboración propia.

### 4.3 Discusión

Los resultados obtenidos en la implementación del modelo muestran que éste una vez entrenado, es capaz de adaptar la aplicación de sus técnicas de preprocesamiento y segmentación al procesamiento de nuevas imágenes de entrada dependiendo de las características obtenidas mediante *descriptores visuales MPEG-7*.

Se encontró que la implementación del modelo es sensible a la selección de la métrica definida como parámetro de similitud entre la segmentación y el *ground truth*. Algunas métricas tienen una definición estricta, donde la calificación de similitud se basa en la concordancia espacial entre los píxeles de la imagen segmentada y la de referencia (métricas de traslape como: índice *Dice*, *F-Measure* entre otras), otras utilizan la similitud con base en sus características geométricas (como CFHu), la elección de la métrica es determinante en la capacidad selectiva del modelo para elegir las secuencias adecuadas para segmentar cada imagen.

El modelo también es sensible a las imágenes con bajo contraste o con texturas demasiado entramadas, esto es una limitante de la manera en cómo operan las técnicas de segmentación por discontinuidades.

Existe un compromiso entre la selección del umbral mínimo de la métrica, el tiempo de entrenamiento y la calidad de la segmentación.

En Tabla 4.20 se dio respuesta a las preguntas de investigación establecidas en el planteamiento del problema (pág. 28) con base en los resultados obtenidos.

Tabla 4.20 Respuestas a las preguntas de investigación planteadas inicialmente.

<b>Pregunta de investigación / Respuesta</b>
<p data-bbox="302 405 1373 583"><b>1. ¿Será posible establecer un modelo con un mecanismo de aprendizaje automático que le permita adaptarse a los diferentes tipos de imágenes de entrada, partiendo de un conjunto predefinido de técnicas de preprocesamiento y segmentación?</b></p> <p data-bbox="302 625 1373 709">R.- Sí fue posible establecer un modelo, la implementación de este con buenos resultados fue con el algoritmo de aprendizaje KNN.</p>
<p data-bbox="302 762 1373 993"><b>2. ¿Será posible diseñar un modelo que pueda aprender a segmentar distintas clases de imágenes, a partir de procesar imágenes de ejemplo mediante la aplicación de un conjunto predefinido de técnicas de preprocesamiento y segmentación, de forma tal, que le permita adaptarse a los diferentes tipos de imágenes de entrada?</b></p> <p data-bbox="302 1035 1373 1318">R.- Sí, es posible, se utilizaron las características de imagen como referencia para asociarlas con las técnicas de segmentación exitosas, estas relaciones fueron las que permitieron entrenar al modelo mediante algoritmos de aprendizaje supervisado (con base en ejemplos) para que aprendiera y reconociera a las imágenes con características similares (descriptores) y así poder emitir una recomendación de las técnicas de segmentación para cada tipo de imagen.</p>
<p data-bbox="302 1356 1373 1486"><b>3. ¿Existirá alguna relación o patrón característico entre las imágenes segmentadas adecuadamente y sus correspondientes técnicas aplicadas para lograrlo?</b></p> <p data-bbox="302 1560 1373 1745">R.- Se encontraron algunas relaciones o patrones característicos en las imágenes a través de sus descriptores visuales MPEG-7 de color dominante (DCD) y de histograma de borde (EHD) y de los descriptores estadísticos; entropía, media y desviación estándar, entre otros.</p>

Tabla 4.20 Respuestas a las preguntas de investigación planteadas inicialmente. Continuación.

Pregunta de investigación / Respuesta
<p><b>4. ¿Existirán técnicas de preprocesamiento y segmentación comunes entre imágenes de la misma categoría?</b></p> <p>R.- Sí, en el caso de que las categorías de imágenes compartan características similares de color, forma y textura. Se debe diferenciar entre la división categórica realizada bajo la percepción humana (categorías tipo: auto, casa, persona, entre otras, las cuales son de alto nivel semántico), y, las características intrínsecas de las imágenes extraídas a través de los descriptores <b>visuales (las cuales son de bajo nivel) y estadísticos.</b></p>
<p><b>5. ¿Existirán características que describan a las imágenes y que den algún indicio sobre las técnicas que se les debería aplicar para segmentarlas adecuadamente?</b></p> <p>R.- En los experimentos se encontró que las características de color dominante y de coherencia de color proporcionadas por el descriptor DCD; junto con la característica de similitud de forma obtenida mediante descriptores de Hu (CFHu); las características de textura proporcionada por los <i>descriptores visuales</i> EHD, HTD y los descriptores estadísticos; entropía, media y desviación estándar, entre otros. Proporcionan suficiente información para seleccionar las técnicas de segmentación recomendables para una imagen o conjunto de imágenes.</p>
<p><b>6. ¿Podría el modelo informar la forma en que realizó la segmentación de las imágenes en casos de éxito y reportar las características de las imágenes donde no lo logró?</b></p> <p>R.- Sí, el modelo implementado puede registrar la secuencia de técnicas exitosas y no exitosas diferenciadas por el valor de su métrica de similitud. Se guarda la información en archivo para análisis posteriores. Las características se describen a través de los <i>descriptores visuales</i> y estadísticos.</p>

Fuente: elaboración propia.

Tabla 4.20 Respuestas a las preguntas de investigación planteadas inicialmente. Continuación.

Pregunta de investigación / Respuesta
<p data-bbox="298 415 1375 552"><b>7. ¿Sería posible utilizar el modelo para segmentar nuevas imágenes identificadas a través de algunas de sus características intrínsecas obtenidas mediante sus descriptores?</b></p> <p data-bbox="298 625 1375 905"><b>R.-</b> Sí es posible, el elemento clave para seleccionar las técnicas adecuadas para su segmentación son sus <i>descriptores visuales</i>. Estos funcionan como llave primaria para buscar en la base de conocimiento del modelo un patrón similar y, aplicar las técnicas que éste le sugiera, si se tiene el conocimiento suficiente, de lo contrario, se informa que no se posee tal conocimiento y no es posible recomendar técnicas de segmentación para el nuevo caso.</p>

Fuente: elaboración propia.

# Capítulo 5

## Conclusiones y trabajo futuro

### 5.1 Objetivos logrados

En la Tabla 5.1 se resumen los objetivos logrados como resultado de esta investigación.

Tabla 5.1 Objetivos logrados.

OBJETIVOS	CUMPLIMIENTO DEL OBJETIVO
<b>1. Definir el concepto del modelo y diseñar la arquitectura correspondiente para su implementación.</b>	Se diseñó el modelo, el cual está integrado por dos etapas, una de entrenamiento y otra de utilización. La implementación se realizó en C++ con bibliotecas de código abierto.
<b>2. Seleccionar las técnicas de aprendizaje supervisado apropiadas para el modelo.</b>	Se seleccionaron cinco algoritmos de aprendizaje supervisado: IB1 (aprendizaje con base en instancias), KNN (K-vecinos más cercanos), SVM (máquina de soporte vectorial ) con <i>kernel</i> lineal, SVM con <i>kernel</i> polinomial y SVM con <i>kernel</i> de base radial.
<b>3. Seleccionar técnicas de segmentación por discontinuidades adecuadas.</b>	Se seleccionaron siete técnicas de segmentación: Canny, Scharr, Sobel, Laplaciano, segmentación morfológica, Prewitt y transformada difusa $F^{-1}$ .

Fuente: elaboración propia.

Tabla 5.1 Objetivos logrados (continuación).

