

Instituto Politécnico Nacional  
Centro de Investigación en Computación



# **Discriminación robusta de imágenes basada en momentos geométricos y el clasificador Gamma**

TESIS

Que para obtener el grado de  
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

Presenta:

**CARLOS OMAR CHÁVEZ SÁNCHEZ**

Directores de tesis:

**Dr. Itzamá López Yáñez**  
**Dr. Amadeo José Argüelles Cruz**

México, D.F., Octubre de 2012



# INSTITUTO POLITÉCNICO NACIONAL

## SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

### ACTA DE REVISIÓN DE TESIS

En la Ciudad de      México, D.F.      siendo las     12:00     horas del día     02     del mes de     Mayo     de     2012     se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

**Centro de Investigación en Computación**

para examinar la tesis titulada:

**“Discriminación robusta de imágenes basada en momentos geométricos y el clasificador Gamma”**

Presentada por el alumno:

**CHÁVEZ**

Apellido paterno

**SÁNCHEZ**

Apellido materno

**CARLOS OMAR**

Nombre(s)

Con registro:

B	0	2	0	8	8	5
---	---	---	---	---	---	---

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

### LA COMISIÓN REVISORA

Directores de Tesis

Dr. Amadeo José Argüelles Cruz

Dr. Itzamá López Yáñez

Dr. Edgardo Manuel Felipe Riverón

Dr. Olexiy Pogrebnyak

Dr. Cornelio Yáñez Márquez

Dr. Oscar Camacho Nieto

PRESIDENTE DEL COLEGIO DE PROFESORES



Dr. Luis Alfonso Villa

INSTITUTO POLITÉCNICO NACIONAL  
CENTRO DE INVESTIGACION

DIRECCION



**INSTITUTO POLITÉCNICO NACIONAL**  
**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

**CARTA CESIÓN DE DERECHOS**

En la Ciudad de México el día 22 del mes Junio del año 2012, el (la) que suscribe Carlos Omar Chávez Sánchez alumno (a) del Programa de Maestría en Ciencias de la Computación con número de registro B020885, adscrito a Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Itzamá López Yáñez y Dr. Amadeo José Argüelles Cruz y cede los derechos del trabajo intitulado Discriminación robusta de imágenes basada en momentos geométricos y el clasificador gamma al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección carcha@mail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Carlos Omar Chávez Sánchez

Nombre y firma

## Dedicatoria

*A mis padres  
por brindarme una maravillosa familia,  
y porque sin su esfuerzo, dedicación y amor,  
no sería nadie en la vida...*

## Agradecimientos

Agradezco a mis asesores *Dr. Itzamá López Yáñez* y *Dr. Amadeo José Argüelles Cruz*, quienes depositaron en mí su confianza y amistad, por brindarme su apoyo y haberme guiado a través de la realización de esta tesis.

De forma especial, agradezco al Dr. Cornelio Yáñez Márquez quién me ayudó a mantener aún viva la esperanza para poder seguir, pero sobretodo a ti Itzamá, por ayudarme a hallar por fin de nuevo mi camino, por mucho tiempo escasamente divisado. Estoy en deuda contigo.

A mis sinodales Dr. Cornelio Yáñez Márquez, Dr. Edgardo Felipe Riverón, Dr. Oleksiy Pogrebnyak y Dr. Óscar Camacho Nieto, por su gran disposición, por sus excelentes aportaciones y crítica constructiva.

También extendo mi agradecimiento a todos los compañeros de estudio y trabajo por los gratos momentos; y al Colegio de profesores, no solo por haberme formado en el CIC, sino porque gracias a ellos pude, más allá de lo académico, conocer un aspecto más de la condición humana.

Agradezco al *IPN*, *CONACYT* y a la *SIP* por haberme apoyado tanto económica como moralmente en la realización de mis estudios de posgrado.

Finalmente, agradezco al *Centro de Investigación en Computación* por todos los años de formación que me ha brindado, por esta gran experiencia.

## Resumen

En el presente trabajo de tesis se muestra el desempeño del clasificador Gamma al ser utilizado con imágenes binarias. Adicionalmente, se pone a prueba el desempeño de los momentos invariantes al ser empleados con el clasificador Gamma como principales descriptores de los objetos en los experimentos realizados.

Las pruebas realizadas con los momentos como descriptores de objetos, pretenden mostrar, además del alto desempeño del clasificador Gamma, la alta discriminación que pueden tener los rasgos obtenidos al ser utilizados como alternativas de clasificación que permiten la identificación de objetos en casos de clases traslapadas o con elementos dispersos.

Aunque los experimentos realizados utilizan la definición de momentos invariantes propuesta originalmente por Hu, la metodología puede ser utilizada con cualquier otra técnica de cálculo de momentos existente.

Los resultados manifiestan que el nuevo algoritmo propuesto para el clasificador Gamma es eficiente y que, en general, dicho clasificador muestra un desempeño competitivo para los conjuntos de datos expuestos. Además, los descriptores probados presentan alta discriminación al obtenerse porcentajes de recuperación altos en los experimentos realizados.

## **Abstract**

This thesis shows the performance of the Gamma classifier to be used with binary images. Additionally, put to the test performance of invariant moments to be employed with the binder as main Gamma characterizing of the objects in the experiments.

Tests using the invariant moments as shape descriptors of objects intend to show, in addition to the high performance of the Gamma classifier, the high discrimination that descriptors can get derived from classification alternatives, which allows the identification of objects in cases of elements with overlapping or dispersed classes.

Although the experiments are based on the calculation of invariant moments, originally proposed by Hu, the methodology can be used with any other technique of calculation of existing moments.

The results show that the new proposed algorithm for the Gamma classifier is efficient and that, in general, the binder shows a competitive performance for any problem. In addition, the descriptors are proven high discrimination to be obtained recovery rates high among the experiments.

# Índice general

<i>Dedicatoria</i> .....	iv
<i>Agradecimientos</i> .....	v
<i>Resumen</i> .....	vi
<i>Abstract</i> .....	vii
<i>Índice general</i> .....	viii
<i>Índice de tablas</i> .....	x
<i>Índice de figuras</i> .....	x
<i>Capítulo 1</i> .....	1
<i>Introducción</i> .....	1
1.1 <i>Objetivo general</i> .....	2
1.1.1 <i>Objetivos específicos</i> .....	2
1.2 <i>Motivación</i> .....	2
1.3 <i>Planteamiento del problema</i> .....	3
1.4 <i>Contribuciones</i> .....	4
1.5 <i>Organización del documento</i> .....	4
<i>Capítulo 2</i> .....	5
<i>Antecedentes</i> .....	5
2.1 <i>Momentos invariantes</i> .....	6
2.2 <i>Momentos radiales</i> .....	6
2.3 <i>Momentos rotacionales</i> .....	6
2.4 <i>Momentos complejos</i> .....	6
2.5 <i>Momentos ortogonales</i> .....	6
2.6 <i>Otros trabajos relevantes</i> .....	8
<i>Capítulo 3</i> .....	10
<i>Marco teórico</i> .....	10
3.1 <i>Momentos geométricos</i> .....	10
3.1.1 <i>Propiedades de los momentos</i> .....	11
3.2 <i>Momentos invariantes</i> .....	13
3.2.1 <i>Invariantes a traslaciones: momentos centrales</i> .....	13
3.2.2 <i>Invariantes a cambios de escala: momentos normalizados</i> .....	14
3.2.3 <i>Invariantes a rotaciones</i> .....	14
3.2.4 <i>Invariantes de Hu</i> .....	14
3.3 <i>Otras características invariantes de los momentos</i> .....	17
3.4 <i>Técnicas de clasificación</i> .....	18
3.4.1 <i>Vecino más cercano (Nearest Neighbour-NN): IB1</i> .....	18
3.4.2 <i>Naïve Bayes</i> .....	19
3.4.3 <i>Bagging</i> .....	20
3.4.4 <i>Tablas de decisión (Decision tables)</i> .....	21
3.4.5 <i>Árboles de decisión (BF-Tree)</i> .....	23
<i>Capítulo 4</i> .....	25
<i>Materiales y métodos</i> .....	25

4.1	Clasificador Gamma .....	25
4.1.1	Operador alfa y operador beta.....	25
4.1.2	Operador $u_\beta$ .....	26
4.1.3	Módulo y congruencia.....	26
4.1.4	Código binario Johnson-Möbius modificado.....	26
4.1.5	Operador Gamma de similitud.....	27
4.1.6	Algoritmo del clasificador Gamma .....	27
4.2	Evaluación del desempeño de clasificación .....	29
Capítulo 5	.....	31
	Método propuesto .....	31
5.1	Algoritmo del clasificador Gamma simplificado.....	31
Capítulo 6	.....	35
	Resultados y discusión .....	35
6.1	Desempeño del clasificador Gamma.....	37
6.2	Factor de olvido.....	38
6.3	Rendimiento del algoritmo propuesto.....	41
6.4	Discusión.....	43
Capítulo 7	.....	45
	Conclusiones y trabajo futuro.....	45
7.1	Trabajo futuro.....	46
Referencias	.....	47

## Índice de tablas

<i>Tabla 3.1.1. Tabla de momentos de la imagen de la Figura 3.1.1</i> .....	16
<i>Tabla 4.1. Definición de los operadores <math>\alpha</math> y <math>\beta</math></i> .....	26
<i>Tabla 6.1. Relación de conjuntos de datos</i> .....	35
<i>Tabla 6.2. Clasificadores evaluados</i> .....	36
<i>Tabla 6.3. Porcentajes de recuperación de los clasificadores analizados</i> .....	37
<i>Tabla 6.4. Porcentajes de recuperación de los clasificadores analizados para estimar el factor de olvido</i> .....	39
<i>Tabla 6.5. Registro de tiempos truncando decimales para un patrón conocido</i> .....	41
<i>Tabla 6.6. Registro de tiempos truncando decimales para un patrón desconocido</i> .....	41
<i>Tabla 6.7. Registro de tiempos con distinto número de rasgos para un patrón conocido</i> .....	42
<i>Tabla 6.8. Registro de tiempos con distinto número de rasgos para un patrón desconocido</i> .....	42

## Índice de figuras

<i>Figura 3.1.1. Región de interés representada por una imagen binaria</i> .....	15
<i>Figura 3.1.2. Objetos de tamaño 100x100 en formato binario, a) rectángulo y b) triángulo</i> .....	16
<i>Figura 3.1.3. Representación gráfica de los objetos evaluados de la Figura 3.1.2.a y 3.1.2.b</i> .....	16
<i>Figura 4.1. k-fold cross validation, con k=4</i> .....	30
<i>Figura 6.1. Gráfica comparativa del desempeño de los clasificadores</i> .....	38
<i>Figura 6.2. Gráfica vertical del desempeño de los clasificadores</i> .....	38
<i>Figura 6.3. Gráfica comparativa del desempeño de los clasificadores para estimar el factor de olvido</i> .....	40
<i>Figura 6.4. Gráfica vertical del desempeño de los clasificadores para estimar el factor de olvido</i> .....	40
<i>Figura 6.5. Gráfica de los tiempos de con distinto nivel de truncamiento para un patrón conocido</i> .....	42
<i>Figura 6.6. Gráfica de los tiempos de con distinto nivel de truncamiento para un patrón desconocido</i> .....	42
<i>Figura 6.7. Gráfica de los tiempos de ejecución considerando distinto número de rasgos para un patrón conocido</i> .....	43
<i>Figura 6.8. Gráfica de los tiempos de ejecución considerando distinto número de rasgos para un patrón desconocido</i> .....	43

# Capítulo 1

## *Introducción*

El reconocimiento de formas es uno de los problemas más importantes en el campo de la visión por computadora. Uno de los problemas básicos en el proceso de reconocimiento de patrones se relaciona con la selección de un conjunto apropiado de atributos numéricos que modele las características de un objeto de interés para propósitos de clasificación [1][2]. De esta manera, las técnicas eficientes de reconocimiento de objetos para la extracción de características únicas de un objeto dado, para su representación y comparación, son de vital importancia en el área del reconocimiento de patrones. Dentro de las técnicas más ampliamente utilizadas debido a su gran aceptación está la basada en momentos [3]-[6].

Cuando se habla de momentos, difícilmente se puede encontrar una única interpretación para estos, al menos no una que sea universalmente utilizada, ya que su significado depende ampliamente del área de aplicación o del contexto en el que se encuentre. Por ejemplo, los momentos tienen una interpretación distinta si se les usa en mecánica clásica o en física, en estadística o en técnicas de conteo, o bien, cuando son utilizados como propiedades numéricas de una imagen en la descripción de características de un "objeto" para su reconocimiento [7][12][13].

En general, los momentos exhiben cantidades numéricas en alguna distancia de un punto o eje de referencia. Son comúnmente usados en estadística para caracterizar la distribución de variables aleatorias; similarmente, en mecánica para caracterizar cuerpos mediante su distribución espacial de masa. El uso de los momentos para el análisis de imágenes es simple, si se considera a una imagen binaria o segmento de una imagen en niveles de gris como una función de distribución de densidad de dos dimensiones. De esta forma, los momentos pueden ser usados para caracterizar un segmento de imagen y extraer propiedades que tengan analogías en estadística y en mecánica [14][36][37].

Los momentos proveen características de un objeto y que representan unívocamente su forma, y han sido extensamente utilizados como descriptores invariantes de una imagen en el reconocimiento de patrones desde 1960 [6][8][11]-[17]. Algunos momentos, o sus derivados, tienen la capacidad de ser invariantes ante ciertas transformaciones como cambios de escala, rotación, traslación, proyecciones, cambios de contraste, ante la presencia de ruido, entre otras [5][7][13][18]-[22]

Un conjunto de momentos invariantes (realmente invariantes derivados de momentos geométricos), inicialmente propuesto por Hu [7][11], es representado por una tupla de siete momentos invariantes derivados a partir de los tres primeros momentos geométricos

centrales. Esta caracterización presenta la propiedad de ser invariante a cambios de escala, rotación y traslación de un objeto.

Aunque es posible la extensión del modelo presentado por Hu, para la obtención de los siguientes momentos de mayor orden (mayores que tres) [23]-[25], con la finalidad de incrementar el número de características de un objeto, estos presentan bajo desempeño en términos de redundancia de información, sumado al error inherente de la aproximación de las operaciones de su cálculo, lo cual conduce a la baja eficacia de la discriminación [10][14][25][26].

El uso de momentos invariantes ha sido ampliamente propuesto en una gran variedad de trabajos de investigación, desde la identificación de aeronaves militares [15], en el reconocimiento de caracteres escritos [27][28], en tareas de identificación de caracteres ópticos e impresos [18][29], hasta el punto de ser contemplados en técnicas de reconocimiento de objetos en imágenes de tres dimensiones [30], en tareas de biometría para la identificación de huellas dactilares [16], palma de la mano [17] y facial [31]; y recientemente para tareas con imágenes médicas [32]. Estos, en general, han mostrado buen desempeño como descriptores. Esto motiva su utilización como caracterizadores en el presente trabajo de tesis para mostrar el desempeño del clasificador Gamma en tareas de identificación que involucran imágenes binarias.

## **1.1 Objetivo general**

Probar el desempeño del clasificador Gamma en problemas que involucran imágenes binarias mediante un estudio comparativo con algunos de los clasificadores más representativos de los diferentes enfoques en el reconocimiento de patrones, utilizando además un nuevo algoritmo del clasificador.

### **1.1.1 Objetivos específicos**

- Mostrar el desempeño del clasificador Gamma en problemas que involucren imágenes binarias
- Realizar un estudio comparativo del clasificador Gamma con clasificadores de distintos enfoques utilizando como caracterizadores los momentos invariantes
- Presentar un nuevo algoritmo del clasificador Gamma el cual elimina parte de las operaciones de codificación de componentes para simplificar su procesamiento

## **1.2 Motivación**

Un problema importante en visión por computadora es el modelado de formas para su reconocimiento. Una manera de describir este problema es la siguiente: dada una función  $f: X \rightarrow \mathcal{R}$ , hallar un conjunto finito de valores que caracterice a  $f$  en forma unívoca, es decir,

que el conjunto de valores debe ser único, y no debe existir  $g \neq f$  que pueda ser caracterizado con el mismo dominio y contra dominio.

Una forma de lograr lo antes mencionado, es mediante la utilización de los momentos geométricos de una función. Cabe hacer notar que dicha correspondencia puede ser únicamente obtenida con un número infinito de momentos en el caso euclidiano, o cuando se tienen al menos tantos momentos como puntos tiene  $X$ , aunque esto no siempre es de utilidad en la práctica, ya que comúnmente es conveniente que distintas funciones compartan el mismo conjunto de valores característicos.

La identificación de objetos, independientemente de que estos presenten ciertas transformaciones, es un tema importante en el reconocimiento de patrones [11], y la aproximación mayormente aceptada es la basada en los momentos geométricos, o simplemente momentos [3][5].

Aunque las primeras investigaciones sobre aplicaciones de momentos para la identificación de objetos, descritos mediante caracterizaciones invariantes, fueron publicadas hace 40 años, un gran número de técnicas e investigaciones han surgido alrededor de dichos momentos, las que son de gran interés aun en la actualidad [26][33]-[38].

Para un objeto dado en una imagen, en la definición original de momentos dada por Hu, se define una caracterización que consiste en una tupla ordenada de números reales, con los siete momentos invariantes derivados a partir de los tres primeros momentos centrales. Dicha caracterización es invariante a cambios de escala, rotación y traslación [7].

De aquí que se considere la utilización de los momentos invariantes para probar mediante el clasificador Gamma, aprovechando su característica de comparación, que los rasgos obtenidos con estos pueden ser altamente discriminantes en tareas de recuperación y clasificación de imágenes. Sumado a la presentación del nuevo algoritmo del clasificador Gamma.

### ***1.3 Planteamiento del problema***

Como se ha mencionado anteriormente, el cálculo de momentos geométricos, así como de sus correspondientes momentos invariantes, es ampliamente utilizado en el procesamiento de imágenes, ya que por medio de estos es posible obtener características que permiten representar numéricamente a los objetos para fines de comparación y clasificación.

Mediante la utilización de un clasificador de alto desempeño, con la capacidad de discriminar objetos de clases linealmente no separables y con dispersión (como el clasificador Gamma), es posible probar que técnicas de extracción de rasgos ampliamente conocidas en la literatura pueden producir realmente caracterizadores que presenten alta discriminación entre objetos.

En conclusión, el propósito del presente trabajo es el de probar el poder de discriminación de los momentos invariantes, así como el desempeño del clasificador Gamma con imágenes binarias, cabe señalar aquí que se presenta un nuevo algoritmo para la simplificación del clasificador Gamma.

## **1.4 Contribuciones**

En la presente tesis, se mostrará un estudio comparativo del clasificador Gamma con algunos de los clasificadores más representativos y de distintos enfoques para mostrar su desempeño en tareas de clasificación que involucran imágenes binarias. Adicionalmente se presenta un nuevo algoritmo del clasificador Gamma cuya implementación es utilizada en la realización del conjunto de experimentos.

