



Universidad Autónoma del Estado de México

Maestría en Ciencias de la Computación

Desarrollo de un Sistema de Identificación de Objetos en
Video

Tesis que presenta

Laura Yadira Domínguez Jalili

Para obtener el Grado de

Maestro en Ciencias en Computación

Asesor de tesis:

Dr. Jair Cervantes Canales

Texcoco, Estado de México

Enero de 2018



Autonomous University of Mexico State

Computer Science Department

Development of an Object Identification System in Video

Submitted by

Laura Yadira Domínguez Jalili

As a fulfillment of the requirement for the degree of
Master in Computer Science

Thesis Advisor

Dr. Jair Cervantes Canales

Texcoco City, Mexico State

January 2018



DICTÁMEN DE AUTORIZACIÓN Y OBTENCIÓN DE GRADO DE MAESTRÍA

Texcoco, México, a 20 de noviembre de 2017

Título del proyecto:

Desarrollo de un Sistema de Identificación de Objetos en Video

Tesista:

Laura Yadira Domínguez Jalili

Dictamen:

No. de revisión: 3

- Rechazado
- Sujeto a modificaciones
- Aceptado, condicionado
- Aceptado

**POSGRADO
TITULACION
RECIBIDO**
 Por Adriana Arevalo
 Texcoco, Méx., a 24 de 11 del 2017

Observaciones generales:

Acceptado para la impresión

Acceptado para la defensa de grado

Tutor Adjunto	Tutor Académico	Tutor Adjunto
Dr. en C. de la Comp. Farid García Lamont	Dr. en C. Jair Cervantes Canales	Dr. en I. S. José Sergio Ruiz Castilla

Dedicada a las personas que amo y son la motivación de mis éxitos y a todos aquellos que estén interesados en esta área y este trabajo les sea de apoyo; muchas veces que yo no comprendía algo siempre tenía la esperanza de encontrar a alguien o algo que me ayudara y entender algo que era tan sencillo pero en su momento era un dolor de cabeza, cuando encontraba un trabajo que me explicaba y mi duda estaba resuelta realmente daba gracias por que esa persona me ayudara, a pesar de que esa persona ni siquiera sabía que en una parte del mundo yo le estaba agradecida.

A ti y a quien me enseñó a que la vida no se vive si no te atreves:

La felicidad la tenemos con las personas que nos aman y amamos, muchas veces buscamos la felicidad en muchos lugares y no observamos que está a nuestro alrededor hasta que pasa algo que mueve nuestro entorno es como nos damos cuenta, pero en la vida no hay vuelta atrás así que demuéstrole a los que son tu felicidad que realmente son importantes para tu mundo feliz. Y algo que aprendí es que la vida hay que vivirla, corre sin motivo alguno solo corre ve lo bello que vas pasando y si ves un enorme charco enfrente brinca y siente el agua entrar en tus zapatos disfruta TODO.

«Laura Dojali.»

Agradecimientos

Agradezco al Dr. Jair, quien es mi tutor académico y un gran amigo, porque ha sido mi apoyo y me ha permitido crecer en muchos ámbitos de mi vida, me permitió aprender todo lo necesario para el desarrollo del proyecto. Le agradezco su tolerancia, su tiempo y su buena enseñanza ya que a pesar de que yo no comprendiera su explicación el volvía a retomar su lápiz y me volvía a explicar; Comprendí que aprender no solo es lograr terminar algo, si no que te lleves el conocimiento sin dudas ya que de esta forma en verdad vas a crear grandes cosas, por lo que agradezco el interés y tenacidad que tiene hacia todos sus alumnos para que aprendan y se superen.

Le agradezco al Dr. Farid quien fue mi revisor y un gran amigo, por su tiempo y dedicación para mi crecimiento académico, también le doy gracias por todas las historias que me ha contado porque siempre aprendo de ellas, por todos los ánimos que me ha dado y por ser un maestro que me hace sentir la calma y todas sus explicaciones son fáciles de comprender y te motiva a seguir aprendiendo más.

Le agradezco al Dr. Sergio quien ha sido un buen tutor y un amigo, por toda la paciencia que me tuvo y por todas las clases que me dio donde siempre las hacia divertidas por contarlas con sus experiencias y siempre compartir lo que leía con ello nos permitía sentir interés hacia el área.

Agradezco a mi padre porque me enseñó que no importa el color en que te pinte el camino, que tu puedes pintar un bello paisaje y seguir adelante hasta la meta y triunfar en todo lo que uno realmente decida, a mi mama porque siempre ante cualquier situación me demuestra su cariño y me apoya para que yo comprenda que todo esta bien, un apoyo incondicional y porque siempre ha confiado en mí por lo que me motiva a no defraudarla. Agradezco a mi abuelita porque sabe como forjar en mí el coraje que necesito para lo que la vida tiene preparado y por su interés que siempre ha dado por que yo tenga una preparación y su frase es que tenga herramientas para defenderme. A mis hermanas porque me han apoyado y me dan ánimos, me hacen sentir que hago bien las cosas. ¡A mis sobrinitos les agradezco todas esas hermosas sonrisas, ese amor y el sentimiento de que soy importante para ellos, son una parte muy grande de mi felicidad, son una gran motivación para mí cada que los veo y llego a casa me reciben con un eufórico tíaaa!! Un abrazo y un hermoso beso. Y a la persona que está conmigo siempre demostrándome que me ama.

Agradezco a CONACYT ya que me permitió realizar mis estudios y subir un escalón más para prepararme, aprender un nuevo camino hacia el desarrollo y creación de nuevas generaciones que transmitan y tengan el gusto por enseñar y motivar a los que apenas comiencen. Gracias porque permiten que los estudiantes se sientan apoyados y tengan el sentimiento de poder lograr más y haya quien crea en ellos como lo hizo CONACYT y mi institución Centro Universitario UAEM Texcoco que me abrió las puertas para este nuevo éxito.

Gracias a Dios por que siempre esta conmigo y sin importar la hora o fecha el siempre me escucha y me hace sentir tranquila.

Abstract

Currently, the identification systems of objects in images and video have been used in multiple fields of research. These systems allow solving several problems depending on the area of application. The algorithms in the state of the art allow to detect objects with different precision, this according to the quality of the images and video, color, pre-processing and kind of algorithm. The present work proposes and implements a system for identifying objects in video. To highlight the performance of the algorithms under a particular problem algorithms are implemented to solve a real problem.

Resumen

En la actualidad los sistemas de identificación de objetos en imágenes y video han sido utilizados en múltiples campos de investigación. Estos sistemas permiten resolver diversos problemas dependiendo del área de aplicación. Los algoritmos existentes en el estado del arte permiten detectar objetos con diferente precisión, esto de acuerdo a la calidad de las imágenes y video, color, pre procesamiento y tipo de algoritmo seleccionado. El presente trabajo propone e implementa un sistema de identificación de objetos en video. Para resaltar el desempeño de los algoritmos bajo un problema en particular se implementan los algoritmos para resolver un problema real.

Índice general

Abstract	VII
Resumen	IX
Índice general	XI
Índice de figuras	XV
Índice de cuadros	XVII
1. Introduction	1
1.1. Problemática	2
1.2. Justificación	3
1.2.1. Objetivo General	3
1.2.2. Objetivos específicos	3
1.3. Hipótesis	4
1.4. Estado del arte	4
1.4.1. Sistemas de identificación de objetos en video	4
2. Preliminares	9
2.1. Procesamiento Digital de Imágenes	9
2.1.1. Transformaciones de intensidad	10
Brillo	11
Contraste	11
Dominio espacial	11
Filtro Mediana	12
Filtro de la Media Ponderada	12
Filtro de la Media o Promedio	12
Filtro Gaussiano	12
2.1.2. Histogramas	12

Expansión de Histogramas	13
Contracción del Histograma	13
Desplazamiento del histograma	13
Ecuación del Histograma	13
2.2. Sistema de Reconocimiento de objetos	15
2.3. Manejo de Frames en Video	16
2.4. Técnicas de Evaluación de resultados	18
2.4.1. <i>Cross-validation</i>	18
2.4.2. F-Measure	19
2.4.3. Área ROC	21
3. Técnicas de detección de objetos	23
3.1. Detector de objetos básico (Método de coincidencia)	23
3.1.1. Suma Cuadrática de Diferencias	24
3.1.2. Correlación cruzada normalizada	24
3.1.3. Técnica piramidal	26
3.2. Algoritmo Viola-Jones	27
3.3. Detector de Harris	31
3.4. Detector de objetos SIFT	33
3.4.1. Detección de extremos en el espacio escala	35
3.4.2. Localización de puntos clave estables	36
3.4.3. Asignación de la orientación	37
3.4.4. Descriptor del punto de interés	37
3.5. Detector de objetos SURF	38
4. Metodología	41
4.1. Detector de objetos simple y piramidal	42
4.1.1. Seleccionar plantilla	43
4.1.2. Pre-procesamiento	44
4.1.3. Búsqueda de Candidatos y Detección	45
Algoritmo Piramidal	47
4.1.4. Refinación de Candidatos	49
4.2. Detector utilizando Viola Jones	51
4.2.1. Ejemplos de entrenamiento	51
4.2.2. Extracción de características	51

4.2.3.	Selección de características	52
4.2.4.	Clasificador	52
4.3.	Detector SIFT	52
4.3.1.	Detección	53
4.3.2.	Descripción	55
4.3.3.	Representación de la imagen	56
4.4.	Detector de objetos utilizando el algoritmo SURF	57
4.4.1.	Detección de puntos	58
4.4.2.	Descriptor SURF	59
5.	Resultados experimentales	61
5.1.	Resultados experimentales de detección sobre logos en imágenes	62
5.1.1.	Conjuntos de datos	62
Entrenamiento	62	
Prueba	62	
5.1.2.	Resultados	62
5.2.	Resultados experimentales de detección sobre logos en video	65
6.	Conclusiones	67
6.1.	Conclusiones de los resultados obtenidos	67
6.2.	Trabajo futuro	68
6.3.	Artículos publicados	69
6.3.1.	Revistas Internacionales con Arbitraje Estricto	69
6.3.2.	Capítulos de Libro	69
A.	Articulos publicados	71

Índice de figuras

2.1. Sistema de detección de objetos	15
2.2. Validación cruzada de 4 iteraciones	19
2.3. Representación de similitud entre dos clases. El punto de corte t determina el comportamiento del clasificador.	21
2.4. Representación de tres Curva ROC con distinta área bajo la curva.	22
3.1. Cálculo de las diferencias entre la imagen de búsqueda y prototipo	24
3.2. Cálculo de las diferencias entre la imagen de búsqueda y prototipo	26
3.3. Cálculo de los rectangulos de una imagen integral	29
3.4. Características basadas en rectangulos	31
3.5. Tipos de esquinas más comunes	31
3.6. Pasos comunes de algoritmos de detección de esquinas	32
4.1. Entrada y salida del detector básico	42
4.2. Pasos comunes de algoritmo de detección de objetos simple	43
4.3. Logos con diferentes cambios en diseño y color	44
4.4. Posibles Candidatos	46
4.5. Detección de objetos por el método básico	47
4.6. Algoritmo Piramidal	48
4.7. técnica piramidal	49
4.8. Refinación de candidatos	50
4.9. Resultados de detección de logo utilizando el algoritmo básico	50
4.10. Detección de objetos utilizando Viola-Jones	51
4.11. Etapas del algoritmo SIFT	53
4.12. Influencia de σ en la dispersión del filtro	54
4.13. Características locales con el método SIFT	57
4.14. Etapas del algoritmo SURF	58
4.15. Dcretización de las derivadas parciales de la matriz Hessiana	58
4.16. Asignación de la orientación	59
4.17. Características locales con el método SURF	60

5.1. Logos de partidos políticos registrados en el INE	61
5.2. Falsos positivos detectados por el algoritmo Viola-Jones con solo 50 imágenes de entrenamiento	64
5.3. Mejora en el algoritmo Viola-Jones	64

Índice de cuadros

2.1. Estructura de la Matriz de Confusión	20
3.1. Tamaños de diferentes imágenes con factor de reescalada $s = 1,2$	27
4.1. Resolución de diferentes platillas utilizadas	45
5.1. Tiempos de detección sobre imágenes de tamaño variable	63
5.2. Desempeño de los diferentes métodos en la detección de logos	63
5.3. Tiempos de detección sobre video	65
5.4. Desempeño de los diferentes métodos en la detección de logos	66

Capítulo 1

Introduction

En la actualidad, los sistemas de reconocimiento de objetos en video o imágenes han obtenido un importante auge debido al abaratamiento de los dispositivos de vigilancia, el aumento en la capacidad de los dispositivos de almacenamiento y mejoras en la velocidad de procesamiento, entre otros. Debido a la presencia de grandes cantidades de datos visuales, el desarrollo de técnicas que identifiquen automáticamente objetos y categoricen estos datos son de excepcional importancia. Las aplicaciones de los sistemas de reconocimiento de objetos son muy amplias, estas van desde reconocimiento de huellas dactilares, identificación de escenas criminales, identificación facial de criminales, identificación de plagas y/o enfermedades en plantas, identificación de cadenas cromosómicas que codifican en proteínas, traducción automática de lenguaje de señas a partir de video hasta reconocimiento automático de cultivos a partir de imágenes.

Por la virtud de ser útiles en reconocer e identificar diferentes formas, incluso estas técnicas se utilizan en las empresas que tienen diversas necesidades, como por ejemplo algunas empresas dedicadas a la producción de alimentos del cual su ingrediente principal puede ser el jitomate, para ello la necesidad de la empresa es identificar los jitomates que le sirven para su producto como también eliminar el jitomate podrido de su cosecha. Otro caso que se puede dar en el mercado es empresas de marketing que realizan sus estudios de acuerdo a la identificación de las señales de una persona que pronuncia en su rostro tras ver un artículo que desea o considera atractivo para obtener ventas elevadas. Por otro lado, el simple hecho de realizar un reconocimiento de forma visual puede implicar texturas de un objeto, la cromaticidad tanto en colores, tonalidades de grises o simple blanco y negro, la forma que implicaría el contorno del objeto; pero los humanos también realizamos reconocimiento de sonidos y hasta de olor, entonces dentro de cada reconocimiento se pretende determinar una recolección de las características que se necesitan identificar e indicar cual es el objeto, sonido o olor que

estamos buscando.

Debido a la necesidad de incrementar el uso de los sistemas de reconocimiento de objetos, la utilidad de estos sistemas ha sido relevante en diversas áreas y tan distintas como medicina, biología, agricultura, zoología, investigación criminal, marketing, etc. Los sistemas de reconocimiento de patrones y/o objetos se conocen por realizar aprendizaje máquina, se busca que las computadoras aprendan a reconocer desde forma visual, o de sonido.

En esta tesis se han revisado variadas técnicas para el desarrollo de sistemas para la identificación de objetos en video, los sistemas implementados emulan la capacidad humana para la identificación automática de objetos a partir de colores y formas. Como un caso particular se contempla un problema que se ha observado en el INE, ya que este tiene problemas con el abuso de los spots políticos por parte de los distintos partidos y el INE requiere de sistemas automáticos que identifiquen y cuenten automáticamente el tiempo de cada partido en televisión; para implementar este sistema se llevó a cabo la identificación de los logos de cada partido político utilizando diversas técnicas de reconocimiento automático de objetos. En los experimentos llevados a cabo, se tuvo en cuenta que los sistemas deben identificar los logos con un reducido tiempo de procesamiento. Los tiempos son muy importantes para el computo porque se debe encontrar una solución eficiente en un tiempo aceptable.

1.1. Problemática

En la actualidad el uso de sistemas para la detección de objetos específicos ha sido un reto importante, la eficiencia de identificación de los objetos y el control de factores influyentes en las imágenes o videos como es la resolución y la iluminación son factores que influyen en la eficiencia de la identificación. Los sistemas deben cumplir en algunos casos con la capacidad de una segmentación adecuada para la separación de los objetos de su fondo y otros objetos que no son de interés, la resolución es cambiante de acuerdo al dispositivo y por lo tanto es difícil trabajar con diferentes tipos de resolución. En un sistema siempre se debe buscar restringir los problemas que este debe resolver de tal forma que cumpla en tiempo y forma. Al ser esto un problema que atañe actualmente se pretende en esta tesis implementar diferentes técnicas de identificación de objetos y obtener un análisis comparativo de ellas. Concretamente se pretende desarrollar un sistema de identificación de logos automático que permita llevar un control de spots de cada partido político. La solución de este problema es importante ya que se consideran

parámetros de dinero y saturación social de la propaganda electoral, debido a que los partidos políticos no respetan un número de spots ni el tiempo de estos en televisión.

1.2. Justificación

Los sistemas de identificación de objetos sirven de mucho en la actualidad ya que permiten el ahorro de dinero y determinan su trabajo con una mayor precisión. Existen sistemas creados para la Seguridad (Seguridad para el reconocimiento de personas en aeropuertos, bancos o establecimientos con un control de acceso, etc) (Wang, Tony , & Shuicheng, Noviembre 2009), Botánica (Vigilancia en el comportamiento de las plantas y su identificación de estas), Etología (Vigilancia e identificación de insectos, aves, murciélagos, entre otros mamíferos; obteniendo una identificación de comportamiento) (Marcovecchio, Stefanazzi, Delrieux, Maguitman, & Ferrero, 2013), Prevención de incendios (Reconocimiento de humo) (Llobregat, 2011) o incluso en Búsquedas por internet (Sivic & Zisserman, 2003), (Toshev, Makadia, & Daniilidis, 2010), (Sivic & Zisserman, 2003), (Dalal & Triggs, 2014). Incluso en México se considera que es necesario un control de spots políticos de acuerdo a cantidad y tiempo de estos, lo cual al ser un problema (Alcántara,2005), (Amparán, El framing del Spot Político , 2011) (Amparán. El framing audiovisual del spot político, 2010) por ello consideramos realizar una aplicación que dé solución a este problema. Finalmente, los resultados aportados en esta tesis permitirán llevar un conteo automático eficiente de los spots que realiza cada partido político en televisión.

1.2.1. Objetivo General

Emplear las técnicas de detección de objetos reconocidas en el estado del arte para la identificación automática de objetos en video y añadir a esto una aplicación para la detección de logos políticos, de tal forma que se tenga una identificación de los partidos además de la cantidad y tiempo de cada spot político en las televisoras.

1.2.2. Objetivos específicos

1. Investigar y determinar las técnicas que se utilizan para el desarrollo de un sistema que identifique objetos en video.

2. Investigar y aplicar las posibles técnicas utilizadas en separación de frames en un video.
3. Identificar las técnicas para reconocer el inicio y fin de un frame.
4. Investigar e Identificar las posibles características que se pueden contemplar para una buena identificación de objetos en video.
5. Investigar los algoritmos de clasificación utilizados en sistemas de identificación.
6. Implementar un algoritmo para la optimización de tiempos para la clasificación.
7. Identificar la tasa de error de cada clasificador para determinar el algoritmo de clasificación óptimo.
8. Realizar la validación de los algoritmos desarrollados para la aplicación.
9. Realizar las pruebas finales para nuestra aplicación

1.3. Hipótesis

¿Es factible el desarrollo de un sistema para identificación de objetos en video?

1.4. Estado del arte

El estado del arte del objetivo de esta investigación podemos clasificarlo en dos grandes bloques. El primero que describe las técnicas de extracción de características más utilizadas para detectar objetos en video y el segundo que describe los sistemas de identificación de objetos en video y sus aplicaciones.

1.4.1. Sistemas de identificación de objetos en video

Los sistemas de identificación de objetos en video, ha sido un tema de importancia para muchos proyectos en distintas áreas como seguridad (Wang et al, 2009), Botanica, Etología (Marcovecchio et al, 2013), Búsqueda (Sivic y Zisserman, 2003), (Toshev et al, 2010), etc. Incluso tuvo un gran auge debido a la necesidad de identificación de peatones para el manejo automático de autos, lo cual beneficiaría a las grandes empresas

fabricantes de autos y a los peatones, ya que se considera que esto reduciría la cifra de accidentes automovilísticos. (Dalal y Triggs, 2014), (Min et al, 2018). Sin embargo, el número de aplicaciones donde se utilizan técnicas para identificación de objetos es muy grande y una área de investigación muy actual (Wang et al, 2017), (Wei et al, 2017), (Yaseen et al, 2018), (Yuan et al, 2017), (Burak et al, 2016), (Xu G. et al, 2017), (Xiao et al, 2017).

Recientemente, la conducción autónoma ha sido ampliamente estudiada y ha demostrado ser prometedora. La detección de objetos peligrosos basada en la visión es una tecnología crucial de conducción autónoma. En trabajos previos, la detección de objetos peligrosos generalmente se formula como un problema típico de detección de objetos y un problema de evaluación de peligros basado en la distancia, por separado (Chen et al, 2017) (Rhee et al, 2017). Una de las áreas más activas en visión artificial es la detección de objetos, que ha mejorado significativamente en los últimos años. Los métodos actuales de detección de objetos de última generación se adhieren principalmente al marco de las regiones con redes neuronales convolucionales. Sin embargo, estos sistemas solo aprovechan las características de apariencia local, ignorando la información contextual (Chu y Cai, 2017). Dalal (Dalal y Triggs, 2014) utiliza SVM para detección de peatones, además de utilizar un descriptor basado en histogramas de gradientes ordenados (HOG); también utilizó el descriptor SIFT y obtuvo buenos resultados, reduciendo las tasas de falsos positivos al utilizar un detector de Haar wavelet. Wang y compañía (Wang et al, 2009) desarrollan un sistema para incrementar la seguridad de los peatones utilizando un detector de peatones basado en HOG-LBP, el sistema hace uso de un manejo parcial de oclusión.

Xu et al (Xu et al, 2017). Presentan una novedosa arquitectura multi-modal de aprendizaje de características profundas para la detección de objetos. La mayoría de los enfoques de detección de objetos existentes constan de dos etapas: estimación de objetividad y reconocimiento de objetos a nivel de región. Las técnicas actuales tratan las dos etapas por separado extrayendo las características RGB y de profundidad individualmente, ignorando así la relación correlacionada entre estas dos modalidades. En este artículo los autores proponen un método para aprovechar al máximo las señales de profundidad y color.

Sivic y Zisserman (Sivic y Zisserman, 2003) desarrollan un sistema para reconocer palabras y localizar todas las ocurrencias en un video; en sus experimentos representan los objetos por un conjunto de descriptores de regiones con puntos invariantes.

Los sistemas de visión son capaces de procesar mayores cantidades de datos en un

menor tiempo, además de gastar menor tiempo que una persona, aunque no sería capaz de reconocer un objeto si este no se le indica con anterioridad. El ser humano es capaz de extraer más información de una imagen, debido a que nosotros recabamos información de diferentes patrones, gracias a ello determinamos con mayor facilidad la separación de los objetos en diversas imágenes y su identificación. Los sistemas procesan la información obtenida y aplican los algoritmos para el procesamiento de las imágenes y determine una correcta identificación del objeto.

Los sistemas de detección de objetos presenta numerosos retos; desde como modelar los objetos de forma que se puedan discriminar unas clases de otras, también se debe contemplar la invarianza de algún cambio del objeto que nos interesa y además de la variabilidad que se tenga en el entorno donde se busque el objeto, tal es como los cambios de iluminación y resoluciones. Las búsquedas de los objetos debe darse de una forma donde consideremos todas las ventanas posibles que puedan contener el objeto y por consecuencia tengamos una búsqueda eficiente.

Uno de los temas de investigación actual se refiere a la detección automática de logos. Se han utilizado variadas técnicas con el fin de tener una búsqueda confiable del logo y el tiempo de búsqueda se reduzca. Algunos proyectos que han compartido este interés por la identificación de logos en video se enumeran a continuación:

1. Mao et al desarrollan un sistema de detección del logotipos de vehículos (Mao et al, 2013). El sistema localiza los logotipos buscando las regiones con la máxima información útil. Los investigadores sostienen que su método es muy rápido y puede lograr una alta tasa de detección, que sería adecuada para aplicaciones en tiempo real.
2. Diligenti et al, proponen un enfoque para reconocer logotipos de empresas corrompidas por el ruido (Diligenti et al, 2001). El enfoque se basa en representaciones estructuradas apropiadas de patrones que, posteriormente, son procesados por redes neuronales recursivas. El sistema que proponen intenta evitar el proceso de extracción de características sin disminuir la eficiencia del sistema.
3. Gori y compañía (Gori et al, 2003) también proponen un sistema de identificación automática de logos. El sistema permite detectar logotipos de empresas corrompidos por ruido artificial. El sistema que proponen permite mejorar el rendimiento de un perceptrón multicapa que funciona como autoasociadores para clasificar

elementos gráficos en presencia de ruido puntual en la imagen. La mejora se obtiene al introducir una norma ponderada en lugar de usar la norma euclidiana para medir la precisión de entrada y salida de la red neuronal.

4. En (Alaei et al, 2016) los autores proponen un sistema de detección automática de imágenes de documentos administrativos en relación con sellos y logotipos. Los autores sostienen que la investigación puede ser útil para los investigadores de la comunidad de análisis de documentos. Ya que puede utilizarse eficazmente para diferentes tareas de recuperación, clasificación y reconocimiento de imágenes de documentos en aplicaciones basadas en documentos.
5. En (Tang y Peng, 2017) los autores proponen un método de correspondencia robusto para obtener los mejores resultados de coincidencia mediante la explotación de la restricción topológica distintiva junto con selección de características. La sólida compatibilidad de funciones locales juega un papel importante en el reconocimiento de imágenes de logotipos. La mayoría de los métodos tradicionales consideran la característica local individual pero ignoran la relación geométrica invariante afín entre las características locales adyacentes, lo cual es esencial para reducir el número de desajustes.
6. Bianco y compañía (Bianco et al, 2017) proponen un método para el reconocimiento de logotipos utilizando técnicas de aprendizaje profundo. En sus experimentos, evalúan el efecto en el rendimiento de reconocimiento del aumento de datos sintético contra el real, y el preprocesamiento de imágenes. Además, investigan los beneficios de diferentes opciones de entrenamiento. Los autores sostienen que el método supera los métodos actuales en el estado del arte.

Capítulo 2

Preliminares

2.1. Procesamiento Digital de Imágenes

El reconocimiento de objetos emplea técnicas de pre-procesamiento de imágenes con el objetivo de mejorar la calidad de la información contenida en una imagen, además de facilitar la búsqueda de información para el reconocimiento de diversos patrones. Al conjunto de técnicas y procesos para hacer resaltar información contenida en una imagen, mediante una computadora se le conoce como procesamiento digital de imágenes (Reyes,2008) (Torres, 1996)

El procesamiento digital de la imagen (PDI), permite eliminar la mayor cantidad de ruido que se obtiene durante la adquisición. El PDI se realiza principalmente en dos áreas: en el mejoramiento de la información pictórica para la interpretación humana y en el procesamiento de datos de la imagen para su almacenamiento. En el PDI se busca mejorar las características de la imagen como es mejora de contorno, color, brillo, tonalidades de grises, emborronamiento, realce, interpolación, etc. En particular en esta tesis solo se redimensiona la imagen y en algunos casos se utilizarón diferentes técnicas como realce, emborronamiento, etc

Para el reconocimiento de patrones, en un sistema podemos mencionar dos formas de trabajo las cuales podran ser:

1. Identificar un objeto de la imagen y desechar el entorno mediante la segmentación, aplicando las técnicas de PDI sobre la imagen segmentada que es el objeto de interes. Cuando se segmenta se puede realizar PDI para mejorar la imagen como puede ser darle mas brillo, resaltar los bordes de está (cuando se tiene la necesidad de trabajar con el contorno de esta imagen), eliminar ruido, emborronar, inclusive cambios en intensidad, etc.

2. Reconocer un objeto dentro de una imagen sin necesidad de segmentar dado que no se tiene previsto realizar cambios en el objeto de forma directa, este proceso es utilizado en esta tesis.

La mejor forma de evitar pre-procesamientos elaborados es concentrarnos en la adquisición de imágenes de alta calidad. Las técnicas de pre-procesamiento no incrementan la información contenida en la imagen. Si la información es medida utilizando la entropía (Shanon,1948), entonces el pre-procesamiento reduce el contenido de información en la imagen. Sin embargo, muchas veces es necesario realizar pre-procesamiento en la imagen para suprimir información que no es relevante en las tareas de análisis.

Existen dos grandes campos de pre-procesamiento de imágenes. El primer campo permite mejorar la imagen, mientras que el segundo reduce la imagen, permitiendo centrar nuestra atención sobre una pequeña área de interés. El objetivo de la mejora de la imagen es hacer transformaciones sobre la imagen para obtener la información necesaria de las propiedades físicas de los objetos de la imagen, permitiendo que sean de mayor utilidad en nuestro sistema. Por otro lado, la detección de objetos en imágenes es un proceso computacionalmente costoso. Para evitarlo en la actualidad se utilizan técnicas de pre-procesado en la imagen, cuyas técnicas permiten reducir considerablemente el área de la imagen con posibilidades de encontrar los objetos reduciendo el costo computacional del proceso total. El preprocesado, por tanto, es una forma de obtener regiones de interés de manera que éstas sufran los procesos de detección ignorando el resto de la imagen. La principal ventaja del preprocesado es por tanto la reducción del coste computacional del análisis del fotograma, aunque en sí también conlleva coste. Además de esto, el preprocesado puede ayudar a evitar falsas detecciones si está bien hecho, aunque si es defectuoso también puede evitar que se analicen zonas en las que hay un logo presente y por tanto evitar la detección de este.

En esta Sección se mencionan diferentes técnicas de PDI que se utilizan a menudo en la detección automática de objetos en imágenes/video.

2.1.1. Transformaciones de intensidad

Las transformaciones de intensidad modifican el brillo en los píxeles. Este tipo de transformaciones dependen de cada uno de los píxeles en sí. En estas transformaciones solo intervienen el píxel de la imagen de entrada para determinar el valor del píxel en misma posición de la imagen de salida.

$$N(x, y) = f(O(x, y)) \quad (2.1)$$

donde $N(x, y)$ es la nueva imagen obtenida al aplicar una función f , y $O(x, y)$ es la imagen original.

Brillo

El aumento de brillo consiste en sumar o restar una constante a las intensidades de la imagen, trabajando pixel a pixel respetando los intervalos de intensidad 0 y 255, k puede ser negativo o positivo.

$$N(x, y) = O(x, y) + K \quad (2.2)$$

Contraste

La idea es llevar el valor mínimo a cero y el valor máximo a 255, para el aumento de contraste o inversa. El contraste es un juego de colores para resaltar o disminuir los tonos de un objeto en la imagen.

$$N(x, y) = (O(x, y) - \text{mín} / (\text{máx} - \text{mín}) * 255) \quad (2.3)$$

donde min representa el valor mínimo de intensidad, max el valor máximo de intensidad.

Dominio espacial

Estas técnicas de dominio espacial operan directamente sobre los píxeles de la imagen, operan con la vecindad de los píxeles, mediante una máscara cuadrada o rectangular. Una máscara es una pequeña imagen con valores predeterminados para cada posición.

El siguiente es un ejemplo de una máscara de 3×3 . La máscara se centra sobre el píxel de interés de forma que el nuevo valor del píxel depende de los píxeles que cubre la máscara.

Filtro Mediana

El filtro de la mediana sustituye el valor del píxel central por el de la mediana, los píxeles de la nueva imagen se generan calculando la mediana del conjunto de píxeles del entorno de vecindad del píxel tomado. Este filtro sirve para eliminar ruido aleatorio como sal y pimienta.

Filtro de la Media Ponderada

En la máscara no son todos los valores 1, se debe dar más peso al valor central para que el resultado no varíe tanto de la imagen original, evitando que se vea borroso.

Filtro de la Media o Promedio

Obtiene el promedio de los píxeles vecinos con $w = 1$; donde todos los valores de la máscara son 1's.

Filtro Gaussiano

Este filtro reduce el ruido tipo Gaussiano, que produce variaciones en la imagen, originado por las diferencias en el sensor y ruido en la digitalización. El filtro en general obtiene mejores resultados que un filtro promedio o media. La desventaja de este filtro es que suaviza las orillas o discontinuidades, generando que se emborronen. El filtro está dado por la siguiente ecuación:

$$G(x, y) = e^{-x^2+y^2/2\pi\sigma^2} \quad (2.4)$$

donde σ ; es la desviación estandar

2.1.2. Histogramas

El histograma de una imagen, representa el número de píxeles en función de los niveles de intensidad; Un histograma representa la probabilidad de ocurrencia de un determinado nivel de intensidad y se define como:

$$P(L) = \frac{N(L)}{TP} \quad (2.5)$$

donde $P(L)$ es la probabilidad de ocurrencia en las intensidades, $N(L)$ número de píxeles de intensidad L , TP número total de píxeles en la imagen.