OBJETIVOS	CUMPLIMIENTO DEL OBJETIVO
<b>4. Seleccionar las herramientas de desarrollo necesarias para la implementación.</b>	Para la implementación del modelo se utilizaron las herramientas: IDE QT, C++ MinGW y las bibliotecas de código abierto OpenCV 4.0.0.
<b>5. Desarrollar las interfaces de ejecutor y supervisor</b>	Los conceptos de ejecutor y supervisor del modelo se implementaron dentro del sistema desarrollado en C++.
<b>6. Definir la base de conocimiento del sistema.</b>	Se definió una base de conocimiento con dos algoritmos de clasificación con base en instancias seleccionables en la implementación: IB1 y KNN.
<b>7. Integrar el modelo de aprendizaje con el de procesamiento de imagen.</b>	Las técnicas de preprocesamiento y de segmentación definidas previamente en el diseño del modelo se integraron con los algoritmos IB1 y KNN en la implementación en <i>software</i> .
<b>8. Probar el uso del conocimiento adquirido y aplicarlo a nuevos casos.</b>	Se realizaron pruebas con un grupo de imágenes extraído del conjunto de imágenes de entrenamiento y con imágenes obtenidas de diversas fuentes, para verificar la funcionalidad del modelo en la segmentación de nuevos casos. El sistema entrega a la salida una composición de la imagen de entrada y su segmentación.

Fuente: elaboración propia.

## 5.2 Productos y aportaciones

### Productos

Ponencia con el tema “**Diseño e Implementación de un Modelo Autoadaptable de Segmentación de Imágenes Digitales por Discontinuidades**” presentada en la Escuela de Inteligencia Artificial y Robótica UTEZ (octubre - 2018).

Ponencia con el tema “**Diseño e Implementación de un Modelo Autoadaptable de Segmentación de Imágenes Digitales por Discontinuidades**” presentada en el 1er coloquio “ARTIFICIAL INTELLIGENCE” realizado en el Instituto Tecnológico de Iguala Guerrero (mayo - 2019).

#### **Presentación de poster en la “Escuela de Inteligencia Artificial y Robótica” (octubre - 2018)**

Se realizó la presentación de un poster en la “Escuela de Inteligencia Artificial y Robótica”, realizada en la UTEZ en el mes de octubre de 2018, en la ciudad de Cuernavaca, Morelos; México.

#### **Presentación de poster en la “Escuela de Inteligencia Artificial y Robótica” (octubre - 2019)**

Se realizó la presentación de un poster en la “Escuela de Inteligencia Artificial y Robótica”, realizada en la UTEZ en el mes de octubre de 2019, en la ciudad de Cuernavaca, Morelos; México.

Los reconocimientos obtenidos y posters presentados en las conferencias se muestran en el Anexo C.

### Aportaciones

1. Se diseñó un modelo con la posibilidad de expandirse para incluir otro tipo de técnicas de segmentación además de la de discontinuidades.
2. Se desarrolló un sistema con una implementación del modelo en una aplicación de computadora para experimentar, evaluar y responder a las preguntas de investigación; teniendo la posibilidad de expansión hacia otras técnicas o reutilizar módulos o bloques de código.
3. Se implementó un algoritmo de detección de bordes con base en la transformada difusa grado 1, F1 (Hodakova, 2014). (El algoritmo requiere optimizarse para mejorar su velocidad de ejecución, por lo que se tomó la decisión de no utilizarlo en los experimentos, sin embargo, quedó integrado en el sistema desarrollado para trabajos futuros).
4. Se implementó una biblioteca en C++ con las métricas de similitud para descriptores del estándar MPEG7.
5. Se actualizó la biblioteca *MPEG7FexLib* de extracción de *descriptores visuales MPEG-7* para que funcionara con OpenCV 4.0. en lugar de la versión 2.2

### 5.3 Conclusiones generales

1. Los hallazgos de esta investigación soportan el cumplimiento del objetivo principal del proyecto de tesis.
2. Se cumplieron los objetivos particulares planteados, tomando en consideración los alcances y limitaciones previstos.
3. Se verificó que el modelo propuesto e implementado puede aprender de los casos ejemplo y seleccionar las técnicas adecuadas para segmentar nuevos casos.
4. Los descriptores seleccionados del estándar MPEG7 son útiles para detectar características de textura color y forma e identificar las imágenes dentro del modelo para aplicar las técnicas de clasificación y reconocimiento de patrones que proporcionen al sistema de visión artificial clásico la adaptabilidad deseada en el objetivo de tesis.

## 5.4 Trabajos futuros

1. Considerar un proceso adicional de generación de *ground truth* en la etapa de utilización, con el fin de procesar los casos rechazados y agregarlos en la etapa de entrenamiento, incrementando la base de conocimiento del modelo, Figura 5.1.

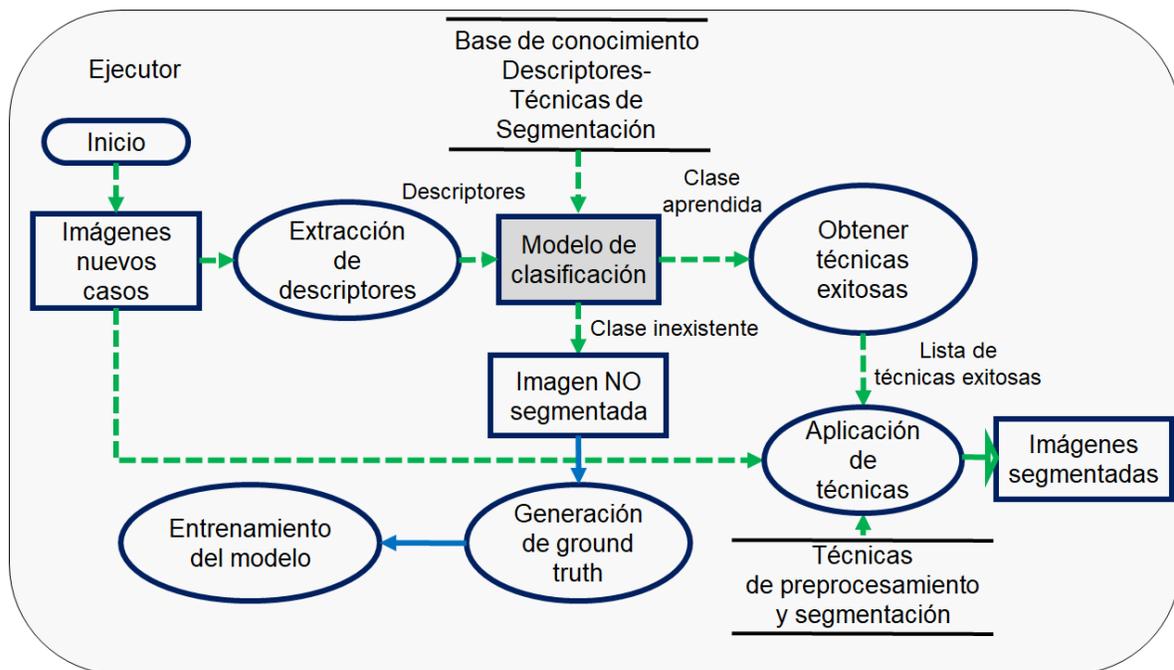


Figura 5.1 Opción de generar el *ground truth* de los nuevos casos para reentrenar el modelo.

2. Buscar en el estado del arte o proponer alguna media de similitud que tome en cuenta las variaciones de forma y de alguna manera la percepción humana, con la suficiente flexibilidad para evaluar los resultados de una segmentación, respecto a un *ground truth*.
3. Afinar los algoritmos de procesamiento y segmentación para mejorar la precisión de las segmentaciones y obtener resultados de las métricas *F-Measure* y coeficiente Dice próximos al valor de umbral.

4. Considerar agregar al modelo la capacidad de ajustar automáticamente los parámetros de las técnicas de preprocesamiento y segmentación utilizadas, con base en los resultados de las segmentaciones obtenidas en la etapa de entrenamiento; de tal manera que sea posible maximizar la métrica de similitud empleada para valorar la segmentación de manera automática y lograr un mejor ajuste en las segmentaciones durante la etapa de utilización.
5. En la implementación del modelo de segmentación agregar otros algoritmos de clasificación que generen modelos de predicción como lo hace la máquina de vector soporte en sus distintas variantes (SVM).
6. Implementar una base de conocimiento más flexible que permita establecer relaciones entre características de la imagen y las técnicas de procesamiento aplicadas, una posibilidad es una base de datos basada en grafos (Borgelt, Kruse, & Steinbrecher, 2009).
7. El modelo se implementó y evaluó con los *descriptores visuales MPEG-7* aplicados de manera global sobre la imagen, se podrían realizar evaluaciones utilizando los descriptores por regiones y extraer información adicional sobre las características de la imagen de manera local.
8. Probar el modelo con otros descriptores adicionales a los MPEG-7.