La utilización del clasificador Gamma con imágenes binarias mostrará su alta competitividad al permitir observar, además, que los caracterizadores obtenidos con los invariantes de Hu pueden ofrecer alto grado de discriminación al trabajar en conjunto con el clasificador Gamma.

## **1.5 Organización del documento**

La presente tesis se encuentra organizada de la siguiente manera: en el presente capítulo, titulado Introducción, se describe el objetivo de la tesis, las motivaciones que impulsan a su realización y las contribuciones. En el capítulo 2, antecedentes, se mencionan algunas de las investigaciones más relevantes en las últimas décadas entorno a la teoría de momentos y su empleo en tareas de clasificación. En el capítulo 3, titulado marco teórico, se mencionan los conceptos importantes relacionados con el cálculo de momentos geométricos. De igual forma se describen de manera introductoria los métodos de clasificación utilizados en el estudio experimental. En el capítulo 4 se describe el material matemático y conceptual que sirve como base para sustentar este trabajo de tesis como un marco teórico para su desarrollo. El desarrollo principal del modelo propuesto y su demostración se brindan en el capítulo 5, para posteriormente mostrar los experimentos realizados en el capítulo 6. Las comparaciones con el método original que llevan a las conclusiones expuestas, se describen en el capítulo 7, dónde también se señalan los trabajos futuros y posibles temas de investigación derivados de la presente investigación.

## Capítulo 2

### ***Antecedentes***

El cálculo de momentos geométricos a imágenes ha sido de gran interés desde su introducción en 1960, siendo el móvil principal encontrar un conjunto finito de momentos que pueda ser suficientemente discriminante de otros conjuntos. Para ello, los investigadores han utilizado gran variedad de conceptos en el diseño de nuevas metodologías para obtener derivaciones de momentos que permitan incrementar las características invariantes o reducir los errores de aproximación de su cálculo.

Varias técnicas han sido creadas a partir de una derivación de las características invariantes de los momentos para la representación de objetos. Dichas técnicas son distinguidas por su definición de momentos, el tipo de datos de imagen empleados, y los métodos para la derivación de valores invariantes de los momentos de una imagen. Varias definiciones de momentos son caracterizadas mediante otras opciones de funciones base, las cuales pueden ser polinomios ortogonales o no ortogonales [10][33][35], y el muestreo de la imagen, que puede ser rectangular o polar [8][39]. Los momentos han sido definidos para dos dimensiones (silueta y borde) [15], tres dimensiones [30][38], e imágenes en niveles de gris [32]. Muchas de las caracterizaciones invariantes logran invariancia a la escala y la traslación del objeto mediante una normalización, dado que ésta es fácilmente completada sobre la base de los momentos de bajo orden (menor que tres) [10][25][34].

Cinco principales técnicas de caracterización invariante basadas en momentos pueden ser identificadas en la investigación actual [24][36][40]. El primer método, denominado en este capítulo como, *momentos invariantes*, está basado en combinaciones no lineales de momentos geométricos o cartesianos de dos dimensiones de bajo orden que mantienen la invariancia ante rotación [7]. Definiciones alternativas de momentos geométricos basadas en la representación polar de la imagen, *momentos rotacionales*, fueron también propuestas como una solución dadas sus propiedades de rotación simples [8][25]. Definiciones de momentos utilizando funciones base no correlacionadas, *momentos ortogonales*, fueron desarrolladas para reducir la redundancia de información que existe con los momentos convencionales [10][33][35][41]. Adicionalmente, los momentos ortogonales tienen transformaciones inversas simples, y pueden ser usados para determinar el número mínimo de momentos requeridos para reconstruir adecuadamente, y así caracterizar únicamente, una imagen dada. Relacionados con los momentos ortogonales, los *momentos complejos* proveen cálculos simples de momentos invariantes de orden arbitrario [9].

El presente capítulo menciona las principales investigaciones relacionadas con estas técnicas.

## **2.1 Momentos invariantes**

En 1960, Hu [7] deriva siete invariantes a partir de los momentos geométricos los cuales son combinaciones lineales de los momentos geométricos normalizados hasta orden tres, mismos que consiguen invariancia ante la traslación, la rotación y el cambio de escala. En dicha investigación Hu prueba los invariantes en el reconocimiento de caracteres [11] y años más tarde se emplean para clasificar imágenes binarias de aeronaves [12][15].

## **2.2 Momentos radiales**

En [8] se presenta una aproximación alternativa al cálculo de momentos invariantes basada en momentos radiales y angulares. Los cuales presentan características invariantes a rotación, traslación, reflexión y cambios de escala con la ayuda de los momentos regulares. Una de las ventajas de tratar con momentos angulares y radiales es que existe una forma de obtener los invariantes directamente, es decir, sin necesidad de realizar cálculos que involucran los momentos regulares como lo hace Hu. Otra de las ventajas ofrecidas es que no se está necesariamente restringido a potencias enteras positivas de  $r$  al generar los invariantes, ya que pueden obtenerse potencias de  $r$  no enteras.

## **2.3 Momentos rotacionales**

La noción de momentos rotacionales se introduce como una alternativa a los momentos regulares [4]. Los momentos rotacionales están basados en la representación de la imagen en coordenadas polares y pueden ser usados para extender la definición de momentos invariantes para un orden arbitrario, de forma que asegura que sus magnitudes no disminuyen significativamente al incrementar el orden [36].

## **2.4 Momentos complejos**

Los momentos complejos se propone en [9] como una forma simple para la derivación de un conjunto de momentos invariantes. Aunque este método provee una forma eficaz de obtención de momentos y que son los menos susceptibles ante la presencia de ruido, entre las metodologías para cálculo de momentos [24], dichos momentos no son, en general, buenos caracterizadores [9], ya que sufren de pérdida de información, supresión, y redundancia de datos, lo cual limita su poder de discriminación. Posteriormente en [14] se redefinen las normalizaciones de imagen para el cambio de escala, traslación, rotación y contraste en términos de momentos complejos, y muestran también que los momentos invariantes son derivables de los momentos complejos de la imagen normalizada.

## **2.5 Momentos ortogonales**

La función base empleada en los momentos geométricos no es ortogonal, lo cual permite un alto costo computacional al realizar la reconstrucción de una imagen a partir de los

momentos regulares. Además, los momentos regulares sufren de alto grado de redundancia de información, y en el caso de los momentos de alto orden existe alta sensibilidad con relación al ruido [18]. Así, en [10] se propone la noción de momentos ortogonales basada en la teoría de polinomios ortogonales para superar los problemas asociados con los momentos regulares. Se introduce la teoría de momentos de Zernike y Legendre, basados en los polinomios de Zernike y Legendre respectivamente. Una importante propiedad de los momentos Zernike, gracias a su ortogonalidad, es la fácil reconstrucción de la imagen, ya que haciendo uso de esta propiedad están capacitados para separar la contribución individual de sus momentos para el proceso de reconstrucción, donde la adición simple de dichas contribuciones individuales genera la imagen reconstruida [19]. Además, dichos caracterizadores pueden ser construidos para un orden alto debido a su independencia [27].

En [24] se probó que los momentos Zernike son menos sensibles a la presencia de ruido en comparación con los momentos de Legendre. En general, los momentos Zernike son capaces de lograr el mejor desempeño con relación a la redundancia de información y la representación de la imagen con respecto a las demás metodologías para cálculo de momentos, momentos regulares, invariantes, rotacionales, angulares y Legendre [34].

El cálculo de los momentos ortogonales requiere una transformación de coordenadas del espacio de la imagen a un subespacio, de círculo unitario para los momentos Zernike y cuadrado para el caso de los momentos Legendre, además de una aproximación al sustituir integrales continuas por las sumatorias discretas lo cual conduce a errores numéricos en el cálculo. Adicionalmente, derivaciones de los momentos Zernike y Legendre logran la invariancia a la traslación y cambio de escala [18][19].

En [33] se propone un nuevo conjunto de caracterizadores de momentos ortogonales basados en los polinomios Tchevichev. La derivación de los momentos Tchevichev soluciona las limitaciones de los momentos ortogonales, ya que no implica aproximaciones numéricas de ningún tipo, dado que el conjunto base es ortogonal dentro del dominio discreto del espacio de coordenadas de la imagen, satisfaciendo exactamente la ortogonalidad. Esta propiedad hace a los momentos Tchevichev superiores a los momentos ortogonales convencionales, tales como momentos Zernike y Legendre, en función de la preservación de las propiedades analíticas necesarias para asegurar redundancia de información reducida en un conjunto de momentos, además de poder producir una reconstrucción de imagen superior [34]. Así, la normalización del espacio requerida en la evaluación de momentos Zernike y Legendre es eliminada, al igual que la necesidad de cualquier aproximación discreta.

Los momentos Tchevichev pueden ser representados con una forma radial-polar de las coordenadas de la imagen, lo cual permite a los descriptores basados en dichos momentos ser invariantes a rotación [34]. En [45] se logran las derivaciones de los momentos Tchevichev para obtener la invariancia ante transformaciones afines y cambio de foco.

En [35] se propone un nuevo conjunto de momentos ortogonales basados en los polinomios discretos de Krawtchouk. Dada la ortogonalidad los momentos Krawtchouk aseguran redundancia de información mínima, y al igual que los momentos Tchevichev, no emplean ninguna aproximación numérica para su derivación, al tener una base discreta. Los resultados muestran que los momentos Krawtchouk pueden ser empleados

para extraer características locales de una imagen, a diferencia de otros momentos ortogonales los cuales generalmente capturan características globales.

Recientemente en [44] se introduce una nueva forma de momentos ortogonales discretos basada en los polinomios Hahn. Particularmente esta definición de momentos es investigada aplicada en la reconstrucción de imágenes. Experimentalmente, los momentos Hahn son comparados contra los otros momentos ortogonales discretos: Tchevichev y Krawtchouk. Los resultados muestran que los momentos Hahn son superiores a los dos métodos de momentos con respecto a la reconstrucción de imágenes.

## **2.6 Otros trabajos relevantes**

Tres principales ramas de investigación se desprenden entorno a la teoría de momentos: La primera, mencionada al inicio de presente capítulo con los trabajos más relevantes al respecto, consiste en reducir el error que conlleva el paso de la función base de su forma continua a discreta y el de la redundancia de información, proponiendo la utilización de funciones base que permiten trabajar con imágenes digitales bajo su mismo dominio discreto.

La segunda vertiente, se relaciona con el empleo de los momentos, o de algunas de sus técnicas derivadas, en casos prácticos de tareas de clasificación y utilizando imágenes de distinto dominio: binarias, en escala de gris, de dos y tres dimensiones, satelitales, médicas, etcétera. Al respecto, en [28] se utilizan los momentos de Hu como descriptores de vocales escritas a mano para su reconocimiento y clasificación utilizando el clasificador K-NN. En [50] presentan un nuevo método para el reconocimiento de objetos en ambientes con y sin ruido, basado en los momentos invariantes modificados y la norma mínima donde obtienen resultados con altos porcentajes de clasificación. De igual forma, en [26] se muestra el desempeño de las derivaciones invariantes a transformaciones afines y a cambio foco de los momentos Tchevichev utilizando imágenes de peces en tres dimensiones con ruido y otras con caracteres obteniendo buen desempeño reflejándose en altos índices de clasificación al emplear el clasificador Euclidiano. En [32] se propone un esquema que permite calcular de forma rápida los momentos geométricos en imágenes en escala de gris. Posteriormente el mismo autor, en [46], emplea los momentos Krawtchouk junto con el clasificador K-NN para clasificar imágenes en escala de gris bajo el esquema ISR como forma de representación de la imagen y facilitar el cálculo de los momentos mostrando resultados con altos porcentajes de clasificación.

La tercera y última vertiente se enfoca en el desarrollo y obtención de nuevas propiedades invariantes de los momentos, ante transformaciones y distorsiones que presenten los objetos de interés o las imágenes que los contienen: transformaciones afines, cambios de contraste, brillo, etcétera. Recientemente, en [20] utilizando como domino imágenes a color y derivan un conjunto de expresiones a partir de combinaciones de los momentos geométricos los cuales consiguen invariancia a cambios fotogramétricos los cuales so probados de experimentalmente con imágenes tomadas en interiores y exteriores mostrando su eficacia. Introduce en [21] un conjunto de momentos los cuales consiguen la invariancia a objetos o imágenes borrosas, mismos que son también invariantes a transformaciones afines. Posteriormente en [22] el mismo autor propone derivaciones de momentos que consiguen ser invariantes ante proyecciones de la imagen. Finalmente, en

[18][19] y [29] se consiguen derivaciones que presentan invariancia a traslación y cambio de escala de los momentos Zernike, Legendre y Tchevichev respectivamente.

Como se ha mostrado, y aunque existen distintas técnicas de momentos, la más utilizada sigue siendo la propuesta inicialmente por Hu debido a la simplicidad de su cálculo. De ahí que, sea esta definición de momentos la que se utilice en el presente trabajo para describir y caracterizar a los objetos.

## Capítulo 3

### Marco teórico

En el presente capítulo se muestran los conceptos básicos, relacionados con la teoría de momentos y de los métodos de clasificación, necesarios para comprender los capítulos siguientes.

#### 3.1 Momentos geométricos

Como el cálculo de momentos proporciona una útil e interesante alternativa en la representación de objetos en una imagen, estos son usados en diferentes áreas del procesamiento de imágenes, tales como: en el análisis de formas y el reconocimiento de patrones, en particular; así como en las áreas donde la extracción y clasificación de las características o patrones de imágenes, y su codificación, son de vital importancia.

Una manera de obtener estos invariantes es mediante el cálculo de momentos geométricos o cartesianos de una imagen [2][4], ya que describen su contenido, distribución y otras características propias de ésta. Están diseñados para obtener tanto información geométrica global como detallada de la imagen.

Los momentos pueden ser determinados para el caso de  $n$  dimensiones. Un ejemplo típico que ilustra el cálculo de momentos es el de una imagen continua de dos dimensiones: el momento geométrico  $m$  de orden  $p+q$ , expresado como  $m_{pq}$ , de una función de distribución de intensidad en el plano real  $f(x,y)$ , está dado por:

$$m_{pq} = \int \int_{x \ y} x^p y^q f(x, y) dx dy \quad (3.1)$$

donde  $x, y \in \mathbb{R}^2$ ,  $f(x,y)$  es la función característica de la imagen y  $p, q = 0, 1, 2, \dots$

Ya que  $p$  y  $q$  toman todos los valores de enteros no negativos, se obtiene un conjunto infinito de momentos que determina unívocamente cada función  $f(x,y)$ , y viceversa, cada valor del conjunto es determinado por  $f(x,y)$ .

Afortunadamente en la práctica, incluso si la función  $f(x,y)$  es complicada, no es necesario un número infinito de momentos para describirla adecuadamente, ya que es permitido discretizar la ecuación (3.1), reemplazando la doble integral por una doble sumatoria, de la siguiente manera:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad (3.2)$$

donde  $f(x,y)$  es la función característica de la imagen,  $x, y \in \mathbb{Z}^2$  son las coordenadas de la imagen, y  $p, q = 0, 1, 2, \dots$

Si además, la función representa una imagen binaria, es decir, la función  $f(x,y)$  es una función característica que toma el valor de 1 o 0 dependiendo de si el píxel, en las coordenadas  $(x,y)$ , pertenece o no a la región de interés,  $P$ ; suponiendo que tiene valor 1, si el píxel pertenece a la región de interés (objeto) y 0 si pertenece a su complemento (fondo), entonces sustituyendo  $f(x,y)=1$  en la ecuación (3.2), se obtiene la versión discreta para el cálculo de los momentos cartesianos de una imagen binaria, de la siguiente manera:

$$m_{pq} = \sum_{x,y \in R} x^p y^q \quad (3.3)$$

Esta definición representa los momentos geométricos simples que si bien proporcionan información por sí mismos, a partir de ellos se pueden calcular otros momentos que cuenten con ciertas características invariantes de interés para el procesamiento de imágenes.

### 3.1.1 Propiedades de los momentos

Los valores de los momentos “simples” de bajo orden representan propiedades geométricas fundamentales de una distribución (objeto de interés en una imagen) o de un cuerpo [1][4].

#### 3.1.1.a Momentos de orden cero: área

Este momento es calculado mediante:

$$m_{00} = \sum_x \sum_y x^0 y^0 \quad (3.4)$$

Se puede observar claramente que el momento de orden cero se calcula como la suma de los píxeles que pertenecen al objeto, si se habla de imágenes binarias. De esta manera, se obtiene la superficie de la imagen, su área o bien, la masa total del objeto, ya que, literalmente se habla de contar el número de píxeles que aparecen en el objeto. Sin embargo, en una imagen en tonos de gris, este momento es una medida de la energía del objeto [24][27].

#### 3.1.1.b Momentos de primer orden: centro de masa

Los dos momentos de primer orden  $\{m_{10}, m_{01}\}$  representan el centro de masa de un objeto. Las coordenadas dadas por  $\{X, Y\}$  son la intersección de las líneas,  $x=X$  y  $y=Y$ , paralelas a los ejes  $x$  y  $y$  respectivamente, y sobre los cuales los momentos de primer orden son cero. Alternativamente,  $x=X$  y  $y=Y$  representan las líneas donde toda la masa puede ser concentrada sin cambios en los momentos de primer orden sobre los ejes  $x$  y  $y$  respectivamente. En términos de coordenadas para el caso de dos dimensiones, el centro de masa está dado por:

$$X = \frac{m_{10}}{m_{00}}, \quad Y = \frac{m_{01}}{m_{00}} \quad (3.5)$$

El centro de masa define la única ubicación con respecto al objeto que puede ser usada como punto de referencia para describir su posición dentro del campo de visión. Si un objeto es posicionado de tal forma que su centro de masa coincide con el origen del campo de visión, entonces los momentos calculados para ese objeto son referidos como momentos centrales (más información de los momentos centrales ver Sección 3.2.1), y se denotan como  $\mu_{pq}$  [36].

### 3.1.1.c Momentos de segundo orden

Los momentos de segundo orden  $\{m_{02}, m_{11}, m_{20}\}$  conocidos también como los momentos de inercia en física, pueden ser usados para determinar varias características útiles de un objeto, en particular, características relacionadas con su orientación, como puede ser: localización de los ejes principales, la elipse objeto y el radio de giro de un objeto.

*Ejes principales:* los ejes principales pueden describirse como el par de ejes en los cuales se localizan los momentos de orden dos, mínimo y máximo ( $m_{20}$  y  $m_{02}$ ), y que representan a los ejes menor y mayor respectivamente. Es términos de momentos, las orientaciones están dadas por:

$$\phi = \frac{1}{2} \tan^{-1} \left( \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (3.6)$$

donde  $\phi$  es el ángulo del eje principal más cercano al eje  $x$  cuyo valor se encuentra en el rango  $-\pi/4 \leq \phi \leq \pi/4$ .

