

**Universidad Central “Marta Abreu” de Las Villas**

**Facultad de Ingeniería Eléctrica**

**Centro de Estudio de Electrónica y Tecnologías de la  
Información**



## **TRABAJO DE DIPLOMA**

**Clasificación de células en la prueba de  
Papanicolaou empleando técnicas de visión  
computacional**

**Autor: José Daniel López Cabrera**

**Tutor: Dr.C. Juan Valentín Lorenzo Ginori**

**Santa Clara**

**2013**

**"Año 55 del triunfo de la Revolución"**

**Universidad Central “Marta Abreu” de Las Villas**

**Facultad de Ingeniería Eléctrica**

**Departamento de Automática y Sistemas Computacionales**



## **TRABAJO DE DIPLOMA**

### **Clasificación de células en la prueba de Papanicolaou empleando técnicas de visión computacional**

**Autor: José Daniel López Cabrera**

jdaniel@uclv.edu.cu

**Tutor: Dr.C. Juan Valentín Lorenzo Ginori**

Profesor Titular Consultante, Investigador Titular

juanl@uclv.edu.cu

**Santa Clara**

**2013**

**"Año 55 del triunfo de la Revolución"**



Hago constar que el presente trabajo de diploma fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de estudios de la especialidad de Ingeniería Biomédica, autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

---

Firma del Autor

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

---

Firma del Autor

---

Firma del Jefe de Departamento  
donde se defiende el trabajo

---

Firma del Responsable de  
Información Científico-Técnica

## **PENSAMIENTO**

*"La investigación de las enfermedades ha avanzado tanto que cada vez es más  
difícil encontrar a alguien que esté completamente sano."*

***ALDOUS HUXLEY***

## **DEDICATORIA**

A mis padres que tanto se han sacrificado por mí. El apoyo de ustedes es mi mayor recompensa.

## **AGRADECIMIENTOS**

A mis padres por hacer posible que yo esté aquí, por su ayuda en todo momento, por ser mi ejemplo a seguir cada día, por su cariño y amistad.

A mi novia Betsy y su familia por su cariño, y por acogerme como parte de la casta.

A mi tutor Juan V. Lorenzo Ginori por ayudarme a llevar a cabo esta investigación.

A todos los profesores que aportaron en mi formación profesional.

A mis compañeros en todo este camino maravilloso que ha sido la universidad.

A todos los que me han apoyado y me han acompañado en estos cinco años.

## **TAREA TÉCNICA**

- Realizar una búsqueda y revisión crítica de los materiales encontrados en la literatura científica sobre el tema. Determinar la posibilidad de clasificar imágenes de citología cervical utilizando rasgos basados en los núcleos celulares.
- Implementar en Matlab y probar algoritmos para la clasificación de células de citología cervical utilizando conjuntos de rasgos basados en la forma y en la textura de las imágenes de los núcleos celulares.
- Evaluar el comportamiento de los algoritmos aplicados en términos de índices de efectividad tales como la predictividad negativa y la medida F.

---

Firma del Autor

---

Firma del Tutor

## **RESUMEN**

En este trabajo se realizó una investigación sobre la clasificación de las células cervicales mediante técnicas de procesamiento digital de imágenes. Se utilizaron tres conjuntos de rasgos, relacionados solamente con los núcleos celulares e ignorando los citoplasmas. El primer conjunto contiene rasgos relacionados con la forma, mientras que el segundo conjunto está basado en textura, el tercer conjunto es la unión de los dos anteriores. En todos los casos se utilizó el Análisis de Componentes Principales como método para reducir la dimensionalidad de los datos. Se trabajó con las imágenes de la base de datos Herlev. Se empleó la programación en Matlab para utilizar los algoritmos clasificadores: clasificador lineal, de Mahalanobis, k vecinos más cercanos y Máquina de Soporte Vectorial. Posteriormente se analizaron los resultados de todos los experimentos realizados en cuanto a: la predictividad negativa y la medida F. Se realizó un análisis estadístico para determinar cuál de los cuatro clasificadores resultó ser el más eficiente y cuáles de los conjuntos de rasgos utilizados fue el mejor.



## TABLA DE CONTENIDOS

PENSAMIENTO .....	i
DEDICATORIA .....	ii
AGRADECIMIENTOS .....	iii
TAREA TÉCNICA.....	iv
RESUMEN .....	v
INTRODUCCIÓN .....	1
Organización del informe .....	4
CAPÍTULO 1. INTRODUCCIÓN A LA CLASIFICACIÓN DE IMÁGENES DE CITOLOGÍA CERVICAL.....	6
1.1 Generalidades sobre el uso de la visión computacional en el reconocimiento y clasificación de imágenes .....	6
1.2 La prueba de Papanicolaou y el cáncer cérvico-uterino.....	7
1.3 El problema del reconocimiento y clasificación de imágenes de citología cervical. Aplicaciones en el control de la calidad en la prueba de Papanicolaou mediante sistemas automatizados .....	8
1.4 Rasgos utilizados para la clasificación de imágenes.....	10
1.5 Generalidades sobre algoritmos clasificadores .....	11
1.5.1 El clasificador lineal .....	12
1.5.2 El clasificador basado en la distancia de Mahalanobis.....	12

1.5.3	El método de los k vecinos más cercanos (knn) .....	12
1.5.4	Máquinas de Soporte Vectorial (SVM) .....	13
1.6	Métodos de evaluación del desempeño de los algoritmos clasificadores .....	14
CAPÍTULO 2. MATERIALES Y MÉTODOS.....		17
2.1	Funciones Matlab para la implementación clasificadores .....	17
2.2	Funciones Matlab para la evaluación de clasificadores .....	19
2.3	La base de datos Herlev .....	20
2.3.1	Uso de matrices de rasgos en la implementación de clasificadores .....	21
2.1.1	Estructura de datos utilizada .....	23
2.4	Reducción de la dimensionalidad de los datos.....	24
2.5	Prueba de hipótesis estadísticas en la evaluación de clasificadores.....	24
2.6	Programas para la clasificación y evaluación .....	25
CAPÍTULO 3. RESULTADOS Y DISCUSIÓN.....		27
3.1	Índices de desempeño obtenidos .....	27
3.2	Resultados del clasificador de Mahalanobis .....	28
3.2.1	Análisis estadístico para el clasificador de Mahalanobis usando la medida F ... .....	29
3.2.2	Análisis estadístico para el clasificador de Mahalanobis usando la predictividad negativa.....	31
3.3	Resultados para las clases 1 y 7 .....	33
3.4	Resultados para la todas las clases .....	33
3.5	Selección del mejor clasificador .....	34
3.5.1	Análisis estadístico para determinar el mejor clasificador usando la medida F . .....	35

3.5.2	Análisis estadístico para determinar el mejor clasificador usando la predictividad negativa.....	37
3.6	Conclusiones del capítulo .....	39
CONCLUSIONES Y RECOMENDACIONES .....		40
	Conclusiones .....	40
	Recomendaciones .....	40
REFERENCIAS BIBLIOGRÁFICAS .....		42
ANEXOS .....		45
Anexo I	Resultados del clasificador lineal .....	45
Anexo II	Resultados del clasificador SVM con kernel rbf $\sigma=2$ .....	48
Anexo III	Resultados del clasificador knn.....	51
Anexo IV	Índices de desempeño en las 20 corridas para todos los clasificadores .....	54
Anexo V	Programa Matlab para la clasificación .....	69

## INTRODUCCIÓN

La prueba de Papanicolaou, conocida también como prueba citológica, es un método de diagnóstico de cáncer cérvico-uterino. Además ofrece un panorama sobre el estado general del útero, el nivel hormonal y la existencia de otras afecciones. Si el resultado de la prueba de Papanicolaou es negativo, la cervix es normal. Pero cuando el resultado es positivo, entonces significa la presencia de células anómalas, no necesariamente que exista un cáncer. Ante esto, es preciso realizar un urgente reconocimiento con otros exámenes: colposcopia o biopsia.

A partir de la experiencia en la aplicación de esta prueba se ha podido determinar que existen diferentes factores que afectan la calidad de los resultados. Principalmente se encuentran los errores en la toma de las muestras, en su procesamiento y en su lectura e interpretación. Como consecuencia de los diferentes errores mencionados, aparece una cierta tasa de falsos negativos en el resultado final. Esto implica, la posibilidad que un carcinoma curable pueda convertirse en cáncer invasivo, sin que la paciente sea sometida a tratamiento alguno. En ello radica la importancia de este trabajo pues con él se pretende realizar una contribución al estudio y desarrollo en nuestro medio de algoritmos y programas de procesamiento digital de imágenes orientados al control de calidad en la prueba de Papanicolaou, específicamente en lo que se refiere a la evaluación de la efectividad de diferentes alternativas de clasificadores, a partir de rasgos significativos en los núcleos de las células en cuestión.

La aparición de falsos negativos motiva la necesidad de asegurar la calidad en la toma y procesamiento de las muestras, además de disponer de métodos efectivos de revisión de las pruebas. Para controlar este efecto adverso, se desarrollan en la actualidad métodos que utilizan el procesamiento digital de imágenes y la visión computacional. Estos podrán ser

empleados en el control de calidad de los resultados de la prueba de Papanicolaou, pues permitirían aprovechar las ventajas que brindan los sistemas automáticos, como es la objetividad que aporta trabajar con imágenes digitales (matrices de datos numéricos).

Este control se logra a partir de la detección de posibles falsos negativos, de modo que estos puedan ser reanalizados por el personal médico especializado y de esta forma se consiga mejorar la sensibilidad de la prueba y la predictividad negativa. El proceso general de análisis automatizado de las imágenes en la prueba de Papanicolaou, consta de tres etapas: la adquisición de la imagen digital, la segmentación o extracción de las regiones de interés (núcleo, citoplasma) y finalmente la clasificación de las células.

Este trabajo tiene entre sus fundamentos lo recogido en los lineamientos del VI Congreso del Partido Comunista de Cuba, en los que se establece que:

131. Sostener y desarrollar los resultados alcanzados en el campo de la biotecnología, la producción médico-farmacéutica, la industria del software y el proceso de informatización de la sociedad, las ciencias básicas, las ciencias naturales, los estudios y el empleo de las fuentes de energía renovables, las tecnologías sociales y educativas, la transferencia tecnológica industrial, la producción de equipos de tecnología avanzada, la nanotecnología y los servicios científicos y tecnológicos de alto valor agregado.

132. Perfeccionar las condiciones organizativas, jurídicas e institucionales para establecer tipos de organización económica que garanticen la combinación de investigación científica e innovación tecnológica, desarrollo rápido y eficaz de nuevos productos y servicios, su producción eficiente con estándares de calidad apropiados y la gestión comercializadora interna y exportadora, que se revierta en un aporte a la sociedad y en estimular la reproducción del ciclo. Extender estos conceptos a la actividad científica de las universidades.

El problema científico de este análisis es el siguiente: ¿Cuáles son los rasgos y clasificadores más efectivos en los núcleos de las células de la prueba de Papanicolaou para disminuir los falsos negativos?

Para responder lo antes expuesto se proponen los siguientes objetivos:

**Objetivo General**

- Desarrollar programas en Matlab para garantizar una mayor calidad en los resultados de la prueba de Papanicolaou.

**Objetivos Específicos**

- Seleccionar los rasgos más efectivos para realizar una certera evaluación a los núcleos de las células.
- Implementar programas en Matlab para realizar la clasificación de las células en normales o anómalas en las imágenes de citología cervical en la prueba de Papanicolaou.
- Comprobar el desempeño de los programas de clasificación realizando un análisis contrastivo con los resultados obtenidos por analistas humanos.

Para lograr la solución del problema expuesto se plantearon las siguientes tareas técnicas:

- Realizar una búsqueda y revisión crítica de los materiales encontrados en la literatura científica sobre el tema. Determinar la posibilidad de clasificar imágenes de citología cervical utilizando rasgos basados en los núcleos celulares.
- Implementar en Matlab y probar algoritmos para la clasificación de células de citología cervical utilizando conjuntos de rasgos basados en la forma y en la textura de las imágenes de los núcleos celulares.
- Evaluar el comportamiento de los algoritmos aplicados en términos de índices de efectividad tales como la predictividad negativa y la medida F.

Para la realización de este proyecto investigativo contamos con la posibilidad de trabajar con un software para el procesamiento digital de imágenes: Matlab (laboratorio de matrices) y sus diferentes cajas de herramientas. Además en el plan de estudio de la carrera de Ingeniería Biomédica se imparten asignaturas vinculadas a este tema, como por ejemplo Procesamiento Digital de imágenes (PDI), PDI avanzado y otras que facilitarán el desempeño satisfactorio del proyecto. Para la investigación contamos además con la base de datos Herlev, que cuenta con un total de 917 imágenes de citología cervical. Estas están

segmentadas y evaluadas por un grupo de especialistas. Dichas imágenes serán la base para el entrenamiento de nuestros clasificadores.

Los resultados de esta investigación son valiosos en el área de la imaginología médica, relacionadas con el análisis de imágenes de microscopía celular y poseen una aplicación práctica para los especialistas e investigadores relacionados con la prueba de Papanicolaou. El método propuesto brinda un soporte y una plataforma de automatización de los procesos que puede contribuir a una buena toma de decisiones por parte de los profesionales de la medicina. De esta manera se persigue desarrollar un método de clasificación automatizado, utilizando tecnologías de visión computacional, con menor costo, mayor eficiencia en términos de tiempo y una disminución de los márgenes de errores cometidos. Por todo lo anterior, se considera que los resultados de este trabajo pueden tener una importancia significativa para el Sistema Nacional de Salud.