## Referencias

- Aha, D., Kibler, D., & Albert, M. (1991). Instance-based Learning Algorithms. *Machine Learning*(6), 37-66.
- Ajeshwari, R. N. (2015). Segmentation of Structured Objects in Image. *International Journal of Computer Science and Technology*.
- B., P., L., Z., Mou, X., & M.H., Y. (Submitted, 2015). *Evaluation of Image Segmentation via Exemplar References*.
- Bastan, M., Cam, H., Gudukbay, U., & Ulusoy, O. (July-September de 2010). BiVideo-7: An MPEG-7 Compatible Video Indexing and Retrieval System. *IEEE MultiMedia*, 17, 62-73. Obtenido de [www.cs.bilkent.edu.tr/~bilmdg/bilvideo-7/Software.html](http://www.cs.bilkent.edu.tr/~bilmdg/bilvideo-7/Software.html)
- Béréziat, D., & Herlin, I. (2009). *Solving ill-posed Image Processing problems using Data Assimilation*. Paris: INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE.
- Borgelt, C., Kruse, R., & Steinbrecher, M. (2009). *Graphical models, methods for data analysis and mining*. West Sussex: John Wiley & Sons Ltd.
- Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Cervantes, A. J. (2006). *Metodología para la estructuración y uso de conocimiento en segmentación de imágenes digitales*. Cuernavaca, Morelos: Cenidet.
- Chai, Y., Lempitsky, V., & Andrew, Z. (s.f.). *Visual Geometry Group*. Obtenido de Segmentations for Flower Image Datasets and Others: <http://www.robots.ox.ac.uk/~vgg/data/bicos/>
- Chao, W.-L. (2011). *Machine Learning Tutorial*. Taiwan: DISP Lab.
- Choi, Y., Won, C. S., Ro, Y. M., & Manjunath, B. S. (2003). Texture Descriptors. En B. S. Manjunath, P. Salembier, & T. Sikora, *Introduction to MPEG-7. Multimedia Content Description Interface* (págs. 213-228). West Sussex England: JOHN WILEY & SONS, LTD.

- Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. *IEEE. CVPR 2004, Workshop on Generative-Model*.
- Frosio, I., & Ratner, E. R. (2015). Adaptive Segmentation based on a Learned Quality Metric. *VISAPP 2015 - International Conference on Computer Vision Theory and Applications*, (págs. 283-292). Berlin.
- Gonzalez, R. C., & Woods, R. E. (2008). *Digital Image Processing* (Tercera ed.). New Jersey, E.U.A.: Pearsons Education, Inc.
- Hadamard, J. (1923). *Lectures on Cauchy's Problem in Linear Partial Differential Equations*. New Haven.: Yale University Press.
- Harrington, P. (2012). *Machine Learning in Action*. Shelter Island, NY: Manning Publications Co.
- Hasan Abdulrahman, B. M. (2017). From Contours to Ground Truth: How to Evaluate Edge Detectors by Filtering. *Journal of WSCG*.
- Hodakova, P. (2014). *Fuzzy (F-)Transform of functions of two variables and its applications in image processing*. Tesis doctoral, Universidad de Ostrava, Ostrava.
- <http://www.imageprocessingplace.com>. (2019). <http://www.imageprocessingplace.com>.  
Obtenido de <http://www.imageprocessingplace.com>:  
[http://www.imageprocessingplace.com/root\\_files\\_V3/image\\_databases.htm](http://www.imageprocessingplace.com/root_files_V3/image_databases.htm)
- Hu, M.-K. (1962). Visual Pattern Recognition by Moment Invariants". *IRE TRANSACTIONS ON INFORMATION THEORY*, 179-186.
- ISO. (January de 2001). MPEG-7 Visual part of eXperimentation Model Version 8. *ISO/IEC JTC1/SC29/WG11/ M6808* . Pisa, Italiy.
- ISO. (October de 2004). MPEG-7 Overview (version 10). *ISO/IEC JTC1/SC29/WG11N6828*. Palma de Mallorca, Spain.
- Krig, S. (2014). *Ground Truth Data, Content, Metrics, and Analysis In: Computer Vision Metrics*. Berkeley, CA: Apress.
- Kumar Abhinav, J. S. (2017). Image Segmentation of Multi-Shaped Overlapping Objects.
- Latecki, L. J., Lakämper, R., & Eckhardt, U. (2000). Shape Descriptors for Non-rigid Shapes with a Single Closed Contour. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

- Mallick, S. (17 de Febrero de 2015). <https://www.learnopencv.com>. Obtenido de <https://www.learnopencv.com>: <https://www.learnopencv.com/blob-detection-using-opencv-python-c/>
- Manjunath, B. S., Salembier, P., & Sikora, T. (2003). *Introduction to MPEG-7. Multimedia Content Description Interface*. Sussex: John Wiley & Sons Ltd.
- Martin, D., Fowlkes, C., Tal, D., & Malik, J. (July de 2001). A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. *Proc. 8th Int'l Conf. Computer Vision*, 2, págs. 416--423.
- Meiling Gong, J. L. (2017). Adaptive image segmentation algorithm under the constraint of edge posterior probability. *IET Computer Vision*, 11(8), pp. 702-709.
- Mirosław Bober, F. P.-Y. (2003). Shape Descriptors. En B. S. Manjunath, P. Salembier, & T. Sikora, *Introduction to MPEG-7. Multimedia Content Description Interface* (págs. 231-260). West Sussex England: JOHN WILEY & SONS, LTD.
- Moeslund, T. (2012). *Introduction to video and image processing*. London: Springer.
- Mukherjee, D., Jonathan Wu, Q., & Wang, G. (2015). A Comparative Experimental Study of Image Feature Detectors and Descriptors. *Machine Vision and Applications*.
- Nixon, M., & Aguado, A. (2008). *Feature Extraction and Image Processing*. San Diego CA.: AP, Elsevier.
- Ohm, J.-R., CieplinskiLeszek, K. H., Krishnamachari, S., Manjunath, B. S., Messing, D. S., & Yamada, A. (2003). Color Descriptors. En B. S. Manjunath, P. Salembier, & T. Sikora, *Introduction to MPEG-7. Multimedia Content Description Interface* (págs. 187-212). West Sussex England: JOHN WILEY & SONS, LTD.
- Pajares, G., & de la Cruz, J. (2008). *Visión por computadora: Imágenes digitales y aplicaciones*. México D.F.: Alfaomega Grupo Editor, S.A. de CV México.
- Peng, B., Zhang, L., Mou, X., & Yang, M.-H. (2017). Evaluation of Segmentation Quality via Adaptive Composition of Reference Segmentations. *IEEE TRANSACTION ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*.
- Pérez, C. (2014). *Recuperación Automatizada de Imágenes mediante la Implementación de Descriptores del Estándar MPEG-7*. Cuernavaca, Morelos: cenidet.