*Elipse objeto:* la elipse circundante es un disco elipsoidal con intensidad constante con la misma masa y momentos de orden dos del objeto original. Si esta elipse se define a través de dos ejes, uno mayor  $a$ , a lo largo del eje  $x$ , y otro menor  $b$ , a lo largo del eje  $y$ , entonces  $a$  y  $b$  pueden determinarse a partir de los momentos de orden dos mediante la siguiente expresión:

$$(a, b) = \sqrt{\frac{2 \left[ \mu_{20} + \mu_{02} \pm \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \right]}{\mu_{00}}} \quad (3.7)$$

Si, además, la elipse es posicionada de forma que su centro coincida con el centro de masa del objeto y que su eje mayor coincida con el eje principal del objeto, entonces los momentos de primero y segundo orden de tal elipse serán iguales a los del objeto original.

*Radio de giro:* el radio de giro de un objeto sobre un eje es la distancia del eje a la línea donde toda la masa de dicho objeto puede ser concentrada sin cambios en los valores de los segundos momentos alrededor de dicho eje. En términos de momentos, los radios de giro sobre los ejes  $x$  y  $y$ , están dados por:

$$R_x = \sqrt{\frac{m_{20}}{m_{00}}}, \quad R_y = \sqrt{\frac{m_{02}}{m_{00}}} \quad (3.8)$$

El radio de giro alrededor del origen es el radio de un círculo centrado en el mismo lugar, donde es posible concentrar toda la masa de un objeto sin alterar el momento de inercia sobre dicho origen. El radio de giro de un objeto esta dado por:

$$R = \sqrt{\frac{\mu_{20} + \mu_{02}}{\mu_{11}}} \quad (3.9)$$

### 3.1.1.d Momentos de tercer orden

Los momentos de tercer orden y posteriores, son usados para describir propiedades de la proyección de un imagen sobre el eje  $x$  o  $y$ , más que propiedades del objeto mismo.

Estos momentos, así como los de orden superior, producen información abstracta difícil de interpretar físicamente. Entre otras cosas, los momentos de orden tres son utilizados para calcular los momentos invariantes, ya que a partir de estos y los de orden menor, se obtienen los siete momentos invariantes de Hu [7], que, como su nombre lo indica, se mantienen invariantes ante rotaciones, traslaciones y cambios de escala de una imagen.

## 3.2 Momentos invariantes

En la práctica, existen ciertas transformaciones que pueden afectar deliberadamente a una imagen, por lo que es importante el cálculo de momentos geométricos invariantes que permitan reconocer imágenes, aun si han sido modificadas por algún tipo de transformación. Los casos típicos de transformaciones a los que está expuesta una imagen, como se mencionó anteriormente, son las transformaciones geométricas de traslación, rotación y cambios de escala.

### 3.2.1 Invariantes a traslaciones: momentos centrales

En el caso de traslaciones es deseable que el valor  $m_{pq}$  se conserve constante ante cambios de posición del objeto sin rotaciones y cambios de escala, en otras palabras es deseable que:

$$m'_{pq} = m_{pq}; x' = x + a; y' = y + b \quad (3.10)$$

La forma de lograr lo anterior es que el origen de referencia sea movido al centroide del objeto. Esto da origen a los llamados momentos centrales. El centro de gravedad, también conocido como promedio o media aritmética, está dado por  $(X, Y)$  calculadas a partir de los momentos de orden uno, con las expresiones (3.5):

$$X = \frac{m_{10}}{m_{00}}, Y = \frac{m_{01}}{m_{00}}$$

las cuales, permiten el cálculo de los momentos invariantes a la traslación, ya que son calculados con respecto al centro de gravedad de la imagen; esto se logra moviendo el origen de referencia al centroide del objeto a partir de la expresión (3.3), de la siguiente manera:

$$\mu_{pq} = \sum_x \sum_y (x - X)^p (y - Y)^q f(x, y) \quad (3.11)$$

Dichos momentos invariantes son denotados por  $\mu_{pq}$  y pueden ser expresados como sigue:

$$\begin{aligned} \mu_{00} &= m_{00} \\ \mu_{10} &= 0 \end{aligned}$$

$$\begin{aligned}
\mu_{01} &= 0 \\
\mu_{20} &= m_{20} - Xm_{10} \\
\mu_{02} &= m_{02} - Ym_{01} \\
\mu_{11} &= m_{11} - Ym_{10} \\
\mu_{30} &= m_{30} - 3Xm_{20} + 2m_{10}X \\
\mu_{12} &= m_{12} - 2Ym_{11} - Xm_{02} + 2Y^2 m_{10} \\
\mu_{21} &= m_{21} - 2Xm_{11} - 2Ym_{20} + 2X^2 m_{01} \\
\mu_{03} &= m_{03} - 3Ym_{02} + 2Y^2 m_{01}
\end{aligned} \tag{3.12}$$

Asimismo, mediante la normalización de los momentos centrales, es posible obtener los momentos que proporcionan invariancia a escala geométrica.

### 3.2.2 Invariantes a cambios de escala: momentos normalizados

Los cambios de escala son obtenidos mediante transformaciones de coordenadas, los momentos invariantes a este tipo de transformación pueden conseguirse al dividir cada momento por un factor de normalización que cancela el efecto de escalamiento. Estos momentos son conocidos también como momentos geométricos normalizados o momentos centrales normalizados. La fórmula de los momentos centrales normalizados, está dada por:

$$\eta = \frac{\mu_{pq}}{\mu_{00}^\gamma} \tag{3.13}$$

donde  $\gamma = (p+q)/2 + 1$  y  $p+q = 2, 3, \dots$

### 3.2.3 Invariantes a rotaciones

Con los momentos centrales pueden ser derivados los primeros cuatro invariantes de Hu, los cuales presentan la invariancia a traslaciones y rotaciones:

$$\begin{aligned}
\phi_1 &= \mu_{20} + \mu_{02} \\
\phi_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11} \\
\phi_3 &= (\mu_{20} + 3\mu_{02})^2 + (3\mu_{21} - \mu_{03})^2 \\
\phi_4 &= (\mu_{30} + \mu_{21})^2 + (\mu_{21} + \mu_{03})^2
\end{aligned} \tag{3.14}$$

Dichos momentos pueden ser también derivados a partir de los momentos complejos [23].

### 3.2.4 Invariantes de Hu

El primer trabajo significativo sobre el empleo de momentos para el procesamiento de imágenes y reconocimiento de patrones, fue realizado por Hu [7][12]. Basado en la teoría de invariantes algebraicos, Hu derivó combinaciones absolutas y relativas de momentos que son invariantes con respecto a cambios de tamaño (escala), posición (traslación) y orientación del objeto (rotación).

El método de momentos invariantes, es derivado de los invariantes algebraicos aplicados a la función de generación de momentos bajo la transformación de rotación. Dicho

conjunto de invariantes, consiste de una serie de combinaciones no lineales de los momentos centrales que se mantienen invariantes ante la rotación.

Hu define las siguientes siete funciones, calculadas a partir de los momentos normalizados hasta tercer orden, los cuales mantienen la invariancia ante la traslación, escala y rotación de un objeto, que resultan al remplazar los momentos centrales, en las ecuaciones para invariantes a rotaciones, por los invariantes a escala  $\eta_{pq}$ , de la siguiente manera:

$$\begin{aligned}
 \phi_1 &= \eta_{20} + \eta_{02} \\
 \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11} \\
 \phi_3 &= (\eta_{20} + 3\eta_{02})^2 + (3\eta_{21} - \eta_{03})^2 \\
 \phi_4 &= (\eta_{30} + \eta_{21})^2 + (\eta_{21} + \eta_{03})^2 \\
 \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
 \phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
 \phi_7 &= (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
 \end{aligned} \tag{3.15}$$

Sin embargo, estas funciones son de orden finito, por lo que, a diferencia de los momentos regulares, no son suficientes para mantener un conjunto completo de descriptores de una imagen. Aunque es posible derivar funciones de mayor orden con motivo de incrementar el número de funciones [11][27], esto resulta en la alta complejidad de los cálculos para dichas derivaciones, aunado a los problemas de redundancia de información y de error inherente al utilizar la aproximación de sumatorias [10], lo cual motiva a mostrar que los caracterizadores obtenidos con una combinación reducida de invariantes puede ser altamente discriminante al ser utilizada con el clasificador Gamma.

En general, como se ha mencionado, los momentos geométricos describen cantidades numéricas en alguna distancia de un punto de referencia o eje. El uso de momentos para análisis de imágenes es sencillo si se considera un segmento de imagen binaria o en niveles de gris como una función de distribución de densidad de dos dimensiones (ver ecuación 3.2); de esta forma los momentos pueden ser usados para caracterizar dicho segmento de imagen.

### Ejemplo 3.1

Dada la imagen binaria de la Figura 3.1.1, los valores de los momentos calculados utilizando las expresiones (3.15), son mostrados en la tabla 3.1.1.

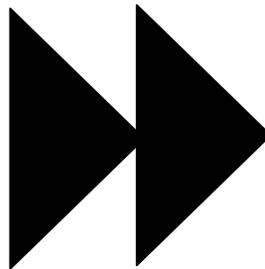


Figura 3.1.1. Región de interés representada por una imagen binaria

#	Valor
$\phi_1$	0.2278
$\phi_2$	0.004114
$\phi_3$	0.000627
$\phi_4$	0.00002417
$\phi_5$	-0.000000002975
$\phi_6$	0.00000155
$\phi_7$	-0.0000000000000000000000004046

Tabla 3.1.1. Tabla de momentos de la imagen de la Figura 3.1.1

**Ejemplo 3.2**

En la figura 3.1.2.a y 3.1.2.b, se muestran dos objetos, representando a dos clases, rectángulo y triángulo. Cada uno con un conjunto de elementos sometidos a distintas transformaciones geométricas, por ejemplo rotación, traslación y cambio de escala, y que en total componen 11 objetos para cada clase.



Figura 3.1.2. Objetos de tamaño 100x100 en formato binario, a) rectángulo y b) triángulo.

En la figura 3.1.3 se observan los elementos de cada clase graficando dos rasgos dados por los momentos de Hu  $\phi_2$  y  $\phi_4$  tomados de las expresiones (3.15), representados en dicha figura por los ejes  $x$  y  $y$  respectivamente. Adicionalmente se muestra el centroide de cada clase (círculo relleno) y los elementos de la clase alrededor de éste (círculos sin relleno). Se muestra también la desviación estándar del grupo, delineada en color gris, y la región de máxima pertenencia, dada por el patrón más alejado del centro, y es indicada por un círculo del mismo color que los elementos de la clase.

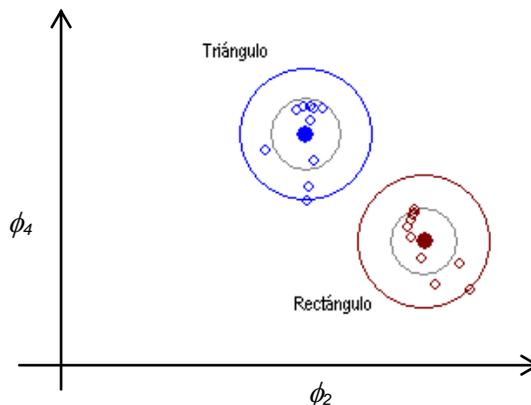


Figura 3.1.3. Representación gráfica de los objetos evaluados de la Figura 3.1.2.a y 3.1.2.b.

Los momentos de Hu graficados en este ejemplo permiten observar su utilidad como descriptores de objetos, ya que los valores numéricos que resultan de su cálculo representan de forma correcta a los objetos, reflejándose en la gráfica del ejemplo por la concentración separada de los dos distintos grupos de objetos o patrones de un mismo tipo, rectángulo y triángulo respectivamente.

Partiendo del teorema de Hu, que dice que si  $f(x,y)$  es una función de segmentos continuos y tiene valores distintos de cero solo en la región finita del plano  $(x,y)$ , entonces los momentos de todos los órdenes existen. Entonces puede ser demostrado que el conjunto de momentos  $\{m_{pq}\}$  está únicamente determinado por  $f(x,y)$ , e inversamente,  $f(x,y)$  está únicamente determinada por  $\{m_{pq}\}$  [7]. Así, dado que un segmento de imagen tiene un área finita, o en el peor caso, es de segmentos continuos, los momentos de todos los órdenes existen y un conjunto de momentos puede ser calculado y describirá de forma única la información contenida en ese segmento de imagen. De ahí que para caracterizar toda la información contenida en un segmento de imagen se requiere de un número infinito de valores de momentos. Parte del presente trabajo de tesis consiste en mostrar que un subconjunto reducido de momentos es suficiente para obtener alta discriminación entre objetos de clases diferentes para un problema y clasificador específicos, de ahí que en los experimentos del Capítulo 5, correspondiente a los resultados y su discusión, se utilicen estas derivaciones de invariantes para obtener los rasgos de los objetos utilizados.

### 3.3 Otras características invariantes de los momentos

Posteriormente se presentó una variación al modelo de momentos invariantes de Hu, los cuales ofrecen adicionalmente invariancia ante cambios de contraste [13]. Tales momentos son también inherentemente invariantes al tamaño, por lo que no requieren normalización de tamaño. Así que, en términos de los momentos de Hu, los primeros seis nuevos invariantes quedan definidos como:

$$\begin{aligned}
 \beta_1 &= \frac{\sqrt{\phi_2}}{\phi_1} \\
 \beta_2 &= \frac{\phi_3 \mu_{00}}{\phi_1 \phi_2} \\
 \beta_3 &= \frac{\phi_4}{\phi_3} \\
 \beta_4 &= \frac{\sqrt{\phi_5}}{\phi_4} \\
 \beta_5 &= \frac{\phi_6}{\phi_1 \phi_4} \\
 \beta_6 &= \frac{\phi_4}{\phi_3}
 \end{aligned} \tag{3.16}$$

Otras investigaciones semejantes han logrado la invariancia de los momentos ante transformaciones afines [5], cambios de foco e iluminación [20], objetos borrosos [21], a proyecciones de los objetos [22], e incluso si hay ruido presente [43].

Como se ha mencionado anteriormente, diversas investigaciones en las últimas décadas han sido dedicadas a la investigación de nuevos invariantes [24][27] por lo que han ganado importancia en los últimos años [44]-[49]. Y como es de notarse, a pesar de que actualmente existen invariantes ante distintas transformaciones, a las cuales puede estar sometido un determinado objeto (ya sea transformaciones geométricas, afines, cambios de foco y de contraste), el número de dichas invariantes y el tipo de estos dependerá directamente del problema y la aplicación a resolver [24][36][38][50].

### **3.4 Técnicas de clasificación**

A continuación se describen las técnicas de clasificación y predicción más representativas correspondientes a algunos de los enfoques y técnicas comúnmente utilizadas en el campo del reconocimiento de patrones y de representación de información. Las distintas técnicas aquí mostradas servirán de referencia, de tal forma que se comprenda el fundamento de cada enfoque descrito utilizado para realizar el estudio comparativo con el clasificador Gamma.

#### **3.4.1 Vecino más cercano (Nearest Neighbour-NN): IB1**

Nearest Neighbour, NN, es un método originado en estadística. Inicialmente considerado para la producción de reglas; es propuesto en [51] donde se muestra un análisis preliminar de las propiedades de los sistemas de  $k$  vecinos más cercanos y se establece la consistencia del método para las variaciones de  $k$  desde 1 hasta  $\infty$ . En dicho estudio se evalúa numéricamente el desempeño de los  $k$  vecinos más cercanos para pequeñas muestras bajo criterios estadísticos de distribución normal [52]. Posteriormente es propuesto como clasificador de patrones en [53] y sometido a análisis en tareas de clasificación y predicción [54]-[56].

Los métodos basados en NN utilizan una métrica que mide la distancia entre un nuevo elemento y el conjunto de elementos almacenados. El nuevo elemento es clasificado según el tipo de su vecino más cercano. Un método basado en NN puro almacena todos los elementos en memoria de forma literal. Éste clasifica los nuevos elementos localizando el elemento más semejante, de entre los almacenados, para posteriormente asignar su clase. Una función de distancia es usada para determinar dicha similitud. Para atributos numéricos, se utiliza comúnmente la distancia Euclidiana, donde cada elemento es tratado como un punto en un espacio característico de  $n$  dimensiones. Se asume que para un punto dado en dicho espacio característico el área alrededor compartirá la misma clase. Además, la distancia Euclidiana asume que todas las características son igualmente importantes, así que comparten la misma escala en el espacio característico, y dicha escala es lineal a lo largo de cada eje [53].

Los métodos basados en NN tomaron popularidad después que en [2][57] se mostrara que los más simples modelos del método podían producir excelentes resultados para una variedad de dominios. En dicha investigación se introduce el método IB1 como parte de una familia de algoritmos creados bajo distintos criterios de procesamiento de las muestras

con el objetivo de corregir o minimizar algunas desventajas mencionadas en [58], las cuales están inherentes en el método original de NN, tales como:

- Alto costo de almacenamiento
- Sensibles a la función de similitud
- Sin criterios para el manejo de atributos faltantes
- Dificultad para trabajar con atributos nominales
- Sin generación de resumen de conceptos

Entre las principales características del algoritmo IB1 está la utilización de la función de distancia Euclidiana y que cada muestra nueva se clasifica de acuerdo con su vecino más cercano bajo esta función y se incorpora al resto de los elementos en el conjunto de datos conforme estas se van introduciendo. La única variación respecto al algoritmo NN puro yace en la normalización lineal de los valores de los atributos, previo al procesado de las muestras, y en el manejo de los atributos faltantes, los cuales se asume que tienen los valores máximos de diferencia para tal característica en todas las otras muestras. En general, se menciona que el desempeño de este modelo es muy bueno, pero al igual que con el algoritmo NN original presenta problemas cuando se trabaja con muestras con ruido, con atributos ausentes o con caracterizadores irrelevantes [57].

### 3.4.2 Naïve Bayes

El algoritmo de Naïve Bayes fue introducido en [2] para su uso en el área del reconocimiento de patrones. Este método es un modelo probabilístico el cual emplea como base principal el teorema de Bayes, y es uno de los más ampliamente utilizados [59].

Este método asume independencia condicional entre las clases, esto es, el efecto de un valor de atributo sobre una clase dada es independiente de los valores de los otros atributos. Esta suposición es hecha para reducir el costo computacional, de ahí su nombre *naïve*. El objetivo de este método es el de clasificar una nueva instancia  $X$  maximizando la probabilidad  $P(X_i/C_i)P(C_i)$ , donde  $i$  es el índice de la clase y  $\pi$  el producto de todas las probabilidades (ver Ecuación 2.17).

$$P(X | C_i) = \prod_{k=1}^n P(X_k | C_i) \quad (3.17)$$

En general, dados un conjunto de tuplas de datos desconocidas, donde cada tupla está representada por un vector de  $n$  dimensiones,  $X=(x_1, x_2, \dots, x_n)$  con  $n$  mediciones hechas sobre la tupla de  $n$  atributos, respectivamente,  $A_1, A_2, \dots, A_n$ ; y  $m$  clases,  $C_1, C_2, \dots, C_m$ . Utilizando el teorema de Bayes, el clasificador Naïve Bayes calcula la probabilidad a posteriori de cada clase condicionada sobre el elemento desconocido  $X$ . De ahí que el elemento desconocido  $X$  le sea asignado la etiqueta de la clase con la máxima probabilidad a posteriori condicionada sobre la misma clase  $X$ . Por lo tanto, el objetivo es maximizar la probabilidad  $P(C_{ij}/X) = P(X_j/C_i)P(C_i) = P(X)$ . De cualquier forma, y dado que la probabilidad  $P(X)$  es constante para todas las clases, solo la probabilidad  $P(X_j/C_i)P(C_i)$  requiere ser maximizada. Si las probabilidades de clase previas no son conocidas entonces se asume comúnmente que las clases son igualmente probables,  $P(C_1) = P(C_2) = \dots = P(C_m)$ , por lo que se debería maximizar  $P(X_j/C_i)$ . De lo contrario se maximiza  $P(X_j/C_i)P(C_i)$ . Las probabilidades previas de clase pueden ser estimadas por  $P(C_i) = s_i/s$ , donde  $s_i$  es el número de tuplas de entrenamiento de clase  $C_i$ , y  $s$  es el número de tuplas de entrenamiento.