### **Organización del informe**

Estructuralmente el trabajo está conformado por: introducción, capitulario, conclusiones y recomendaciones, referencias bibliográficas y anexos.

En la introducción queda definida la importancia, actualidad y necesidad del tema que se aborda y se dejan explícitos los objetivos y tareas a realizar.

El capítulo I se titula: Introducción a la clasificación de imágenes de citología cervical. En este capítulo se abordan las generalidades sobre el uso de visión computacional en el reconocimiento y clasificación de imágenes, las aplicaciones en el control de la calidad mediante sistemas automatizados en la prueba de Papanicolaou. Se realiza una introducción sobre la prueba de Papanicolaou y el cáncer cérvico-uterino. Además se mencionan los algoritmos clasificadores y los rasgos utilizados en el estudio. En el capítulo II se proponen posibles soluciones a los problemas encontrados para lograr una mejor clasificación utilizando un conjunto de rasgos basados en la forma y la textura de los núcleos celulares de las imágenes de la base de datos Herlev. En el capítulo III se presentan y discuten resultados luego de obtener los índices de desempeño para los clasificadores y los conjuntos de rasgos utilizados en la clasificación. Se realizó un análisis estadístico para determinar el clasificador de mejor comportamiento.

En las conclusiones se exponen los principales resultados en concordancia con los objetivos planteados. Las recomendaciones dan indicaciones para trabajos futuros sobre el tema. La bibliografía incluye referencias utilizadas en el trabajo. Los anexos reúnen información sobre los estudios realizados que no quedaron plasmados en el cuerpo del trabajo.



## **CAPÍTULO 1. INTRODUCCIÓN A LA CLASIFICACIÓN DE IMÁGENES DE CITOLOGÍA CERVICAL**

Esta sección resume la importancia de la realización periódica de la prueba de Papanicolaou para el diagnóstico y prevención de cualquier tipo de anomalía existente en la zona del cuello uterino o vaginal. Para el control de calidad, se propone el desarrollo de equipos automatizados con los cuales se reduce notablemente los debidos errores de lectura e interpretación. A pesar que la incorporación de estos equipos automáticos disminuye la aparición de falsos negativos, se pretende orientar el trabajo hacia otro método que evalúa la calidad de la prueba mediante el procesamiento digital de imágenes, utilizando la segmentación de los núcleos, la extracción de características de los mismos y el análisis de los datos resultantes.

### **1.1 Generalidades sobre el uso de la visión computacional en el reconocimiento y clasificación de imágenes**

La visión artificial, también conocida como visión computacional, es un sub-campo de la inteligencia artificial. El objetivo de esta técnica es lograr que el ordenador sea capaz de entender una acción o interpretar las características de una imagen. Dentro de las actividades que se persiguen en este campo se encuentran la detección, segmentación, localización y reconocimiento de ciertos objetos en imágenes; búsqueda de imágenes digitales por su contenido y seguimiento de un objeto en una secuencia de imágenes. Estos objetivos se consiguen por medio de diferentes funciones de reconocimiento de patrones.

La utilización de esta técnica puede proporcionar ventajas a la hora de realizar una valoración. Se establece un proceder constante, sin dependencia del estado anímico, cansancio o subjetividad del observador humano. De este modo una misma imagen podrá

ser analizada siempre de igual forma y no existirá la variabilidad que caracteriza el análisis por diferentes observadores. En la imagen digital están registrados detalles imperceptibles para el ojo humano por lo que se logrará en muchos casos un análisis más preciso y con mayor información.

Con métodos basados en la visión computacional, queda resuelta la problemática de la subjetividad a la hora de emitir un criterio sobre si una imagen contiene o no algún tipo de anomalía. Asimismo, se alivia la carga de trabajo de los analistas humanos, los que tienen que observar diariamente gran cantidad de imágenes de citología cervical, debido al elevado número de féminas que se realizan la prueba de Papanicolaou. Esto determina que exista entonces la posibilidad de que ciertas células sean clasificadas como negativas cuando en realidad no lo son, estos son los llamados falsos negativos.

## **1.2 La prueba de Papanicolaou y el cáncer cérvico-uterino.**

La prueba de Papanicolaou conocida normalmente por prueba citológica, debe su nombre a Georgios Papanicolaou, médico griego que fue pionero en citología y quien desarrolló una prueba para la detección temprana de cáncer en el cuello uterino [1]. El mismo es un examen donde se toman muestras de células epiteliales en la zona de transición del cuello uterino en busca de anomalías celulares que orienten la presencia de una posible neoplasia de cuello uterino. Dicha prueba es un análisis de gran importancia para la detección y diagnóstico precoz de cáncer cervical.

El cáncer cervical o de cérvix uterino es en la actualidad el segundo tipo de cáncer más común en mujeres. En la región de América Latina y el Caribe sus tasas de incidencia se hallan dentro de las más altas del planeta. En Cuba, ha venido ocupando entre los lugares tres y cuatro en orden de ocurrencia y desgraciadamente del tercero al quinto en mortalidad. La aparición de este tipo de cáncer en las diferentes áreas geográficas está estrechamente relacionada con las condiciones climatológicas y el estilo de vida que lleven las pacientes. En este último punto inciden factores tales como: relaciones sexuales tempranas, higiene inadecuada, niveles socioeconómico y educacional bajos, uso prolongado de anticonceptivos orales, dieta, tabaquismo y promiscuidad en ambos sexos.

El cáncer cervical casi nunca muestra indicios de existencia, por tanto la realización sistemática de la prueba de Papanicolaou juega un papel fundamental para su temprana

detección. No obstante, la aparición de una hemorragia o flujo anormal de la vagina requiere de investigaciones médicas. Estos síntomas pueden ser ocasionados por diversos problemas pero pudiera existir la posibilidad de un cáncer cervical. Si se realiza el diagnóstico a tiempo de lesiones premalignas y del cáncer cérvico-uterino, no se verá afectada la salud y la capacidad de trabajo de la mujer. Esto puede lograrse con el perfeccionamiento de los métodos de diagnóstico de este tipo de neoplasia.

El hecho de que un gran número de pruebas sean clasificadas como negativas repetidamente, puede llevar al analista a emitir un falso criterio o clasificación. En un frotis típico se pueden encontrar hasta 300000 células, lo que limita la productividad del observador a no más de unas 60-80 citologías por día de trabajo.

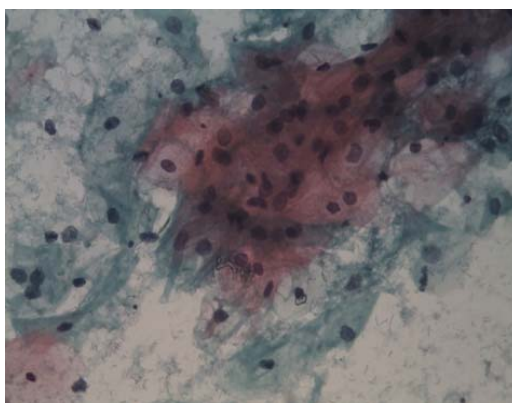


Figura 1. Imágenes de un campo, correspondiente a un frotis en la prueba de Papanicolaou convencional.

### **1.3 El problema del reconocimiento y clasificación de imágenes de citología cervical. Aplicaciones en el control de la calidad en la prueba de Papanicolaou mediante sistemas automatizados**

En la actualidad el reconocimiento y clasificación de imágenes presentan dificultades. Estas vienen dadas porque las imágenes a analizar, normalmente pueden estar contaminadas con ruido. Los objetos que en ellas están contenidas poseen variabilidad en cuanto a la posición, orientación y rotación. Además se ven afectadas por la variabilidad de la iluminación lo cual incide directamente sobre el contraste, la textura, y la uniformidad de los colores en dichas imágenes. Por ser muy variables los patrones biológicos, no existe un único método para realizar una certera evaluación de las imágenes de microscopía celular.

El proceso general de análisis automatizado de la imagen en la prueba de Papanicolaou (tanto la convencional como la que emplea citología de base líquida), consta de tres etapas: la adquisición de la imagen digital, la segmentación o extracción de las regiones de interés (núcleo, citoplasma) a partir de esta y finalmente la clasificación de las células.

Dentro de los mayores desafíos existentes para determinar un algoritmo eficiente para la segmentación en imágenes de citología cervical está el solapamiento de las células [2]. En la literatura actual podemos encontrar una gran variedad de técnicas de segmentación. La transformada watershed es una de las técnicas más poderosas en cuanto a métodos de segmentación [3]. Se basa en el análisis de una representación tridimensional de la superficie, por ejemplo la intensidad en una imagen [4]. La implementación de algoritmos basados en la transformada watershed se realiza mediante operaciones de procesamiento morfológico de imágenes que pueden ser implementados empleando diferentes algoritmos tales como: inmersión o distancias topográficas. El procesamiento morfológico es además una herramienta comúnmente usada para extraer y procesar componentes de la imagen que resultan útiles en la representación y descripción de las formas [5].

La combinación del método de pseudo coloreado con el algoritmo de crecimiento de regiones a partir de semillas da como resultado la extracción de rasgos de pseudo-color. Con este algoritmo se logra extraer los tamaños del núcleo y el citoplasma así como los valores de grises en células cervicales [6].

Los métodos que utilizan el procesamiento digital de imágenes y la visión computacional han comenzado a desempeñar en los últimos años un importante papel. Giménez en [7] realiza una valoración del estado de estas técnicas, desde su aparición a finales del siglo XX hasta el año 2003. El *Autopap*, es un sistema automatizado que realiza el control de calidad en la prueba de Papanicolaou. Dicho sistema realiza el análisis de las muestras en dos modalidades: el cribado primario y el secundario. Cuando se realiza el cribado primario se persigue diagnosticar altos porcentajes de casos como negativos, reduciendo considerablemente la carga de trabajo de los analistas humanos. En una valoración de esta técnica se muestra que aunque su valor predictivo negativo es muy alto, todavía existe una pequeña tasa de falsos negativos [8]. Cuando la técnica se utiliza para el cribado secundario, *Autopap* se emplea para el control de calidad del trabajo realizado por los analistas humanos. Esto consiste en evaluar los casos que fueron seleccionados como

negativos y determinar de ellos cual pudiera ser positivo, disminuyendo así los falsos negativos. La prueba de Papanicolaou ha evolucionado durante los primeros años del siglo XXI, hacia la adopción de la citología de base líquida como el método estándar para preparar las muestras [9]. Cibas en [10] analiza los nuevos sistemas hacia los cuales evolucionó el *Autopap*, por ejemplo el *BD FocalPoint<sup>TM</sup> Slide Profiler*, que posee una mayor productividad y efectividad en el cribado primario. Austin en [11] realiza un estudio del empleo de este sistema y de su efectividad para las células glandulares atípicas. El *ThinPrep Imaging System* es otro sistema que está destinado a desarrollar la productividad del analista humano hasta más de 300 láminas por día. Posee un incremento significativo de la sensibilidad pero al mismo tiempo cierta reducción de la especificidad. Finalmente, se cita el *FocalPoint GS Imaging System*, similar en su concepción al *ThinPrep*, en este caso con una productividad de 170 láminas por jornada de 8 horas. Todos estos sistemas se encuentran instalados en varios laboratorios de los EE. UU.

#### **1.4 Rasgos utilizados para la clasificación de imágenes**

En las imágenes médicas, las regiones que son objeto de investigación para la detección de anomalías se caracterizan por sus irregularidades. En general se hacen difíciles de clasificar, por lo que es necesario utilizar para estos propósitos una gran diversidad de rasgos que puedan diferenciar unas regiones de otras. La calidad del resultado obtenido en la clasificación depende generalmente de la capacidad que tengan los rasgos extraídos de caracterizar la zona de interés en el estudio.

La representación de una región tiene dos opciones: representar la región en términos de sus características externas por sus fronteras, el denominado criterio de forma, o representar la región en términos de sus características internas utilizando los píxeles incluidos en su interior llamado criterio de textura [4]. La elección del esquema de representación debe ser determinada por el tipo de tarea a resolver de tal modo que los datos resulten más ventajosos. En algunas circunstancias pudiera ser necesario usar ambos tipos de representación, lo que coincide con Srisuk [12]. Los rasgos seleccionados como descriptores de la región deben ser tan insensibles como sea posible, a las variaciones en tamaño, traslación y rotación. La mayor parte de los descriptores satisfacen una o más de estas propiedades.

En el estudio de regiones se han empleado descriptores que genéricamente se agrupan como simples, topológicos y de textura, aunque en la práctica es común usar combinadamente tanto los descriptores de frontera como los de región [13]. Dentro de las orientaciones significativas en la descripción de una región es cuantificar su contenido de textura. El descriptor de textura suministra medidas de las propiedades de las imágenes tales como suavidad, rugosidad y regularidad. Los descriptores simples son área, perímetro y compacidad. El área de una zona se define como el número de píxeles en la región. El perímetro de la región sería la longitud de sus fronteras. La medida de la compacidad en una región se calcula como  $(\text{perímetro})^2/\text{área}$ . La compacidad es una medida adimensional y por tanto es insensible a los cambios uniformes de escala. Con la excepción de los errores cometidos en la rotación digital, la compacidad también es insensible a la rotación. Otras medidas simples usadas como descriptores de una región incluyen la media y la mediana de los niveles de gris, el mínimo y el máximo de los niveles de gris y el número de píxeles con valores por encima y por debajo de la media.