Expansión de Histogramas

Consiste en distribuir las frecuencias de la aparición de los píxeles en todo el ancho del histograma, el valor de la intensidad más baja sea cero y la intensidad más alta sea el valor máximo. Para expandir los niveles de gris de un histograma se define:

$$N(x, y) = \left[\frac{O(x, y) - \text{mín}}{\text{máx} - \text{mín}} \right] [MAX - MIN] + MIN \quad (2.6)$$

donde $O(x, y)$ es el nivel de gris de la imagen entrada, max valor máximo del nivel gris de la imagen, min representa el valor mínimo del nivel de gris de la imagen entrada, MAX valor máximo posible de los niveles de gris (255), MIN valor mínimo posible de los niveles de gris (0).

Contracción del Histograma

La contracción de histograma no produce un realzado sino una disminución del contraste de la imagen.

$$N(x, y) = \left[\frac{MAX - MIN}{\text{máx} - \text{mín}} \right] [O(x, y) - \text{mín}] + MIN \quad (2.7)$$

Desplazamiento del histograma

Técnica que se utiliza para aclarar o oscurecer una imagen, creando una relación entre los valores de los niveles de gris mediante una adición o sustracción de una constante que será el desplazamiento del histograma.

$$N(x, y) = O(x, y) + C \quad (2.8)$$

donde C es una constante, $O(x, y)$ el valor del píxel

Ecualización del Histograma

La ecualización del histograma es una técnica para armonizar la distribución del nivel de luminosidad de la imagen, da como resultado que todos los niveles de gris tengan la misma frecuencia. Formalmente, el algoritmo para realizar la ecualización es el siguiente:

Algorithm 1 Algoritmo de Ecuación de histograma)

Require: Imagen de entrada $O(x, y)$

Ensure: Imagen ecualizada $N(x, y)$

- 1: Para una imagen de tamaño $N \times M$ con G niveles de gris (a menudo 256) crear un arreglo H de longitud G inicializado con valores 0.
- 2: A partir de la imagen: escanear cada pixel e incrementar cada miembro de H , si el pixel p tiene una intensidad g_p , realizar

$$H[g_p] = H[g_p] + 1. \quad (2.9)$$

- 3: A partir del histograma acumulado de la imagen H_c

$$H_c[0] = H[0], H_c[p] = H_c[p - 1] + H[p], p = 1, 2, \dots, G - 1 \quad (2.10)$$

- 4: Asignar

$$T[p] = \text{round} \frac{G - 1}{NM} H_c[p]. \quad (2.11)$$

Este paso permite una rápida implementación al construir una tabla de múltiplos de $\frac{(G-1)}{NM}$.

- 5: Reescanear la imagen y escribir una imagen de salida con niveles de grises g_q , donde $g_q = T[g_p]$. Esta representación asume que el rango de intensidad de las imágenes fuente y destino se encuentran en el intervalo $[0, G - 1]$
-

2.2. Sistema de Reconocimiento de objetos

El reconocimiento de objetos es un área de la visión por computadoras y consiste en el desarrollo de técnicas para encontrar e identificar objetos en una imagen o secuencia de video. Los humanos reconocemos una multitud de objetos en imágenes con poco esfuerzo, a pesar del hecho que la imagen del objeto puede variar un poco en diferentes puntos de vista, en diferentes tamaños o escala e incluso cuando están trasladados o rotados. Incluso, es relativamente fácil para un humano reconocer objetos cuando están parcialmente obstruidos desde algún punto. No obstante, esta tarea es un desafío para los sistemas de visión por computadoras. Para este problema se han implementado muchos métodos durante múltiples décadas. Las técnicas implementadas emulan la capacidad humana de reconocer objetos sin importar la escala, rotación, posición e incluso diferentes perspectivas o una visión obstruida del objeto. Una parte importante del reconocimiento de objetos en imágenes se ocupa de los procesos relacionados con el fin de extraer información que establezca propiedades acerca de un objetos físicos o abstractos en una imagen, un esquema simple del proceso de reconocimiento de objetos en imágenes puede ser observado en la Figura 2.1.

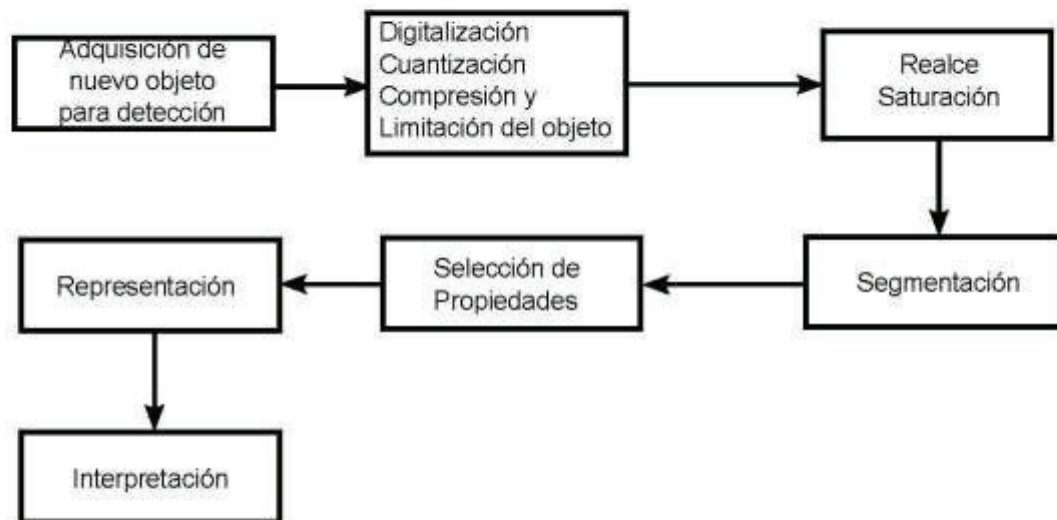


FIGURA 2.1: Sistema de detección de objetos

El proceso empieza por definir el objeto a detectar, una vez definido se delimita el objeto en la imagen y se extraen las características que lo definen. Este conjunto de características puede ser obtenido mediante cualquiera de las técnicas en el estado del arte. Cada grupo de características posee una capacidad discriminativa diferente.

En general, un sistema de reconocimiento de objetos completo consiste de sensores que recogen las observaciones a clasificar, un sistema de extracción de características que transforma la información observada en valores numéricos o simbólicos y un sistema de clasificación o descripción que, basado en las características extraídas, clasifica la medición.

Las características de la imagen pueden ser:

1. topológicas: número de componentes conexas, agujeros, etc.
2. geométricas: área, perímetro, curvatura, etc.
3. estadísticas: momentos

Un patrón es un conjunto de características, mientras que una clase de patrones es un conjunto de patrones "similares". El objetivo del reconocimiento de patrones es asignar un patrón a la clase a la que pertenece (lo más automáticamente posible).

En la actualidad, uno de los principales desafíos del reconocimiento de objetos es generar algoritmos capaces de extraer características acordes al problema con una gran capacidad discriminativa, y definir descriptores que permitan hacer una representación de dichas características. Este es uno de los desafíos fundamentales de la visión computacional y diversos acercamientos han sido implementados a lo largo de décadas para solucionarlo.

2.3. Manejo de Frames en Video

Un video es una sucesión de imágenes presentadas en secuencias. El ojo humano es capaz de distinguir 20 imágenes por segundo y crear un efecto de película a partir de los 14 fotogramas por tal motivo en la tele blanco y negro se presentaban a 16 Frames por segundo (FPS), hoy en día las Televisoras utilizan 24 FPS y en el cine se han utilizado hasta 60 FPS pero eso no es un límite porque algunas cámaras ya proyectan hasta 1000 FPS.

Al aumentar ésta cantidad se perciben los movimientos con mayor contraste y sin efecto de pixeleo, ya que no da lugar a que el ojo perciba los cambios. La fluidez dependerá de la cantidad de imágenes por segundo, esto es la frecuencia de cuadros, frames por segundo por lo tanto las imágenes que captura la cámara en cada segundo presenta mayor calidad en la película por ejemplo si se graba a 60 FPS el video esta compuesto

de 60 fotos por unidad de tiempo, en cambio si se graba a 30, solo son 30 fotos por segundo.

Fotograma es el termino utilizado como traducción de Frame, FPS es una medida de como se muestra el video en movimiento, el término se aplica igualmente a la película y video digital, cada fotograma es una imagen fija. (Konigsberg, 2004), (Carrasco, 2010)

La resolución de una imagen es el número de píxeles por pulgada que contenga, sabemos que una pulgada equivale a 2.54 centímetros; La resolución se expresa en puntos por pulgada (PPP), la imagen se apreciará mejor al contener mayor número de píxeles por pulgada. Por ejemplo una resolución de 400 dpi (dots per inch en ingles) significa que la imagen contiene 400 píxeles de ancho y 400 píxeles de alto, por lo que tiene 160,000 (400*400 ppp) entonces una resolución total de 1,6 Mp, la resolución se expresa en Megapíxel que equivale a 1 millón de píxeles.

Se conocen como efectos de transición para la cinematografía a la forma o manera de unir dos planos consecutivos. Los diferentes tipos de transiciones entre planos son :

1. Corte:

El corte hace que el espectador interprete una nueva imagen sin perder la continuidad, se realiza un ensamblado de una imagen con otra.

2. Entradas y salidas de campo:

Cuando un personaje u objeto en movimiento debe salir o entrar en el campo de encuadre entre planos consecutivos.

3. Fundidos:

El fundido es una transición en negro entre planos, la primer imagen va desapareciendo paulatinamente hasta convertirse la pantalla en negro; se utiliza para separar temporalmente los episodios o secuencias del relato, en este caso para separar los spots y programación de la televisión.

4. Encadenados:

Se mezclan dos planos, mientras uno se desvanece el otro va apareciendo de forma paulatina, se mezclan las dos imágenes; se utiliza para mostrar el paso del tiempo en periodos breves.

5. Desenfoque:

Desenfoca dos planos entre si, utilizado para los cambios espaciales.

6. Cortinillas:

Técnica de invasión de una imagen a otra, desde cualquier ángulo, un nuevo plano aparece y empuja al plano anterior hasta reemplazarlo por el primero, ya no se utiliza en grandes producciones.

7. Barrido:

Es un movimiento de cámara rápido que no se distingue con claridad el contenido, el efecto es muy parecido con el desenfoque.

La transición en la que nos enfocaremos es Fundido, ya que es la técnica que utiliza la televisora para realizar la separación entre planos que en nuestro caso los spots de los partidos políticos, para determinar el tiempo de duración de cada spot. El espacio oscuro nos ayuda a determinar inicio y fin de los spots.

2.4. Técnicas de Evaluación de resultados

A fin de evaluar la exactitud de predicción de un modelo particular o para evaluar y comparar diferentes modelos, se considera el rendimiento relativo en el conjunto de datos de prueba. Debido a la importancia fundamental de la clasificación y detección en muchas situaciones prácticas, se deben tomar ciertas medidas para seleccionar un modelo. Por esta razón, varias métricas de rendimiento se proponen en la literatura. Estas medidas, son reconocidas como un elemento importante en toda gestión de calidad. Para validar nuestros resultados utilizamos varias medidas de desempeño.

En esta Sección se describen algunas métricas de desempeño importantes que son frecuentemente utilizadas.

2.4.1. *Cross-validation*

La validación cruzada (*Cross-validation*), es un método para evaluar y comparar los algoritmos de aprendizaje mediante la división de datos en dos segmentos. El primero se utiliza para entrenar el modelo y el segundo para validar el modelo.

El algoritmo primero divide los datos en k partes iguales. Después realiza k iteraciones de entrenamiento, tomando en cada iteración como conjunto de prueba un

subconjunto diferente y construyendo el modelo con los subconjuntos restantes. La Figura 2.2 muestra un ejemplo con 4 iteraciones. El índice de error estimado es la media de todos los errores obtenidos en cada entrenamiento.



FIGURA 2.2: Validación cruzada de 4 iteraciones

La ventaja de evaluar a partir de k combinaciones de datos de entrenamiento y prueba hace que el método sea más preciso. Sin embargo, en una evaluación con un valor alto en k el proceso es lento al momento de computar. La elección del número de iteraciones depende de la medida del conjunto de datos, aunque lo más común es utilizar la validación cruzada de 10 iteraciones.

2.4.2. F-Measure

F-Measure no es más que la media armónica entre precisión y exhaustividad. La precisión representa el nivel de confianza del clasificado ya que es el porcentaje de datos clasificados correctamente. La exhaustividad representa la cobertura del clasificador, es decir, la cantidad de datos que clasifica frente a los no clasificados y clasificados. Cuando un sistema clasifica todos los datos en una sola categoría, este puede tener una exhaustividad alta, sin embargo, si la clasificación es incorrecta la precisión será baja. Tanto la precisión como exhaustividad están basadas en la matriz de confusión que está formada por cuatro casos:

Verdaderos positivos (TP): Es el caso de los datos positivos que han sido clasificados como positivos, es decir, es el total de datos que han sido clasificados correctamente.

Verdaderos negativos (TN): Es el caso de los casos negativos que ha sido clasificados como negativos, es decir, representan el número de datos clasificados en otra categoría correctamente.

Falsos Positivos (FP): Es el caso de los datos negativos que ha sido clasificados como positivos.

Falsos Negativos (FN): Es el caso de los datos positivos que han sido clasificados como negativos.

Partiendo de estos casos se puede formar la matriz de confusión como se muestra en la Tabla 2.1

CUADRO 2.1: Estructura de la Matriz de Confusión

	Positivos	Negativos
Positivos	TP: Verdaderos Positivos	FN: Falsos Negativos
Negativos	FP: Falsos Positivos	TN: Verdaderos Negativos

La técnica de matrices de confusión, no solo permite conocer el error del modelo predictivo, sino que también muestra el tipo de predicciones correctas e incorrectas cuando se aplica el modelo sobre el conjunto de prueba. Las predicciones correctas estas representadas por la diagonal principal, sin en cambio los elementos ubicados fuera de la diagonal principal, indican los errores de asignación.

Dada la matriz de confusión se pueden obtener la precisión y exhaustividad con las siguientes ecuaciones:

$$precision = \frac{TP}{TP + FP} \quad (2.12)$$

$$exhaustividad = \frac{TP}{TP + FN} \quad (2.13)$$

Para calcular F-Measure de una clase j con otra clase i primero se define la ecuación siguiente:

$$F_{ij} = \frac{2 * precision(i, j) * exhaustividad(i, j)}{precision(i, j) + exhaustividad(i, j)} \quad (2.14)$$

entonces F-Measure de un conjunto dado es calculado como sigue:

$$F - Measure = \sum \frac{n_i}{n} \max(F_{ij}) \quad (2.15)$$

donde n es el número de todo el conjunto de datos y n_i es el número de datos de la clase i .

El rango de los valores calculados esta entre 0 y 1. Un valor F-measure alto indica una mayor calidad de clasificación.

2.4.3. Área ROC

Cuando los errores llevan asociada una pérdida que puede cuantificarse, es posible aplicar otra técnica de validación como el análisis de la curva ROC (por sus siglas en inglés, *Receiver Operating Characteristics*). Los gráficos ROC son útiles para visualizar el desempeño de los clasificadores y se utilizan comúnmente en la toma de decisiones médicas, aunque en los últimos años se han utilizado cada vez más en el aprendizaje automático (Tom 2006). El método consiste en un gráfico que ayuda a visualizar la disyuntiva entre la tasa de verdaderos positivos y la tasa de falsos positivos de un clasificador. La tasa de verdaderos positivos se representa en el eje las y , y la tasa de falsos positivos se representa en el eje de las x . La Figura 2.3 muestra un ejemplo de similitud que puede existir entre dos clases.

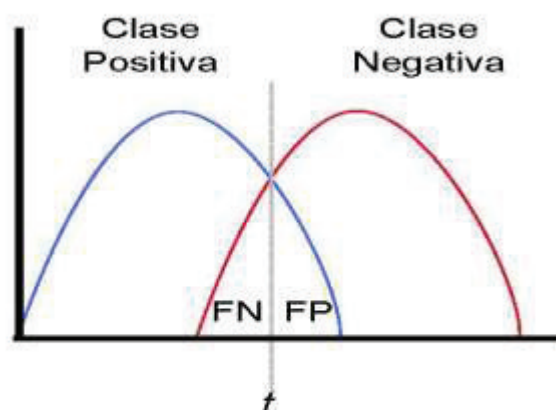


FIGURA 2.3: Representación de similitud entre dos clases. El punto de corte t determina el comportamiento del clasificador.

El comportamiento de pruebas depende del punto de corte t . Si este se desplaza a la derecha (Clase Negativa) disminuye la tasa de falsos positivos y aumenta la tasa de falsos negativos. Inversamente si se desplaza a la izquierda (Clase Positiva) aumenta la tasa de falsos positivos pero disminuye la tasa de falsos negativos. Entonces, para caracterizar el comportamiento entre estas dos clases se utilizan las curvas ROC. Un ejemplo de curva ROC se muestra en la Figura 2.4 .

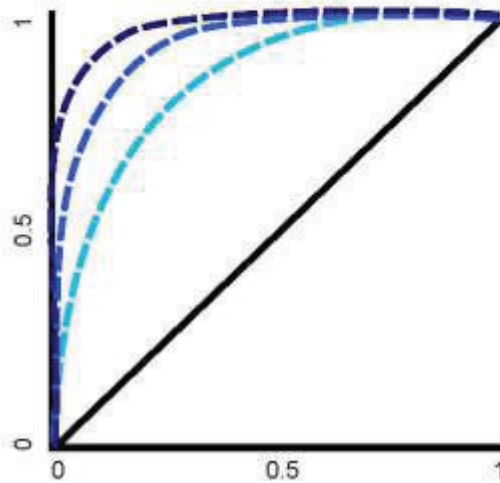


FIGURA 2.4: Representación de tres Curva ROC con distinta área bajo la curva.

Si la prueba fuera perfecta, es decir que no exista solapamiento entre clases, la curva solo tiene un punto $(0,1)$. Sin embargo, si la prueba fuera mala, la curva sería una diagonal de $(0,0)$ a $(1,1)$. La Figura 2.4 muestra un ejemplo de distintos tipos de solapamiento y las tipos de curvas ROC que se generan, en la gráfica mientras más oscuro es el color de la línea más área bajo la curva posee.

El parámetro para evaluar la bondad de la prueba, es el área bajo la curva ROC que toma valores entre 1 (prueba perfecta) y 0.5 (prueba fallida). Esta área puede interpretarse como la probabilidad de que un conjunto de datos ante un clasificador funcione correctamente.

Capítulo 3

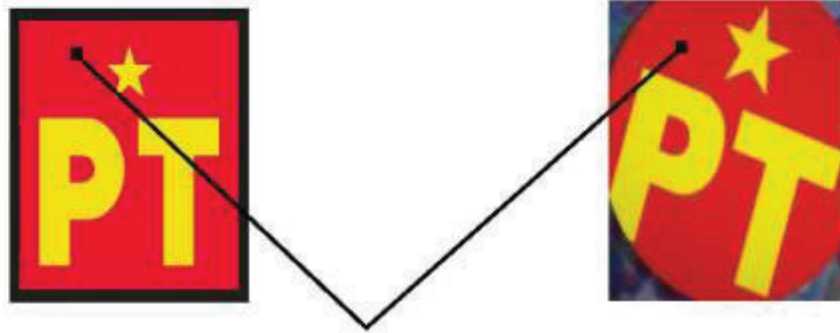
Técnicas de detección de objetos

La detección de objetos es una área actual de mucha efervescencia, en los últimos años se han desarrollado diversas técnicas para detectar objetos bajo diferentes escenarios y condiciones. Se busca que los sistemas sean eficientes y cumplan con el objetivo en su totalidad haciendo énfasis en su exactitud y rapidez. Actualmente la detección de objetos en movimiento en secuencias de video es un paso importante para el desarrollo e investigación, ya que implica una vertiente nueva a explotar. Vivimos en un mundo donde la búsqueda de información avanza constantemente en cualquier área o lugar; desde seguimiento de personas, video-vigilancia, control de tráfico, nuevos proyectos de manejo automático de autos y hasta en la agricultura, como separación de producto de calidad o producto inmaduro. Además de no perder la importancia de la medicina o cirujías en nuestros tiempos, donde se está avanzando rápido y se busca mayor eficacia en la detección de objetos bajo una cámara y con el uso de sistemas que permitan identificar con eficiencia y arrojar datos importantes previos que son de utilidad. En este Capítulo se muestran las diferentes técnicas de detección automática de objetos. En la detección de objetos se encuentran muchos retos, tales como mejorar el poder discriminativo de los diferentes métodos bajo condiciones diferentes de iluminación, escalado y rotación. En este Capítulo describimos los métodos de detección de objetos más utilizados en el estado del arte.

3.1. Detector de objetos básico (Método de coincidencia)

El enfoque más sencillo de detección de objetos es la técnica de detección de coincidencias que utiliza una imagen prototipo del objeto a localizar. En este método, la idea básica es ir comparando los píxeles en el prototipo con la imagen de búsqueda. El prototipo es seleccionado en una fase previa al proceso de reconocimiento. La técnica mide las diferencias de cada píxel entre la imagen de búsqueda y el prototipo como se

muestra en la Figura 3.1. La tarea de reconocimiento es encontrar la posición precisa del objeto en una imagen. La tarea es llevada a cabo moviendo la plantilla pixel a pixel para cada posible posición en imagen. Existen diferentes técnicas para medir esta diferencia. A continuación se muestran algunas de las más implementadas.



$$Dist_{DC}(x, y) = \sum_{x', y'} [P(x', y') - I(x + x', y + y')]^2$$

FIGURA 3.1: Cálculo de las diferencias entre la imagen de búsqueda y prototipo

3.1.1. Suma Cuadrática de Diferencias

Esta técnica mide la diferencia de cada pixel en el prototipo y la imagen donde se buscará, cada una de estas diferencias la eleva al cuadrado. La formula que utilizamos es la siguiente:

$$Dist_{SCD}(x, y) = \sum_{x', y'} [P(x', y') - I(x + x', y + y')]^2 \quad (3.1)$$

donde (x, y) nos define la posición de cada pixel, $P(x', y')$ representa la posición del pixel en el prototipo, $I(x + x', y + y')$ la posición del pixel en la imagen donde se busca el prototipo. Este modelo depende exclusivamente de la intensidad del área cubierta por el prototipo de la imagen de entrenamiento. Una gran desventaja de esta técnica es que es muy sensible a las variaciones de brillo.

3.1.2. Correlación cruzada normalizada

La correlación cruzada normalizada permite obtener un mejor grado de pertenencia entre el prototipo y la imagen de búsqueda, esta es definida mediante:

$$Dist_{CCN}(a,b) = \frac{\sum_{k,l} (I(k+a, l+b) - \bar{I}) \cdot (P(k,l) - \bar{P})}{\sqrt{(\sum_{k,l} (I(k+a, l+b) - \bar{I})^2) \cdot (\sum_{k,l} (P(k,l) - \bar{P})^2)}} \quad (3.2)$$

donde $Dist_{CCN}(x,y)$ es el coeficiente de correlacion cruzada normalizado en el intervalo $[a,b]$ entre la imagen y el prototipo. $I(k,l)$ y $P(k,l)$ denotan la intensidad de la imagen y el prototipo respectivamente, \bar{I} y \bar{P} sus respectivas medias. Debido a que el denominador sirve como termino de normalización, este puede ir de -1 a 1. Valores positivos altos indican que la imagen y el prototipo son muy similares, un valor de 0 indica que imagen y prototipo son se correlacionan, mientras que valores negativos indican un contenido inverso.

El método de correlación cruzada normalizada tiene la ventaja de ser muy sencillo y fácil de implementar. Aunado a ello, el mismo procedimiento puede ser aplicado a cualquier tipo de objeto y no existe restricción alguna acerca de la apariencia del objeto.

Sin embargo, existen varias desventajas, primero el coeficiente de correlación decrece significativamente cuando el objeto contenido en la imagen de búsqueda es una versión rotada o escalada del prototipo, i.e. el método de correlación cruzada no es invariante a rotación y escalado. Segundo, el método solo es robusto con respecto a cambios de iluminación lineal, sin embargo cambios de iluminacion no lineal no son previstos por este método. Además, el método es sensible al desorden y la oclusión: como solo se calcula un valor de similitud global, es muy difícil distinguir si los valores bajos de $Dist_{CCN}$ se originan porque el objeto buscado no está presente en la imagen de búsqueda (pero tal vez un objeto bastante similar) o porque existen variaciones causadas por cambios de iluminación no lineal.

En otras palabras, la correlación cruzada normalizada no posee un gran poder discriminativo. Aunado a ello, la estrategia no es recomendable para tareas de clasificación donde el número de clases es muy grande, debido a que el proceso de búsqueda en la imagen debe ser repetido para cada clase, esto provoca tiempos de ejecución bastante altos.

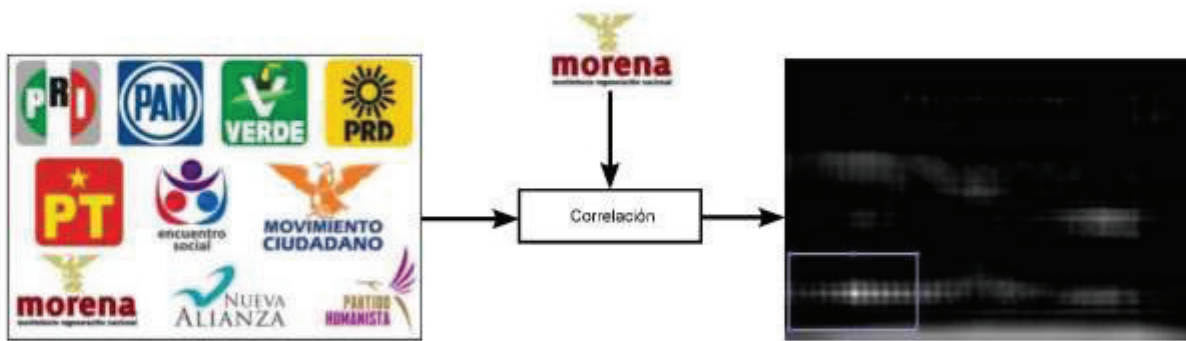


FIGURA 3.2: Cálculo de las diferencias entre la imagen de búsqueda y prototipo

3.1.3. Técnica piramidal

La detección de objetos en una imagen tiene muchas dificultades, desde una mala identificación hasta no poder realizarla debido a que los objetos no van a ser del mismo tamaño en todas las tomas, estos cambian y al obtener nosotros las ventanas canónicas para comparar con nuestro template muchas veces no se encuentra el objeto completo en la ventana o es muy pequeño a comparación de nuestro template, por lo cual los resultados de identificación son nulos o muy malos. Para ello nosotros utilizamos la técnica piramidal.

La técnica piramidal nos permite el manejo de ventanas canónicas a escala, de tal forma que la identificación de un objeto dentro de una imagen llega a ser más exacta y fácil de realizar. Esta técnica no realiza un re-escalado de las ventanas, lo que hace es re-escalar la imagen por este motivo recibe el nombre de técnica piramidal, ya que realiza un tipo de piramide con las imágenes a distintas escalas desde la más grande a una pequeña.

Para realizar un parámetro de re-escalado utilizaremos: s^i donde $i = 0$ para la imagen original.

s^i donde $i = 0$ para la imagen original; s es el que controla los niveles entre imágenes.

Se le asigna un valor a s . Que es el grado de re-escalada que se aplica a la imagen.

El tope del re-escalado de la imagen, se da cuando la ventana canónica es más grande que la imagen. En la base de la piramide se encuentra la imagen original y a la punta de la piramide esta la imagen más pequeña re-escalada y en cada una de estas se va deslizando la ventana canónica.

CUADRO 3.1: Tamaños de diferentes imágenes con factor de reescalada $s = 1,2$

s^i	i	Ancho	Ancho	Alto	Alto
$1,2^0 = 1$	0	600/1	600	420/1	420
$1,2^1 = 1,2$	1	600/1.2	500	420/1.2	350
$1,2^2 = 1,44$	2	600/1.44	417	420/1.44	292
$1,2^3 = 1,728$	3	600/1.728	347	420/1.728	243
$1,2^4 = 2,073$	4	600/2.073	289	420/2.073	202
$1,2^5 = 2,488$	5	600/2.488	241	420/2.488	169
$1,2^6 = 2,985$	6	600/2.985	201	420/2.985	141
$1,2^7 = 3,583$	7	600/3.583	167	420/3.583	117
$1,2^8 = 4,299$	8	600/4.299	139	420/4.299	98

La Tabla 3.1 nos muestra los resultados; tenemos una ventana canónica de 70x120. Por lo tanto en $i = 7$ aún no hay problema en lo ancho ya que es mayor que el valor de la ventana canónica pero por el contrario en lo alto es 3 puntos menor que la canónica; por lo cual, no nos sirve llegar hasta ese nivel de re-escalado. En nuestro ejemplo nuestra piramide tendrá 7 niveles y el último nivel de re-escalado será para $i = 6$.

Para concluir se realiza el barrido por toda la imagen de la ventana canónica en busca de identificar nuestro template. Al tener una fotografía, sabemos que hay objetos cercanos y otros lejanos, por lo tanto al momento de buscar en las imágenes de la piramide, encontraremos a los objetos más cercanos en la punta de la piramide, y los objetos mas lejanos se detectarán en la base de la piramide. No hemos de olvidar que el tamaño de la imagen canónica la determinamos nosotros y este tamaño es el que determinará el tamaño mínimo de los objetos que vayamos a detectar.

3.2. Algoritmo Viola-Jones

El algoritmo de detección de caras de Viola-Jones fue desarrollado debido a la necesidad de una detección rápida dentro de una amplia variedad de imágenes (Viola y Jones,2001). Esta misma estructura puede ser aplicada a la detección de objetos en general y también a tareas de seguimiento de objetos en imágenes.

Primero la imagen es mapeada a una nueva representación llamada imagen integral, que es introducida con el objetivo de permitir un rápido cálculo de características simples. El número de características que pueden ser calculadas es muy grande, incluso más grande que el número de píxeles en la imagen. Ya que el número de características

es mayor al necesario, es necesario identificar un subconjunto de las mejores características.

En la segunda parte del método, un algoritmo de aprendizaje basado en Adaboost es implementado para seleccionar un pequeño número de características muy discriminativas, esto nos permite obtener un conjunto eficiente de clasificadores. La selección de características es realizada utilizando un algoritmo AdaBoost modificado en los que un clasificador debil puede depender de una sola característica (Tieu y Viola, 2004).

En la tercera etapa, los clasificadores son ordenados en una secuencia de cascada iniciando con simples pero rápidos clasificadores utilizados para fácilmente rechazar hipótesis de detección de objetos, para obtener clasificadores más complejos y por lo tanto más poderosos. Esta estrategia mejora drásticamente la velocidad de detección de objetos.

El algoritmo Viola-Jones no funciona directamente con los datos de intensidades de la imagen. El algoritmo calcula y usa una imagen integral para calcular rápidamente las respuestas de los filtros simples basados en regiones a muchas escalas. Los filtros específicos dependen del objeto y se pueden usar para codificar el conocimiento específico del problema. Es importante destacar que estas características se pueden calcular en tiempo constante una vez que se forma la imagen integral. Para facilitar el cálculo rápido de las características del rectángulo, se construye la imagen integral. La imagen integral es una representación matricial que mantiene la información global de la imagen (Viola y Jones, 2001). Esta es construida de tal forma que sus valores $ii(i, j)$ en la posición (i, j) representan la suma de todos los valores que se encuentran arriba y a la izquierda del pixel (i, j) en la imagen original.

$$ii(i, j) = \sum_{k \leq i, l \leq j} f(k, l), \quad (3.3)$$

donde f es la imagen original, la imagen integral puede ser eficientemente calculada en una simple pasada usando recurrencias.