Para reducir cálculos al evaluar  $P(X_j/C_i)$ , es considerada la suposición de la independencia condicional de clase. Así, se supone que los valores de los atributos son condicionalmente independientes unos de otros, dada la etiqueta de la clase de la tupla, por ejemplo, que no hay relación de dependencia entre los atributos.

Este método se caracteriza por su alta exactitud y desempeño en comparación con otros métodos de clasificación. En [60][61] se muestran los resultados de una comparación exhaustiva del método Naïve Bayes con algoritmos de árboles de decisión, basados en instancias (IB), y de reglas sobre conjuntos de datos estándar, donde se muestra el gran desempeño de este método incluso sobre conjuntos de datos con sustancial dependencia de atributos.

La importancia de este clasificador radica en su base bayesiana y sobre la cual (en [2], pp.17) se afirma lo siguiente: “Empezamos [con la Teoría de la Decisión Bayesiana] considerando el caso ideal en el cual la estructura estadística inherente a las categorías es perfectamente conocida. Mientras que este tipo de situación raramente se presenta en la práctica, [esta teoría] nos permite determinar el clasificador óptimo (Bayesiano) contra el cual podemos comparar a todos los demás clasificadores...”.

### 3.4.3 Bagging

Desarrollos en la teoría de Machine learning han guiado a métodos para mejorar el desempeño o extender las capacidades de los esquemas de aprendizaje básicos. Este tipo de mejoras de desempeño son conocidos como esquemas de meta clasificación o meta aprendizaje, y es que estos esquemas operan a partir de la salida de la ejecución repetida de algún método de clasificación tomado como base [62].

El método Bootstrap aggregating, o, de su acrónimo, Bagging [62], es un método de predicción perteneciente a los meta clasificadores y cuyo principal funcionamiento consiste en generar múltiples versiones de un mismo clasificador, comúnmente arboles de decisión, tal que utiliza estas versiones para obtener un clasificador compuesto, mejorado. Este clasificador obtenido promedia los resultados de todas las versiones cuando se trata de predecir un valor numérico y somete a votación general para clasificar cuando se trata de un elemento. Las versiones múltiples son formadas haciendo replicas del conjunto de aprendizaje mediante Bootstrap y las utiliza como nuevos conjuntos de aprendizaje.

Bagging es uno de los más populares y efectivos métodos de aprendizaje compuesto [63]. La técnica se basa en la reducción de varianza, cuyo principal objetivo es la corrección de la inestabilidad provocada por la incertidumbre de estimación de parámetros, y del modelo mismo, utilizado en el método, de ahí que sean utilizados sobre métodos de clasificación inestables y de alta varianza [64]. Se dice que un clasificador es inestable si un pequeño cambio en el conjunto de aprendizaje implica cambios significativos en el clasificador [63].

Diversas teorías han sido propuestas sobre la efectividad del método Bagging para clasificación basada en la descomposición del sesgo y de la varianza [65]. Asimismo, resultados de pruebas sobre conjuntos de datos reales y simulados usando árboles de clasificación y de regresión, así como selección de subconjuntos en regresión lineal, han mostrado que Bagging puede dar ganancias substanciales en exactitud, donde el

elemento vital es la inestabilidad del método [63][66][67]. Además de indicarse que también reduce la porción de sesgo del error. Dado que Bagging es una técnica de reducción de varianza y tiende a trabajar bien para métodos con alta varianza como redes neuronales y métodos basados en árboles, de ahí que no mejore el desempeño de métodos con baja varianza como regresión lineal y análisis de discriminantes lineales.

### 3.4.4 Tablas de decisión (Decision tables)

Las tablas de decisión, o Decision tables en inglés, son especificaciones de conjuntos mutuamente disjuntos de reglas de decisión. En aplicaciones típicas, las tablas de decisión son diseñadas manualmente, basadas en el conocimiento previo de los vectores de entrada posibles y de las decisiones asociadas [68]. En algunas áreas de aplicación, por ejemplo en sistemas de control complejos, este tipo de conocimiento no está disponible, o aún peor, la relación entre entradas y decisiones no es funcional. Este problema fue inicialmente tratado bajo la idea de tablas de decisión en el contexto de la teoría de conjuntos rugosos, en el cual, los datos experimentales, más que el conocimiento humano, son usados como una base para la derivación o entrenamiento en una tabla de decisión [69]. Este esquema es extendido en [70]-[72] para hacer uso de información probabilística de las decisiones en la tabla de decisión, conduciendo a la introducción de tablas de decisión probabilística [73] en el contexto del modelo de conjuntos rugosos de precisión variable [72][74]. Uno de los puntos clave de interés cuando se derivan tablas de decisión de los datos, es la forma de aprendizaje o de construcción de la tabla.

Una tabla de decisión esta constituida por columnas, que son etiquetadas por los atributos, los renglones, conformados por los objetos de interés, y las entradas de la tabla, representando los valores de los atributos. Los atributos de la tabla de decisión están divididos en dos grupos disjuntos llamados atributos de decisión y de condición respectivamente. Donde cada renglón de la tabla induce una regla de decisión, la cual especifica una solución o decisión (acción, resultados, salidas, etc.) si alguna o algunas condiciones son satisfechas. Si una regla determina únicamente una decisión en términos de condiciones, se dice entonces que la regla es cierta. De otra forma la decisión es incierta. Las reglas de decisión están fuertemente conectadas por aproximaciones o derivaciones. En resumen, las reglas de decisión ciertas describen una aproximación de decisión en términos de condiciones, mientras que reglas de decisión inciertas refieren a las regiones en la frontera de las decisiones [75].

Con cada regla de decisión se asocian dos probabilidades condicionales llamadas coeficientes de certeza y de cobertura. El coeficiente de certeza expresa la probabilidad condicional de que un objeto, que pertenece a la clase de decisión especificada por la regla de decisión, satisface dichas condiciones. El coeficiente de cobertura da la probabilidad condicional de las razones para una decisión dada. A su vez, estos dos coeficientes satisfacen el teorema de Bayes, lo cual ofrece una forma de interpretación de los datos.

De esta forma de [76] se tiene que una tabla de decisión se puede denotar como  $S=(U, C, D)$ , donde  $C$  y  $D$  son los conjuntos disjuntos de condiciones y atributos, respectivamente [70][71].

Sea entonces  $S=(U, C, D)$  una tabla de decisión. Se tiene que cada  $x \in U$  determina una secuencia  $c_1(x), \dots, c_n(x), d_1(x), \dots, d_m(x)$ , donde  $\{c_1, \dots, c_n\}=C$  y  $\{d_1, \dots, d_m\}=D$ .

La secuencia es llamada regla de decisión inducida por  $x$  (en  $S$ ) y se denota por  $c_1(x), \dots, c_n(x) \rightarrow d_1(x), \dots, d_m(x)$  o de forma simplificada  $C \rightarrow_x D$ .

El número  $\sup_x(C, D) = |A(x)| = |C(x) \cap D(x)|$  se llamará soporte de la regla de decisión  $C \rightarrow_x D$  y el número:

$$\sigma_x(C, D) = \frac{\sup_x(C, D)}{|U|} \quad (3.18)$$

Será referido como fuerza de la regla de decisión  $C \rightarrow_x D$  donde  $|X|$  denota la cardinalidad de  $X$ .

Con cada regla de decisión  $C \rightarrow_x D$  se asocia el factor de certeza de la regla de decisión, denotado  $cer_x(C, D)$  y se define como:

$$cer_x(C, D) = \frac{|C(x) \cap D(x)|}{|C(x)|} = \frac{\sup_x(C, D)}{|C(x)|} = \frac{\sigma_x(C, D)}{\pi(C(x))} \quad (3.19)$$

donde  $\pi(C(x)) = \frac{|C(x)|}{|U|}$ .

El factor de certeza puede ser interpretado como una probabilidad condicional de que  $y$  pertenezca a  $D(x)$  dada  $y$  perteneciente a  $C(x)$ , simbólicamente  $\pi_x(D, C)$ .

Si  $cer_x(C, D) = 1$ , entonces  $C \rightarrow_x D$  es llamado una regla de decisión cierta; si  $0 < cer_x(C, D) < 1$  la regla de decisión será referida como una regla de decisión incierta.

Además, el factor de cobertura de la regla de decisión, denotado como  $cov_x(C, D)$ , se define como:

$$cov_x(C, D) = \frac{|C(x) \cap D(x)|}{|D(x)|} = \frac{\sup_x(C, D)}{|D(x)|} = \frac{\sigma_x(C, D)}{\pi(D(x))} \quad (3.20)$$

Donde  $\pi(D(x)) = \frac{|D(x)|}{|U|}$ , de igual forma  $cov_x(C, D) = \pi_x(D, C)$ .

Si  $C \rightarrow_x D$  es una regla de decisión entonces  $D \rightarrow_x C$  es llamada una regla de decisión inversa. Las reglas de decisión inversas pueden ser usadas como explicaciones o razones para una decisión.

De esta forma, si  $C \rightarrow_x D$  es una regla de decisión, entonces:

$$\bigcup_{y \in D(x)} \{C(y) : C(y) \subseteq D(x)\} \quad (3.21)$$

Es la regla de aproximación más baja de la clase de decisión  $D(x)$ , en las clases de condición  $C(x)$ , mientras que el conjunto:

$$\bigcup_{y \in D(x)} \{C(y) : C(y) \cap D(x) \neq \emptyset\} \quad (3.22)$$

indica la aproximación más alta de la clase de decisión en las clases de condición  $C(y)$ .

Las aproximaciones y las reglas de decisión son dos diferentes métodos para expresar propiedades de los datos. Las aproximaciones son apropiadas para expresar propiedades topológicas de los datos, mientras las reglas de decisión describen en una forma simple patrones no visibles de los datos.

### 3.4.5 Árboles de decisión (BF-Tree)

Los árboles de decisión, o en inglés BF-Tree, son considerados una de las técnicas más populares para la representación del conocimiento [75][77]. Este modelo analiza sistemáticamente la información contenida en el conjunto de datos fuente para extraer las reglas de los valores y las relaciones usadas comúnmente para propósitos de clasificación o predicción, siguiendo el esquema de divide y vencerás [78]. En comparación con otras técnicas de clasificación, los árboles de decisión son ampliamente aplicados en varias áreas dada su robustez [76][77].

Un árbol de decisión consiste en partir de forma recursiva el conjunto de datos hasta que todos los datos queden identificados dentro de una clase. Una estructura de árbol de decisión está formada por una raíz y nodos terminales y no terminales. La mayoría de los clasificadores basados en árboles de decisión desempeñan la clasificación en dos fases: la primera es la construcción o creación del árbol, y la segunda, la de podado o reducción de instancias. La construcción del árbol se realiza de arriba hacia abajo, fase durante la cual el árbol es recursivamente particionado hasta que todas las instancias perteneciente a una clase sean identificadas. En la fase de podado, el árbol en su totalidad es cortado de abajo a arriba para prevenir sobreclasificación de elementos y mejorar la exactitud de dicho árbol. Normalmente se mejora la exactitud de predicción y clasificación del algoritmo al reducir el conjunto y al eliminar las instancias con ruido junto con las que producen sobreclasificación.

Dentro de las técnicas de representación de árboles está la del enfoque BF-Tree, en la cual la creación de ramas es realizada mediante la selección de la mejor partición o división del conjunto de datos, asignando cada partición a cada rama que extienda del nodo raíz. En este enfoque la selección de estos subconjuntos está basada en el algoritmo Boosting [79], el cual es utilizado para expandir una rama localizando la mejor partición del conjunto disponible, más que por un orden fijo. De esta manera se obtiene la máxima reducción de la impureza entre todos los nodos disponibles para dividir. Para lograr esta reducción, además, el BF-Tree utiliza los criterios de medición de impureza de los nodos basados en el índice *gain* y *gini*, obteniendo así el mejor nodo en la fase de crecimiento del árbol de decisión [80]. Dado que se selecciona el mejor primero (*best first*, BF), este modelo permite la utilización de criterios como Cross validation para seleccionar el número de expansiones, de tal forma que el podado previo y posterior sobre el árbol de decisión pueda ser realizado, eliminando de esta forma los elementos que conducen a clasificaciones erróneas, además de ramas innecesarias.

Durante el proceso de construcción, tres cosas deben ser consideradas: la primera es encontrar el mejor atributo para representar cada nodo; la segunda es encontrar qué nodo en la lista de nodos será expandido posteriormente; y la tercera es establecer el punto en el que el árbol deberá dejar de crecer.

Un BF-Tree es construido de la siguiente forma:

1. Se selecciona un atributo para ubicar el nodo raíz del árbol e iniciar la creación de ramas para este atributo bajo algún criterio;
2. Se parte el conjunto de instancias de prueba en subconjuntos, uno para cada rama (aunque el árbol puede tener  $n$  número de ramas, es común que estas sean dos). Este paso es repetido para la rama seleccionada usando únicamente aquellas instancias de su subconjunto;
3. De cada iteración, se calcula el mejor subconjunto de entre todos los subconjuntos disponibles para la expansión;
4. El proceso se repite hasta que todos los nodos son puros o se alcanza un número de expansiones específico.

# Capítulo 4

## ***Materiales y métodos***

En el presente capítulo se muestra el material requerido para entender los conceptos utilizados en los capítulos siguientes y que sirven como fundamento para definir el nuevo algoritmo del clasificador Gamma. Además, se muestran algunos conceptos adicionales referentes a los criterios utilizados para realizar el estudio comparativo del Capítulo 6 correspondiente a los resultados y discusión.

### **4.1 Clasificador Gamma**

En este apartado se presentan los conceptos, materiales y métodos que se requieren para describir y fundamentar teóricamente el clasificador Gamma.

El clasificador Gamma, inicialmente introducido en [81], es un clasificador de patrones de alto desempeño que está basado en las memorias asociativas Alfa-Beta y cuyo principal operador es el operador de similitud  $\gamma$ . Este clasificador, que pertenece al enfoque asociativo de reconocimiento de patrones, trabaja en conjunto con los operadores  $\alpha$  y  $\beta$  y el operador  $u_{\beta}$ , además de hacer uso de la codificación Johnson-Möbius modificada para la codificación de los patrones.

A continuación se describen las bases teóricas del clasificador Gamma. De igual forma al final de la sección se muestra el algoritmo de dicho clasificador. Todas las definiciones mostradas en este apartado son tomadas directamente de [82].

#### **4.1.1 Operador alfa y operador beta**

Los dos operadores principales de las memorias asociativas alfa-beta, para los conjuntos  $A=\{0,1\}$  y  $B=\{0,1,2\}$ , están representados por las siguientes tablas de verdad (Tabla 4.1) [82]:

El operador  $\alpha$  es utilizado en la fase de aprendizaje, mientras que el operador  $\beta$  es útil durante la fase de recuperación.

$\alpha : A \times A \rightarrow B$		
x	y	$\alpha(x,y)$
0	0	01
0	1	00
1	0	10
1	1	01

$\beta : B \times A \rightarrow A$		
X	y	$\alpha(x,y)$
00	0	0
00	1	0
01	0	0
01	1	1
10	0	1
10	1	1

Tabla 4.1. Definición de los operadores  $\alpha$  y  $\beta$

### 4.1.2 Operador $u_\beta$

**Definición 4.1** [82]. Sean el conjunto  $A=\{0,1\}$ , un número  $n \in \mathbb{Z}^+$  y  $x \in A^n$  un vector binario de dimensión  $n$ , con la  $i$ -ésima componente representada por  $x_i$ . Se define el operador  $u_\beta(x)$  como:

$$u_\beta(x) = \sum_{i=1}^n \beta(x_i, x_i) \quad (4.1)$$

Nótese que  $u_\beta(x)$  tiene como argumento de entrada un vector binario  $x$  de  $n$  dimensiones y la salida es un número entero no negativo, resultado de la expresión (4.1).

### 4.1.3 Módulo y congruencia

**Definición 4.2** [82]. Sean  $a$  un número entero y  $m$  un número entero positivo. Se denota por  $a \bmod m$  al residuo de dividir  $a$  entre  $m$ .

Dicho de otra manera,  $a \bmod m$  es el número entero  $r$  tal que  $a = qm + r$  y  $0 \leq r < m$ .

Cabe mencionar que cuando dos números tienen el mismo residuo con respecto a un módulo  $m$  dado, se dice que son congruentes.

Nótese que  $a \equiv b \bmod m$ , si y sólo si  $a \bmod m = b \bmod m$ .

### 4.1.4 Código binario Johnson-Möbius modificado

**Algoritmo 4.1** [82]. Algoritmo del Código Johnson-Möbius modificado:

1. Sea un conjunto de números reales:

$$\{r_1, r_2, \dots, r_i, \dots, r_n\}$$

donde  $n$  es un número entero positivo fijo.

2. Si uno de los números del conjunto (por ejemplo  $r_i$ ) es negativo, se crea un nuevo conjunto transformando a través de restar  $r_i$  a cada uno de los  $n$  números:

$$\{t_1, t_2, \dots, t_i, \dots, t_n\}$$

donde  $t_j = r_j - r_i \forall j \in \{1, 2, \dots, n\}$  y particularmente  $t_i = 0$ .

*Nota: si hay más de un negativo, se trabaja con el menor.*

3. Escoger un número fijo  $d$  de decimales y truncar cada uno de los números del conjunto transformado (los cuales son no negativos) precisamente a  $d$  decimales.
4. Realizar un escalamiento de  $10d$  en el conjunto del paso 3, para obtener un conjunto de  $n$  enteros no negativos:
 
$$\{e_1, e_2, \dots, e_i, \dots, e_n\}$$
 donde  $e_m$  es el número mayor.
5. Así, el código Johnson-Möbius modificado para cada  $j=1, 2, \dots, n$  se obtiene al generar  $(e_m - e_j)$  ceros concatenados por la derecha con  $e_j$  unos.