### **1.5 Generalidades sobre algoritmos clasificadores**

Un clasificador es un mecanismo que ofrece una clase etiquetada como salida a partir de un conjunto de características tomadas como entradas. Existen diferentes tipos de técnicas para el aprendizaje. Una de ellas es el aprendizaje supervisado, el cual consiste en aprender una función, a partir de ejemplos etiquetados anteriormente, que establezca una correspondencia entre las entradas y las salidas deseadas del sistema. En esta técnica se basan los cuatro clasificadores utilizados en este trabajo. Existe además el aprendizaje no supervisado, en este caso el algoritmo aprende a partir de patrones de entradas para los que no se especifiquen los valores de sus salidas. La principal desventaja que se puede apuntar de esta técnica es la toma de decisiones a la hora de escoger un patrón entre todos los proporcionados.

Ambas técnicas han sido utilizadas para la clasificación de imágenes en citología cervical. Dentro de los clasificadores basados en aprendizaje supervisado más citados en la bibliografía se encuentran las redes neuronales artificiales (RNA) [14], [15] las máquinas de soporte vectorial (SVM) [16], [17], [18], k vecinos más cercanos (knn) [19], así como el

clasificador Bayesiano [20]. Con respecto a los que se basan en aprendizaje no supervisado, tenemos *Spectral Clustering* [21] y *fuzzy C-means* [22].

### 1.5.1 El clasificador lineal

El clasificador lineal toma una decisión de clasificación, utilizando el valor de una combinación lineal de sus características. Las características de un objeto son típicamente presentadas en un vector llamado vector de rasgos o características. Si la entrada del clasificador es un vector de características reales  $\vec{x}$ , entonces el resultado de salida es:

$$y = f\left(\sum_j w_j x_j\right) \quad (1)$$

donde  $w_j$  es un vector real de pesos y  $f$  es una función que convierte el producto punto a punto de los dos vectores en la salida deseada. El vector de pesos  $w_j$  aprende de un conjunto de muestras de entrenamiento. Normalmente  $f$  es una función simple que mapea todos los valores por encima de un cierto umbral a la primera clase y el resto a la segunda clase. Los clasificadores lineales se utilizan regularmente en ambientes donde la rapidez de la clasificación es importante, ya que este clasificador es de los rápidos.

### 1.5.2 El clasificador basado en la distancia de Mahalanobis

Formalmente, la distancia de Mahalanobis entre dos variables aleatorias con la misma distribución de probabilidad  $\vec{x}$  y  $\vec{y}$  con matriz de covarianza  $\Sigma$  se define como:

$$dm(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y})} \quad (2)$$

Su utilidad radica en que es una forma de determinar la similitud entre dos variables aleatorias multidimensionales. Se diferencia de la distancia euclidiana en que tiene en cuenta la correlación entre las variables aleatorias. Se utiliza cuando las variables no son independientes. Es invariante por transformaciones lineales no singulares, en particular por cambios de escala.

### 1.5.3 El método de los k vecinos más cercanos (knn)

El método de los k vecinos más cercanos (knn, *k nearest neighbors*), es un procedimiento de clasificación no paramétrico. En esta técnica se estima el valor de la función de densidad

de probabilidad o directamente la probabilidad a posteriori de que un elemento  $x$  pertenezca a la clase  $C_j$  a partir de la información proporcionada por un conjunto de prototipos. Es un algoritmo de clasificación supervisado. En el proceso de aprendizaje no se hace ninguna suposición acerca de la distribución de las variables predictoras. La fase de entrenamiento del algoritmo radica en almacenar los vectores característicos y las etiquetas de las clases de los ejemplos de entrenamiento. En la fase de clasificación, la evaluación de un ejemplo del que no se conoce su clase, es representada por un vector en el espacio de rasgos. Se calcula la distancia entre los vectores almacenados y el nuevo vector y se seleccionan los  $k$  ejemplos más cercanos. El nuevo ejemplo es clasificado con la clase que más se repite en los vectores seleccionados.

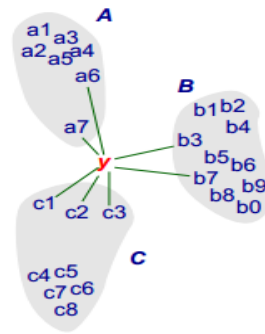


Figura 2. Regla de los knn con  $k=7$ . La variable 'y' es clasificada dentro del conjunto C por mayoría de vecinos.

Este método considera que los vecinos más cercanos nos dan la mejor clasificación y esto se hace utilizando todos los atributos. El problema de dicha suposición es que es posible que se tengan muchos atributos irrelevantes que dominen sobre la clasificación: dos atributos relevantes perderían peso entre otros veinte irrelevantes. Para corregir el posible sesgo se puede asignar un peso a las distancias de cada atributo, dándole así mayor importancia a los atributos más relevantes. Otra posibilidad consiste en tratar de determinar o ajustar los pesos con ejemplos conocidos de entrenamiento. Finalmente, antes de asignar pesos es recomendable identificar y eliminar los atributos que se consideran irrelevantes.

#### 1.5.4 Máquinas de Soporte Vectorial (SVM)

Las máquinas de soporte vectorial constituyen una poderosa y robusta herramienta destinadas a labores de clasificación [18], [23]. Este algoritmo se basa en construir hiperplanos en un espacio multidimensional para separar las diferentes clases. Para la



construcción del hiperplano óptimo se emplea un algoritmo de entrenamiento iterativo, el cual se utiliza para minimizar una función de error. De acuerdo a la función de error, existen dos tipos de vectores de soporte para clasificación: tipo 1 (*C-SVM*) y tipo 2 (*nu-SVM*).

Dentro de los núcleos que se pueden emplear están, entre otros, las funciones polinómicas, funciones gaussianas de base radial o sigmoideas. En esta investigación se utiliza un núcleo gaussiano de base radial (rbf) de orden 2. Se determinó usar esta variante luego de hacer varias pruebas informales y determinar que era la que mejores resultados mostraba. Esta función del núcleo es la que utiliza el algoritmo clasificador para separar los datos y crear el hiperplano entre ellos. Los datos son evaluados y de acuerdo a su resultado es que se asignan a su clase de pertenencia.

### 1.6 Métodos de evaluación del desempeño de los algoritmos clasificadores

Luego de disponer de un estándar dorado adecuado, en este caso la base de datos Herlev [24], la que contiene las regiones anómalas para el problema dado, es posible realizar la evaluación de los clasificadores. La valoración de un algoritmo de clasificación requiere la definición de medidas de similitud para comparar los datos de referencia con los resultados obtenidos en el estudio.

Tabla 1. Matriz de confusión

		Clasificación realizada por analistas humanos	
		Anómalas $X_i = 1$	Normales $X_i = 0$
Clasificadas por el Algoritmo	Anómalas $y_i = 1 (+)$	<i>VP</i>	<i>FP</i>
	Normales $y_i = 0 (-)$	<i>FN</i>	<i>VN</i>

En la Tabla 1 se muestran las siguientes definiciones:

*VP* = cantidad de ocurrencias de  $y_i = 1 \mid x_i = 1$  (células anómalas detectadas como anómalas), conocidas como verdaderos positivos.

$FP$  = cantidad de ocurrencias de  $y_i = 1 \mid x_i = 0$  (células normales detectadas como anómalas), conocidas como falsos positivos.

$FN$  = cantidad de ocurrencias de  $y_i = 0 \mid x_i = 1$  (anomalías detectadas como no-anomalías), conocidas como falsos negativos.

$VN$  = cantidad de ocurrencias de  $y_i = 0 \mid x_i = 0$  (no-anomalías detectadas como no-anomalías), conocidas como verdaderos negativos.

A partir de estas definiciones se pueden calcular diferentes indicadores de desempeño como la sensibilidad, la especificidad, la predictividad positiva, la predictividad negativa y la tasa de clasificación correcta.

La sensibilidad ( $Se$ ) o tasa de detección correcta es la probabilidad de que las anomalías sean detectadas correctamente. Se define como

$$Se = p(Y = 1 \mid X = 1) = 1 - p(Y = 0 \mid X = 1) = \frac{VP}{VP + FN} = 1 - \frac{FN}{VP + FN}. \quad (3)$$

La especificidad ( $S_p$ ) es la probabilidad de detectar correctamente las regiones normales, se calcula como:

$$S_p = p(Y = 0 \mid X = 0) = 1 - p(Y = 1 \mid X = 0) = \frac{VN}{FP + VN} = 1 - \frac{FP}{FP + VN}. \quad (4)$$

La predictividad positiva ( $P_p$ ) es la probabilidad de que los positivos hayan sido correctamente detectados. Se calcula como:

$$P_p = p(X = 1 \mid Y = 1) = 1 - p(X = 0 \mid Y = 1) = \frac{VP}{VP + FP} = 1 - \frac{FP}{VP + FP}. \quad (5)$$

La predictividad negativa ( $P_n$ ) o tasa bayesiana negativa: es la probabilidad de que los negativos hayan sido correctamente detectados. Se calcula como:

$$P_n = p(X = 0 \mid Y = 0) = 1 - p(X = 1 \mid Y = 0) = \frac{VN}{FN + VN} = 1 - \frac{FN}{FN + VN}. \quad (6)$$

La tasa de clasificación correcta ( $Cr$ ) es la proporción del número total de casos que fueron clasificados correctamente.

$$Cr = p(X=Y) = 1 - p(X \neq Y) = \frac{VP+VN}{VP+FP+FN+VN} = 1 - \frac{FP+FN}{VP+FP+FN+VN}. \quad (7)$$

La tasa de error ( $Er$ ) es el número total de casos que fueron clasificados incorrectamente.

$$Er = \frac{FP+FN}{VP+FP+FN+VN} = 1 - \frac{VP+VN}{VP+FP+FN+VN}. \quad (8)$$

La medida F ( $F_{measure}$ ) combina sensibilidad y especificidad. Es la media armónica de la sensibilidad y la especificidad.

$$D_C = F_{measure} = \frac{2S_e S_p}{S_e + S_p} = \frac{2VP}{2VP+FP+FN}. \quad (9)$$

Se observa que para mejorar la clasificación debe ocurrir una disminución de los falsos negativos ( $Fn$ ), con lo cual la sensibilidad y la predictividad negativa aumentarán, al igual que si disminuyen los falsos positivos ( $Fp$ ), la especificidad y la predictividad positiva también aumentarían. Reducir la tasa de falsos negativos aumenta la seguridad del diagnóstico para el paciente, mientras que disminuir la tasa de falsos positivos tiene un efecto económico pues evita seguir tratando a pacientes que en realidad están sanas.

## CAPÍTULO 2. MATERIALES Y MÉTODOS

En la investigación se trabaja con las imágenes de la base de datos Herlev [24] como referencia para los métodos de clasificación. Dichas imágenes han sido previamente ordenadas en 7 clases sobre la base de una clasificación realizada por expertos humanos. A partir de esto se definen rasgos representativos de los núcleos de las células para poder clasificarlas como normales (sin anomalía) o anómalas (con algún tipo de desviación de la normalidad). Para implementar el algoritmo clasificador se toma como base el clasificador lineal. En la implementación de los algoritmos clasificadores (clasificador lineal, de Mahalanobis, knn y SVM) se emplea la programación en Matlab, utilizando herramientas como las funciones *crossvalind*, *classify*, *svmtrain*, *svmclassify*, *knnclassify*, *classperf* y algunas utilidades de la clase de datos *cell array*.

### 2.1 Funciones Matlab para la implementación clasificadores

Para la clasificación de las células en el estudio se utilizan 5 clasificadores. El clasificador lineal y el de la distancia de Mahalanobis, ambos fueron implementados mediante la función *classify* de Matlab. El tercer y cuarto clasificador emplean una máquina de soporte vectorial (SVM), mediante las funciones *svmtrain* y *svmclassify* de Matlab. El primero de estos dos utiliza kernel lineal y el otro kernel gaussiano de base radial. El quinto clasificador está basado en el método de vecinos más cercanos, el cual fue implementado mediante la función *knnclassify*, con 3 vecinos cercanos.

La función *classify* tiene como argumento los siguientes parámetros [25]:

*class = classify (sample,training,group)*

Esta función permite clasificar cada una de las filas de los datos de muestra (*sample*) en uno de los grupos usados en el entrenamiento (*training*). Los datos de muestra y de

entrenamiento deben tener el mismo número de columnas para poder realizar la clasificación. Otra variante de utilizar *classify* es especificando el tipo de función discriminante que se desea, de no hacerse este paso, Matlab toma por defecto el clasificador lineal.

```
class = classify(sample,training,group,'type')
```

'type' : permite especificar el tipo de clasificador. Estas pueden ser:

- 'linear': acomoda una densidad normal multivariada para cada grupo con una estimación mancomunada de covarianza.
- diaglinear: similar al lineal, pero con un estimado diagonal de la matriz de covarianza.
- quadratic: las densidades de normalidad multivariadas con estimados de covarianza estratificados por grupos.
- diagquadratic: similar al cuadrático, pero con una estimación diagonal de la matriz de covarianza.
- mahalanobis: usa la distancia de Mahalanobis con estimaciones estratificadas de covarianza.

La función *svmtrain* tiene como argumentos los siguientes parámetros [25]:

```
svmstruct = svmtrain(training,group)
```

*training*: matriz de datos de entrenamiento

*group*: cell array que contiene en sus filas la clasificación de la matriz de datos entrenamiento, debe tener la misma cantidad de filas que *training*. La variable *group* es un *cell array* de cadenas de caracteres de la misma longitud que el conjunto de entrenamiento que está definido para los dos grupos. En general *group* puede ser un vector columna, un arreglo de caracteres o un *cell array* de cadenas. En este trabajo se emplea un grupo de tipo numérico (matriz de rasgos) y el otro grupo de tipo *cell array* (normales y anómalas).

