# Universidad de Oriente

## Facultad de Ingeniería Eléctrica

### Departamento de Telecomunicaciones



# TRABAJO DE DIPLOMA

## Reconocimiento de armas en imágenes de rayos X mediante Saco de Palabras Visuales.


**Autor: David Castro Piñol**

**Tutor: Frank Sanabria Macias**

**Cotutor: Felipe Rodriguez Arias**



**Santiago de Cuba**

**Junio, 2015**

# Universidad de Oriente

## Facultad de Ingeniería Eléctrica

### Departamento de Telecomunicaciones



# TRABAJO DE DIPLOMA

## Reconocimiento de armas en imágenes de rayos X mediante Saco de Palabras Visuales.


**Autor:David Castro Piñol**

**Tutor: Frank Sanabria Macias**

Profesor Asistente, Centro de Estudios de Neurociencias, Procesamiento de Imágenes y Señales, Facultad de Ingeniería Eléctrica, fsanm77@fie.uo.edu.cu

**Cotutor: Felipe Rodriguez Arias**

Profesor Asistente, Centro de Estudios de Neurociencias, Porcesamiento de Imágenes y Señales, Facultad de Ingeniería Eléctrica, farias@fie.uo.edu.cu



**Santiago de Cuba**

**Junio, 2015**


# COMPROMISO DEL AUTOR

Hago constar que el presente trabajo de diploma es de mi autoría exclusivamente, no constituyendo copia de ningún trabajo realizado anteriormente y las fuentes usadas para la realización del trabajo se encuentran referidas en la bibliografía. Doy mi consentimiento a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización del Tutor o Institución.

Firma del Autor

# Pensamiento

*Para conseguir la paz, se necesita valor,*
*mucho más que para hacer la guerra.*
*Papa Francisco*

# Dedicatoria

*Dedico este trabajo:*
*A mi familia querida: mi padre, mi madre, mi hermana*
*y al sobrino(a) que está en camino.*
*A mis amigos de todos estos años de estudio*
*y a mis amistades de Pastoral Juvenil.*

# Agradecimientos

*A toda la familia por el apoyo, educación y constante cercanía.*
*Al tutor por su excelente labor y guía en la disciplina de la investigación.*
*A los profesores del departamento y del CENPIS*
*que se han preocupado por enseñar.*
*A demás profesores de otras etapas*
*como Omar por el inglés que me enseñó*
*como Orestes por la matemática que aprendí*
*y otros que han sabido enseñar en tiempos difíciles.*
*A la Compañía de Jesús por su educación en formar hombres para los demás.*
*A todos ellos y a todos lo que*
*de una forma u otra han influido en este trabajo*
*Muchas Gracias!*

# Resumen

El diseño de un sistema automático que reconozca objetos peligrosos en imágenes de rayos X de equipos de inspección ha sido un problema complejo en los últimos años. La inspección de equipajes por rayos X presenta limitantes en cuanto a la eficiencia en el reconocimiento de objetos peligrosos y la demora que se toma el proceso. No existe una herramienta software que detecte automáticamente la presencia de armas en imágenes de rayos X y facilite el trabajo del operador de inspección. En este trabajo se desarrolló e implementó un algoritmo para el reconocimiento de armas cortas en imágenes de rayos X usando el método Saco de Palabras Visuales. Para realizar esto se implementó una etapa de pre-procesado, se construyó el vocabulario de palabras visuales que tuviera el mejor comportamiento frente a este tipo de imágenes, se representó un conjunto de imágenes mediante los histogramas de palabras visuales y se realizó el entrenamiento de un clasificador de tipo Máquina de Soporte Vectorial. Este algoritmo se desarrolló sobre la plataforma Matlab y con el apoyo de la biblioteca de funciones VLFeat. Se realizaron diversos experimentos variando los parámetros del método obteniéndose como mejor resultado una razón de verdaderos positivos de un 97.12 % y una razón de falsos positivos de 7.4 %. Estos resultados muestran que el algoritmo implementado puede servir de apoyo al personal de inspección, aumentar la rapidez del proceso y mejorar la eficiencia en el reconocimiento de armas en las imágenes de rayos X del sistema de inspección de equipajes.

**Palabras clave**: Saco de Palabras Visuales, Máquina de Soporte Vectorial, imágenes de rayos X

# Abstract

An automatic system's design that recognizes dangerous objects in baggage X-ray images has been a complex problem in recent years. X-ray inspection has difficulties because of the low efficiency in automatic recognition of dangerous objects and inspection process delay. It doesn't exist a software application that automatically detects weapons in those images and reduce the workload of screeners. In this project was developed and implemented an algorithm for recognizing handguns in X-ray images using the Bag of Visual Words method. In order to achieve this, it was implemented a preprocess, was built a vocabulary of visual words with the better performance for this kind of images, it was represented a set of images by histograms of visual words and it was trained a Support Vector Machine classifier. This algorithm was developed in Matlab platform using VLFeat library. It was performed several experiments handling tunable parameters, getting the most relevant result a true positive rate of 97.12 % and a false positive rate of 7.4 %. These results show that the implemented algorithm could be a support for inspection screeners and hence increase inspection speed and increase the efficiency of weapons recognition in X-ray images of inspection system.

**Keywords**: Bag of Visual Words, Suport Vector Machines, X ray images

# Índice

**Introducción** **1**

**1. Estudio Teórico de las imágenes de rayos X y BoVW** **5**

1.1. Introducción a los equipos de rayos X . . . . . 5

1.2. Imágenes de rayos X . . . . . 8

1.3. El método BoVW . . . . . 9

1.4. Extracción de Características . . . . . 10

1.4.1. Algoritmo SIFT . . . . . 11

1.4.2. DSIFT una variante rápida de SIFT . . . . . 14

1.5. Agrupación de características mediante k-mean . . . . . 14

1.6. Cuantificación Vectorial y obtención del histograma de palabras visuales . 17

1.7. Parámetros de BoVW . . . . . 17

1.7.1. Tamaño del Vocabulario . . . . . 18

1.7.2. Información espacial . . . . . 19

1.7.3. Técnicas de asociación de pesos en los histogramas . . . . . 20

1.8. Etapa de Clasificación . . . . . 22

1.8.1. Fundamentos de las Máquinas de Soporte Vectorial (SVM) . . . . . 23

1.8.2. El truco del kernel . . . . . 25

1.8.3. Redefinición del problema mediante la función de costo . . . . . 27

1.9. Métricas de evaluación . . . . . 29

1.9.1. Curva ROC . . . . . 29

1.9.2. Validación cruzada . . . . . 32

1.9.3. Promediado de curvas ROC . . . . . 34

**2. Aplicación de BoVW en imágenes de rayos X** **35**

2.1. Introducción . . . . . 35

2.2. Extracción de características usando PHOW . . . . . 36

2.3. Implementacion del pre-procesamiento . . . . . 38

2.4. Construcción de vocabularios . . . . . 41

2.5. Cálculo de los histogramas de palabras visuales . . . . . 43

2.6. Entrenamiento de la SVM . . . . . 45

2.6.1. Funciones de pérdida . . . . . 45

2.6.2. Mapas de kernels homogéneos . . . . . 46

2.7. Implementación de la validación cruzada . . . . . 49

2.8. Promediado de curvas ROC . . . . . 49

**3. Resultados Experimentales** **52**

3.1. Introducción . . . . . 52

3.2. Descripción de la base de datos . . . . . 52

3.3. Evaluación del clasificador . . . . . 54

3.3.1. Evaluación del pre-procesado y de los kernels . . . . . 55

3.3.2. Evaluación de las diferentes funciones de pérdidas . . . . . 55

3.3.3. Evaluación de los vocabularios . . . . . 56

3.3.4. Evaluación del algoritmo frente al reconocimiento de armas solapadas 58

3.4. Análisis y discusión de los resultados . . . . . 60

**Conclusiones** **65**

**Recomendaciones** **66**

**Referencias Bibliográficas** **67**

# Índice de figuras


1. *Dos ejemplos de imágenes de rayos X que contienen armas de fuego. Son difíciles de interpretar debido al inusual punto de vista.* . . . . . 2

1.1. *Esquema de equipo de inspección.* . . . . . 6
1.2. *Usando dos niveles de energía para obtener una imagen de pseudo-color.* . . . . 7
1.3. *Imagen en pseudo-color entregada por el equipo de inspección de rayos X* . . . . 8
1.4. *Diagrama general de un algoritmo clasificador basado en BoVW* . . . . . 10
1.5. *Extracción de los puntos clave.* . . . . . 11
1.6. *Ejemplo de puntos clave SIFT. Características.* . . . . . 13
1.7. *Descriptor SIFT* . . . . . 13
1.8. *Geometría del descriptor DSIFT.* . . . . . 15
1.9. *Ejemplo de un vocabulario de palabras visuales. Los puntos negros son los centroides, los grupos de color son las palabras visuales.* . . . . . 15
1.10. *Aplicación de k-means. Se aprecian cuatro palabras visuales con sus correspondientes centroides* $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$ *y* $\mathbf{w}_4$. . . . . 16
1.11. *Cuantificación vectorial y obtención del histograma de palabras visuales.* . . . . 17
1.12. *Formas de dividir la imagen para incluir la información espacial.* . . . . . 19
1.13. *Beneficios del pesado suave.* . . . . . 22
1.14. *Representación del hiperplano de separación en el proceso de entrenamiento de una SVM* . . . . . 26
1.15. *Expansión de los datos mediante el mapeo de características* . . . . . 26
1.16. *Dibujo para entender las razones de la curva ROC* . . . . . 30
1.17. *Matriz de confusión* . . . . . 31
1.18. *Ejemplo de curva ROC.* . . . . . 32
1.19. *Divisiones del conjunto de datos para una validación cruzada de 3 segmentos* . . 33

2.1. *De izquierda a derecha: a) Puntos SIFT, b) puntos PHOW, c) zoom sobre los puntos PHOW* . . . . . 37
2.2. *a) Imagen original. b) Regiones que quedan después de hacer la segmentación* . 38
2.3. *a) Imagen binaria b) Resultado del cierre sobre la imagen binaria* . . . . . 39
2.4. *a) Filtrado por áreas b) Dilatación de la imagen* . . . . . 39
2.5. *Características extraídas después del pre-procesado* . . . . . 41
2.6. *a) Características para el vocabulario universal, b) características para el vocabulario metálico.* . . . . . 42
2.7. *Histogramas de palabras visuales de un objeto arma y otro no arma* . . . . . 44
2.8. *Solapamiento de caracteristicas visuales* . . . . . 44

3.1. *Muestra de las imágenes negativas preparadas* . . . . . 53
3.2. *Situaciones de las imágenes positivas utilizadas* . . . . . 54
3.3. *Curvas ROC de diferentes kernels* . . . . . 56
3.4. *Curvas ROC de diferentes funciones de pérdida* . . . . . 57
3.5. *Curvas ROC con diferentes vocabularios* . . . . . 57
3.6. *Armas que tienden a ser mal clasificadas* . . . . . 59
3.7. *Curva ROC con armas solapadas* . . . . . 60

# Índice de tablas

1.1. *Fórmulas de pesos para los histogramas de palabras visuales.* . . . . . 21

2.1. *Diferentes funciones de pérdida que se utilizan en SVM* . . . . . 46

3.1. *Distribución de las armas* . . . . . 54

3.2. *Efecto del pre-procesado* . . . . . 55

3.3. *Distribución para el experimento con armas solapadas* . . . . . 59

3.4. *Comparación con otros estudios* . . . . . 62

# Introducción

Los rayos X se descubrieron en el año 1895 por el físico Wilhelm Conrad Röntgen haciéndolo meritorio del premio Nobel en el año 1901 por su descubrimiento. Los rayos X permitieron ver e identificar las partes internas de los objetos. No solamente han sido desarrollados para su uso en imágenes médicas sino también en pruebas no intrusivas NDT (*non-destructive testing*) para materiales u objetos donde el propósito es analizar las partes internas que son indetectables a simple vista. Existe un conjunto de aplicaciones de rayos X que expone D. Mery 2013 [1] en una revisión del estado del arte en el tema. Aplicaciones como el análisis de productos alimenticios, inspección de piezas de automóviles, control de calidad de las soldaduras, inspección de cargas de contenedor e inspección de equipajes.

Las imágenes de Rayos X constituyen una importante tecnología para aplicaciones de seguridad en los equipos de inspección presentes en puertos y aeropuertos. Tradicionalmente estas imágenes son grabadas por equipos de Rayos X en puntos de seguridad donde hay un personal entrenado para detectar materiales ilícitos. Este personal se encuentra constantemente inspeccionando el contenido de las maletas.

El sistema de inspección del país, hasta el momento arroja muchos falsos positivos, es decir se activan alarmas cuando no hay objetos peligrosos y esto hace lento el proceso de inspección. Además el personal entrenado debe procesar un enorme flujo de imágenes que genera agotamiento visual, aumentando las posibilidades de cometer errores, por lo que debe ser relevado de su puesto de trabajo cada cierto intervalo de tiempo. Razones por las cuales aproximadamente la tercera parte de las maletas hay que revisarla nuevamente, según Padrón en 2012 [2]. De manera que se hace necesario el diseño de un sistema semiautomático del proceso de inspección para reducir la carga de trabajo, mejorar la eficiencia en la clasificación y aumentar la velocidad de inspección, Bastan en 2011 [3].

Se habla de un sistema semiautomático porque el objetivo no es desplazar al personal entrenado sino fortalecer y complementar su trabajo de manera que no tenga que revisar todas las imágenes sino solo una porción de ellas.

Para darle solución a esta situación se hace necesario la utilización de algoritmos de visión por computadora, que es el campo donde se aplica el reconocimiento de patrones para permitir que una computadora pueda "entender" el contenido de las imágenes. Existe una notable diferencia entre las imágenes de rayos X y las del espectro visible puesto que las imágenes de rayos X poseen un alto grado de desorden por la cantidad de objetos que se pueden encontrar en ellas en diversas posiciones, además por las características intrínsecas de estas imágenes muchos objetos aparecen superpuestos unos con otros. Pueden resultar ruidosas debido a la baja energía de los rayos emitidos por el equipo, además de esto se pueden encontrar objetos desde diversos puntos de vista (ver figura 1) de manera que la interpretación del contenido de las imágenes se torna un problema complejo Bastan [3].

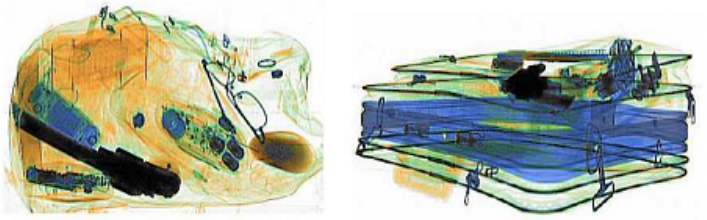

Figura 1: *Dos ejemplos de imágenes de rayos X que contienen armas de fuego. Son difíciles de interpretar debido al inusual punto de vista.* (Fuente: [4])

### Antecedentes del problema

En los últimos años se han realizado investigaciones de algoritmos de visión por computadora para aplicarlos a las imágenes de rayos X que entregan los equipos de inspección. Uno de los métodos que ha tenido muy buenos resultados es Saco de Palabras Visuales o *Bag-of-Visual-Words* (BoVW) propuesto por Csurka en 2004 [5] para búsqueda de imágenes por contenido y clasificación de objetos en imágenes de espectro visible. El trabajo realizado por Bastan, 2011 [3] y Bastan, 2013 [6] y el de Turcsany, 2013 [4] aplican el método BoVW en el contexto de imágenes de rayos X para reconocer objetos peligrosos. Existe una bibliografía limitada sobre el tema y las investigaciones realizadas que utilizan

BoVW difieren entre ellas.

El método Saco de Palabras Visuales BoVW está inspirado en la representación de textos mediante la frecuencia de aparición de palabras claves, conocido también como *Bag-of-Words* (BoW). Este es un algoritmo ampliamente usado para la clasificación de textos y constituye la base del concepto aplicado a imágenes. Esto quiere decir que muchas de las técnicas utilizadas para la clasificación de textos son también aplicables al problema de la clasificación de escenas y objetos en imágenes Yang 2007 [7].

El método Saco de Palabras Visuales BoVW está constituido por dos fases. La primera se basa en la construcción del vocabulario de palabras visuales a partir de una base de datos de imágenes mediante un algoritmo que agrupe las características extraídas. La segunda es la representación de una nueva imagen mediante un histograma de palabras visuales apoyándose en el vocabulario de palabras visuales construido. Con esta representación de las imágenes se puede construir un clasificador binario mediante una etapa de entrenamiento. El clasificador más utilizado con BoVW es la Máquina de Soporte Vectorial o *Support Vector Machine*(SVM) Vapnik 1998 [8].

El CENPIS (Centro de Estudios de Neurociencias, Procesamiento de Imágenes y Señales) se encuentra actualmente desarrollando un proyecto de investigación relacionado con la detección automática de objetos peligrosos en imágenes de rayos X. Se han estudiado las características de las mismas y el formato de las imágenes que retornan los equipos de rayos X de energía dual. Pero no se tiene la implementación de un algoritmo de clasificación basado en BoVW. Este trabajo se inserta dentro de dicho proyecto y se centra en el reconocimiento de armas de fuego, específicamente para armas cortas en imágenes de rayos X.

El algoritmo de clasificación basado en BoVW para el reconocimiento de armas que se propone en este trabajo se utilizaría sobre una ventana deslizante Viola 2001 [9] para la detección de estos objetos en las imágenes de rayos X. En el proceso de entrenamiento del algoritmo clasificador no se utilizaron las imágenes completas, sino las instancias de la ventana deslizante. Este trabajo solo abarca la implementación del algoritmo de reconocimiento.

**Problema**

La inexistencia en el CENPIS de un algoritmo que permita la representación de imágenes de rayos X usando el método de Saco de Palabras Visuales y también el reconocimiento de armas de fuego en dichas imágenes.

**Hipótesis**

Si se implementa la representación mediante el método de Saco de Palabras Visuales y se entrena un clasificador del tipo SVM, basado en la representación anterior, en el contexto de imágenes de rayos X, se puede obtener un algoritmo efectivo para reconocer armas cortas en imágenes de rayos X.

**Objetivo General**

Desarrollar e implementar un algoritmo para reconocimiento de armas cortas en imágenes de rayos X usando el método de Saco de Palabras Visuales.

**Objetivos Específicos**

• Desarrollar e implementar el método de Saco de Palabras Visuales para la representación de imágenes.

• Entrenar y evaluar un clasificador para armas cortas usando la representación de Saco de Palabras Visuales.

**Tareas**

• Estudio del estado del arte en el reconocimiento de objetos en imágenes de rayos X.

• Estudiar la teoría del método Saco de Palabras Visuales.

• Construir vocabularios de palabras visuales usando una base de datos de imágenes de rayos X que contengan varios tipos de objetos, incluidas armas cortas.

• Lograr una representación de las imágenes mediante histogramas de palabras visuales.

• Entrenar un clasificador binario SVM con los histogramas de palabras visuales.

• Evaluar el clasificador binario SVM.

• Analizar los resultados de la evaluación del clasificador SVM.

# Capítulo 1

# Estudio Teórico de las imágenes de rayos X y BoVW

## 1.1. Introducción a los equipos de rayos X

Los equipos de inspección por rayos X se basan en el principio de diferenciar los objetos de una muestra o valija a partir de la absorción de rayos X que presenten los materiales que lo componen [10]. Esencialmente estos equipos se componen de un emisor de los rayos X, un conjunto de sensores para cuantificar la energía del rayo al atravesar cada una de las regiones del objeto y una banda transportadora para desplazar el objeto a inspeccionar. El resto del equipo se compone de dispositivos de control, de protección y de interacción con el operador.

En la figura 1.1 se muestra el esquema de un equipo de inspección por rayos X. El modo de funcionamiento, de manera simplificada, de estos equipos para formar la imagen del objeto bajo análisis es el siguiente. Los sensores se encuentran usualmente en distribución en fila o en forma de L, alineados con el emisor en un plano perpendicular a la dirección de movimiento de la banda transportadora, registran el valor de la intensidad de rayo que llega a él en un instante dado. El conjunto de estos registran una columna de la imagen (cada sensor corresponde a un pixel de la imagen). En la medida que la banda transportadora se desplaza, se van generando las distintas columnas de una imagen.