El principal uso de la imagen integral es agilizar el cálculo de las características de rectángulo en la imagen. Este tipo de características es utilizado para una rápida identificación de objetos. Como se muestra en la Figura 3.4 cualquier suma rectangular puede ser calculada usando cuatro referencias del arreglo. Luego, la diferencia entre dos rectángulos requiere ocho referencias.

3.2. Algoritmo Viola-Jones

Algorithm 2 Algoritmo de construcción de imagen integral

Require: Imagen de entrada

Ensure: Imagen integral $ii(i, j)$

- 1: Sea $s(i, j)$ la suma acumulativa de una fila, sea $s(i, -1) = 0$
- 2: Sea $ii(i, j)$ una imagen integral y $ii(-1, j) = 0$.
- 3: Usando una simple pasada fila por fila a través de la imagen, para cada pixel (i, j) en la imagen calcular la suma acumulativa de filas $s(i, j)$ y el valor de la imagen integra $ii(i, j)$ usando las recurrencias.

$$s(i, j) = s(i, j - 1) + f(i, j), \quad (3.4)$$

$$ii(i, j) = ii(i - 1, j) + s(i, j), \quad (3.5)$$

- 4: Después de alcanzar el pixel de la esquina inferior derecha de la imagen a través de una pasada en la imagen, la imagen integral es construida.
-

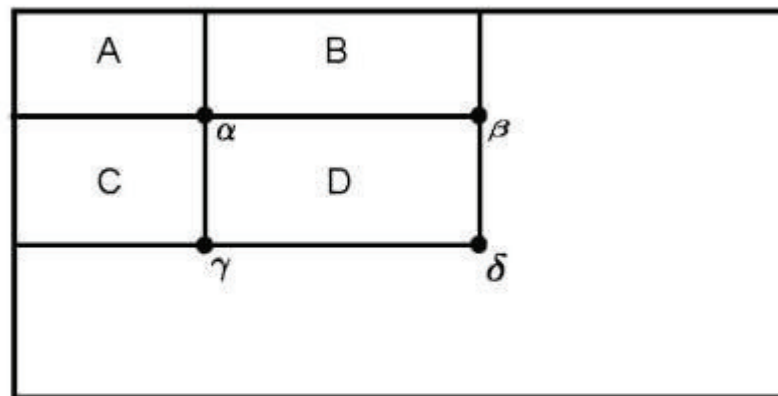


FIGURA 3.3: Cálculo de los rectángulos de una imagen integral

La determinación del rectángulo de características multi-escala provee características discriminativas que facilitan un aprendizaje efectivo. Teniendo este enorme conjunto de características disponible, es deseable que solo un pequeño número de de estas características sean utilizadas en una combinación para mantener un clasificador eficiente. Un pequeño conjunto de estas características puede ser seleccionado utilizando el algoritmo AdaBoost. Un rectángulo de características es inicialmente seleccionado usando un clasificador debil. Para cada característica seleccionada, el clasificador debil encuentra una frontera óptima minimizando el número de ejemplos mal clasificados en el conjunto de entrenamiento. Para cada clasificador debil basado en una sola característica f_j y una frontera t_j . El proceso completo del algoritmo de aprendizaje AdaBoost es mostrado en el algoritmo 3

Algorithm 3 Algoritmo de selección de características AdaBoost y aprendizaje del clasificador)

Require: Considerar un problema de clasificación binaria, un conjunto de entrenamiento con ejemplos positivos y negativos y sus identificadores de clase $U_i \in \{-1, 1\}$.

Ensure: $S(x_i)$

- 1: Inicializar K , el número de características a ser identificadas
- 2: Sea $k = 1$; para cada ejemplo x_i , inicializar pesos

$$w_{1,i} = \frac{1}{2q}, w_i = -1, \quad (3.6)$$

$$w_{1,i} = \frac{1}{2p}, w_i = 1 \quad (3.7)$$

- 3: Para $k \neq 1$, normalizar los pesos para producir una distribución de probabilidad

$$w_{k,i} := \frac{w_{k,i}}{\sum_{l=1}^{p+q} w_{k,l}} \quad (3.8)$$

- 4: Para cada característica f_j , entrenar un clasificador $h_{k,j}$ restringido a usar una sola característica. Evaluar su error de clasificación $\epsilon_{k,j}$ sobre el conjunto de entrenamiento considerando los pesos actuales $w_{k,i}$ asociados con cada ejemplo x_i

$$\epsilon_{k,j} = \frac{1}{2} \sum_i w_{k,i} |h_j(x_i - w_i)| \quad (3.9)$$

- 5: Seleccionar el clasificador $h_{k,j}$ con el error mas pequeño $\epsilon_{k,j}$
- 6: Actualizar los pesos para todos los ejemplos x_i

$$w_{k+1,i} = w_{k,i} \beta_k^{1-E_i}, \quad (3.10)$$

donde $\beta_k = \frac{\epsilon_{k,j}}{(1-\epsilon_{k,j})}$ y $E_i = 0$ si x_i es clasificado correctamente,

$E_i = 1$ de cualquier otra manera

- 7: hacer $k := k+1$
- 8: si $k \leq K$, regresar al paso 4
- 9: el clasificador mas fuerte $S(x_i)$ es definido como

$$S(x_i) = 1 \text{ para } \sum_{k=1}^K \alpha_k h_{k,j}(x_i) \geq \frac{1}{2} \sum_{k=1}^K \alpha_k, \quad (3.11)$$

$$S(x_i) = -1 \text{ de otra manera,} \quad (3.12)$$

donde $\alpha_k = \log\left(\frac{1}{\beta_k}\right)$ y j denota las características que son utilizadas en el k -ésimo clasificador debil $h_{k,j}$ respectivamente.

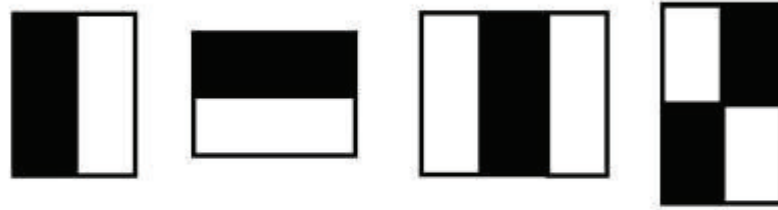


FIGURA 3.4: Características basadas en rectángulos

3.3. Detector de Harris

El detector de Harris es un detector invariante a cambios de iluminación e invariante a transformaciones afines. Este detector ha sido utilizado en diversas aplicaciones que van desde detección, reconocimiento y categorización de objetos (Dorko y Schmid, 2003) (Leibe et al, 2004), Reconocimiento en vídeo (Liu et al, 2009), Reconocimiento de modelos y seguimiento de objetos en movimiento (Dorko y Schmid, 2003) (Nowak et al, 2006).

El algoritmo fue diseñado para detectar esquinas. Una esquina puede definirse como un punto donde se intersectan dos o más bordes con direcciones diferentes. La Figura esquinas muestra algunos de los tipos de esquinas más comunes.



FIGURA 3.5: Tipos de esquinas más comunes

Un detector de bordes debe satisfacer algunos criterios para ser categorizado como un detector ideal, estas se enumeran a continuación:

1. Todas las esquinas deben ser detectadas
2. No deben detectarse falsas esquinas
3. Deben localizarse con exactitud
4. El detector debe poseer una buena estabilidad

5. El detector debe ser invariante a transformaciones geométricas y robusto
6. Debe ser computacionalmente barato

Los métodos de detección de esquinas basados en correlación son los métodos más utilizados. algunos métodos involucran derivadas para detectar la razón de cambio de la dirección del gradiente y la magnitud del gradiente. Los algoritmos de detección de esquinas normalmente utilizan tres pasos principales como lo muestra la Figura 4.8.

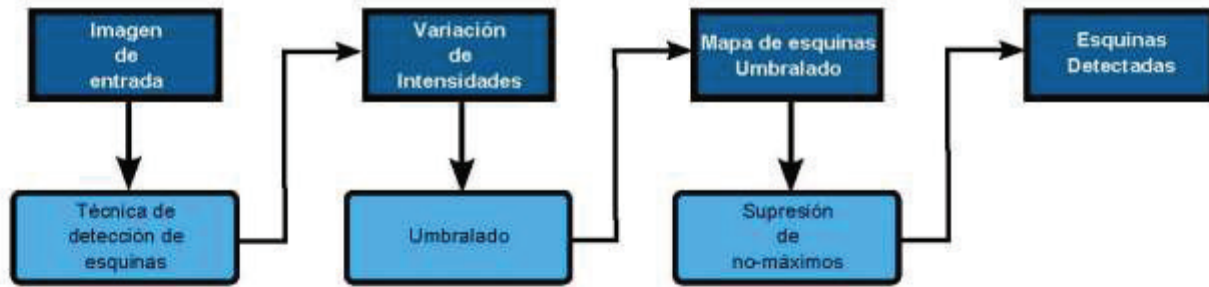


FIGURA 3.6: Pasos comunes de algoritmos de detección de esquinas

El algoritmo de detección de esquinas de Harris calcula la variación de la intensidad a través del gradiente de la imagen, ponderando cada pixel mediante un kernel Gaussiano de la siguiente manera:

$$V_{u,v}(x,y) = \sum_{i \in W} g_i \left(u \frac{\partial I}{\partial x} + v \frac{\partial I}{\partial y} \right)^2 \quad (3.13)$$

$$V_{u,v}(x,y) = \sum_{i \in W} g_i \left(\left(u \frac{\partial I}{\partial x} \right)^2 + 2uv \frac{\partial I_i}{\partial x} \frac{\partial I_i}{\partial y} + \left(v \frac{\partial I}{\partial y} \right)^2 \right) \quad (3.14)$$

$$V_{u,v}(x,y) = Au^2 * 2Cuv + Bv^2 \quad (3.15)$$

donde $A = g_\sigma * \left(\frac{\partial I}{\partial x} \right)^2$, $B = g_\sigma * \left(\frac{\partial I}{\partial y} \right)^2$, $C = g_\sigma * \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \right)$

La variación de la intensidad puede ser descrita como

$$V_{u,v}(x,y) = Au^2 * 2Cuv + Bv^2 \quad (3.16)$$

$$V_{u,v}(x,y) = \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \quad (3.17)$$

3.4. Detector de objetos SIFT

$$\text{donde } M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

Las variaciones de intensidad se pueden calcular como:

$$C(x, y) = \det(M) - k(\text{trace}(M))^2 \quad (3.18)$$

donde $\det(M) = AB - C^2$, $\text{trace}(M) = A + B$ y k es una constante

El algoritmo de detección de esquinas de Harris es descrito en el algoritmo 4

Algorithm 4 Algoritmo de detección de esquinas de Harris)

Require: Imagen en escala de grises $I(x, y)$, σ del kernel Gaussiano, valor de k y umbral T .

Ensure: Mapa que indica la posición de cada esquina

1: Para cad pixel (x, y) calcular la matriz de autocorrelación M donde

$$A = g_\sigma * \left(\frac{\partial I}{\partial x}\right)^2, B = g_\sigma * \left(\frac{\partial I}{\partial y}\right)^2, C = g_\sigma * \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y}\right)$$

$$\text{con: } \left(\frac{\partial I}{\partial x}\right) = I * h_1 * h_2^2 \text{ y } \left(\frac{\partial I}{\partial y}\right) = I * h_2 * h_1^2$$

donde: $h_1 = [0,0376950,2491530,4263750,2491530,037659]$ y $h_2 = [0,1096040,2766910,00000 - 0,276691 - 0,109604]$, y el tamaño del kernel Gaussiano es $6\sigma + 1$

2: Construir el mapa de variaciones para cada pixel (x, y) como:

$$C(x, y) = \det(M) - k(\text{trace}(M))^2 \quad (3.19)$$

3: Umbralizar el mapa de intensidades $C(x, y) \geq T$

4: Realizar supresión de no-máximos para encontrar máximos locales

3.4. Detector de objetos SIFT

El algoritmo Scale Invariant Feature Transform (SIFT) fue desarrollado por Lowe (Lowe, 1999). Este método es capaz de detectar y describir puntos característicos estables en una imagen. Originalmente, este algoritmo fue desarrollado para el reconocimiento de objetos en imágenes estáticas (Lowe, 1999), aunque sus aplicaciones han sido múltiples y numerosas desde entonces (Brown and Lowe, 2003) (Sivic y Zisserman, 2003) (Fergus et al, 2003) (Dorko y Schmid, 2003) (Leibe et al, 2004). Entre algunas de ellas se encuentran: reconocimiento de objetos con y sin oclusión (Lowe, 1999), categorización de objetos (Dorko y Schmid, 2003) (Leibe et al, 2004), modelado, reconocimiento y seguimiento en vídeo (Brown and Lowe, 2003), entre otras.

El algoritmo SIFT es capaz de encontrar características relevantes en una imagen, estas características son invariantes a la escala, rotación y a pequeñas variaciones en la pose en 3D.

Los puntos obtenidos son llamados puntos SIFT y son obtenidos mediante un complejo proceso de detección. La detección es llevada a cabo a través de filtros de diferencias de Gaussianos de espacio-escala. El espacio de escala de una imagen es una pirámide formada por todos los posibles reescalados de la imagen. El algoritmo estudia máximos o mínimos locales a partir de la diferencia de Gaussianos en la propia escala y las contiguas y toma el punto central y su vecindario como región detectada. Obtener la pirámide completa es un trabajo computacionalmente muy caro. sin embargo, es posible reducir el cálculo a sólo unos cuantos niveles discretizados, calculados con un operador de suavizado Gaussiano, doblando en cada paso el nivel de suavizado, consiguiendo así un escalado efectivo del 50%.

El algoritmo se compone de cuatro pasos principales:

1. Cálculo del espacio de escala. La primera etapa del algoritmo realiza una búsqueda sobre las diferentes escalas y dimensiones de la imagen identificando posibles puntos de interés. Esto se lleva a cabo mediante una serie de suavizados Gaussianos sobre la imagen original que se van agrupando en octavas. SIFT es utilizado en imágenes en escala de grises, el vector de características de 128 elementos que define cada píxel, contiene información sobre cómo se distribuyen los niveles de intensidad alrededor de cada punto de interés.
2. Localización de los puntos clave. En esta etapa, se busca localizar candidatos a puntos SIFT. Cada posible punto de una característica es encontrado como el punto de máxima respuesta (máximo o mínimo) de la diferencia de Gaussianas (DoG), comparándolo con sus vecinos en la imagen y con los puntos vecinos de las imágenes adyacentes del espacio de escala. El espacio-escala de la imagen original es la convolución de funciones Gaussianas 2D con distinto factor de escala (σ) con la imagen. Una vez realizada la comparación, es necesario Filtrar los puntos para eliminar los puntos de bajo contraste y los que se encuentran en bordes. La diferencia de Gaussianas produce una fuerte respuesta en torno a los bordes de la imagen. Al conjunto de las imágenes Gaussianas suavizadas junto con las imágenes DOG se le llama octava.

3. Asignación de la orientación. Para dotar a los puntos de interés de independencia a la rotación, se aplica un post-proceso, que consiste en calcular la mejor aproximación del vector gradiente a una de las direcciones discretizadas. Así, se compara la magnitud y dirección de dicho vector con unas cuentas direcciones discretizadas, habitualmente 36 direcciones con 10 grados de amplitud. Todas las operaciones posteriores son realizadas sobre los datos transformados según la orientación, escala y localización dentro de la imagen asignados en esta etapa.
4. Descriptor del Punto de Interés: La última etapa hace referencia a la representación de los puntos clave como una medida de los gradientes locales de la imagen en las proximidades de dichos puntos clave y respecto de una determinada escala.

Por último, y con el fin de fortalecer la estabilidad de los puntos obtenidos en este método, se eliminan aquellos puntos de bajo contraste o ubicación más sensible al ruido. Con el tiempo, este método ha pasado a incorporarse al algoritmo de SIFT como una etapa más pese a no encontrarse en el enunciado original.

La detección de los llamados puntos de interés o puntos que exhiben algún tipo de características importantes como esquinas. Después de ello, para cada punto de interés, un vector de características llamado descriptor de region es calculado. Cada descriptor de region caracteriza la información disponible de la imagen en un vecindario local alrededor de un punto de interés.

3.4.1. Detección de extremos en el espacio escala

La primera etapa tiene obtiene los puntos candidatos de la imagen que puedan ser identificados de forma repetida bajo diferentes vistas del mismo objeto.

El espacio-escala Gaussiano de una imagen $I_s(x; y; \sigma)$ es definido como la convolución de funciones 2D Gaussianas $G(x; y; \sigma)$ de diferentes valores σ con la imagen original $I(x; y)$, esto es:

$$I_s(x; y; \sigma) = G(x; y; \sigma) * I(x; y) \quad (3.20)$$

donde (x, y) son las coordenadas de la imagen I y σ representa el factor de escala, mientras que $G(x; y; \sigma)$ esta dado por:

$$\frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-(x^2+y^2)/2\sigma^2} \quad (3.21)$$

La función DoG es obtenida sustrayendo escalas posteriores de σ en cada octava

$$D(x; y; \sigma) = I_s(x; y; k\sigma) - I_s(x; y; \sigma) \quad (3.22)$$

donde k es una constante multiplicativa del factor de escala, comúnmente el rango de los valores de k es fijado entre 1.1 y 1.4. El espacio de escala puede ser explorado mediante las variaciones de σ . Máximos y mínimos locales son detectados comparando $D(x; y; \sigma)$ en el punto $(x; y; \sigma)$ con sus ocho vecinos de la misma escala σ y las regiones 3×3 de los dos vecinos centrados en la posición (x, y)

El conjunto de las octavas es construido mediante el muestreo sucesivo de la imagen original por un factor de 2. Cada una de las octavas es dividida en un número entero de subniveles o escalas s .

Los máximos y mínimos de las imágenes diferencia-de-gaussiana son detectados mediante la comparación de un pixel con sus 26 vecinos en las regiones de 3×3 de las escalas actual y anterior y posterior.

3.4.2. Localización de puntos clave estables

La segunda etapa se obtiene los puntos estables. Los puntos que son vulnerables al ruido y que podrían no ser detectados bajo pequeños cambios de iluminación son excluidos utilizando los siguientes criterios:

1. Se aplica un proceso de umbralización para eliminar los puntos con bajo contraste. Los puntos con valores menores al umbral D son excluidos. Lowe recomienda el valor de $D = 0,03$
2. Los puntos situados sobre bordes, tienen demasiada inestabilidad ante pequeños ruidos. Para eliminarlos, se calcula la matriz del Hessiano sobre la localización y escala del punto en estudio:

$$H = \begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial x \partial y} \\ \frac{\partial^2 D}{\partial x \partial y} & \frac{\partial^2 D}{\partial y^2} \end{bmatrix} \quad (3.23)$$

donde D es la imagen DoG $D(x; y; \sigma)$ respecto de la escala s . Las derivadas se calculan mediante la resta del valor de los puntos vecinos. Se puede demostrar que la siguiente desigualdad permite la localización de los puntos en los bordes:

Aquellos puntos que no satisfagan dicha desigualdad son descartados. El valor fijado es de $r = 10$. Tras descartar los puntos inestables, al resto de puntos clave se les asigna una orientación.

3.4.3. Asignación de la orientación

La técnica SIFT permite obtener puntos invariantes a rotación.

Esta invarianza se consigue asignando a cada uno de los puntos una orientación basada en las propiedades locales de la imagen y representando el descriptor respecto de esta orientación. Para cada punto de interés se calcula la magnitud del gradiente, m , y su orientación, θ , de la siguiente manera:

$$m(x, y) = \sqrt{(I_s(x+1, y) - I_s(x-1, y))^2 + (I_s(x, y+1) - I_s(x, y-1))^2} \quad (3.24)$$

$$\theta(x, y) = \arctan \frac{I_s(x, y+1) - I_s(x, y-1)}{I_s(x+1, y) - I_s(x-1, y)} \quad (3.25)$$

donde I_s representa la imagen gaussiana suavizada cuya escala resulta más próxima a la escala del punto de interés actual.

Para la orientación del gradiente, se crea un histograma con 36 compartimentos, cada uno de ellos con una longitud de 10 para cubrir el rango de los 360 posibles. El compartimento cuyo valor es mas alto se corresponde con la dirección dominante del gradiente y por lo tanto es elegido como orientación dominante. Sin embargo se ha de tener en cuenta la posibilidad de que exista más de una dirección dominante. Es por ello que cualquier compartimento con un valor de más del 80% del valor de la magnitud principal se considerará también como dirección dominante.

Los puntos que contengan más de una dirección dominante poseen mayor estabilidad.

3.4.4. Descriptor del punto de interés

Esta etapa obtiene un vector de características para cada punto de interés. Para evitar cambios bruscos en el descriptor ante pequeños cambios en la posición de la ventana se realiza un muestreo de las orientaciones y magnitudes del gradiente de la imagen sobre regiones de 16×16 alrededor del punto de interés.

Se analizan las muestras de cada región de 16×16 formando histogramas de orientaciones resumiendo el contenido en sub-regiones de 4×4 . Cada uno de los histogramas se compone de 8 compartimentos, que almacenan las orientaciones posibles proporcionales a 45 donde la magnitud de cada flecha representa el valor acumulado para cada bin. Por lo tanto se obtienen 16 histogramas respecto de las orientaciones de los puntos de cada región para cada uno de los puntos de interés.

Finalmente el descriptor de cada punto de interés está formado por un vector que contiene los valores de las 8 orientaciones de los 4×4 histogramas componiendo un vector de características de $4 \times 4 \times 8 = 128$ elementos.

Por último, se normaliza la magnitud del gradiente del vector de características y se limita el valor de cada componente a un valor máximo, esto permite hacerlo robusto frente a los cambios de iluminación

3.5. Detector de objetos SURF

El descriptor SURF (Speeded-Up Robust Features) fue desarrollado por Herbert Bay et al. (Bay et al, 2006) como un detector de puntos de interés. El detector SURF es robusto a transformaciones comunes de la imagen como rotación, desenfoque, cambio de escala y cambios de iluminación. El descriptor SURF presenta una gran similitud el descriptor de puntos SIFT. Sin embargo, este presenta las dos mejoras siguientes:

1. Menor complejidad computacional del algoritmo, lo que se traduce en una obtención de los puntos más agil.
2. Una técnica más robusta a transformaciones de la imagen

Debido a su menor complejidad computacional, el tiempo de procesamiento es reducido, esto permite que se utilice en aplicaciones relacionadas con video (Domke et al, 2006) (Huang et al, 1997).

Las etapas que comprende el descriptor SURF pueden ser resumidas como sigue

1. La normalización o longitud de los vectores de características de los puntos de interés es considerablemente menor. Pues emplea vectores con una dimensionalidad de 64, la mitad de la longitud del descriptor SIFT.
2. El descriptor SURF utiliza siempre la misma imagen, la original.

3. Utiliza el determinante de la matriz Hessiana para calcular la posición y la escala de los puntos de interés

Esta etapa es análoga a la del descriptor SIFT en la detección de puntos de interés. Sin embargo, SURF no utiliza diferentes medidas para el cálculo de la posición y la escala de los puntos de interés individualmente, sino que utiliza el valor del determinante de la matriz Hessiana en ambos casos. Dado un punto $p = (x, y)$ de la imagen I , la matrix Hessiana $H(p, \sigma)$ del punto p se define como:

$$H(p, \sigma) = \begin{bmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{xy}(p, \sigma) & L_{yy}(p, \sigma) \end{bmatrix} \quad (3.26)$$

donde $L_{xx}(p, \sigma)$ representa la convolución de la derivada parcial de segundo orden de la Gaussiana $\frac{\partial^2}{\partial x^2}g(\sigma)$ con la imagen I en el punto p .

En la etapa de descripción se amplía la capacidad de descripción de los puntos de interés y se agrega la información del color, información con la que no cuenta el descriptor SIFT. Para ello se proponen las técnicas de descriptores basados en histogramas donde se concatena los histogramas de los diferentes canales y la otra es utilizar descriptores basados en descriptores sobre imágenes de intensidad donde se concatena la salida de un descriptor de forma aplicado a cada canal.

El descriptor SURF utiliza la imagen en su tamaño original, utilizando filtros de la segunda derivada pero cambiando el tamaño del filtro en piramide, el costo será el mismo sin importar el tamaño del filtro.

En la etapa de descripción SURF contempla tres fases: escala, orientación y descripción. Las cuales nos permite tener un vector que describa los cambios de cada punto de interés.

En escala se basa en analizar para cada punto un conjunto de vecinos de 3×3 o 4×4 , para la orientación se utilizan filtros de Haar en dirección de x e y ; además del uso de imágenes integrales donde podemos obtener la suma de un solo espacio variable en toda la imagen.

Pasos para Orientación:

1. Calcular la respuesta a filtros de Haar
2. Se selecciona un vecindaje circular de tamaño de 6 vecinos.
3. Sumamos las respuestas en un arco de 60 grados en desplazamiento, donde se obtiene un vector de dirección, se asigna la dirección del vector de mayor magnitud.

Pasos de Descripción:

1. Centramos una región cuadrada en el punto de interés la cual subdividimos en 16 regiones
2. Se toma cada una de las regiones y se coloca una malla cada 5x5 píxeles y recalculamos los filtros de Haar (h_x y h_y).
3. Se suman las magnitudes del filtro de Haar en verticales y en horizontales
4. Se tendrá un vector que es la concatenación de las $\sum h_x, \sum h_y, \sum |h_x|, \sum |h_y|$ será un vector de 64 valores.

Capítulo 4

Metodología

El uso de la tecnología para la solución de problemas se ha caracterizado por ser una vertiente que mejora y realiza diversas tareas, tanto en áreas de investigación como en cotidianas por lo que el uso de los sistemas y técnicas de procesamiento de imágenes y reconocimiento de patrones son una solución a los problemas. Aunado a esto los medios de comunicación han aumentado el uso de cámaras web, cámaras fotográficas y uso de dispositivos móviles, por lo cual el uso de videos y fotografías en los sistemas de reconocimiento de patrones sigue en evolución.

Se menciona cada día más el uso de reconocimiento por Iris, huella dactilar para el desbloqueo móvil (seguridad), reconocimiento de personas en aeropuertos (seguridad), uso de sistemas para el campo agrícola tanto en investigación (identificación de enfermedades en los árboles, plagas, etc.) como en seguridad (para los espacios protegidos), en área médica (identificación de tumores por cámara en tiempo real), en la política (uso de sistemas para la seguridad de los votos, en esta tesis se realizó un sistema para el reconocimiento de logos de los partidos políticos para un control en los spots). El desarrollo de estos sistemas se plantean para investigación y campo de producción en las empresas.

En este Capítulo se muestran la metodología llevada a cabo en los experimentos. En los experimentos llevados a cabo se trabajó con diferentes técnicas de identificación automática de objetos, cada uno de los procesos se detalla en este Capítulo.

La recolección de los videos para nuestras pruebas, fue en realidad un primer paso en el desarrollo de los sistemas de detección de objetos. Los logos fueron tomados de los comerciales en televisión de cada partido político (temporada 2017B).

Las técnicas utilizadas son las siguientes:

1. Detector de objetos simple y piramidal
2. Detector de objetos de Viola-Jones

3. Detector de objetos SIFT
4. Detector de objetos SURF

4.1. Detector de objetos simple y piramidal

El detector de objetos simple es el enfoque más sencillo (ver Figura 4.1). Solo es necesaria una plantilla o prototipo del objeto de búsqueda y la imagen donde se buscará dicho objeto.

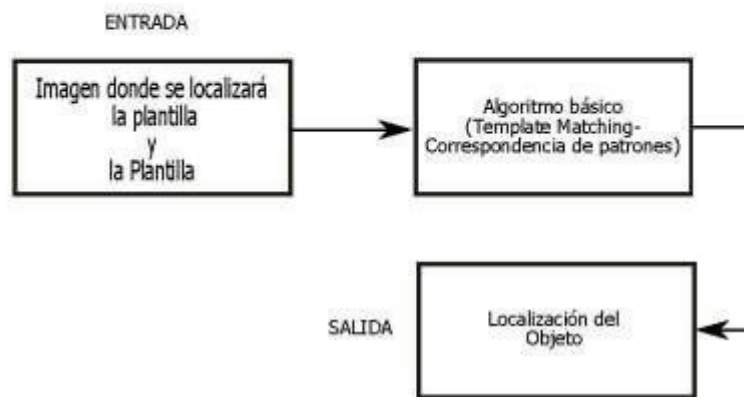


FIGURA 4.1: Entrada y salida del detector básico

La técnica utiliza una imagen prototipo del objeto a localizar y desliza el prototipo sobre la imagen de búsqueda cuadro a cuadro hasta pasarla por toda la imagen. En este método, la idea básica es ir comparando los píxeles en el prototipo con la imagen de búsqueda. La técnica mide las diferencias de cada píxel entre la imagen de búsqueda y el prototipo. La tarea de reconocimiento es encontrar la posición precisa del objeto en una imagen. La tarea es llevada a cabo moviendo la plantilla píxel a píxel para cada posible posición en imagen. Existen diferentes técnicas para medir esta diferencia. El modelo propuesto consiste de siete pasos generales. En la Figura 4.2 se muestra de una forma más puntual el diagrama de bloques del método utilizado. En cada apartado siguiente se explica cada apartado de la metodología propuesta.



FIGURA 4.2: Pasos comunes de algoritmo de detección de objetos simple

4.1.1. Seleccionar plantilla

En el primer paso, la Figura 4.2 muestra como es necesario seleccionar una plantilla o prototipo. Debido a que en algunas ocasiones, como sucede nuestro caso de aplicación, los logos cambian su diseño y en algunas ocasiones tambien su color con el tiempo, es indispensable seleccionar una plantilla adecuada para que el margen de error sea mínimo.



FIGURA 4.3: Logos con diferentes cambios en diseño y color

La Figura 4.3 muestra un ejemplo claro de las variaciones en el objeto a detectar. La Figura muestra logos del mismo partido político pero con cambios en su diseño, forma y color. Ya que esto sucede a menudo, aún cuando exista un patrón, escoger una plantilla inadecuada podría llevar a tener un pobre desempeño en la detección del objeto deseado. No obstante, en nuestro caso, los cambios no son substanciales, son cambios muy pequeños en el color y forma del objeto que no inciden en el desempeño del detector.

4.1.2. Pre-procesamiento

En el procesamiento de imágenes se manejan técnicas para mejora de imágenes desde eliminar ruido, borrosidad hasta cambios en brillo, suavizado, realzado, tonalidades de color, etc. Dado que el proyecto se trataba de identificar el logo en las imágenes (video), los logos de los partidos políticos no son cambiantes, los colores y la simbología es la misma, por lo tanto no fue necesario aplicar alguna mejora de imagen a los logos. Para las imágenes en donde se realiza la búsqueda de los logos. Tampoco fue necesario el uso de técnicas para la mejora de imagen, dado que estas imágenes son los frames que componen el video. La calidad de las imágenes de los videos son dadas por la resolución de la cámara, en nuestro proyecto utilizamos una resolución de 480×640 píxeles por lo que nuestros resultados fueron exitosos sin necesidad de aplicar estos tipos de preprocesamiento.

Sin embargo, se utilizaron técnicas para convertir las imágenes de color en imágenes en escala de grises tanto en las imágenes del video como en los logos; las técnicas que se utilizaron fueron promedio, tonalidad y luminosidad. La técnica de promedio es la más utilizada, no tiene gran costo computacional, pero al realizar las pruebas se determinó que la técnica de luminosidad es más eficaz para nuestro proyecto, por lo tanto utilizamos la técnica de luminosidad para nuestras pruebas posteriores.

4.1. Detector de objetos simple y piramidal

Otra técnica de pre-procesamiento utilizada fue el reescalado de las imágenes. Cada logo fué extraído de internet con distintas resoluciones para utilizarlo como plantilla o prototipo y realizar las pruebas de identificación. Se determinó que las mejores resoluciones de los logos para la búsqueda en imágenes entre 80 y 180 píxeles de ancho y alto (no se da un tamaño exacto debido a que los logos son diferentes para cada partido político). Ya que la mayoría de los partidos políticos utilizan logos completamente cuadrados, estos pueden ser utilizados con resoluciones de 80×80 , sin embargo algunos de ellos utilizan medidas diferentes a los demás y estos deben ser reescalados a otras resoluciones. La Tabla 4.1 muestra los tamaños estandar que se utilizaron para cada logo de cada partido político. Por otro lado, las imágenes o frames obtenidos para buscar el logo en ellas fueron tomadas con una cámara de video con resolución 480×640 (480 filas con 640 columnas) para cámaras que obtenían imágenes con resoluciones muy grandes, se reescalaron a imágenes de tamaño 480×640 .