### 4.1.5 Operador Gamma de similitud

El operador Gamma de similitud generalizado se relaciona, en primer lugar, a los operadores fundamentales de las memorias asociativas alfa-beta, aunque de una forma novedosa, de ahí que se nombre con la siguiente letra del alfabeto griego, gamma. En segundo lugar, dado que este operador indica si dos vectores son parecidos o no, considerando un grado de similitud  $\theta$ , es que se le denomina de similitud. En este sentido, el argumento  $\theta$  indica la tolerancia para la cual dos vectores, al compararlos, sean considerados similares –no obstante si son diferentes-. Por otra parte, este operador generaliza al operador Gamma de similitud, permitiéndole operar con vectores binarios de dimensiones diferentes. De [82] se toma la siguiente definición:

**Definición 4.3** [82]. Sean el conjunto  $A=\{0, 1\}$ , dos números  $n, m \in \mathbb{Z}^+$ ,  $n \leq m$ ,  $x \in A^n$  y  $y \in A^m$  dos vectores binarios,  $n$ -dimensionales y  $m$ -dimensional respectivamente, con la  $i$ -ésima componente representada por  $x_i$  y  $y_i$ , respectivamente, y además,  $\theta$  un número entero no negativo. Se define el operador gamma de similitud generalizado  $\gamma_g(x, y, \theta)$  como:

$$\gamma_g(x, y, \theta) = \begin{cases} 1 & \text{si } m - u_\beta[\alpha(x, y) \bmod 2] \leq \theta \\ 0 & \text{en otro caso} \end{cases} \quad (4.2)$$

Donde si  $m > n$ , se truncan los  $m - n$  bits izquierdos de  $y$  antes de realizar el cálculo. Nótese que  $\gamma(x, y, \theta)$  tiene como argumentos de entrada dos vectores binarios  $n$ -dimensionales  $x$  y  $y$ , y un número entero no negativo  $\theta$ .

### 4.1.6 Algoritmo del clasificador Gamma

Como se mencionó al inicio de este apartado, este clasificador hace uso de los operadores fundamentales de las memorias asociativas alfa-beta, así como del operador Gamma de similitud generalizado, que a su vez utiliza los conceptos del operador  $u_\beta$  y de módulo, además de poner en práctica de forma adicional sus propiedades al ser utilizado con patrones codificados con el código Johnson-Möbius modificado, para, de esta forma, llevar a cabo de manera eficaz y eficiente la tarea de clasificación de patrones. De ahí que este clasificador se haya denominado Gamma, y utilizando la ampliamente conocida base de datos *iris plant* en los experimentos realizados, se mostró su alto desempeño en cuanto a la clasificación y recuperación de patrones, por lo que es utilizado en el presente trabajo de tesis para probar experimentalmente su desempeño con imágenes binarias

haciendo uso de los momentos invariantes. De igual forma se propone un nuevo algoritmo para su simplificación.

A continuación se presenta el algoritmo del clasificador Gamma, tal y como se propuso originalmente en [82].

**Algoritmo 4.2** [82]. Sean los números  $k, m, n, p \in \mathbb{Z}^+$ ,  $\{x^\mu \mid \mu=0,1,2,\dots,p\}$  el conjunto fundamental de patrones de cardinalidad  $p$ , donde  $\forall \mu x^\mu \in \mathbb{R}^n$  y  $y \in \mathbb{R}^n$  es un vector real de  $n$  dimensiones a ser clasificado. Se asume que el conjunto fundamental está particionado en  $m$  clases diferentes, donde cada clase tiene cardinalidad  $k_i, i=1,2,\dots,m$ , por lo que

$$\sum_{i=1}^m k_i = p \quad (4.3)$$

De ahí que para clasificar el patrón  $y$ , se realiza lo siguiente:

1. Codificar cada componente de cada patrón del conjunto fundamental con el código Johnson-Möbius modificado, obteniéndose el valor  $e_m$  para cada componente como:

$$e_m = \vee_{i=1}^{\rho} x_j^i \quad (4.4)$$

donde  $\vee$  indica el valor máximo de cada componente.

Así, la componente  $x_j^i$  se transforma en un vector binario de dimensión  $e_m(j)$ .

2. Calcular el parámetro de paro  $\rho$  de la siguiente forma:

$$\rho = \wedge_{j=1}^n e_m(j) \quad (4.5)$$

donde  $\wedge$  indica el valor mínimo del vector.

3. Codificar cada componente con el código Johnson-Möbius modificado, utilizando las mismas condiciones que se utilizaron para codificar las componentes de los patrones fundamentales. En caso que alguna componente del patrón a clasificar sea mayor al  $e_m$  correspondiente ( $y_\xi > e_m(\xi)$ ), igualar esa componente a  $e_m(\xi)$  y guardar su valor anterior en la variable  $nmax_\xi$ .
4. Realizar una transformación de índices en los patrones del conjunto fundamental, de manera que el índice único que tenía un patrón originalmente en el conjunto fundamental, por ejemplo  $x^\mu$ , se convierta en dos índices, uno para la clase (por ejemplo la clase  $i$ ) y otro para el orden que le corresponde a ese patrón dentro de dicha clase (por ejemplo  $\omega$ ). Bajo estas condiciones ejemplificadas, la notación para el patrón  $x^\mu$ , será ahora, con la transformación,  $x^{i\omega}$ . Lo anterior se realiza para todos los patrones del conjunto fundamental.
5. Inicializar  $\theta = 0$ .
6. Realizar la operación  $\gamma_g(x_j^{i\omega}, y_j, \theta)$  para cada clase y para cada componente de cada uno de los patrones fundamentales que corresponden a esa clase, y del patrón a clasificar, considerándose  $nmax_\xi$  como la dimensión del patrón binario  $y_\xi$ .

7. Calcular la suma ponderada  $c_i$  de los resultados obtenidos en el paso 6, para cada clase  $i = 1, 2, \dots, m$ :

$$c_i = \frac{\sum_{\omega=1}^{k_i} \sum_{j=1}^n \gamma_g(x_j^{i\omega}, y_j, \theta)}{k_i} \quad (4.6)$$

8. Si existe más de un máximo entre las sumas ponderadas por clase, se incrementa  $\theta$  en 1 y se repiten los pasos 6 y 7 hasta que exista un máximo único, o se cumpla con la condición de paro  $\theta \geq \rho$ .
9. Si existe un máximo único, denotado por  $\vee$ , se asigna al patrón a clasificar la clase correspondiente a dicho máximo:

$$C_y = C_j \text{ tal que } \vee_{i=1}^m c_i = c_j \quad (4.7)$$

10. En caso contrario, si  $\lambda$  es el índice más pequeño de la clase que corresponde a uno de los máximos, se asigna al patrón a clasificar la clase  $C_\lambda$ .

Nótese que como los patrones con los que se opera  $\gamma_g(x_j^{i\omega}, y, \theta)$  en este algoritmo están codificados con el código Johnson-Möbius modificado, todos los patrones en la  $i$ -ésima componente tendrán la misma dimensión  $e_m(j)$ , siendo la única excepción cuando  $y_\xi > e_m(\xi)$ , en cuyo caso se trunca a  $e_m(\xi)$  y  $\gamma_g(x_j^{i\omega}, y_j, \theta)$  toma  $mgamma_\xi$  como dimensión de  $y_\xi$ .

## 4.2 Evaluación del desempeño de clasificación

El desempeño de los clasificadores es medido en términos de porcentaje de recuperación. El clasificador predice la clase de cada instancia etiquetándolas como correcta, si esta fue exitosa, o errónea en el caso contrario. Así, el factor de recuperación es la proporción de aciertos realizado sobre un conjunto completo de instancias y es utilizado para medir de forma estadística el desempeño de un clasificador.

Uno de los métodos más ampliamente utilizados en la literatura para la estimación del desempeño de un clasificador es el llamado Cross validation [80].

Cross validation es una técnica estadística utilizada en el campo de la inteligencia artificial para estimar el desempeño de un clasificador, la cual consiste en partir el conjunto de datos de prueba del clasificador en determinado número ( $k$ ) de partes (*folds*). De tal forma que una de las partes sea tomada como conjunto de prueba, y las ( $k-1$ ) restantes para conjunto de datos de entrenamiento. Este procedimiento es repetido para cada una de los subconjuntos obtenidos ( $k$ ). Al final, el factor resultante es obtenido al aplicar la media aritmética sobre los resultados de cada una de las ( $k$ ) iteraciones (Figura 4.1).

Cross validation es una variante del método Holdout, el cual consiste en dividir el conjunto de datos de muestra en dos partes de forma que cada uno se emplee en la fase de entrenamiento y de recuperación respectivamente [83]. El factor final es el resultado de recuperación de este único proceso. Aunque este método es muy rápido, dado que sólo requiere la iteración de un único conjunto de prueba, este no se considera un método

confiable ya que la muestra considerada para realizar la fase de entrenamiento podría no ser la de mejor calidad de las del conjunto en cuanto a la utilidad de la información provista por sus elementos.

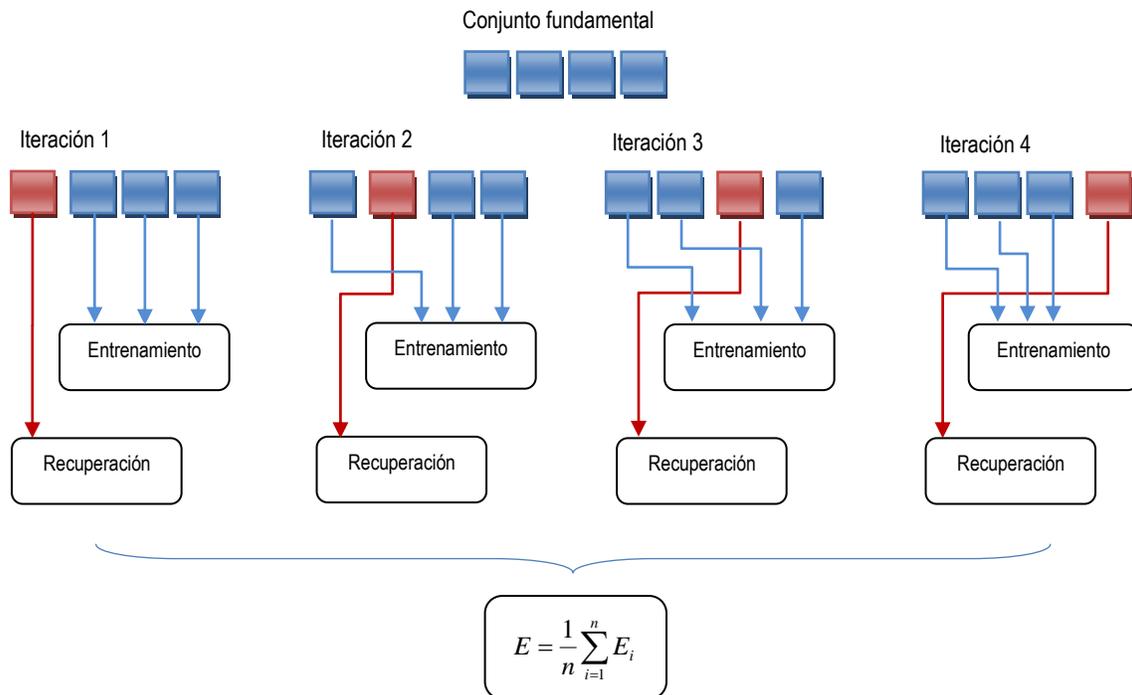


Figura 4.1. *k-fold cross validation*, con  $k=4$ .

Una derivación directa de Cross validation es Leave one out, la cual consiste en realizar la partición del conjunto en muestras del tamaño del conjunto a evaluar, es decir, que se hacen tantas muestras como haya elementos en el conjunto, tal que la cardinalidad de cada subconjunto es 1, de ahí el nombre del método, dejando únicamente un elemento para la etapa de prueba y el  $(n-1)$  restante para entrenamiento. Cada elemento o patrón en el conjunto es considerado en ambas etapas, entrenamiento y recuperación por lo menos una vez, haciéndolo un método muy preciso, pero lento. Similar a Cross validation, el proceso es repetido  $n$  veces y de la misma manera el resultado es obtenido al aplicar la media aritmética de todos los resultados de las iteraciones.

A pesar de que el método Leave one out involucra la evaluación de todos los elementos en el conjunto, de ahí su alto costo computacional, y a diferencia de los anteriores, este presenta dos ventajas: la primera, al tener como conjuntos de entrenamiento grandes cantidades de datos en cada iteración se incrementa la posibilidad de que el clasificador sea más preciso; y la segunda, es que el procedimiento se vuelve determinístico al eliminar las muestras aleatorias, produciendo los mismos resultados sin importar las veces que se repita la evaluación.

# Capítulo 5

## *Método propuesto*

En el Capítulo 3 se dieron algunos métodos para extracción de rasgos de una imagen a partir del cálculo de momentos geométricos, rasgos con propiedades invariantes a algunas transformaciones lineales de la imagen.

En el Capítulo 4 se introdujeron los conceptos necesarios para poder desarrollar el presente capítulo, el de principal interés de este trabajo de tesis, en el cual se muestra experimentalmente el desempeño del clasificador Gamma con imágenes binarias, utilizando como descriptores de estas, distintas combinaciones probadas de los momentos invariantes de Hu.

Finalmente, se introduce un nuevo algoritmo para simplificar el clasificador Gamma. Este algoritmo ofrece entre sus principales mejoras, la reducción de los pasos que involucran la codificación de las componentes en código Johnson-Möbius modificado, paso cuyo cálculo presenta alta latencia, en términos de costo computacional.

### **5.1 Algoritmo del clasificador Gamma simplificado**

El clasificador Gamma, aunque ha mostrado tener un desempeño sobresaliente en comparación con los mejores clasificadores de la literatura, mostrando tener resultados semejantes, e incluso superiores para algunos experimentos, presenta inconvenientes en cuanto al consumo de recursos computacionales requeridos para tratar componentes numéricas con cifras compuestas por conjuntos grandes de dígitos. Este problema se presenta específicamente al codificar en código Johnson-Möbius modificado, ya que de forma general esta codificación requiere que se modele, en unos, el número entero de la componente previamente escalada y trasladada. Sin mencionar los cálculos posteriores para procesar el patrón desconocido a clasificar con el operador  $\alpha$  para cada elemento de la componente codificada. Si bien las operaciones a las cuales son sometidas las componentes codificadas son simples (operadores  $\alpha$  y  $\beta$ , módulo y  $u_{\beta}$ ), lo que eleva el costo computacional son las largas iteraciones sobre las codificaciones de las componentes haciendo compleja y poco factible su implementación con números grandes.

De ahí que, y como parte de las contribuciones de este trabajo, se muestre un nuevo algoritmo para el clasificador Gamma que permite reducir en gran medida el costo computacional implicado en la codificación de las componentes y sus operaciones con el patrón a clasificar. Mediante la eliminación de dicha codificación y permitiendo el trato directo con las componentes en términos de enteros positivos, se consigue la

simplificación del algoritmo, dejando así el esfuerzo de procesado, únicamente, en las iteraciones de comparación del conjunto fundamental contra el patrón a clasificar.

A continuación se muestra el algoritmo del clasificador Gamma con el criterio propuesto de mejora para su simplificación, el cual fue implementado para ser probado en la ejecución de los conjuntos de experimentos descritos en el siguiente capítulo, correspondiente a los resultados y discusión:

**Definición 5.1.** Sean dos números  $n, m \in \mathbb{Z}^+$ ,  $n \leq m$ ,  $x$  y  $y \in \mathbb{Z}^+$ , y además,  $\theta$  un número entero no negativo. Se define el operador gamma de similitud simplificado  $\gamma_s(x, y, \theta)$  de la siguiente manera:

$$\gamma_s(x, y, \theta) = \begin{cases} 1 & \text{si } m - (\wedge(\text{ceros}X, \text{ceros}Y) + \wedge(\text{unos}X, \text{unos}Y)) \leq \theta \\ 0 & \text{en otro caso} \end{cases} \quad (5.1)$$

donde  $\wedge$  indica el valor mínimo de entre los valores dados; si  $m > n$ , se truncan los  $m-n$  bits izquierdos de  $y$  antes de realizar el cálculo; y  $\text{ceros}X = (e_m - x)$ ,  $\text{unos}X = (x)$ ,  $\text{ceros}Y = (e_m - y)$  y  $\text{unos}Y = (y)$ . Con  $e_m$  el máximo valor de todas las componentes para el índice dado.

Nótese que  $\gamma_s(x, y, \theta)$  tiene como argumentos de entrada dos enteros positivos  $x$  y  $y$ , y un número entero no negativo  $\theta$ .

**Algoritmo 5.1** Sean los números  $k, m, n, p \in \mathbb{Z}^+$ ,  $\{x^\mu \mid \mu=0,1,2,\dots,p\}$  el conjunto fundamental de patrones de cardinalidad  $p$ , donde  $\forall \mu x^\mu \in \mathbb{R}^n$  y  $y \in \mathbb{R}^n$  es un vector real de  $n$  dimensiones a ser clasificado. Se asume que el conjunto fundamental está particionado en  $m$  clases diferentes, donde cada clase tiene cardinalidad  $k_i$ ,  $i=1, 2, \dots, m$ , por lo que

$$\sum_{i=1}^m k_i = p \quad (5.2)$$

De ahí que para clasificar el patrón  $y$ , se realiza lo siguiente:

1. Calcular el parámetro de paro  $\rho$  de la siguiente forma:

$$\rho = \wedge_{j=1}^n e_m(j) \quad (5.3)$$

donde  $\wedge$  indica el valor mínimo del vector.

2. En caso que alguna componente del patrón a clasificar sea mayor al  $e_m$  correspondiente ( $y_\xi > e_m(\xi)$ ), igualar esa componente a  $e_m(\xi)$  y guardar su valor anterior en la variable  $nmax_\xi$ .
3. Realizar una transformación de índices en los patrones del conjunto fundamental, de manera que el índice único que tenía un patrón originalmente en el conjunto fundamental, por ejemplo  $x^\mu$ , se convierta en dos índices, uno para la clase (por ejemplo la clase  $i$ ) y otro para el orden que le corresponde a ese patrón dentro de dicha clase (por ejemplo  $\omega$ ). Bajo estas condiciones ejemplificadas, la notación para el patrón  $x^\mu$  será ahora, con la transformación,  $x^{i\omega}$ . Lo anterior se realiza para todos los patrones del conjunto fundamental.
4. Inicializar  $\theta = 0$ .

5. Calcular la suma ponderada  $c_i$  para cada clase  $i = 1, 2, \dots, m$ :

$$c_i = \frac{\sum_{\omega=1}^{k_i} \sum_{j=1}^n \gamma_s(x_j^{i\omega}, y_j, \theta)}{k_i} \quad (5.4)$$

6. Si existe más de un máximo entre las sumas ponderadas por clase, se incrementa  $\theta$  en 1 y se repiten los pasos 5 y 6 hasta que exista un máximo único, o se cumpla con la condición de paro  $\theta \geq \rho$ .
7. Si existe un máximo único, denotado por  $\vee$ , asignar al patrón a clasificar la clase correspondiente a dicho máximo:

$$C_y = C_j \text{ tal que } \vee_{i=1}^m c_i = c_j \quad (5.5)$$

8. En caso contrario, si  $\lambda$  es el índice más pequeño de la clase que corresponde a uno de los máximos, se asigna al patrón a clasificar la clase  $C_\lambda$ .