La tarea de *svmtrain* es entrenar el clasificador SVM tomando datos etiquetados que se introducen mediante la variable *group*. Por consiguiente, *svmstruct* contiene información acerca del clasificador entrenado, incluyendo los vectores de soporte. Estos datos son usados luego por *svmclassify* para realizar la clasificación de las células.

Esta función se puede utilizar también de la siguiente forma:

```
Svmtrain(...,'Kernel_Function',Kfun)
```

Aquí se permite especificar la función *Kfun*, que usa el mapa de los datos de entrenamiento dentro del espacio *Kernel*. *Kfun* puede ser una de las siguientes cadenas:

- '*linear*' Kernel Lineal
- '*quadratic*' Kernel Cuadrático
- '*polynomial*' Kernel Polinomial (por defecto orden 3)
- '*rbf*' Kernel gaussiano con función de base radial (por defecto factor de escala  $\sigma = 1$ )
- '*mlp*' Kernel de perceptron de capa múltiple (por defecto escala y parámetro del bias [1,-1])

En nuestro estudio utilizamos kernel rbf con sigma igual dos ( $\sigma=2$ ).

La función *svmclassify* tiene como argumento los siguientes parámetros [25]:

*Group=svmclassify(svmstruct,sample)*

*svmstruct*: creada usando *svmtrain*

*sample*: matriz de datos de prueba (a ser clasificado)

Su objetivo es clasificar utilizando la máquina de soporte vectorial. Clasifica cada fila de los datos de prueba con la información del clasificador de máquina de soporte vectorial mediante la estructura *svmstruct* creada usando *svmtrain*. La matriz de datos de prueba debe tener el mismo número de columnas como los datos que se usaron en el entrenamiento del clasificador en *svmtrain*. En *Group* queda el resultado de la clasificación realizada por la SVM.

## 2.2 Funciones Matlab para la evaluación de clasificadores

Para la evaluación de la calidad de estos clasificadores se trabaja con las funciones *crossvalind* y *classperf* de Matlab.

La función *classperf* provee una interfaz para seguir el desempeño durante la validación de los clasificadores (lineal, Mahalanobis, knn, SVM) y acumula los resultados de estos. A continuación se ilustran los parámetros de su argumento [25]:

*cp = classperf(..., 'Positive', PositiveValue, 'Negative', NegativeValue)*

*classperf(cp,class,test)*

*cp* es el objeto de evaluación del clasificador, que primeramente se inicializa.

*PositiveValue* es un vector numérico o un arreglo de nombres que indican los casos positivos en los objetos que se estén analizando (clases 4, 5, 6, 7). En *NegativeValue* se encuentran los casos que no tienen anomalía (clases 1, 2, 3).

La evaluación del desempeño los clasificadores se determina en esta investigación a través de los siguientes parámetros:

- Tasa de error
- Tasa de clasificación correcta
- Sensibilidad
- Especificidad
- Predictividad positiva
- Predictividad negativa
- Medida F

La función *crossvalind* tiene como argumento los siguientes parámetros [25]:

$\text{Indices} = \text{crossvalind}('K\text{fold}', N, K)$

La validación cruzada divide las  $N$  muestras a clasificar en  $K$  grupos ordenados aleatoriamente. Se utilizan  $K-1$  grupos para entrenamiento y el grupo restante para la evaluación. Con este proceso siempre se toma un subconjunto diferente para la evaluación hasta clasificar todos los  $K$ . Esta técnica es simple y garantiza conocer el grado de generalización adquirido por los clasificadores. En esta investigación se usó una validación cruzada con 20 grupos (*20-Fold Cross-validation*) [26]. Este método tiene como limitación que los datos utilizados para el entrenamiento de los clasificadores entre una iteración y otra, tienen una alta correlación.

### 2.3 La base de datos Herlev

Una minuciosa base de datos de imágenes de citología cervical se confeccionó en el hospital danés Herlev. En su contenido se encuentran un total de 917 imágenes, agrupadas en 7 clases diferentes [24]. El panel de analistas que clasificó las imágenes estuvo conformado por técnicos experimentados y doctores en medicina. Para determinar la clasificación de las células, cada una de estas fue analizada por dos técnicos. Para las muestras de difícil diagnóstico se consultó la opinión del médico, en caso de no existir

unanimidad de criterio se descartaron esas imágenes. Las imágenes fueron segmentadas manualmente por los especialistas de esta institución, quedando entonces solo una célula por imagen, igualmente las más contradictorias fueron desechadas. La técnica usada para la toma de las imágenes digitales fue el frotis convencional. El objetivo perseguido por los especialistas con este trabajo fue brindar una fuente de información para estudios de clasificación automatizada en imágenes de citología cervical. En este estudio, para la clasificación, las células fueron agrupadas en 2 categorías: normales (sin anomalía) y anómalas (con algún tipo de desviación de la normalidad). El 73,6% de las imágenes corresponden a células con algún tipo de anomalía; la composición precisa de las clases se muestra en la Tabla 2.

Tabla 2. Distribución de la base de datos Herlev.

Clases	Categorías	Tipo de célula	Cantidad de células	Subtotal
1	Normal	Superficial escamosa epitelial	74	
2	Normal	Intermedia escamosa epitelial	70	
3	Normal	Columnar escamosa epitelial	98	242
4	Anormal	Ligera escamosa displasia	182	
5	Anormal	Moderada escamosa displasia	146	
6	Anormal	Severa escamosa displasia	197	
7	Anormal	Carcinoma in situ	150	675

### 2.3.1 Uso de matrices de rasgos en la implementación de clasificadores

Para este trabajo se seleccionaron dos conjuntos de rasgos diferentes, basados ambos sólo en el núcleo de las células de citología cervical, principal diferencia comparada con el trabajo de Jantzen et al [24]. El primer conjunto consta de 9 rasgos y se le llamará R1, estos se obtienen casi todos con la función *regionprops* de Matlab. Los rasgos que no están



contenidos en dicha función son la relación  $(\sqrt{(\text{Área})})/\text{Perímetro}$ , y la entropía de Shannon. Los nueve rasgos basados en la forma (excepto *relac* y *ent*) se muestran en la Tabla 3.

Tabla 3. Conjunto de rasgos 1 utilizado en la clasificación, relacionado con la forma.

Rasgos	Nombre en matriz de rasgos
$(\sqrt{(\text{Área})})/\text{Perímetro}$	relac
Entropía.	ent
Intensidad Media.	statsxi.MeanIntensity
Intensidad Máxima.	statsxi.MaxIntensity
Intensidad Mínima.	statsxi.MinIntensity
Solidez.	statsxi.Solidity
Excentricidad.	statsxi.Eccentricity
Área.	statsxi.Area
Perímetro	statsxi.Perimeter

El segundo conjunto de rasgos está basado en la textura de los núcleos celulares y se le llamará R3. Para calcularlos se determinó primeramente el mayor cuadrado inscrito en el núcleo de cada célula [27] (ver Figura 3).

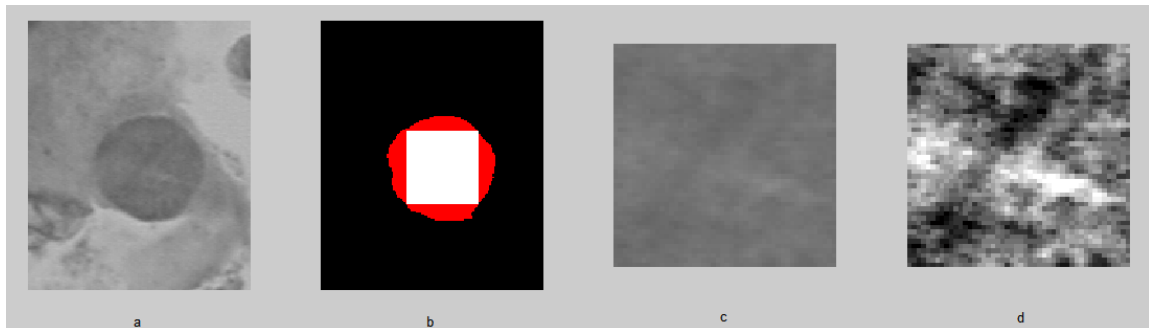


Figura 3. En esta figura se observa en ‘a’ la imagen original en escala de grises. En ‘b’ el máximo cuadrado inscrito en el núcleo. En ‘c’ la parte del núcleo que coincide con el mayor cuadrado inscrito. En ‘d’ luego de aplicarle a ‘c’ un ajuste de la intensidad con *imadjust*. Estas imágenes pertenecen a la clase 4 (anómala).

Luego se aumenta el contraste de cada una de estas nuevas imágenes cuadradas, mediante la función *imadjust*. Para crear la matriz de co-ocurrencia se empleó la función *graycomatrix*. El offset utilizado para el cálculo de esta matriz fue 1, con ángulos de  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$ . Se construyó una matriz resultante con los máximos de las matrices de los ángulos. Con la matriz de máximos es con la se pueden calcular entonces los coeficientes

de Haralick. Los rasgos basados en textura que se utilizaron para la clasificación mediante la función *GLCM\_Features1* se muestran en la Tabla 4.

Tabla 4. Conjunto de rasgos número 3 utilizado en la clasificación, relacionados con textura.

Autocorrelación	(out.autoc)
Contraste	(out.contr)
Correlación de Matlab	(out.corm)
Correlación de la función	(out.corrp)
Prominencia del Cluster	(out.cprom)
Sombra del Cluster	(out.cshad)
Disimilitud	(out.dissi)
Energía	(out.energ)
Entropía	(out.entro)
Homogenidad de Matlab	(out.homom)
Homogenidad de la función	(out.homop)
Máxima probabilidad	(out.maxpr)
Suma de cuadrados: Varianza	(out.sosvh)
Promedio de la suma	(out.savgh)
Varianza de la suma	(out.svarh)
Entropía de la suma	(out.senth)
Diferencia de varianza	(out.dvarh)
Diferencia de entropía	(out.denth)
Información de la medida de correlación1	(out.inf1h)
Información de la medida de correlación2	(out.inf2h)
Diferencia Inversa Normalizada (INN)	(out.indnc)
Diferencia inversa normalizada	(out.idmnc)

### 2.1.1 Estructura de datos utilizada

En la realización de esta investigación se trabajó con la utilidad *cell array* de Matlab para organizar los datos manejados en la clasificación. Contamos con varios conjuntos de datos destinados a probar los algoritmos clasificadores, los cuales fueron creados de forma similar en otro trabajo [27]. Nuestros *arrays* cuentan con una fila y dos columnas, la posición (1,1) es la matriz de rasgos y la posición (1,2) representa el arreglo de nombres. La matriz de rasgos está compuesta por N filas y M columnas, N es el número de células que se estén analizando y M la cantidad de rasgos seleccionados. Si utilizamos la base de datos completa y los rasgos geométricos, la matriz de rasgos tendría N=917 y M=9. El arreglo de

nombres que corresponde a la posición (1,2) en el *array* original, está formada por un único vector columna. El contenido de dicho vector hace referencia a cada una de las siete clases de cada célula, quedando estas etiquetadas. El número de filas de la matriz de rasgos y el de arreglo de nombres tienen que ser iguales.

## 2.4 Reducción de la dimensionalidad de los datos

Con el fin de mejorar la clasificación de las células cervicales, se propuso reducir la dimensionalidad de las matrices de rasgos [28]. La técnica usada para la obtención de los nuevos subconjuntos de rasgos está basada en un esquema lineal. En la literatura se puede encontrar también esquemas no lineales para la reducción de la dimensionalidad [12], [29]. En este trabajo la técnica utilizada fue el Análisis de Componentes Principales (PCA) [28]. En principio se cuenta con un primer conjunto de rasgos relacionados principalmente con la forma de los núcleos celulares, el cual consta de nueve rasgos (ver Tabla 3). El segundo conjunto de rasgos está relacionado con la textura de los núcleos celulares y consta de veintidós rasgos (ver Tabla 4). El tercer conjunto no es más que la unión de los rasgos textura con los rasgos basados en la forma. A estos tres conjuntos de rasgos se le aplica la técnica PCA para así quedar conformados los rasgos utilizados en la clasificación.

## 2.5 Prueba de hipótesis estadísticas en la evaluación de clasificadores

Para determinar la superioridad de un clasificador con respecto a los demás se utiliza primeramente la prueba de Friedman. Dicho test es el equivalente no paramétrico de un diseño de medidas repetidas para una muestra o un análisis de varianza de dos factores con una observación por casilla. En la prueba de Friedman se contrasta la hipótesis nula de que las  $k$  variables relacionadas procedan de la misma población. En cada caso, las  $k$  variables se ordenan en un rango de 1 a  $k$ . El estadístico de contraste se basa en estos rangos. Para un tamaño constante de muestra, mientras mayor sea el valor de chi-cuadrado, mayor es la diferencia entre la suma de rango de cada variable y su valor esperado. Los grados de libertad son iguales al número de variables menos 1. En este estudio la significación asintótica debe ser  $p < 0.05$ , para afirmar que un clasificador es realmente superior a los otros.

Luego de establecer un ranking de los cuatro clasificadores utilizados en la investigación, se realiza una validación adicional mediante la prueba de Wilcoxon [30]. Se enfrentan dos a dos los clasificadores utilizados. Aquí se comparan las distribuciones de dos variables relacionadas. Si las dos variables tienen una distribución similar, el número de diferencias positivas y negativas no difiere de forma significativa. En la prueba de rangos con signo de Wilcoxon se toma como hipótesis nula que las medianas de los valores obtenidos para la predictividad negativa y la medida F por los dos algoritmos no difieren. Este análisis estadístico se realizó mediante el software SPSS 15.0 [31].