CUADRO 4.1: Resolución de diferentes platillas utilizadas

Partido	Ancho	Alto
Morena	180	45
PAN	80	80
PRI	80	80
PRD	80	80
PT	80	80
Verde	80	80
Movimiento ciudadano	80	80
Nueva alianza	80	80
Encuentro social	126	100

4.1.3. Búsqueda de Candidatos y Detección

La búsqueda de candidatos, se refiere a las posibles ventanas que tienen píxeles que coinciden con la plantilla (logo). En la siguiente imagen se explica como se visualizan los posibles candidatos.

El logo que se toma de ejemplo es el de encuentro social; como se aprecia en la imagen, tenemos ventanas de colores, en el logo de encuentro social lo encierra una ventana de color morado, las ventanas de los posibles candidatos son los que tienen partes del logo o que por alguna razón el coeficiente de correlación es muy parecido, como serían la ventana azul, roja, verde claro y la ventana amarilla.



FIGURA 4.4: Posibles Candidatos

Una ventana del tamaño de nuestro prototipo la iremos deslizando por las filas y columnas de la imagen, esto es, por las coordenadas x e y con un desplazamiento de 5 píxeles. Esto es, no vamos buscando píxel a píxel, sino que buscamos una ventana al inicio y damos un salto de 5 píxeles en la coordenada x , buscamos otra ventana y damos otro salto hasta llegar al final de la fila. Una vez terminada la primera fila damos un salto de 5 píxeles en la coordenada y para seguir la búsqueda y realizamos este proceso hasta cubrir la búsqueda en toda la imagen.

Dar saltos en las coordenadas x e y de 1 píxel en lugar de 5 causa problemas serios en la complejidad del algoritmo, haciendo más tardada la búsqueda e imposibilitándolo para dar resultados en tiempo real. De forma general, tenemos dos maneras de estimar la semejanza entre el template (plantilla) y cada ventana de la imagen, tenemos el cálculo de las diferencias (Disimilitud) y el cálculo de similitudes (Similitud); Cada que deslizamos la ventana sobre la imagen aplicamos una de las dos técnicas. En este proyecto nos inclinamos por la técnica de similitud (correlación cruzada normalizada), esta técnica trabaja revisando los puntos de coincidencia entre la plantilla y la imagen, cuando son iguales la similitud da un valor de 255, cuando la similitud es completamente baja el valor es 0. Se obtiene la similitud de cada ventana, las similitudes más altas son las que tomaremos en cuenta y serán nuestros posibles candidatos, después

4.1. Detector de objetos simple y piramidal

revisamos los candidatos y realizamos una refinación obteniendo solo los candidatos más similares a la plantilla y obtenemos la detección del logo.

La Figura 4.8 muestra dos ejemplos de la forma como se busca en la imagen el prototipo. La Figura muestra como se desliza la búsqueda en las coordenadas x e y .

Ventanas Deslizantes



FIGURA 4.5: Detección de objetos por el método básico

Algoritmo Piramidal

La técnica definida en la subsección anterior tenía un margen de error alto, ya que si nuestro logo no era del mismo tamaño exacto al de la imagen, entonces el logo no sería encontrado, ya que no habría coincidencia por lo que fué necesario implementar un algoritmo piramidal para tener resultados eficientes y evitar problemas en la detección del objeto que puede presentarse en múltiples escalas.

El algoritmo piramidal es más eficiente que el detector básico, el desarrollo de este algoritmo es muy parecido al detector básico sin embargo tiene la ventaja de trabajar a diferentes escalas, por lo que los resultados son más exitosos.

La Figura 4.6 contiene el diagrama que muestra el proceso que sigue este algoritmo de detección:

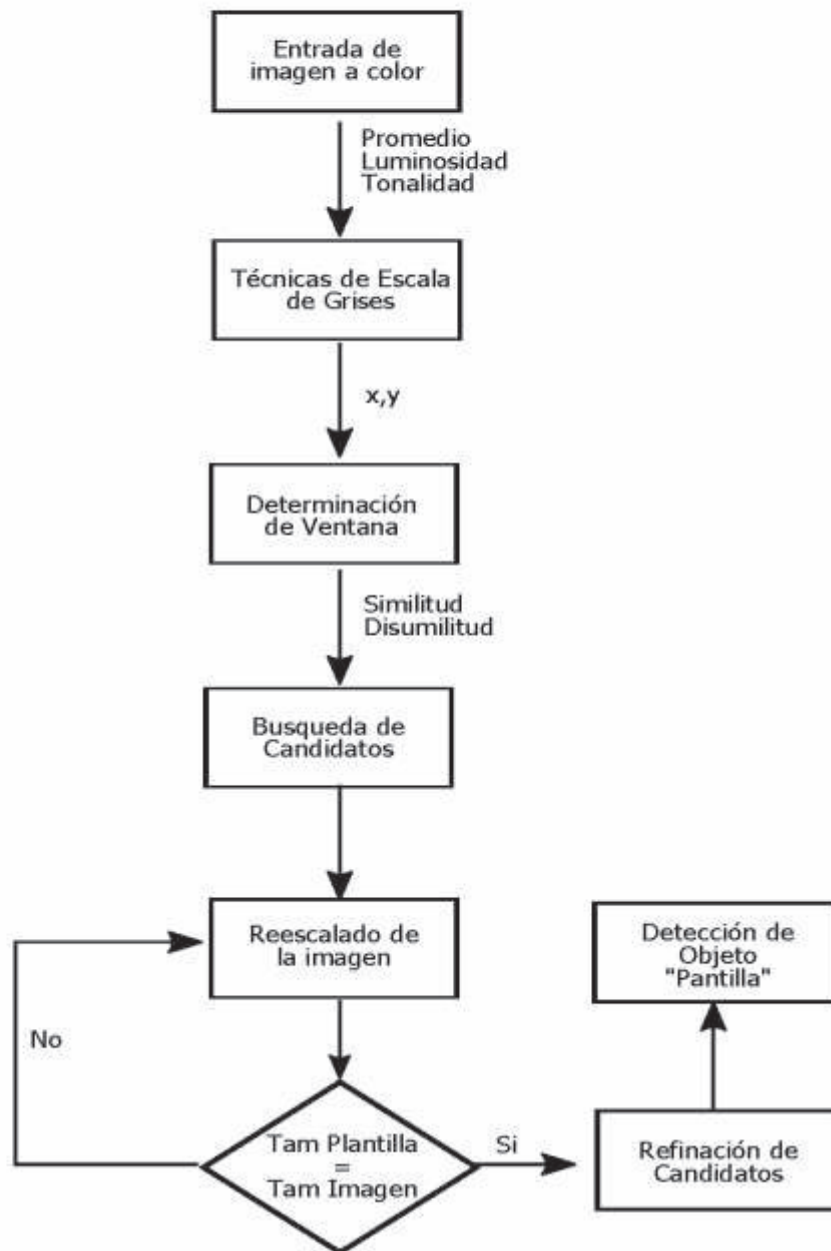


FIGURA 4.6: Algoritmo Piramidal

En una imagen tenemos varios objetos algunos de nuestro interés y otros sin importancia, estos objetos pueden aparecer a distintas distancias por lo que sus tamaños varían de acuerdo a la cercanía y lejanía de este, por lo que el algoritmo anterior solo encontraba los objetos que fueran del tamaño de la ventana, debido a esto el uso del reescalado es de mucha utilidad ya que nuestra imagen es la que cambiará de tamaño, de su tamaño original se realizará el primer barrido en busca de la coincidencia de nuestra plantilla y se irá reduciendo su tamaño y se realizará nuevamente nuestro

4.1. Detector de objetos simple y piramidal

barrido en busca del logo, el tamaño más pequeño en el que nuestra imagen se podra reescalar sera el tamaño de nuestra plantilla. La Figura 4.7 muestra un ejemplo de cómo se va reescalando la imagen original.

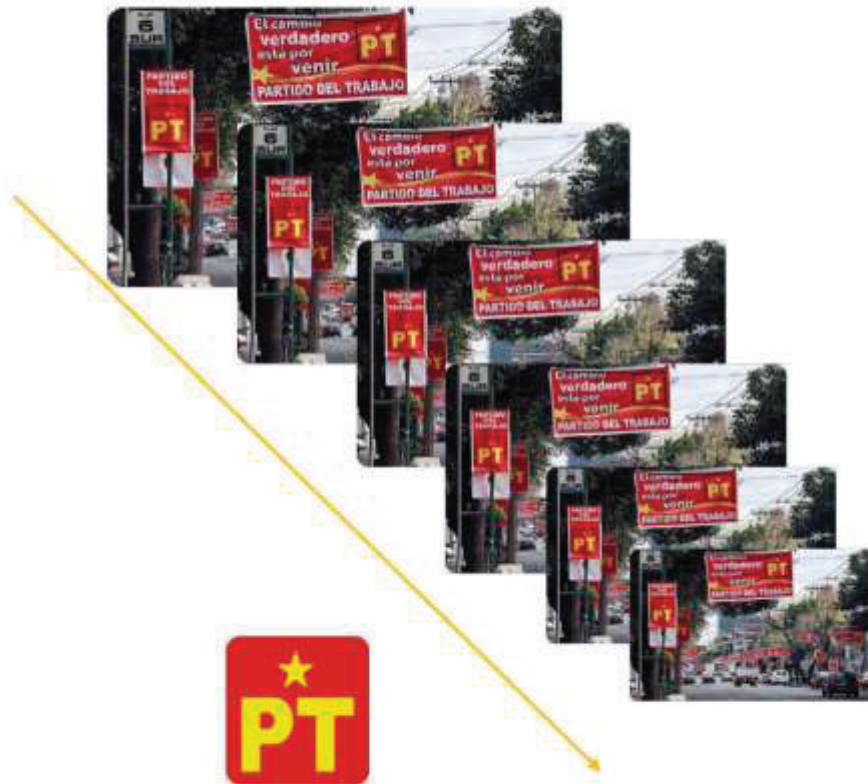


FIGURA 4.7: técnica piramidal

En cada reescalado, se vuelve a realizar la búsqueda de candidatos, realizamos el barrido con nuestra ventana sobre la imagen a diferente escala, cuando tenemos listo todos los posibles candidatos realizamos la refinación de los candidatos hasta llegar a la detección de nuestro logo.

4.1.4. Refinación de Candidatos

En esta parte el algoritmo elimina una gran cantidad de candidatos que se repiten. La Figura 4.8 muestra un conjunto de ventanas obtenidas después de que el algoritmo de correlación cruzada pasa por la imagen. Aquellas subimágenes cecanas al logo presentan una similaridad muy grande, sin embargo, esto es debido a que se encuentran muy cerca del logo. Es necesario eliminar todas aquellas ventanas que se encuentran

muy cerca del logo y obtener solo la que tiene una similitud muy grande al logo, esto es refinar la búsqueda de candidatos.



FIGURA 4.8: Refinación de candidatos

La Figura 4.9 muestra los resultados de la implementación del algoritmo básico en varias imágenes de prueba.



FIGURA 4.9: Resultados de detección de logo utilizando el algoritmo básico

4.2. Detector utilizando Viola Jones

El algoritmo Viola-Jones fué originalmente diseñado para detectar rostros en imágenes. El algoritmo obtiene una eficiencia bastante buena en la mayoría de las pruebas a las que ha sido sometido. En esta tesis, se utilizó el algoritmo para detección de logos.

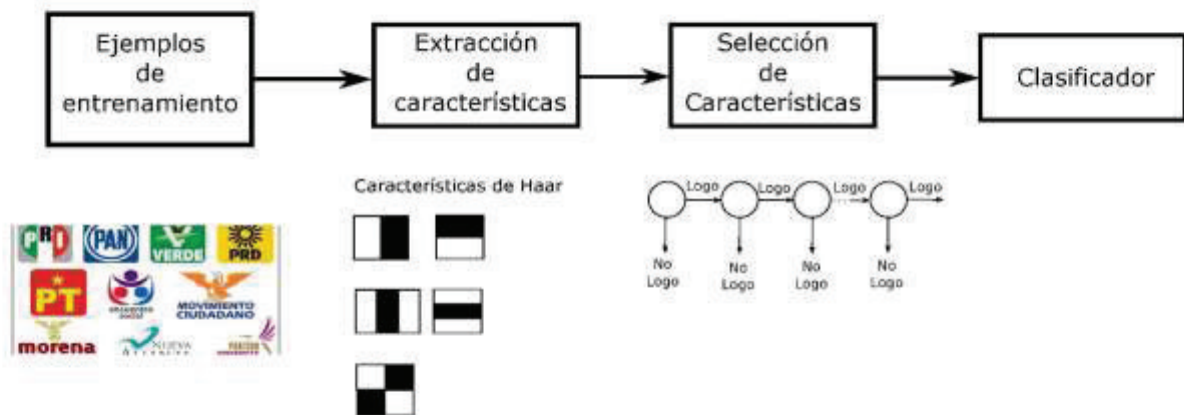


FIGURA 4.10: Detección de objetos utilizando Viola-Jones

La Figura 4.10 muestra el algoritmo general utilizado en la metodología. Los pasos del algoritmo se enumeran de la siguiente manera:

4.2.1. Ejemplos de entrenamiento

En esta parte se obtuvieron diversas imágenes de internet de los distintos logos. Las imágenes obtenidas fueron variaciones de logos de un mismo partido político y se entrenó individualmente cada logo, esto es, se obtuvieron alrededor de 40 imágenes de un logo de un partido político (imágenes positivas) y el mismo número de imágenes que no pertenecen al logo (imágenes negativas). Las variaciones de los logos fueron modificaciones en el color y diseño, pero también pequeñas modificaciones en la rotación de los logos.

4.2.2. Extracción de características

En esta parte el algoritmo extrae características utilizando la técnica de imagen integral descrita en el algoritmo 2 del Capítulo 3 .

4.2.3. Selección de características

Utilizando el algoritmo 3 visto en el Capítulo 3. Eliminamos características que afectan el desempeño del clasificador o que no aportan información al clasificador. el método basado en características de tipo Haar selecciona una gran cantidad de clasificadores débiles de un conjunto significativamente mayor de candidatos de clasificadores débiles para construir un clasificador fuerte. El clasificador final fuerte se puede describir como:

$$S(x) = h_{13}^1(x) + h_7^2(x) + h_{108}^3(x) + \dots + h_{83}^T(x) \quad (4.1)$$

donde un superíndice significa el número de secuencia del clasificador débil seleccionado y un subíndice se refiere al índice de candidatos. Este método requiere la selección de un gran número de clasificadores débiles, T , incluso del grupo de candidatos significativamente mayor. Debido a que el proceso de selección es independiente, el método requiere un gran número de operaciones computacionales.

4.2.4. Clasificador

El clasificador más fuerte es obtenido utilizando el algoritmo 3 descrito en el Capítulo 3.

4.3. Detector SIFT

El detector de objetos SIFT es una técnica que nos permite detectar una imagen mediante características locales, para después describirlos mediante un histograma de gradientes orientados. Una característica local es una propiedad que describe las propiedades de un pixel con relación a sus pixeles vecinos, proporcionando información relevante sobre partes de la imagen como regiones o contornos. Esto permite identificar la imagen aún cuando se presentan orientaciones, posiciones y escalas diferentes. De forma general el algoritmo puede ser descrito en la Figura 4.11. Las tres fases intermedias (detección, descripción y representación de la imagen) son un tanto complejas y estas han sido ya descritas en detalle en el Capítulo 3. En esta Sección solo se describen los parámetros utilizados en la implementación.



FIGURA 4.11: Etapas del algoritmo SIFT

4.3.1. Detección

Este primer paso se realiza encontrando puntos de interés de contornos y regiones, para ello se aplica un filtro gaussiano con una desviación estandar diferente a la misma imagen, el resultado serán imágenes procedentes de la imagen original con distintos grados de suavidad. Al comparar las imágenes gaussianas podemos identificar puntos de interés donde detectamos contornos y regiones.

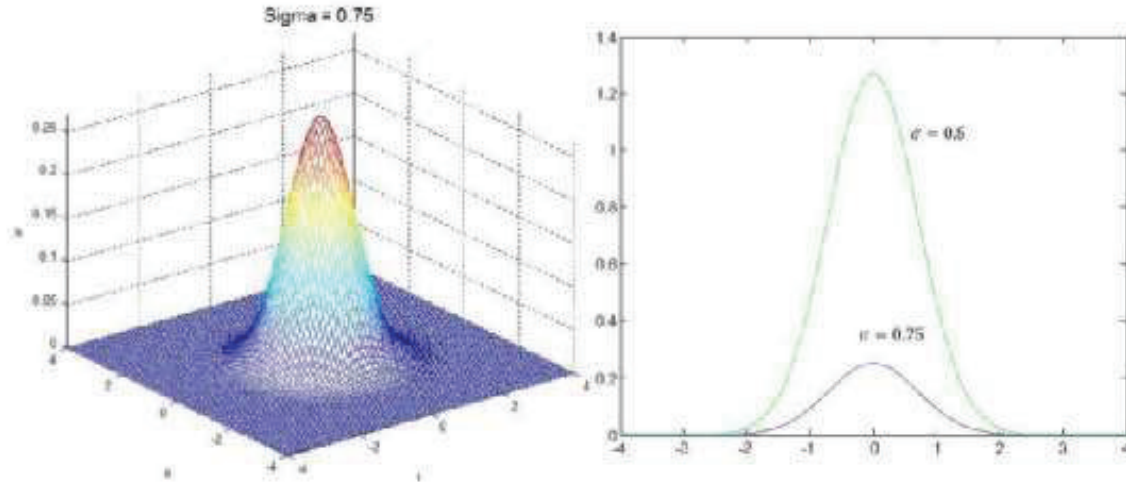
En el caso del filtro Gaussiano, este depende de la desviación estandar del valor de sigma, ya que sigma determina el nivel de suavidad que se aplicará, para un sigma más grande el suavizado es mayor. Un filtro se puede aplicar en una imagen utilizando el método de convolución con la función gaussiana; para ello se explicará primero la convolución:

La convolución es una operación que se aplica en dos funciones, f y g . Esto produce una nueva función que da el grado de solapamiento de la función g sobre la función f .

$$(f * g)(x, y) = \sum_{x', y'} g(x', y') \cdot f(x - x', y - y') \quad (4.2)$$

donde f representa la señal de entrada y g función básica.

En la función Gaussiana, el valor de los pixeles disminuyen cuando en la función nos alejamos del centro (ver Figura 4.12), esto indica que el punto central pondera con sus pixeles vecinos, dando esto un nuevo pixel que es una mezcla de los pixeles vecinos en la ubicación del pixel central, cuando obtenemos la imagen con el filtro gaussiano observamos que los pixeles pierden valor y la imagen se suaviza.

FIGURA 4.12: Influencia de σ en la dispersión del filtro

La convolución o filtro Gaussiano se utiliza para cambiar el aspecto global de una imagen, suavizando los contornos o agudizándolos; utilizamos el filtro Gaussiano para extraer las características locales de una imagen y utilizamos a f como la imagen original y a g como el filtro que utilizaremos para obtener la característica local deseada. Obtendremos el solapamiento entre un punto de la imagen y un punto. Sustituyendo valores para realizar la convolución de nuestra imagen queda así:

$$c(x, y) = I(x, y) * \text{filtro}(x, y) \quad (4.3)$$

$$c(x, y) = \sum_{x', y'} \text{filtro}(x', y') \cdot I(x - x', y - y') \quad (4.4)$$

Encontrar la diferencia de imágenes modificadas con filtros Gaussianos solo nos permite encontrar y detectar los cambios de intensidad entre diferentes imágenes suavizadas con filtros Gaussianos con diferentes valores de σ . Para encontrar los puntos de interés es necesario encontrar los valores máximos y mínimos de la diferencia de dos imágenes de filtro Gaussiano. Para que el método SIFT pueda detectar los objetos bajo diferentes escalas, SIFT utiliza el método piramidal al igual que el método de detección de objetos básico visto anteriormente. El método piramidal permite generar una secuencia de imágenes de menor tamaño y en cada una repite el procedimiento. En la base de la piramide se puede detectar puntos de interés pequeños y en la cima de la piramide se pueden detectar mejor los puntos de interés grandes.

4.3.2. Descripción

En la detección se obtuvieron los puntos de interés, que son las zonas relevantes de la imagen. Cada punto de interés tiene asociado el nivel de la pirámide y la escala en donde se detectó por lo tanto ya podemos rastrear los puntos sin importar el tamaño de la imagen.

En general, para obtener la descripción de cada característica local es necesario:

1. Obtener la magnitud y la orientación del gradiente en una ventana de 16×16 ,
2. Calcular un histograma de 8 bins con las orientaciones del gradiente para cada una de las 16 regiones
3. Normalizar el vector de 128 elementos después de concatenar los histogramas de 8 valores para cada uno de las 16 divisiones realizadas de la región

En el paso de la descripción, se debe seleccionar la región de interés o punto, para cada punto se debe determinar la escala del nivel de la pirámide y el valor de sigma en el suavizado; una vez identificado el punto de interés y determinada su información se realizará una ventana de tamaño fijo en todas las imágenes de la pirámide. Se tienen los niveles bajos de la pirámide que son las regiones pequeñas en la imagen original y los niveles altos de la pirámide son las regiones grandes en la imagen original.

Para obtener el descriptor es necesario obtener el gradiente de cada ventana obtenida por las características locales. El gradiente es el cambio de la intensidad de la imagen a una determinada dirección para cada píxel, donde la dirección es el cambio de intensidad con un valor máximo. El gradiente se caracteriza por la dirección (horizontal, vertical o diagonal) donde va el cambio de intensidad máximo y la magnitud del cambio en la dirección de máxima variación.

Para calcular el gradiente: Diferencia de intensidad de los píxeles en dirección horizontal y vertical.

$$dx = I(x + 1, y) - I(x - 1, y) \quad (4.5)$$

$$dy = I(x, y + 1) - I(x, y - 1) \quad (4.6)$$

Para calcular la dirección y la magnitud del gradiente se utilizan estas formulas:

$$\theta(x, y) = \arctan\left(\frac{dy}{dx}\right) \quad (4.7)$$

$$m(x, y) = \sqrt{dx^2 + dy^2} \quad (4.8)$$

Una vez obtenidos la magnitud y el gradiente, es necesario obtener un histograma de orientación de los gradientes basandose primero en determinar un valor de acuerdo a la orientación de cada gradiente en cada píxel. Se realiza una división de rango de las orientaciones tal que de 0 grados a 45 grados le asignaremos un valor fijo de 1, de 46 a 90 grados el valor fijo será de 2, de 91 a 135 grados será 3, de 135 a 180 grados será 4, de 181 a 225 será 5, de 226 a 270 será 6, de 271 a 315 será 7, de 316 a 360 será el valor fijo de 8. Se asigna a cada píxel un intervalo en función de la orientación del gradiente, bajo los valores fijos se obtiene el histograma que será la suma de todos los pixeles correspondiente a una magnitud $h = (h_1, \dots, h_n)$.

EL objetivo método SIFT trabaja creando un vector de características de 128 elementos para cada punto de interés en este vector se obtiene información de la posición en coordenadas de la imagen, la escala que tiene y la orientación que se tiene alrededor de este punto. Esto permite al método reconocer los objetos presentes en una imagen mediante la identificación de los puntos de interés que predominan en una imagen.

4.3.3. Representación de la imagen

En este paso, es necesario realizar un refinamiento de los puntos de interes, eliminando aquellos que tienen un contraste bajo o eliminando puntos de interes poco localizados en contornos. Además es necesario normalizar el conjunto de descriptores obtenido y obtener solo aquellos que son más importantes para la clasificación. La Figura 4.13 muestra el resultado de utilizar el método SIFT sobre los distintos logos de partidos políticos. Los puntos dibujados en la imagen solo son las características locales detectadas, aun falta el paso de descripción y representación de la imagen a partir de esos puntos descritos.



FIGURA 4.13: Características locales con el método SIFT

4.4. Detector de objetos utilizando el algoritmo SURF

SURF es un descriptor que es similar a SIFT debido a que también obtiene puntos característicos y posteriormente describe la región alrededor de dichos puntos.

Sin embargo, algunas modificaciones hace de este método computacionalmente menos complejo. Este método se caracteriza porque el espacio escala lo crea mediante wavelets de Haar, en lugar de una pirámide de diferencias de gaussianas. Además, los puntos característicos se detectan con el determinante del Hessiano y la región de interés se describe como sumas, absolutas y no, de derivadas de primer orden, en lugar de un histograma orientado de gradientes.

Aunque se considera que la capacidad de detectar y caracterizar regiones es similar o en algunos casos inferior a SIFT debido a que no detecta rasgos muy puntuales, es más rápido al basar su detección en wavelets de Haar. La Figura 4.17 muestra como los pasos del algoritmo son idénticos al algoritmo SIFT. Sin embargo, como se dijo antes el algoritmo realiza algunas modificaciones para agilizar el proceso de detección.

En general, el algoritmo SURF es muy similar al método SIFT como lo muestra la Figura 4.17. Sin embargo existen pequeños cambios que describiremos en esta Sección.

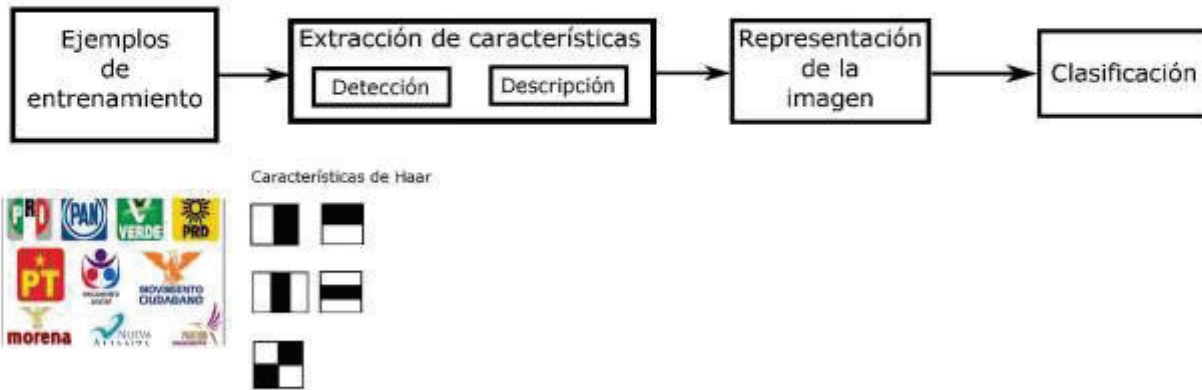


FIGURA 4.14: Etapas del algoritmo SURF

4.4.1. Detección de puntos

El algoritmo SURF detecta los puntos de interés basado en la imagen integral. Esto permite reducir el número de cálculos (complejidad computacional) que se realizan al obtener la matriz Hessiana.

La matriz Hessiana de escala σ se define como:

$$H(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix} \quad (4.9)$$

$$L(x, y, \sigma) = \left[\frac{\partial^2 G(x, y, \sigma)}{\partial x} * I(x, y) \right] \quad (4.10)$$

Estos filtros permiten analizar la imagen en el espacio-escala. El método SURF introduce aproximaciones de estos filtros utilizando la imagen integral. La Figura 4.15 muestra los filtros L_{yy} y L_{xy} y a la derecha sus aproximaciones D_{yy} y D_{xy}

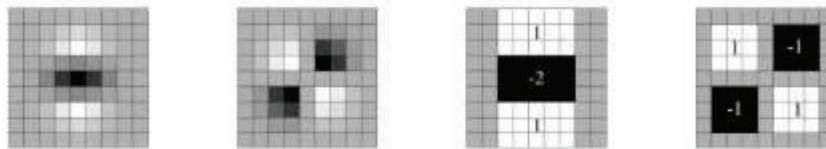


FIGURA 4.15: Dcretización de las derivadas parciales de la matriz Hessiana

Una vez obtenidos D_{xx}, D_{xy} y D_{yy} y aplicados a la imagen, se calcula el determinante aproximado de la matriz Hessiana. Los máximos locales del determinante son considerados puntos de interés.

$$\det(H_{approx}) = D_{xx}D_{yy} - 0,9D_{xy}^2 \quad (4.11)$$

Para que el algoritmo SURF detecte objetos bajo diferentes escalas es necesario utilizar filtros de distinto tamaño. Filtros de tamaño mayor representan escalas más gruesas. Los puntos de interés son obtenidos buscando los máximos locales de la matriz hessiana, en escala y espacio y luego interpolando para obtener una ubicación precisa de los puntos de interés.

4.4.2. Descriptor SURF

En el método SURF los puntos de interés son obtenidos aplicando filtros de Haar en el área de cada punto de interés. Esto permite obtener una representación de la distribución de intensidad.

El descriptor se obtiene de forma similar al descriptor SIFT, calculando la respuesta de los descriptores de Haar en lugar de obtener el gradiente en la vecindad, esto permite obtener una reducción considerable del tiempo de cálculo.

Alrededor del punto de interés se asigna la orientación usando una vecindad circular, sobre la cual se calcula la respuesta de los wavelets, ponderada por una gaussiana centrada en el punto de interés. Luego se busca, dentro de la vecindad circular, la ventana de tamaño $\frac{\pi}{3}$, donde la suma de la respuesta es máxima (ver Figura 4.16).

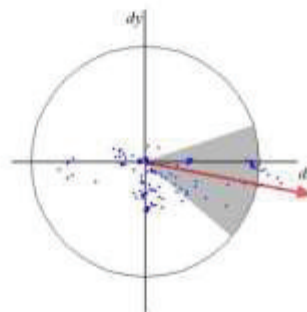


FIGURA 4.16: Asignación de la orientación

El descriptor se construye obteniendo una ventana cuadrada de tamaño $20s$, donde s es el factor de escala del punto de interés, sobre esta ventana se calcula la respuesta a los filtros Haar de primer grado, ponderada por una gaussiana centrada en el punto de interés. La región cuadrada de tamaño $20s$ es dividida en subregiones de 4×4 lo que permite preservar la información espacial. Para cada subregión 4×4 se calcula la

respuesta de los filtros de Haar y se obtienen respuestas horizontales d_x y respuestas verticales d_y . A partir de de estas respuestas se obtiene

$$v = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right) \quad (4.12)$$

Para cada una de las 16 regiones se obtiene un vector de 64 dimensiones, definido como:

$$d_i^I = \begin{bmatrix} d_i^I(1) \\ d_i^I(2) \\ \vdots \\ d_i^I(64) \end{bmatrix} \quad (4.13)$$

La Figura 4.17 muestra el resultado de utilizar el método SURF sobre los distintos logos de partidos políticos. Los puntos dibujados en la imagen solo son las características locales detectadas, aun falta el paso de descripción y representación de la imagen a partir de esos puntos descritos.



FIGURA 4.17: Características locales con el método SURF

Capítulo 5

Resultados experimentales

En esta sección se muestran los resultados experimentales de cada uno de los métodos utilizados en nuestro proyecto, así como una comparativa global de los resultados obtenidos en cada uno de los métodos. En los experimentos se implementaron los métodos y se probaron para detectar logos de los partidos políticos de México en video. La Figura 5.1 muestra los logos de los partidos políticos con los que se trabajo en los experimentos.



FIGURA 5.1: Logos de partidos políticos registrados en el INE

En los experimentos realizados se trabajo con dos experimentos generales. Uno para la detección de los logos en imágenes y el otro experimento sobre videos en tiempo real. Los resultados se enlistan a continuación

5.1. Resultados experimentales de detección sobre logos en imágenes

5.1.1. Conjuntos de datos

Entrenamiento

Para el caso específico de detector básico, SIFT y SURF se utilizó una imagen para cada logo de los partidos políticos con los que se trabajó como entrenamiento. La imagen de entrenamiento se seleccionó con la imagen definida por cada partido político como su logo oficial. Las imágenes para este caso no tenían ninguna deformación o rotación. Se seleccionaron logos sin ninguna rotación o transformación.

Para el caso del algoritmo Viola-Jones se utilizaron varias imágenes como entrenamiento. Las imágenes que se utilizaron fueron imágenes con el logo sin ninguna transformación, así como imágenes con ligeras transformaciones de rotación y oclusión para mejorar el desempeño del algoritmo.