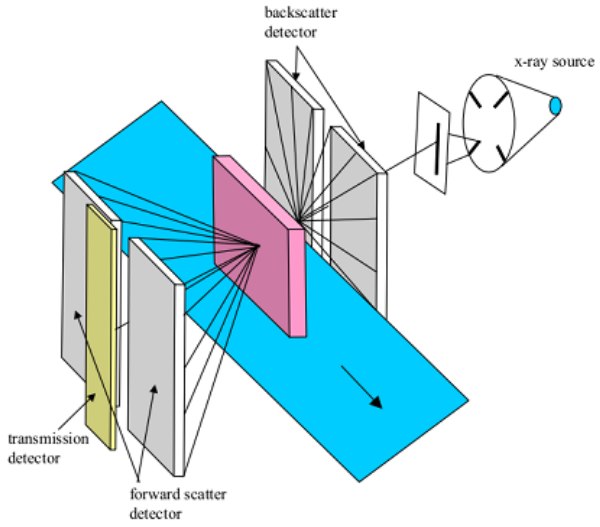


Figura 1.1: *Esquema de equipo de inspección.* (Fuente: [10])

Para identificar los diferentes objetos presentes en la muestra es necesario relacionar la absorción registrada en cada píxel con las propiedades del material de que este se compone. Entre las propiedades de un material que inciden en el nivel de absorción de los rayos X, en un material puro, están el número atómico y la densidad [11].

En la ecuación (1.1) se expresa la relación entre la intensidad del rayo emitido, el número atómico, la densidad de la sustancia y la intensidad del rayo luego de atravesar el material.

$$I(E) = I_0 e^{-\mu z} \tag{1.1}$$

Donde, $I_0$ es la intensidad del rayo emitido, $z$ es el ancho del material por donde pasa el rayo y $\mu$ es el coeficiente de atenuación lineal asociado con el material. El coeficiente $\mu$ puede ser modelado como $\mu = \alpha(Z, E)\rho$, donde $\rho$ es la densidad del material y $\alpha(Z, E)$ es el coeficiente de atenuación de masa que depende del número atómico del material $Z$ y de la energia $E$ de los fotones de rayos X [12].

Dado que la mayoría de los materiales en la naturaleza no son puros, se define el número atómico efectivo $Z_{eff}$ como el valor promedio de los números atómicos de las diferentes sustancias que lo componen, ponderado por las proporciones de cada sustancia en el material [13].

Para representarlas, cada valor posible de estas propiedades, es asociado a un color de manera que pueda visualizarse como una imagen. Este tipo de imágenes son conocidas como imagen en pseudo-color, falso color o imágenes de equipos de energía dual.

Los equipos de rayos X de energía dual para formar la imagen realizan una combinación de dos radiográficos adquiridos a dos diferentes niveles de energía para obtener la densidad y el número atómico de los materiales. Esto también provee información de la composición de los materiales. Las imágenes de baja y alta energía son fusionadas con la ayuda de una tabla de consulta (*look up table*) para obtener esta imagen de color que facilita la interpretación del contenido de las maletas. En la figura 1.2 se aprecia la obtención de la imagen de pseudo-color.

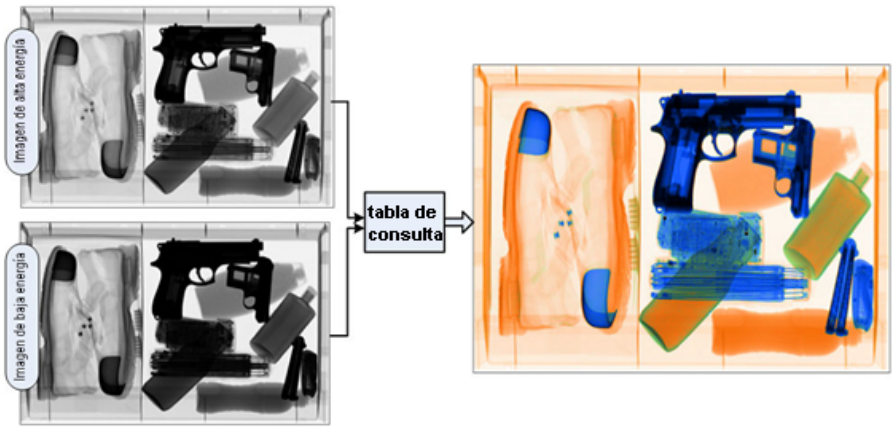


Figura 1.2: *Usando dos niveles de energía para obtener una imagen de pseudo-color.* (Fuente: [3])

Existen equipos que retornan múltiples vistas del equipaje desde diferentes ángulos, posibilitando al operador reconocer con mayor facilidad los objetos. Las múltiples vistas se obtienen muchas veces por un manipulador robótico que posiciona el objeto de análisis o el equipo mismo de adquisición para obtener varias vistas [12]. Estos equipos se conocen como 3D. En el presente trabajo se exploran solo imágenes de equipos de energía dual. La utilización de múltiples vistas aumenta considerablemente la calidad de los resultados de un algoritmo clasificador. Las recientes investigaciones sobre reconocimiento de objetos en múltiples vistas son los trabajos de Bastan 2013 [6], Franzel [14] y Mery [15] [16]

## 1.2. Imágenes de rayos X

Los equipos de inspección por rayos X actuales tienen la capacidad de entregar varios tipos de imágenes en pseudo-color, en función del tipo de objeto que se desee enfatizar. En este trabajo se usó únicamente como modelo de imagen en pseudo-color, uno de los modelos más comunes de encontrar en equipos de este tipo. En este modelo, los materiales orgánicos ($Z_{eff} < 11$) se muestran con colores de tono naranja, los materiales metálicos ($Z_{eff} > 18$) con colores de tono azul mientras que para materiales con valores de $Z_{eff}$ intermedio ($11 < Z_{eff} < 18$) se usan colores de tono verde. En la figura 1.3 se muestra una imagen tomada usando este modelo de color.

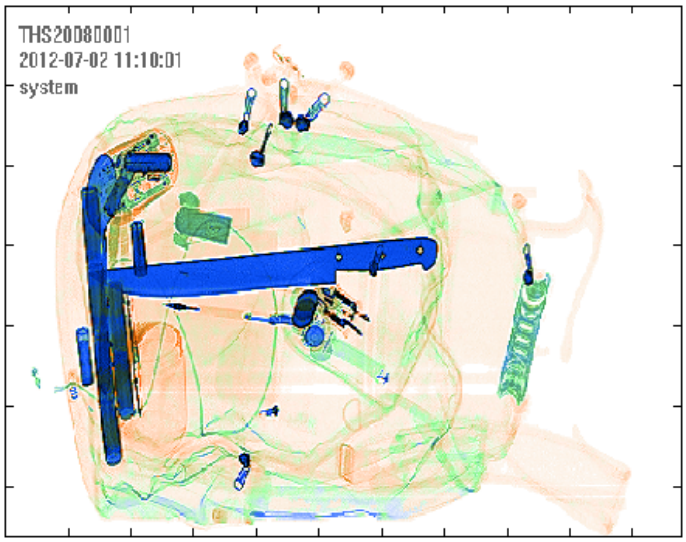


Figura 1.3: *Imagen en pseudo-color entregada por el equipo de inspección de rayos X*

Las imágenes de rayos X son muy diferentes de las imágenes que contienen el espectro visible principalmente porque:

1) Las imágenes de rayos X son imágenes transparentes de poca textura, el valor de los píxeles representa la atenuación de múltiples objetos.

2) Pueden estar muy desordenadas.

3) Son ruidosas debido a la baja energía propia de las imágenes de rayos X [3].

Los objetos en las imágenes de espectro visible son opacos y se ocultan entre ellos. Por el contrario los rayos X penetran los objetos. Como resultado los objetos atenúan la señal

y afectan los valores finales de intensidad. A pesar de las dificultades que poseen estas imágenes de rayos X se puede aprovechar la información extra como son las imágenes de falso color, las múltiples vistas que devuelven algunos equipos y el color específico (que está relacionado con el material) para tareas de clasificación tal como especifica Bastan [3].

## 1.3. El método BoVW

A continuación se procede a explicar la teoría del método BoVW para posteriormente aplicarlo en el contexto de imágenes de rayos X de equipos de energía dual en el reconocimiento de armas de fuego.

Existe una tendencia a utilizar puntos clave o puntos de interés local en la clasificación de imágenes. Estos puntos clave son regiones de una imagen que contienen información relevante que pueden ser representados por varios descriptores. Después de haberlos detectados en una colección de imágenes mediante un algoritmo de extracción de características, los descriptores de los puntos clave, son agrupados mediante un algoritmo de agrupación de características (generalmente k-means) en una cierta cantidad de grupos (*clusters*). Tratando a cada uno de estos grupos como una "palabra visual" que representan un patrón local relacionado con los puntos clave de ese grupo. Al conjunto de estos grupos se les llama "vocabulario de palabras visuales" que describe todos los patrones locales de un conjunto de imágenes. De manera que una nueva imagen puede ser representada a través del conteo (histograma) de las palabras visuales del vocabulario presentes en dicha imagen. A esta representación se le conoce como "Saco de Palabras Visuales" (*Bag-of-Visual-Words* BoVW). Para ello es necesario asociar cada descriptor de la nueva imagen a una o varias palabras visuales del vocabulario, este proceso se conoce como cuantificación vectorial Yang en 2007 [7]. En la figura 1.4 se muestra el diagrama del algoritmo clasificador mediante BoVW.

Posteriormente se procede a construir un clasificador con estos histogramas de palabras visuales que son los vectores de rasgos que caracterizan a las imágenes. El clasificador se construye mediante una etapa de aprendizaje llamada entrenamiento, con histogramas de un conjunto de imágenes positivas y negativas es decir que pertenezcan a dos clases diferentes, que presenten diferencias visuales. De esta forma el clasificador aprende a dis-

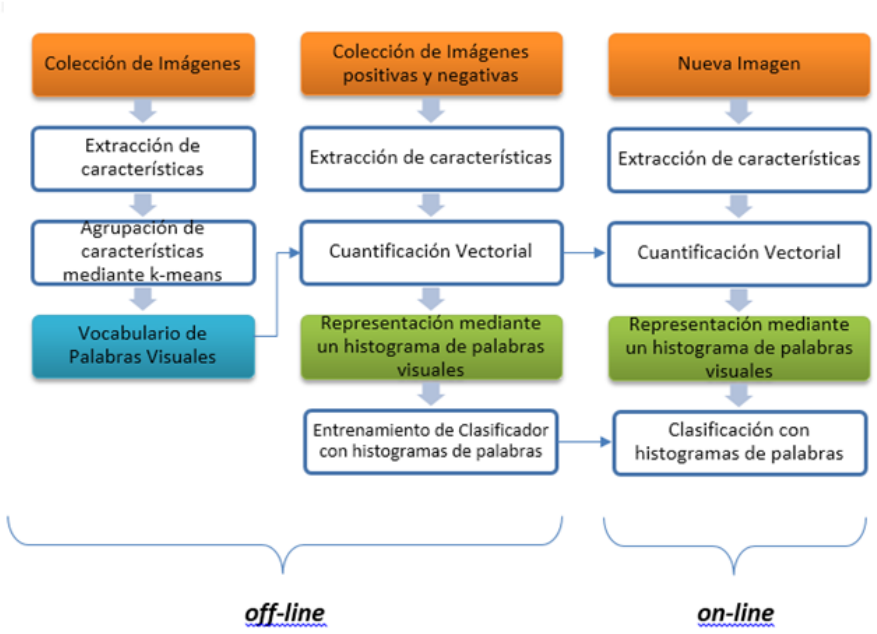


Figura 1.4: *Diagrama general de un algoritmo clasificador basado en BoVW*

cernir de manera que cuando reciba un nuevo histograma de palabras visuales sepa a que clase pertenece la imagen correspondiente.

Resumiendo, el método BoVW está constituido por dos fases. La primera está relacionada con la construcción del vocabulario de palabras visuales y el entrenamiento del clasificador que se realiza *off-line*, durante el entrenamiento del algoritmo. La segunda es la representación de una nueva imagen con un histograma de palabras visuales para determinar a qué clase pertenece que se realiza *on-line* en el momento del reconocimiento. La principal ventaja del método BoVW es su simplicidad, su eficiencia computacional y su invariancia a transformaciones como la oclusión y la iluminación según Csurka en 2004 [5].

## 1.4. Extracción de Características

A continuación se procede a explicar con más detalle los pasos del método BoVW. El primer paso es la extracción de características de un conjunto de imágenes, este consiste

en hallar los descriptores de los puntos clave de una colección de imágenes. Se puede utilizar para ello un algoritmo de extracción de características en imágenes, entre los más conocidos se encuentran SIFT [17], HOG [18], SURF [19], detector de Harris [20], MSER [21], entre otros. El detector de puntos clave y el descriptor no necesariamente tienen que pertenecer al mismo algoritmo de los mencionados.

Los descriptores extraídos deberían ser invariantes a las variaciones que son irrelevantes en la tarea de clasificación (transformaciones de la imagen, variaciones de iluminación y oclusión) pero deben llevar información suficiente para ser discriminativo al nivel de la clase [5].

En la figura 1.5 se aprecia un ejemplo de extracción de características en regiones elípticas de un conjunto de imágenes, $R^d$ representa el espacio de características.

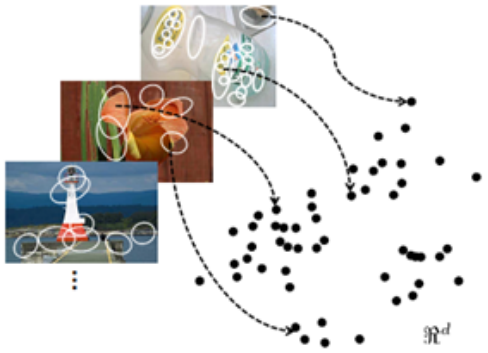

Figura 1.5: *Extracción de los puntos clave.* (Fuente: [22])

### 1.4.1. Algoritmo SIFT

El algoritmo SIFT Transformada Característica Invariante a Escala (*Scale Invariant Feature Transform*) fue presentado por Lowe en 2004 [17]. Este método permite la extracción de puntos de interés o puntos clave distintivos de una imagen que pueden ser utilizados en el reconocimiento de objetos en imágenes. Su éxito frente a otros métodos utilizados radica en que estos puntos de interés son invariantes a escala, rotación, cambios de puntos de vista 3D, adición de ruido, y es parcialmente invariante a cambios de iluminación. Estas características hacen a este método lo suficientemente robusto para la detección y el reconocimiento de objetos en imágenes y video [17].

Para detectar los puntos SIFT lo primero que se hace es representar la imagen en diferentes escalas y tamaños mediante filtrados sucesivos con la función gausiana de diferentes escalas (sigmas). Posteriormente se ejecuta la Diferencia de Gausianos (que es la resta de dos imágenes filtradas consecutivas) para comenzar a identificar los puntos de interés. Para mayor detalle revisar Lowe [17].

Los puntos clave de SIFT son una región circular de la imagen con una orientación. En la figura 1.6 se pueden observar un ejemplo donde se detectaron diferentes puntos clave. Los puntos clave tienen las siguientes características:

• Las coordenadas del centro del punto clave $x$ y $y$.Determinan la posición del punto clave.
• La escala es el parámetro $\sigma$ de la función gausiana con la cual se realiza el filtrado de la imagen, para mayor detalles revisar Lowe [17]. La escala permite que el punto clave sea invariante a cambios de tamaños de la imagen.
• La orientación que expresa la dirección del punto en radianes. Permite que el punto clave sea invariante a rotación.

Los puntos con una región circular mayor corresponden a puntos detectados en escalas mayores, donde la imagen es filtrada con un $\sigma$ mayor de la función gausiana y viceversa para puntos con una menor región circular. En la figura 1.6 se pueden observar estas características.

El algoritmo SIFT construye sobre cada punto clave un descriptor que no es más que un histograma espacial en 3D de los gradientes de la imagen. En la figura 1.7 se observa la construcción del descriptor SIFT.

Sobre el punto clave se proyecta una ventana gausiana circular (ver figura 1.7 centro), dentro se puede observar que hay 16 regiones (rectángulos en magenta grueso). Se obtiene un histograma de orientación para cada una de estas regiones. Los 360 grados de orientación se cuantifican en 8 niveles, donde cada nivel corresponde a un cambio en la orientación de 45 grados. La longitud de cada flecha corresponde a la suma de las magnitudes de los gradientes que se encuentran próximos a esa dirección dentro de la región. Las coordenadas espaciales son cuantificadas en cuatro niveles, donde cada nivel corres-

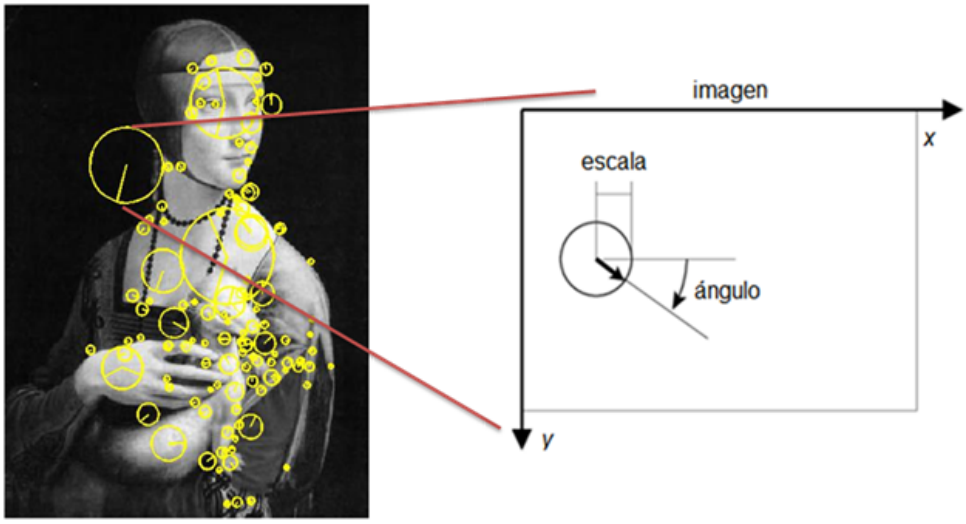


Figura 1.6: *Ejemplo de puntos clave SIFT. Características.*

ponde a una región [17]. Finalmente el descriptor está constituido por un arreglo de 4x4 de histogramas con 8 orientaciones en cada cuadrado. El descriptor SIFT está conformado por 4x4x8=128 valores característicos para cada punto clave.

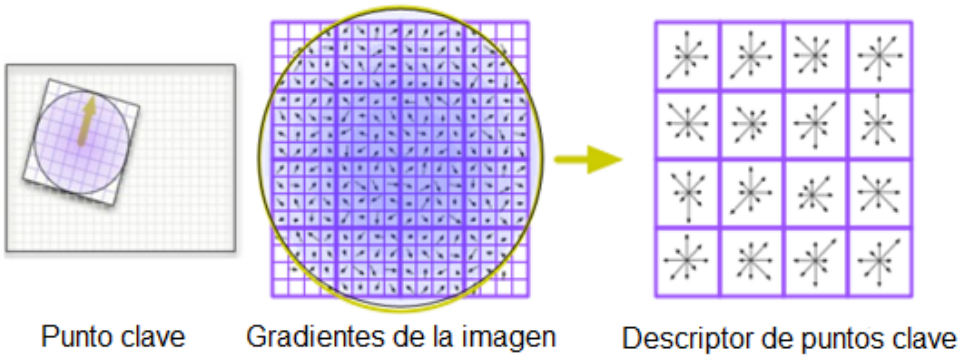


Figura 1.7: *Descriptor SIFT*

Entre los parámetros del algoritmo se encuentra el *factor de magnificación* que determina el tamaño del descriptor multiplicándolo por la escala en que se encuentra el punto clave. Es decir el tamaño del descriptor depende directamente del factor de magnificación y de la escala en que se encuentra el punto clave.

El descriptor SIFT es uno de los descriptores más utilizados en la extracción de características y ha tenido muy buenos resultados en aplicaciones de visión por computadora.