Nótese los patrones con los que opera la nueva definición  $\gamma_s(x_j^{i\omega}, y, \theta)$ , que en este algoritmo trabajan con las componentes escaladas y trasladadas directamente como resultado de las dos restas y la suma, obteniéndose de esta forma el equivalente al operador  $u_\beta$  y eliminando completamente las operaciones relacionadas con los vectores binarios y de los operadores  $\alpha$  y  $\beta$ . De ahí que el alto costo computacional, al tratar con números grandes expresados en vectores binarios de gran tamaño, sea eliminado.

**Ejemplo 5.1** Sean los patrones fundamentales:

$$x^1 = \begin{pmatrix} 4.8 \\ 3.4 \\ 1.6 \\ 0.2 \end{pmatrix}, x^2 = \begin{pmatrix} 4.4 \\ 3.0 \\ 1.3 \\ 0.2 \end{pmatrix}, x^3 = \begin{pmatrix} 6.4 \\ 2.9 \\ 4.3 \\ 1.3 \end{pmatrix}, x^4 = \begin{pmatrix} 6.0 \\ 2.7 \\ 5.1 \\ 1.6 \end{pmatrix}, x^5 = \begin{pmatrix} 6.3 \\ 3.3 \\ 6.0 \\ 2.5 \end{pmatrix} \text{ y } x^6 = \begin{pmatrix} 5.8 \\ 2.7 \\ 5.1 \\ 1.9 \end{pmatrix},$$

agrupados en tres clases  $C_1 = \{x^1, x^2\}$ ,  $C_2 = \{x^3, x^4\}$  y  $C_3 = \{x^5, x^6\}$ ;

$$\text{Clasificar el patrón } y = \begin{pmatrix} 4.6 \\ 3.4 \\ 1.4 \\ 0.3 \end{pmatrix}$$

Se puede observar que la dimensión de los patrones es  $n=4$ , y que se tienen  $m=3$  clases, donde las clases agrupan  $k_1=2$ ,  $k_2=2$  y  $k_3=2$  patrones, respectivamente. Se realizan las operaciones de escalamiento de componentes requeridas para proceder con el algoritmo, quedando los patrones de la siguiente manera:

$$x^1 = \begin{pmatrix} 48 \\ 34 \\ 16 \\ 2 \end{pmatrix}, x^2 = \begin{pmatrix} 44 \\ 30 \\ 13 \\ 2 \end{pmatrix}, x^3 = \begin{pmatrix} 64 \\ 29 \\ 43 \\ 13 \end{pmatrix}, x^4 = \begin{pmatrix} 60 \\ 27 \\ 51 \\ 16 \end{pmatrix}, x^5 = \begin{pmatrix} 63 \\ 33 \\ 60 \\ 25 \end{pmatrix}, x^6 = \begin{pmatrix} 58 \\ 27 \\ 51 \\ 19 \end{pmatrix} \text{ y } y = \begin{pmatrix} 46 \\ 34 \\ 14 \\ 3 \end{pmatrix}$$

Entonces siguiendo los pasos del algoritmo:

1. Dado que  $e_m(1)=64$ ,  $e_m(2)=34$ ,  $e_m(3)=60$ ,  $e_m(4)=25$ , se tiene que el parámetro de paro  $\rho$ :

$$\rho = \bigwedge_{j=1}^3 e_m(j) = 25$$

2. Se utilizan las mismas  $e_m(j)$ , las cuales se mantienen igual, dado que ninguna componente del patrón a clasificar es mayor.
3. Se agrupan los patrones fundamentales de acuerdo con sus clases  $C_1=\{x^1, x^2\}$ ,  $C_2=\{x^3, x^4\}$  y  $C_3=\{x^5, x^6\}$ , de la siguiente manera:  $x^1 \rightarrow x^{11}$ ,  $x^2 \rightarrow x^{12}$ ,  $x^3 \rightarrow x^{21}$ ,  $x^4 \rightarrow x^{22}$ ,  $x^5 \rightarrow x^{31}$  y  $x^6 \rightarrow x^{32}$ .
4. Se asigna  $\theta = 0$ .
5. Se realiza el cálculo para clasificar el patrón  $y$ .

$$\begin{array}{cccc} \gamma_s(x_1^{11}, y_1^1, 0)=0 & \gamma_s(x_2^{11}, y_2^1, 0)=0 & \gamma_s(x_3^{11}, y_3^1, 0)=0 & \gamma_s(x_4^{11}, y_4^1, 0)=0 \\ \gamma_s(x_1^{12}, y_1^1, 0)=0 & \gamma_s(x_2^{12}, y_2^1, 0)=0 & \gamma_s(x_3^{12}, y_3^1, 0)=0 & \gamma_s(x_4^{12}, y_4^1, 0)=0 \\ \gamma_s(x_1^{21}, y_1^1, 0)=0 & \gamma_s(x_2^{21}, y_2^1, 0)=0 & \gamma_s(x_3^{21}, y_3^1, 0)=0 & \gamma_s(x_4^{21}, y_4^1, 0)=0 \\ \gamma_s(x_1^{22}, y_1^1, 0)=0 & \gamma_s(x_2^{22}, y_2^1, 0)=0 & \gamma_s(x_3^{22}, y_3^1, 0)=0 & \gamma_s(x_4^{22}, y_4^1, 0)=0 \\ \gamma_s(x_1^{31}, y_1^1, 0)=0 & \gamma_s(x_2^{31}, y_2^1, 0)=0 & \gamma_s(x_3^{31}, y_3^1, 0)=0 & \gamma_s(x_4^{31}, y_4^1, 0)=0 \\ \gamma_s(x_1^{32}, y_1^1, 0)=0 & \gamma_s(x_2^{32}, y_2^1, 0)=0 & \gamma_s(x_3^{32}, y_3^1, 0)=0 & \gamma_s(x_4^{32}, y_4^1, 0)=0 \end{array}$$

Se suman los resultados anteriores y se tiene que  $C_1=0$ ,  $C_2=0$  y  $C_3=0$ . Dado que no hay un resultado superior a los otros, se incrementa  $\theta = 1$ , y se realizan de nuevo los cálculos:

$$\begin{array}{cccc} \gamma_s(x_1^{11}, y_1^1, 1)=0 & \gamma_s(x_2^{11}, y_2^1, 1)=0 & \gamma_s(x_3^{11}, y_3^1, 1)=0 & \gamma_s(x_4^{11}, y_4^1, 1)=0 \\ \gamma_s(x_1^{12}, y_1^1, 1)=0 & \gamma_s(x_2^{12}, y_2^1, 1)=0 & \gamma_s(x_3^{12}, y_3^1, 1)=0 & \gamma_s(x_4^{12}, y_4^1, 1)=0 \\ \gamma_s(x_1^{21}, y_1^1, 1)=0 & \gamma_s(x_2^{21}, y_2^1, 1)=0 & \gamma_s(x_3^{21}, y_3^1, 1)=0 & \gamma_s(x_4^{21}, y_4^1, 1)=0 \\ \gamma_s(x_1^{22}, y_1^1, 1)=0 & \gamma_s(x_2^{22}, y_2^1, 1)=0 & \gamma_s(x_3^{22}, y_3^1, 1)=1 & \gamma_s(x_4^{22}, y_4^1, 1)=0 \\ \gamma_s(x_1^{31}, y_1^1, 1)=0 & \gamma_s(x_2^{31}, y_2^1, 1)=0 & \gamma_s(x_3^{31}, y_3^1, 1)=0 & \gamma_s(x_4^{31}, y_4^1, 1)=0 \\ \gamma_s(x_1^{32}, y_1^1, 1)=1 & \gamma_s(x_2^{32}, y_2^1, 1)=0 & \gamma_s(x_3^{32}, y_3^1, 1)=1 & \gamma_s(x_4^{32}, y_4^1, 1)=1 \end{array}$$

Nuevamente se suman los resultados anteriores y se tiene que  $C_1=0$ ,  $C_2=1/2$  y  $C_3=3/2$ .

6. Existe un máximo único tal que  $\bigvee_{i=1}^3 c_i = c_3 = \frac{3}{2}$

Finalmente, al patrón  $y$  se le asigna la clase  $C_3$ . ■

# Capítulo 6

## Resultados y discusión

En éste capítulo se detalla el conjunto de ejercicios realizados, con la finalidad de mostrar experimentalmente el desempeño del clasificador Gamma con imágenes binarias caracterizadas con los invariantes de Hu, utilizando un nuevo algoritmo del clasificador Gamma. Este capítulo evalúa el desempeño del clasificador Gamma con distintos conjuntos de prueba y a continuación se presenta el detalle.

Los experimentos realizados se implementaron en un programa de computadora utilizando Borland C++ Builder 6.0 como plataforma de desarrollo. Dicho programa se ejecutó en una computadora personal con microprocesador de la familia CORE i7 de Intel de 64 bits a 2.9 MHz, con 6 GB de memoria DDR3 SDRAM a 1333 MT/s y con disco duro de 750 GB a 5400 rpm, sobre el sistema operativo *Windows 7 Home Edition* x64 bits.

En cuanto a los conjuntos de datos de las imágenes utilizadas en los experimentos, se explotaron fuentes de datos creadas de forma artificial, las cuales son descritas en la tabla 6.1. Cada clase en los conjuntos de prueba está conformada por un conjunto de elementos los cuales presentan alguna transformación geométrica, por ejemplo rotación, traslación o cambio de escala. Los criterios considerados para aplicar estas transformaciones geométricas fueron: para cambio de escala, replicación de píxeles, y para la rotación, el del punto más cercano. Para todos los experimentos mencionados se utilizaron imágenes binarias (1 bit de profundidad) en formato *.bmp*, de tamaño *100x100* píxeles, salvo en el caso del conjunto de datos número 2 donde se tratan imágenes entre *100x100* y *200x200* píxeles. Las transformaciones geométricas fueron realizadas utilizando el editor de imágenes *Microsoft Photo Editor*.

#	CONJUNTO DE DATOS	ELEMENTOS	CLASES	INSTANCIAS POR CLASE	OBSERVACIONES
1	Dígitos 0-9	110	10	11	Imágenes de 100x100 con tipo de letra Arial
2	Alfabeto en mayúsculas (A-Z)	286	26	11	Imágenes de entre 100x100 y 200x200 con tipo de letra Arial
3	Alfabeto en mayúsculas (A-Z)	286	26	11	Imágenes de 100x100 con tipo de letra Times New Roman
4	Alfabeto en minúsculas (a-z)	286	26	11	Imágenes de 100x100 con tipo de letra Times New Roman

Tabla 6.1. Relación de conjuntos de datos.

Los algoritmos de clasificación presentados en los estudios comparativos son tomados de la plataforma de Machine Learning, WEKA [83][84]. Misma que se utilizó como medio

estandarizado de pruebas. WEKA es una herramienta informática, la cual tiene como objetivo principal servir de apoyo para la aplicación de técnicas de representación de información en el área de Machine Learning para una variedad de problemas del mundo real. WEKA esta orientada a proveer un marco de trabajo para los especialistas. En general, esta herramienta provee un ambiente integrado el cual no sólo da fácil acceso a distintas técnicas de representación de información sino que además incorpora herramientas para el pre y post procesamiento, esenciales cuando se trabaja con conjuntos de datos reales. Asimismo, y para complementar las herramientas básicas, WEKA ofrece un conjunto de utilidades adicionales para la manipulación de datos, visualización de resultados, vinculación con bases de datos, con técnicas de evaluación de desempeño como Cross validation y comparación de distintos conjuntos de reglas, motivo por el cual se utilizará de apoyo en la parte experimental de este trabajo.

Para la parte experimental se seleccionaron seis de los algoritmos pertenecientes a los enfoques de clasificación más representativos descritos en el capítulo de antecedentes de este trabajo, implementados en WEKA, los cuales se muestran en la tabla 6.2.

#	CLASIFICADOR
1	IB1
2	<i>Naïve Bayes</i>
3	<i>Bagging</i>
4	<i>Decision Table</i>
5	<i>BF Tree</i>
6	<i>Multilayer Perceptron</i>

Tabla 6.2. Clasificadores evaluados.

Parte importante del reconocimiento de patrones consiste en seleccionar un conjunto apropiado de atributos que nos permitan representar de forma única a nuestras imágenes. Las técnicas basadas en momentos han sido ampliamente usadas, dadas sus cualidades discriminantes, de ahí que, y como se ha mencionado en los capítulos anteriores, estos se han utilizado en este trabajo como caracterizadores, específicamente los invariantes de Hu,  $\Phi_2$  y  $\Phi_4$ , tomados de las expresiones (3.15), debido a que experimentalmente han mostrado tener mayor poder de discriminación [24].

El capítulo se organiza de la siguiente manera: se presenta de forma inicial, en el apartado 6.1, la comparación del desempeño del clasificador Gamma con los algoritmos arriba mencionados, mostrando dos ejecuciones para cada conjunto de prueba donde se manejan dos distintas longitudes para truncar los decimales de los atributos, 6 y 8 decimales. De igual forma, en el apartado 6.2 se presenta la comparación del desempeño de los clasificadores para señalar el impacto del llamado factor de olvido en estos. La evaluación del desempeño de los clasificadores para el primer experimento es realizada mediante el criterio de estimación Leave one out [2][80][83], mientras que para el segundo caso es mediante la recuperación de cada uno de los elementos utilizados para entrenamiento. Por último, se comparan los tiempos de respuesta del algoritmo Gamma simplificado propuesto contra el del Gamma original para señalar la diferencia de la optimización en términos de tiempo de ejecución de la recuperación de un patrón de prueba cualquiera.

## 6.1 Desempeño del clasificador Gamma

En este apartado se muestran los resultados correspondientes a los porcentajes de recuperación de los conjuntos de datos de la tabla 6.1 para los clasificadores listados en la tabla 6.2, bajo el siguiente procedimiento:

1. Configurar un conjunto de  $m$  clases de imágenes, cada una con un número de elementos formados por transformaciones geométricas de la imagen original o representante de la clase. Particularmente con las que los momentos de Hu presentan invariancia, esto es, rotación, traslación y cambio de escala.
2. Obtener los siete momentos invariantes de cada una de las imágenes que conforman el conjunto fundamental, esto es, para cada una de las imágenes miembro de cada una de las clases en el conjunto.
3. Escoger una combinación de momentos con la cual se desea probar al clasificador. Específicamente la conformada por los momentos  $\Phi_2$  y  $\Phi_4$  de las expresiones de Hu de las expresiones (3.15).
4. Ejecutar el análisis de la combinación dada mediante la técnica de evaluación Leave one out, utilizando el algoritmo 5.1 para el clasificador Gamma simplificado.
5. Para la misma combinación, probar con cada uno de los métodos de clasificación de la tabla 6.2 utilizando la plataforma WEKA para comparar el resultado obtenido.
6. Asentar en una tabla comparativa los resultados agregando los comentarios y conclusiones correspondientes.

La tabla 6.3 ilustra los porcentajes de clasificación de los conjuntos de datos de la tabla 6.1 para cada clasificador. El criterio de estimación utilizado para realizar la evaluación es Leave one out estratificado. Cabe mencionar que para que un conjunto se considere estratificado este debe tener en cada clase o tipo que lo conforman igual número de elementos, es decir, que cada una de las clases esta balanceada. Los porcentajes más altos están mostrados en negritas. Se puede notar que el incremento en el número de decimales de 6 a 8 para truncar los valores no representa impacto significativo en el porcentaje de clasificación mostrado para los clasificadores, excepto para el clasificador Gama, el cual muestra una variación considerable con lo que logra superar claramente a los demás clasificadores. Sin embargo, para el clasificador Gama la precisión considerada si conlleva a que se obtengan mejores porcentajes de recuperación.

CLASIFICADOR	Porcentaje de Clasificación (Leave one out)								PROM
	Dígitos (0-9)		Alfabeto (A-Z) - Arial		Alfabeto (A-Z) – Times New Roman		Alfabeto (a-z) – Times New Roman		
	6	8	6	8	6	8	6	8	
<b>Clasificador Gamma</b>	<b>92.73%</b>	<b>100.00%</b>	81.82%	<b>98.95%</b>	<b>95.10%</b>	<b>100.00%</b>	<b>95.45%</b>	<b>100.00%</b>	<b>96%</b>
Bagging	83.64%	84.55%	<b>95.10%</b>	95.45%	89.51%	89.51%	86.71%	86.71%	<b>89%</b>
IB1	80.00%	80.00%	94.06%	94.06%	87.06%	87.06%	87.76%	87.76%	<b>87%</b>
Náive Bayes	75.45%	76.36%	83.57%	84.97%	84.27%	84.27%	88.46%	87.76%	<b>83%</b>
Decision Table	82.73%	82.73%	77.27%	76.57%	71.68%	70.98%	68.18%	68.18%	<b>75%</b>
BF Tree	0.00%	0.00%	84.27%	71.33%	86.36%	86.71%	88.11%	88.11%	<b>63%</b>
Multilayer Perceptron	67.27%	67.27%	54.90%	54.90%	62.24%	62.24%	43.71%	43.71%	<b>57%</b>

Tabla 6.3. Porcentajes de recuperación de los clasificadores analizados

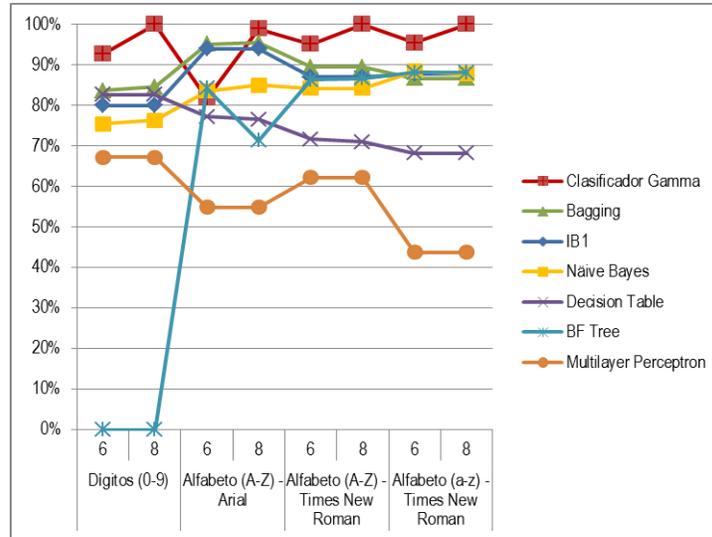


Figura 6.1. Gráfica comparativa del desempeño de los clasificadores

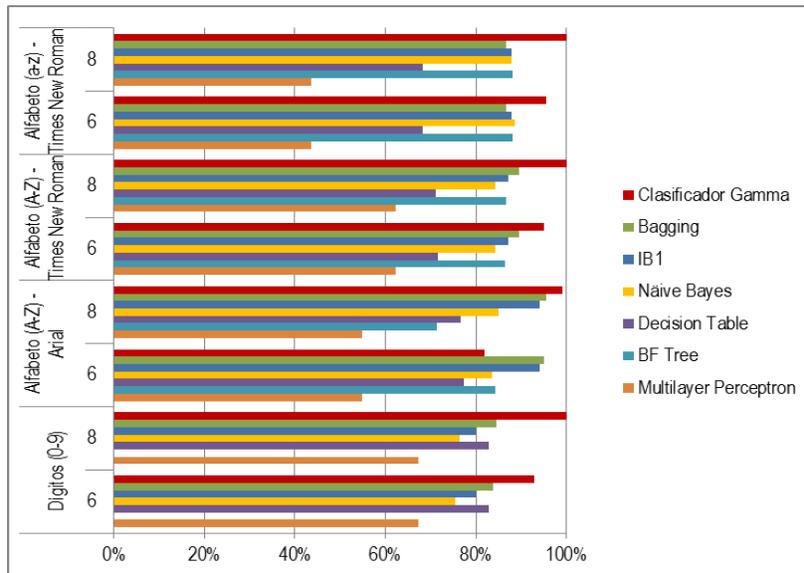


Figura 6.2. Gráfica vertical del desempeño de los clasificadores

El clasificador Gamma mostró tener el mejor desempeño, demostrando ser bastante apto para trabajar con imágenes binarias. De los experimentos realizados se observa que su desempeño no sólo es competitivo, sino que incluso sus porcentajes de clasificación resultan ser mayores que los obtenidos por los clasificadores probados, como se muestra en las gráficas 6.1 y 6.2.