- Perfilieva, I. (2015). F-transform. En W. P. Janusz Kacprzyk, *Springer Handbook of Computational Intelligence* (págs. 113-129). Berlin: Springer.
- Qian, C. L. (2015). Measuring Image Similarity Based on Shape Context. *International Journal of Multimedia and Ubiquitous Engineering*, 127-134.
- Qiang, S., Guoying, L., Jingqi, M., & Hongmei, Z. (2016). An Edge-Detection Method Based on Adaptive Canny Algorithm and Iterative Segmentation Threshold. *2nd International Conference on Control Science and Systems Engineering*, (págs. 64-67).
- R, B., & Kannan, V. (2009). Efficient use of MPEG-7 Color Layout and Edge Histogram Descriptors in CBIR Systems. *Global Journal of Computer Science and Technology*, 157-163.
- Raajan, P., Muthuselvi, S., & Agnes, S. A. (2015). An Adaptive Image Enhancement using Wiener Filtering with Compression and Segmentation. *International Journal of Computer Applications*, 15-19.
- Rayar, F. (2017). *ImageNet MPEG-7 Visual Descriptors*. Université Francois Rabelais of Tours, LI EA-6300, France.
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorensen, W. (1996). *Modelado y diseño orientados a objetos* (Primera ed.). Madrid: Prentice-Hall.
- Salahat, E., & Qasaimeh, M. (2017). Recent Advances in Features Extraction and Description Algorithms: A Comprehensive Survey.
- Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. New York: Cambridge University Press. Obtenido de <http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning>
- Smith, S., & Brady, J. (1997). SUSAN - a new approach to low level image processing. *International Journal of Computer Vision (IJCV)*, pp. 45–78.
- Sobel, I., & Feldman, G. (1968). A  $3 \times 3$  isotropic gradient operator for image processing. *Presented at a talk at the Stanford Artificial Project*.
- Suárez, S. D. (2018). *Sintonización de una Red Totalmente Conectada para Segmentación de Dos Clases de Objetos en Imágenes*. Cuernavaca, Morelos: cenidet.
- Surya, D., & Satheesh, J. (2016). Performance Evaluation of Image Segmentation using Objective Methods. *Indian Journal of Science and Technology*.

- Troncoso, R. P. (2007). *Indexado y Recuperación de Imágenes por Contenido Maestría en Ciencias en Ciencias Computacionales*. Cuernavaca, Morelos: cenidet.
- Unnikrishnan, R., Pantofaru, C., & Hebert, M. (2007). Toward Objective Evaluation of Image Segmentation Algorithms. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 926-944.
- Wang, Z. (2017). Image segmentation by combining the global and local properties. *Expert Systems With Applications Journal*, 87, 30-40.
- Won, C. S., Park, D. K., & Jeon, Y. S. (Febrero de 2002). Efficient Use of MPEG-7 Edge Histogram Descriptor. *ETRI Journal*, 1(24), 23-30.
- www.xnview.com*. (2019). Obtenido de <https://www.xnview.com/en/xnconvert/#downloads>
- Yeghiazaryana, V., & Voiculescu, I. (2019). A family of boundary overlap metrics for the evaluation of medical image segmentation. *Journal of Medical Imaging*.
- Young, I. T., Gerbrands, J. J., & J. van Vliet, L. (2007). *Fundamentals of Image Processing*. Delft, Países Bajos: Delft University of Technology.

# ANEXOS

## Anexo A

### Combinación de técnicas de preprocesamiento y segmentación

Clases generadas a partir de la combinación de técnicas de preprocesamiento y segmentación. En la tabla anexa se detalla el resultado de las combinaciones al seleccionar 11 técnicas de preprocesamiento y 7 de segmentación produciendo 77 combinaciones posibles mas la combinación [0,0] correspondiente a la clase 0.

<b>COMBINACIÓN DE TÉCNICAS DE PREPROCESAMIENTO Y SEGMENTACIÓN</b>
Clase 0: [0,0] = NA,NA (NO EXISTE UNA COMBINACIÓN ADECUADA PARA SEGMENTAR LA IMAGEN EN LOS TÉRMINOS DE LA MÉTRICA DE SIMILITUD)
Clase 1: [1,1] = ECUALIZACION HISTOGRAMA, SOBEL
Clase 2: [1,2] = ECUALIZACION HISTOGRAMA, CANNY
Clase 3: [1,3] = ECUALIZACION HISTOGRAMA, SCHARR
Clase 4: [1,4] = ECUALIZACION HISTOGRAMA, LAPLACIANO
Clase 5: [1,5] = ECUALIZACION HISTOGRAMA, GRADIENTE MORFOLOGICO
Clase 6: [1,6] = ECUALIZACION HISTOGRAMA, PREWITT
Clase 7: [1,7] = ECUALIZACION HISTOGRAMA, FUZZY_F1
Clase 8: [2,1] = FILTRO MEDIANA_K7, SOBEL
Clase 9: [2,2] = FILTRO MEDIANA_K7, CANNY
Clase 10: [2,3] = FILTRO MEDIANA_K7, SCHARR
Clase 11: [2,4] = FILTRO MEDIANA_K7, LAPLACIANO
Clase 12: [2,5] = FILTRO MEDIANA_K7, GRADIENTE MORFOLOGICO
Clase 13: [2,6] = FILTRO MEDIANA_K7, PREWITT
Clase 14: [2,7] = FILTRO MEDIANA_K7, FUZZY_F1
Clase 15: [3,1] = FILTRO BLUR GAUSIANO_K3SX0SY0, SOBEL
Clase 16: [3,2] = FILTRO BLUR GAUSIANO_K3SX0SY0, CANNY
Clase 17: [3,3] = FILTRO BLUR GAUSIANO_K3SX0SY0, SCHARR
Clase 18: [3,4] = FILTRO BLUR GAUSIANO_K3SX0SY0, LAPLACIANO
Clase 19: [3,5] = FILTRO BLUR GAUSIANO_K3SX0SY0, GRADIENTE MORFOLOGICO
Clase 20: [3,6] = FILTRO BLUR GAUSIANO_K3SX0SY0, PREWITT

<b>COMBINACIÓN DE TÉCNICAS DE PREPROCESAMIENTO Y SEGMENTACIÓN</b>
Clase 21: [3,7] = FILTRO BLUR GAUSIANO_K3SX0SY0, FUZZY_F1
Clase 22: [4,1] = FILTRO BLUR HOMOGENEO_K3, SOBEL
Clase 23: [4,2] = FILTRO BLUR HOMOGENEO_K3, CANNY
Clase 24: [4,3] = FILTRO BLUR HOMOGENEO_K3, SCHARR
Clase 25: [4,4] = FILTRO BLUR HOMOGENEO_K3, LAPLACIANO
Clase 26: [4,5] = FILTRO BLUR HOMOGENEO_K3, GRADIENTE MORFOLOGICO
Clase 27: [4,6] = FILTRO BLUR HOMOGENEO_K3, PREWITT
Clase 28: [4,7] = FILTRO BLUR HOMOGENEO_K3, FUZZY_F1
Clase 29: [5,1] = FILTRO BILATERAL_K3, SOBEL
Clase 30: [5,2] = FILTRO BILATERAL_K3, CANNY
Clase 31: [5,3] = FILTRO BILATERAL_K3, SCHARR
Clase 32: [5,4] = FILTRO BILATERAL_K3, LAPLACIANO
Clase 33: [5,5] = FILTRO BILATERAL_K3, GRADIENTE MORFOLOGICO
Clase 34: [5,6] = FILTRO BILATERAL_K3, PREWITT
Clase 35: [5,7] = FILTRO BILATERAL_K3, FUZZY_F1
Clase 36: [6,1] = FILTRO BLUR MEDIANA_K3, SOBEL
Clase 37: [6,2] = FILTRO BLUR MEDIANA_K3, CANNY
Clase 38: [6,3] = FILTRO BLUR MEDIANA_K3, SCHARR
Clase 39: [6,4] = FILTRO BLUR MEDIANA_K3, LAPLACIANO
Clase 40: [6,5] = FILTRO BLUR MEDIANA_K3, GRADIENTE MORFOLOGICO
Clase 41: [6,6] = FILTRO BLUR MEDIANA_K3, PREWITT
Clase 42: [6,7] = FILTRO BLUR MEDIANA_K3, FUZZY_F1
Clase 43: [7,1] = FILTRO BLUR MEDIANA_K5, SOBEL
Clase 44: [7,2] = FILTRO BLUR MEDIANA_K5, CANNY
Clase 45: [7,3] = FILTRO BLUR MEDIANA_K5, SCHARR
Clase 46: [7,4] = FILTRO BLUR MEDIANA_K5, LAPLACIANO
Clase 47: [7,5] = FILTRO BLUR MEDIANA_K5, GRADIENTE MORFOLOGICO
Clase 48: [7,6] = FILTRO BLUR MEDIANA_K5, PREWITT
Clase 49: [7,7] = FILTRO BLUR MEDIANA_K5, FUZZY_F1
Clase 50: [8,1] = FILTRO BLUR GAUSIANO_K5SX0SY0, SOBEL
Clase 51: [8,2] = FILTRO BLUR GAUSIANO_K5SX0SY0, CANNY
Clase 52: [8,3] = FILTRO BLUR GAUSIANO_K5SX0SY0, SCHARR
Clase 53: [8,4] = FILTRO BLUR GAUSIANO_K5SX0SY0, LAPLACIANO
Clase 54: [8,5] = FILTRO BLUR GAUSIANO_K5SX0SY0, GRADIENTE MORFOLOGICO
Clase 55: [8,6] = FILTRO BLUR GAUSIANO_K5SX0SY0, PREWITT
Clase 56: [8,7] = FILTRO BLUR GAUSIANO_K5SX0SY0, FUZZY_F1