## 2.6 Programas para la clasificación y evaluación

El programa empleado para realizar la clasificación de los diferentes algoritmos se describe a continuación:

1. Cargar el fichero que contiene la matriz de rasgos y el arreglo de los nombres abreviados de las 7 clases.
2. Se crean dos *cell array* nombrados *posval* y *negval*, dentro de los cuales se declaran los nombres de las células que presentan algún tipo de anomalía (clase4, clase5, clase6, clase7) y las que no lo presentan (clase1, clase2, clase3) (ver anexo IV). En *posval* quedan los nombres de las células que serían positivas (anómalas) y en *negval* las negativas (normales) para realizar una clasificación binaria.
3. Se seleccionan de forma aleatoria los conjuntos de prueba y de entrenamiento con  $k=20$ , mediante *crossvalind* de la forma:  

```
indices = crossvalind('Kfold',ARREGLO_DE_NOMBRES,20);
```

Estos conjuntos serán siempre los mismos para cada uno de los algoritmos de clasificación pues, para cada corrida de dichos algoritmos, se utilizan los mismos valores de *indices*, logrando obtener datos pareados.
4. Se inicializa el objeto de clasificación con:  

```
cp = classperf(name_class,'Positive',posval,'Negative',negval);
```
5. Se reduce la dimensionalidad de los datos mediante la alternativa PCA.
6. Se entrena el clasificador usando la alternativa *linear* a través de la función *classify*.

7. Se utiliza un ciclo *for* para garantizar que el proceso de evaluación se repita  $K$  veces, mediante validación cruzada (*crossvalind*) las muestras numeradas obtendrán siempre distribuciones diferentes.
8. Se evalúa la calidad del clasificador mediante los valores de la tasa de error, la tasa de clasificación correcta, la sensibilidad, la especificidad, la predictividad positiva, la predictividad negativa y la medida F, mediante *classperf*.
9. Se realizan las mismas operaciones a partir del paso 4 hasta el paso 7, empleando el clasificador de distancia de Mahalanobis.
10. Se realizan las mismas operaciones a partir del paso 4 hasta el paso 7, empleando el clasificador SVM.
11. Se realizan las mismas operaciones a partir del paso 4 hasta el paso 7, empleando el clasificador de vecinos más cercanos (knn).

En todos los casos se realiza la clasificación con los mismos datos de entrenamiento y de prueba, logrando datos pareados.

## CAPÍTULO 3. RESULTADOS Y DISCUSIÓN

En este capítulo se analizan los resultados de todos los experimentos realizados en cuanto a los índices de efectividad: tasa de error, la tasa de clasificación correcta, la sensibilidad, la especificidad, la predictividad positiva, la predictividad negativa y la medida F (se usó  $k=20$  en cada corrida). Se emplea primeramente el conjunto de rasgos uno (R1) relacionados con la forma, a estos se le aplica una reducción de la dimensionalidad mediante la técnica de análisis de componentes principales (PCA) y se le llama R2. Luego se analiza el conjunto de rasgos dos (R3) relacionados con textura al cual se utiliza con reducción de dimensionalidad. Finalmente se hace la unión de R1 y R3 obteniendo un nuevo conjunto de rasgos (R4), el que será utilizado en la clasificación luego de haberle aplicado una reducción de la dimensionalidad. Se ejecuta un análisis estadístico mediante la prueba de Friedman para determinar cuál de los algoritmos utilizados es el más eficiente en la clasificación de las células con rasgos de los núcleos celulares. Para una validación adicional de los métodos se efectúa la prueba de Wilcoxon, enfrentando dos a dos los clasificadores utilizados en el estudio.

### 3.1 Índices de desempeño obtenidos

Con el objetivo de comprobar el comportamiento de los clasificadores, se observan los ambientes de las 7 clases (ver Figura 4). Se decide modelar experimentos que clasifiquen simplemente a un par de clases que estén suficientemente apartadas entre sí y además que correspondan a conjuntos diferentes (normal o anómala). Las pruebas se realizan con las clases 3 y 4. Luego de determinar el clasificador que tiene mejor comportamiento se realiza con él un análisis para la base de datos completa.

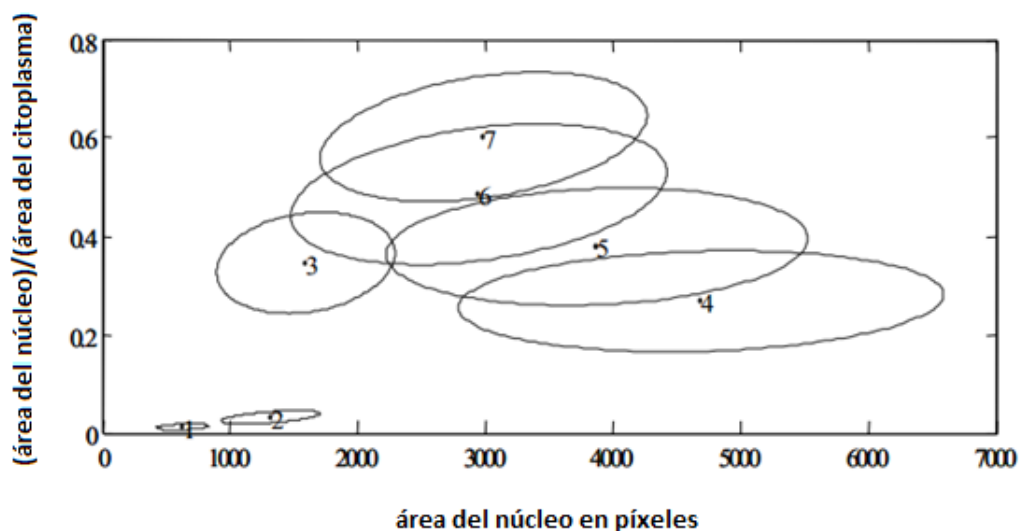


Figura 4. Distribución de las siete clases de la base de datos Herlev, basadas solamente en dos rasgos.

Los resultados alcanzados para todos los clasificadores se ordenan en una matriz, que posteriormente es analizada mediante la función *normfit* de Matlab. Esta función devuelve la media y desviación estándar de los índices de desempeño para cada clasificador. Los resultados se muestran en el anexo IV.

Para la selección del mejor clasificador se tiene en cuenta el costo computacional de cada algoritmo. Se calculó el tiempo que demoró el algoritmo en realizar las 20 iteraciones previstas.

### 3.2 Resultados del clasificador de Mahalanobis

Las pruebas efectuadas al clasificador, basado en la distancia de Mahalanobis, se realizaron con cuatro conjuntos de rasgos. El primer conjunto R1 (ver Tabla 3) está basado en la forma, el conjunto de rasgos dos R2, es el mismo R1 pero con reducción de la dimensionalidad. El conjunto R3 (ver Tabla 4) está basado en la textura, pero cuando se usa en la clasificación se le reduce la dimensionalidad. El último conjunto de rasgos que se usó, se le llamó R4 que no es más que la unión de R1 y R3. Para emplear R4 se reduce su dimensionalidad. Se ejecutaron varias pruebas previas para determinar que valores en la

reducción de la dimensionalidad (RD) resultaban ventajosos. Los valores obtenidos fueron: en el caso R1 una RD=7, en el caso R3 y R4 una RD=17.

Tabla 5. Desempeño del clasificador de distancia de Mahalanobis, utilizando las clases tres y cuatro de la base de datos Herlev.

	Clasificador Mahalanobis			
	rasgos geométricos (R1)	rasgos de textura(R3)	unión de rasgos(R4)	
	sin RD	RD=7	RD=17	RD=17
<b>Er</b>	0,075	0,090	0,166	<b>0,035</b>
<b>Cr</b>	0,925	0,910	0,834	<b>0,965</b>
<b>Se</b>	0,943	0,891	0,745	<b>0,947</b>
<b>Sp</b>	0,891	0,946	<b>1,000</b>	0,998
<b>Pp</b>	0,942	0,969	<b>1,000</b>	0,999
<b>Pn</b>	0,895	0,823	0,677	<b>0,910</b>
<b>medida F</b>	0,916	0,917	0,854	<b>0,971</b>

En la Tabla 5 se puede observar el comportamiento del clasificador basado en la distancia de Mahalanobis. En ella queda registrada la media de los índices de desempeño calculados para este clasificador con los cuatro conjuntos de rasgos usados. Se evidencia que existe una mejoría significativa cuando se utiliza el conjunto de rasgos R4. Para determinar la veracidad de esta mejoría, sometemos estos resultados a un análisis estadístico.

### 3.2.1 Análisis estadístico para el clasificador de Mahalanobis usando la medida F

El análisis estadístico se realiza con el programa SPSS a través de pruebas no paramétricas para k muestras relacionadas. Se utiliza en primera instancia la prueba de Friedman, porque los algoritmos de clasificación se realizan para datos pareados. Por este medio se evalúa la calidad del clasificador para los distintos conjuntos de rasgos. Esta prueba nos devuelve un posible ranking del comportamiento del clasificador para las cuatro variantes de rasgos, donde el valor más alto indica un mejor comportamiento. Para corroborar el ranking se aplica la prueba de Wilcoxon. Aquí se enfrentan de dos en dos los algoritmos de la forma primero con segundo, segundo con tercero etc. De este modo se determina si existe diferencia estadística entre los métodos que se enfrentaron. Los parámetros a analizar en cuanto a índices de efectividad son: la predictividad negativa ( $Pn$ ) y la medida F.



Tabla 6. Prueba de Friedman realizada al clasificador de distancia de Mahalanobis, utilizando la Medida F.

	Rango promedio
Mahala (R1)	2,80
mahala RD (R2)	2,23
mahala tex (R3)	1,00
mahala unión (R4)	3,98

Según los resultados de la Tabla 6, el conjunto de rasgos R4 es el de mejor comportamiento en la clasificación utilizando la medida F, empleando el algoritmo basado en la distancia de Mahalanobis. En segundo lugar está R1, en tercero R2 y en cuarto R3. La baja significación asintótica que se aprecia en la Tabla 7 expresa que es poco probable que los resultados obtenidos se deban al azar. Se rechaza la hipótesis nula de que ambos conjuntos provienen de la misma población.

Tabla 7. Estadísticos de contraste de la prueba de Friedman para el clasificador de distancia de Mahalanobis, utilizando la Medida F.

N	20
Chi-cuadrado	55,372
gl	3
Sig. asintót.	,000

Al aplicar la prueba de los rangos de Wilcoxon siguiendo el orden obtenido en la prueba de Friedman se tiene la Tabla 8. Finalmente en la Tabla 9 se aprecia que para el primer caso, la significación asintótica tiene un valor muy bajo, lo que afirma que los rasgos de textura más los rasgos geométricos (R4) son capaces de discriminar mejor entre células anómalas y normales.

Tabla 8. Prueba de los rangos con signo de Wilcoxon para el clasificador de distancia de Mahalanobis, utilizando la Medida F.

		N	Rango promedio	Suma de rangos
mahala unión - mahala	Rangos negativos	0(a)	,00	,00
	Rangos positivos	20(b)	10,50	210,00
	Empates	0(c)		
	Total	20		
mahala RD - mahala	Rangos negativos	16(d)	8,81	141,00
	Rangos positivos	4(e)	17,25	69,00
	Empates	0(f)		
	Total	20		
mahala tex - mahala RD	Rangos negativos	20(g)	10,50	210,00
	Rangos positivos	0(h)	,00	,00
	Empates	0(i)		
	Total	20		

a mahala unión &lt; mahala

b mahala unión &gt; mahala

c mahala unión = mahala

d mahala RD &lt; mahala

e mahala RD &gt; mahala

f mahala RD = mahala

g mahala tex &lt; mahala RD

h mahala tex &gt; mahala RD

i mahala tex = mahala RD

Tabla 9. Estadísticos de contraste de la prueba de Wilcoxon para el clasificador de distancia de Mahalanobis, utilizando la Medida F.

	mahala unión - mahala	mahala RD - mahala	mahala tex - mahala RD
Z	-3,920(a)	-1,344(b)	-3,920(b)
Sig. asintót. (bilateral)	,000	,179	,000

a Basado en los rangos negativos.

b Basado en los rangos positivos.

c Prueba de los rangos con signo de Wilcoxon

### 3.2.2 Análisis estadístico para el clasificador de Mahalanobis usando la predictividad negativa.

Un análisis estadístico similar al anterior se realiza para la predictividad negativa ( $P_n$ ). En este caso la prueba de Friedman indica que al igual que ocurrió con la medida F, R4 tiene un comportamiento superior. Igualmente como se muestra en la Tabla 11, los estadísticos de contraste reflejan una baja significación asintótica y un alto valor de chi cuadrado. En el caso de la prueba de Wilcoxon, nuevamente se muestra un bajo valor de la significación asintótica (ver Tabla 13) por lo que se puede declarar que el conjunto de rasgos R4 es superior al resto.

Tabla 10. Prueba de Friedman realizada al clasificador de distancia de Mahalanobis, utilizando la Predictividad Negativa.

	Rango promedio
mahala	3,10
mahala RD	2,03
mahala tex	1,00
mahala unión	3,88

Tabla 11. Estadísticos de contraste de la prueba de Friedman para el clasificador de distancia de Mahalanobis, utilizando la Predictividad Negativa.

N	20
Chi-cuadrado	57,000
gl	3
Sig. asintót.	,000

Tabla 12. Prueba de los rangos con signo de Wilcoxon para el clasificador de distancia de Mahalanobis, utilizando la Predictividad Negativa.