## 6.2 Factor de olvido

El factor de olvido es un fenómeno que se presenta cuando un clasificador es incapaz de clasificar un patrón a pesar de que éste haya formado parte de su conjunto de

entrenamiento [81]. Al ser ésta una característica cuya presencia es deseable que sea mínima o nula, en general, esta presente en prácticamente todos los clasificadores. A diferencia de la gran mayoría de clasificadores, el clasificador Gamma, al pertenecer al enfoque asociativo del reconocimiento de patrones, es capaz de recuperar o clasificar todos los elementos del conjunto fundamental, esto es, reconocer de forma correcta cada patrón con el que se ha aprendido.

El procedimiento utilizado es el siguiente:

1. Configurar un conjunto de  $m$  clases de imágenes, cada una con un número de elementos formados por transformaciones geométricas de la imagen original o representante de la clase. Particularmente con las que los momentos de Hu presentan invariancia, esto es, rotación, traslación y cambio de escala.
2. Obtener los siete momentos invariantes de cada una de las imágenes que conforman el conjunto fundamental, esto es, para cada una de las imágenes miembro de cada una de las clases en el conjunto.
3. Escoger una combinación de momentos con la cual se desea probar al clasificador. Específicamente la conformada por los momentos  $\Phi_2$  y  $\Phi_4$  de las expresiones (3.15).
4. Evaluar el desempeño del clasificador utilizando el algoritmo 5.1 para el clasificador Gamma simplificado. El criterio para evaluar el desempeño fue el de utilizar los conjuntos fundamentales de los conjuntos de datos de la tabla 6.1 como datos de entrenamiento para posteriormente tomar uno a uno los patrones para su clasificación. El proceso se repitió para todos los elementos en los conjunto de datos y el resultado final se obtuvo de aplicar la media aritmética a todos los resultados de las  $p$  iteraciones de cada conjunto de datos. Donde  $p$  corresponde a la cardinalidad de cada conjunto.
5. Para la misma combinación, probar con cada uno de los métodos de clasificación de la tabla 6.2 utilizando la plataforma WEKA para comparar el resultado obtenido.
6. Asentar en una tabla comparativa los resultados agregando los comentarios y conclusiones correspondientes.

En la tabla 6.4 se ilustran los porcentajes de clasificación obtenidos.

CLASIFICADOR	Porcentaje de Clasificación (Factor de olvido)								PROM	
	Digitos (0-9)		Alfabeto (A-Z) - Arial		Alfabeto (A-Z) - Times New Roman		Alfabeto (a-z) - Times New Roman			
	6	8	6	8	6	8	6	8		
<i>IB1</i>	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100%
<b>Clasificador Gamma</b>	94.55%	100.00%	94.06%	99.65%	98.95%	100.00%	98.95%	100.00%	100.00%	98%
<i>Bagging</i>	97.27%	97.27%	97.20%	97.20%	94.76%	94.76%	96.15%	96.15%	96.15%	96%
<i>Decision Table</i>	88.18%	88.18%	86.36%	90.56%	88.81%	88.81%	87.06%	87.06%	87.06%	88%
<i>BF Tree</i>	70.00%	70.00%	90.21%	90.91%	95.80%	96.15%	95.10%	94.76%	94.76%	88%
<i>Näive Bayes</i>	76.36%	84.55%	88.11%	89.16%	87.76%	89.51%	91.96%	92.31%	92.31%	87%
<i>Multilayer Perceptron</i>	76.36%	76.36%	58.04%	58.04%	65.38%	65.38%	50.70%	50.70%	50.70%	63%

Tabla 6.4. Porcentajes de recuperación de los clasificadores analizados para estimar el factor de olvido

Se puede notar que, y de forma semejante a los resultados del apartado anterior, el incremento en el número de decimales de 6 a 8 para truncar los valores representa una

variación significativa en los porcentajes de clasificación mostrados. De los cuales, particularmente el clasificador Gamma resulta beneficiado reflejándose en el aumento de sus porcentajes emparejando al IB1 (Gráficas 6.3 y 6.4).

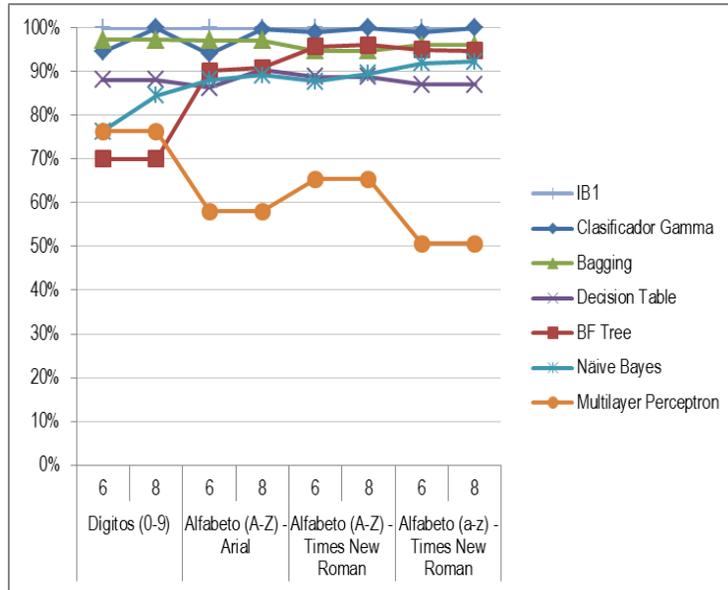


Figura 6.3. Gráfica comparativa del desempeño de los clasificadores para estimar el factor de olvido

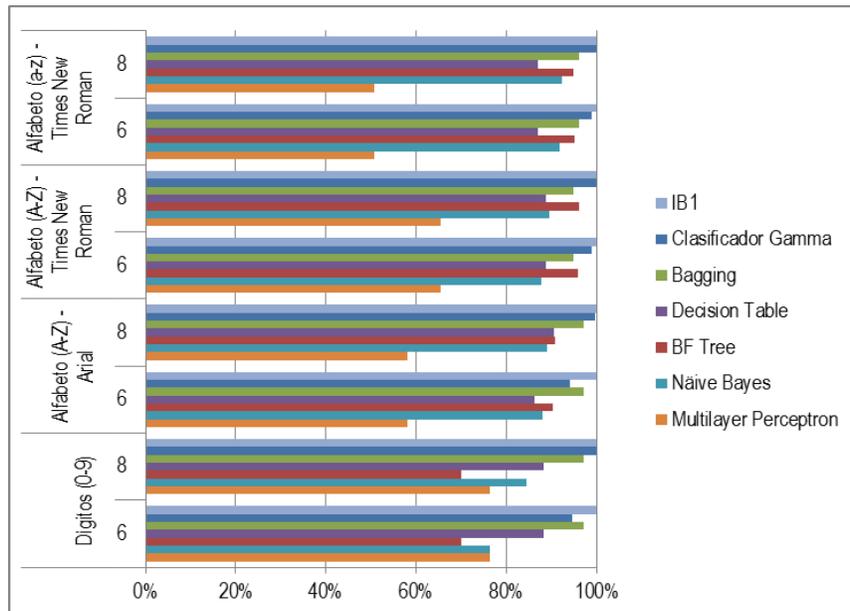


Figura 6.4. Gráfica vertical del desempeño de los clasificadores para estimar el factor de olvido

El clasificador Gamma junto con el IB1 demuestran ser robustos al factor de olvido, al observarse el impacto de este reflejado en el desempeño del resto de los clasificadores, los cuales consiguen tener muy bajos porcentajes de recuperación considerando que el conjunto de prueba es el mismo con el que se entrenó.

### 6.3 Rendimiento del algoritmo propuesto

Dentro de las contribuciones del presente trabajo esta la de proponer un nuevo algoritmo para el clasificador Gamma. De ahí que este apartado tenga como objetivo presentar los resultados en lo referente a los tiempos registrados de los algoritmos del clasificador Gamma original y el Gamma simplificado. Los algoritmos son sometidos a dos conjuntos de experimentos: el primero consiste en comparar los tiempos de ambos algoritmos incrementando el número de decimales a truncar las componentes, esto para 4, 6 y 8 decimales; y el segundo, consiste en mostrar los tiempos de los algoritmos al incrementar el número de rasgos a considerar de los patrones.

Los experimentos son realizados utilizando el conjunto de prueba conformado por el alfabeto en minúsculas con tipo de letra Times New Roman descrito en la tabla 6.1 y utilizado en los apartados anteriores. Asimismo, cada experimento tiene dos casos: en el primer caso se evalúa un patrón conocido, y para el segundo, uno desconocido, es decir, que se toma el tiempo de recuperación al clasificar un patrón tomado del conjunto fundamental y para el segundo se toma un patrón que no pertenece al conjunto, respectivamente.

En las tablas 6.5 y 6.6 se muestran los tiempos de las ejecuciones truncando las componentes de los patrones a 4, 6 y 8 decimales, utilizando un patrón conocido y uno desconocido respectivamente.

ALGORITMO	Tiempo de ejecución por número de decimales (ms)		
	4	6	8
Gamma Original	0.016	0.67	66.924
Gamma Simplificado	0	0	0.016

Tabla 6.5. Registro de tiempos truncando decimales para un patrón conocido

ALGORITMO	Tiempo de ejecución por número de decimales (ms)		
	4	6	8
Gamma Original	0.015	0.702	69.811
Gamma Simplificado	0	0.031	2.745

Tabla 6.6. Registro de tiempos truncando decimales para un patrón desconocido

En las gráficas 6.7 y 6.8 correspondientes a las tablas 6.5 y 6.6 respectivamente, se observa que el algoritmo propuesto obtiene tiempos menores que los del algoritmo original, además de notarse que los tiempos obtenidos con el algoritmo original aumentan en orden mayor en comparación con el algoritmo simplificado, cuando el número de decimales truncados es incrementado, particularmente de 6 a 8 decimales.

Cabe hacer mención que la precisión del tiempo tomado es de milisegundos. De ahí, que los tiempos indicados como 0, en las tabla de resultados, sugieran tiempos menores a un milisegundo.

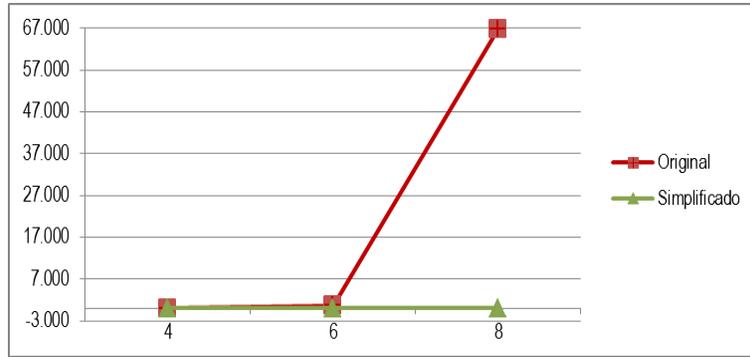


Figura 6.5. Gráfica de los tiempos de con distinto nivel de truncamiento para un patrón conocido.

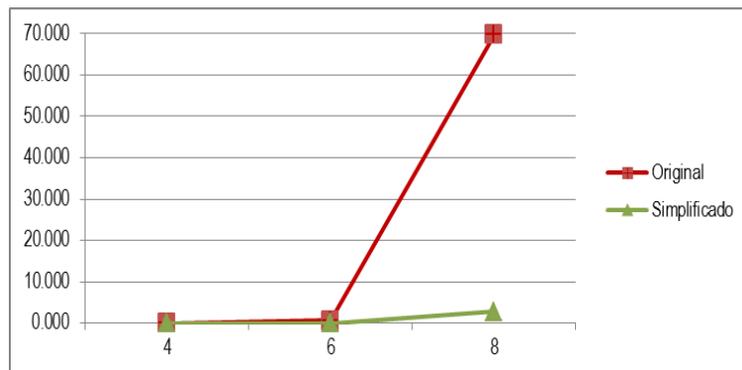


Figura 6.6. Gráfica de los tiempos de con distinto nivel de truncamiento para un patrón desconocido

En las tablas 6.7 y 6.8 se muestran los tiempos de las ejecuciones para distinto número de rasgos de los patrones, utilizando un patrón conocido y uno desconocido respectivamente. Para los experimentos realizados se utilizó truncamiento de 6 decimales para cada una de las componentes.

Tiempo de ejecución por número de rasgos (ms)				
ALGORITMO	$\{\Phi_1\}$	$\{\Phi_1, \Phi_2\}$	$\{\Phi_1, \Phi_2, \Phi_3, \Phi_4\}$	$\{\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5, \Phi_6, \Phi_7\}$
Gamma Original	0.67	1.342	1.888	1.981
<b>Gamma Simplificado</b>	0	0	0	0

Tabla 6.7. Registro de tiempos con distinto número de rasgos para un patrón conocido

Tiempo de ejecución por número de rasgos (ms)				
ALGORITMO	$\{\Phi_i\}$	$\{\Phi_1, \Phi_2\}$	$\{\Phi_1, \Phi_2, \Phi_3, \Phi_4\}$	$\{\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5, \Phi_6, \Phi_7\}$
Gamma Original	0.702	1.404	1.857	2.012
<b>Gamma Simplificado</b>	0.031	0	0	0

Tabla 6.8. Registro de tiempos con distinto número de rasgos para un patrón desconocido

En las graficas 6.9 y 6.10 correspondientes a las tablas 6.7 y 6.8 respectivamente, se observa que el algoritmo propuesto obtiene tiempos más bajos que los del algoritmo original. Se puede notar que los tiempos del algoritmo original, sin bien no crecen de

manera abrupta como en los mostrados en las tablas 6.5 y 6.6 del experimento anterior, estos incrementan de igual forma que con el número de rasgos considerados de las componentes mientras que para el algoritmo simplificado propuesto estos se mantienen sin gran cambio.

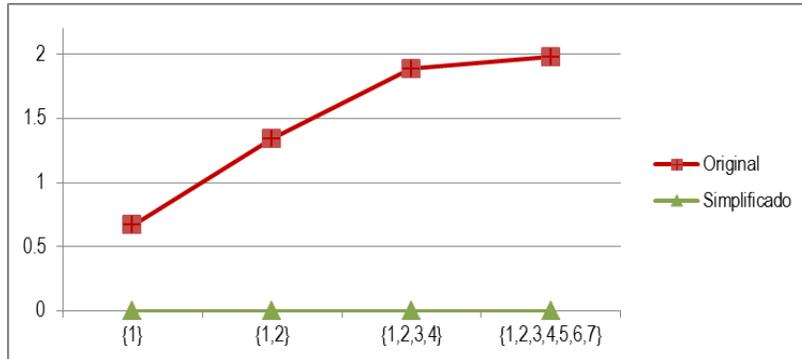


Figura 6.7. Gráfica de los tiempos de ejecución considerando distinto número de rasgos para un patrón conocido

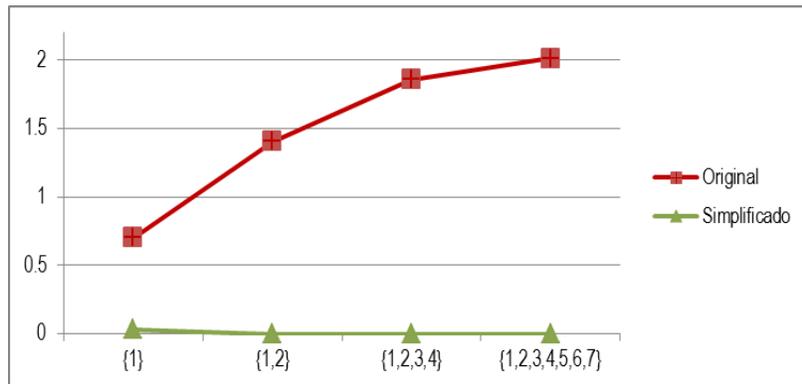


Figura 6.8. Gráfica de los tiempos de ejecución considerando distinto número de rasgos para un patrón desconocido

## 6.4 Discusión

Los casos expuestos de clasificación mediante el uso de momentos invariantes de una imagen, permitió probar el desempeño del nuevo algoritmo del clasificador Gamma. Dado que con el uso de los momentos invariantes se obtuvieron valores de componentes muy variables, y en general su escala y traslación resultaron en números enteros relativamente grandes, se pudo explotar correctamente la mejora propuesta. Se validó que efectivamente el rendimiento en términos del costo computacional mejoró, reflejándose en la reducción de tiempos de clasificación de los patrones presentados.

Asimismo, se puede notar en los experimentos que no necesariamente gran cantidad de momentos son requeridos para poder obtener alta discriminación entre objetos, siendo que únicamente se utilizaron dos rasgos para caracterizar a los objetos en los experimentos.

En resumen, el clasificador Gamma fue el que obtuvo más altos porcentajes de clasificación, incluso superior en comparación a los de los demás clasificadores, mostrando de esta forma un nivel de desempeño altamente competitivo (Gráficas 6.1 y 6.2).

En cuanto a los tiempos registrados, de forma general nos permiten observar que el número de rasgos de las componentes no influye de forma drástica en los tiempos de clasificación usando el algoritmo original del clasificador (Tablas 6.7 y 6.8) a diferencia de los tiempos obtenidos cambiando el número de decimales a truncar de la componentes (Tablas 6.5 y 6.6), los cuales obligadamente son altos dada la cantidad de recursos computacionales utilizados para contener y procesar las componentes.