Prueba

En todas las pruebas para el caso de detección en imágenes se evaluó el desempeño sobre dos conjuntos de imágenes. El primer conjunto son imágenes sin ninguna transformación o rotación, mientras que el segundo conjunto contiene imágenes con pequeñas rotaciones, transformaciones u oclusiones.

5.1.2. Resultados

Se obtuvieron resultados tanto en el desempeño de los métodos de detección como en el tiempo que se llevaron para detectar el logo. La Tabla 5.1 muestra los resultados en el tiempo de entrenamiento de cada método sobre un conjunto de 100 imágenes con tamaño diferente. Estos resultados en el tiempo de detección no son el tiempo que ocupan los métodos en cada imagen, sino el tiempo llevado a cabo en detectar los logos en las 100 imágenes. En los resultados el algoritmo básico es el método que ocupa el menor tiempo en detectar el logo, mientras que el método SIFT es el que ocupa el mayor tiempo para detectar los logos en las 100 imágenes.

5.1. Resultados experimentales de detección sobre logos en imágenes

CUADRO 5.1: Tiempos de detección sobre imágenes de tamaño variable

	Básico	Viola-Jones	SIFT	SURF
Tiempo	13.72	45.2978	172.379	53.019

En cuanto a los resultados de desempeño se realizaron pruebas sobre los dos conjuntos mencionados. Los resultados son mostrados en la Tabla 5.2. En la Tabla, $C1$ representa el conjunto de datos 1, definido anteriormente, Precisión, F-Measure y AUC-ROC son medidas de desempeño definidas en el Capítulo 3.

CUADRO 5.2: Desempeño de los diferentes métodos en la detección de logos

	Básico	Viola-Jones	SIFT	SURF
Precisión C1	100	100	100	100
F-Measure C1	1.00	1.00	1.00	1.00
AUC-ROC C1	1.00	1.00	1.00	1.00
Precisión C2	73	92	97	97
F-Measure C2	0.7	0.87	0.96	0.96
AUC-ROC C2	0.75	0.92	0.98	0.98

Los resultados sobre el conjunto de datos 1 (conjunto de logos sin ninguna transformación en los logos) muestran que los algoritmos siempre detectan el logo de forma eficiente, este sería el caso más simple pues el logo está en las condiciones óptimas, sin ninguna pequeña rotación u oclusión. Estos resultados son importantes pues en la vida real los videos donde se piensa detectar los logos no tienen ninguna transformación. Sin embargo, cuando los logos tienen alguna rotación o transformación los resultados son muy diferentes. El desempeño es severamente afectado en el caso del detector básico. Es importante mencionar que el algoritmo Viola-Jones se entreno primero con 50 imágenes, sin embargo en los resultados obtenidos en el conjunto de datos 2 (C2) el algoritmo tenía muchos fallos al detectar los positivos. La imagen 5.2 muestra los distintos falsos positivos que el algoritmo detecto. Esto se debio principalmente a que el conjunto de datos de entrenamiento era muy pequeño. Al aumentar y mejorar el conjunto de datos con más imágenes y bajo diferentes condiciones la precisión mejor significativamente. Es importante aclarar que es muy posible que al aumentar el número de imágenes en el conjunto de datos de entrenamiento y la variedad de estas o las condiciones bajo las que se encuentran, es muy probable que el desempeño se vea gratamente mejorado.



FIGURA 5.2: Falsos positivos detectados por el algoritmo Viola-Jones con solo 50 imágenes de entrenamiento

La imagen 5.3 nos hace inferir como al aumentar el tamaño de la base de datos de entrenamiento la precisión de detección se ve mejorada incluso ante transformaciones complejas como la que se aprecia en la imagen.



FIGURA 5.3: Mejora en el algoritmo Viola-Jones

Es importante señalar que en nuestros experimentos no se realizaron pruebas sobre imágenes con un tamaño estandar. Esto podría haber ayudado a mejorar el análisis sobre el tiempo de detección de los diferentes algoritmos.

5.2. Resultados experimentales de detección sobre logos en video

La base de datos de prueba para este caso consta de videos de spots de partidos políticos comprende un total de 27 videos grabados. Cada video tiene una duración media de 23 segundos y un frame rate de 25 imágenes por segundo.

Los videos han sido grabados a una resolución de 640×480 píxeles, sin embargo es posible obtener las imágenes a una menor resolución para que los algoritmos puedan detectar los logos en tiempo real y el tiempo de análisis no sea tan elevado.

Cada tipo de video lo hemos subdividido a su vez en dos grupos: un conjunto de videos compuesto por secuencias donde aparecen objetos que se mueven y otro en las que no existe objeto alguno moviéndose.

La Tabla 5.3 muestra los resultados de tiempo de detección. Este es obtenido, verificando el tiempo en el que aparece el primer logo en el video y el tiempo que tarda el algoritmo en reconocer el objeto. Los tiempos en la Tabla son reportados en segundos. En los experimentos realizados con videos de 640×480 los resultados obtenidos muestran que el algoritmo básico tiene ventaja sobre los otros algoritmos, esto es por el número de operaciones que realiza para detectar un logo.

CUADRO 5.3: Tiempos de detección sobre video

	Básico	Viola-Jones	SIFT	SURF
Tiempo	0.119 s	0.328	2.044	0.528

En cuanto a los resultados de desempeño se realizaron pruebas sobre los dos conjuntos mencionados. Los resultados son mostrados en la Tabla 5.2. En la Tabla, $C1$ representa el conjunto de datos de videos sin movimiento definido anteriormente, Precisión, F-Measure y AUC-ROC son medidas de desempeño definidas en el Capítulo 3.

Como puede observarse en la Tabla 5.4, los desempeños caen con respecto a la detección en las imágenes. esto es posiblemente debido a que cuando los videos tienen algun logo un poco rotado no lo detectan. El algoritmo que más sufre con este problema es el algoritmo básico. Sin embargo, es importante mencionar que el tiempo de respuesta es el menor de todos los algoritmos. Esto último es algo que se toma muy en cuenta a la hora de tomar una decisión. En nuestro caso, no es muy importante que el método sea robusto a cambios en la rotación y ante problemas de oclusión pues los spots generalmente se presentan sin estos problemas.

CUADRO 5.4: Desempeño de los diferentes métodos en la detección de logos

	Básico	Viola-Jones	SIFT	SURF
Precisión C1	88	91	93	91
F-Measure C1	0.91	0.940	0.95	0.94
AUC-ROC C1	0.904	0.92	0.93	0.92
Precisión C2	42	78	89	87
F-Measure C2	0.45	0.75	0.85	0.84
AUC-ROC C2	0.47	0.79	0.87	0.85

Es importante señalar que mientras más grande es el frame recogido del video es mas lento el proceso de detección, esto porque el número de operaciones necesarias para detectar los logos se incrementa significativamente.

Capítulo 6

Conclusiones

En este último capítulo se muestran las conclusiones del trabajo realizado en esta tesis, se describen también algunas líneas de investigación posibles para un trabajo futuro y los artículos publicados y por publicar de los resultados obtenidos en este trabajo de investigación. La mayoría de las conclusiones se obtienen a partir de los experimentos realizados y de los resultados obtenidos. En las conclusiones también se describe la influencia de los parámetros utilizados en los resultados de los diferentes algoritmos utilizados.

6.1. Conclusiones de los resultados obtenidos

En esta tesis se han realizado diferentes pruebas con diferentes algoritmos en el estado del arte para la detección de objetos en tiempo real. Como una aplicación real se implementaron 4 algoritmos para la detección de logos de partidos políticos. A través de los experimentos realizados se pudo observar que las técnicas actuales permiten detectar los logos de los partidos políticos perfectamente, bajo diferentes condiciones de escala y rotación sin ningún problema, sin embargo en la mayoría de los casos y en especial en el problema real estudiado no es necesario detectar los logos bajo condiciones de rotación, solamente bajo diferentes condiciones de escalado.

Los métodos de detección de objetos más sencillos como el detector básico que utilizan medidas de diferencia o similitud para detectar los objetos son más vulnerables al ruido, sin embargo, en algunos casos tienen un menor coste computacional. Algunos pasos previos a la búsqueda de la plantilla (logo) en la imagen permiten agilizar el proceso de detección al reducir el número de iteraciones. Para probar el sistema propuesto se dividieron las pruebas en dos grupos. 1) Se probaron los métodos con imágenes de logos de los distintos partidos políticos y 2) se detectaron los logos en tiempo real con grabaciones de video directas y/o haciendo pasar una hoja con el logo a identificar.

El objetivo de hacer estas pruebas fue el de evaluar el desempeño de los algoritmos bajo condiciones cambiantes como movimiento, cambio de escala y rotación del objeto o logo a detectar. Los resultados obtenidos se muestran en el Capítulo de resultados experimentales. Es importante aclarar que mientras el algoritmo básico convencional no es capaz de obtener un buen desempeño bajo cambios de rotación y visión parcial del objeto, el algoritmo es muy efectivo cuando no existe ningún cambio y en la situación real que se implementó (detección de logos de partidos políticos) no se ve afectado su desempeño, pues los logos de los partidos siempre aparecen completos y sin rotación alguna.

Bajo estas condiciones la técnica de detección básica obtiene muy buenos resultados y se puede optimizar para llegar a funcionar en tiempo real, siempre y cuando las imágenes tomadas o el tamaño de los frames sea de un tamaño adecuado. En los experimentos realizados se obtuvieron frames de tamaño 640×480 . Sin embargo, en experimentos realizados sobre imágenes con mayor resolución el algoritmo de búsqueda tiene una eficiencia computacional muy baja y es casi imposible que el algoritmo funcione en tiempo real. Esto debido a que el algoritmo invierte una gran cantidad de tiempo en buscar de forma iterativa la reducción del error y la correlación de las imágenes.

Es importante mencionar que el desempeño del algoritmo Viola-Jones puede verse mejorado si es entrenado con más imágenes. En nuestros experimentos ya no se realizaron más pruebas, pero podría ser realizado en un trabajo futuro.

Los desempeños de los algoritmos SIFT y SURF son muy buenos cuando los logos se encuentran rotados, transformados o incluso cuando se encuentra oculta alguna parte del logo.

6.2. Trabajo futuro

En este Capítulo se describen las conclusiones obtenidas al utilizar diferentes métodos para detección de objetos. Durante el desarrollo de la tesis se cumplió con los objetivos marcados. Sin embargo, a lo largo de la investigación se enfrentaron diversos problemas que podrían ser resueltos en el futuro. Esto podría ayudar a mejorar la detección de objeto y/o reducir el tiempo de detección, además de realizar un estudio más profundo de la influencia de los parámetros utilizados en la precisión de clasificación. A continuación se enumeran algunos puntos que podrían estudiarse en el futuro:

1. Escalar el sistema propuesto para enfrentar el problema de multclasificación. Existen diferentes técnicas para resolver este problema, sin embargo, por falta de tiempo no se implementó alguna de ellas, unicamente se trabajo con detección de objetos binaria.
2. Uno de los trabajos futuros es realizar un análisis exhaustivo para detectar el tamaño mínimo necesario de la plantilla. Esto puede ayudar a mejorar el tiempo de detección del objeto.
3. Estalecer las condiciones bajo las que los algoritmos son más efectivos.
4. Detectar los inicios y finales de los spots para contalibilizar el tiempo de cada partido político en televisión
5. Probar el algoritmo SURF con los distintos métodos de Detección y selección de puntis de interés.
6. Mejorar los tiempos de respuesta para cada unos de los algoritmos vistos en esta tesis.

6.3. Artículos publicados

6.3.1. Revistas Internacionales con Arbitraje Estricto

Jair Cervantes, Jesus Taltempa, Farid García, Arturo Yee, Jose S. Ruiz Castilla y Laura D. Jalili, "Comparative analysis of the techniques used in a recognition system of plant leaves". Revista Iberoamericana de Automática e Informática RIAI, Volume 14, Issue 1, JanuaryMarch 2017, Pages 104-114. factor de impacto:0.450

6.3.2. Capítulos de Libro

Laura D. Jalili, Morales A., Cervantes J., Ruiz-Castilla J.S. (2016) "Improving the Performance of Leaves Identification by Features Selection with Genetic Algorithms". In: Figueroa-García J., Lopez-Santana E., Ferro-Escobar R. (eds) Applied Computer Sciences in Engineering. WEA 2016. Communications in Computer and Information Science, vol 657. Springer.

Apéndice A

Artículos publicados

- Jair Cervantes, Jesus Taltempa, Farid García, Arturo Yee, Jose S. Ruiz Castilla y Laura D. Jalili, "Comparative analysis of the techniques used in a recognition system of plant leaves". Revista Iberoamericana de Automática e Informática RIAI, Volume 14, Issue 1, January-March 2017, PP 104-114.



ScienceDirect

Disponible en www.sciencedirect.com

Revista Iberoamericana de Automática e Informática industrial 14 (2017) 104–114



Análisis Comparativo de las técnicas utilizadas en un Sistema de Reconocimiento de Hojas de Planta

Jair Cervantes^{a*}, Jesús Taltempa^a, Farid García-Lamont^a, José S. Ruiz Castilla^a, Arturo Yee Rendon^b, Laura D. Jalili^a

^aPrograma de Investigación, Universidad Autónoma del Estado de México, Proleguante de An. Zumpango s/n, Pteco. El Tejocote, Texcoco, México, 52346.

^bFacultad de Ciencias de la Computación, Universidad Autónoma de Tlaxcala, Cuicuilco, Tlaxcala, México, 38015

Resumen

El desarrollo de sistemas de identificación de hojas de plantas es un reto actual que comprende numerosas aplicaciones que van desde alimentación, medicina, industria y medio ambiente. En la literatura actual, se han propuesto varias técnicas con el objetivo de identificar plantas en diversos campos de aplicación. Sin embargo, las técnicas actuales están restringidas al reconocimiento e identificación de tipos de plantas limitados, utilizando descriptores de características específicos. En este artículo, se realiza un análisis comparativo de diversos métodos de extracción de características (texturales, cromáticas y geométricas) y clasificación sobre conjuntos de plantas muy similares y disimiles entre sí. Dose conjuntos de hojas con características de forma similares son estudiados utilizando varios clasificadores. Se analiza el desempeño de diferentes combinaciones de características en cada conjunto. Los resultados obtenidos muestran que para incrementar el desempeño de los clasificadores estudiados, es necesaria una combinación de las diferentes técnicas de extracción de características, esta necesidad es mayor cuando se trabaja con conjuntos de hojas con características muy similares. Además, se muestra el mejor desempeño de un clasificador con otro.

Palabras Clave:

Clasificación, Descriptores, SVM, Conjuntos de Datos, Características.



ScienceDirect

Disponibile en www.sciencedirect.com

Revista Iberoamericana de Automática e Informática Industrial 14 (2017) 104–114



Análisis Comparativo de las técnicas utilizadas en un Sistema de Reconocimiento de Hojas de Planta

Jair Cervantes^{a,*}, Jesús Taltempa^a, Farid Garfía-Lamont^a, José S. Ruiz Castilla^a, Arturo Yee Rendón^b, Laura D. Jalili^b

^a *Posgrado e Investigación, Universidad Autónoma del Estado de México, Prolongación de Av. Zumpango 98, Pisco, El Tajón, Toluca, México, 52946.*

^b *Facultad de Ciencias de la Computación, Universidad Autónoma de Puebla, Chilcatlán, Puebla, México, 80013*

Resumen

El desarrollo de sistemas de identificación de hojas de plantas es un reto actual que comprende numerosas aplicaciones que van desde alimentación, medicina, industria y medio ambiente. En la literatura actual, se han propuesto varias técnicas con el objetivo de identificar plantas en diversos campos de aplicación. Sin embargo, las técnicas actuales están restringidas al reconocimiento e identificación de tipos de plantas limitados, utilizando descriptores de características específicos. En este artículo, se realiza un análisis comparativo de diversos métodos de extracción de características (texturales, cromáticas y geométricas) y clasificación sobre conjuntos de plantas muy similares y disímiles entre sí. Doce conjuntos de hojas con características de forma similares son estudiados utilizando varios clasificadores. Se analiza el desempeño de diferentes combinaciones de características en cada conjunto. Los resultados obtenidos muestran que para incrementar el desempeño de los clasificadores estudiados, es necesaria una combinación de las diferentes técnicas de extracción de características, esta necesidad es mayor cuando se trabaja con conjuntos de hojas con características muy similares. Además, se muestra el mejor desempeño de un clasificador con otro.

Palabras Clave:

Clasificación, Descriptores, SVM, Conjuntos de Datos, Características.

1. Introducción

En la actualidad la implementación de algoritmos de visión abarca casi cualquier campo de aplicación que uno pudiera imaginar. La agronomía parece no ser la excepción. El desarrollo de sistemas de visión que detecten automáticamente e identifiquen las plantas es un reto actual que tiene numerosas aplicaciones. Estas pueden ir desde simple identificación, detección de plagas, detección de enfermedades, identificación de plantas por personal de aduanas, identificación de la planta para cuidado y protección de pesticidas, etc.

La identificación de plantas no es un reto fácil debido a que existen muchas características foliares que los botánicos emplean para identificar las plantas, además de la enorme cantidad de diferentes plantas, muchas hojas poseen y/o comparten una o varias propiedades como: forma, tamaño, textura, color, aun cuando pertenecen a plantas diferentes. Aunado a ello existen otros factores que puede influir en algunas características de

la planta como: cromaticidad, color y textura, algunas de estas características se ven afectadas principalmente por nivel de madurez de la hoja, humedad, enfermedades, etc.

El desarrollo de algoritmos de identificación automática de hojas ha sido direccionado en años recientes por varios investigadores. Sin embargo, en reconocimiento de patrones los métodos de identificación pueden restringirse a descriptores geométricos, texturales, cromáticos y de venación.

En este artículo, estudiamos la influencia de los diferentes tipos de características (geométricas, texturales y cromáticas) en la precisión de clasificación. Además se analiza el desempeño de varios clasificadores sobre conjuntos de imágenes con características geométricas muy similares y disímiles entre sí.

En la metodología utilizada empleamos técnicas de extracción de características geométricas invariantes a escalado, traslación y rotación, técnicas de extracción de características texturales y cromáticas. En la fase de clasificación utilizamos un clasificador Bayesiano, el algoritmo de aprendizaje Backpropagation para redes neuronales y Máquinas de Vectores de Soporte (SVM). Los resultados obtenidos muestran la influencia de los tipos de características en la precisión de clasificación.

* Autor en correspondencia.
 Correos electrónicos: jcervante@uasam.mx (Jair Cervantes),
talte_209@hotmail.com (Jesús Taltempa), fglamont@yahoo.com.mx
 (Farid Garfía-Lamont), jsruizruiz@gmail.com (José S. Ruiz Castilla),
yeearendon@gmail.com (Arturo Yee Rendón),
jalili.anders@yahoo.com.mx (Laura D. Jalili)

2. Estado del Arte

Las plantas juegan un importante papel para la vida y el desarrollo humano, no solo son de interés en investigaciones de botánica, sino también en otras ramas, tales como la agricultura (Zhang and Lei, 2011), (Asraf et al., 2012), (Husin et al., 2012), (Xia et al., 2013), ecología vegetal (Zhang and Lei, 2011), (Zhang et al., 2013), (Du et al., 2013), medicamentos basado en plantas (Du et al., 2007), (Husin et al., 2012), (Zhang et al., 2013), (Gwo et al., 2013), conservación natural y también en muchas situaciones de interés público. En el mundo, existen aproximadamente una variedad de 310 000 a 420 000 especies de plantas (Du et al., 2007) sin tomar en cuenta que aún existen muchas que no han sido clasificadas. Por esta razón, identificar una planta a partir de imágenes de hojas no es una tarea trivial. Las técnicas de reconocimiento de patrones empleadas actualmente involucran técnicas de medición de características morfológicas y de textura de los objetos contenidos en la imagen y el desarrollo de sistemas capaces de reconocer el objeto a partir de aquellas características de medida. Es bien sabido que la mejor forma de extraer características válidas es basándose en la imagen de la hoja de la planta. En la literatura actual se ha mostrado que la forma externa de la hoja provee rica información para clasificar. Varios estudios, se han enfocado a la extracción de características y métodos de reconocimiento de patrones para la hoja, utilizando cuatro importantes características para clasificación, que son: La forma (Du et al., 2007) (Hu y Collomose, 2013) (Novotny, 2013) (Xia et al., 2013) (Gwo et al., 2013), textura (Sampallo, 2003) (Asraf et al., 2012) (Husin et al., 2012), el color (Zhang and Lei, 2011) (Cerruti et al., 2013), y la venación de la hoja (Du et al., 2013) (Laresse et al., 2014) (Laresse et al., 2014b).

La forma de la hoja es una de las características más importantes de la hoja de la planta y los dos enfoques básicos para este tipo de análisis son los basados en contorno y basados en la región. El enfoque basado en la región suele utilizar descriptores de momentos, que incluyen momentos geométricos, momentos de Zernike y momentos de Legendre. El enfoque basado en el contorno usualmente obtiene el contorno con métodos basados en la curvatura de la hoja (Zhang and Lei, 2011). Otros estudios han utilizado una combinación de características geométricas y textuales, permitiéndoles incluso utilizar hojas secas, mojadas o deformes (Husin et al., 2012). Los descriptores utilizados por los distintos autores van desde descriptores básicos como perímetro, área, circularidad y elipsidad (Sampallo, 2003), hasta descriptores invariantes como momentos de Hu y descriptores de Fourier para reconocimiento de contorno de la hoja (Novotny, 2013). Recientemente se han propuesto sistemas para extraer características que describen variaciones del borde de la hoja, utilizando descriptores invariantes a traslación, rotación y tamaño (Gwo et al., 2013).

La textura de la hoja puede ser definida como las características que la hoja posee en su superficie y que se manifiesta en imágenes digitales a través de variaciones de escala de grises en la imagen, a partir de donde se extraen las matrices de co-ocurrencia que servirán para obtener los descriptores textuales (Ma y Manjunath, 1996) (Manjunath y Ma, 1996).



Figura 1: Conjunto de hojas empleado

Otros estudios usan el color como característica de comparación de imágenes, ya que una simple similitud de color entre dos imágenes puede ser medida comparando sus histogramas de color (Zhang and Lei, 2011). En (Cerruti et al., 2013) utilizan como alternativa el espacio de color $L^*a^*b^*$, que muestra colores más consistentes y presenta más o menos el mismo eje para toda la hoja a diferencia del espacio RGB. Aunque los enfoques de clasificación como la forma, textura y color son válidos, no se ha documentado la influencia de cada tipo de características en el desempeño de los algoritmos de clasificación.

Los primeros estudios de reconocimiento de plantas utilizaron la cromaticidad de la planta como un descriptor importante para comparar imágenes. Descriptores muy simples de cromaticidad pueden obtener el color promedio en la región previamente segmentada de la hoja, gradiente promedio en el borde o la similitud de color entre dos imágenes que se puede medir mediante la comparación de sus histogramas de color. Descriptores de color más complejos utilizan momentos de invariancia comúnmente utilizados para obtener características geométricas pero incorporándoles la información de la variables de color de la hoja (Vesters y Cooper, 2000), (Smeulders et al., 2000), (Tico et al., 2000). Sin embargo, un problema recurrente en las hojas de las plantas es que la cromaticidad en las hojas no es estática, esta es variable con respecto al tiempo y comúnmente con respecto a otros factores más. Uno de los enfoques más empleados en la actualidad consiste en analizar la forma de la hoja extrayendo características geométricas como tamaño, elongación, elipsidad, área, longitud, diámetro, reoangularidad, esfericidad, excentricidad, etc. (Chaki y Parekh,

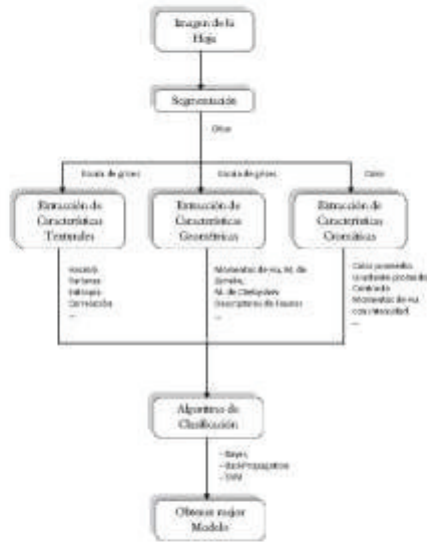


Figura 2: Diagrama de metodología

2012) (Park y Kim, 2004) (Xiang et al., 2007) (Kadir et al., 2012) (Kaur, 2012). Algunos autores han agregado a estos descriptores geométricos básicos, momentos de Hu y momentos de Fourier mejorando el desempeño de los clasificadores (Numar et al., 2012) (Hearn, 2009). Otros autores consideran además de cromaticidad y forma, la textura de la hoja (Novotny, 2013) (Cope et al., 2012) o utilizan combinaciones de descriptores para mejorar el desempeño de clasificación (Gwo et al., 2013) (Husin et al., 2012) (Zhang and Lei, 2011) (Brandstadt y Van Bang, 2006).

Existe gran cantidad de información en la literatura sobre identificación de plantas y/o árboles a partir de hojas. Sin embargo, en la mayoría de los sistemas actuales es fácil identificar dos problemas: 1) No existe un estudio completo sobre identificación complejo de plantas, es decir, un estudio que muestre la eficiencia de un algoritmo sobre conjuntos de imágenes muy similares entre sí. 2) Los trabajos actuales utilizan características texturales, cromáticas y geométricas. Sin embargo, no existen trabajos que realicen un análisis sobre la influencia de las características en el desempeño del clasificador. En este artículo se estudia la influencia de los tipos de descriptores en la precisión de clasificación en dos tipos generales de hojas; conjuntos de imágenes de hojas disímiles entre sí y conjuntos de hojas de imágenes con propiedades muy similares entre sí.

3. Metodología

La metodología utilizada es mostrada en la Figura 2. Los pasos son los habituales en cualquier sistema de reconocimiento

a partir de características. En los experimentos realizados se obtuvieron características geométricas, cromáticas y texturales, no se incluyeron características de venación debido a que el método tiene algunos fallos cuando las imágenes de hojas no contienen una venación prominente.

Primero, las imágenes son pre procesadas y segmentadas. Regularmente en pre procesamiento se emplea una máscara Gaussiana para obtener una buena segmentación (Gonzalez y Woods, 2010) (Sonka et al., 1993) (Nixon y Aguado, 2002). En las simulaciones realizadas se aplicó este tipo de máscaras. Sin embargo, debido a que las imágenes fueron tomadas en un ambiente controlado, se notó que la fase de preprocesamiento no era necesaria para este conjunto de datos.

En general, la segmentación autónoma es una de las tareas más difíciles en el procesamiento de imágenes. En imágenes de hojas a menudo están rodeadas de zonas verdes en el fondo. Sin embargo, las imágenes utilizadas son hojas en ambientes totalmente controlados (imágenes con únicamente la hoja con fondo blanco). Se realizaron pruebas de segmentación con los algoritmos de segmentación frontera adaptativa, Otsu y segmentación utilizando una fase de análisis de componentes principales (PCA) y no se obtuvieron diferencias entre estos por la naturaleza del conjunto de datos, por ello finalmente se utilizó el algoritmo de Otsu para realizar la segmentación.

3.1. Técnicas de segmentación

La región de la hoja en cada imagen fue segmentada empleando los siguientes pasos 1) Cálculo de alto contraste en escala de grises a partir de combinación lineal óptima de los componentes de color en RGB (Gonzalez y Woods, 2010) (Sonka et al., 1993); 2) Estimar una frontera óptima empleando momentos acumulativos de orden cero y de primer orden (método de Otsu) (Gonzalez y Woods, 2010) (Sonka et al., 1993); 3) Operaciones morfológicas para rellenar posibles espacios vacíos en la imagen segmentada (Gonzalez y Woods, 2010) (Sonka et al., 1993) (Nixon y Aguado, 2002). Todo esto con el objetivo de obtener una buena segmentación aún cuando existan cambios en las condiciones globales de brillo. Al segmentar la imagen, el sistema utilizado puede utilizar únicamente la región de la hoja, determinar sus bordes y calcular las propiedades mediante la extracción de características.

3.2. Extractores de Características

La extracción de características es un proceso crítico en cualquier sistema de reconocimiento de patrones. La precisión en la extracción de características tiene una gran influencia en los resultados de la identificación finales.

Una vez segmentada la región se extraen sus características. La extracción de características nos permite representar la imagen mediante un conjunto de valores numéricos con gran poder discriminativo, eliminando características redundantes y reduciendo la dimensionalidad de la imagen. Las características obtenidas son capaces de asociar rangos muy similares a imágenes similares, asociar rangos diferentes a imágenes diferentes, además de ser invariantes a escalado, rotación y traslación, permitiendo al clasificador reconocer objetos a pesar de tener diferente tamaño, posición y orientación. Todas estas características

juegan un rol importante en el desempeño del algoritmo y permiten al clasificador discriminar de una forma apropiada entre distintas clases (Nixon y Aguado, 2002).

Características geométricas

Las características geométricas son uno de los rasgos visuales más importantes y utilizados para clasificar un objeto. Las características geométricas proveen información del tamaño y forma de la región previamente segmentada.

Las características geométricas más elementales proveen información con significado intuitivo de las propiedades básicas de la región a reconocer, estas son: área de la región, redondez de la hoja, longitud del borde de la hoja, elongación definida por la longitud y ancho de la hoja, las coordenadas x e y del centro de gravedad, rectangularidad, proyección (sobre las componentes x e y), excentricidad, centro de gravedad (componentes x e y), factor Danielson, diámetro equivalente, longitud de ejes (x e y), orientación, solidez, extensión, área convexa, área rellenada, elipsidad (varianza, orientación, excentricidad, área, eje mayor, eje menor, centro de elipse x e y).

Sin embargo, un sistema eficiente de clasificación de hojas debe permitir reconocer las hojas independientemente de su orientación, localización y tamaño, i.e. debe ser invariante a escalado, rotación y traslación.

Los momentos son comúnmente utilizados en reconocimiento de imágenes, estos permiten reconocer imágenes independientemente de su rotación, traslación o inversión. Aunque los momentos de Hu no proporcionan información intuitiva de las propiedades básicas de la hoja, pues estos se obtienen a partir de combinación de múltiples características de la región segmentada, estos proporcionan información discriminativa para comparar dos hojas. La Figura 3 muestra los momentos de Hu obtenidos con dos hojas muy parecidas, pero de diferente familia. Los momentos de Hu geométricos son casi idénticos para ambas hojas, sin embargo los momentos de Hu con intensidad varían significativamente. Los momentos de Hu geométricos podrían discriminar perfectamente bien para dos hojas con formas geométricas diferentes, pero fallan al comparar formas similares, estos deben ser ayudados por otras características.

Los momentos invariantes fueron inicialmente introducidos por Hu (Hu, 1962) y consisten de 7 momentos (ϕ_1, \dots, ϕ_7), otras características invariantes comúnmente utilizadas son momentos de Flusser (F_1, \dots, F_4) (Flusser y Suk, 1993) (Mingqiang et al, 2008), Momentos R (R_1, \dots, R_{10}) (Liu et al., 2008), descriptores de Fourier (primeros 8 descriptores) (Mingqiang et al, 2008) (Nixon y Aguado, 2002). Los momentos de Hu de orden $(p + q)$ de una función de intensidad $f(x, y)$ son obtenidos a partir de los momentos ordinarios.