		N	Rango promedio	Suma de rangos
mahala unión - mahala	Rangos negativos	2(a)	19,50	39,00
	Rangos positivos	18(b)	9,50	171,00
	Empates	0(c)		
	Total	20		
mahala RD - mahala	Rangos negativos	20(d)	10,50	210,00
	Rangos positivos	0(e)	,00	,00
	Empates	0(f)		
	Total	20		
mahala tex - mahala RD	Rangos negativos	20(g)	10,50	210,00
	Rangos positivos	0(h)	,00	,00
	Empates	0(i)		
	Total	20		

- a mahala unión < mahala
- b mahala unión > mahala
- c mahala unión = mahala
- d mahala RD < mahala
- e mahala RD > mahala
- f mahala RD = mahala
- g mahala tex < mahala RD
- h mahala tex > mahala RD
- i mahala tex = mahala RD

Tabla 13. Estadísticos de contraste de la prueba de Wilcoxon para el clasificador de distancia de Mahalanobis, utilizando la Predictividad Negativa.

	mahala unión - mahala	mahala RD - mahala	mahala tex - mahala RD
Z	-2,464(a)	-3,920(b)	-3,920(b)
Sig. asintót. (bilateral)	,014	,000	,000

a Basado en los rangos negativos.

b Basado en los rangos positivos.

c Prueba de los rangos con signo de Wilcoxon

### 3.3 Resultados para las clases 1 y 7

Con el objetivo de explorar el comportamiento de los rasgos se realizan experimentos similares a los hechos para las clases 3 y 4, en este caso se utilizan como base las clases 1 y 7. Los resultados luego de utilizar el clasificador de Mahalanobis sobre este par de clases distantes se muestran en la Tabla 14.

Tabla 14. Resultados del clasificador de Mahalanobis para las clases 1 y 7

	Clasificador Mahalanobis			
	rasgos geométricos	rasgos de textura	unión de rasgos	
	sin RD	RD=7	RD=17	RD=17
<b>Er</b>	0,043	0,025	0,083	<b>0,022</b>
<b>Cr</b>	1,000	1,000	0,941	<b>1,000</b>
<b>Se</b>	0,957	0,975	0,917	<b>0,978</b>
<b>Sp</b>	0,869	0,924	0,870	<b>0,935</b>
<b>Pp</b>	0,940	0,964	0,936	<b>0,969</b>
<b>Pn</b>	1,000	1,000	0,881	<b>1,000</b>
<b>medida F</b>	0,929	0,960	0,904	<b>0,966</b>

### 3.4 Resultados para la todas las clases

Al realizar la clasificación utilizando toda la base de datos, los resultados obtenidos no fueron favorables. La principal dificultad se tiene con el conjunto de rasgos de textura, lo que afecta los índices de desempeño para la unión de los rasgos de textura y geométricos (ver Tabla 15). Esto requiere profundizar en el comportamiento de los rasgos para todas clases. En las recomendaciones se sugerirán algunas líneas de continuidad del trabajo que tomen en cuenta la necesidad de dar solución a estos problemas.

Tabla 15. Resultados del clasificador de Mahalanobis para toda la base de datos

	<b>Clasificador Mahalanobis</b>			
	rasgos geométricos	rasgos de textura	unión de rasgos	
	sin RD	RD=7	RD=17	RD=17
<b><i>Er</i></b>	0,585	0,608	0,621	<b>0,555</b>
<b><i>Cr</i></b>	0,415	0,392	0,379	<b>0,445</b>
<b><i>Se</i></b>	0,857	0,857	0,857	<b>0,889</b>
<b><i>Sp</i></b>	0,88	<b>0,892</b>	0,577	0,828
<b><i>Pp</i></b>	0,953	<b>0,958</b>	0,852	0,936
<b><i>Pn</i></b>	0,689	0,692	0,587	<b>0,73</b>
<b><i>medida F</i></b>	0,868	<b>0,873</b>	0,689	0,856

### 3.5 Selección del mejor clasificador

Para precisar quiénes competirán por el título de mejor clasificador, se arregló un estudio similar al ejecutado para el clasificador de Mahalanobis, además se tuvo en cuenta el costo computacional de cada algoritmo. Se determinó para cada algoritmo cuál era el conjunto de rasgos que tenía un comportamiento superior a los demás (ver anexo I, II y III) mediante los análisis estadísticos correspondientes. Los mejores resultados obtenidos se muestran en la Tabla 16. El análisis estadístico que se le realizará a los datos, está basado en la medida F y en la Predictividad Negativa de los cuatro clasificadores en competencia. El estudio elaborado con estos datos es semejante al ilustrado para el clasificador de Mahalanobis, solo que en este caso se determinará el mejor clasificador pues ya se encontraron las mejores variantes de rasgos. En tres de los cuatro casos, el mejor conjunto de rasgos resultó ser R4, lo que demuestra que la textura incide positivamente en la clasificación de células en la prueba de Papanicolaou.

Para determinar el costo computacional, se calculó el tiempo que demoraron en cada corrida los cuatro algoritmos con los conjuntos de rasgos mostrados en la Tabla 17. El clasificador SVM resultó ser el de peor comportamiento, mostrando alto consumo de tiempo comparados con los tres restantes. Los clasificadores (Mahalanobis, lineal, knn) se comportaron de manera similar, destacando al knn que fue el más rápido en correr.

Tabla 16. Desempeño de todos los clasificadores utilizando el conjunto de rasgos que resultó ser el mejor para cada uno de ellos.

	clasificador			
	lineal	mahalanobis	svm	knn
	unión de rasgos	unión de rasgos	unión de rasgos	rasgos geométricos
	RD=17	RD=17	RD=17	RD=7
<i>Er</i>	0,079	<b>0,035</b>	0,053	0,150
<i>Cr</i>	0,921	<b>0,965</b>	0,947	0,850
<i>Se</i>	0,956	0,947	<b>0,960</b>	0,916
<i>Sp</i>	0,856	<b>0,998</b>	0,922	0,726
<i>Pp</i>	0,927	<b>0,999</b>	0,9587	0,863
<i>Pn</i>	0,913	0,910	<b>0,925</b>	0,824
<i>medida F</i>	0,902	<b>0,971</b>	0,940	0,809

Tabla 17. Tiempo que demora la clasificación para las mejores variantes de los distintos clasificadores.

	clasificador			
	lineal	mahalanobis	SVM	Knn
	R4	R4	R4	R2
	tiempo en seg	0,152	0,151	14,619 0,115

### 3.5.1 Análisis estadístico para determinar el mejor clasificador usando la medida F

En la Tabla 18 se aprecia que el clasificador de Mahalanobis está en el primer lugar del escalafón de la prueba de Friedman, seguido por la máquina de soporte vectorial, el clasificador lineal y el método de los vecinos más cercanos en ese orden. En la Tabla 19 se contempla un alto valor de chi cuadrado y una baja significación asintótica, indica que se rechaza la hipótesis de que los clasificadores pertenezcan a poblaciones iguales. Esto se reafirma en la Tabla 21 con una muy baja significación asintótica.

Tabla 18. Prueba de Friedman para determinar el mejor clasificador, utilizando la Medida F.

	Rango promedio
SVM_rbf	2,98
C_LINEAL	2,03
Mahalanobis	4,00
C_knn	1,00

Tabla 19. Estadísticos de contraste de la prueba de Friedman para todos los clasificadores utilizando la medida F.

N	20
Chi-cuadrado	59,714
gl	3
Sig. asintót.	,000

Tabla 20. Prueba de los rangos con signo de Wilcoxon para el todos los clasificadores, utilizando la Medida F.

		N	Rango promedio	Suma de rangos
Mahalanobis - SVM_rbf	Rangos negativos	0(a)	,00	,00
	Rangos positivos	20(b)	10,50	210,00
	Empates	0(c)		
	Total	20		
C_LINEAL - SVM_rbf	Rangos negativos	19(d)	10,00	190,00
	Rangos positivos	0(e)	,00	,00
	Empates	1(f)		
	Total	20		
C_knn - C_LINEAL	Rangos negativos	20(g)	10,50	210,00
	Rangos positivos	0(h)	,00	,00
	Empates	0(i)		
	Total	20		

a Mahalanobis < SVM\_rbf

b Mahalanobis > SVM\_rbf

c Mahalanobis = SVM\_rbf

d C\_LINEAL < SVM\_rbf

e C\_LINEAL > SVM\_rbf

f C\_LINEAL = SVM\_rbf

g C\_knn < C\_LINEAL

h C\_knn > C\_LINEAL

i C\_knn = C\_LINEAL

Tabla 21. Estadísticos de contraste de la prueba de Wilcoxon para todos los clasificadores utilizando la Medida F.

	Mahalanobis - SVM_rbf	C_LINEAL - SVM_rbf	C_knn - C_LINEAL
Z	-3,920(a)	-3,823(b)	-3,920(b)
Sig. asintót. (bilateral)	,000	,000	,000

### 3.5.2 Análisis estadístico para determinar el mejor clasificador usando la predictividad negativa

Las pruebas realizadas para determinar el mejor algoritmo de clasificación usando como índice de desempeño la predictividad negativa son semejantes a las anteriormente realizadas para la medida F. De igual manera se obtiene un ranking encabezado en este caso por la máquina de soporte vectorial (ver Tabla 22), seguido por el clasificador de distancia de Mahalanobis, el clasificador lineal y por último el de vecinos más cercanos. La baja significación asintótica y el alto valor de chi cuadrado en la Tabla 23 hacen pensar que existe diferencia estadística entre el SVM y el resto de los algoritmos. Luego de la prueba de Wilcoxon (ver Tabla 24 y Tabla 25) se descubre que entre los dos primeros clasificadores la significación asintótica no está por debajo de 0.05, por lo que se acepta la hipótesis nula de que los clasificadores provienen de una misma población.

Tabla 22. Prueba de Friedman para determinar el mejor clasificador, utilizando la Predictividad Negativa (PN).

	Rango promedio
C_LINEAL unión RD	2,33
SVM_rbf unión RD	3,43
mahala unión	3,25
C_knn RD	1,00

Tabla 23. Estadísticos de contraste de la prueba de Friedman para todos los clasificadores utilizando la Predictividad Negativa (PN).

N	20
Chi-cuadrado	44,608
gl	3
Sig. asintót.	,000



Tabla 24. Prueba de los rangos con signo de Wilcoxon enfrentando dos a dos los clasificadores. Se utiliza la Predictividad Negativa (PN) como índice de desempeño. RD

		N	Rango promedio	Suma de rangos
mahala unión - SVM_rbf unión RD	Rangos negativos	11(a)	13,00	143,00
	Rangos positivos	9(b)	7,44	67,00
	Empates	0(c)		
	Total	20		
mahala unión - C_LINEAL unión RD	Rangos negativos	4(d)	18,25	73,00
	Rangos positivos	16(e)	8,56	137,00
	Empates	0(f)		
	Total	20		
C_knn RD - C_LINEAL unión RD	Rangos negativos	20(g)	10,50	210,00
	Rangos positivos	0(h)	,00	,00
	Empates	0(i)		
	Total	20		

a mahala unión < SVM\_rbf unión RD

b mahala unión > SVM\_rbf unión RD

c mahala unión = SVM\_rbf unión RD

d mahala unión < C\_LINEAL unión RD

e mahala unión > C\_LINEAL unión RD

f mahala unión = C\_LINEAL unión RD

g C\_knn RD < C\_LINEAL unión RD

h C\_knn RD > C\_LINEAL unión RD

i C\_knn RD = C\_LINEAL unión RD

Tabla 25. Estadísticos de contraste de la prueba de Wilcoxon para todos los clasificadores utilizando la Predictividad Negativa (PN).

	mahala unión - SVM_rbf unión RD	mahala unión - C_LINEAL unión RD	C_knn RD - C_LINEAL unión RD
Z	-1,419(a)	-1,195(b)	-3,920(a)
Sig. asintót. (bilateral)	,156	,232	,000

a Basado en los rangos positivos.

b Basado en los rangos negativos.

c Prueba de los rangos con signo de Wilcoxon

### **3.6 Conclusiones del capítulo**

De acuerdo a los resultados obtenidos en las tablas, se puede concluir que el clasificador de Mahalanobis con el conjunto de rasgos R4 tiene un comportamiento superior a los otros clasificadores en cuanto a la medida F. En el caso de la predictividad negativa, no existen evidencias estadísticas para determinar la superioridad entre SVM y Mahalanobis, pero el SVM fue el que tuvo el peor tiempo de corrida cuando se analizaba el costo computacional en los diferentes algoritmos.

## CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

Después de examinar los resultados de esta investigación se puede arribar a las siguientes conclusiones:

- 1 El conjunto de rasgos más efectivos para determinar un alto desempeño de los clasificadores al separar las clases 3 y 4 o 1 y 7 resultó ser R4, la unión de rasgos geométricos con rasgos de textura.
- 2 Conforme con los métodos estadísticos: la prueba de Friedman y la de Wilcoxon, realizada a los 4 clasificadores implementados para un par de clases distantes en el espacio de parámetros (Normal Columnar y Anormal Ligera), se observó que el algoritmo clasificador que utiliza la distancia de Mahalanobis se comportó de forma superior al SVM, al clasificador lineal y al knn.
- 3 Los resultados al tratar de extender las pruebas a la base de datos completa no fueron satisfactorios, lo que indica la necesidad de continuar profundizando en la definición de los rasgos de textura a emplear y la forma de adquirirlos, así como los métodos más apropiados para reducir la dimensionalidad.

### Recomendaciones

A partir de los resultados obtenidos en esta investigación, se recomienda para estudios posteriores:

- 1 Realizar las pruebas con otros algoritmos clasificadores que no se incluyeron en este trabajo. Especialmente con otras variantes de SVM.