<b>COMBINACIÓN DE TÉCNICAS DE PREPROCESAMIENTO Y SEGMENTACIÓN</b>
Clase 57: [9,1] = FILTRO BOX_K3, SOBEL
Clase 58: [9,2] = FILTRO BOX_K3, CANNY
Clase 59: [9,3] = FILTRO BOX_K3, SCHARR
Clase 60: [9,4] = FILTRO BOX_K3, LAPLACIANO
Clase 61: [9,5] = FILTRO BOX_K3, GRADIENTE MORFOLOGICO
Clase 62: [9,6] = FILTRO BOX_K3, PREWITT
Clase 63: [9,7] = FILTRO BOX_K3, FUZZY_F1
Clase 64: [10,1] = FILTRO WIENER_K3, SOBEL
Clase 65: [10,2] = FILTRO WIENER_K3, CANNY
Clase 66: [10,3] = FILTRO WIENER_K3, SCHARR
Clase 67: [10,4] = FILTRO WIENER_K3, LAPLACIANO
Clase 68: [10,5] = FILTRO WIENER_K3, GRADIENTE MORFOLOGICO
Clase 69: [10,6] = FILTRO WIENER_K3, PREWITT
Clase 70: [10,7] = FILTRO WIENER_K3, FUZZY_F1
Clase 71: [11,1] = FILTRO MODA, SOBEL
Clase 72: [11,2] = FILTRO MODA, CANNY
Clase 73: [11,3] = FILTRO MODA, SCHARR
Clase 74: [11,4] = FILTRO MODA, LAPLACIANO
Clase 75: [11,5] = FILTRO MODA, GRADIENTE MORFOLOGICO
Clase 76: [11,6] = FILTRO MODA, PREWITT
Clase 77: [11,7] = FILTRO MODA, FUZZY_F1

## Anexo B

### Registro de técnicas de preprocesamiento y segmentación aplicadas

El sistema desarrollado genera un registro de las técnicas aplicadas a cada imagen durante la fase de entrenamiento. El registro contiene dos listados, uno muestra el orden en que se fueron aplicando las combinaciones y la clase asignada, la información generada tiene el siguiente formato definido:

**Clase (etiqueta de clase): [ TECNICA DE PREPROCESAMIENTO, TÉCNICA DE SEGMENTACIÓN ]: valor métrica (accuracy)**

El segundo listado está ordenado respecto al valor máximo de la métrica obtenida al aplicar las secuencias de las técnicas, el formato se define como:

**valor métrica (accuracy): [ TECNICA DE PREPROCESAMIENTO, TÉCNICA DE SEGMENTACIÓN ]: Clase (etiqueta de clase)**

El registro también indica la combinación de mayor métrica con el formato:

**valor métrica (accuracy): [ TECNICA DE PREPROCESAMIENTO, TÉCNICA DE SEGMENTACIÓN ]: Clase (etiqueta de clase) <----- Secuencia de máximo valor métrico (accuracy)**

Como ejemplo, se presentan dos registros, uno correspondiente a la primer imagen que procesó el sistema en la fase entrenamiento, cuando la base de conocimiento estaba vacía (imagen **100.png**).

El segundo registro corresponde a otra imagen (**104.png**) cuando el sistema tiene información en su base de conocimiento, se observa que el número de técnicas aplicadas es menor (en algunos casos ) respecto a las aplicadas a la primer imagen.

Imagen: 100.png

Combinación de técnicas aplicadas = 66

**[Orden de aplicación de las secuencias de técnicas]**

- Clase (1): [ ECUALIZACION HISTOGRAMA, SOBEL ]: 0.628620 (accuracy)
- Clase (2): [ ECUALIZACION HISTOGRAMA, CANNY ]: 0.535029 (accuracy)
- Clase (3): [ ECUALIZACION HISTOGRAMA, SCHARR ]: 0.567460 (accuracy)
- Clase (4): [ ECUALIZACION HISTOGRAMA, LAPLACIANO ]: 0.615151 (accuracy)
- Clase (5): [ ECUALIZACION HISTOGRAMA, GRADIENTE MORFOLOGICO ]: 0.688125 (accuracy)
- Clase (6): [ ECUALIZACION HISTOGRAMA, PREWITT ]: 0.655810 (accuracy)
- Clase (7): [ FILTRO MEDIANA\_K7, SOBEL ]: 0.857143 (accuracy)
- Clase (8): [ FILTRO MEDIANA\_K7, CANNY ]: 0.595356 (accuracy)
- Clase (9): [ FILTRO MEDIANA\_K7, SCHARR ]: 0.734457 (accuracy)
- Clase (10): [ FILTRO MEDIANA\_K7, LAPLACIANO ]: 0.953901 (accuracy)
- Clase (11): [ FILTRO MEDIANA\_K7, GRADIENTE MORFOLOGICO ]: 0.878566 (accuracy)
- Clase (12): [ FILTRO MEDIANA\_K7, PREWITT ]: 0.861998 (accuracy)
- Clase (13): [ FILTRO BLUR GAUSIANO\_K3SX0SY0, SOBEL ]: 0.751381 (accuracy)
- Clase (14): [ FILTRO BLUR GAUSIANO\_K3SX0SY0, CANNY ]: 0.552678 (accuracy)
- Clase (15): [ FILTRO BLUR GAUSIANO\_K3SX0SY0, SCHARR ]: 0.624759 (accuracy)
- Clase (16): [ FILTRO BLUR GAUSIANO\_K3SX0SY0, LAPLACIANO ]: 0.726453 (accuracy)
- Clase (17): [ FILTRO BLUR GAUSIANO\_K3SX0SY0, GRADIENTE MORFOLOGICO ]: 0.760275 (accuracy)
- Clase (18): [ FILTRO BLUR GAUSIANO\_K3SX0SY0, PREWITT ]: 0.759649 (accuracy)
- Clase (19): [ FILTRO BLUR HOMOGENEO\_K3, SOBEL ]: 0.757696 (accuracy)
- Clase (20): [ FILTRO BLUR HOMOGENEO\_K3, CANNY ]: 0.560801 (accuracy)
- Clase (21): [ FILTRO BLUR HOMOGENEO\_K3, SCHARR ]: 0.636750 (accuracy)
- Clase (22): [ FILTRO BLUR HOMOGENEO\_K3, LAPLACIANO ]: 0.723213 (accuracy)
- Clase (23): [ FILTRO BLUR HOMOGENEO\_K3, GRADIENTE MORFOLOGICO ]: 0.752949 (accuracy)
- Clase (24): [ FILTRO BLUR HOMOGENEO\_K3, PREWITT ]: 0.762444 (accuracy)
- Clase (25): [ FILTRO BILATERAL\_K3, SOBEL ]: 0.726962 (accuracy)
- Clase (26): [ FILTRO BILATERAL\_K3, CANNY ]: 0.538864 (accuracy)
- Clase (27): [ FILTRO BILATERAL\_K3, SCHARR ]: 0.606288 (accuracy)
- Clase (28): [ FILTRO BILATERAL\_K3, LAPLACIANO ]: 0.723221 (accuracy)
- Clase (29): [ FILTRO BILATERAL\_K3, GRADIENTE MORFOLOGICO ]: 0.739160 (accuracy)
- Clase (30): [ FILTRO BILATERAL\_K3, PREWITT ]: 0.741449 (accuracy)
- Clase (31): [ FILTRO BLUR MEDIANA\_K3, SOBEL ]: 0.766338 (accuracy)
- Clase (32): [ FILTRO BLUR MEDIANA\_K3, CANNY ]: 0.550371 (accuracy)
- Clase (33): [ FILTRO BLUR MEDIANA\_K3, SCHARR ]: 0.624322 (accuracy)
- Clase (34): [ FILTRO BLUR MEDIANA\_K3, LAPLACIANO ]: 0.796572 (accuracy)