Los momentos de orden $(p + q)$ son definidos como:

$$m_{pq} = \sum_x \sum_y x^p y^q \rho(x, y). \quad (1)$$

donde $\rho(x, y)$ es definida por la región segmentada. Los momentos de orden pequeño describen la forma de la región. Por ejemplo m_{00} describe el área de la región segmentada, mientras que m_{01} y m_{10} definen las coordenadas x e y del centro de

gravedad. Sin embargo, los momentos $m_{02}, m_{03}, m_{11}, m_{12}, m_{20}, m_{21}$ y m_{30} son invariantes a traslación, rotación e inversión. Los momentos centrales son invariantes a desplazamiento y pueden ser calculados mediante

$$\mu_{pq} = \sum_{i,j \in R} (i - \bar{i})^p (j - \bar{j})^q \quad (2)$$

donde p, q pertenecen a la región segmentada y el centro de gravedad de la región es definido por:

$$\bar{i} = \frac{m_{10}}{m_{00}}, \bar{j} = \frac{m_{01}}{m_{00}} \quad (3)$$

Los momentos de Hu pueden ser obtenidos de la siguiente manera:

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ \phi_4 &= (\eta_{30} - 3\eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{20} + \eta_{12})(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 + \\ &\quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})(3\eta_{20} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \\ \phi_6 &= (\eta_{30} - \eta_{03})(\eta_{20} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 + \\ &\quad 4(\eta_{11}(\eta_{20} + \eta_{12})(\eta_{21} + \eta_{03})) \\ \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{20} + \eta_{12})(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 - \\ &\quad (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})(3\eta_{20} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \end{aligned} \quad (4)$$

donde $\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^p}$, $t = \frac{p+q}{2} + 1$.

Algunas veces es necesario tener características invariantes a traslación, rotación, escala y transformaciones afines. Los momentos de Flusser obtienen características derivadas a partir de los momentos centrales de segundo y tercer orden que son invariantes a transformaciones afines (Flusser y Suk, 1993), estas pueden ser obtenidas a partir de las siguientes ecuaciones:

$$\begin{aligned} I_1 &= \frac{\mu_{20}\mu_{02} - \mu_{11}^2}{\mu_{00}^4} \\ I_2 &= \frac{\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\mu_{12}^2 + 4\mu_{21}^2\mu_{03} - 3\mu_{21}^2\mu_{12}^2}{\mu_{00}^{10}} \\ I_3 &= \frac{\mu_{20}(\mu_{21}\mu_{03} - \mu_{12}^2) - \mu_{11}(\mu_{30}\mu_{03} - \mu_{21}\mu_{12}) + \mu_{02}(\mu_{30}\mu_{12} - \mu_{21}^2)}{\mu_{00}^7} \\ I_4 &= (\mu_{30}^3\mu_{03}^2 - 6\mu_{30}^2\mu_{21}\mu_{12}\mu_{03} - 6\mu_{30}\mu_{02}\mu_{21}\mu_{12}\mu_{03} + 9\mu_{20}^3\mu_{02}\mu_{12}^2 + \\ &\quad 12\mu_{20}\mu_{11}^2\mu_{21}\mu_{03} + 6\mu_{20}\mu_{11}\mu_{02}\mu_{12}\mu_{03} - 18\mu_{20}\mu_{11}\mu_{02}\mu_{21}\mu_{12} - \\ &\quad 8\mu_{11}^3\mu_{03}\mu_{03} - 6\mu_{20}\mu_{10}^2\mu_{03}\mu_{12} + 9\mu_{20}\mu_{10}^2\mu_{21} + 12\mu_{11}^2\mu_{02}\mu_{20}\mu_{12} - \\ &\quad 6\mu_{11}\mu_{02}^2\mu_{20}\mu_{21} + \mu_{02}^3\mu_{10}^2)/(\mu_{00}^{11}) \end{aligned} \quad (5)$$

Obtener características que sean invariantes a cambios de luz es muy importante para identificar hojas de la misma clase

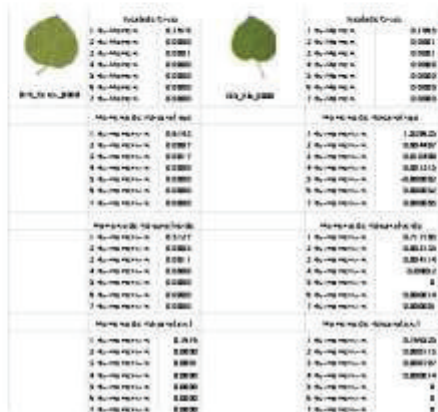


Figura 3: Comparación de características (momentos de Hu geométricos y con intensidad) de dos hojas

con cambios en la iluminación, características invariantes a iluminación pueden ser obtenidos con los momentos R. (Liu et al., 2008) descritos a continuación:

$$\begin{aligned}
 R_1 &= \frac{\sqrt{\phi_2}}{\phi_1}, R_2 = \frac{\phi_1 + \sqrt{\phi_2}}{\phi_1 - \sqrt{\phi_2}}, R_3 = \frac{\sqrt{\phi_3}}{\sqrt{\phi_4}}, \\
 R_4 &= \frac{\sqrt{\phi_5}}{|\sqrt{\phi_3}|}, R_5 = \frac{\sqrt{\phi_4}}{|\sqrt{\phi_3}|}, R_6 = \frac{|\phi_6|}{\phi_1 \cdot \phi_5}, R_7 = \frac{|\phi_6|}{\phi_1 \cdot \sqrt{|\phi_5|}}, \\
 R_8 &= \frac{|\phi_6|}{\phi_1 \cdot \sqrt{|\phi_5|}}, R_9 = \frac{|\phi_6|}{\sqrt{\phi_2} \cdot \sqrt{|\phi_5|}}, R_{10} = \frac{|\phi_6|}{\phi_3 \cdot \phi_4}
 \end{aligned} \tag{6}$$

En total 57 características geométricas fueron extraídas de cada imagen. El vector de características geométricas X_g obtenido puede ser representado como:

$$X_g = [x_1, x_2, \dots, x_{57}] \tag{7}$$

$$X_g = [x_{g1}, x_{g2}, x_F, x_R, x_{DF}] \tag{8}$$

donde x_g representa las características geométricas elementales descritas anteriormente ($x_{g1} = [x_1, \dots, x_{28}]$), x_{g2} representa las características invariantes de Hu ($x_{g2} = [x_{29}, \dots, x_{32}]$), x_F representa los 4 momentos invariantes de Flusser obtenidos ($x_F = [x_{36}, \dots, x_{39}]$), x_R representa los momentos invariantes a cambios en iluminación en la región segmentada ($x_R = [x_{40}, \dots, x_{49}]$), mientras que x_{DF} representa los primeros 8 descriptores de Fourier obtenidos de la región segmentada ($x_{DF} = [x_{50}, \dots, x_{57}]$). Una descripción detallada sobre los descriptores de Fourier puede ser obtenida en (Mingjiang et al, 2008) (Nixon y Aguado, 2002).

Características texturales

Las características texturales proveen información del arreglo espacial de colores o intensidades en la imagen. Los algo-

ritmos de extracción de características texturales buscan patrones repetitivos básicos con estructuras periódicas o aleatorias en imágenes. Estas estructuras dan lugar a una propiedad que puede ser rugosidad, aspereza, granulación, finura, suavidad, etc. Debido a que una textura repite un patrón a lo largo de una superficie, las texturas son invariantes a desplazamientos, ello explica porqué la percepción visual de una textura es independiente de la posición. En este artículo, se utilizaron características texturales de Haralick. Estos extractores toman en cuenta la distribución de valores de intensidad en la región obteniendo la media y rango de las siguientes variables: media, mediana, varianza, suavidad, sesgo, curtosis, correlación, energía o entropía, contraste, homogeneidad, y correlación. Las características texturales más importantes son las características de Haralick y los Patrones Binarios Locales (LBP por sus siglas en inglés -Local Binary Patterns-) (He y Wang, 1990) (Wang y He, 1990). Los descriptores de Haralick utilizados son descritos a continuación:

$$f_1 = \sum_i \sum_j [p(i, j)^2] \tag{9}$$

$$f_2 = \sum_{n=0}^{N_x-1} n^2 \left\{ \frac{\sum_{i=1}^{N_x} \sum_{j=1}^{N_x} p(i, j)}{|\lambda - \beta| = n} \right\} \tag{10}$$

$$f_3 = \frac{\sum_{i=1}^{N_x} \sum_{j=1}^{N_x} [(ij)p(i, j) - \mu_x \mu_y]}{\sigma_x \sigma_y} \tag{11}$$

$$f_4 = \sum_i \sum_j (i - \mu_x)^2 p(i, j) \tag{12}$$

$$f_5 = \sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i, j) \tag{13}$$

$$f_6 = \sum_{l=2}^{2N_x} i P_{x,y}(l) \tag{14}$$

$$f_7 = \sum_{l=2}^{2N_x} (i - f_6)^2 P_{x,y}(l) \tag{15}$$

$$f_8 = - \sum_{l=2}^{2N_x} P_{x,y}(l) \log(P_{x,y}(l)) \tag{16}$$

$$f_9 = - \sum_i \sum_j p(i, j) \log(p(i, j)) \tag{17}$$

$$f_{10} = \sum_{l=0}^{N_x-1} (i - f_9)^2 P_{x-y}(l) \tag{18}$$

$$f_{11} = - \sum_{l=0}^{N_x-1} P_{x-y}(l) \log(P_{x-y}(l)) \tag{19}$$

$$f_{12} = \frac{HXY - HXY1}{\max(HX, HY)} \tag{20}$$

$$f_{13} = (1 - e^{-2(NXY1 - NXY)})^{\frac{1}{2}} \tag{21}$$

$$f_A = (\text{Segundo valor propio mas grande de } Q)^{\frac{1}{2}} \quad (22)$$

donde $p(i, j)$ define el valor de la matriz de coocurrencia en la posición (i, j) . N_p = Niveles de intensidad trabajados en la región segmentada de la imagen. $p_x(i)$ = i -ésima entrada en una matriz de probabilidad original de coocurrencia que es obtenida mediante la suma de filas en $p(i, j)$. $p_y(j)$ = j -ésima entrada en una matriz de probabilidad original de coocurrencia que es obtenida mediante la suma de columnas en $p(i, j)$. R = Número total de elementos en la matriz de coocurrencia. μ_x, μ_y = Media de P_x y P_y . σ_x, σ_y = Desviación estándar de P_x y P_y . H_x, H_y = Entropía de P_x y P_y . En total se obtuvieron 219 descriptores texturales de cada imagen. 73 por cada canal de color utilizado. El vector de características Texturales X_t obtenido puede ser representado como:

$$X_t = [x_1, x_2, \dots, x_{219}] \quad (23)$$

$$X_t = [x_{LBP}, x_{RH}, x_{GLBP}, x_{GH}, x_{LBP}, x_{BH}] \quad (24)$$

donde $x_{LBP}, x_{GLBP}, x_{BLBP}$ representan las características LBP obtenidas en el canal de color R,G y B respectivamente. x_{RH}, x_{GH} y x_{BH} representan las características texturales de Haralick obtenidas en los canales R,G y B respectivamente. Una descripción detallada sobre los descriptores de LBP puede ser obtenida en (He y Wang, 1990) (Wang y He, 1990).

Características cromáticas

Las características cromáticas proveen información de la intensidad del color de una región segmentada, éstas características pueden ser calculadas por cada canal de intensidad, por ejemplo, rojo, verde, azul, escala de grises, tono (Hue), saturación (Saturation) e intensidad (Value), etc.

Las características que se emplearon fueron características de intensidad elementales como: media, varianza, mediana, kurtosis, coeficiente de variación, moda, asimetría, media Laplaciana, gradiente en la frontera y momentos de Hu con intensidad (ϕ_1, \dots, ϕ_7) . Los momentos de Hu integran información de la variable de color en combinación con la forma geométrica de la región de la hoja segmentada. En total 39 características, sin embargo, al trabajar en el espacio RGB se extrajeron 39 características por cada componente, en total 117 características. El vector de características cromáticas X_c obtenido puede ser representado como:

$$X_c = [x_1, x_2, \dots, x_{117}] \quad (25)$$

$$X_c = [x_{Rc}, x_{Bc}, x_{Gc}, x_{GRc}, x_{BRc}, x_{BRc}] \quad (26)$$

donde x_{Rc}, x_{Gc}, x_{Bc} representan las características de color elementales obtenidas en el canal de color R,G y B respectivamente. x_{GRc}, x_{BRc} y x_{BRc} representan las características de color utilizando momentos de Hu, obtenidas en los canales R,G y B respectivamente. Una descripción detallada sobre los momentos de Hu con intensidad puede ser obtenida en (Hu, 1962) (Mangqiang et al, 2008).

3.3. Algoritmos de clasificación

En los experimentos realizados se compararon los resultados con tres técnicas utilizadas para clasificación, se utilizó un clasificador Bayesiano, regresión logística, el algoritmo de aprendizaje *Backpropagation* para redes neuronales y Máquinas de Vectores de Soporte (SVM). Los algoritmos se describen en esta subsección.

Máquinas de Vectores Soporte (SVM)

Las SVM son una de las técnicas de clasificación más utilizadas en los últimos años. Las características clave de las SVMs son el uso de *kernels* al trabajar en conjuntos no-lineales, la ausencia de los mínimos locales, la solución depende de un pequeño subconjunto de datos y el poder discriminativo del modelo obtenido al optimizar el margen de separabilidad entre clases, estas características permiten a las SVM obtener resultados muy competitivos en comparación con otros clasificadores. Formalmente las SVM pueden ser definidas de la siguiente manera:

Assumiendo que un conjunto de datos de entrenamiento X es dado como:

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \quad (27)$$

i.e. $X = [x_i, y_i]_{i=1}^n$ donde $x_i \in \mathbb{R}^d$ y $y_i \in \{+1, -1\}$. Entrenar una SVM permite resolver un problema de programación cuadrática como sigue:

$$\begin{aligned} \max_{\alpha} & -\frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{K}(x_i, x_j) + \sum_{i=1}^l \alpha_i \\ \text{sujeito a: } & \sum_{i=1}^l \alpha_i y_i = 0, \quad C \geq \alpha_i \geq 0, i = 1, 2, \dots, l \end{aligned} \quad (28)$$

donde $C > 0$, $\alpha_i = [\alpha_1, \alpha_2, \dots, \alpha_l]^T$, $\alpha_i \geq 0, i = 1, 2, \dots, l$, son coeficientes que corresponden a x_i , x_i con α_i diferentes a cero que son llamados Vectores Soporte(SV). La función \mathbf{K} es una función. Las SVM son clasificadores lineales, es decir, se clasifica entre dos conjuntos de datos mediante la construcción de una línea que separa dos clases. Cuando esto no es posible, se utiliza una función llamada *kernel* que transforma el espacio de entrada a un espacio altamente dimensional, en donde los conjuntos pueden ser separados linealmente después de la transformación. Sin embargo, la elección de una función está restringida a aquellas que satisfacen las condiciones de Mercer (Vapnik, 1998).

Sea S el conjunto de SV obtenidos después del entrenamiento, entonces el hiperplano óptimo es dado por:

$$\sum_{i \in S} (\alpha_i y_i) \mathbf{K}(x_i, x_j) + b = 0 \quad (29)$$

y la función de decisión óptima es definida como

$$f(x) = \text{sign}(\sum_{i \in S} (\alpha_i y_i) \mathbf{K}(x_i, x_j) + b) \quad (30)$$

donde $x = [x_1, x_2, \dots, x_d]$ son los datos de entrada, α_i y y_i son los multiplicadores de Lagrange. Un nuevo objeto x puede ser clasificado empleando (30). El vector x_i es dado en la forma

Subconjunto	Tamaño oculto	Ejemplos	Clase oculta
1	10		Verde
2	6		Verde
3	14		Verde
4	17		Verde
5	12		Verde
6	3		Verde
7	7		Verde
8	9		Verde
9	11		Verde
10	10		Verde
11	17		Verde

Figura 4. Conjunto de hojas utilizado

de producto punto. Existe un multiplicador de Lagrange α para cada punto de entrenamiento. Cuando el máximo margen del hiperplano es encontrado, solamente los puntos más cercanos al hiperplano satisfacen $\alpha > 0$. Estos puntos son los vectores soporte (SV).

Redes neuronales

Una red neuronal es un conjunto de neuronas artificiales interconectadas que utilizan modelos matemáticos para procesar información. Las múltiples conexiones entre las neuronas forman un sistema adaptivo cuyos pesos se actualizan mediante un particular algoritmo de aprendizaje. Las redes neuronales han sido utilizadas en numerosos campos de aplicación con distintos algoritmos de aprendizaje (Portillo et al., 2009) (Jimenez et al., 2013) (Rossomando et al., 2010) (Valverde y Paez, 2007). De los distintos algoritmos de aprendizaje, uno de los más

utilizados es el algoritmo de aprendizaje *backpropagation* (BP). Para llevar a cabo el proceso de aprendizaje, el algoritmo de aprendizaje BP cambia iterativamente los pesos entre las neuronas minimizando el error cuadrático entre la salida deseada y la obtenida con los pesos actuales. Cada uno de los ejemplos del conjunto de entrenamiento $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ son utilizados para ajustar los pesos en la red. Al ser presentado un ejemplo, la señal es propagada hacia adelante de la red hasta que la salida es obtenida. La salida de la j -ésima unidad oculta es calculada como:

$$o_j^h = f_j^h(ne_{oj}^h) = \frac{1}{1 + \exp(-ne_{oj}^h)} \quad (31)$$

donde $ne_{oj}^h = \sum w_j^h x_{oj} + \theta_j^h$, w_j^h es el peso de la conexión de la j -ésima neurona de entrada a la j -ésima neurona oculta. θ_j^h y f_j^h representan el sesgo y la función de activación de la j -ésima neurona oculta. Por otro lado, la salida de la k -ésima neurona está dado por

$$o_{ok}^s = f_k^s(ne_{ok}^s) = \frac{1}{1 + \exp(-ne_{ok}^s)} \quad (32)$$

donde los superíndices h y s se refieren a las cantidades en las capas ocultas y de salida respectivamente. El error entre la salida actual y la salida deseada es calculado para ajustar los pesos mediante $E_o = \frac{1}{2} \sum_{k=1}^C (t_{ok} - o_{ok}^s)$. El procedimiento de ajuste es obtenido a partir del método de gradiente descendente para reducir la magnitud del error. El procedimiento es primeramente aplicado a los pesos en la capa de salida y retropropagado a través de la red hasta que los pesos en la primera capa han sido ajustados $\Delta w_{kj}^s = -\eta \frac{\partial E_o}{\partial w_{kj}^s}$ y $\Delta w_j^h = -\eta \frac{\partial E_o}{\partial w_j^h}$. Este procedimiento es realizado para cada ejemplo en el conjunto de datos hasta que se cumple un criterio de paro. Para un estudio profundo del algoritmo puede referirse a (Rumelhart et al., 1986) (Werbos, 1994).

Naive Bayes

Los clasificadores Bayesianos se basan en la teoría de decisión de Bayes. El principio de Bayes provee una metodología fundamental para resolver problemas de clasificación de patrones cuando la distribución de probabilidad de los patrones es conocida. Un clasificador Bayesiano utiliza un enfoque probabilístico para asignar la clase a un ejemplo. Sea C la clase de un objeto que pertenece a un conjunto de m clases (C_1, C_2, \dots, C_m) y X_k un objeto con k características $X_k = [x_1, x_2, \dots, x_k]$, en nuestro caso es el conjunto de características que definen a una hoja. el clasificador Bayesiano, calcula la probabilidad condicional posterior $p(C_i | X_k)$ utilizando la regla de Bayes:

$$p(C_i | X_k) = \frac{p(X_k | C_i) p(C_i)}{p(X_k)}, i = 1, 2, \dots, m. \quad (33)$$

En la Ec. 33 $p(X_k | C_i)$, $p(C_i)$ y $p(X_k)$ son calculadas a partir de los datos de entrenamiento.

Tabla 2: Desempeño con características cromáticas, texturales y geométricas.

Subconjunto	Cromáticas					Texturales					Geométricas							
	Bayes	BP	EL	SVM _{RF}	SVM _L	SVM _P	Bayes	BP	EL	SVM _{RF}	SVM _L	SVM _P	Bayes	BP	EL	SVM _{RF}	SVM _L	SVM _P
CH.1	88.7	94.21	88.38	94.91	88.43	92.83	36.28	47.78	42.97	34.902	32.49	32.95	81.39	93.56	84.83	95.408	77.45	94.89
CH.2	81.57	84.58	82.17	88.28	82.81	87.84	36.87	68.34	70.84	73.053	30.71	71.82	63.58	90.19	87.64	92.941	82.48	92.45
CH.3	88.22	94.73	89.27	95.03	78.73	94.61	37.18	74.86	71.86	77.518	31.29	77.19	78.06	89.32	83.63	91.918	79.38	90.92
CH.4	86.25	94.56	89.61	95.62	83.29	95.16	36.72	67.32	65.91	75.881	37.68	72.34	74.70	90.74	85.73	93.409	69.63	91.26
CH.5	93.67	95.56	92.82	95.72	91.42	94.19	38.39	71.84	68.81	72.841	22.31	70.93	83.44	95.11	88.33	97.289	81.41	95.56
CH.6	99.37	100	96.62	100	96.14	100	75.53	91.71	76.53	93.368	70.19	90.61	93.88	99.37	98.31	99.742	84.67	97.28
CH.7	96.56	96.79	94.35	97.45	91.42	95.33	48.25	82.39	81.27	85.082	44.35	82.42	97.87	98.75	96.45	95.733	93.96	97.39
CH.8	95.85	97.83	96.18	98.56	93.92	97.84	56.40	84.23	83.85	87.875	49.58	87.69	93.26	98.31	96.84	98.041	81.31	96.91
CH.9	88.38	90.64	90.38	90.87	84.72	90.37	38.39	65.77	63.21	69.69	39.73	67.32	64.59	93.80	87.46	94.410	79.93	91.37
CH.10	86.31	95.85	92.48	96.36	81.38	95.34	39.54	78.89	75.39	82.871	35.14	81.53	80.21	93.10	86.84	94.182	78.24	92.72
CH.11	77.62	92.33	88.31	93.91	75.83	91.73	33.72	63.41	62.84	88.161	31.27	67.56	74.83	88.35	78.03	91.324	71.85	91.08
CH.12	79.37	85.72	87.24	97.75	72.21	94.31	58.23	61.55	58.52	65.585	61.38	65.14	79.86	86.93	83.81	89.695	73.37	87.38

De acuerdo a la teoría de Bayes (Ng y Jordan, 2002) (Russell y Norvig, 2003), para una observación dada (X_i), la clase a la que pertenece está dada por la probabilidad posterior máxima:

$$f(X_i) = \arg \max_j p(C_j|X_i) \quad (34)$$

Para un estudio más exhaustivo puede referirse a (Ng y Jordan, 2002) (Russell y Norvig, 2003).

Regresión logística

La regresión logística es utilizada para modelar las probabilidades de clase posterior sin tener que aprender las densidades de clase condicional, facilitando la clasificación en pequeños conjuntos de entrenamiento y una menor complejidad. Sea $\pi_i = p(Y_i = 1|X_i)$ donde X_i es un vector de tamaño $1 \times (p + 1)$ con el primer elemento igual a 1 y los elementos restantes corresponden a las características extraídas de la hoja para el ejemplo i . El modelo de regresión logística relaciona π_i con las características mediante la función:

$$\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1 - \pi_i}\right) = X_i \beta \quad (35)$$

donde $\beta = (\beta_0, \beta_1, \dots, \beta_p)$ es el vector de coeficientes de regresión. Un estudio detallado y diferentes algoritmos de regresión logística pueden ser encontrados en (Borges et al., 2011), (Bernardo y Smith, 1994), (Dempster et al., 1977).

Tabla 1: Subconjuntos de datos.

Subconjuntos	Tamaño	Clases
CH.1	1095	19
CH.2	348	8
CH.3	962	14
CH.4	2481	27
CH.5	782	12
CH.6	186	3
CH.7	428	7
CH.8	832	9
CH.9	1248	13
CH.10	1632	20
CH.11	4398	37
CH.12	9425	90

4. Resultados Experimentales

En esta sección se muestran la técnica de selección de parámetros, normalización de datos y los resultados experimentales obtenidos con el sistema utilizado.

4.1. Conjunto de datos

En los experimentos realizados se utilizó el conjunto de datos ICL, que es una colección de hojas de la Universidad de Hefei (Intelengine.cn, 2016). El conjunto de datos contiene 16849 imágenes de hojas de 220 especies.

Con el objetivo de realizar un análisis comparativo de los clasificadores y las características extraídas separamos los conjuntos en dos tipos, trivial (cuyas formas pueden ser fácilmente diferenciadas) y complejas (cuyas formas son muy similares entre sí). A partir del conjunto de datos original se obtuvo un conjunto trivial y 11 conjuntos complejos. Los 12 subconjuntos de datos son descritos en la Tabla 1.

El conjunto de hojas trivial contiene hojas muy distintas entre sí, se puede inferir que dada la gran diferencia en las características de forma, el clasificador distinguirá entre una y otra hoja muy fácilmente. Por otro lado, en el conjunto de hojas complejo podría ser un reto para un clasificador distinguir entre una hoja y otras, dado que sus características son muy similares. Sin embargo, es cierto que cada clasificador utiliza técnicas muy distintas que pueden tener variaciones en sus resultados.

El proceso de separación de conjuntos, se realizó de forma manual de acuerdo a las semejanzas y diferencias notables a simple vista. El total de especies asociadas al conjunto trivial fue de 90 de las 220 con las que se contaba. La Figura 1 muestra un ejemplo de hoja de las 90 familias asociadas al conjunto trivial.

Para el conjunto de hojas complejas, se formaron subconjuntos a partir de las hojas que tenían gran similitud pero con diferente clase. A partir de las 220 especies de hojas inicial se formaron 11 subconjuntos distintos con 169 especies. El subconjunto con menos especies contiene solo 3 y el subconjunto con más especies asociadas contiene 37.

La Figura 4 muestra cada subconjunto creado con algunos ejemplos de las familias de hojas que se tomaron para formar cada grupo, además se describe la razón de similitud que se tomó en cuenta para asignar cada hoja y se muestran 3 hojas como ejemplo aunque, el total de cada subconjunto se describe en Familias Asociadas.

4.2. Normalización de datos

El vector final de características T fue almacenado en una matriz de tamaño $m \times 393$ que contiene m imágenes con 393 características. 57 características geométricas, 219 texturales y 117 cromáticas.

Todas las características extraídas fueron normalizadas con la relación:

$$f_{ij} = \frac{T_{ij} - \mu_j}{\sigma_j}$$

donde $i = 1, \dots, m$ y $j = 1, \dots, n$, μ_j y σ_j representan la media y desviación estándar de la j -ésima característica, T_{ij} representa la j -ésima característica del i -ésimo vector, m es el número de imágenes y n el número de características. Las características normalizadas tienen media cero y desviación estándar igual a 1.

4.3. Selección de parámetros

La selección de parámetros es un paso muy importante, ya que una buena selección de parámetros tiene un efecto considerable en el desempeño del clasificador.

En todos los clasificadores utilizados se obtuvieron los parámetros óptimos mediante validación cruzada y búsqueda de malla.

4.4. Resultados

En los experimentos realizados, todos los conjuntos de datos fueron normalizados y se utilizó validación cruzada con $k = 10$ para validar resultados, esto es, se realizaron 10 pruebas con 90% y 10% de los datos para entrenamiento y prueba respectivamente (los datos de prueba son excluyentes para cada una de las pruebas), el promedio de las 10 pruebas es el que se reporta en este artículo. La Tabla 2 muestra los resultados obtenidos con características geométricas, texturales y cromáticas, cada una como características individuales. En la Tabla 2, CH_i define el subconjunto de datos de hojas utilizado, Bayes representa los resultados obtenidos utilizando el algoritmo Naive Bayes, BP representa a los desempeños utilizando el algoritmo de aprendizaje backpropagation, RL regresión logística, mientras que SVM_{RBF}, SVM_L y SVM_P representan los resultados obtenidos con SVM utilizando los kernels RBF, lineal y Polinomial respectivamente. Para cada clasificador utilizado, se reportan las precisiones obtenidas con cada conjunto individual de características. La métrica utilizada para evaluar el desempeño del clasificador fue precisión y esta se obtiene de los aciertos del clasificador entre el total del conjunto de datos.

En los resultados obtenidos, no es posible inferir que la similitud entre hojas afecte significativamente a los clasificadores, pues los resultados mostrados en las Tablas 2, 3 y 4 nos permiten deducir que los desempeños de los clasificadores obtenidos con imágenes muy similares entre sí y disímiles, no son contrastantes. Sin embargo, es posible apreciar que las características texturales son poco discriminativas para la mayoría de los conjuntos de datos, excepto para el conjunto CH₆. Dos posibles razones son que el tamaño del conjunto de datos es muy pequeño y que aun cuando las hojas son muy parecidas entre sí,

las características obtenidas logran discriminar entre las 3 clases de este conjunto.

La Tabla 3 muestra los resultados obtenidos con combinaciones de características. En los resultados es posible apreciar una mejora en la precisión de clasificación en comparación con los desempeños obtenidos con características no combinadas. Utilizar únicamente características texturales no produce resultados satisfactorios. Sin embargo, cuando son combinadas las características texturales con geométricas o cromáticas, estas obtienen muy buenos desempeños.

La Tabla 4 muestra los resultados obtenidos con todas las características (cromáticas, texturales y geométricas). Sin embargo, se puede notar que utilizar todas las características no precisamente infiere en una buena clasificación, ya que los resultados fueron mejores al utilizar solo características geométricas con cromáticas. Un estudio sobre más exhaustivo utilizando técnicas de selección de características podría ayudar a comprender mejor la influencia de cada característica en el desempeño de los clasificadores.

Tabla 4. Desempeño con características geométricas, texturales y cromáticas. Cromáticas-Texturales-Geométricas

Subconjunto	Bayes	BP	RL	SVM _R	SVM _L	SVM _P
CH ₁	91.909	96.207	91.21	98.524	89.34	96.29
CH ₂	87.739	90.038	88.41	94.941	86.23	94.67
CH ₃	93.797	96.992	95.36	98.882	89.15	97.36
CH ₄	90.940	96.253	94.71	97.183	88.22	94.53
CH ₅	95.784	98.145	96.18	99.153	95.18	99.25
CH ₆	98.742	100	99.48	100	92.70	100
CH ₇	98.535	98.954	98.83	99.062	91.74	98.64
CH ₈	96.354	98.177	97.74	99.828	91.82	98.03
CH ₉	95.196	93.886	92.24	97.893	89.97	96.84
CH ₁₀	90.027	98.292	93.54	98.531	88.95	97.79
CH ₁₁	86.482	93.807	88.26	95.917	81.85	94.58
CH ₁₂	88.488	96.19	91.82	98.526	83.92	97.35

5. Conclusiones

En este artículo presentamos un análisis comparativo de la influencia de las características en el desempeño de varios clasificadores. El sistema extrae un conjunto de características basadas en propiedades geométricas, texturales y cromáticas. En los resultados obtenidos se aprecia que una combinación de las diferentes características es necesaria para obtener una buena precisión de clasificación. Aunque no siempre es necesario incluir la combinación de todas las características, ya que algunas de estas afectan la precisión de clasificación. Por lo tanto se debe de hacer una buena combinación de estas. En este trabajo se demostró que para los conjuntos utilizados la mejor combinación son las características geométricas con las características cromáticas.

English Summary

Comparative Analysis of the Techniques Used in a Recognition System of Plant Leaves.

Tabla 3: Desempeño con características cromáticas-texturales, cromáticas-geométricas y texturales-geométricas.