- 2 Utilizar otros algoritmos para la reducción de la dimensionalidad de los datos, considerando PCA con kernels no lineales y algoritmos de selección de rasgos.
- 3 Evaluar de forma exhaustiva los resultados que se obtendrían con otros parámetros de cálculo de las matrices de co-ocurrencia (offset, ángulo).
- 4 Utilizar otros rasgos relacionados con la textura que no se utilizaron en este estudio, como pueden ser los momentos granulométricos.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] G. N. Papanicolaou, «A new procedure for staining vaginal smears», *Science*, vol. 95, n.º 2469, pp. 438–439, 1942.
- [2] S. Noraini Sulaiman, N. A. Mat Isa, y I. Yusoff, «Overlapping cells separation method for cervical cell images», presentado en 10th International Conference on Intelligent Systems Design and Applications, 2010, pp. 1218-1222.
- [3] M. Orozco Monteagudo, C. Mihai, T. Geerink, H. Sahli, y A. Taboada Crispí, «Pap-Smear Cell Nucleus Extraction by using Morphological Watershed», presentado en V Latin American Congress on Biomedical Engineering, 2011.
- [4] R. E. Wood y R. C. González, *Digital Image Processing*, 3ra ed. Singapore: Pearson Education, 2004.
- [5] M. E. Plissiti, C. Nikou, y A. Charchanti, «Automated Detection of Cell Nuclei in Pap Smear Images Using Morphological Reconstruction and Clustering», *Ieee Trans. Inf. Technol. Biomed.*, vol. 15, n.º 2, pp. 233-241, 2011.
- [6] S. N. Sulaiman, N. A. Mat Isa, y N. H. Othman, «Pseudo Color Features Extraction Technique for Cervical Cancer of Pap Smear Images», presentado en 10th International Conference on Intelligent Systems Design and Applications (ISDA), 2010, pp. 314 - 319.
- [7] J. A. Giménez-Mas, P. Sanz-Moncasi, J. Alfaro-Torres, C. Hörndler, y E. Urbiola-Marcilla, «Evaluación de dispositivos automatizados para diagnóstico citológico en la prevención del cáncer de cérvix», *Rev. Española Patol.*, vol. 35, pp. 301-313, 2003.
- [8] P. Bulgaresi, M. P. Carlaggi, G. M. Troni, y S. Clatto, «Quality control of the Autopap screening system employed as a primary screening device: rapid review of smears coded as no further review», *Tumori*, vol. 92, pp. 276-278, 2006.
- [9] S. J. Coon y B. M. Craig, «Assessing Human and Economic Benefits of Cancer Prevention», en *Fundamentals of Cancer Prevention*, D. S. Alberts y L. M. Hess, Eds. Berlin Heidelberg: Springer-Verlag, 2008, p. 533.
- [10] E. S. Cibas, «Cervical Cytology Automation: the U.S. Experience», presentado en 17th International Congress of Cytology, Edinburgh, Scotland, 2010.
- [11] R. Austin, «Computer-Assisted Papanicolaou Imaging», *Cancer Cytopathol.*, vol. 118, pp. 65-67, 2010.

- [12] S. Srisuk, M. Petrou, R. Fooprateep, K. Sunat, y W. Kurutach, «A Combination of Shape and Texture Classifiers for a Face Verification System», *Lect. Notes Comput. Sci.*, vol. 30, 2004.
- [13] J. L. Gil Rodríguez, «Estado Actual de la Representación y Análisis de Textura en Imágenes», *Serie Azul*, 2008.
- [14] N. A. Mat Isa, M. Y. Mashor, y N. H. Othman, «An automated cervical pre-cancerous diagnostic system», *Artif. Intell. Med.*, vol. 42, pp. 1-11, 2008.
- [15] I. A. Yusoff, N. A. Mat Isa, N. H. Othman, S. N. Sulaiman, y Y. Jusman, «Performance of neural network architectures: Cascaded MLP versus extreme learning machine on cervical cell image classification», presentado en 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010), 2010, pp. 308-311.
- [16] S. Osowski, R. Siroic, T. Markiewicz, y K. Siwek, «Application of Support Vector Machine and Genetic Algorithm for Improved Blood Cell Recognition», *Ieee Trans. Instrum. Meas.*, vol. 58, pp. 2159-2168, 2009.
- [17] H. Po-Chi, C. Yung-Kuan, C. Po-Chou, C. Yung-Fu, C. Rung-Ching, y H. Y.-R. Huang, «Quantitative Assessment of Pap Smear Cells by PC-Based Cytopathologic Image Analysis System and Support Vector Machine», en *ICMB 2008, LNCS*, D. Zhang, Ed. Springer-Verlag Berlin Heidelberg, 2007, pp. 192-199.
- [18] N. Dehak, R. Dehak, P. Kenny, N. Brammer, P. Ouellet, y P. Dumouchel, «Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification», *LNCS*, 2009.
- [19] Y. Marinakis, G. Dounias, y J. Jantzen, «Pap smear diagnosis using a hybrid intelligent scheme focusing on genetic algorithm based feature selection and nearest neighbor classification», *Comput. Biol. Med.*, vol. 39, pp. 69-78, 2009.
- [20] D. Riana y A. Murni, «Performance evaluation of Pap smear cell image classification using quantitative and qualitative features based on multiple classifiers», presentado en Proceedings of the International Conference on Advanced Computer Science and Information Systems, 2009.
- [21] M. I. Jordan y Y. Weiss, «Analysis and an algorithm on spectral clustering», *Adv. Neural Inf. Process. Syst.*, vol. 14, pp. 849-856, 2002.
- [22] C. Bishop, «Pattern recognition and machine learning», *Springer*, 2006.
- [23] S. Abe, «Support vector machines for pattern classification», New York, 2010.
- [24] J. Jantzen, J. Norup, G. Dounias, y B. Bjerregaard, «Pap-smear Benchmark Data For Pattern Classification», presentado en Nature Inspired Smart Information Systems (NiSIS), 2005, pp. 1-9.
- [25] *Matlab Statistics Toolbox*. The Mathworks, 2009.
- [26] S. Arlot y A. Celisse, «A survey of cross-validation procedures for model selection», *Stat. Surv.*, vol. 4, pp. 40-79, 2010.

- 
- [27] S. B. Huergo Suárez, «Desarrollo de herramientas de cálculos de rasgos para la clasificación de imágenes de citología cervical», pregrado, Universidad Central «Marta Abreu» de Las Villas, Santa Clara, Cuba, 2013.
- [28] M. E. Plissiti y C. Nikou, «Cervical Cell Classification Based Exclusively on Nucleus Feature», A. Campilho., Springer-Verlag Berlin Heidelberg, 2012, pp. 483-490.
- [29] B. Scholkopf, A. Smola, y K. R. Muller, «Nonlinear component analysis as a kernel eigenvalue problem», *Neural Comput.*, pp. 1299–1319, 1998.
- [30] I. R. Miller, J. E. Freund, y R. Johnson, *Probabilidad y estadística para ingenieros*, vol. 2. Prentice-Hall Hispanoamericana, 1992.
- [31] *Tutorial SPSS*. IBM, 2006.

## ANEXOS

### Anexo I Resultados del clasificador lineal

Tabla 26. Desempeño del clasificador lineal utilizando los rasgos geométricos, los rasgos de textura y la unión de ambos conjuntos de rasgos.

	Clasificador lineal			
	rasgos geométricos	rasgos de textura	unión de rasgos	
	sin RD	RD=7	RD=17	RD=17
<b>Er</b>	0,114	0,124	0,176	<b>0,079</b>
<b>Cr</b>	0,886	0,876	0,824	<b>0,921</b>
<b>Se</b>	0,936	0,866	0,907	<b>0,956</b>
<b>Sp</b>	0,793	<b>0,895</b>	0,668	0,856
<b>Pp</b>	0,895	<b>0,939</b>	0,838	0,927
<b>Pn</b>	0,868	0,781	0,793	<b>0,913</b>
<b>medida F</b>	0,857	0,880	0,767	<b>0,902</b>

Tabla 27. Prueba de Friedman realizada al clasificador lineal, utilizando la medida F.

	Rango promedio
C_LINEAL	2,08
C_LINEAL RD	3,00
C_LINEAL tex RD	1,00
C_LINEAL unión RD	3,93

Tabla 28. Estadísticos de contraste de la prueba de Friedman para el clasificador lineal utilizando la medida F.

N	20
Chi-cuadrado	56,819
gl	3
Sig. asintót.	,000



Tabla 29. Prueba de los rangos con signo de Wilcoxon para el clasificador lineal, utilizando la medida F.

		N	Rango promedio	Suma de rangos
C_LINEAL unión RD - C_LINEAL RD	Rangos negativos	1(a)	1,00	1,00
	Rangos positivos	19(b)	11,00	209,00
	Empates	0(c)		
	Total	20		
C_LINEAL RD - C_LINEAL	Rangos negativos	1(d)	16,00	16,00
	Rangos positivos	19(e)	10,21	194,00
	Empates	0(f)		
	Total	20		
C_LINEAL tex RD - C_LINEAL	Rangos negativos	20(g)	10,50	210,00
	Rangos positivos	0(h)	,00	,00
	Empates	0(i)		
	Total	20		

a C\_LINEAL unión RD < C\_LINEAL RD

b C\_LINEAL unión RD > C\_LINEAL RD

c C\_LINEAL unión RD = C\_LINEAL RD

d C\_LINEAL RD < C\_LINEAL

e C\_LINEAL RD > C\_LINEAL

f C\_LINEAL RD = C\_LINEAL

g C\_LINEAL tex RD < C\_LINEAL

h C\_LINEAL tex RD > C\_LINEAL

i C\_LINEAL tex RD = C\_LINEAL

Tabla 30. Estadísticos de contraste de la prueba de Wilcoxon para el clasificador lineal, utilizando la medida F.

	C_LINEAL unión RD - C_LINEAL RD	C_LINEAL RD - C_LINEAL	C_LINEAL tex RD - C_LINEAL
Z	-3,883(a)	-3,323(a)	-3,920(b)
Sig. asintót. (bilateral)	,000	,001	,000

a Basado en los rangos negativos.

b Basado en los rangos positivos.

c Prueba de los rangos con signo de Wilcoxon

Tabla 31. Prueba de Friedman realizada al clasificador lineal, utilizando la Predictividad Negativa.

	Rango promedio
C_LINEAL	3,00
C_LINEAL RD	1,35
C_LINEAL Tex	1,70
C_LINEAL unión RD	3,95

Tabla 32. Estadísticos de contraste de la prueba de Friedman para el clasificador lineal, utilizando la Predictividad Negativa.

N	20
Chi-cuadrado	52,837
gl	3
Sig. asintót.	,000

Tabla 33. Prueba de los rangos con signo de Wilcoxon para el clasificador lineal, utilizando la Predictividad Negativa.

		N	Rango promedio	Suma de rangos
C_LINEAL unión RD - C_LINEAL	Rangos negativos	0(a)	,00	,00
	Rangos positivos	19(b)	10,00	190,00
	Empates	1(c)		
	Total	20		
C_LINEAL Tex - C_LINEAL	Rangos negativos	19(d)	10,00	190,00
	Rangos positivos	0(e)	,00	,00
	Empates	1(f)		
	Total	20		
C_LINEAL Tex - C_LINEAL RD	Rangos negativos	7(g)	12,86	90,00
	Rangos positivos	13(h)	9,23	120,00
	Empates	0(i)		
	Total	20		

a C\_LINEAL unión RD < C\_LINEAL

b C\_LINEAL unión RD > C\_LINEAL

c C\_LINEAL unión RD = C\_LINEAL

d C\_LINEAL Tex < C\_LINEAL

e C\_LINEAL Tex > C\_LINEAL

f C\_LINEAL Tex = C\_LINEAL

g C\_LINEAL Tex < C\_LINEAL RD

h C\_LINEAL Tex > C\_LINEAL RD

i C\_LINEAL Tex = C\_LINEAL RD

Tabla 34. Estadísticos de contraste de la prueba de Wilcoxon para el clasificador lineal, utilizando la Predictividad Negativa.

	C_LINEAL unión RD - C_LINEAL	C_LINEAL Tex - C_LINEAL	C_LINEAL Tex - C_LINEAL RD
Z	-3,823(a)	-3,823(b)	-,560(a)
Sig. asintót. (bilateral)	,000	,000	,575

a Basado en los rangos negativos.

b Basado en los rangos positivos.

c Prueba de los rangos con signo de Wilcoxon

## Anexo II Resultados del clasificador SVM con kernel rbf $\sigma=2$

Tabla 35. Desempeño del clasificador SVM utilizando los rasgos geométricos, los rasgos de textura y la unión de ambos conjuntos de rasgos.

	Clasificador SVM			
	rasgos geométricos	rasgos de textura	unión de rasgos	
	sin RD	RD=7	RD=17	RD=17
<b>Er</b>	0,056	0,069	0,107	<b>0,053</b>
<b>Cr</b>	0,944	0,931	0,893	<b>0,947</b>
<b>Se</b>	0,956	0,938	<b>0,962</b>	0,960
<b>Sp</b>	<b>0,923</b>	0,919	0,764	0,922
<b>Pp</b>	0,9586	0,956	0,886	<b>0,9587</b>
<b>Pn</b>	0,918	0,890	0,920	<b>0,925</b>
<b>medida F</b>	0,939	0,928	0,848	<b>0,940</b>

Tabla 36. Prueba de Friedman realizada al clasificador SVM, utilizando la medida F.

	Rango promedio
SVM_rbf	3,48
SVM_rbf RD	2,30
SVM_rbf tex	1,00
SVM_rbf unión RD	3,23

Tabla 37. Estadísticos de contraste de la prueba de Friedman para el clasificador SVM, utilizando la Medida F.