Se ha demostrado que tiene mejor desempeño frente a otros en diferentes contextos según Mikolajczyk en 2005 [23].

### 1.4.2. DSIFT una variante rápida de SIFT

DSIFT es una versión rápida de SIFT para un conjunto denso de puntos de interés (definidos a priori). Este algoritmo es aproximadamente equivalente a ejecutar SIFT en una cuadrícula de puntos densa a una escala y orientación fija. La principal ventaja de la versión densa de SIFT es que es mucho más rápida que el SIFT original [24]. Basado en el SIFT original, DSIFT tiene algunas nuevas características que explican el porqué es más rápido que el SIFT estándar:

**1.** La localización de cada punto clave no es por la característica del gradiente del píxel, sino por una ubicación predefinida.

**2.** La escala de cada punto clave es la misma, también es predefinida.

**3.** La orientación de cada punto clave es siempre cero.

Con estas nuevas características DSIFT puede extraer mayor cantidad de puntos clave en menos tiempo de lo que lo hace SIFT [24].

DSIFT especifica el tamaño del descriptor mediante el parámetro tamaño de ranura o *bin size* el cual controla el tamaño de la ranura espacial de SIFT en píxeles. En el descriptor del SIFT estándar, el tamaño de la ranura está relacionado con la escala del punto clave y por el factor de magnificación, el cual por defecto está en 3. Como consecuencia un descriptor de DSIFT con el tamaño de la ranura igual a 5 corresponde a un punto clave de SIFT a escala 5/3=1.66 [24]. En la figura 1.8 se aprecia la geometría del descriptor DSIFT.

## 1.5. Agrupación de características mediante k-mean

Después de la extracción de las características principales se hace necesario utilizar un algoritmo de agrupamiento para organizar y agrupar por similitudes los datos. Para lograr esto se utiliza el algoritmo de agrupamiento *k-means* que es probablemente el más utilizado para la construcción del vocabulario de palabras visuales. Su propósito es dividir

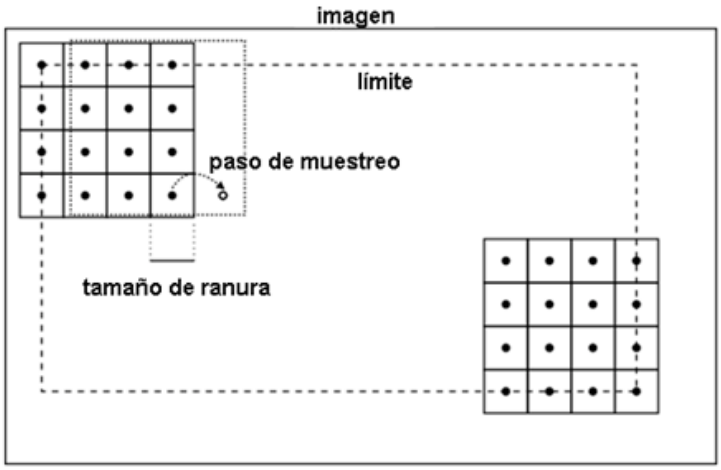


Figura 1.8: *Geometría del descriptor DSIFT.* (Fuente: [24])

un conjunto de vectores en $k$ grupos distintos alrededor de un vector media común. Es decir se trata de encontrar $k$ centros distintos (de ahí el nombre *k-means*) conocidos también como centroides. Estos centroides deben representar al patrón compartido por los puntos claves en ese grupo [5]. Estos grupos son las llamadas *palabras visuales* y una colección de estas palabras se conoce como *vocabulario visual*. El número de grupos o palabras visuales determinan el tamaño del vocabulario visual. Al conjunto de los centroides se le conoce como *codebook* [25] [26].

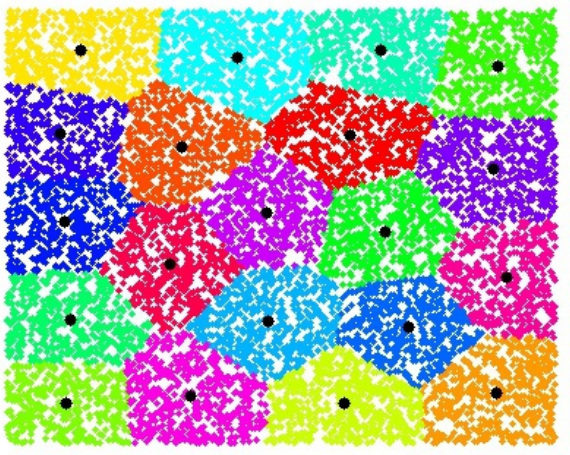

Figura 1.9: *Ejemplo de un vocabulario de palabras visuales. Los puntos negros son los centroides, los grupos de color son las palabras visuales.* (Fuente: [25])

Dado un conjunto de observaciones $(x_1, x_2, \ldots, x_n)$, donde cada observación es un vector real de $d$ dimensiones, *k-means* construye una partición de las observaciones en $k$ conjuntos $S = S_1, S_2, \ldots, S_k$ con el fin de minimizar la suma del cuadrado de las distancias de los puntos al centro de su grupo [26].

$$\arg_{\mathbf{s}} \min \Sigma_{i=1}^{k} \Sigma_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \mu_i\|^2 \tag{1.2}$$

El algoritmo *k-means* busca $k$ vectores $\mu_1, \ldots, \mu_k$ de manera que sea minimizado el error acumulativo. Donde $\mu_i$ (los centroides) es la media de puntos en $S_i$. El algoritmo es implementado mediante un proceso iterativo donde se pretende alcanzar el mínimo, ver expresión (1.2). Pero no siempre se obtiene el mismo resultado.

Según se muestra en la figura 1.10 cada centroide $\mathbf{w}$ representa a todos los descriptores en cada palabra visual. Este se obtiene a través del promedio de la ubicación de todos los elementos de su grupo.

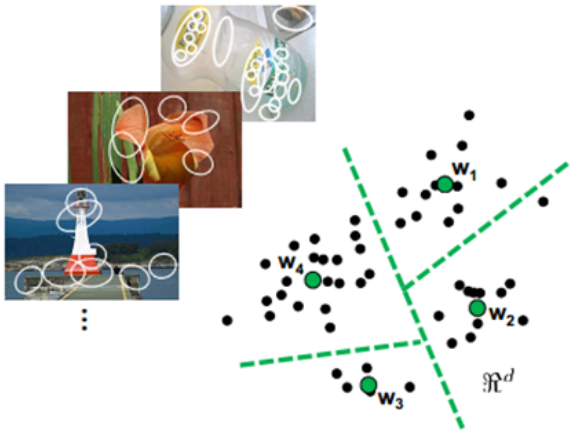


Figura 1.10: *Aplicación de k-means. Se aprecian cuatro palabras visuales con sus correspondientes centroides* $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$ *y* $\mathbf{w}_4$. (Fuente: [22])

## 1.6. Cuantificación Vectorial y obtención del histograma de palabras visuales

En el próximo paso se desea expresar una nueva imagen de entrada mediante la frecuencia de repetición de sus características visuales. Usando la cuantificación vectorial, cada nuevo descriptor, es asignado al centroide más cercano, es decir cada descriptor se asocia a la palabra visual más cercana del vocabulario como muestra la figura 1.11. Posteriormente se procede a construir un histograma de palabras visuales. El histograma de palabras visuales es una representación gráfica de la cantidad de descriptores que fueron asociados a cada palabra visual. Este histograma es el vector de rasgos o vector característico que se utiliza en el entrenamiento del algoritmo clasificador. En la figura 1.11 se observa cómo se construye el histograma. Se puede observar que la palabra visual $\mathbf{w}_2$ es a la que mayor cantidad de descriptores se le asoció, un total de 6 descriptores. Mientras que a $\mathbf{w}_1$ y $\mathbf{w}_3$ solo cuentan un descriptor.

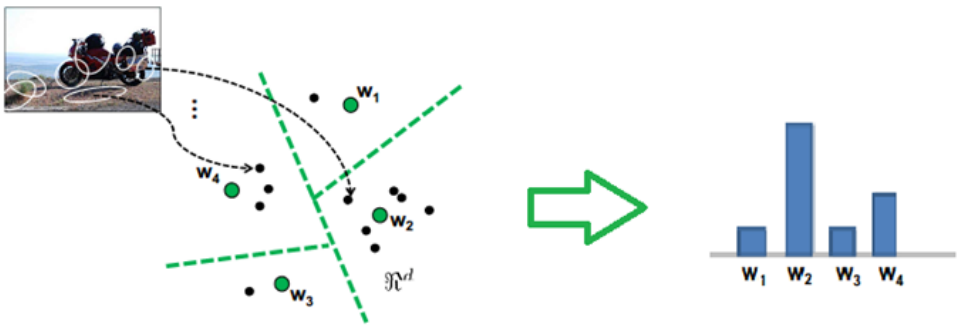


Figura 1.11: *Cuantificación vectorial y obtención del histograma de palabras visuales.* (Fuente: [22])

## 1.7. Parámetros de BoVW

Con el objetivo de profundizar en el método BoVW, en esta sección se presenta un análisis de algunos los principales parámetros que se tienen en cuenta hasta la construcción del histograma de palabras visuales.

### 1.7.1. Tamaño del Vocabulario

El tamaño del vocabulario está determinado por la cantidad de grupos que se desean en el proceso de agrupamiento del algoritmo k-means. Elegir el tamaño adecuado del vocabulario representa un compromiso entre discriminación y generalización. Un vocabulario pequeño que contenga pocas palabras visuales no es muy discriminatorio porque diferentes puntos clave pueden pertenecer a una misma palabra visual. A medida que se incrementa el tamaño del vocabulario las características son más discriminativas pero también son menos generalizables y aumenta la posibilidad de que haya palabras visuales que agrupen únicamente ruido, es decir, puntos clave similares pueden pertenecer a diferentes palabras visuales [5] [7] . En la literatura no hay un consenso exacto sobre el tamaño que debe tener el vocabulario. También depende de las imágenes que se utilicen y de la aplicación. Los trabajos que se han hecho comprenden tamaños entre varios cientos hasta decenas de miles [7], aunque se ha demostrado que con vocabularios de mayor tamaño es más probable que sea más precisa la clasificación sin quitar de vista el costo computacional [27] [26]. A mayor tamaño del vocabulario mayor separabilidad lineal de las clases, incluso puede llegar a un punto en que no sea necesaria la utilización de un kernel en el entrenamiento del algoritmo clasificador [26].

Existen otras técnicas para la construcción de un vocabulario como son las investigaciones de Perronnin en 2006 [28] donde define el concepto de vocabulario universal y vocabulario adaptado. Un vocabulario universal es aquel que se construye con todas las imágenes posibles de todas las clases. Un histograma construido con este vocabulario no es lo suficientemente poderoso como para ayudar a distinguir las clases. Un histograma adaptado es aquel que se construye con imágenes que pertenecen a una misma clase. Si una imagen pertenece a una clase $c$ es recomendable describirla con las palabras visuales de esa clase $c$ que con las palabras del vocabulario universal, pero si la imagen pertenece a otra clase entonces las palabras del vocabulario universal la describirán mejor [28]. Se pueden construir histogramas combinados en base a estos vocabularios de manera que una imagen puede ser descrita con un conjunto de estos histogramas. Cada uno contiene una contribución del vocabulario universal y del adaptado.

### 1.7.2. Información espacial

La principal desventaja de BoVW es la carencia de información espacial entre las palabras visuales porque el histograma solo cuenta la ocurrencia de ellas no su relación espacial. La falta de relación espacial entre las palabras visuales puede ser tanto una ventaja como una desventaja en dependencia de la aplicación. Por un lado, mostrando la ocurrencia de las palabras visuales, no la relación espacial entre ellas, se obtiene una significante flexibilidad a los puntos de vista y cambios de posición. Por el otro lado la relación espacial entre los puntos clave puede ser un importante factor discriminatorio que BoVW pierde [22]. Si se tiene una imagen y se divide en varias piezas como si de un rompecabezas se tratase donde cada pieza corresponde a una palabra visual, el histograma de la imagen armada y desarmada serán iguales. Esto ocurre si no se tiene en cuenta la información espacial.

Existen algunas técnicas para incluir la información espacial ya que la misma puede contener información útil para la tarea de clasificación. Una forma es dividir la imagen en rectángulos de iguales dimensiones, se calculan las palabras visuales de cada una de estas regiones y se concatenan las características de estas regiones dentro de un vector característico de mayor dimensión [7]. Dividiendo la imagen en $m\ x\ n$ regiones aumenta la dimensionalidad del vector característico o vector de rasgos a $m\ x\ n$ veces, elevando el costo computacional.

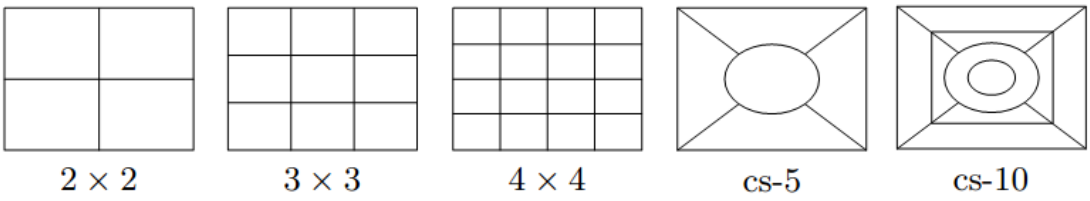


Figura 1.12: *Formas de dividir la imagen para incluir la información espacial.* (Fuente: [29])

Existen otras formas de dividir la imagen a parte de la división en rectángulos como se puede observar en la figura 1.12. Esta propuesta fue publicada por Viitaniemi en 2009 [29]. Además de introducir la técnica de *soft tiling* frente al tradicional *hard tiling* demostrando su superioridad en las pruebas que realizó. El *soft tiling* [29] consiste en asignar los puntos

de interés no solamente a una región espacial de la imagen (tradicional *hard tiling*) sino en diferentes regiones en dependencia de la cercanía del punto de interés a la región. Esto se determina mediante un conjunto de máscaras suavizadas. El *soft tiling* es muy parecido al pesado suave que se utiliza en la cuantificación vectorial para la construcción del histograma de palabras visuales que se explica en la próxima sección.

Otra técnica utilizada para incluir la información espacial es mediante una pirámide espacial que propone Lazebnik en 2006 [30] y que se menciona también en [29]. Esta consiste en concatenar los vectores característicos obtenidos de diferentes niveles de divisiones de la imagen, es decir de divisiones rectangulares de *1x1*, *2x2* y *4x4*. Donde se obtiene un vector característico de mayor dimensionalidad que cuando se utiliza un solo nivel.

Cuando se utilizan técnicas para incluir información espacial hay que tener en cuenta el tamaño del vocabulario. Es mejor utilizar vocabularios pequeños cuando se divide la imagen en muchas regiones (como por ejemplo *6x6*) que vocabularios de gran tamaño y viceversa [29].

En la práctica, por la dimensión del descriptor asociado a cada punto, muchas de las características extraídas se solapan. Implícitamente esto provee cierta dependencia geométrica, o sea, cierta dependencia espacial entre los descriptores. Esto quiere decir que aunque no se tuviese en cuenta la mayor parte de la información espacial, siempre hay una pequeña porción presente [22].

### 1.7.3. Técnicas de asociación de pesos en los histogramas

En el trabajo de Yang [7] se mencionan tres términos de peso para los histogramas basados en las técnicas usadas en el área de reconocimiento de textos. El primero *tf* (*term frequency*), el segundo *idf* (*inverse document frequency*) y el tercer término es el factor de normalización. Estos términos se encuentran resumidos en la tabla 1.1. Resaltar que $tf_i$ es el número de veces que una palabra visual $t_i$ aparece en una imagen. En la tabla 1.1 $N$ es el número total de imágenes y $n_i$ es el número de imágenes que tienen la palabra visual $t_i$. Se ha usado los vectores que cuentan la cantidad de palabras visuales para la clasificación de imágenes (es decir $tf$) y $tf$, *idf* para la búsqueda de imágenes por contenido.

Tabla 1.1: *Fórmulas de pesos para los histogramas de palabras visuales.* (Fuente: [7])

| **Nombre** | **Factores** | **Valor para $t_i$** |
|---|---|---|
| bxx | binario | 1 si $t_i$ está presente, 0 si no |
| txx | $tf$ | $tf_i$ |
| txc | $tf$, normalización | $\frac{tf_i}{\Sigma_i tf_i}$ |
| tfx | $tf$, $idf$ | $tf_i \log(N/n_i)$ |
| tfc | $tf$,$idf$, normalización | $\frac{tf_i \log(N/n_i)}{\Sigma_i tf_i \log(N/n_i)}$ |

El mejor esquema de peso en la clasificación de textos no garantiza el mejor rendimiento en la clasificación de imágenes, principalmente por el factor de normalización. Mientras el factor de normalización tiene la ventaja de eliminar la diferencia de tamaño entre las imágenes, puede tener un efecto negativo debido a que no en todas las imágenes se extraen la misma cantidad de puntos clave [7].

También existen otras técnicas para asignar pesos a los histogramas. En la representación de BoVW original dos descriptores son considerados idénticos si son asignados a la misma palabra visual (centroide) o completamente diferentes si son asignados a diferentes palabras visuales. Puede suceder que dos rasgos visuales muy cercanos, sean asignados a palabras visuales diferentes (ver figura 1.13). En la práctica este tipo de asignación (asignación dura) conduce a errores de cuantificación debido a la variabilidad en el descriptor. Variabilidad que puede venir de diversas fuentes como el ruido, cambios de iluminación o inestabilidad en el proceso de detección [31].

El pesado suave o asignación suave (*soft-assignment*) fue propuesto por Philbin en 2008 [31] para contrarrestar el error de cuantificación. Consiste en asignar un peso a las $r$ palabras visuales más cercanas (típicamente $r = 3$). Este peso está determinado por la expresión:

$$\mathbf{w}(d) = e^{-\frac{d^2}{2\sigma^2}} \tag{1.3}$$

Donde $d$ es la distancia Euclideana del descriptor al centroide y $\sigma$ es el parámetro del método [31].

En la figura 1.13 se muestran un conjunto de descriptores (números) y palabras visuales con su centroide (letras). Si se utiliza asignación dura los descriptores 1, 2, 3 serían asignados a la palabra visual A y 4 a C, nunca 3 y 4 serían asignados a la misma palabra visual estando incluso muy cerca en el espacio de características. Sin embargo en el pesado suave 3 y 4 serían asignados a A, B y C con sus correspondientes pesos.

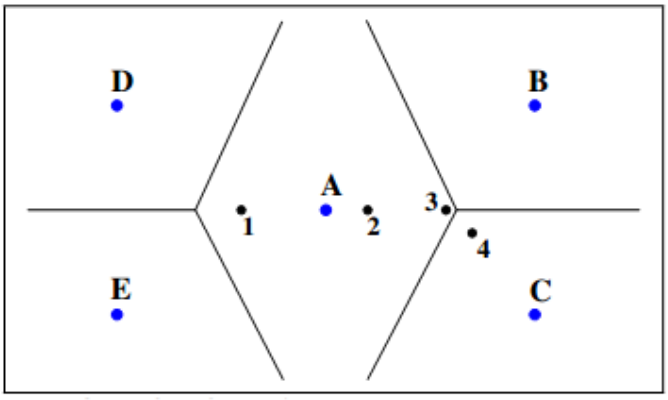


Figura 1.13: *Beneficios del pesado suave.* (Fuente: [31])

## 1.8. Etapa de Clasificación

La clasificación en visión por computadora consiste en reconocer que objeto u objetos pertenecen a una o varias clases. Es el proceso de decisión mediante el cual se le asigna a los objetos su pertenencia a una clase.

La etapa de clasificación en los algoritmos supervisados se basa en dos pasos separados para predecir las clases a la que pertenecen las imágenes: entrenamiento y prueba. Durante el entrenamiento son enviadas al algoritmo clasificador imágenes etiquetadas, es decir imágenes que se conoce a que clase pertenece, para construir el clasificador. Posteriormente son utilizadas las imágenes de prueba, para evaluar cuán preciso es el clasificador en la decisión de determinar a qué clase pertenece.