Subconjunto	Cromáticas-Texturales					Cromáticas-Geométricas					Texturales-Geométricas							
	Bayes	BP	RL	SVM _G	SVM _L	Bayes	BP	RL	SVM _G	SVM _L	Bayes	BP	RL	SVM _G	SVM _L			
CH.1	87.357	93.426	88.26	94.264	85.52	94.10	92.371	97.242	93.79	97.593	88.41	96.91	86.412	93.803	88.15	95.276	82.49	95.13
CH.2	82.061	84.732	81.82	86.253	80.92	85.68	87.858	99.584	88.69	92.476	81.32	91.78	88.122	91.869	89.04	91.421	84.87	90.53
CH.3	90.413	95.864	90.32	96.421	87.53	93.97	92.51	96.948	94.12	97.93	87.37	95.38	85.150	93.945	83.61	94.285	82.85	93.71
CH.4	87.261	94.005	87.58	95.218	84.21	93.74	91.429	97.262	91.83	95.825	90.82	95.72	84.332	92.370	85.45	94.720	80.45	92.89
CH.5	92.342	95.952	91.34	95.301	88.43	93.39	96.333	95.111	97.43	96.015	90.68	95.16	90.219	94.603	91.46	95.086	86.54	94.79
CH.6	98.742	100	97.75	100	94.25	99.82	98.742	100	98.76	100	91.45	99.83	96.226	99.371	97.90	99.371	89.71	98.58
CH.7	96.861	95.48	96.93	97.289	91.59	96.97	95.853	96.568	95.26	97.717	93.18	96.61	96.380	96.233	93.74	88.760	91.28	97.79
CH.8	95.052	96.135	95.58	96.211	90.18	95.35	96.844	96.211	95.71	97.823	93.55	97.30	96.614	96.697	94.52	96.958	92.89	96.19
CH.9	87.772	91.484	87.04	92.108	83.84	90.82	94.41	95.072	94.91	95.238	91.23	94.89	87.554	92.139	86.35	90.174	83.51	88.65
CH.10	87.748	95.926	88.19	95.332	82.34	95.71	89.77	95.294	91.29	96.163	87.26	95.83	90.505	92.628	90.72	91.939	86.16	90.91
CH.11	80.537	92.569	82.22	93.169	74.41	92.42	83.815	93.889	83.83	94.925	80.19	94.97	84.039	92.655	87.18	92.817	79.86	92.15
CH.12	81.0788	95.342	83.26	95.865	76.28	94.59	87.785	95.233	86.98	96.721	84.59	96.71	86.340	89.204	87.54	93.204	82.94	91.74

Abstract

The development of vision systems for identifying plants by leaves is an important challenge which has numerous applications ranging from food, medicine, industry and environment. Recently, several techniques have been proposed in the literature in order to identify plants in various fields of application. However, current techniques are restricted to the recognition and identification of plants using specific descriptors. In this paper, is accomplished a comparative analysis using different methods of feature extraction (textural, chromatic and geometric) and different methods of classification. The experiments are executed on very similar plants. Twelve sets of leaves with similar shape characteristics are studied using several classifiers. The performance of different combinations of classifiers-descriptors are analyzed in detail for each set. The results show that a combination of different feature extraction techniques is necessary in order to improve the performance. This combination of descriptors is more necessary when the leaves have similar characteristics.

Keywords:

Classification Descriptors SVM Data Sets

Agradecimientos

Agradecemos a los revisores anónimos que permitieron mejorar significativamente este trabajo de investigación. Este estudio fue financiado por la Secretaría de Investigación de la Universidad Autónoma del Estado de México con el proyecto de investigación 3771/2014/CIB.

Referencias

Azaf M.H., Noorizawati M.T., Shah Rizam M.S.B., 2012. A Comparative Study in Kernel-Based Support Vector Machine of Oil Palm Leaves Nutrient Disease, *Procedia Engineering*, 41, 1353-1359. DOI: 10.1016/j.proeng.2012.07.821

Bernardo J. and Smith A., 1994. *Bayesian Theory*, Wiley.

Brandt A., Van Fang L., 2006. Structure and linear time recognition of 3-leaf powers, *Information Processing Letters*, 96(6), 133-138. DOI: 10.1016/j.ipl.2006.01.004

Borje J., Blouca D.J. and Marçal A., 2011. Bayesian hyperspectral image segmentation with a discriminative class learning, *IEEE Transactions on Geoscience and Remote Sensing*, 49(5), 2151-2164. DOI: 10.1109/TGRS.2010.2097269

Cerutti G., Tougre L., Mille J., Vacavant A., Coquin D., 2013. Understanding leaves in natural images: A model-based approach for tree species identification, *Computer Vision and Image Understanding*, 117(10), 1482-1501. DOI: 10.1016/j.cviu.2013.07.003

Chaki, J., Paekki, K., 2012. Designing an automated system for plant leaf recognition, *International Journal of Advances in Engineering Technology*, 2(1), 149-158. DOI: 10.1.1.667.5268

Cope J. S., Corney D., Clark J. Y., Remagnino P., Wilkin P., 2012. Plant species identification using digital morphometrics: A review, *Expert Systems with Applications*, 39(8), 7562-7573. DOI: 10.1016/j.eswa.2012.01.073

Dempster A., Laird N. and Rubin D., 1977. Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society*, 1(39), 1-38. DOI: 10.2307/2344876

Do J.X., Wang X.F., Zhang G.J., 2007. Leaf shape based plant species recognition, *Applied Mathematics and Computation*, 185 (2), 883-893. DOI: 10.1016/j.amc.2006.07.072

Do J.X., Zhai Ch.M., Wang Q.P., 2013. Recognition of plant leaf image based on fractal dimension features, *Neurocomputing*, 116, 150-156. DOI: 10.1016/j.neucon.2012.08.028

Fusser, J., Suk, T., 1993. Pattern recognition by affine moment invariants, *Pattern Recognition* 26(1), 167-174. DOI: 10.1016/0021-8203(93)90098-H

Gonzalez R.C. and Woods R.E., 2010. *Digital Image Processing Using MATLAB*, Pearson.

Gwo Ch.T., Wei Ch.H., Li Y., 2013. Robust matching of edge features for leaf recognition, *Computers and Electronics in Agriculture*, 91, 124-134. DOI: 10.1016/j.compag.2012.12.005

Haralick R.M., 1979. Statistical and Structural Approaches to Texture, *Proc. IEEE*, 67, 786-804. DOI: 10.1109/PROC.1979.11328

Haralick R.M., Shanmugan K. and Dinstein I., 1973. Textural Features for Image Classification, *IEEE Transactions On System, Man, Cybernetics*, 6, 610-621. DOI: 10.1109/TSMC.1973.4309314

He D.C. and Wang L., 1990. Texture Unit, Texture Spectrum, And Texture Analysis, *Geoscience and Remote Sensing*, *IEEE Transactions on*, 28, 509-512. DOI: 10.1109/IGARSS.1989.575836

Hearn D.J., 2009. Shape analysis for the automated identification of plants from images of leaves, *Taxon*, 58, 934-954.

Hu M.K., 1962. Visual pattern recognition by moment invariants, *IRE Trans. Inform. Theory*, 8, 179-187. DOI: 10.1109/TIT.1962.1057692

Hu R., Collomouse J., 2013. A performance evaluation of gradient field HOG descriptor for sketch based image retrieval, *Computer Vision and Image Understanding*, 117(7), 790-806. DOI: 10.1016/j.cviu.2013.02.005

Husin Z., Shakaff A.Y.M., Aziz A.H.A., Farook R.S.M., Jaafar M.N., Hashim U., Hanun A., 2012. Embedded portable device for herb leaves recognition using image processing techniques and neural network algorithm, *Computers and Electronics in Agriculture*, 89, 18-29. DOI: 10.1016/j.compag.2012.07.009

- Intelligence.cn. (2016). *Intelligence.cn*. [online]. Intelligent Computing Laboratory, Chinese Academy of Sciences Homepage. Disponible en <http://www.intelligence.cn/English/Assets/>.
- Jiménez M.E., Sánchez A., Carvajal H., Blanco J., Sáenz J.C., 2013. Emisión Acústica y Redes Neuronales para Modelado y Caracterización del Proceso de Soldadura por Fricción Agitada. *Revista Iberoamericana de Automática e Informática Industrial* *10*(4), 434-440.
DOI: 10.1016/j.riai.2013.09.008
- Kadir, A., Nugroho, L. E., Susanto, A., y Santosa, P.L., 2012. Experiments of distance measurements in a foliage plant retrieval system. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 5, 356-363.
- Kang, G., y Kang, G., 2012. Classification of biological species based on leaf architecture. *International Journal of Engineering Research and Development*, 135-42.
DOI: 10.1.1.642.4985
- Larosa M., Nantas R., Cravetto R., Arango M., Gallo C., Granitto E.M., 2014. Automatic classification of legumes using leaf vein image features. *Pattern Recognition*, 47(1), 158-168.
DOI: 10.1016/j.patcog.2013.06.012
- Larosa M., Bazzà A., Cravetto R., Arango M., Gallo C., Granitto E.M., 2014. Multiscale recognition of legume varieties based on leaf venation images. *Expert Systems with Applications*, Volume 41(10), 4638-4647.
DOI: 10.1016/j.eswa.2014.01.029
- Liu J., Liu Y., y Yan C., 2008. Feature extraction technique based on the perceptual invariability. *Proceedings of the Fifth International Conference on Puzzy Systems and Knowledge Discovery*, Shandong, China, 551-554.
DOI: 10.1109/ikdd.2008.282
- Ma W., Manjunath B., 1996. Texture features and learning similarity. in: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 425-430.
DOI: 10.1109/cvpr.1996.517107
- Manjunath B. y Ma W., 1996. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 837-842.
DOI: 10.1109/34.831808
- Yang M., Kpalma K. and Ronin J., 2008. A Survey of Shape Feature Extraction Techniques. *Pattern Recognition Techniques*, INTECH Open Access Publisher.
- Ng, A.Y.; Jordan, M.I., 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. *Advances in Neural Information Processing Systems* 14, MIT-Press, 841-848.
- Nixon M. y Aguado A., 2002. *Feature Extraction and Image Processing*. Academic Press.
- Novotný P., Suk T., 2013. Leaf recognition of woody species in Central Europe. *Biosystems Engineering*, 115(4), 444-452.
DOI: 10.1016/j.biosysteng.2013.04.007
- Kumar N., Bellhumeur P.N., Biswas A., 2012. Leafsnap: a computer vision system for automatic plant species identification. in: *Proc. ECCV 2012*, 502-516.
DOI: 10.1007/978-3-642-33709-9_6
- Park J.S., Kim T.-Y., 2004. Shape-based image retrieval using invariant features. in: K. Aizawa, Y. Nakamura, S. Satoh, (Eds.), *Advances in Multimedia Information Processing-PCM 2004*, Berlin/Heidelberg Lecture Notes in Computer Science, pp. 146-153
DOI: 10.1007/978-3-540-30542-2_9
- Portillo E., Cabanes I., Marco M., Zubizarreta A., 2009. Aplicación de Redes Neuronales en la Detección de Regímenes Degradados en el Proceso Wedm. *Revista Iberoamericana de Automática e Informática Industrial* *6*(1), 39-50.
DOI: 10.1016/S1867-7912(09)70075-5
- Rosomando F.G., Soría C., Carelli R., 2010. Control de Robots Móviles con Incertidumbres Dinámicas usando Redes de Base Radial. *Revista Iberoamericana de Automática e Informática Industrial* *7*(4), 28-35.
DOI: 10.1016/S1867-7912(10)70067-1
- Rumelhart, D.E.; Hinton, G.E.; Williams, R.J., 1986. Learning representations by back-propagating errors. *Nature*, 323(6088): 533-536.
DOI: 10.1038/323536a0
- Russell S. y Norvig P., 2003. [1995]. *Artificial Intelligence: A Modern Approach* (2nd ed.). Prentice Hall.
- Sampallo C., 2003. Reconocimiento de tipos de hojas. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 7(21), 55-62. Asociación Española para la Inteligencia Artificial España.
- Smeulders A.W.M., Worring M., Sarti S., Gupta A., Jain R., 2000. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 1349-1380.
DOI: 10.1109/34.896972
- Sonka M., Hlavac V., Boyle R., 1993. *Image Processing, Analysis and Machine Vision*. Springer.
- Tico M., Haverinen T., Kuosmanen R., 2000. A method of color histogram creation for image retrieval. in: *Proceedings of the Nordic Signal Processing Symposium (NORSIG-2000)*, Kolmarden, Sweden, 157-160.
- Valverde R., Gachet D., 2007. Identificación de sistemas dinámicos utilizando redes neuronales RBF. *Revista Iberoamericana de Automática e Informática Industrial* *4*(2), 33-42.
DOI: 10.1016/S1867-7912(07)70207-9
- Vapnik V., 1995. *The Nature of Statistical Learning Theory*. Springer.
- Venters C., Cooper D., 2000. *A Review of Content-based Image Retrieval Systems*. Technical Report, Manchester Visualization Centre, Manchester Computing, University of Manchester, Manchester, UK.
- Wang L. y He D., 1990. Texture Classification Using Texture Spectrum. *Pattern Recognition*, 23(8), 905-910.
DOI: 10.1016/0031-3208(90)90195-8
- Werbos P.J., 1994. *The Roots of Backpropagation. From Ordered Derivatives to Neural Networks and Political Forecasting*. New York, NY: John Wiley Sons, Inc.
- Xin Ch., Lee J, Li Y., Song Y., Chang B., Chen T.S., 2013. Plant leaf detection using modified active shape models. *Biosystems Engineering*, 116(1), 23-35.
DOI: 10.1016/j.biosysteng.2013.06.008
- Du J.X., Wang X.P., Zhang G., 2007. Leaf shape based plant species recognition. *Applied Mathematics and Computation*, 185(2), 883-893.
DOI: 10.1016/j.amc.2006.07.072
- Zhang S., Lei Y.K., 2011. Modified locality linear discriminant embedding for plant leaf recognition. *Neurocomputing*, 74(16), 2284-2290.
DOI: 10.1016/j.neucom.2011.05.007
- Zhang S., Lei Y., Dong T., Zhang X.P., 2013. Label propagation based supervised locality projection analysis for plant leaf classification. *Pattern Recognition*, 46(7), 1891-1897.
DOI: 10.1016/j.patcog.2013.01.015

- Laura D. Jalili, Morales A., Cervantes J., Ruiz-Castilla J.S. (2016) "Improving the Performance of Leaves Identification by Features Selection with Genetic Algorithms". In: Figueroa-García J., Lopez-Santana E., Ferro-Escobar R. (eds) Applied Computer Sciences in Engineering, WEA 2016. Communications in Computer and Information Science, vol 657. Springer.

Selección de Atributos para Mejorar el Desempeño en Identificación de Hojas

Laura D. Jalili, Alfredo Morales, Jair Cervantes, José Sergio Ruiz Castilla

Universidad Autónoma del Estado de México

Posgrado e Investigación

UAEMEX-Texcoco

joervantes@uaemex.mx

Abstract—El desarrollo de sistemas de visión para identificación de hojas de plantas a partir de imágenes es un reto actual muy importante. Una de las principales líneas de investigación es la mejora en el desempeño. El desempeño de los sistemas de identificación automática está directamente ligado a las características extraídas. Las características extraídas permiten identificar o clasificar el objeto. Sin embargo, en algunas ocasiones el aumento de características introduce ruido afectando el desempeño del sistema. En este trabajo de investigación, se propone un algoritmo genético para extraer las características más discriminativas. La técnica propuesta disminuye el ruido introducido en la dimensionalidad del conjunto de datos y mejora el desempeño en la identificación de las plantas. En los resultados experimentales, se comparan los resultados utilizando técnicas clásicas de selección de características. Los resultados experimentales muestran que la técnica propuesta obtiene una mejora significativa en el desempeño del sistema de identificación.

I. INTRODUCCIÓN

En la actualidad, los algoritmos de visión han sido implementados en muchas áreas de investigación. Una de esas áreas es la agronomía. El desarrollo de sistemas de visión que detecten automáticamente e identifiquen las plantas es un reto actual que tiene numerosas aplicaciones. Estas pueden ir desde simple identificación, detección de plagas, detección de enfermedades, identificación de plantas por personal de aduanas, identificación de la planta para cuidado y protección de pesticidas, etc.

El desarrollo de algoritmos de identificación automática de hojas ha sido direccionado por varios investigadores en años recientes. Sin embargo, la identificación de plantas no es un reto fácil debido a diversos problemas para identificar las plantas como: las características foliares que los botánicos emplean para identificar las plantas; la enorme cantidad de diferentes plantas; muchas plantas poseen y/o comparten una o varias propiedades como: forma, tamaño, textura, color, aún cuando pertenecen a plantas diferentes; algunas características de color y textura en las hojas de las plantas son modificadas por el nivel de madurez de la hoja, humedad, enfermedades, etc.

En los últimos años, varios investigadores han desarrollado metodologías para identificar las plantas a partir de las hojas. La mayoría de estos sistemas se han enfocado en extraer características geométricas, texturales y de cromaticidad para identificar la hoja. Las características extraídas van desde

geométricas básicas, hasta invariantes a escalado, traslación y rotación o descriptores de Fourier.

Sin embargo, el desempeño de un clasificador no depende del número de características del conjunto de datos, sino del poder discriminativo de las características. El uso de características no adecuadas introduce ruido y provoca que el clasificador aprenda una hipótesis sesgada, degradando el desempeño del clasificador. Los datos con ruido degradan el rendimiento de algoritmos de aprendizaje al sesgar los límites de decisión o sobreajuste del modelo mediante la incorporación de puntos de datos incorrectos.

Esta investigación está motivada por estas desventajas. El algoritmo propuesto utiliza un algoritmo genético para obtener los atributos que mejor ayudan al desempeño del clasificador. El algoritmo propuesto reduce el ruido introducido al clasificador al hacer una búsqueda de combinaciones de atributos. La técnica propuesta explota el poder de búsqueda de los algoritmos genéticos, esta búsqueda es guiada utilizando diferentes métricas de desempeño. Los resultados experimentales obtenidos muestran que el algoritmo propuesto puede ayudar a mejorar el desempeño de los clasificadores. Finalmente, se muestran las conclusiones sobre el impacto discriminativo de las características en la precisión de clasificación.

II. ESTADO DEL ARTE

Las plantas juegan un importante papel para la vida y el desarrollo humano, en diferentes áreas de investigación como: la agricultura [19] [20] [21] [25], ecología vegetal [19] [23] [26], medicamentos basado en plantas [18] [21] [23] [27], conservación natural y también en muchas situaciones de interés público. En el mundo, existen aproximadamente una variedad de 310 000 a 420 000 especies de plantas [18], sin tomar en cuenta que aún existen muchas que no han sido clasificadas. Por esta razón, identificar una planta a partir de imágenes de hojas no es una tarea trivial. Las técnicas de reconocimiento de patrones empleadas actualmente involucran técnicas de medición de características morfológicas y de textura de los objetos contenidos en la imagen y el desarrollo de sistemas capaces de reconocer el objeto a partir de aquellas características de medida. Es bien sabido que la mejor forma de extraer características válidas es basándose en la imagen de la hoja de la planta. En la literatura actual se ha mostrado que la forma externa de la hoja provee rica información para



Fig. 1. Conjunto de hojas empleado

clasificar. Así varios estudios, se han enfocado a la extracción de características y métodos de reconocimiento de patrones para la hoja, utilizando cuatro importantes características para clasificación, que son: La forma [18] [22] [24] [25] [27], textura [17] [20] [21], el color [19] [28], y la venación de la hoja [26] [29] [30].

La forma de la hoja es una de las características más importantes de la hoja de la planta y los dos enfoques básicos para este tipo de análisis son los basados en contorno y basados en la región. El enfoque basado en la región suele utilizar descriptores de momentos, que incluyen momentos geométricos, momentos de Zernike y momentos de Legendre. El enfoque basado en el contorno usualmente obtiene el contorno con métodos basados en la curvatura de la hoja [19]. Otros estudios han utilizado una combinación de características geométricas y texturales, permitiéndoles incluso utilizar hojas secas, mojadas o deformes [21]. Los descriptores utilizados por los distintos autores van desde descriptores básicos como perímetro, área, circularidad y elipsidad [17], hasta descriptores invariantes como momentos de Hu y descriptores de Fourier para reconocimiento de contorno de la hoja [24]. Recientemente se han propuesto sistemas para extraer características que describen variaciones del borde de la hoja, utilizando descriptores invariantes a traslación, rotación y tamaño [27].

La textura de la hoja puede ser definida como las características que la hoja posee en su superficie y que manifiesta como variaciones de escala de grises en la imagen, a partir de donde se extraen las matrices de coocurrencia que servirán

para obtener los descriptores texturales [13] [14]. Otros estudios usan el color como característica de comparación de imágenes, ya que una simple similitud de color entre dos imágenes puede ser medida comparando sus histogramas de color [19]. En [28] utilizan como alternativa el espacio de color $L^* a^* b^*$, que muestra colores más consistentes y presenta más o menos el mismo eje para toda la hoja a diferencia del espacio RGB. Aunque los enfoques de clasificación como la forma, textura y color son válidos, no se ha documentado la influencia de cada tipo de características en el desempeño de los algoritmos de clasificación.

Los primeros estudios de reconocimiento de plantas utilizaron la cromaticidad de la planta como un descriptor importante para comparar imágenes. Descriptores muy simples de cromaticidad pueden obtener el color promedio en la región previamente segmentada de la hoja, gradiente promedio en el borde o la similitud de color entre dos imágenes que se puede medir mediante la comparación de sus histogramas de color. Descriptores de color más complejos utilizan momentos de invariancia comúnmente utilizados para obtener características geométricas pero incorporándoles la información de la variables de color de la hoja [1] [2] [3]. Sin embargo, un problema recurrente en las hojas de las plantas es que la cromaticidad de estas no es estática, esta es variable con respecto al tiempo y comúnmente con respecto a otros factores más. Uno de los enfoques más empleados en la actualidad consiste en analizar la forma de la hoja extrayendo características geométricas como tamaño, elongación, elipsidad, área, longitud, diámetro, rectangularidad, esfericidad, excentricidad, etc. [4] [5] [6] [7] [8]. Algunos autores han agregado a estos descriptores geométricos básicos, momentos de Hu y momentos de Fourier mejorando el desempeño de los clasificadores [9] [10] [11]. Otros autores consideran además de cromaticidad y forma, la textura de la hoja [24] [12] o utilizan combinaciones de descriptores para mejorar el desempeño de clasificación [27] [21] [19] [15].

Por otro lado, los algoritmos genéticos (AG) son heurísticas bastante utilizadas en los últimos años. Los AG fueron originalmente introducidos en la literatura por John Holland en los 70's. Los AG son algoritmos de optimización muy populares basados en la mecánica de la selección natural y la evolución. En particular, son procedimientos de búsqueda heurística que modifican el valor de los individuos comúnmente codificados como cadenas binarias. éstos han sido aplicados profusamente en muchos campos como bio informática, predicción [31], finanzas [32], control de procesos bio-químicos [33], manufactura [34], vehículos autónomos [35], robotica [36], etc.

Concretamente en problemas de clasificación los genéticos han sido utilizados para obtener parámetros óptimos (tuning) [37], reducir la dimensionalidad del conjunto de datos [38], mejorar la precisión de clasificación [39] y hasta para generar conjuntos de reglas compactas a partir de datos de entrada [40].

Existe gran cantidad de información en la literatura sobre identificación de plantas y/o arboles a partir de hojas. Sin embargo, en la mayoría de los sistemas actuales es fácil



Fig. 2. Diagrama de metodología propuesta

Identificar dos problemas:

- Los trabajos actuales utilizan características texturales, cromáticas y geométricas o combinaciones anteriores. Sin embargo, no existen trabajos que realicen un análisis sobre la influencia de las características en el desempeño del clasificador.
- Aunque las técnicas basadas en AG mejoran el desempeño de los clasificadores no existen en la literatura técnicas que ayuden a seleccionar los mejores atributos en conjuntos de datos de hojas.

En esta investigación se implementa un AG para identificar las características más discriminativas del conjunto de datos, esto permite estudiar la influencia de los descriptores en la precisión de clasificación.

III. MÉTODO PROPUESTO

La metodología del sistema propuesto es mostrada en la Figura 2. Los primeros pasos son los habituales en cualquier sistema de reconocimiento a partir de características. En los experimentos realizados se obtuvieron características geométricas, cromáticas y texturales, no se incluyeron características de venación debido a que el método tiene algunos fallos cuando las imágenes de hojas no contienen una venación prominente. Debido a las condiciones de espacio se definen brevemente los pasos que son comunes en los sistemas de clasificación y que son llevados a cabo también por el método propuesto.

Primero, las imágenes son preprocesadas y segmentadas. Regularmente en preprocesamiento se emplea una máscara Gaussiana para obtener una buena segmentación. En las simulaciones realizadas se aplicó este tipo de máscaras. Sin embargo, debido a que las imágenes fueron tomadas en un ambiente controlado, se notó que la fase de preprocesamiento no era necesaria para este conjunto de datos.

En general, la segmentación autónoma es una de las tareas más difíciles en el procesamiento de imágenes. En imágenes de hojas a menudo están rodeadas de zonas verdes en el fondo. Sin embargo, las imágenes utilizadas son hojas en ambientes totalmente controlados (imágenes de únicamente la hoja con fondo blanco). Se realizaron pruebas de segmentación con los algoritmos de segmentación frontera adaptativa, Otsu y segmentación utilizando una fase de análisis de componentes principales (PCA) y no se obtuvieron diferencias entre estos por la naturaleza del conjunto de datos, por ello finalmente se utilizó el algoritmo de Otsu para realizar la segmentación.

A. Técnicas de segmentación

La región de la hoja en cada imagen fue segmentada empleando los siguientes pasos 1) Cálculo de alto contraste en escala de grises a partir de combinación lineal óptima de los componentes de color en RGB; 2) Estimar una frontera óptima empleando momentos acumulativos de orden cero y de primer orden (método de Otsu). 3) Operaciones morfológicas para rellenar posibles espacios vacíos en la imagen segmentada. Todo esto con el objetivo de obtener una buena segmentación aún cuando existan cambios en las condiciones globales de brillo. Al segmentar la imagen, el sistema propuesto puede utilizar únicamente la región de la hoja, determinar sus bordes y calcular las propiedades mediante la extracción de características.

B. Extractores de Características

Una vez segmentada la región se extraen sus características. La extracción de características nos permite representar la imagen mediante un conjunto de valores numéricos con gran poder discriminativo, eliminando características redundantes y reduciendo la dimensionalidad de la imagen. Las características obtenidas son capaces de asociar rangos muy similares a imágenes similares; asociar rangos diferentes a imágenes diferentes, además de ser invariantes a escalado, rotación y traslación, permitiendo al clasificador reconocer objetos a pesar de tener diferente tamaño, posición y orientación. Todas estas características juegan un rol importante en el desempeño del algoritmo y permiten al clasificador discriminar de una forma apropiada entre distintas clases.

1) *Características geométricas:* Las características geométricas son los rasgos visuales más importantes y utilizados para clasificar plantas. Las características geométricas más elementales describen las propiedades básicas de la región a reconocer, estas son; área de la región, redondez de la hoja, longitud del borde de la hoja, elongación definida por la longitud y ancho de la hoja, las coordenadas x e y del centro de gravedad, densidad, definida por la longitud

de los bordes de la hoja y el área de esta. Sin embargo, un sistema eficiente de clasificación de hojas debe permitir reconocer las hojas independientemente de su orientación, localización y tamaño, i.e. debe ser invariante a escalado, rotación y traslado.

Los momentos son comúnmente utilizados en reconocimiento de imágenes, estos permiten reconocer imágenes independientemente de su rotación, traslación o inversión. Los momentos invariantes fueron inicialmente introducidos por Hu [16]. Los dos momentos de Hu de orden $(p+q)$ de una función de intensidad $f(x,y)$ son definidos como:

Los momentos de orden $(p+q)$ son definidos como:

$$m_{pq} = \sum_x \sum_y x^p y^q \rho(x,y) \quad (1)$$

donde $\rho(x,y)$ es definida por la región segmentada. Los momentos de orden pequeño describen la forma de la región. Por ejemplo m_{00} describe el área de la región segmentada, mientras que m_{01} y m_{10} definen las coordenadas x e y del centro de gravedad. Sin embargo, los momentos m_{02} , m_{03} , m_{11} , m_{12} , m_{20} , m_{21} y m_{30} son invariantes a traslación, rotación e inversión. Los momentos centrales son invariantes a desplazamiento y pueden ser calculados mediante

$$\mu_{pq} = \sum_{i,j \in R} (i-\bar{i})^p (j-\bar{j})^q \quad (2)$$

donde p, q pertenecen a la región segmentada y el centro de gravedad de la región es definido por:

$$\bar{i} = \frac{m_{10}}{m_{00}}, \bar{j} = \frac{m_{01}}{m_{00}} \quad (3)$$

Los momentos de Hu pueden ser obtenidos de la siguiente manera:

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ \phi_4 &= (\eta_{30} - 3\eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\ &\quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ \phi_6 &= (\eta_{30} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + \\ &\quad 4(\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})) \\ \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - \\ &\quad (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (4)$$

donde $\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q}{2}}}$, $t = \frac{t+1}{2} + 1$.

Otras características empleadas fueron descriptores de elipse, convexidad de región, momentos de Flusser y orientación, en total 57 características geométricas fueron extraídas de cada imagen.

2) *Características texturales*: Los algoritmos de extracción de características texturales buscan patrones repetitivos básicos con estructuras periódicas o aleatorias en imágenes. Estas estructuras dan lugar a una propiedad que puede ser rugosidad, aspereza, granulación, fineza, suavidad, etc. Debido a que una textura repite un patrón a lo largo de una superficie, las texturas son invariantes a desplazamientos, ello explica porqué la percepción visual de una textura es independiente de la posición. En este artículo, se utilizaron características texturales de Haralick. Estos extractores toman en cuenta la distribución de valores de intensidad en la región obteniendo la media y rango de las siguientes variables: media, mediana, varianza, suavidad, sesgo, curtosis, correlación, energía o entropía, contraste, homogeneidad, y correlación. Los descriptores de Haralick utilizados son descritos a continuación:

$$f_1 = \sum_i \sum_j [p(i,j)^2] \quad (5)$$

$$f_2 = \sum_{n=0}^{N_x-1} n^2 \left\{ \frac{\sum_{i=1}^{N_x} \sum_{j=1}^{N_x} p(i,j)}{|i-j|=n} \right\} \quad (6)$$

$$f_3 = \frac{\sum_{i=1}^{N_x} \sum_{j=1}^{N_x} [i/j p(i,j) - \mu_x \mu_y]}{\sigma_x \sigma_y} \quad (7)$$

$$f_4 = \sum_i \sum_j (i - \mu_x)^2 p(i,j) \quad (8)$$

$$f_5 = \sum_i \sum_j \frac{1}{1 + (i-j)^2} p(i,j) \quad (9)$$

$$f_6 = \sum_{i=2}^{2N_x} i P_{x+y}(i) \quad (10)$$

$$f_7 = \sum_{i=2}^{2N_x} (i - f_6)^2 P_{x+y}(i) \quad (11)$$

$$f_8 = - \sum_{i=2}^{2N_x} P_{x+y}(i) \log\{P_{x+y}(i)\} \quad (12)$$

$$f_9 = - \sum_i \sum_j p(i,j) \log\{p(i,j)\} \quad (13)$$

$$f_{10} = \sum_{i=0}^{N_x-1} (i - f_9)^2 P_{x-y}(i) \quad (14)$$

$$f_{11} = - \sum_{i=0}^{N_x-1} P_{x-y}(i) \log\{P_{x-y}(i)\} \quad (15)$$

$$f_{12} = \frac{HXY - HXY1}{\max\{HX, HY\}} \quad (16)$$

$$f_{13} = (1 - e^{-2(HXY2 - HXY)})^{\frac{1}{2}} \quad (17)$$

$$f_{14} = (\text{Segundo eigenvalor mas grande de } Q)^{\frac{1}{2}} \quad (18)$$

donde $p(i, j)$ = define el valor de la matriz de coocurrencia en la posición (i, j) . N_g = Niveles de grises trabajados en la región segmentada de la imagen. $p_x(i)$ = i -ésima entrada en una matriz de probabilidad original de coocurrencia que es obtenida mediante la suma de filas en $p(i, j)$. $p_y(j)$ = j -ésima entrada en una matriz de probabilidad original de coocurrencia que es obtenida mediante la suma de columnas en $p(i, j)$. R = Numero total de elementos en la matriz de coocurrencia. μ_x, μ_y = Media de P_x y P_y . σ_x, σ_y = Desviación estandar de P_x y P_y . H_x, H_y = Entropía de P_x y P_y . En total se obtuvieron 14 descriptores textuales de cada imagen.

3) *Características cromáticas*: Las características cromáticas proveen información de la intensidad del color de una región segmentada. éstas características pueden ser calculadas por cada canal de intensidad, por ejemplo, rojo, verde, azul, escala de grises, tono (Hue), saturación (Saturation) e intensidad (Value), etc. Las características que se emplearon fueron; características de intensidad estandar, éstas describen la media, desviación estandar de la intensidad, primera y segunda derivada derivada en la región segmentada, Momentos de Hu con información de intensidad, características de Gabor basadas en funciones de gabor en 2D. En los experimentos realizados se obtuvieron 122 características por cada canal. Ya que los experimentos fueron realizados en RGB solo se utilizaron 366 características cromáticas.