Clase (35): [ FILTRO BLUR MEDIANA\_K3, GRADIENTE MORFOLOGICO ]: 0.788823 (accuracy)  
 Clase (36): [ FILTRO BLUR MEDIANA\_K3, PREWITT]: 0.777404 (accuracy)  
 Clase (37): [ FILTRO BLUR MEDIANA\_K5, SOBEL ]: 0.820061 (accuracy)  
 Clase (38): [ FILTRO BLUR MEDIANA\_K5, CANNY ]: 0.579400 (accuracy)  
 Clase (39): [ FILTRO BLUR MEDIANA\_K5, SCHARR ]: 0.683772 (accuracy)  
 Clase (40): [ FILTRO BLUR MEDIANA\_K5, LAPLACIANO ]: 0.902105 (accuracy)  
 Clase (41): [ FILTRO BLUR MEDIANA\_K5, GRADIENTE MORFOLOGICO ]: 0.841194 (accuracy)  
 Clase (42): [ FILTRO BLUR MEDIANA\_K5, PREWITT]: 0.827286 (accuracy)  
 Clase (43): [ FILTRO BLUR GAUSIANO\_K5SX0SY0, SOBEL]: 0.767321 (accuracy)  
 Clase (44): [ FILTRO BLUR GAUSIANO\_K5SX0SY0, CANNY]: 0.574593 (accuracy)  
 Clase (45): [ FILTRO BLUR GAUSIANO\_K5SX0SY0, SCHARR]: 0.654441 (accuracy)  
 Clase (46): [ FILTRO BLUR GAUSIANO\_K5SX0SY0, LAPLACIANO ]: 0.731017 (accuracy)  
 Clase (47): [ FILTRO BLUR GAUSIANO\_K5SX0SY0, GRADIENTE MORFOLOGICO ]: 0.760265 (accuracy)  
 Clase (48): [ FILTRO BLUR GAUSIANO\_K5SX0SY0, PREWITT]: 0.765477 (accuracy)  
 Clase (49): [ FILTRO BOX\_K3, SOBEL ]: 0.757696 (accuracy)  
 Clase (50): [ FILTRO BOX\_K3, CANNY ]: 0.560801 (accuracy)  
 Clase (51): [ FILTRO BOX\_K3, SCHARR ]: 0.636750 (accuracy)  
 Clase (52): [ FILTRO BOX\_K3, LAPLACIANO ]: 0.723213 (accuracy)  
 Clase (53): [ FILTRO BOX\_K3, GRADIENTE MORFOLOGICO ]: 0.752949 (accuracy)  
 Clase (54): [ FILTRO BOX\_K3, PREWITT]: 0.762444 (accuracy)  
 Clase (55): [ FILTRO WIENER\_K3, SOBEL]: 0.757740 (accuracy)  
 Clase (56): [ FILTRO WIENER\_K3, CANNY]: 0.563149 (accuracy)  
 Clase (57): [ FILTRO WIENER\_K3, SCHARR]: 0.636773 (accuracy)  
 Clase (58): [ FILTRO WIENER\_K3, LAPLACIANO]: 0.747241 (accuracy)  
 Clase (59): [ FILTRO WIENER\_K3, GRADIENTE MORFOLOGICO ]: 0.763673 (accuracy)  
 Clase (60): [ FILTRO WIENER\_K3, PREWITT]: 0.762485 (accuracy)  
 Clase (61): [ FILTRO MODA, SOBEL ]: 0.766995 (accuracy)  
 Clase (62): [ FILTRO MODA, CANNY ]: 0.541917 (accuracy)  
 Clase (63): [ FILTRO MODA, SCHARR ]: 0.623418 (accuracy)  
 Clase (64): [ FILTRO MODA, LAPLACIANO ]: 0.769383 (accuracy)  
 Clase (65): [ FILTRO MODA, GRADIENTE MORFOLOGICO ]: 0.778883 (accuracy)  
 Clase (66): [ FILTRO MODA, PREWITT ]: 0.779643 (accuracy)

0.953901 (accuracy): [ FILTRO MEDIANA\_K7, LAPLACIANO ]: Clase (10) <-----  
 Secuencia de máximo valor métrico (accuracy)

### **[Ordenamiento descendente de las secuencias por valor de métrica]**

0.953901 (accuracy): [ FILTRO MEDIANA\_K7, LAPLACIANO ]: Clase (10)  
 0.902105 (accuracy): [ FILTRO BLUR MEDIANA\_K5, LAPLACIANO ]: Clase (40)  
 0.878566 (accuracy): [ FILTRO MEDIANA\_K7, GRADIENTE MORFOLOGICO ]: Clase (11)

0.861998 (accuracy): [ FILTRO MEDIANA\_K7, PREWITT ]: Clase (12)  
0.857143 (accuracy): [ FILTRO MEDIANA\_K7, SOBEL ]: Clase (7)  
0.841194 (accuracy): [ FILTRO BLUR MEDIANA\_K5, GRADIENTE MORFOLOGICO ]:  
Clase (41)  
0.827286 (accuracy): [ FILTRO BLUR MEDIANA\_K5, PREWITT]: Clase (42)  
0.820061 (accuracy): [ FILTRO BLUR MEDIANA\_K5, SOBEL ]: Clase (37)  
0.796572 (accuracy): [ FILTRO BLUR MEDIANA\_K3, LAPLACIANO ]: Clase (34)  
0.788823 (accuracy): [ FILTRO BLUR MEDIANA\_K3, GRADIENTE MORFOLOGICO ]:  
Clase (35)  
0.779643 (accuracy): [ FILTRO MODA, PREWITT ]: Clase (66)  
0.778883 (accuracy): [ FILTRO MODA, GRADIENTE MORFOLOGICO ]: Clase (65)  
0.777404 (accuracy): [ FILTRO BLUR MEDIANA\_K3, PREWITT]: Clase (36)  
0.769383 (accuracy): [ FILTRO MODA, LAPLACIANO ]: Clase (64)  
0.767321 (accuracy): [ FILTRO BLUR GAUSIANO\_K5SX0SY0, SOBEL ]: Clase (43)  
0.766995 (accuracy): [ FILTRO MODA, SOBEL ]: Clase (61)  
0.766338 (accuracy): [ FILTRO BLUR MEDIANA\_K3, SOBEL ]: Clase (31)  
0.765477 (accuracy): [ FILTRO BLUR GAUSIANO\_K5SX0SY0, PREWITT]: Clase (48)  
0.763673 (accuracy): [ FILTRO WIENER\_K3, GRADIENTE MORFOLOGICO ]: Clase (59)  
0.762485 (accuracy): [ FILTRO WIENER\_K3, PREWITT]: Clase (60)  
0.762444 (accuracy): [ FILTRO BLUR HOMOGENEO\_K3, PREWITT]: Clase (24)  
0.762444 (accuracy): [ FILTRO BOX\_K3, PREWITT]: Clase (54)  
0.760275 (accuracy): [ FILTRO BLUR GAUSIANO\_K3SX0SY0, GRADIENTE  
MORFOLOGICO ]: Clase (17)  
0.760265 (accuracy): [ FILTRO BLUR GAUSIANO\_K5SX0SY0, GRADIENTE  
MORFOLOGICO ]: Clase (47)  
0.759649 (accuracy): [ FILTRO BLUR GAUSIANO\_K3SX0SY0, PREWITT]: Clase (18)  
0.757740 (accuracy): [ FILTRO WIENER\_K3, SOBEL ]: Clase (55)  
0.757696 (accuracy): [ FILTRO BLUR HOMOGENEO\_K3, SOBEL ]: Clase (19)  
0.757696 (accuracy): [ FILTRO BOX\_K3, SOBEL ]: Clase (49)  
0.752949 (accuracy): [ FILTRO BLUR HOMOGENEO\_K3, GRADIENTE  
MORFOLOGICO ]: Clase (23)  
0.752949 (accuracy): [ FILTRO BOX\_K3, GRADIENTE MORFOLOGICO ]: Clase (53)  
0.751381 (accuracy): [ FILTRO BLUR GAUSIANO\_K3SX0SY0, SOBEL ]: Clase (13)  
0.747241 (accuracy): [ FILTRO WIENER\_K3, LAPLACIANO ]: Clase (58)  
0.741449 (accuracy): [ FILTRO BILATERAL\_K3, PREWITT ]: Clase (30)  
0.739160 (accuracy): [ FILTRO BILATERAL\_K3, GRADIENTE MORFOLOGICO ]: Clase  
(29)  
0.734457 (accuracy): [ FILTRO MEDIANA\_K7, SCHARR ]: Clase (9)  
0.731017 (accuracy): [ FILTRO BLUR GAUSIANO\_K5SX0SY0, LAPLACIANO ]: Clase  
(46)  
0.726962 (accuracy): [ FILTRO BILATERAL\_K3, SOBEL ]: Clase (25)  
0.726453 (accuracy): [ FILTRO BLUR GAUSIANO\_K3SX0SY0, LAPLACIANO ]: Clase  
(16)  
0.723221 (accuracy): [ FILTRO BILATERAL\_K3, LAPLACIANO ]: Clase (28)  
0.723213 (accuracy): [ FILTRO BLUR HOMOGENEO\_K3, LAPLACIANO ]: Clase (22)  
0.723213 (accuracy): [ FILTRO BOX\_K3, LAPLACIANO ]: Clase (52)