En el estado del arte se han desarrollado diversos métodos de aprendizaje basándose en la representación de Saco de Palabras Visuales. Estos métodos pueden ser divididos en dos tipos: modelos generativos y los modelos discriminativos. Los modelos generativos

desarrollados en el contexto de reconocimiento de textos son adaptados al contexto de visión por computadora. Ejemplo de ellos se encuentran el clasificador Naïve Bayes y los modelos Bayesianos Jerárquicos. Entre los modelos discriminativos se encuentran las Máquinas de Soporte Vectorial (SVM) también utilizadas en la clasificación de textos.

### 1.8.1. Fundamentos de las Máquinas de Soporte Vectorial (SVM)

Debido a su forma de representar los datos mediante histogramas normalizados, el algoritmo clasificador más utilizado en BoVW, son las Máquinas de Soporte Vectorial (*Support Vector Machines*, SVM). Las SVM fueron introducidas por Vladimir Vapnik en 1995. Las máquinas de soporte vectorial son un conjunto de algoritmos de aprendizaje supervisado. Una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. Los algoritmos SVM pertenecen a la familia de los clasificadores lineales. También se le conoce como clasificador de margen máximo.

Existen infinitos hiperplanos de separación entre dos conjuntos finitos de datos (cuando el problema es separable). Los clasificadores SVM encuentran un hiperplano que separe dos conjuntos finitos de datos con el máximo margen Vapnik [8]. El margen es definido como las distancia del punto más cercano de entrenamiento al hiperplano de separación.

Dada las siguientes muestras de entrenamiento $\{\mathbf{x}_i, y_i\}$,$i = 1, \ldots, l$ donde ($\mathbf{x}_i \in \mathbf{R}^d$) y las etiquetas $y_i \in \{-1, 1\}$. Todos los hiperplanos en $\mathbf{R}^d$ son parametrizados por el vector $\mathbf{w}$ y la constante $b$, expresado en la ecuación:

$$\mathbf{w^T x} + b = 0 \tag{1.4}$$

Notar que $\mathbf{w}$ es el vector ortogonal al hiperplano. Dado un hiperplano ($\mathbf{w}$ y $b$ representan los parámetros del hiperplano ver figura 1.14) que separe los datos entonces la función de clasificación es:

$$\mathbf{f(x)} = sign(\mathbf{w^T x} + b) \tag{1.5}$$

Esta función de decisión separa correctamente los datos de entrenamiento (y con suerte

otros datos que no ha visto todavía). Dado un hiperplano representado por $(\mathbf{w}, b)$ es igualmente expresado por los pares $\{\lambda\mathbf{w}, \lambda b\}$ para $\lambda \in \mathbf{R}^+$ se define el hiperplano canónico como aquel que separa los datos del hiperplano de una distancia de al menos 1. Entonces satisface:

$$y_i(wx_i + b) \geq 1 \forall i \tag{1.6}$$

Todos los hiperplanos tienen una "distancia funcional"$\geq 1$ que no puede ser confundida con distancia geométrica o distancia euclidiana (también conocido como margen). Para un hiperplano dado todos los pares $\{\lambda\mathbf{w}, \lambda b\}$ definen el mismo hiperplano pero cada uno tiene una distancia funcional diferente hacia un punto de datos dado. Para obtener la distancia geométrica del hiperplano hacia un punto, se debe normalizar por la magnitud $\mathbf{w}$, Boswell en 2002 [32].

$$d((\mathbf{w}, b), x_i) = \frac{y_i(wx_i + b)}{\|\mathbf{w}\|} \geq \frac{1}{\|\mathbf{w}\|} \tag{1.7}$$

Lo que se desea encontrar es el hiperplano que tenga la máxima distancia geométrica al punto de datos más cercano. Esto se logra en la ecuación (1.7) minimizando $\|\mathbf{w}\|$ [32]. El principal método para hacer esto es con los multiplicadores de Lagrange. De la solución del problema se obtiene que el hiperplano óptimo puede ser escrito como:

$$\mathbf{w} = \Sigma_i \alpha_i y_i \mathbf{x}_i \tag{1.8}$$

$$b = y_i - \mathbf{w}\mathbf{x}_i \tag{1.9}$$

$$0 \leq \alpha_i \leq C \ \forall i \tag{1.10}$$

Donde $\alpha = (\alpha_1 \ldots \alpha_l)$ es el vector de los $l$ multiplicadores no negativos de Lagrange y C es una constante. La ecuación significa que el vector $\mathbf{w}$ es una combinación lineal de las muestras de entrenamiento. Los vectores de rasgos de entrada $\mathbf{x}_i$ en el caso de BoVW son los histogramas de palabras visuales. Es decir, si se está utilizando BoVW hay que tomar

los histogramas de palabras visuales pertenecientes al conjunto de imágenes de dos clases para realizar el entrenamiento de la SVM. Puede mostrarse también que:

$$\alpha_i(y_i(\mathbf{w}\mathbf{x}_i + b) - 1) = 0 \ \forall i \tag{1.11}$$

Otra manera de expresar esta idea consiste en que cuando la distancia funcional es estrictamente mayor que 1 (cuando $y_i(\mathbf{w}^\mathbf{T}x_i + b) > 1$) entonces $\alpha_i = 0$. De manera que solo los datos cercanos contribuyen a formar $\mathbf{w}$. Estas muestras de entrenamiento para las cuales $\alpha_i > 0$ son las conocidas como vectores de soporte. Las mismas son las únicas necesarias para definir y encontrar el óptimo hiperplano (si se rechazan todos los demás vectores y se realiza el entrenamiento con los de soporte únicamente el hiperplano de separación será el mismo). Los vectores de soporte son los casos que están en la frontera decisión. Ver figura 1.14. Entonces la función de clasificación es:

$$\mathbf{w}^\mathbf{T}\mathbf{x} + b = \Sigma_i \alpha_i y_i \mathbf{x}_i \mathbf{x} + b \tag{1.12}$$

Los datos de un problema no son siempre linealmente separables. Los clasificadores SVM tienen dos formas para resolver este problema. Una forma es mediante la expansión del espacio de los datos $\mathbf{X}$ hacia otro espacio característico, donde el problema sea linealmente separable, a través de un mapeo de características. Ver el ejemplo de la figura 1.15. La otra alternativa consiste en permitir muestras mal clasificadas como parte de la solución y penalizarlas en proporción a su distancia a la frontera de decisión.

### 1.8.2. El truco del kernel

Un mapa de características (*feature map*) $\Psi(\mathbf{x})$ es una función que mapea a un vector $\mathbf{x}$ del espacio de características original $\mathbf{X}$, hacia un nuevo espacio vectorial a través de un producto escalar, de manera que:

$$\forall \ \mathbf{x}, \mathbf{y} : K(\mathbf{x}, \mathbf{y}) = \langle \Psi(\mathbf{x}), \Psi(\mathbf{y}) \rangle \tag{1.13}$$

Este segundo espacio de características puede tener una dimensión mayor o incluso infinita. Una de las ventajas de SVM es que puede ser formulada completamente en

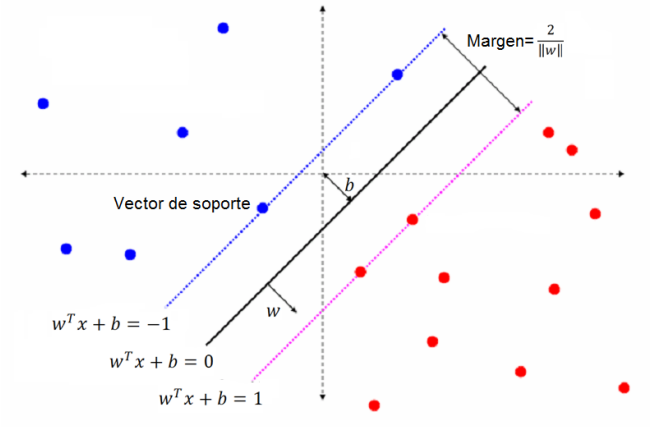


Figura 1.14: *Representación del hiperplano de separación en el proceso de entrenamiento de una SVM*

términos de productos escalares en el segundo espacio de características. De esta forma se introduce la función núcleo o *kernel* como el producto escalar entre dos mapas de características, ver la ecuación (1.13) Introduciendo la función *kernel* entonces la función de clasificación quedaría:

$$f(x) = sign(\Sigma_i y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b) \tag{1.14}$$

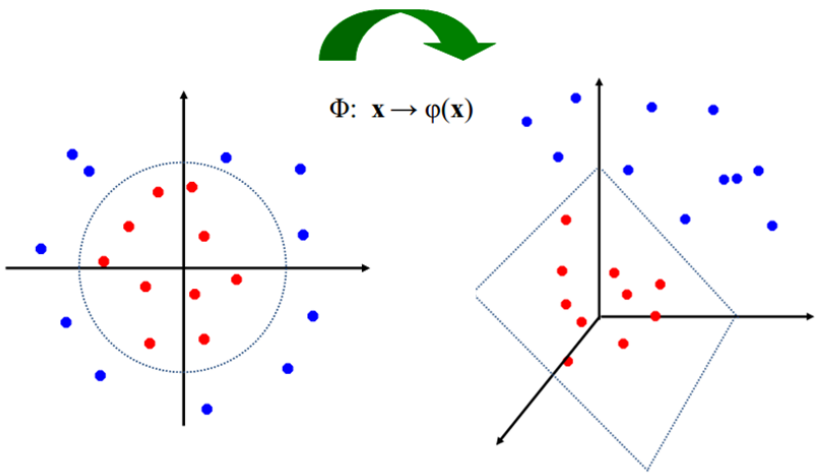


Figura 1.15: *Expansión de los datos mediante el mapeo de características*

Se puede apreciar que la diferencia con la ecuación (1.12) es la presencia del kernel $K(\mathbf{x}, \mathbf{x}_i)$ que sustituye el producto escalar $\mathbf{x}\mathbf{x}_i$

El kernel depende de la aplicación que se desee desarrollar y necesita ser determinado por el usuario. Entre los kernels más comunes se encuentran:
Lineal: $K(\mathbf{x}, \mathbf{y}) = x^T y$
Polinomio: $K(\mathbf{x}, \mathbf{y}) = (x^T y + c)^n$
Función radial base RBF: $K(\mathbf{x}, \mathbf{y}) = e^{-\gamma \|x-y\|^2}$
$\chi^2$ o (*Chi-squared*): $K(\mathbf{x}, \mathbf{y}) = e^{-\gamma \chi^2(x,y)}$ donde $\chi^2(x, y) = \Sigma_j \frac{(x_j - y_i)^2}{x_j + y_i}$
Hellinger: $K(\mathbf{x}, \mathbf{y}) = \sqrt{xy}$
Intersección de histogramas: $K(\mathbf{x}, \mathbf{y}) = min\{x, y\}$

### 1.8.3. Redefinición del problema mediante la función de costo

La otra alternativa para la no linealidad de las clases consiste en permitir errores de clasificación en la formulación de la SVM. Para poder tratar con datos que no son linealmente separables el análisis previo puede ser generalizado introduciendo la función de pérdida $\ell_i \geq 0$ de modo que la ecuación (1.6) es modificada a:

$$y_i(wx_i + b) \geq 1 - \ell_i \ \forall i \tag{1.15}$$

La función de pérdida $\ell_i \neq 0$ en (1.15) es para aquellos para los cuales el punto $x_i$ no satisface la expresión (1.6). La función de pérdida depende de $\langle \mathbf{w}, \mathbf{x} \rangle$ y se conoce como bisagra o *hinge* la cual en la SVM estándar es:

$$\ell_i(\langle \mathbf{w}, \mathbf{x} \rangle) = max\{0, 1 - y_i \langle \mathbf{w}, \mathbf{x} \rangle\} \tag{1.16}$$

Es cero solo si $\langle \mathbf{w}, \mathbf{x} \rangle$ es al menos -1 o como máximo 1, dependiendo de la etiqueta $y_i$. Entonces el término $\Sigma_{i=1}^{n} \ell_i(\langle \mathbf{w}, \mathbf{x} \rangle)$ puede ser tomado como algún tipo de medida del error en la clasificación.

Ajustar $\mathbf{w}$ usando únicamente los datos de entrenamiento es usualmente insuficiente. Para que el producto escalar pueda clasificar casos no entrenados, es preferible hacer un compromiso entre la exactitud del ajuste con la regularidad de la aprendida función $\langle \mathbf{w}, \mathbf{x} \rangle$ [24]. La regularidad en la formulación estándar es medida por la norma del vector

$\|\mathbf{w}\|^2$. Promediando la función de pérdida sobre todas las muestras de entrenamiento y adicionando el regularizador pesado por el parámetro $\lambda$ da la función de costo o función objetivo:

$$E(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{n}\Sigma_{i=1}^{n}\ell_i(\langle \mathbf{w}, \mathbf{x}\rangle) \tag{1.17}$$

De manera que el problema del hiperplano óptimo es redefinido como la solución al problema de minimizar la función de costo. Resaltar que esta función objetivo es convexa, de manera que existe un óptimo global. El objetivo es encontrar el valor de $\mathbf{w}$ que minimiza la función de costo.

Otra forma de representar la función objetivo es mediante el parámetro $C$ (visto anteriormente en la expresión (1.10)) que es usado para pesar la pérdida en vez del regularizador. La función objetivo quedaría:

$$E_C(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\Sigma_{i=1}^{n}\ell_i(\langle \mathbf{w}, \mathbf{x}\rangle) \tag{1.18}$$

La constante $C$ es el peso de error que penaliza las muestras mal clasificadas. Cuando $C = \infty$ , el hiperplano óptimo será aquel que separe completamente los datos (si existe). Para un valor finito de $C$ el problema se concentra en encontrar un clasificador de "margen suave" (*soft-margin*) que conduce a que algunos de los datos sean mal clasificados. Altos valores para $C$ corresponde a darle mayor importancia a clasificar los datos de entrenamiento correctamente. Valores bajos para $C$ corresponde a un hiperplano con "mayor flexibilidad" que intenta minimizar el error. Valores finitos de $C$ son comunes cuando los datos no son fácilmente separables [32].

Este es el único parámetro libre de ser ajustado en la formulación de la SVM. El ajuste de éste parámetro puede hacer un balance entre la maximización del margen y el permitir errores de clasificación.

La función $E_C(\mathbf{w})$ y $E(\mathbf{w})$ en $\lambda$ son equivalentes (proporcionales) si:

$$\lambda = \frac{1}{nC} \tag{1.19}$$

$$C = \frac{1}{n\lambda} \tag{1.20}$$

## 1.9. Métricas de evaluación

En esta sección se describe la teoría de las métricas que se utilizarán para la evaluación del algoritmo clasificador. Se presentan las curvas ROC, la validación cruzada, el promediado de curvas ROC y su importancia.

### 1.9.1. Curva ROC

Para medir la eficacia del clasificador en la etapa de prueba existen diversas métricas de evaluación. Entre ellas se encuentra la tradicional curva ROC (*Receiver Operating Characteristic*). La curva ROC es una técnica para visualizar, organizar y seleccionar clasificadores binarios basado en sus rendimientos. Ha sido usada en la teoría de detección de señales para representar el compromiso entre la razón de aciertos positivos y la razón de falsas alarmas de clasificadores. También se ha usado para visualizar y analizar el comportamiento de sistemas de diagnósticos, en la medicina y aprendizaje de máquina Fawcett en 2004 [33]. El análisis ROC se relaciona de forma directa y natural con el análisis de costo/beneficio en toma de decisiones diagnósticas.

Una curva ROC es una representación gráfica de la razón de verdaderos positivos (TPR[1], de sus siglas en inglés) frente a la razón de falsos positivos (FPR, de sus siglas en inglés) según se varía el umbral de discriminación obtenidas de las expresiones:

$$TPR = \frac{TP}{TP + FN} \tag{1.21}$$

$$FPR = \frac{FP}{FP + TN} \tag{1.22}$$

Donde $TP$,$TN$,$FN$ y $FP$ son los aciertos positivos, aciertos negativos, falsos negativos y falsos positivos respectivamente. Es decir:

[1]También se le conoce como *hit rate* y *recall*

- $TP$ es el número de muestras que son correctamente clasificadas como positivas.
- $TN$ es el número de muestras que son correctamente clasificadas como negativas.
- $FP$ es el número de muestras que son incorrectamente clasificadas como positivas.
- $FN$ es el número de muestras que son incorrectamente clasificadas como negativas.

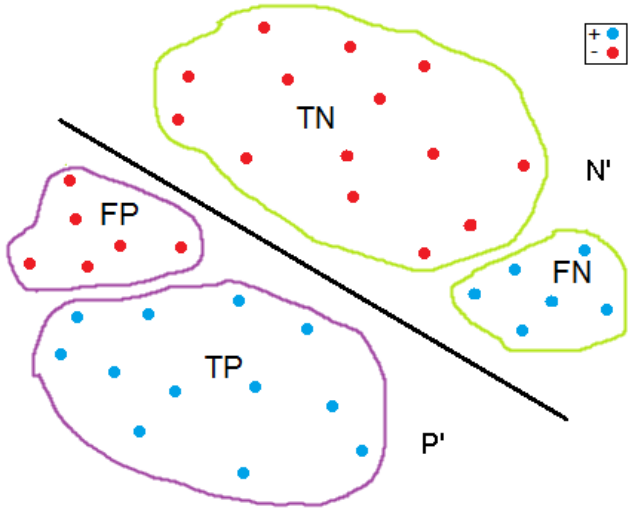


Figura 1.16: *Dibujo para entender las razones de la curva ROC*

En la figura 1.16 aparece un dibujo que representa la distribución de las muestras después de haber realizado una etapa de prueba en el clasificador para comprender mejor estos índices. Los puntos azules son las muestras positivas y los rojos son las muestras negativas (el color corresponde con el etiquetado). Las regiones $P'$ y $N'$ corresponden a las regiones etiquetadas como positivas y negativas por un clasificador. Los cuatro grupos son las cuatro combinaciones explicadas. A estos cuatro valores ($TP$, $TN$, $FP$ y $FN$) representados en forma matricial (ver figura 1.17) se les conoce como matriz de confusión.

Por definición $P = TP + FN$, $N = TN + FP$. La $TPR$ mide hasta qué punto un clasificador o prueba diagnóstica es capaz de detectar o clasificar los casos positivos correctamente. La $FPR$ también se le conoce como falsa alarma, indica el porciento de muestras detectadas incorrectamente como positivas. En la figura 1.18 se visualiza un ejemplo de curva ROC. Se puede apreciar como las razones $FPR$ y $TPR$ corresponden a los ejes $x$ y $y$ respectivamente.

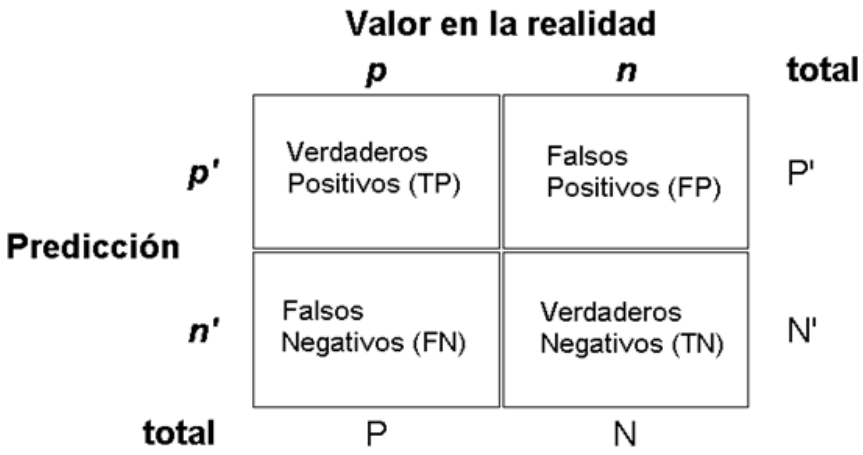


Figura 1.17: *Matriz de confusión*

El AUC (*Area Under the Curve*) área bajo la curva es un indicador de la calidad general de la curva ROC. En la figura 1.18 AUC=95.82 %. La AUC en la curva ROC de un clasificador ideal es 1. Otro indicador es EER (*Equal Error Rate*) razón de error igual, es el punto en la curva ROC que corresponde con tener una probabilidad igual de errores en la clasificación de una muestra positiva o negativa. En la figura del ejemplo EER=9.62 %.[28] El AUC se puede interpretar como la probabilidad de que un clasificador ordenará o puntuará una instancia positiva elegida aleatoriamente más alta que una negativa [33].