C. Selección de características

Con el objetivo de eliminar algunas características que no contribuyen al desempeño del clasificador, en esta investigación se utilizó un algoritmo genético para extraer las mejores combinaciones de características.

En el algoritmo propuesto cada conjunto de características por hoja, forma un vector definido por el número de descriptores o características. El número de características de cada conjunto de datos define el tamaño de cada cadena binaria necesaria para implementar el algoritmo genético. La relación que existe entre cada cadena binaria con el conjunto de características, es que el 1 se toma como característica empleada y el 0 como ausencia de esa característica. La aptitud de cada individuo se toma de la precisión que se obtiene al clasificar el conjunto correspondiente a dicha cadena.

En la metodología propuesta se toma el individuo con mejor aptitud y este pasa intacto a la siguiente generación, se utilizó cruce de dos puntos y probabilidad de mutación de 0.08. La Figura 3 muestra un ejemplo de cadenas cromosómicas utilizadas y el desempeño del clasificador al utilizar las características marcadas con 1.

La dimensionalidad del conjunto de datos es un factor importante en el desempeño de los clasificadores. Algunos atributos pueden en ocasiones afectar el desempeño del clasificador en lugar de ayudarlo. El uso adecuado de atributos ayuda a mejorar el desempeño de un clasificador. Este problema ha sido tratado por varios autores, este problema es habitual en reconocimiento de patrones y es comúnmente llamado curso de la dimensionalidad. Un factor importante al momento

de reducir características, es eliminar aquellas que no son importantes para el clasificador o encontrar la combinación de atributos que optimiza el desempeño del clasificador.

La selección de características o reducción de dimensionalidad es regularmente planteado como un problema de optimización. En los últimos años, los algoritmos genéticos (AGs) han sido extensamente utilizados para resolver problemas de optimización. Los AGs parten de la premisa de emplear la evolución natural como un procedimiento de optimización. Se caracterizan por representar las soluciones al problema que abordan en forma de cadenas binarias. Esas representaciones binarias les aporta características muy importantes de eficiencia. Sin embargo, es necesario disponer de un método para pasar esa representación binaria al espacio de búsqueda natural de cada problema.

Formalmente, dado un conjunto de patrones n -dimensional, la tarea del Algoritmo Genético es encontrar un conjunto de atributos en un espacio k -dimensional que maximice un criterio de optimización, donde $k \ll n$. Los patrones obtenidos son evaluados basados en dos condiciones la dimensionalidad del conjunto de datos y la separación entre clases o la precisión de clasificación.

El esquema general de un algoritmo genético propuesto es el siguiente:

Entrada:	Conjunto de datos X y clase t.
Salida:	Mejores atributos
1:	Crear población inicial
2:	Calcular aptitud de población inicial
3:	WHILE condición de paro no se cumpla DO
4:	Selección de individuos para la reproducción
5:	Cruza de individuos
6:	Mutación de individuos
7:	Calcular aptitud de nueva generación END

La estructura se describe con más detalle a continuación:

D. Población inicial

Para ejecutar un AG, se requiere de una población de individuos. Cada individuo, es un candidato a ser la solución del problema tratado que permite llegar a la solución.

Cada individuo de la población se representa con una cadena binaria. Los individuos de la población inicial suelen ser cadenas de 0's y 1's generados de forma completamente aleatoria, esto es importante porque permite dotar al algoritmo genético de población con suficientemente variedad, para poder explorar todas las zonas del espacio de búsqueda. Las cadenas se denominan *genotipo* del individuo y que es análoga al *cromosoma* en el sistema biológico. Cada *genotipo* representa un punto π en el espacio de búsqueda del problema. A cada punto π se le denomina *fenotipo*. Se usa el termino *gen* para referirse a la codificación de una determinada característica del individuo. Cada *gen* puede tomar distintos valores que son llamados *alelos*. Para referirse a una determinada posición de la cadena binaria se usa el término *locus*.



Fig. 3. Generación de individuos

En nuestro caso, el tamaño del cromosoma depende del número original de características que tiene el conjunto de datos. De acuerdo al número de características de cada conjunto de datos, se formó el tamaño de cada cadena binaria que se necesitó para implementar el algoritmo genético. La relación que existe entre cada cadena cromosómica con el conjunto de atributos, es que el 1 se toma como característica empleada y el 0 como ausencia de ese atributo. En la Figura 3 se muestra un ejemplo que muestra el genotipo y fenotipo las cadenas utilizadas.

E. Selección de individuos

La idea básica de selección, es utilizar una distribución de probabilidad de selección de una cadena, donde la probabilidad es directamente proporcional a la función de aptitud del individuo. La aptitud de cada individuo se toma de la precisión que se obtiene al clasificar el conjunto correspondiente con dicha cadena de atributos. Es decir, el proceso de selección debe favorecer la cantidad de copias de los individuos más adaptados. En este trabajo solo se utilizó la técnica de selección de ruleta.

F. Cruza

El proceso de cruza provee un mecanismo para heredar rasgos a su descendencia donde intervienen ambos padres, este es un método de fusión sobre la información genética de dos individuos. El mecanismo de cruza utilizado en los experimentos fue cruza de dos puntos.

G. Mutación

La mutación es un proceso donde el material genético puede ser alterado en forma aleatoria, debidamente a un error en la reproducción o la deformación de genes. A diferencia de la genética humana, la probabilidad en un algoritmo genético es mayor y evita que el algoritmo quede atrapado en mínimos locales. La forma más sencilla de mutación consiste en cambiar el valor de una de las posiciones de la cadena. Si el valor es cero pasa a uno, y si es uno pasa a cero.

H. Condición de paro

Es necesario especificar las condiciones en las que el algoritmo deja de evolucionar y se presenta la mejor solución encontrada. La condición de paro más sencilla, se presenta al detectar que la mayor parte de la población ha convergido a una forma similar, careciendo de la suficiente diversidad para que tenga sentido continuar con la evolución. En el método propuesto, el algoritmo finaliza cuando después de 10 iteraciones no hubo mejora. La cadena con mejor precisión se almacena en un archivo de texto junto con la precisión obtenida.



Fig. 4. Cadenas cromosómicas y su aptitud

I. Elitismo

De toda la población, se toma el individuo con mejor resultado de clasificación en la generación y este pasa intacto a la siguiente generación. Se continúa con un método de selección y se realiza el método de cruza con los individuos seleccionados para la nueva generación. Se realiza el mismo procedimiento para la siguiente generación, y si ahora se obtiene un mejor resultado que el anterior, está reemplaza al individuo que tenía la mejor precisión. Si no hubo cadena que mejore la aptitud, la cadena anterior vuelve a permanecer intacta en la nueva generación.

J. Técnicas de Clasificación

La clasificación, consiste en detectar o reconocer un patrón en términos de propiedades o rasgos. El reconocimiento de patrones es una de las tareas más importantes. Sin embargo, también es una de las tareas más complejas. En los experimentos realizados se compararon los resultados con tres clasificadores, se utilizó un clasificador Bayesiano, el algoritmo de aprendizaje Backpropagation y Máquinas de Vectores de Soporte (SVM).

IV. RESULTADOS EXPERIMENTALES

En esta sección se muestran la técnica de selección de parámetros, normalización de datos y los resultados experimentales obtenidos con el sistema propuesto.

A. Conjunto de datos

En los experimentos realizados se utilizó el conjunto de datos ICL, que es una colección de hojas de la Universidad de Hefei. El conjunto de datos contiene 16849 imágenes de hojas de 220 especies. Las imágenes de las hojas fueron segmentadas utilizando el método de Otsu y se extrajeron las características antes mencionadas.

Todas las características extraídas fueron normalizadas con la relación:

$$f_{ij} = \frac{T_{ij} - \mu_j}{\sigma_j}$$

donde $i = 1, \dots, m$ y $j = 1, \dots, n$, μ_j y $\sigma_j T_{ij} = \sigma_j$ representan la media y desviación estándar de la j -ésima característica, T_{ij} representa la j -ésima característica del i -ésimo vector, m es el número de imágenes y n el número de características. Las características normalizadas tienen media cero y desviación estándar igual a 1.

B. Selección de parámetros

La selección de parámetros es un paso muy importante, ya que una buena selección de parámetros tiene un efecto considerable en el desempeño del clasificador.

En todos los clasificadores utilizados se obtuvieron los parámetros óptimos mediante validación cruzada y búsqueda de malla.

C. Resultados

En los experimentos realizados, todos los conjuntos de datos fueron normalizados y se utilizó validación cruzada con $k = 10$. La Tabla 1 muestra los resultados obtenidos con características geométricas, texturales y cromáticas, cada una como características individuales. CH_i define el conjunto de datos de hojas utilizado. Para cada clasificador utilizado, se reportan las precisiones obtenidas con cada conjunto individual de características. La métrica utilizada para evaluar el desempeño del clasificador fue precisión y esta se obtiene de los aciertos del clasificador entre el total del conjunto de datos.

En los resultados obtenidos, no es posible inferir que la similitud entre hojas afecte significativamente a los clasificadores, los desempeños de los clasificadores utilizados con imágenes muy similares entre sí y disimilares, no son contrastantes. Sin embargo, es posible apreciar que las características texturales son poco discriminativas para la mayoría de los conjuntos de datos, excepto para el conjunto CH₆. Una posible razón es que el tamaño del conjunto de datos es muy pequeño (solo tres clases).

Tabla 1. Desempeños con diferentes técnicas de selección de características.

Clasificador		ODS	RG	SFS	SBS
Bayes	Acc	88.84	94.08	92.31	91.78
	TP	0.888	0.941	0.923	0.918
	Recall	0.87	0.991	0.90	0.92
BP	Acc	90.72	94.32	92.73	92.27
	TP	0.91	0.943	0.93	0.92
	Recall	0.92	0.993	0.94	0.92
SVM	Acc	91.37	95.65	92.71	92.38
	TP	0.914	0.957	0.93	0.92
	Recall	0.90	0.999	0.95	0.91
LR	Acc	82.56	89.03	85.61	86.07
	TP	0.826	0.890	0.86	0.86
	Recall	0.827	0.871	0.84	0.87

La Tabla 1 muestra los resultados obtenidos con las diferentes técnicas. En la Tabla, ODS representa los resultados obtenidos con el conjunto de datos original, RG representa los resultados obtenidos con el Genético propuesto, SFS significa selección secuencial hacia adelante y SBS Selección secuencial hacia atrás. En los resultados es posible apreciar una mejora en la precisión de clasificación en comparación con los desempeños obtenidos con el conjunto de datos original y en comparación con las otras técnicas de selección de atributos. En todas las pruebas con los diferentes clasificadores, los resultados con la técnica propuesta mejoran los resultados

obtenidos con las otras técnicas. Estos resultados permiten resaltar la utilidad del método propuesto. En los resultados obtenidos se mejoró la precisión en todas las cadenas al utilizar el genético, es importante señalar que aún cuando el número de características disminuyó significativamente, en todos los resultados obtenidos la combinación de los tres tipos de características es muy necesaria.

V. CONCLUSIONES

En este artículo, se propone un algoritmo de selección de características para mejorar el desempeño de clasificadores para identificación de plantas a partir de hojas. El método propuesto ayuda a mejorar el desempeño del clasificador al eliminar atributos que introducen ruido al clasificador. El método propuesto utiliza un algoritmo genético para eliminar atributos utilizando como aptitud una métrica de desempeño. El método propuesto realiza una búsqueda de los mejores atributos y/o combinación de mejores atributos, eliminando aquellos atributos que afectan su desempeño. Los experimentos obtenidos muestran que el método propuesto genera notables resultados al eliminar atributos que no aportan información. La principal ventaja del método propuesto es su facilidad de implementación y su fácil uso sobre conjuntos de datos de pequeño y mediano tamaño. La reducción de características es importante para mejorar los tiempos de respuesta que tarda el sistema para reconocer una nueva hoja. Los resultados obtenidos muestran que el sistema de reconocimiento empleado mejora en todos los casos la precisión obtenida con todas las características al eliminar características individuales o combinación de características que introducen ruido al clasificador.

RECONOCIMIENTOS

Este estudio fue financiado por la Secretaría de Investigación de la Universidad Autónoma del Estado de México con el proyecto de investigación 3771/2014/CIB.

REFERENCIAS

- [1] C.C. Vennera, D.M. Cooper, A Review of Content-based Image Retrieval Systems, Technical Report, Manchester Visualization Centre, Manchester Computing, University of Manchester, Manchester, UK, 2000.
- [2] A.W.M. Smeulders, M. Worring, S. Sanini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (2000) 1349-1380.
- [3] M. Theo, T. Haverizen, P. Kucsmann, A method of color histogram creation for image retrieval, in: Proceedings of the Nordic Signal Processing Symposium (NORSIG-2000), Kolmarden, Sweden, 2000, pp. 157-160.
- [4] Chaki, J., Fureth, R. (2012). "Designing an automated system for plant leaf recognition" *International Journal of Advances in Engineering Technology*, 2(1), 149-158.
- [5] J.S. Park, T.Y. Kim, "Shape-based image retrieval using invariant features", in: K. Aizawa, Y. Nakamura, S. Satoh, (Eds.), *Advances in Multimedia Information Processing-PCM 2004*, Berlin/Heidelberg, Lecture Notes in Computer Science, 2004, pp. 146-153.
- [6] Ji-Xiang Du, Xiao-Feng Wang, Guo-Jun Zhang, Leaf shape based plant species recognition, *Applied Mathematics and Computation*, Volume 185, Issue 2, 15 February 2007, Pages 883-890, ISSN 0096-3003.
- [7] Kadir, A., Nugroho, L. E., Susanto, A., Santosa, P. I. (2012a). "Experiments of distance measurements in a foliage plant retrieval system", *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 5, 256-263.

- [8] Kaur, G., Kaur, G. (2012). "Classification of biological species based on leaf architecture". *International Journal of Engineering Research and Development*, 1:35-42.
- [9] N. Kumar, P.N. Belhumeur, A. Biswas, et al. Leafsnap: a computer vision system for automatic plant species identification. in: *Proc. ECCV 2012*, 2012, pp. 502-516.
- [10] C.S. Sumathi, A.V.S. Kumar, "Edge and texture fusion for plant leaf classification", *International Journal of Computer Science and Telecommunications*, 3 (2012), pp. 6-9.
- [11] D.J. Hees "Shape analysis for the automated identification of plants from images of leaves" *Taxon*, 58 (2009), pp. 934-954.
- [12] James S. Cope, David Corney, Jonathan Y. Clark, Paolo Remagnino, Paul Wilkin, Plant species identification using digital morphometrics: A review, *Expert Systems with Applications*, Volume 39, Issue 8, 15 June 2012, Pages 7562-7573.
- [13] W. Ma, B. Manjunath, Texture features and learning similarity, in: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 1996, pp. 425-430.
- [14] B. Manjunath, W. Ma, Texture features for browsing and retrieval of image data, *IEEE Trans. Pattern Anal. Mach. Intell.* 18 (1996) 837-842.
- [15] Anders Brandstadt, Van Bang Le, Structure and linear time recognition of 3-leaf powers, *Information Processing Letters*, Volume 98, Issue 4, 31 May 2006, Pages 133-138.
- [16] M.K. Hu, Visual pattern recognition by moment invariants, *IRE Trans. Inform. Theory*, 8 (1962), pp. 179-187.
- [17] Guillermo Sampallo, Reconocimiento de tipos de hojas. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, vol. 7, num. 21, 2003, pp. 35-62. *Asociacion Espanola para la Inteligencia Artificial Espana*.
- [18] Ji-Xiang Du, Xiao-Feng Wang, Guo-Jun Zhang, Leaf shape based plant species recognition, *Applied Mathematics and Computation*, Volume 183, Issue 2, 15 February 2007, Pages 883-893, ISSN 0096-3003.
- [19] Shaowen Zhang, Ying-Ke Lei, Modified locally linear discriminant embedding for plant leaf recognition, *Neurocomputing*, Volume 74, Issues 14-15, July 2011, Pages 2284-2290, ISSN 0925-2312.
- [20] H. Muhammad Araf, M.T. Noorizwati, M.S.B. Shah Rizam, A Comparative Study in Kernel-Based Support Vector Machine of Oil Palm Leaves Nutrient Disease, *Procedia Engineering*, Volume 41, 2012, Pages 1353-1359, ISSN 1877-7058.
- [21] Z. Husn, A.Y.M. Shakaff, A.H.A. Aziz, R.S.M. Farook, M.N. Jafar, U. Hashim, A. Harun, Embedded portable device for herb leaves recognition using image processing techniques and neural network algorithm, *Computers and Electronics in Agriculture*, Volume 89, November 2012, Pages 13-19, ISSN 0168-1699.
- [22] Rui Hu, John Colomese, A performance evaluation of gradient field HOG descriptor for sketch based image retrieval, *Computer Vision and Image Understanding*, Volume 117, Issue 7, July 2013, Pages 790-806, ISSN 1077-3142.
- [23] Shaowen Zhang, Yingke Lei, Tianbao Dong, Xiao-Feng Zhang, Label propagation based supervised locality projection analysis for plant leaf classification, *Pattern Recognition*, Volume 46, Issue 7, July 2013, Pages 1891-1897, ISSN 0031-3203.
- [24] Petr Novotny, Tomas Suk, Leaf recognition of woody species in Central Europe, *Biosystems Engineering*, Volume 115, Issue 4, August 2013, Pages 444-452, ISSN 1537-5110.
- [25] Chunlei Xia, Jang Myung Lee, Yan Li, Yoo-Han Song, Bu-Keun Chung, Tae-Soo Chon, Plant leaf detection using modified active shape models, *Biosystems Engineering*, Volume 116, Issue 1, September 2013, Pages 23-35, ISSN 1537-5110.
- [26] Ji-xiang Du, Chuan-Min Zhai, Qing-Fang Wang, Recognition of plant leaf image based on fractal dimension features, *Neurocomputing*, Volume 116, 20 September 2013, Pages 150-156, ISSN 0925-2312.
- [27] Chih-Ying Gwo, Chia-Hung Wei, Yue Li, Rotary matching of edge features for leaf recognition, *Computers and Electronics in Agriculture*, Volume 91, February 2013, Pages 124-134, ISSN 0168-1699.
- [28] Guillaume Cerutti, Laure Tougnon, Julien Mille, Antoine Vacavant, Didier Coquin, Understanding leaves in natural images - A model-based approach for tree species identification, *Computer Vision and Image Understanding*, Volume 117, Issue 10, October 2013, Pages 1482-1501, ISSN 1077-3142.
- [29] Monica G. Larese, Rafael Namias, Roque M. Craviotto, Miriam R. Arango, Carina Gallo, Pablo M. Granitto, Automatic classification of legumes using leaf vein image features, *Pattern Recognition*, Volume 47, Issue 1, January 2014, Pages 158-168, ISSN 0031-3203.
- [30] Monica G. Larese, Ariel E. Baya, Roque M. Craviotto, Miriam R. Arango, Carina Gallo, Pablo M. Granitto, Multiscale recognition of legume varieties based on leaf venation images, *Expert Systems with Applications*, Volume 41, Issue 10, August 2014, Pages 4638-4647, ISSN 0957-4174.
- [31] Ch. Miza, G.Ch. Srakoulis, V. Mardiris, I. Karafyllidis, N. Glykos, R. Sandaltonopolos, Reconstruction of DNA sequences using genetic algorithms and cellular automata: Towards mutation prediction?, *Biosystems*, Volume 92, Issue 1, pp 61-68, April 2008.
- [32] R. Mokhtab-Rafei, S.M. Manzari, S. Bostanian, Financial health prediction models using artificial neural networks, genetic algorithm and multivariate discriminant analysis: Iranian evidence, *Expert Systems with Applications*, Volume 38, Issue 8, pp 10210-10217, August 2011.
- [33] Antonio C. Caputo, Pacifico M. Pelagaggi, Mario Palumbo, Economic optimization of industrial safety measures using genetic algorithms, *Journal of Loss Prevention in the Process Industries*, Vol. 24, Issue 5, pp 541-551, September 2011.
- [34] Rui Zhang, Pei-Chann Chang, Cheng Wu, A hybrid genetic algorithm for the job shop scheduling problem with practical considerations for manufacturing costs: Investigations motivated by vehicle production, *International Journal of Production Economics*, Available online 8 November 2012, ISSN 0925-5273.
- [35] Jeremy Breen, P. de Souza, G.P. Tamra, R. Ollington, Onboard assessment of XRF spectra using genetic algorithms for decision making on an autonomous underwater vehicle, *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, Vol. 269, Issue 12, pp 1341-1345, 15 June 2011.
- [36] Ragib Kokar, A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization, *Information Sciences*, Vol 222, pp 528-543, 10 February 2013.
- [37] K. Asan Mohideen, G. Saravarakumar, K. Valarmathi, D. Devaraj, T.K. Radhakrishnan, Real-coded Genetic Algorithm for system identification and tuning of a modified Model Reference Adaptive Controller for a hybrid tank system, *Applied Mathematical Modelling*, Vol 37, Issue 6, pp 3829-3847, 15 March 2013.
- [38] Amelia Zafra, Mykola Pechenizkyi, Sebastian Ventura, HyDR-M: A hybrid algorithm to reduce dimensionality in multiple instance learning, *Information Sciences*, Volume 222, Pages 282-301, 10 February 2013.
- [39] Jui-Sheng Chou, Min-Yuan Cheng, Yu-Wei Wu, Improving classification accuracy of project dispute resolution using hybrid artificial intelligence and support vector machine models, *Expert Systems with Applications*, Volume 40, Issue 6, pp 2263-2274, May 2013.
- [40] Bilash Kant Sarker, Shub Sanjay Saha, Kripasindhu Chandruri, Selecting informative rules with parallel genetic algorithm in classification problem, *Applied Mathematics and Computation*, Volume 218, Issue 7, Pages 3247-3264, 1 December 2011.
- [41] Yakoub Bani and Farid Melgani, Semisupervised PSO-SVM Regression for Biophysical Parameter Estimation, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 45, No. 6, June 2007.
- [42] Mario Koppen, The Curse of Dimensionality, 5th Online World Conference on Soft Computing in Industrial Applications (WSC5), September 4-18 2000.

Bibliografía

- [Alaei et al, 2016] Alireza Alaei, Partha Pratim Roy, Umapada Pal, Logo and seal based administrative document image retrieval: A survey, *Computer Science Review*, Volume 22, November 2016, Pages 47-63, ISSN 1574-0137.
- [Alcantara, 2005] Alcántara, J. C. (2005). El encanto audiovisual en la política: el spot político electoral televisivo. Flacso, México.
- [Amparan, 2010] Amparán, A. C. (2010). El framing audiovisual del spot político. *Cultura y Representaciones Sociales*, 5-9.
- [Amparan:2011] Amparán, A. C. (2011). El framing del Spot Político. México D.F.: Porrúa.
- [Bay et al, 2006] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *Proceedings of the ninth European Conference on Computer Vision*, May 2006.
- [Bermeo:2006] Bermeo Campos Nadia Lizbeth, C. B. (2006). Robots Controlados por Internet para la Localización de Objetos. *Revista Digital Universitaria DGSCA-UNAM*, 7- 8.
- [Bianco et al, 2017] Simone Bianco, Marco Buzzelli, Davide Mazzini, Raimondo Schettini, Deep learning for logo recognition, *Neurocomputing*, Volume 245, 5 July 2017, Pages 23-30, ISSN 0925-2312.
- [Brown and Lowe, 2003] Brown, M. Lowe, D. G.: "Recognising panoramas." In: *Computer Vision (ICCV)*., 2003, S. 1218
- [Burak et al, 2016] Burak Altinoklu, Ilkay Ulusoy, Sibel Tari, A probabilistic sparse skeleton based object detection, *Pattern Recognition Letters*, Volume 83, Part 3, 1 November 2016, Pages 243-250, ISSN 0167-8655.

- [Carrasco, 2010] Carrasco, Jorge (2010). Cine y televisión digital. Manual técnico. Barcelona: Edicions de la Universidad de Barcelona. ISBN 978-84-475-3457-9.
- [Chen et al, 2017] Yaran Chen, Dongbin Zhao, Le Lv, Qichao Zhang, Multi-task learning for dangerous object detection in autonomous driving, *Information Sciences*, Available online 10 August 2017, ISSN 0020-0255.
- [Chu y Cai, 2017] Wenqing Chu, Deng Cai, Deep feature based contextual model for object detection, *Neurocomputing*, Available online 21 September 2017, ISSN 0925-2312.
- [Dalal y Triggs, 2014] Dalal, N., Triggs, B. (2014). Histograms of Oriented Gradients for Human Detection. *INRIA, Rhone-Alps*, 384-392.
- [Diligenti et al, 2001] M. Diligenti, M. Gori, M. Maggini, E. Martinelli, Adaptive graphical pattern recognition for the classification of company logos, *Pattern Recognition*, Volume 34, Issue 10, October 2001, Pages 2049-2061, ISSN 0031-3203.
- [Domke et al, 2006] J. Domke and Y. Aloimonos, "Deformation and viewpoint invariant color histograms," in *British Machine Vision Conference*, vol. 2, pp. 509–518, 2006.
- [Dorko and Schmid, 2003] Dorkó, G. Schmid, C.: "Selection of scale-invariant parts for object class recognition." In: *Computer Vision (ICCV), Proceedings. Ninth IEEE International Conference on.*, 2003, S. 634—639
- [Fergus et al, 2003] Fergus, R.; Perona, P. Zisserman, A.: "Object class recognition by unsupervised scale-invariant learning." In: *Computer Vision and Pattern Recognition (CVPR), Proceedings. IEEE Computer Society Conference on.*, 2003, S. II-264.
- [Gori et al, 2003] M. Gori, M. Maggini, S. Marinai, J.Q. Sheng, G. Soda, Edge-backpropagation for noisy logo recognition, *Pattern Recognition*, Volume 36, Issue 1, January 2003, Pages 103-110, ISSN 0031-3203.

- [Huang et al, 1997] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), pp. 762–768, IEEE Computer Society, 1997.
- [Konigsberg, 2005] Konigsberg, Ira (2004). Diccionario técnico Akal de cine. Madrid: Akal. ISBN 9788446019022.
- [Leibe et al, 2004] Leibe, B. Schiele, B.: "Scale-invariant object categorization using a scale-adaptive mean-shift search: Springer." Pattern Recognition (CVPR), on., 2004, S. 145—153
- [Liu et al, 2009] Liu, J.; Luo, J. Shah, M.: "Recognizing realistic actions from videos "in the wild". In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on., 2009, S. 1996—2003
- [Lowe, 1999] Lowe, D.: "Object recognition from local scale-invariant features" .: In: Computer Vision (ICCV) . The Proceedings of the Seventh IEEE International Conference on., 1999, S. 1150-1157 Vol. 2
- [Lopez, 2012] López, C. V. (2012). Los spots políticos televisivos de los candidatos a la Presidencia de México en las elecciones del año 2012. Universidad Autónoma de Tlaxcala.
- [Llobregat, 2011] Llobregat, R. R. (Noviembre de 2011). Procesado Digital de Imágenes de video para la detección de humo. Universidad Politecnica de Valencia. Tesis de grado.
- [Mao et al, 2013] Songan Mao, Mao Ye, Xue Li, Feng Pang, Jinglei Zhou, Rapid vehicle logo region detection based on information theory, Computers Electrical Engineering, Volume 39, Issue 3, April 2013, Pages 863-872, ISSN 0045-7906.
- [Marcovecchio et al, 2013] Marcovecchio, D., Stefanazzi, N., Delrieux, C., Maguitman, A., Ferrero, A. (2013). Tracking de múltiples objetos aplicado a insectos. XV WorkShop de Investigadores en Ciencias de la Computación.
- [Min et al, 2018] Weidong Min, Yu Zhang, Jing Li, Shaoping Xu, Recognition of pedestrian activity based on dropped-object detection, Signal Processing, Volume 144, March 2018, Pages 238-252, ISSN 0165-1684.

- [Muñoz, 2015] Muñoz, L. M. (2015). Del voto corporativo al marketing político, Su introducción en México Marketing Politico.
- [Nowak et al, 2006] Nowak, E.; Jurie, F. Triggs, B.: "Sampling strategies for bag-of-features image classification: Springer." : Computer Vision (ICCV) on., 2006, S. 490—503
- [Pla, 2013] Pla, I. L. (2013). El encanto audiovisual en la política: el spot político-electoral televisivo. Investigaciones Jurídicas de la UNAM.
- [Reyes, 2008] Reyes, J. (2008). Reconocimiento de Patrones . Stanford University.
- [Rhee et al, 2017] Phill Kyu Rhee, Enkhbayar Erdenee, Shin Dong Kyun, Minhaz Uddin Ahmed, Songguo Jin, Active and semi-supervised learning for object detection with imperfect data, Cognitive Systems Research, Volume 45, October 2017, Pages 109-123, ISSN 1389-0417.
- [Shanon, 1948] Shannon C. E. A mathematical theory of communication. Bell System Technical Journal, 27:379-423, 623-656, 1948.
- [Sivic and Zisserman, 2003] Sivic, J. Zisserman, A.: "Video Google: A text retrieval approach to object matching in videos." In: Computer Vision (ICCV. Ninth IEEE International Conference on., 2003, S. 1470-1477
- [Tang y Peng, 2017] Panpan Tang, Yuxin Peng, Exploiting distinctive topological constraint of local feature matching for logo image recognition, Neurocomputing, Volume 236, 2 May 2017, Pages 113-122, ISSN 0925-2312.
- [Tieu and Viola, 2004] Tieu K. and Viola P. Boosting image retrieval, volume 56, pages 781-796, 2004.
- [Torres, 1996] Torres, A. D. (1996). Procesamiento digital de imágenes. Perfiles Educativos, , 2-11. Vargas, A. M. (2013). Reconocimiento de Patrones .
- [Toshev et al, 2010] Toshev, A., Makadia, A., Daniilidis, K. (2010). Shape-based Object Recognition in Videos Using 3D Synthetic Object Models. IEEE International, 220-228.

- [Viola and Jones, 2001] Viola P. and Jones M. Rapid object detection using a boosted cascade of simple features. In Proceedings IEEE Conf on Computer Vision and Pattern Recognition, pages 511U-518, Kauai, Hawaii, 2001.
- [Wang et al, 2009] Wang, X., Tony , H., Shuicheng, Y. (Noviembre 2009). An HOG-LBP Human Detector with Partial Occlusion Handling. Proceedings IEEE Internacional Conference on Computer Vision.
- [Wang et al, 2017] Gang Wang, Yongdong Zhang, Jintao Li, High-level background prior based salient object detection, Journal of Visual Communication and Image Representation, Volume 48, October 2017, Pages 432-441, ISSN 1047-3203,
- [Wei et al, 2017] Hui Wei, Chengzhan Yang, Qian Yu, Contour segment grouping for object detection, Journal of Visual Communication and Image Representation, Volume 48, October 2017, Pages 292-309, ISSN 1047-3203.
- [Xiao et al, 2017] Fen Xiao, Liangchan Peng, Lei Fu, Xieping Gao, Salient object detection based on eye tracking data, Signal Processing, Volume 144, March 2018, Pages 392-397, ISSN 0165-1684.
- [Xu G. et al, 2017] Guili XU, Zhengbing WANG, Yuehua CHENG, Yupeng TIAN, Chao ZHANG, A man-made object detection algorithm based on contour complexity evaluation, Chinese Journal of Aeronautics, Volume 30, Issue 6, December 2017, Pages 1931-1940, ISSN 1000-9361.
- [Xu et al, 2017] Xiangyang Xu, Yuncheng Li, Gangshan Wu, Jiebo Luo, Multi-modal deep feature learning for RGB-D object detection, Pattern Recognition, Volume 72, December 2017, Pages 300-313, ISSN 0031-3203.
- [Yaseen et al, 2018] Muhammad Usman Yaseen, Ashiq Anjum, Omer Rana, Richard Hill, Cloud-based scalable object detection and classification in video streams, Future Generation Computer Systems, Volume 80, March 2018, Pages 286-298, ISSN 0167-739X.
- [Yuan et al, 2017] Zehuan Yuan, Tong Lu, Chew Lim Tan, Learning discriminated and correlated patches for multi-view object detection using sparse coding, Pattern Recognition, Volume 69, September 2017, Pages 26-38, ISSN 0031-3203.

- [Tom, 2006] . Zean Yuan, Tong Lu, Chew Lim Tan, rse coding, Pattern Recognition, Volume 69, September 2017, .