0.688125 (accuracy): [ ECUALIZACION HISTOGRAMA, GRADIENTE MORFOLOGICO ]: Clase (5)  
 0.683772 (accuracy): [ FILTRO BLUR MEDIANA\_K5, SCHARR ]: Clase (39)  
 0.655810 (accuracy): [ ECUALIZACION HISTOGRAMA, PREWITT ]: Clase (6)  
 0.654441 (accuracy): [ FILTRO BLUR GAUSIANO\_K5SX0SY0, SCHARR]: Clase (45)  
 0.636773 (accuracy): [ FILTRO WIENER\_K3, SCHARR]: Clase (57)  
 0.636750 (accuracy): [ FILTRO BLUR HOMOGENEO\_K3, SCHARR]: Clase (21)  
 0.636750 (accuracy): [ FILTRO BOX\_K3, SCHARR]: Clase (51)  
 0.628620 (accuracy): [ ECUALIZACION HISTOGRAMA, SOBEL ]: Clase (1)  
 0.624759 (accuracy): [ FILTRO BLUR GAUSIANO\_K3SX0SY0, SCHARR]: Clase (15)  
 0.624322 (accuracy): [ FILTRO BLUR MEDIANA\_K3, SCHARR ]: Clase (33)  
 0.623418 (accuracy): [ FILTRO MODA, SCHARR ]: Clase (63)  
 0.615151 (accuracy): [ ECUALIZACION HISTOGRAMA, LAPLACIANO ]: Clase (4)  
 0.606288 (accuracy): [ FILTRO BILATERAL\_K3, SCHARR ]: Clase (27)  
 0.595356 (accuracy): [ FILTRO MEDIANA\_K7, CANNY ]: Clase (8)  
 0.579400 (accuracy): [ FILTRO BLUR MEDIANA\_K5, CANNY ]: Clase (38)  
 0.574593 (accuracy): [ FILTRO BLUR GAUSIANO\_K5SX0SY0, CANNY ]: Clase (44)  
 0.567460 (accuracy): [ ECUALIZACION HISTOGRAMA, SCHARR ]: Clase (3)  
 0.563149 (accuracy): [ FILTRO WIENER\_K3, CANNY]: Clase (56)  
 0.560801 (accuracy): [ FILTRO BLUR HOMOGENEO\_K3, CANNY]: Clase (20)  
 0.560801 (accuracy): [ FILTRO BOX\_K3, CANNY]: Clase (50)  
 0.552678 (accuracy): [ FILTRO BLUR GAUSIANO\_K3SX0SY0, CANNY]: Clase (14)  
 0.550371 (accuracy): [ FILTRO BLUR MEDIANA\_K3, CANNY ]: Clase (32)  
 0.541917 (accuracy): [ FILTRO MODA, CANNY ]: Clase (62)  
 0.538864 (accuracy): [ FILTRO BILATERAL\_K3, CANNY ]: Clase (26)  
 0.535029 (accuracy): [ ECUALIZACION HISTOGRAMA, CANNY ]: Clase (2)

-----  
 Imagen: 104.png

Combinación de técnicas aplicadas = 1

**[Orden de aplicación de las secuencias de técnicas]**

Clase (1): [ FILTRO MEDIANA\_K7, GRADIENTE MORFOLOGICO ]: 0.960861 (accuracy)

0.960861 (accuracy): [ FILTRO MEDIANA\_K7, GRADIENTE MORFOLOGICO ]: Clase (11) <----- Secuencia de máximo valor métrico (accuracy)

Ordenamiento descendente de las secuencias por valor de métrica

**[Ordenamiento descendente de las secuencias por valor de métrica]**

0.960861 (accuracy): [ FILTRO MEDIANA\_K7, GRADIENTE MORFOLOGICO ]: Clase (1)

-----

## Anexo C

Reconocimiento por participación en el evento “Escuela de Inteligencia Artificial y Robótica” UTEZ (octubre - 2018).



Reconocimiento por participación en el 1er coloquio "ARTIFICIAL INTELLIGENCE" realizado en el Instituto Tecnológico de Iguala Guerrero (mayo - 2019).



Poster presentado en la “Escuela de Inteligencia Artificial y Robótica” UTEZ (octubre - 2018)



# Modelo Autoadaptable para Segmentación de Imágenes Digitales por Discontinuidades



Ing. José Manuel Pérez Tovar  
Asesor: Dr. Raúl Pinto Elías

## RESUMEN

La segmentación de imágenes digitales es de gran importancia para una variedad de tareas en el procesamiento de imágenes y visión artificial. Juega un papel importante en la detección, reconocimiento, y seguimiento de objetos en aplicaciones como: Imágenes médicas, localización de objetos en imágenes de satélite, reconocimiento facial, reconocimiento de huellas dactilares, sistemas de control de tráfico y detección de luz de frenado en autos, entre otros.

- Los sistemas de visión artificial se diseñan generalmente para una aplicación específica.

A través de un modelo autoadaptable, se busca que un sistema de visión artificial sea capaz de segmentar distintas imágenes dentro de su dominio del problema, sin la necesidad de reprogramar o reconfigurar al sistema.

## INTRODUCCIÓN:

La segmentación permite descomponer la imagen en sus partes constituyentes. Es un proceso mediante el cual se extrae de una imagen digital cierta información subyacente. Este proceso es una etapa medular en el análisis de imagen y un tema abierto a la investigación.

- La segmentación por discontinuidades está orientada a encontrar los bordes que separan regiones diferentes en la imagen.

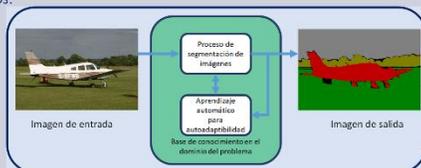
Mediante un mecanismo de aprendizaje supervisado integrado al proceso de segmentación y a través de un proceso de entrenamiento con un conjunto de imágenes ejemplo, se busca generar un modelo capaz de procesar nuevos casos.