Existen otros términos asociados con las curvas ROC como es la precisión y la exactitud o *accuracy*. La precisión [2] es igual al por ciento de muestras positivas que fueron correctamente clasificadas sobre el total de muestras que fueron clasificadas como correctas. Es decir:

$$PPV = \frac{TP}{TP + FP} \tag{1.23}$$

Existe una curva llamada *Precision Recall* que presenta la relación de $TPR$ con la precisión. El eje $x$ sería $TPR$ y el eje $y$ la precisión [33].

La exactitud da una medida general de la eficiencia del clasificador en clasificar correctamente tanto muestras positivas como negativas. Está determinada por la expresión:

[2]También se le conoce como valor positivo predictivo (*positive predictive value* $PPV$)

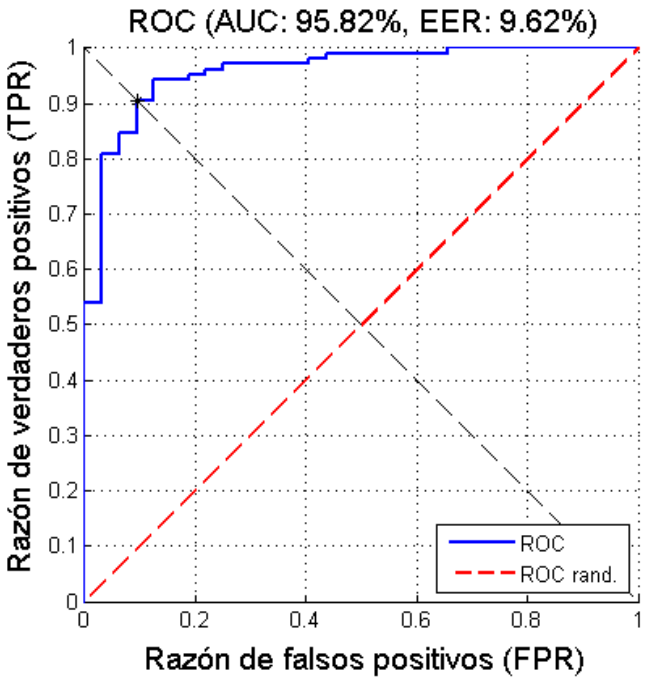


Figura 1.18: *Ejemplo de curva ROC.* (Fuente: [34] )

$$ACC = \frac{TP + TN}{P + N} \tag{1.24}$$

### 1.9.2. Validación cruzada

Existen diversas técnicas para evaluar los resultados de una prueba estadística. Entre ellas se encuentra la validación cruzada. La validación cruzada garantiza que los datos son independientes de la partición entre conjuntos de entrenamiento y prueba. Se utiliza para evitar presentar un resultado afectado por el sobre ajuste del algoritmo de entrenamiento de un clasificador. De manera que da una mayor confiabilidad estadística a los resultados.

La validación cruzada es una mejora del método de retención (*holdout*). El método de retención consiste en dividir los datos en dos conjuntos mutuamente exclusivos denominados entrenamiento y prueba. Es común designar la 2/3 de los datos al conjunto de entrenamiento y el restante 1/3 al conjunto de prueba. El conjunto de entrenamiento se le entrega al algoritmo que construye los parámetros del clasificador y el conjunto de prueba evalúa la precisión en el clasificador. El método de retención es un estimador pe-

simista porque solo una porción de los datos es utilizada en el proceso de entrenamiento. De manera que puede ocurrir un sobre ajuste en el proceso de entrenamiento, es decir que se restrinjan demasiado los parámetros para el conjunto de entrenamiento y se pierda generalización. Entonces los resultados finales pierden confiabilidad y generalización. La evaluación puede depender en gran medida de cómo es la división entre datos de entrenamiento y de prueba, y por lo tanto puede ser significativamente diferente en función de cómo se realice esta división. Debido a estas carencias aparece el concepto de validación cruzada.

Existen dos tipos de validación cruzada: validación cruzada de k segmentos (*k-fold cross-validation*) y la validación cruzada dejando uno fuera (*Leave-one-out*) [35]. La validación cruzada de k segmentos, algunas veces llamada estimación de rotación, consiste en dividir aleatoriamente los datos en k subconjuntos mutuamente exclusivos de aproximadamente igual tamaño. Cada subconjunto se utiliza como datos de prueba y los restantes k-1 como datos de entrenamiento. El algoritmo del clasificador es entrenado y evaluado k veces con cada uno de los subconjuntos seleccionados. Finalmente se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado. Este método es muy preciso puesto que se evalúa a partir de k combinaciones de datos de entrenamiento y de prueba pero tiene una desventaja, y es que, a diferencia del método de retención, es lento desde el punto de vista computacional. En la figura 1.19 aparece un ejemplo de una validación cruzada de 3 segmentos.


Figura 1.19: *Divisiones del conjunto de datos para una validación cruzada de 3 segmentos*

La validación cruzada dejando uno fuera o *Leave-one-out cross-validation* (LOOCV) consiste en separar los datos tal que la prueba se realiza sobre una sola muestra y el entrenamiento con todas las restantes. De manera que habrá tantas iteraciones como

muestras en la base de datos. Es un caso particular de la validación cruzada de k segmentos donde k sería igual al número total de muestras. Este tipo de validación cruzada es muy robusta al utilizar la mayor cantidad de muestras posibles como datos de entrenamiento, pero tiene la desventaja de ser muy costosa computacionalmente.

En principios se pude ajustar un clasificador para que siga aumentando sus índices de clasificación con los datos de prueba, pero a partir de cierto punto tiene la desventaja que aumentan los errores de clasificación frente a nuevos datos (sobre-entrenamiento, pierde generalización), de ahí la importancia de hacer una validación cruzada.

### 1.9.3. Promediado de curvas ROC

Si se está trabajando con curvas ROC y se aplica validación cruzada se va a obtener una curva ROC por cada iteración. De manera que estas curvas ROC hay que promediarlas para obtener el resultado final de la validación cruzada. Para ello existen dos técnicas de promediado de curvas ROC que propone Fawcett en 2004 [33]: el promediado vertical (VA de sus siglas en inglés) y el promediado por umbral (TA de sus siglas en inglés).

El promediado vertical (VA) consiste en promediar los diferentes valores de TPR obtenidos de una validación cruzada de k segmentos. En este método cada curva ROC es tratada como una función de manera que: $tpr = R_i(fpr)$ Esto se realiza seleccionando el máximo $tpr$ para cada $fpr$ e interpolando entre los puntos cuando sea necesario. La curva ROC promedio es la función: $R'(fpr) = mean[R_i(fpr)]$ [33].

El promediado por umbral debe generar un conjunto de umbrales para muestrear, para cada umbral encuentra el punto correspondiente de cada curva ROC y los promedia. Es decir para un umbral dado el algoritmo selecciona de cada curva ROC el punto de mayor puntaje menor o igual que el umbral. Estos puntos son luego promediados separadamente a través de los ejes $X$ y $Y$ de la curva [33].

# Capítulo 2

# Aplicación de BoVW en imágenes de rayos X

## 2.1. Introducción

En este capítulo se presenta la implementación de un algoritmo basado en el método Saco de Palabras Visuales en el contexto de imágenes de rayos X que permita el reconocimiento de objetos desarrollado sobre la plataforma MATLAB 2014ª y la biblioteca de funciones VLFeat [24]. Esta herramienta permite la construcción de un vocabulario de palabras visuales sobre una base de datos de imágenes de rayos x para poder representar estas imágenes mediante los histogramas de palabras visuales. Se realiza el diseño de un clasificador basado en SVM y se presentan los métodos desarrollados para el entrenamiento y evaluación del clasificador.

La biblioteca de funciones VLFeat [24] tiene implementaciones de alta calidad de algoritmos comunes de visión por computadora para desarrollar el modelo Saco de Palabras Visuales. Dicha biblioteca de funciones está desarrollada bajo la licencia GNU GPL con lo cual permite el acceso al código (código abierto), incluye muchas implementaciones optimizadas[1] de algoritmos complejos de visión por computadora, es flexible y portable [36]. Muchas de sus implementaciones son más rápidas que las nativas que utiliza MATLAB[2]

[1]Principalmente los algoritmos de extracción de características.

[2]Como el k-means

.Como VLFeat es de código abierto tiene la ventaja de que pueden modificarse las funciones presentes y adaptarlas a la aplicación deseada. También se utiliza un conjunto de funciones que se apoyan en VLFeat presentes en [37]. Aunque no es objetivo principal de este trabajo realizar un análisis exhaustivo del costo computacional, es conveniente para la realización de los experimentos, debido al elevado volumen de datos manejados.

## 2.2. Extracción de características usando PHOW

Para la extracción de características el algoritmo a utilizar debe priorizar la cantidad de características extraídas que contengan información relevante frente al costo computacional, debido a las condiciones de seguridad de la aplicación. Según los experimentos de Bastan 2011 [3] el algoritmo SIFT tiene mejor rendimiento y arroja mejores resultados que otros. En el trabajo de Turcsany [4], utilizan SURF para la extracción de las características. Este método tiene la ventaja de ser más rápido que SIFT pero con menor cantidad de puntos claves. En base a las exigencias de la aplicación en este trabajo se utiliza el algoritmo PHOW (*Pyramid Histogram Of Visual Words*) propuesto por Bosch en 2007 [38] que calcula una mayor cantidad de puntos a cuatro escalas fijas (definidas a priori) y está basado en los descriptores de SIFT. Esta mayor cantidad de puntos que arroja es conveniente para la aplicación, debido a que como los objetos en las imágenes de rayos X poseen menor textura que en las imágenes del espectro visible es necesario obtener más información de ellos para su clasificación. Además a mayor cantidad de puntos extraídos mayor será la precisión de la clasificación según Nowak en 2006 [27] y Chatfield en 2011 [26].

PHOW es una variante de los descriptores de DSIFT extraídos a múltiples escalas. Permite extraer los descriptores en los tres canales de una imagen en colores y después los concatena. PHOW consiste en calcular los descriptores SIFT en una cuadrícula con espacio de M píxeles. A cada uno de estos puntos los descriptores son calculados sobre cuatro regiones circulares fijas. De manera que cada punto es representado por cuatro descriptores SIFT dado que se calculan sobre cuatro escalas fijas [38]. Estas escalas son definidas modificando el ancho de la ranura espacial del descriptor SIFT a 4, 6, 8 y 10

píxeles respectivamente. Cada valor de estas escalas es usado como el tamaño de la ranura (*bin size*) para la función *vl_dsift*. La orientación es fijada a un valor constante [26]. En la figura 2.1 se puede apreciar una imagen que muestra los puntos SIFT y los puntos PHOW en donde se pueden ver sus diferencias.

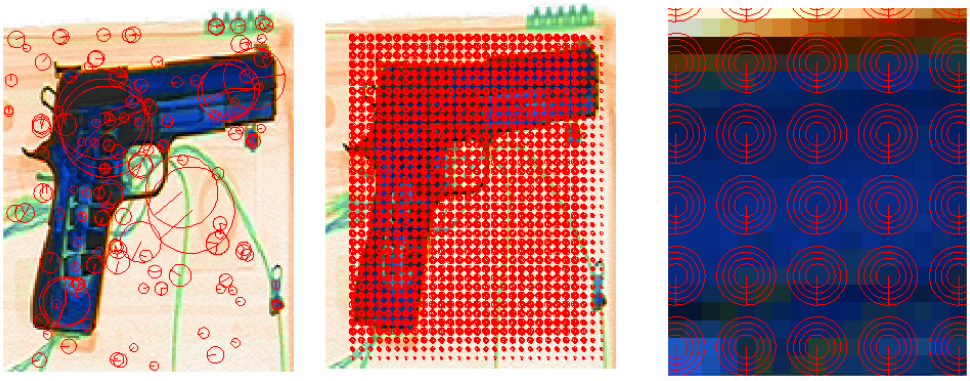

Figura 2.1: *De izquierda a derecha: a) Puntos SIFT, b) puntos PHOW, c) zoom sobre los puntos PHOW*

La biblioteca de funciones VLFeat [24] implementa PHOW en la función *vl_phow* que ejecuta la función *vl_dsift* tantas veces como escalas se hayan seleccionado. Los descriptores que devuelve tienen el mismo formato que los que devuelve *vl_sift*. El vector de los puntos clave contiene las coordenadas del centro de los descriptores, el contraste y el tamaño de la ranura de cada descriptor. La función permite elegir el modelo de color que puede ser escala de grises (*gray*), RGB o HSV.

En resumen, teniendo en cuenta las propiedades de estas imágenes se extrajeron los puntos clave espaciados cada 4 píxeles (M=4) implementado en la función *vl_phow*. También se seleccionaron los canales HSV (*Hue, Saturation, Value*) para la extracción de los puntos. Las características de las imágenes de pseudo-color de rayos X parecen indicar que quedan mejor reflejadas en estos canales. El tono de color está en el canal H, que contiene información del número atómico efectivo, mientras que en la intensidad (en canal V) contiene información de la densidad y el ancho del material [3].

## 2.3. Implementacion del pre-procesamiento

Las armas de fuego en general están compuestas por partes metálicas [3] aunque también algunas poseen partes no metálicas. Son objetos con poca textura, pero existen otros objetos [4] que presentan menos textura aún, por lo que su clasificación no será tan compleja. La poca textura presente en las armas de fuego y las partes metálicas que las componen son características que se puede aprovechar para su reconocimiento. Las imágenes de rayos X en pseudo-color poseen características visuales que necesitan ser aprovechadas. El color se corresponde con el número atómico efectivo del material. Los metales tienen un color que va desde un azul claro hasta un azul oscuro [5] . De manera que constituye una ventaja para la tarea de clasificación si se logran extraer únicamente las características de los objetos de interés (objetos de color azul). De esta forma se distinguen de las características del fondo de poca importancia y aumenta considerablemente la precisión de la clasificación. Se ha demostrado que esta estrategia ha tenido buenos resultados según Bastan [3] y Turcsany [4]. Para hacer una segmentación basada en color se utilizó el método de la esfera (*sphere*) presentado en el libro de Gonzales [39]. Primeramente se calculó experimentalmente un color intermedio mediante el promediado de diferentes tonalidades de azul tomadas de varias armas. Se fijó el radio de la esfera y se mostraron solo aquellos píxeles que se encuentran dentro de dicha esfera en el espacio RGB. En la figura 2.2 se aprecian las regiones tomadas después de esta segmentación.

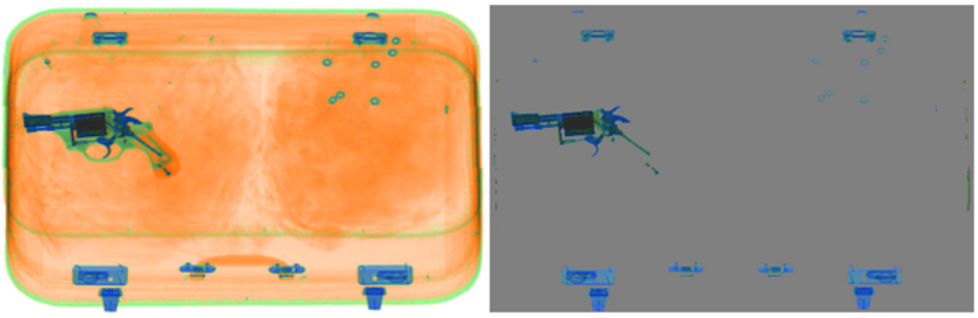

Figura 2.2: *a) Imagen original. b) Regiones que quedan después de hacer la segmentación*

[3] a excepción de las armas de plástico de reciente aparición construidas con impresoras 3D

[4] Como las botellas según explica Bastan [6]

[5] en dependencia del nivel de atenuación del metal

Posteriormente se construyó una imagen binaria donde están resaltadas las zonas de interés. Mediante la operación morfológica de cierre (con un disco de diámetro 6 píxeles como elemento estructural), se cierran los huecos que se encuentran dentro de las regiones en blanco como se muestra en la figura 2.3.

Figura 2.3: *a) Imagen binaria b) Resultado del cierre sobre la imagen binaria*

Después de realizar el cierre quedan puntos y regiones pequeñas que es poco probable que pertenezcan a armas de fuego. Estas regiones son filtradas eliminando aquellas con un área menor a determinado umbral. Para ello se realizó un experimento donde se calculó el área mínima del conjunto de armas de la base de datos. En la figura 2.4 a) se observa cómo queda la máscara binaria después de un filtrado por áreas.

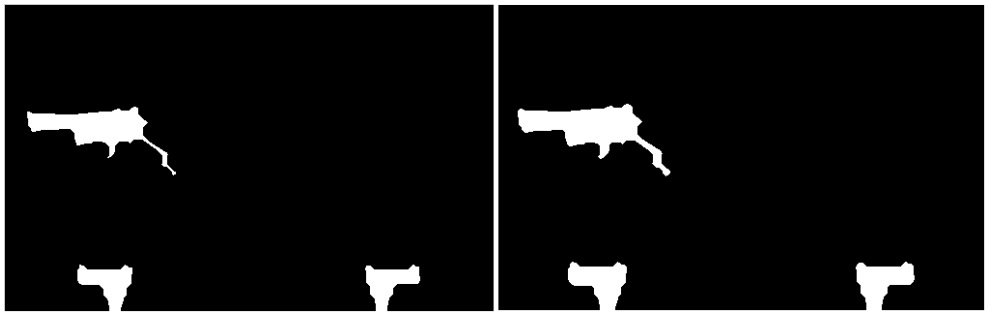

Figura 2.4: *a) Filtrado por áreas b) Dilatación de la imagen*

Muchas de las armas tienen regiones que no son metálicas como es el caso de la figura 2.2 a). Estas partes aunque son la minoría no deben ser rechazadas. Para incluir mejor estas partes no metálicas se realizó la operación morfológica de dilatación, con un disco de diámetro 2 píxeles como elemento estructural. El resultado de esta operación se puede

apreciar en la figura 2.4 b). Además el tamaño del descriptor SIFT en cada punto clave abarca regiones no metálicas.

Finalmente la figura 2.5 muestra el resultado de aplicar el procedimiento de reducción del área de búsqueda antes de aplicar la extracción de características con el algoritmo PHOW. Todo el pre-procesado se le añadió como opción a la función *computeFeatures.m* mediante una función llamada *colorSegmentation* ya que la biblioteca de funciones VLFeat [24] no incluía la posibilidad de hacer este tipo de pre-procesado. El pre-procesado constituye un paso importante para incrementar el rendimiento de la clasificación. Se utiliza en las imágenes de entrenamiento del clasificador, en las de prueba y en la construcción de algunos vocabularios. El pseudo-código de esta implementación aparece en el algoritmo 1.

**Algoritmo 1** Pre-procesamiento

**Entrada:** imagen a color $I(x, y)$,CoordCent (coordenadas centrales de la esfera en el espacio RGB), $R0$(radio de la esfera); $areaMin$ (área mínima de un arma).

**Salida:** iMask(máscara binaria)

1: Inicilizar $maskColor(x, y)$ a cero.
2: **for** cada píxel $I(x, y)$ **do**
3:  **if** $dist <= R0$ **then**
4:  $maskColor(x, y) = 1$
5:  **end if**
6: **end for**
7: $iMaskClose(x, y) \leftarrow$ Cierre de iMaskColor con un disco de radio 6
8: $Rconn1 : Nconn \leftarrow$ Obtener regiones conexas en $iMaskClose(x, y)$
9: **for** cada región conexa $Rconn(i)$ **do**
10:  **if** $areaRegionActual(i) > areaMin$ **then**
11:  $iMaskArea(x, y) \leftarrow$ Copiar Píxeles de $Rconn(i)$
12:  **end if**
13: **end for**
14: $iMask(x, y) \leftarrow$ Dilatación de $iMaskArea$ con un disco de diámetro 2

Se presume que esta etapa de pre-procesado también aumentaría la eficiencia de un algoritmo de detección objetos, como es el caso de la ventana deslizante de Viola en

2001 [9]. Es decir se analizarían solo las regiones que contienen objetos metálicos en vez de recorrer con la ventana deslizante toda la imagen. La ventana deslizante solo se aplicaría en las regiones metálicas con un área de gran tamaño (determinada a priori).