# Capítulo 7

## ***Conclusiones y trabajo futuro***

En el presente trabajo se mostró experimentalmente el desempeño del clasificador Gamma en tareas que involucran imágenes binarias mediante un estudio comparativo con algunos de los modelos más representativos entre los distintos enfoques de clasificación en el área de reconocimiento de patrones: *Naïve Bayes*, *Multilayer Perceptron*, *IB1*, *Bagging*, *Decision Table* y *BF-Tree*. En dicho estudio se probó, primeramente, el desempeño del clasificador Gamma en comparación con los algoritmos mencionados, mostrando su alta competitividad. Posteriormente, se mostró el impacto del factor de olvido en los clasificadores con una comparación del desempeño de estos al ser entrenados con todo el conjunto fundamental de cada conjunto para ser probados al recuperar los elementos aprendidos. Donde el clasificador Gamma, al igual que el IB1, mostró gran robustez, por encima de los otros enfoques. Por último, y dado que se presentó adicionalmente un nuevo algoritmo para el clasificador Gamma, se realizó una comparación de los tiempos de procesamiento para los conjuntos de prueba utilizando el algoritmo original y el simplificado, donde se pudo observar la mejora en cuanto al rendimiento y consumo de recursos computacionales en términos de reducción de tiempo.

En general, el clasificador Gamma fue probado experimentalmente en tareas de clasificación de imágenes mediante los momentos invariantes como caracterizadores. A continuación se exponen las conclusiones obtenidas.

1. De los resultados obtenidos de la verificación experimental, se pudo observar que el desempeño del clasificador Gamma es altamente competitivo al trabajar en tareas de clasificación de imágenes binarias, y como se mostró, puede ser superior a otros clasificadores.
2. Adicionalmente, y como consecuencia del punto anterior, los descriptores basados en momentos invariantes mostraron tener alta discriminación al trabajar con el clasificador Gamma, dados los altos porcentajes de clasificación en las pruebas realizadas.
3. Se ha desarrollado un nuevo algoritmo para el clasificador Gamma el cual reduce el número de operaciones y de ahí los recursos computacionales requeridos para su cálculo, el cual ha sido probado experimentalmente con los momentos invariantes como caracterizadores. La eliminación de los pasos para codificar en código Johnson-Möbius modificado y sus subsecuentes operaciones, permiten al clasificador Gamma trabajar directamente con números enteros. Esto reduce de forma considerable el tiempo de ejecución, en comparación con el algoritmo original.

## **7.1 Trabajo futuro**

En el presente trabajo se utilizaron imágenes binarias y los momentos invariantes de Hu, los cuales son invariantes ante cambios de escala, traslación y rotación. Por ello, en lo referente a los descriptores, el clasificador Gamma podría ser probado empleando otros invariantes: invariantes al cambio de contraste e iluminación, invariantes ante la presencia de ruido, entre otras. Incluso ser probado con alguna de las técnicas de momentos que han mostrado tener mejor desempeño; por ejemplo, los polinomios de Zernike, Tchebichev, y recientemente Krawtchouk.

En lo referente a imágenes, el clasificador Gamma podría ser probado utilizando conjuntos de imágenes en escala de grises, con siluetas, con imágenes reales de algún otro dominio o incluso con imágenes en tercera dimensión.

## Referencias

- [1] R.C. Gonzalez, R.E. Woods. "*Digital Image Processing*". Second edition. Prentice-Hall, (2002).
- [2] Duda, R. O., Hart, P. E., Stork, D. G. "*Pattern Classification*". John Wiley & Sons, USA (2001).
- [3] J. Wood, "*Invariant Pattern Recognition: A Review*", Pattern Recognition Society, 29, pp. 1-17 (1995).
- [4] R. J. Prokop and A. P. Reeves, "*A Survey of Moment-based techniques for Unoccluded Object Representation and Recognition*", Graphical Models Image Process, 54, pp. 438-460, (1981).
- [5] J. Flusser and T. Suk, "*Pattern Recognition by Affine Moment Invariants*", Pattern Recognition, 26, pp. 167-174, (1993).
- [6] Pejnovic, P. Stojiljkovic, Z. "*Object recognition by invariants*". Pattern Recognition. Vol.II. Pattern Recognition Methodology and Systems, Proceedings, 11th IAPR International Conference on. Conference Publications, pp. 434–437, (1992).
- [7] M.-K. Hu, "*Pattern Recognition by Moments Invariants*", IRE Transactions On Information Theory, 49, pp. 1428, (1961).
- [8] S. Reddi, "*Radial and Angular moment Invariants for Image Identification*", IEEE Trans. Pattern Anal. Mach. Intell., 3, pp. 240-242, (1981).
- [9] Y. Abu-Mustafa and D. Psaltis, "*Image Normalization by Complex Moments*", IEEE Trans. Pattern Anal. Machine Intell, PAMI-7, pp. 46-55, (1984).
- [10] M. Teague, "*Image Analysis Via the General Theory of Moments*", J. Opt. Soc. Am., 70, pp. 920-930, (1980).
- [11] M.-K. Hu, "*Visual Pattern Recognition by Moments Invariants*", IRE Transactions On Information Theory, 8, (1962).
- [12] F. L. Alt, "*Digital Pattern Recognition by Moments*", J. Assoc. Computing Machinery, 9, pp. 240-258, (1962).
- [13] S. Maitra, "*Moment Invariants*", Proc IEEE, 67, pp. 697-699, (1979).
- [14] Y. Abu-Mustafa and D. Psaltis, "*Recognitive Aspects of the Moment Invariants*", IEEE Trans. Pattern Anal. Mach. Intell., 6, pp. 5598-5706, (1984).
- [15] S. Dudani, K. Breeding and R. McGhee, "*Aircraft Identification by Moment Invariants*", IEEE Trans. Comput., 26, pp. 39-45, (1977).
- [16] Ju Cheng Yang, Dong Sun Park. "*A fingerprint verification algorithm using tessellated invariant moment features*". Neurocomputing for Vision Research. Advances in Blind Signal Processing. Volume 71, Issues 10–12, pp. 1939–1946, June 2008.

- [17] Jin Soo Noh, Kang Hyeon Rhee, "*Palprint Identification Algorithm Using Hu Invariant Moments And Otsu Binarization*", Computer and Information Science, 2005. Fourth Annual ACIS International Conference on, pp. 94-99 (2005).
- [18] C.-W Chong, P. Raveendran and .R. Mukundan, "*Translation Invariants of Zernike Moments*", Pattern Recognition, 36, pp. 1765-1773, (2003).
- [19] C.-W Chong, P. Raveendran and .R. Mukundan, "*Translation and Scale Invariants of Legendre Moments*", Pattern Recognition, 37, pp. 119-129, (2004).
- [20] F. Mindru, T. Tuytelaars, L. V. Gool and T. Moons, "*Moment Invariants for Recognition Under Changing Viewpoint and Illumination*", Computer Vision and Image Unders. Volume 94, pp. 3–27, (2004).
- [21] Tomáš Suk, Jan Flusser. "*Combined blur and affine moment invariants and their use in pattern recognition*". Pattern Recognition. Volume 36, Issue 12, pp. 2895–290, December 2003.
- [22] Tomas Suk, Jan Flusser. "*Projective Moment Invariants*". IEEE Transactions on pattern analysis and machine intelligence, Volume 26, Issue 10, pp. 1364-1367, October, 2004.
- [23] Y. Li, "*Reforming the Theory of Invariant Moments for Pattern Recognition*", Pattern Recognition, 25, pp.723-730, (1992).
- [24] C. Teh and R. Chin, "*On Image Analysis by the Methods of Moments*", IEEE Trans. Pattern. Anal. Mach. Intell., 10, pp. 496-512, (1988).
- [25] J. Flusser. "*On the independence of rotation moment invariants*". Pattern Recognition, Volume 33, Issue 9, pp. 1405–1410, September 2000.
- [26] Wai-hong Wong, Wan-Chi- Siu, Kin-Man Lam. "*Generation of moment invariants and their uses for character recognition*". Pattern Recognition Letters, 16 (1995), pp. 115-123.
- [27] S. O. Belkasim, M. Shridhar and Ahmadi, "*Pattern Recognition With Moment Invariants: A Comparative Study and New Results*", Pattern recognition, 12, pp. 1117-1138, (1991).
- [28] Sangame S.K.1, Ramteke R.J.1, Rajkumar Benne. "*Recognition of isolated handwritten Kannada vowels*". Advances in Computational Research, ISSN: 0975–3273, Volume 1, Issue 2, 2009, pp-52-55.
- [29] Hongqing Zhua, Huazhong Shua, Ting Xiaa, Limin Luoa, Jean Louis Coatrieux. "*Translation and scale invariants of Tchebichef moments*". Pattern Recognition, 40, pp. 2530 – 2542, (2007).
- [30] F.A. Sadjadi, E.L. Hall. "*Three-dimensional moment invariants*", IEEE Trans. Pattern Anal. Machine Intell. 2, 127–136, (1980).
- [31] Y. Zhu, L.C. De Silva, C.C. Ko. "*Using moment invariants and HMM in facial expression recognition*". Pattern Recognition Letters. Volume 23, Issues 1–3, pp. 83–91, January 2002.
- [32] George A. Papakostas, E. G. Karakasis, Dimitris E. Koulouriotis, "*Efficient and Accurate Computation of Geometric Moments on Gray-Scale Images*". Pattern Recognition, 41, pp. 1895-1904, (2008).

- [33] R. Mukundan, S. H. Ong and P. A. Lee, "*Image Analysis by Tchebichev Moments*", IEEE Transactions on Image Processing, 10, pp. 1357-1369, (2001).
- [34] R. Mukundan, "*A New Class of Rotational Invariants Using Discrete Orthogonal Moments*", Signal and Image Processing, pp. 23-25, (2004).
- [35] Paramesran R., Seng-Huat Ong, "*Image Analysis by Krawtchouk Moments*", IEEE Trans on Image Processing, Volum 12, Issue 11, pp. 1367-1377 (2003).
- [36] S.X. Liao, M. Pawlak. "*On image analysis by moments*". IEEE Transactions on Pattern Analysis and Machine Intelligence, 18, pp. 254–266, (1996).
- [37] Jan Flusser. "*Moment Invariants in Image Analysis*". World Academy of Science, Engineering and Technology, 11, (2005).
- [38] Dong Hu, Hua Li, "*Geometric Moment Invariants*", Pattern Recognition. Vol. 41, pp. 240-249, (2008).
- [39] Qiguang Miao, Juan Liua, Weisheng Lib, Junjie Shia, Yiding Wang. "Three novel invariant moments based on radon and polar harmonic transforms". Optics Communications. Volume 285, Issue 6, pp. 1044–1048, March 2012.
- [40] Zhu Mingfa, Santosh Hasani, Surendra Bhattarai, Harpreet Singh, "*Pattern recognition with moment invariants on a machine vision system*". Pattern Recognition Letters, Volume 9, Issue 3, pp. 175-180, (1989).
- [41] Leida Lia, Shushang Lia, Ajith Abrahamb, Jeng-Shyang Panc. "*Geometrically invariant image watermarking using Polar Harmonic Transforms*". Information Sciences. Volume 199, pp. 1–19, September 2012.
- [42] Zhihu Huang, Jinsong Leng. "*Analysis of Hu's Moment Invariants on Image Scaling and Rotation*". Computer Engineering and Technology (ICET), 2nd International Conference on. Conference Publications. Volume 7, pp. 476 -480, April 2010.
- [43] I. Balsev, "*Noise Tolerance of Moment Invariants in Pattern Recognition*", Pattern Recognition Letters, August (1998).
- [44] Hongqing Zhua, Huazhong Shua, Jian Zhoua, Limin Luo, J.L. Coatrieuxb. "Image analysis by discrete orthogonal dual Hahn moments". Pattern Recognition Letters. Volume 28, Issue 13, 1, pp. 1688–1704 (2007).
- [45] Qian Li, Hongqing Zhu, Qian Liu. "Image recognition by combined affine and blur Tchebichef moment invariants". Image and Signal Processing (CISP), 2011 4th International Congress on. October 2011. Volume 3 Pp. 1517 – 1521. Conference Publications.
- [46] G.A. Papakostas, E.G.Karakasis, D.E.Koulouriotis. "Novel moment invariants for improved classification performance in computer vision applications". Pattern Recognition. Volume 4, Issue 3, Pp. 58-68, (2010).
- [47] R. Muralidharan, C. Chandrasekar. "Object Recognition using SVM-KNN based on Geometric Moment Invariant". International Journal of Computer Trends and Technology- July to Aug. Issue 2011.

- [48] Chao Kan, Mandyam D. Srinath. "Invariant character recognition with Zernike and orthogonal Fourier–Mellin moments". *Pattern Recognition*. Volume 35, Issue 1, January 2002, pp. 143–154. Shape representation and similarity for image databases.
- [49] Mohamed Rizon, Haniza Yazid, Puteh Saad, Ali Yeon Md Shakaff, Abdul Rahman Saad, Mohd Rozailan Mamat, Sazali Yaacob, Hazri Desa y M. Karthigayan, "Object Detection Using Geometric Invariant Moment", *American Journal of Applied Sciences*, 2 (6):, pp. 1876-1878, (2006).
- [50] Lei Zhang, Jiexin Pu and Jia Yu. "Object Recognition Based on Modified Invariant Moments". *Proceedings of the 2009 IEEE, International Conference on Mechatronics and Automation*. August 9 - 12, Changchun, China.
- [51] Fix, E., Hodges, J.L. "*Discriminatory analysis, nonparametric discrimination: Consistency properties*". Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.
- [52] E. Fix and J. L. Hodges, "*Discriminatory analysis, small sample performance*". Tech. Rep. Tex. Project 21-49-004, Rept. 11, USAF School of Aviation Medicine, Randolph Field, 1952.
- [53] Aha, D., and D. Kibler. "*Instance-based learning algorithms*", *Machine Learning*, Volume 6, pp. 37-66, (1991).
- [54] T. M. Cover and P. E. Hart, "*Nearest neighbor pattern classification*". *IEEE Trans. Information Theory*, vol. 13, no. 1, pp. 21–27 (1967).
- [55] D. W. Aha. "*Incremental, instance-based learning of independent and graded concept descriptions*". In *Sixth International Workshop on Machine Learning*, Detroit, MI, Morgan Kaufmann (1989).
- [56] Desathary, B., "*Nearest neighbor (NN) norms: NN pattern classification techniques*". IEEE Press, 1990.
- [57] Aha, D. "Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms". *International Journal of Man-Machine Studies*, 36: 267–287. 1992.
- [58] Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, C.J. "*Classification and regression trees*". Belmont, CA: Wadsworth International Group, (1984).
- [59] D.J. Hand and K. Yu. "*Idiot's Bayes - not so stupid after all?*" *International Statistical Review*, 69(3):385-398, (2001).
- [60] George H. John and Pat Langley. "*Estimating Continuous Distributions in Bayesian Classifiers*". *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. pp. 338-345. Morgan Kaufmann, San Mateo, (1995).
- [61] P. Domingos and M. Pazzani. "*On the optimality of the simple Bayesian classifier under zero-one loss*". *Machine Learning*, 29:103-130, (1997).
- [62] Ian H. Witten, Eibe Frank, "*Data mining: practical machine learning tools and techniques with Java implementations*", Morgan Kaufmann Publishers Inc., San Francisco, CA, (2000).
- [63] Breiman, L. "*Bagging predictors*". *Machine Learning*, Vol. 24, 123–140, (1992).

- [64] Tuv, E. "Ensemble learning. In *Feature Extraction*". 207, 187–204 of *Studies in Fuzziness and Soft Computing*. Springer Berlin, (2006).
- [65] Opitz, D., and Maclin, R. "Popular ensemble methods: An empirical study". *Journal of Artificial Intelligence Research* 11: 169-198, (1999).
- [66] Guohua Liang, Xingquan Zhu, and Chengqi Zhang. "An Empirical Study of Bagging Predictors for Different Learning Algorithms". *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*. The Centre for Quantum Computation & Intelligent Systems, FEIT, University of Technology, Sydney, NSW (2007).
- [67] Xiaoyuan Su Taghi M. Khoshgoftarr Xingquan Zhu. "VoB Predictors: Voting on Bagging Classifications". *Computer Science and Engineering*, Florida Atlantic University.
- [68] Pawlak, Z. "Rough set theory and its applications to data analysis". *Cibernetics and systems*. Vol. 29, Issue: 7, pp.661-688, (1998).
- [69] Marek, W. Pawlak, Z. "One dimension learning". *Fundamenta informaticae*, vol.8, No. 1. pp 83-88, (1985).
- [70] Slezak, D. Zierko, W. "Attribute reduction in the bayesian version of variable precision rough set model". *Proc. Of RSKD2003*. Elsevier, ENTCS 82/4, (2004).
- [71] Yao, Y.Y. Wong, S.K.M."A decision theoretic framework for approximating concepts". *Intl. Journal of Man-Machine studies*". Vol. 37, pp. 793-809. (1992).
- [72] Ziarko, W. "Variable precision rough sets model". *Journal of Computer and Systems Sciences*". Vol. 46, No. 1. pp. 39-59. (1993).
- [73] Ziarko, W."Acquisition of hierarchy-structured probabilistic decision tables and rules from data". *Proc. Of IEEE Intl. Conference on Fuzzy Systems*". Pp.593-603. (2001).
- [74] W. Ziarko, "On Learnability of Decision Tables". In *Proceedings of Rough Sets and Current Trends in Computing*, pp. 394-401, (2004).
- [75] Venkatadri M., Lokanatha C. Reedy. "A comparative study on decision tree classification algorithms in data minning". *International journal of computer applications in engineering, technology and sciences (IJ-CA-ETS)*. Vol. 2, Issue 2. pp. 24-29, (2010).
- [76] M. J. Berry and G. S. Linoff. "Mastering data mining", New York: John Wiley & Sons, (2000).
- [77] K-M. Osei-Bryson, "Post-pruning in decision tree induction using multiple performance measures", *Computers & Operations Research*, No. 34: pp. 3331-3345, (2007).
- [78] Haijian Shi. "Best-first decision tree learning". Hamilton, NZ, (2007).
- [79] Jerome Friedman, Trevor Hastie, Robert Tibshirani. "Additive logistic regression: A statistical view of boosting". *Annals of statistics*. 28(2), pp. 337-407, (2000).
- [80] Heidelberg.Witten, I. H., and Frank, E. "Data Mining: Practical Machine Learning Tools and Techniques", Second Edition. Morgan Kaufmann, (2005).

- [81] I. López Yáñez. "Clasificador Automático de Alto Desempeño". Tesis de Maestría. Instituto Politécnico Nacional. Centro de Investigación en Computación. México. pp.18-41, (2007).
- [82] I. López Yáñez. "Teoría y Aplicaciones del Clasificador Asociativo Gamma". Tesis de Doctorado. Instituto Politécnico Nacional. Centro de Investigación en Computación. México. (2011).
- [83] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten. "*The WEKA Data Mining Software: An Update*". SIGKDD Explorations. Vol. 11, pp. 10-18, (2009).
- [84] G. Holmes, A. Donkin, and I.H. Witten. "*WEKA: A Machine Learning Workbench*". In Proc Second Australia and New Zealand Conference on Intelligent Information Systems, Brisbane, Australia, (1994).