## OBJETIVO:

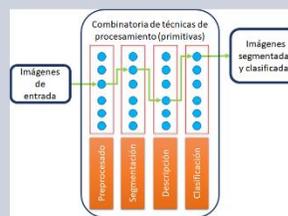
- Diseño e implementación de un modelo autoadaptable para segmentación de imágenes digitales por discontinuidades.

## METODOLOGÍA:

- Para la realización del proyecto se implementa un conjunto base de técnicas de procesamiento de imagen o primitivas, más las técnicas de aprendizaje supervisado, que permitan desarrollar el modelo requerido para proporcionar la auto adaptabilidad al proceso de segmentación de imágenes, previamente establecido.
- Se aplica el conjunto de técnicas de procesamiento sobre un grupo de imágenes de ejemplo de las cuales se conoce su segmentación (original y *ground truth*).
- Se clasifican las imágenes buscando patrones similares entre sus características, de tal forma que el conocimiento adquirido en este proceso de aprendizaje permita procesar nuevos casos.
- El desarrollo se realiza con herramientas de distribución libre, Eclipse C++ y las librerías de procesamiento de imágenes OpenCV, entre otros.



## RESULTADOS:



Técnicas	Imagen	Ground Truth	CLAS	Desviaciones
P1,S2,D3,C1	Imagen 1	GT 1	C1	D1,D3,D5
	Imagen 2	GT 2	C1	D1,D3,D5
	Imagen 3	GT 3	C1	D1,D3,D5
P3,S4,D3,C2	Imagen 4	GT 4	C1	D1,D3,D5
	Imagen 5	GT 5	C2	D2,D4
	Imagen 6	GT 6	C2	D2,D4
P3,S4,D3,C2	Imagen 7	GT 7	C2	D2,D4
	Imagen 8	GT 8	C2	D2,D4
	Imagen 9	GT 9	C3	D2,D4

Flujo del procesamiento de la imagen desde su preprocesado hasta su clasificación.

En el desarrollo del trabajo se han logrado diferentes resultados en la implementación de técnicas de preprocesamiento de imágenes y filtrado para la segmentación, utilizando las bibliotecas OpenCV.

Se han implementado técnicas de eliminación de ruido y mejoramiento de imagen, para detectar los bordes de mejor manera.

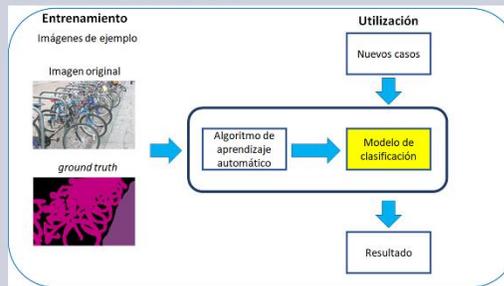


Aplicación de filtro de mediana para eliminación de ruido sal y pimienta, técnica de preprocesado.



Aplicación de filtros derivativos para detección de bordes en la imagen, técnicas de segmentación.

El trabajo realizado se enfoca en obtener el modelo adecuado que permita procesar nuevos casos de segmentación de imágenes por discontinuidades, mediante aprendizaje supervisado.



## REFERENCIAS:

- [1] R. E. W. Rafael C. González, Digital Image Processing, Tercera ed., New Jersey: Pearson Education, Inc., 2008.
- [2] G. P. M. y J. M. d. I. C. García, Visión por computadora: Imágenes digitales y aplicaciones, México D.F.: Alfaomega Grupo Editor, S.A. de CV México, 2008.
- [3] J. S. S. R. S. a. 4. K. Jimmy Singla, «TECHNIQUE OF IMAGE SEGMENTATION IN DIGITAL IMAGE PROCESSING,» Bookman International Journal of Software Engineering, vol. 1, nº 2, pp. 50-52, 2012.

Poster presentado en la “Escuela de Inteligencia Artificial y Robótica” UTEZ (octubre - 2019)



**TECNM**  
TECNOLOGICO NACIONAL DE  
MEXICO

## Modelo Autoadaptable para Segmentación de Imágenes Digitales por Discontinuidades



**cenidet**<sup>(C)</sup>  
Centro Nacional de Investigación y Desarrollo Tecnológico

Ing. José Manuel Pérez Tovar  
Asesor: Dr. Raúl Pinto Elías

**RESUMEN**

La segmentación de imágenes digitales es de gran importancia para una variedad de tareas en el procesamiento de imágenes y visión artificial. Juega un papel importante en la detección, reconocimiento y seguimiento de objetos en aplicaciones como: imágenes médicas, localización de objetos en imágenes de satélite, reconocimiento facial, reconocimiento de huellas dactilares, sistemas de control de tráfico y detección de luz de frenado en autos, entre otros.

- Los sistemas de visión artificial se diseñan generalmente para una aplicación específica.

A través de un modelo autoadaptable, se busca que un sistema de visión artificial sea capaz de segmentar distintas imágenes dentro de su dominio del problema, sin la necesidad de reprogramación o reconfiguración.

**INTRODUCCIÓN:**

La segmentación permite descomponer la imagen en sus partes constituyentes. Es un proceso mediante el cual se extrae de una imagen digital cierta información subyacente. Este proceso es una etapa medular en el análisis de imagen y un tema abierto a la investigación.

- La segmentación por discontinuidades está orientada a encontrar los bordes o fronteras que separan los diferentes objetos en una imagen.

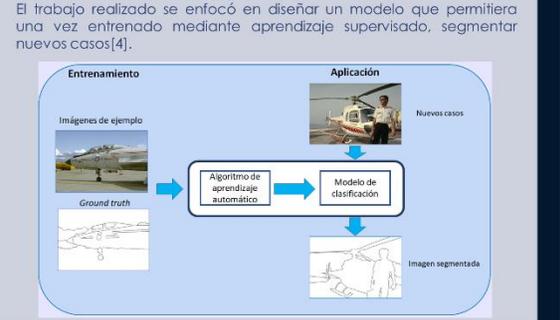
Mediante un mecanismo de aprendizaje supervisado integrado al proceso de segmentación y a través de un proceso de entrenamiento con un conjunto de imágenes ejemplo, se busca generar un modelo capaz de procesar nuevos casos.

**OBJETIVO:**

- Diseño e implementación un modelo autoadaptable de segmentación de imágenes digitales por discontinuidades.

**METODOLOGÍA:**

- Para la realización del proyecto se implementa un conjunto base de técnicas de procesamiento de imagen o primitivas, más las técnicas de aprendizaje supervisado, que permitan desarrollar el modelo requerido para proporcionar la auto adaptabilidad al proceso de segmentación de imágenes, previamente establecido.
- Se aplica el conjunto de técnicas de procesamiento sobre un grupo de imágenes de ejemplo de las cuales se conoce su segmentación (original y ground truth).
- Se clasifican las imágenes buscando patrones similares entre sus atributos característicos o descriptores, de tal forma que el conocimiento adquirido en este proceso permita procesar nuevos casos.
- La implementación se realiza con herramientas de distribución libre, Qt C++ y las bibliotecas de procesamiento de imágenes (OpenCV) entre otros.



**REFERENCIAS:**

[1] Frédéric Rayar, "ImageNet MPEG-7 Visual Descriptors," Université Francois Rabelais de Tours, LI EA-6300, France, 2017.

[2] R. E. W. Rafael C. González, Digital Image Processing, Tercera ed., New Jersey: Pearson Education, Inc., 2008.

[3] G. P. M. y J. M. d. L. C. García, Visión por computadora: Imágenes digitales y aplicaciones, México D.F.: Alfaomega Grupo Editor, S.A. de CV México, 2008.

[4] J. S. S. R. a. 4. K. Jimmy Singla, «TECHNIQUE OF IMAGE SEGMENTATION IN DIGITAL IMAGE PROCESSING,» Bookman International Journal of Software Engineering, vol. 1, nº 2, pp. 50-52, 2012.