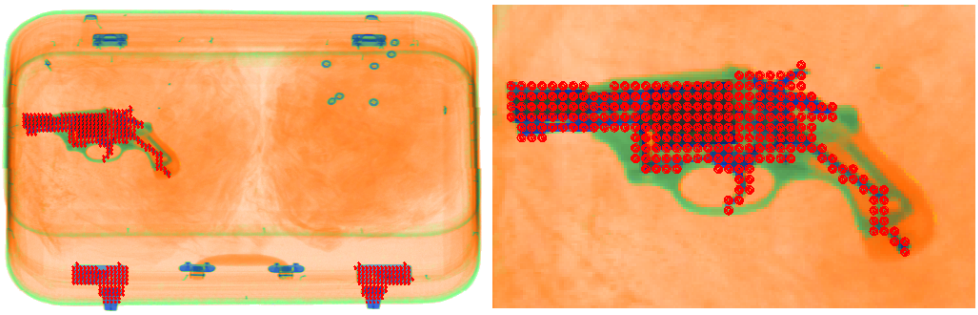

Figura 2.5: *Características extraídas después del pre-procesado*

En la bibliografía consultada se hace referencia a una segmentación de primer plano y a la operación morfológica de dilatación antes de extraer las características pero no se menciona la utilización de otras técnicas como las utilizadas en este trabajo. No hay mención de haberse utilizado un filtrado por áreas. Aunque no hay mucha diferencia en realizar una segmentación de primer plano frente a las técnicas utilizadas aquí.

## 2.4. Construcción de vocabularios

El algoritmo utilizado para el agrupamiento de características fue el clásico k-means que tiene mejores prestaciones frente a otros según Bastan [3] y se ha convertido en el algoritmo más usado en la construcción del vocabulario. La función *vl_kmeans* implementa eficientemente el algoritmo k-means. Por defecto usa el algoritmo de Lloyd que es un método de optimización que alterna entre la búsqueda del mejor promedio dada la asignación y busca la mejor asignación dado el promedio. También k-means soporta dos algoritmos adicionales. La variante de Elkan [40] que es igual al anterior solo que utiliza una técnica de aceleración mediante la desigualdad triangular que evita muchas comparaciones especialmente en las iteraciones finales. Usualmente se toma de 4 a 5 veces menos tiempo que la variante de Lloyd. También se encuentra la variante ANN (*Approximate Nearest Neighbor*) que es más eficiente que Elkan en problemas donde el número de muestras

supere al millón [25].

En este trabajo se construyeron diversos vocabularios de diferentes tamaños como 1000, 3000 y 5000. Se construyeron dos vocabularios por cada tamaño diferenciándose en las regiones donde se extrajeron las características antes de aplicar el k-means. Estos dos vocabularios son el vocabulario universal y el vocabulario metálico. El vocabulario universal se construyó con las características extraídas de todas las imágenes de la base de datos mediante el algoritmo PHOW, tomando los puntos sobre todas las regiones. Para el vocabulario metálico se decidió extraer las características de todas las imágenes en las regiones de interés que quedaron después de realizar el pre-procesado (explicado en la sección 2.3). En la figura 2.6 se aprecian estas diferencias entre ambos vocabularios. Esto se hizo debido a que se espera una mejor representación de las imágenes con un vocabulario construido sobre las regiones metálicas de los objetos de interés según se infiere en Perronin [28]. Se utilizó el algoritmo Elkan dentro de k-means para acelerar la construcción del vocabulario y se tomaron todas las 948 imágenes presentes en la base de datos.

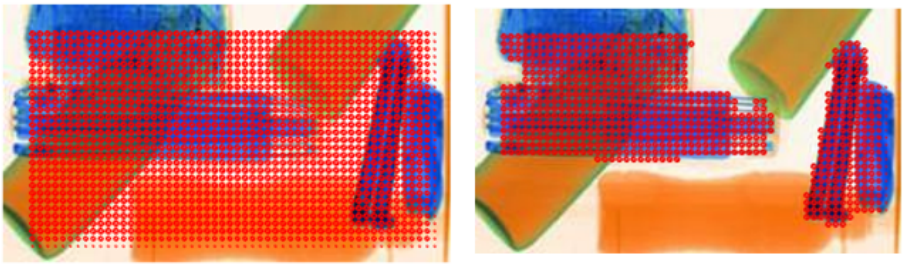

Figura 2.6: *a) Características para el vocabulario universal, b) características para el vocabulario metálico.*

Para la construcción del vocabulario se desarrolló un script en matlab *ConstructVocabulary.m*, que a su vez llama a la función *computeVocabularyFromImageList.m*. A esta función se le pasa como parámetro el conjunto de imágenes y el tamaño del vocabulario, que se le incorporó a la función original de [37]. La función *computeVocabularyFromImageList* realiza el agrupamiento utilizando *vl-kmeans* y, también utiliza la función *vl-kdtreebuild* que es la encargada de implementar kdtree. Un kdtree o árbol k-d es una estructura de datos utilizada para acelerar el proceso de k-means.

## 2.5. Cálculo de los histogramas de palabras visuales

Para el cálculo de los histogramas de palabras visuales se prepararon dos conjuntos de imágenes de la base de datos. Uno que contenía todas las armas después de haberlas recortado de las imágenes originales y otro conjunto que no contiene armas, obtenido después de haber recortado aleatoriamente regiones donde no hay armas.

Estos dos conjuntos constituyen las imágenes positivas y las negativas respectivamente. Las imágenes que no contienen armas en esta selección tienen objetos metálicos ya que son los objetos de interés donde es más probable encontrar un arma y no son eliminadas por el procedimiento de reducción de área de búsqueda anteriormente descrito. A todas estas imágenes se le extraen las características mediante PHOW después de haber pasado la etapa de pre-procesado. Con estas características se construyen los vectores de rasgos es decir los histogramas de palabras visuales por cada imagen.

La función *computeHistogramsFromImageList.m* calcula los histogramas de un conjunto de imágenes dado un vocabulario visual que se le pasa por parámetro. A esta función se le incorporó un parámetro para indicar que se utilizará la extracción de características después de hacer el pre-procesado. La función a su vez ejecuta reiteradamente la función *computeHistogram* de VLFeat [24] que es la encargada de construir estos histogramas. La figura 2.7 muestra los histogramas de palabras visuales de un objeto arma y otro no arma construidos con esta función.

La función *computeHistogram* fue mejorada en este trabajo dando la posibilidad de construir un histograma con la técnica de pesado suave, ya que esta implementación no aparece en la biblioteca de funciones VLFeat [24]. A la función *computeHistogram* se le dio la posibilidad de recibir por parámetro el vector que contiene las distancias de los descriptores a las 3 palabras visuales más cercanas para junto con la expresión (1.3) se pueda construir un histograma con la técnica de pesado suave.

Los histogramas se construyeron usando asignación dura además de usar el factor de normalización, es decir, como aparece en la sección 1.7.3, se está usando tf con normalización que aparece en la tabla 1.1 Aunque se realizaron algunos experimentos utilizando pesado suave en la construcción de los histogramas pero no hubo diferencia significativa.

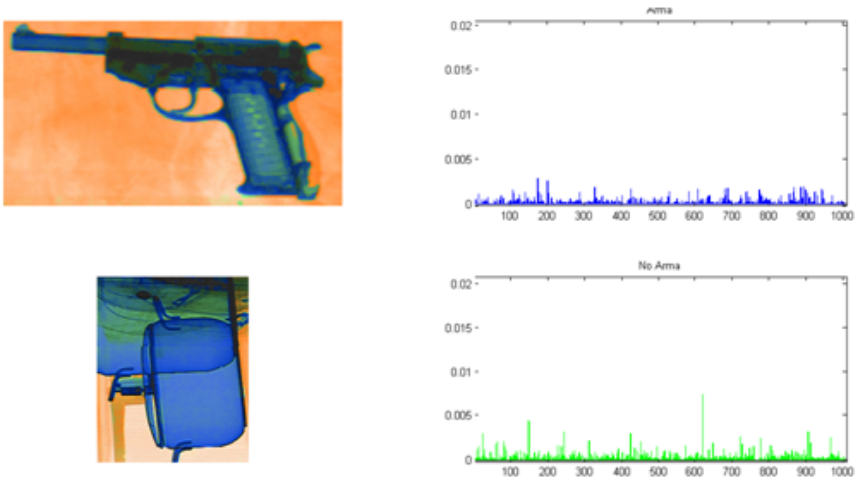


Figura 2.7: *Histogramas de palabras visuales de un objeto arma y otro no arma*

Se cree que esto se debe principalmente a la falta de claridad en el parámetro $\sigma$ visto en la expresión (1.3) en el contexto de imágenes de rayos X y al tamaño del vocabulario adecuado para que el pesado suave sea satisfactorio en este contexto. Por lo que este aspecto necesita mayor investigación.

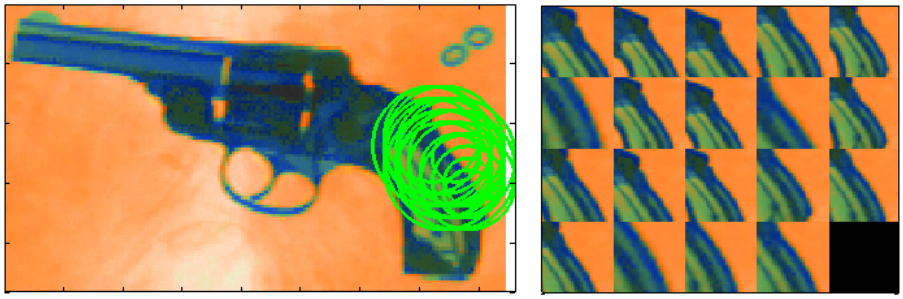

Figura 2.8: *Solapamiento de caracteristicas visuales*

Después de haber calculados los histogramas con la función *computeHistogram* se utilizó la función *removeSpatialInformation* para remover la información espacial ya que se desea flexibilidad frente a la posición de las armas en las imágenes. Esto se realiza debido a que la función *computeHistogram* por defecto incluye información espacial. De todas formas implícitamente hay información espacial porque la densa cuadrícula de puntos de PHOW permite que se solapen las características visuales (visto en la sección 2.2). Aunque también no deja de ser un área que necesite mayor investigación en este contexto.

En la figura 2.8 se demuestra el solapamiento implícito de características visuales que pertenecen a una misma palabra visual.

## 2.6. Entrenamiento de la SVM

Para el entrenamiento del clasificador se hizo necesaria la utilización de la función *vl_svmtrain* que brinda VLFeat [24]. La forma básica para operar con esta función es $[WB] = vl\_svmtrain(X, Y, LAMBDA)$ donde $X$ son los vectores de datos para entrenamiento es decir el conjunto de histogramas de las imágenes de entrenamiento. $Y$ es el vector de las etiquetas que toma valores binarios (+1 o -1) para cada histograma de palabras visuales indicando si pertenece a la clase de armas cortas o no. También se encuentra $LAMBDA$ que es el parámetro de regularización del algoritmo. Se realizaron algunas pruebas para encontrarel mejor valor para $LAMBDA$, que resultó ser 0.0001. La función devuelve el vector de pesos $W$ y el bias $B$ que determinan la frontera de decisión, tal que el valor de $W'X(:, i) + B$ tiene el mismo signo que $Y(i)$ para todo $i$. De manera que si se desea saber si un histograma $H$ de una imagen (que no pertenezca al conjunto de imágenes de entrenamiento) pertenece o no a la clase arma corta, solo bastaría determinar el signo de $W'H + B$. Si es mayor que 0 entonces pertenece a la clase y si no viceversa.

### 2.6.1. Funciones de pérdida

Entre los parámetros que se pueden modificar en el entrenamiento de la SVM se encuentra la función de pérdida introducida en la sección 1.8.3. Existe una gran variedad de funciones de pérdidas que son comúnmente utilizadas en las SVM. Las funciones de pérdida difieren en:

- Su propósito (algunas son más apropiadas para clasificación otras para regresión)
- Su suavidad (la cual afecta cuán rápido la función objetivo puede ser minimizada)
- Su interpretación estadística

Las variantes de función de pérdida que proporciona la función *vl_svmtrain* para el entrenamiento de los datos aparecen en la tabla 2.1.

Tabla 2.1: *Diferentes funciones de pérdida que se utilizan en SVM*

| **Nombre** | **Pérdida** $\ell_i(\mathbf{z})$ | **Descripción** |
|---|---|---|
| Hinge | $max\{0, 1 - y_i\mathbf{z}\}$ | Función de pérdida estándar de SVM |
| Square hinge | $max\{0, 1 - y_i\mathbf{z}\}^2$ | Esta versión es más suave y puede arrojar el problema numéricamente más fácil |
| Square o L2 | $(y_i - \mathbf{z})^2$ | Esta produce el modelo de regresión cadena |
| Linear o L1 | $\lvert y_i - \mathbf{z}\rvert$ | Otra función de pérdida buena para regresión, usualmente más robusta pero más difícil de optimizar que la anterior. |
| Insensitive | $max\{0, \lvert y_i - \mathbf{z}\rvert - \epsilon\}$ | Esta es una variante de la anterior, propuesta en la formulación original de Regresión de Vectores de Soporte. Puede arrojar a una escasa selección de vectores de soporte. |
| Logistic | $\log(1 + e^{-y_i\mathbf{z}})$ | Corresponde a la regresión logística regularizada. |

En el proceso de aprendizaje de la SVM se trata de encontrar $w$ para que la función de costo $E(w)$ sea mínima. Existe una docena de métodos para hacer esto. VLFeat implementa dos métodos diseñados para trabajar con SVM lineales: *Stochastic Gradient Descent* (SGD) Shalev 2011 [41] y *Stochastic Dual Coordinate Ascent* (SDCA) Shalev 2013 [42]. Estos métodos mediante un proceso de optimización resuelven encontrar el mínimo de la función de costo. El método que se escogió, que es el que aparece por defecto, es el SDCA ya que es más eficiente [42] y de reciente aparición.

### 2.6.2. Mapas de kernels homogéneos

Resulta que en la mayoría de los casos se hace necesaria la utilización de un kernel. Porque las clases no son linealmente separables y hay que expandir los datos a una dimensión superior a través de un kernel para poder encontrar una separación de las clases lo más lineal posible.

Mientras los kernels lineales son muy eficientes en el entrenamiento, los kernels no lineales tienden a brindar mejores resultados en la precisión de la clasificación Chatfield [26]. Una clase de kernels que son casi tan eficientes como los lineales pero usualmente con mayor precisión son los kernels homogéneos aditivos Vedaldi 2012 [43].

En las últimas versiones de la biblioteca de funciones VLFeat [24] se incorporaron algunos de estos kernels que pueden utilizarse con la función *vl_svmtrain*. Los kernels homogéneos que tiene VLFeat son el Intersección, el $\chi^2$ y el Jensen-Shannon.

Los mapas de kernels homogéneos son una aproximación lineal de dimensión finita de los kernels homogéneos de Intersección el $\chi^2$ y el Jensen-Shannon. Estos kernels son frecuentemente usados en aplicaciones de visión por computadora porque son particularmente apropiados a los datos en el formato de histogramas, lo que cual es conveniente ya que se está trabajando con histogramas de palabras visuales.

Sea $x, y \in \mathbf{R}_+$ ser números no negativos escalares y $k(x, y) \in \mathbf{R}$ ser un kernel homogéneo. Para datos vectoriales $\mathbf{x}, \mathbf{y} \in {\mathbf{R}_+}^d$ los kernels homogéneos son definidos como una combinación aditiva de kernels escalares $K(\mathbf{x}, \mathbf{y}) = \Sigma_{i=1}^{d} k(x_i, y_i)$.

El mapa de kernels homogéneos de orden $n$ es una función vectorial $\mathbf{\Psi}(x) \in \mathbf{R}^{2n+1}$ de manera que para cualquier $x, y \in R_+$ se cumple la siguiente aproximación:

$$k(x, y) \approx \langle \mathbf{\Psi}(x), \mathbf{\Psi}(y) \rangle \tag{2.1}$$

Usar herramientas de análisis lineal como las máquinas de soporte vectorial con los datos que han sido codificados por mapa de kernels homogéneos es entonces aproximadamente equivalente a usar un método basado en el correspondiente kernel no lineal.

**Grado de homogeneidad**

Cualquier kernel homogéneo $k_1(x, y)$ puede ser extendido a lo que se conoce como $\gamma$-homogéneo kernel $k_\gamma(x, y)$ por la definición:

$$k_\gamma(x, y) = (xy)^{\frac{\gamma}{2}} \frac{k_1(x, y)}{\sqrt{xy}} \tag{2.2}$$

De manera que variando este valor regula el grado de homogeneidad, cuando $\gamma = 1$ el

kernel es homogéneo puro. Para valores pequeños de $\gamma$ aumenta la no linealidad del kernel esto es beneficioso en algunas aplicaciones [43].

Para utilizar estos kernels se construye una estructura de datos mediante la función *vl_svmdataset*. La función recibe por parámetro la estructura *hom* donde se indica el tipo de kernel, el orden de la expansión de los datos a una dimensión mayor, el tipo de ventana y el valor de $\gamma$

El orden de la expansión se especifica en $hom.order = N$ de manera que cada vector de los datos de entrenamiento es expandido a un vector de dimensión $2N+1$. En las pruebas que se realizaron se escogió $N = 2$

El tipo de ventana se especifica en *hom.window* que puede adoptar un valor de dos posibles: $RECTANGULAR$ o $UNIFORM$. (Para mayor detalles sobre la ventana revisar Vedaldi [43]) El valor de $\gamma$ se especifica en *hom.gamma* donde se selecciona el grado de homogeneidad del kernel. El valor estándar es 1. Se hicieron algunos experimentos variando el grado de homogeneidad y la ventana pero no se presentan debido a que no se obtuvieron variaciones significativas.

También se utiliza la función *vl_homkermap* en la expansión de los datos de prueba antes de realizar el producto escalar con $W$ para determinar la puntuación (*scores*) de cada imagen de prueba.

Además de utilizar la implementación de los tres kernels homogéneos que trae la biblioteca de funciones VLFeat [24], se implementó el kernel de Hellinger[6]. Este es el kernel homogéneo más simple [43]. Es un kernel de fácil implementación, se interpreta que:

$$K(x, y) = \sqrt{xy} = \langle \Psi(x), \Psi(y) \rangle = \sqrt{x}\sqrt{y} \tag{2.3}$$

De manera que solo bastaría realizar el entrenamiento de la SVM con la raíz cuadrada de los histogramas de entrenamiento y posteriormente evaluar con la raíz cuadrada de los histogramas de prueba.

[6]es conocido también como el coeficiente de Bhattacharyya [43]

## 2.7. Implementación de la validación cruzada

Después de haber calculado los histogramas de palabras visuales de las imágenes positivas y negativas de la base de datos se procede a realizar la evaluación del método de clasificación usando la técnica de validación cruzada. La biblioteca de funciones VLFeat [24] no tiene implementada la validación cruzada de manera que se ha hecho necesario implementarla en MATLAB.

Se eligió la validación cruzada de $k$ segmentos. A medida que se aumenta la cantidad de segmentos mayor es la confiabilidad, pero según el trabajo de Kohavi 1995 [35] no hay muchos cambios en la varianza de los resultados de una validación cruzada cuando se varía el número de segmentos $k$.

Lo primero que se realizó fue la construcción de un vector que contiene los índices de los histogramas y realizar una permutación aleatoria de estos. Se construyó un vector para cada conjunto positivo y negativo y se guardaron en un fichero *.mat* con el nombre de $PermHist$. Posteriormente estos vectores son divididos en $k$ partes iguales y en un ciclo de $k$ iteraciones se seleccionaron los histogramas de entrenamiento y prueba en cada caso. Esto se realizó mediante una selección de los índices de los histogramas de prueba (de cada uno de los $k$ intervalos) y los restantes índices corresponden a los histogramas de entrenamiento.

En cada una de las $k$ iteraciones se realiza el entrenamiento del algoritmo clasificador con el subconjunto seleccionado de histogramas y a su vez se realiza la evaluación con el subconjunto de histogramas de prueba. Este proceso de evaluación se repite para cada uno de los kernels evaluados.

## 2.8. Promediado de curvas ROC

Para realizar el promediado de las curvas ROC se utilizó la función *perfcurve* disponible en MATLAB 2014a. Esta función calcula los intervalos de confianza usando tanto el promediado vertical como el promediado por umbral. En este trabajo se utilizó el promediado por umbral. La sintaxis de la llamada a la función es:

*[fpr,tpr,T,AUC]=perfcurve(labels,scores,posclass);*

Para retornar los intervalos de confianza las variables de entrada *labels* y *scores* deben ser arreglos de celdas (*cell array*). De esta manera se realiza el promediado por umbral. La variable de entrada *posclass* debe ser del mismo tipo que *labels* indicando cuál es la clase positiva. Las variables $fpr$ y $tpr$ son una matrix de $m$ por 3 donde la primera columna son los valores promediados y la segunda y tercera son el valor inferior y superior del intervalo de confianza. Lo mismo ocurre con $AUC$ que es el área bajo la curva. La variable $T$ es el vector columna de los umbrales. Esta función se ejecutó para cada uno de los cuatro kernels y se graficaron las cuatro curvas ROC para determinar cuál de ellas brinda un mejor resultado.

Se creó una estructura llamada *folds* donde se almacenaron los resultados de cada iteración y los resultados del promediado de cada curva ROC. Incluye también los parámetros utilizados en la etapa del entrenamiento de la SVM, como son: el parámetro de regularización *lambda*, el orden de la expansión de los mapas de kernel homogéneos, el grado de homogeneidad $\gamma$ y la función de pérdida. El pseudo-código de la validación cruzada con curvas ROC aparece en el algoritmo 2.

**Algoritmo 2** Validación Cruzada con Curvas ROC

**Entrada:** $kfolds$ (número de particiones del conjunto de datos), $hist$ (histogramas), $indPos, indNeg$ (listado de los índices de histogramas positivos y negativos), $numPos, numNeg$ (cantidad de positivos y negativos)

**Salida:** tpr,fpr

1: $\Delta_{Pos} = \frac{numPos}{kfolds}$;

2: $\Delta_{Neg} = \frac{numNeg}{kfolds}$;

3: $intervalsPos \leftarrow$ Dividir indPos en $kfolds$ partes $(\{1, \ldots, \Delta_{Pos}\}, \{\Delta_{Pos} + 1, \ldots, 2\Delta_{Pos}\}, \ldots, \{kfols\Delta_{Pos} + 1, \ldots, numPos\})$

4: $intervalsNeg \leftarrow$ Dividir indNeg en $kfolds$ partes $(\{1, \ldots, \Delta_{Neg}\}, \{\Delta_{Neg} + 1, \ldots, 2\Delta_{Neg}\}, \ldots, \{kfols\Delta_{Neg} + 1, \ldots, numNeg\})$

5: $permHistPos \leftarrow$ Realizar permutación aleatorea de elementos positivos

6: $permHistPos \leftarrow$ Realizar permutación aleatorea de elementos negativos

7: **for** cada iteración $i$, $i < kfolds$ **do**

8: $testIdxPos \leftarrow permHistPos(intervalsPos(i))$, conjunto de positivos para prueba

9: $testIdxNeg \leftarrow permHistNeg(intervalsNeg(i))$, conjunto de negativos para prueba

10: $trainIdxPos \leftarrow setdiff(permHistPos, testIdxPos)$, conjunto de positivos restantes para entrenamiento

11: $trainIdxNeg \leftarrow setdiff(permHistNeg, testIdxNeg)$, conjunto de positivos restantes para entrenamiento

12: **for** cada kernel $kern = 1$, $kern <$ cantidad de kernels **do**

13: $[w, bias] \leftarrow$ Entrenar Clasificador con histogramas de entrenamiento $(hist([trainIdxPostrainIdxNeg]))$

14: $testScores\{kern, iter\} \leftarrow$ Evaluar Clasificador con histogramas de Prueba $(hist(ttestIdxPos)$

15: **end for**

16: **end for**

17: **for** cada kernel $kern = 1$, $kern <$ cantidad de kernels **do**

18: $[tpr, fpr] \leftarrow$ Calcular y promediar las curvas ROC usando $testScores\{kern, :\}$

19: **end for**

# Capítulo 3

# Resultados Experimentales

## 3.1. Introducción

En este capítulo se presentan los resultados obtenidos de diversos experimentos realizados para buscar los mejores valores de los parámetros de entrenamiento para las exigencias de la aplicación. Lo primero que se presenta es una descripción de la base de datos, la cantidad de imágenes y sus peculiaridades. Después se presentan los resultados obtenidos de la construcción de diversos clasificadores basados en SVM. Estos resultados son de experimentos que se realizaron con y sin pre-procesado, con los cuatro kernels[1] y con diferentes funciones de pérdidas. Se hizo al final una prueba para seleccionar el vocabulario con mejor comportamiento. Finalmente se hace un análisis de los resultados seleccionados y se comparan con otros estudios referenciados.

## 3.2. Descripción de la base de datos

La base de datos fue confeccionada por un conjunto de imágenes de equipos de inspección de rayos X de energía dual entregadas por el CENPIS. La base de datos tiene un total de 948 ficheros de imágenes. Se procesaron en MATLAB 2014a en el formato de fichero de imagen PNG con resoluciones que varían alrededor de 1000x600. Muchas de estas imágenes son las que se utilizan en el entrenamiento visual del personal de inspección esto

[1]Los tres kernels homogéneos que están implementados en VLFeat y la implementación del Hellinger

tiene gran importancia porque le confiere cierta validez, confiabilidad y generalización a la base de datos.

Para la validación cruzada se prepararon un conjunto de 312 imágenes positivas y 567 imágenes negativas. Las imágenes positivas son aquellas que pertenecen a la clase arma corta que es la unión de las clases revolvers y pistolas, estas contienen solamente la región de la imagen donde está contenida el arma. Las imágenes negativas fueron generadas mediante una selección aleatoria de regiones[2] en las imágenes que contenían otros tipos de objetos. Posteriormente se seleccionaron de ese conjunto los objetos y/o parte de ellos que contenían regiones metálicas ya que la extracción de características se realiza sobre estas regiones de interés debido al pre-procesado. En la figura 3.1 se observa una muestra de estas imágenes con objetos metálicos.

Figura 3.1: *Muestra de las imágenes negativas preparadas*

Se puede apreciar en la figura 3.2 una muestra de las diversas situaciones en que se encuentran las armas en el conjunto de imágenes positivas utilizadas. Estas imágenes de armas poseen gran diversidad en cuanto a que presentan: oclusión propia, cambios de posición, pistolas y revolvers de diferentes tipos, algunas con leves transformaciones geométricas que estuvieron presente en el momento de adquisición, otras parcialmente desarmadas, con diversos grados de solapamiento con otros objetos metálicos, algunas con partes no metálicas y una combinación de estas características. Esto se evidencia en la distribución que aparece en la tabla 3.1 de las armas presentes en la base de datos

[2]Como si fuese una ventana deslizante

para realizar los experimentos. En la figura 3.2 aparecen etiquetadas cada una de estas situaciones descritas en la tabla 3.1

Tabla 3.1: *Distribución de las armas*

| | **Situación de las armas** | **Cantidad de imágenes** |
|---|---|---|
| **1** | Oclusión propia | 123 |
| **2** | Solapadas con objetos metálicos | 94 |
| **3** | Partes no metálicas | 30 |
| **4** | Error en el momento de la adquisición | 15 |
| **5** | Parcialmente desarmadas | 12 |
| **6** | Vista frontal simple | 92 |

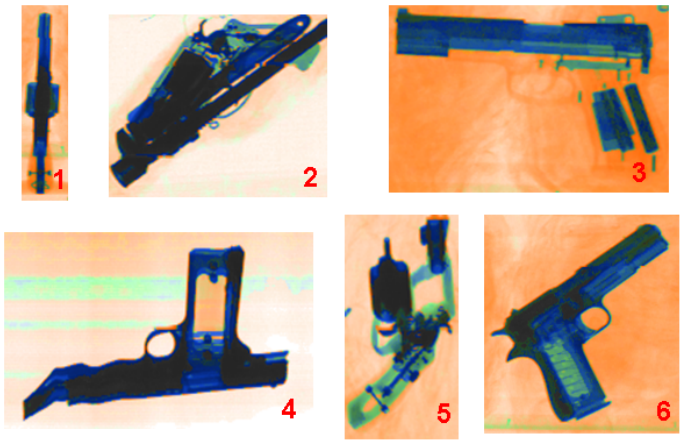


Figura 3.2: *Situaciones de las imágenes positivas utilizadas*

## 3.3. Evaluación del clasificador

Después del aprendizaje del clasificador con el conjunto de imágenes de entrenamiento se procede a realizar la etapa de prueba. Es decir el clasificador recibe un conjunto de imágenes de prueba (pertenecientes a ambas clases) diferentes a las imágenes de entrenamiento y debe ser capaz de discernir las imágenes que pertenecen a una clase u otra. De manera que la evaluación consiste en medir cuán efectivo fue este proceso de distinción

entre las clases durante la etapa de prueba del clasificador. Para este proceso de evaluación se utilizó validación cruzada de 3 segmentos es decir el proceso de entrenamiento y prueba se realiza tres veces sobre la base de datos y el resultado final que se presenta es el promedio de las curvas ROC.

### 3.3.1. Evaluación del pre-procesado y de los kernels

Lo primero que se comprobó fue el efecto positivo del pre-procesado que lo resume la tabla 3.2. Para esta prueba se utilizó el vocabulario universal de 1000 centroides. Como era de esperar el efecto del pre-procesado sobre todos los kernels es positivo puesto que para cada uno de ellos mejora la respuesta general de la curva AUC y disminuye EER. De manera que siempre se va a utilizar la etapa de pre-procesado para el resto de los experimentos.

Tabla 3.2: *Efecto del pre-procesado*

| **Pre-procesado** | $\chi^2$ | **Intersección** | **Jensen-Shannon** | **Hellinger** |
|---|---|---|---|---|
| **No** | AUC: 97.12 % | AUC: 95.82 % | AUC: 95.37 % | AUC: 96.12 % |
| | EER: 9.38 % | EER: 9.62 % | EER: 9.38 % | EER: 9.43 % |
| **Si** | AUC: 98.11 % | AUC: 97.58 % | AUC: 96.86 % | AUC: 97.98 % |
| | EER: 6.45 % | EER: 7.83 % | EER: 8.29 % | EER: 7.07 % |

En la figura 3.3 se observan las curvas ROC de la tabla después de haber realizado el pre-procesamiento. Se destaca que los kernels que mejor comportamiento presentan en la clasificación son el $\chi^2$ y el Hellinger. Siendo superior el $\chi^2$. Ya en algunas investigaciones se había reportado la superioridad de este kernel pero en bases de datos de imágenes de espectro visible [26]. También en la tabla el kernel $\chi^2$ es superior en todos los casos.

### 3.3.2. Evaluación de las diferentes funciones de pérdidas

Teniendo seleccionado el kernel de mejor comportamiento se procede a la selección de la función de pérdida (introducida en la sección 1.8.3). En las referencias consultadas no

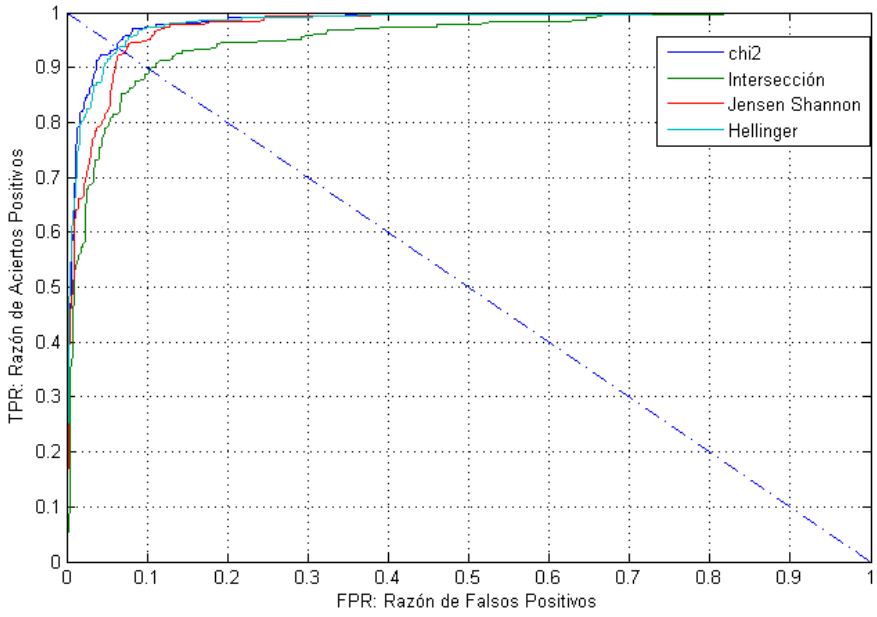


Figura 3.3: *Curvas ROC de diferentes kernels*

se menciona la selección de este parámetro. En la figura 3.4 aparecen las curvas ROC utilizando el kernel $\chi^2$ donde se ha variado en cada una la función de pérdida. Se puede observar en la leyenda que las funciones de pérdida con mayor AUC son L1 y L2. Pero para las exigencias de la aplicación hay que priorizar la TPR sobre FPR es decir la elección del punto de operación debe estar por encima de la diagonal principal del espacio ROC.

Aunque la diferencia en cuanto a AUC no es significativa, se decidió seleccionar como función de pérdida a L2 debido a que es menos costosa computacionalmente que L1 [24]. Se seleccionó en la figura 3.4 un punto de operación con una TPR=96.47 % y una FPR=6.35 % a modo de ejemplo.

### 3.3.3. Evaluación de los vocabularios

En la sección 2.4 se había comentado que se construyeron vocabularios universales y metálicos de tamaños 1000, 3000 y 5000 constituyendo un total de seis vocabularios de palabras visuales. En la figura 3.5 aparecen las curvas ROC resultantes utilizando cada uno de estos vocabularios.

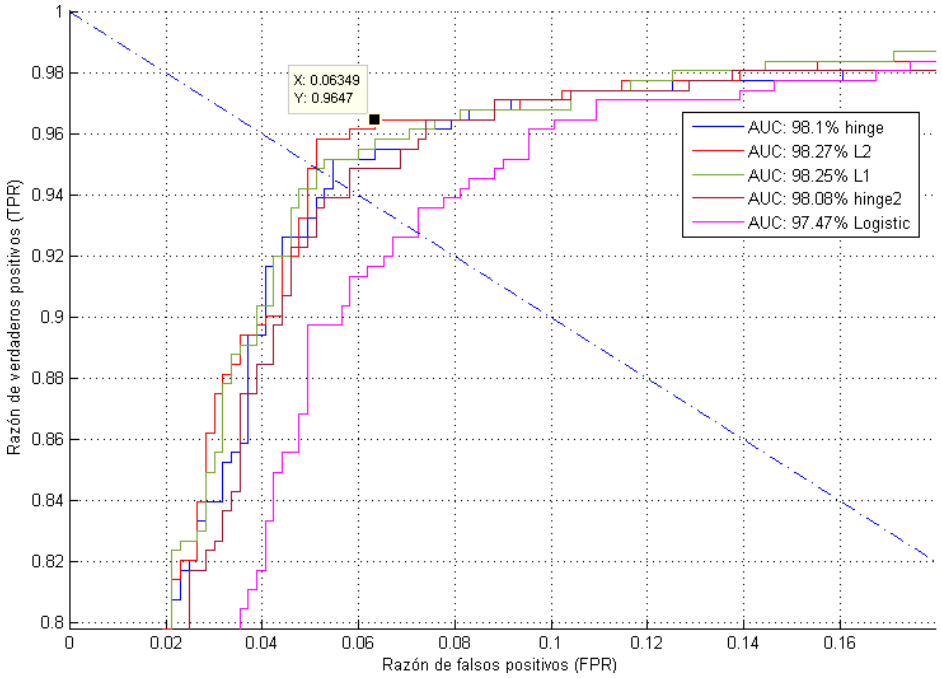


Figura 3.4: *Curvas ROC de diferentes funciones de pérdida*

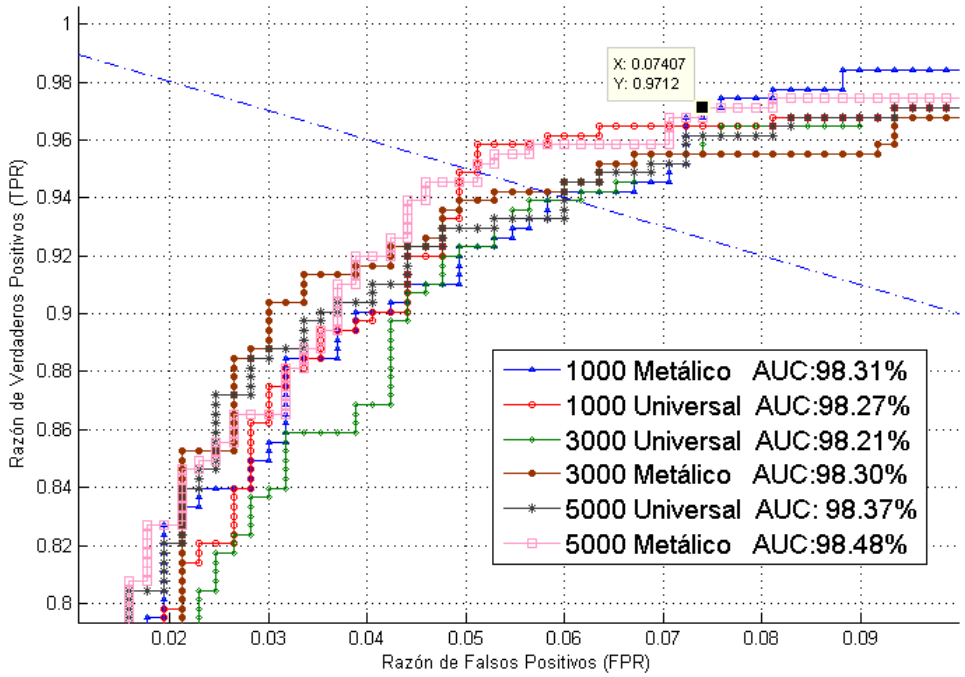


Figura 3.5: *Curvas ROC con diferentes vocabularios*

Se puede observar como el AUC de las curvas pertenecientes a los vocabularios metálicos son superiores que la de los vocabularios universales y a medida que aumenta el tamaño del vocabulario es superior la diferencia en cada par de vocabularios con igual tamaño.
Por encima de la diagonal principal se destacan los vocabularios: 1000 universal, 1000 metálico y 5000 metálico. Se seleccionó el punto de operación con TPR=97.12 % y FPR=7.4 % que aparece señalado en la figura 3.5. Este punto pertenece simultáneamente a las curvas del vocabulario metálico de 1000 y 5000 palabras visuales respectivamente. Pero es preferible utilizar el vocabulario metálico de tamaño 1000 frente al de 5000 ya que los histogramas construidos generarían menos cálculo computacional según mostró Uijlings en 2009 [44], de manera que este es el vocabulario propuesto.
El punto de operación seleccionado con TPR=97.12 % y FPR=7.4 % tiene una precisión de PPV= 87.57 % y una exactitud general de ACC=94.2 %

### 3.3.4. Evaluación del algoritmo frente al reconocimiento de armas solapadas

Como el algoritmo a implementar es para una aplicación de seguridad se hace necesario analizar las características de las armas que fueron mal clasificadas. En la figura 3.6 aparecen armas que tienden a ser mal clasificadas (FN), es decir armas cortas que el algoritmo le cuesta reconocer. Fueron tomadas de cada una de las tres iteraciones de la validación cruzada. Se puede observar que prevalecen armas solapadas con otros objetos metálicos y armas con oclusión propia como era de esperar.

Para poder medir cuán efectivo es el algoritmo propuesto frente a armas con diferentes grados de solapamiento se realizó una prueba usando el método de retención o *holdout* (es decir realizando una sola partición de datos de entrenamiento y prueba). Se utilizaron todas las imágenes de armas sin solapamiento en el conjunto positivo de entrenamiento. Pero las imágenes que poseían armas que se encontraban solapadas con diferentes objetos metálicos a diferentes grados de solapamiento son las que utilizaron en el conjunto positivo de prueba. Este experimento no se realiza para tomar ninguna decisión sino para mostrar el peor comportamiento del algoritmo clasificador para esta situación de armas solapadas.

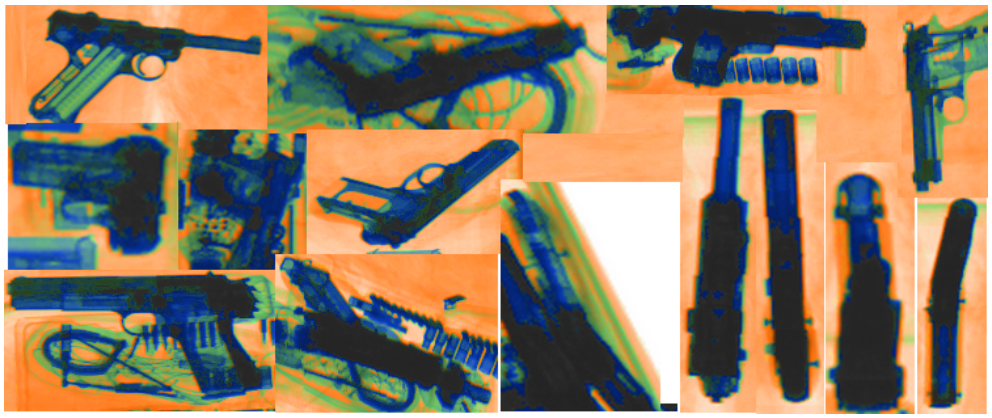

Figura 3.6: *Armas que tienden a ser mal clasificadas*

Tabla 3.3: *Distribución para el experimento con armas solapadas*

| Base de datos | Positivas | Negativas |
|---|---|---|
| **Entrenamiento** | 218 | 378 |
| **Prueba** | 94 | 189 |

La tabla 3.3 muestra la distribución de las imágenes utilizadas en este experimento.

Como era de esperar en la figura 3.7 la respuesta de la curva es pobre en comparación con los resultados presentados con la validación cruzada. Aparece marcado un punto de operación con un TPR=88.83 % y FPR=12.7 %. Este resultado se puede tomar como la medida de la peor precisión que tiene el algoritmo frente al reconocimiento de armas con solapamiento. La implementación reconoce un 88.83 % de las armas solapadas. En realidad la eficiencia sería un poco mayor debido a que en el proceso de entrenamiento se contaría con una cierta cantidad de armas que poseen solapamiento, como se realizó en el experimento anterior con validación cruzada. Este resultado demuestra la eficiencia del algoritmo en el reconocimiento de armas solapadas.

El decremento en el rendimiento del resultado anterior muestra que la tarea más difícil es el reconocimiento de armas solapadas con otros objetos. De manera que constituye un área que necesita mayor investigación en el desarrollo de un sistema de visión por computador para imágenes de rayos X. Es muy probable que las armas solapadas sean los

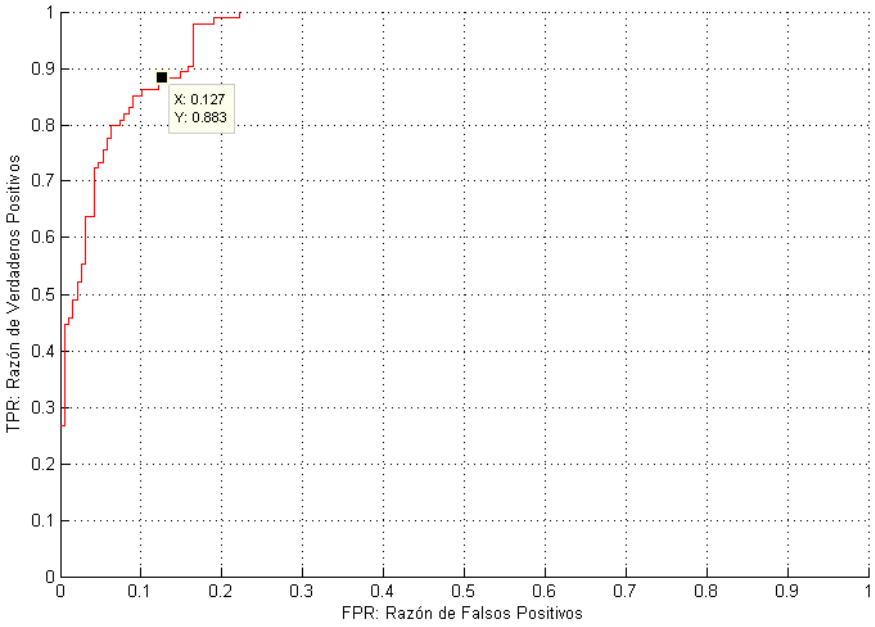


Figura 3.7: *Curva ROC con armas solapadas*

vectores de soporte porque son las más difíciles de reconocer y por ende son las que van a contribuir mayormente a definir los parámetros del hiperplano de separación. Razón por la que deben ser incluidas en el proceso de entrenamiento.

En cuanto a las armas con oclusión propia se sugieren dos soluciones. La primera es la utilización de un equipo de rayos X que retorne múltiples vistas, pero esto depende más del hardware de la tecnología. La segunda idea pudiera ser entrenar un clasificador utilizando únicamente las armas con oclusión propia y combinando este con el propuesto se podría distinguirlas con mayor facilidad.

## 3.4. Análisis y discusión de los resultados

Después de obtener esta serie de resultados y de hacer la mejor selección hay que analizar su significado y cuánto puede aportar a la implementación de esta aplicación. Para hacer un mejor análisis se siguen los parámetros generales del método BoVW que utiliza Turcsany [4] para realizar una comparación con otros estudios. Estos estudios utilizan el mismo método general de Saco de Palabras Visuales BoVW para reconocer armas

de fuego (armas cortas) en imágenes de rayos X y se utilizan las mismas métricas para evaluar. No hay que perder de vista que se utilizan diferentes bases de datos de manera que una comparación no puede ser estricta mirando únicamente los resultados obtenidos. El objetivo de esta comparación está enfocado en realizar un análisis sobre los parámetros del algoritmo. La tabla 3.4 muestra los resultados de tres estudios y los principales parámetros del método utilizado en cada uno. Aunque también se realiza una comparación con los resultados del reciente trabajo de Bastan [6].
Las principales diferencias en los tres estudios se concentran en: la extracción de características, tanto en la detección de los puntos como en el descriptor, el tamaño y el tipo de vocabulario construido, el tipo de kernel junto con otros parámetros que se pueden modificar en el entrenamiento de la SVM y en la experimentación con la base de datos.

A primera instancia se puede apreciar que los resultados que presenta este estudio se encuentran relativamente cercanos a los presentados por Turcsany [4]. Para realizar una comparación hay que considerar una serie de puntos que influyen.
Hay una diferencia en el pre-procesado de este estudio frente a los demás ya que no es una segmentación de primer plano sino que se hace una segmentación por color con operaciones morfológicas. No hay mucha diferencia en la utilización de ambos métodos de pre-procesado, el propuesto tiene la novedad de que descarta ciertas regiones debido al filtrado por áreas aunque esto no demuestra superioridad.

En la extracción de características se utilizó PHOW como detector de puntos (basado en DSIFT) que realiza un muestreo denso de puntos en una región definida a priori. Es bien conocido los beneficios de un muestreo denso de puntos en la clasificación [26] [27], principalmente para este tipo de imágenes de poca textura como sugiere también Bastan 2013 [6]. Los descriptores SIFT utilizados en este estudio tienen la ventaja de tener un tamaño de 384 por extraerse en los tres canales HSV y representan mejor la información que un descriptor SIFT estándar (de tamaño 128) sobre un solo canal.

El vocabulario utilizado en este trabajo es de tamaño 1000 y es metálico. Se demostró que este tipo de vocabulario representa mejor la información que uno universal. Por esta razón y por el tamaño del vocabulario se entiende que el resultado de este trabajo es superior al Bastan [3]. El vocabulario propuesto por Turcsany [4] utiliza otra técnica, es la concatenación de dos vocabularios de 600 palabras construidos con las clases positiva

Tabla 3.4: *Comparación con otros estudios*

| **Parámetro** | **Bastan** [3] | **Turcsany** [4] | **Este estudio** |
|---|---|---|---|
| Pre-procesado | Segmentación de primer plano | Segmentación de primer plano | Segmentación por color +op. morfológicas con +filtrado por área |
| Detección de puntos | DoG + Harris | SURF | PHOW |
| Descriptor | SIFT | SURF | SIFT |
| Generación del vocabulario | k-means tradicional | *Class-specific online k-means* | k-means tradicional sobre características metálicas |
| Tamaño del vocabulario | 200 | 1200 | 1000 Metálico |
| Cuantificación vectorial | SW2 | HA | HA |
| Clasificador | SVM | SVM | SVM |
| Kernel | Intersección | RBF Gausiano | $\chi^2$ |
| Experimentación | Método de retención | Validación cruzada de 3 segmentos | Validación cruzada de 3 segmentos |
| Scanner | Rayos X de energía dual | Rayos X de una sola energía | Rayos X de energía dual |
| Datos | 208 entrenamiento (52 pos; 156 neg) 764 prueba (40 pos; 724 neg) | 850 pos; 10000 neg | 312 pos; 567 neg |
| TPR | 70 % | 99.07 % | 97.12 % |
| FPR | | 4.31 % | 7.4 % |

y negativa respectivamente.

Bastan [3] utiliza la técnica de asociación de pesos de histograma conocida como pesado suave SW2 (comentado en la sección 1.7.3) que se ha probado que es superior al pesado duro en determinado contexto [31]. Pero es probable que no haya sido eficiente en el contexto que la aplica debido principalmente al tamaño de su vocabulario que es de 200 palabras. Con un vocabulario pequeño la técnica del pesado suave no es muy eficiente [31]. No obstante esta técnica requiere mayor investigación en el área de imágenes de rayos X.

Hay que tener en cuenta también que las imágenes que utiliza Turcsany [4] son en escala de grises . La utilización de un kernel homogéneo en la implementación que ofrece VLFeat [24] con los mapas de kernels homogéneos permite que la etapa de clasificación sea más eficiente y tan rápida como si se tratase de un kernel lineal [26]. Se obtuvo como resultado en el presente estudio que el kernel $\chi^2$ es superior al Intersección propuesto por Bastan [3] [6] donde él plantea lo contrario (que el kernel Intersección es superior al $\chi^2$). No se realizó un experimento para compararlo con el RBF Gausiano propuesto por Turcsany [4], pero se cree que el kernel $\chi^2$ es superior a este. Ya se ha reportado su superioridad por Jiang en 2007 [45], Zhang en 2007 [46] y Uijlings en 2009 [44] pero en otros contextos de imágenes.

En las referencias consultadas no hay alusión de haberse realizado un experimento en busca de la mejor función de pérdida. Se pudo comprobar en este estudio cómo este parámetro influye en la calidad de los resultados.
En este trabajo se muestra la dificultad que presenta el reconocimiento de armas solapadas y se da una medida de la peor precisión del algoritmo en este contexto. En las referencias consultadas no hay un análisis del reconocimiento frente a las armas que aparecen solapadas con objetos metálicos. No incluyen pruebas que lo midan directamente. Dado que es más probable encontrarse esta situación, como un intento para esconder armas, tiene gran importancia analizar la eficiencia del algoritmo en esta situación.

El trabajo de Bastan [6] utiliza la métrica AP (*Average Precision*) que es el área bajo la curva *Precision-Recall* [31] para presentar sus resultados. Para poder realizar una comparación con su trabajo se utilizó la misma función *perfcurve*, indicando en sus parámetros de entrada que calcule la curva *Precision-Recall*. Donde se obtuvo como resultado

96.48 % de AP. Este resultado supera al mejor resultado de 94.6 % que presenta Bastan [6] para el caso de clasificación con una única vista en armas cortas. El resultado de Bastan [6] fue alcanzado con otro algoritmo de extracción de características, un vocabulario de 5000 palabras universal sin realizar validación cruzada y con el kernel de Intersección de histogramas.

Otra de las dificultades presentes en esta área de investigación es la inexistencia de una base de datos estándar (ya mencionado por Mery en 2014 [12]) de este tipo de imágenes. Las bases de datos utilizadas en los trabajos comparados son diferentes, principalmente en cuanto a cantidad de imágenes y puntos de vista. De todas formas aunque la base de datos utilizada en este estudio no posee gran cantidad de imágenes en comparación con algunos trabajos (como el de Turcsany [4] y el de Bastan [6]), sí aparecen imágenes representativas de ambas clases. De diferentes puntos de vista, imágenes rotadas, pistolas y revolvers de diferentes tipos, algunas pistolas con leves transformaciones geométricas que estuvieron presentes en el momento de adquisición, algunos revolvers levemente desarmados, armas con oclusión propia y armas solapadas con objetos metálicos.
Aunque el trabajo de Bastan [6] ofrece una descripción de las armas presentes en su base de datos, en los demás estudios no está explícita una descripción de la base de datos como la presentada. Es de gran importancia mostrar esto puesto que no se cuenta con una base de datos estándar de las imágenes de rayos X presentes en los equipos de energía dual.

# Conclusiones

Al término de este trabajo de diploma se llegó a las siguientes conclusiones:

1. Se desarrolló e implementó un algoritmo para el reconocimiento de armas cortas en imágenes de rayos X usando el método Saco de Palabras Visuales con pruebas mediante el método de validación cruzada, alcanzando un resultado general con razón de verdaderos positivos de 97.12 % y razón de falsos positivos de 7.4 % (falsas alarmas). El algoritmo propuesto reduciría en un 26 % las falsas alarmas del método de inspección de equipajes actual con esta tecnología.

2. Se desarrolló e implementó el método Saco de Palabras Visuales para representación de imágenes de rayos X. La mejor representación se obtuvo con un vocabulario de 1000 palabras visuales con filtrado para objetos metálicos.

3. Se entrenó y evaluó un clasificador del tipo Máquina de Soporte Vectorial para el reconocimiento de armas cortas en imágenes de rayos X. Los mejores resultados se obtuvieron con el kernel $\chi^2$ y la función de pérdida L2.

4. Los resultados alcanzados en este trabajo de diploma muestran que es posible implementar un sistema de visión por computadora que reconozca armas de fuego, que facilite el trabajo de los operadores y que el proceso de inspección sea más rápido, preciso en el reconocimiento de objetos peligrosos, económico y seguro.

# Recomendaciones

1. Investigar una alternativa para aumentar la eficiencia en el reconocimiento de armas solapadas con objetos metálicos.

2. Implementar y evaluar el algoritmo propuesto de ventana deslizante para la detección de estos objetos en las imágenes de rayos X.

# Referencias Bibliográficas


[1] D. Mery, "X-ray testing: The state of the art," *The e-Journal of Non-Destructive Testing & Ultrasonics*, vol. 18, no. 9, 2013.

[2] L. P. León, "Descriptores para la caracterizacion cuantitativa de la calidad del sistema de inspeccion aduanal de equipajes por rayos-x: Modelo cx5030t," Ph.D. dissertation, FACULTAD DE INGENIERÍA ELÉCTRICA. Instituto Superior Politécnico "José Antonio Echeverría", 2012.

[3] M. Ba*s*tan, M. R. Yousefi, and T. M. Breuel, "Visual words on baggage x-ray images," in *Computer Analysis of Images and Patterns*. Springer, 2011, pp. 360–368.

[4] D. Turcsany, A. Mouton, and T. P. Breckon, "Improving feature-based object recognition for x-ray baggage security screening using primed visualwords," in *Industrial Technology (ICIT), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1140–1145.

[5] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, no. 1-22, 2004, pp. 1–2.

[6] M. Ba*s*tan, W. Byeon, and T. M. Breuel, "Object recognition in multi-view dual energy x-ray images," in *British Machine Vision Conference BMVC*, 2013.

[7] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*. ACM, 2007, pp. 197–206.

[8] V. N. Vapnik and V. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 1.

[9] M. Jones and P. Viola, "Robust real-time object detection," in *Workshop on Statistical and Computational Theories of Vision*, 2001.

[10] X. Shi, "Improving object classification in x-ray luggage inspection," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2000.

[11] G. Heitz and G. Chechik, "Object separation in x-ray image sets," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2093–2100.

[12] D. Mery, "Computer vision technology for x-ray testing," *Insight-Non-Destructive Testing and Condition Monitoring*, vol. 56, no. 3, pp. 147–155, 2014.

[13] Q. Lu, "The utility of x-ray dual-energy transmission and scatter technologies for illicit material detection," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 1999.

[14] T. Franzel, U. Schmidt, and S. Roth, *Object detection in multi-view X-ray images*. Springer, 2012.

[15] D. Mery and V. Riffo, "Automated object recognition in baggage screening using multiple x-ray views," in *52nd Annual Conference of the British Institute for Non-Destructive Testing, Telford*, 2013.

[16] D. Mery, G. Mondragon, V. Riffo, and I. Zuccar, "Detection of regular objects in baggage using multiple x-ray views," *Insight-Non-Destructive Testing and Condition Monitoring*, vol. 55, no. 1, pp. 16–20, 2013.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[19] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[20] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Computer Visionï¿½ECCV 2002*. Springer, 2002, pp. 128–142.

[21] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[22] B. Grauman, Kristen Leibe, *Visual object recognition*. Morgan and Claypool, 2011.

[23] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1615–1630, 2005.

[24] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[25] B. Vedaldi, A. Fulkerson, "Running k-means," http://www.vlfeat.org/tutorials/K-means-clustering, [ultimo acceso mayo 2014].

[26] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *British Machine Vision Conference*, 2011.

[27] F. J. E. Nowak and B. Triggs, "Sampling strategies for bag-of-features image classification," in *Proceedings of the European Conference on Computer Vision*, 2006, pp. 490–503.

[28] F. Perronnin, C. Dance, G. Csurka, and M. Bressan, "Adapted vocabularies for generic visual categorization," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 464–475.

[29] V. Viitaniemi and J. Laaksonen, "Spatial extensions to bag of visual words," in *Proceedings of the ACM International Conference on Image and Video Retrieval*. ACM, 2009, p. 37.

[30] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 2169–2178.

[31] M. I. J. S. J. Philbin, O. Chum and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Computer Vision and Pattern Recognition (CVPR)*, 2008.

[32] D. Boswell, "Introduction to support vector machines: University of carlifornia," *San Diego*, 2002.

[33] T. Fawcett, "Roc graphs: Notes and practical considerations for researchers," *Machine learning*, vol. 31, pp. 1–38, 2004.

[34] B. Vedaldi, A. Fulkerson, "Plotting ap and roc curves," http://www.vlfeat.org/tutorials/Plotting-AP-and-ROC-curves, [ultimo acceso mayo 2014].

[35] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2, 1995, pp. 1137–1145.

[36] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *Proceedings of the international conference on Multimedia*. ACM, 2010, pp. 1469–1472.

[37] A. Vedaldi and A. Zisserman, "Image classification practical," http://www.di.ens.fr/willow/events/cvml2011/materials/practical-classification/, [ultimo acceso mayo 2014].

[38] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," 2007.

[39] R. Gonzalez and R. Woods, *Digital Image Processing*. Addison-Wesley, 2008.

[40] C. Elkan, "Using the triangle inequality to accelerate k-means," in *ICML*, vol. 3, 2003, pp. 147–153.

[41] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal estimated sub-gradient solver for svm," *Mathematical programming*, vol. 127, no. 1, pp. 3–30, 2011.

[42] S. Shalev-Shwartz and T. Zhang, "Stochastic dual coordinate ascent methods for regularized loss," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 567–599, 2013.

[43] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 3, pp. 480–492, 2012.

[44] J. R. Uijlings, A. W. Smeulders, and R. J. Scha, "Real-time bag of words, approximately," in *Proceedings of the ACM international Conference on Image and Video Retrieval*. ACM, 2009, p. 6.

[45] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM, 2007, pp. 494–501.

[46] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International journal of computer vision*, vol. 73, no. 2, pp. 213–238, 2007.