N	20
Chi-cuadrado	45,422
gl	3
Sig. asintót.	,000

Tabla 38. Prueba de los rangos con signo de Wilcoxon para el clasificador SVM, utilizando la medida F.

		N	Rango promedio	Suma de rangos
SVM_rbf unión RD - SVM_rbf	Rangos negativos	10(a)	9,20	92,00
	Rangos positivos	9(b)	10,89	98,00
	Empates	1(c)		
	Total	20		
SVM_rbf unión RD - SVM_rbf RD	Rangos negativos	5(d)	7,00	35,00
	Rangos positivos	15(e)	11,67	175,00
	Empates	0(f)		
	Total	20		
SVM_rbf tex - SVM_rbf RD	Rangos negativos	20(g)	10,50	210,00
	Rangos positivos	0(h)	,00	,00
	Empates	0(i)		
	Total	20		

a SVM\_rbf unión RD < SVM\_rbf

b SVM\_rbf unión RD > SVM\_rbf

c SVM\_rbf unión RD = SVM\_rbf

d SVM\_rbf unión RD < SVM\_rbf RD

e SVM\_rbf unión RD > SVM\_rbf RD

f SVM\_rbf unión RD = SVM\_rbf RD

g SVM\_rbf tex < SVM\_rbf RD

h SVM\_rbf tex > SVM\_rbf RD

i SVM\_rbf tex = SVM\_rbf RD

Tabla 39. Estadísticos de contraste de la prueba de Wilcoxon para el clasificador SVM utilizando la medida F.

	SVM_rbf unión RD - SVM_rbf	SVM_rbf unión RD - SVM_rbf RD	SVM_rbf tex - SVM_rbf RD
Z	-,121(a)	-2,613(a)	-3,920(b)
Sig. asintót. (bilateral)	,904	,009	,000

a Basado en los rangos negativos.

b Basado en los rangos positivos.

c Prueba de los rangos con signo de Wilcoxon

Tabla 40. Prueba de Friedman realizada al clasificador SVM, utilizando la Predictividad Negativa.

	Rango promedio
SVM_rbf	3,38
SVM_rbf RD	1,70
SVM_rbf tex	1,98
SVM_rbf unión RD	2,95

Tabla 41. Estadísticos de contraste de la prueba de Friedman para el clasificador SVM utilizando la Predictividad Negativa (PN).

N	20
Chi-cuadrado	22,833
gl	3
Sig. asintót.	,000

Tabla 42. Prueba de los rangos con signo de Wilcoxon para el clasificador SVM utilizando la Predictividad Negativa (PN).

		N	Rango promedio	Suma de rangos
SVM_rbf unión RD - SVM_rbf	Rangos negativos	13(a)	8,69	113,00
	Rangos positivos	6(b)	12,83	77,00
	Empates	1(c)		
	Total	20		
SVM_rbf unión RD - SVM_rbf tex	Rangos negativos	6(d)	11,17	67,00
	Rangos positivos	13(e)	9,46	123,00
	Empates	1(f)		
	Total	20		
SVM_rbf tex - SVM_rbf RD	Rangos negativos	12(g)	8,33	100,00
	Rangos positivos	8(h)	13,75	110,00
	Empates	0(i)		
	Total	20		

a SVM\_rbf unión RD < SVM\_rbf

b SVM\_rbf unión RD > SVM\_rbf

c SVM\_rbf unión RD = SVM\_rbf

d SVM\_rbf unión RD < SVM\_rbf tex

e SVM\_rbf unión RD > SVM\_rbf tex

f SVM\_rbf unión RD = SVM\_rbf tex

g SVM\_rbf tex < SVM\_rbf RD

h SVM\_rbf tex > SVM\_rbf RD

i SVM\_rbf tex = SVM\_rbf RD

Tabla 43. Estadísticos de contraste de la prueba de Wilcoxon para el clasificador SVM utilizando la Predictividad Negativa.

	SVM_rbf unión RD - SVM_rbf	SVM_rbf unión RD - SVM_rbf tex	SVM_rbf tex - SVM_rbf RD
Z	-,724(a)	-1,127(b)	-,187(b)
Sig. asintót. (bilateral)	,469	,260	,852

a Basado en los rangos positivos.

b Basado en los rangos negativos.

c Prueba de los rangos con signo de Wilcoxon

### Anexo III Resultados del clasificador knn

Tabla 44. Desempeño del clasificador KNN utilizando los rasgos geométricos, los rasgos de textura y la unión de ambos conjuntos de rasgos.

	Clasificador KNN			
	rasgos geométricos	rasgos de textura	unión de rasgos	
	sin RD	RD=7	RD=17	RD=17
<b><i>Er</i></b>	0,159	<b>0,150</b>	0,288	0,179
<b><i>Cr</i></b>	0,841	<b>0,850</b>	0,712	0,821
<b><i>Se</i></b>	0,916	<b>0,916</b>	0,841	0,907
<b><i>Sp</i></b>	0,699	<b>0,726</b>	0,470	0,660
<b><i>Pp</i></b>	0,851	<b>0,863</b>	0,749	0,834
<b><i>Pn</i></b>	0,818	<b>0,824</b>	0,612	0,789
<b><i>medida F</i></b>	0,792	<b>0,809</b>	0,600	0,762

Tabla 45. Prueba de Friedman realizada al clasificador KNN, utilizando la medida F.

	Rango promedio
C_knn	3,03
C_knn RD	3,98
C_knn tex	1,00
C_knn unión RD	2,00

Tabla 46. Estadísticos de contraste de la prueba de Friedman para el clasificador KNN utilizando la medida F.

N	20
Chi-cuadrado	59,714
gl	3
Sig. asintót.	,000

Tabla 47. Prueba de los rangos con signo de Wilcoxon para el clasificador KNN utilizando la Medida F.

		N	Rango promedio	Suma de rangos
C_knn RD - C_knn	Rangos negativos	0(a)	,00	,00
	Rangos positivos	19(b)	10,00	190,00
	Empates	1(c)		
	Total	20		
C_knn unión RD - C_knn	Rangos negativos	20(d)	10,50	210,00
	Rangos positivos	0(e)	,00	,00
	Empates	0(f)		
	Total	20		
C_knn unión RD - C_knn tex	Rangos negativos	0(g)	,00	,00
	Rangos positivos	20(h)	10,50	210,00
	Empates	0(i)		
	Total	20		

a C\_knn RD < C\_knn

b C\_knn RD > C\_knn

c C\_knn RD = C\_knn

d C\_knn unión RD < C\_knn

e C\_knn unión RD > C\_knn

f C\_knn unión RD = C\_knn

g C\_knn unión RD < C\_knn tex

h C\_knn unión RD > C\_knn tex

i C\_knn unión RD = C\_knn tex

Tabla 48. Estadísticos de contraste de la prueba de Wilcoxon para el clasificador KNN utilizando la Mediada F.

	C_knn RD - C_knn	C_knn unión RD - C_knn	C_knn unión RD - C_knn tex
Z	-3,823(a)	-3,920(b)	-3,920(a)
Sig. asintót. (bilateral)	,000	,000	,000

a Basado en los rangos negativos.

b Basado en los rangos positivos.

c Prueba de los rangos con signo de Wilcoxon

Tabla 49. Prueba de Friedman realizada al clasificador KNN, utilizando la Predictividad Negativa (PN).

	Rango promedio
C_knn	3,03
C_knn RD	3,98
C_knn tex	1,00
C_knn unión RD	2,00

Tabla 50. Estadísticos de contraste de la prueba de Friedman para el clasificador KNN utilizando la Predictividad Negativa (PN).

N	20
Chi-cuadrado	59,714
gl	3
Sig. asintót.	,000

Tabla 51. Prueba de los rangos con signo de Wilcoxon para el clasificador KNN utilizando la Predictividad Negativa (PN).

		N	Rango promedio	Suma de rangos
C_knn RD - C_knn	Rangos negativos	0(a)	,00	,00
	Rangos positivos	19(b)	10,00	190,00
	Empates	1(c)		
	Total	20		
C_knn unión RD - C_knn	Rangos negativos	20(d)	10,50	210,00
	Rangos positivos	0(e)	,00	,00
	Empates	0(f)		
	Total	20		
C_knn unión RD - C_knn tex	Rangos negativos	0(g)	,00	,00
	Rangos positivos	20(h)	10,50	210,00
	Empates	0(i)		
	Total	20		

a C\_knn RD < C\_knn

b C\_knn RD > C\_knn

c C\_knn RD = C\_knn

d C\_knn unión RD < C\_knn

e C\_knn unión RD > C\_knn

f C\_knn unión RD = C\_knn

g C\_knn unión RD < C\_knn tex

h C\_knn unión RD > C\_knn tex

i C\_knn unión RD = C\_knn tex

Tabla 52. Estadísticos de contraste de la prueba de Wilcoxon para el clasificador KNN utilizando la Predictividad Negativa (PN).

	C_knn RD - C_knn	C_knn unión RD - C_knn	C_knn unión RD - C_knn tex
Z	-3,823(a)	-3,920(b)	-3,920(a)
Sig. asintót. (bilateral)	,000	,000	,000

a Basado en los rangos negativos.

b Basado en los rangos positivos.

c Prueba de los rangos con signo de Wilcoxon



#### Anexo IV Índices de desempeño en las 20 corridas para todos los clasificadores

Tabla 53. Índices de desempeño para el clasificador de Mahalanobis, utilizando el conjunto de rasgos 1.

er	cr	se	sp	pp	pn	h_mean
0,071	0,929	1,000	0,800	0,900	1,000	0,889
0,103	0,897	0,947	0,800	0,900	0,889	0,867
0,093	0,907	0,929	0,867	0,929	0,867	0,897
0,071	0,929	0,946	0,895	0,946	0,895	0,920
0,057	0,943	0,957	0,917	0,957	0,917	0,936
0,060	0,940	0,964	0,897	0,946	0,929	0,929
0,061	0,939	0,953	0,912	0,953	0,912	0,932
0,080	0,920	0,945	0,872	0,932	0,895	0,907
0,079	0,921	0,939	0,886	0,939	0,886	0,912
0,079	0,921	0,945	0,878	0,935	0,896	0,910
0,072	0,928	0,950	0,887	0,941	0,904	0,917
0,065	0,935	0,955	0,897	0,946	0,912	0,925
0,077	0,923	0,933	0,905	0,949	0,877	0,919
0,082	0,918	0,930	0,897	0,944	0,871	0,913
0,081	0,919	0,927	0,904	0,948	0,868	0,915
0,080	0,920	0,925	0,910	0,951	0,866	0,917
0,076	0,924	0,929	0,916	0,954	0,874	0,922
0,071	0,929	0,933	0,920	0,956	0,880	0,927
0,071	0,929	0,931	0,925	0,958	0,878	0,928
0,068	0,932	0,934	0,929	0,960	0,883	0,931

Tabla 54. Índices de desempeño para el clasificador de Mahalanobis, utilizando el conjunto de rasgos 2.

er	cr	se	sp	pp	pn	h_mean
0,071	0,929	0,889	1,000	1,000	0,833	0,941
0,103	0,897	0,842	1,000	1,000	0,769	0,914
0,093	0,907	0,857	1,000	1,000	0,789	0,923
0,089	0,911	0,865	1,000	1,000	0,792	0,928
0,086	0,914	0,891	0,958	0,976	0,821	0,924
0,083	0,917	0,909	0,931	0,962	0,844	0,920
0,082	0,918	0,906	0,941	0,967	0,842	0,923
0,107	0,893	0,890	0,897	0,942	0,814	0,894
0,103	0,897	0,890	0,909	0,948	0,816	0,900
0,093	0,907	0,901	0,918	0,953	0,833	0,910
0,085	0,915	0,910	0,925	0,958	0,845	0,917
0,077	0,923	0,918	0,931	0,962	0,857	0,925
0,093	0,907	0,891	0,937	0,964	0,819	0,913
0,097	0,903	0,891	0,926	0,958	0,818	0,908
0,095	0,905	0,891	0,932	0,961	0,819	0,911
0,094	0,906	0,890	0,936	0,963	0,820	0,913
0,088	0,912	0,897	0,940	0,965	0,830	0,918
0,087	0,913	0,896	0,943	0,967	0,830	0,919
0,086	0,914	0,896	0,946	0,969	0,830	0,920
0,082	0,918	0,901	0,949	0,970	0,838	0,924

Tabla 55. Índices de desempeño para el clasificador de Mahalanobis, utilizando el conjunto de rasgos 3.

er	cr	se	sp	pp	pn	h_mean
0,357	0,643	0,556	0,800	0,833	0,500	0,656
0,310	0,690	0,632	0,800	0,857	0,533	0,706
0,256	0,744	0,679	0,867	0,905	0,591	0,761
0,268	0,732	0,676	0,842	0,893	0,571	0,750
0,257	0,743	0,717	0,792	0,868	0,594	0,753
0,274	0,726	0,691	0,793	0,864	0,575	0,738
0,245	0,755	0,719	0,824	0,885	0,609	0,768
0,250	0,750	0,712	0,821	0,881	0,604	0,763
0,262	0,738	0,695	0,818	0,877	0,590	0,752
0,279	0,721	0,681	0,796	0,861	0,574	0,734
0,281	0,719	0,680	0,792	0,861	0,568	0,732
0,280	0,720	0,682	0,793	0,862	0,568	0,733
0,269	0,731	0,697	0,794	0,865	0,581	0,742
0,265	0,735	0,703	0,794	0,865	0,587	0,746
0,267	0,733	0,715	0,767	0,852	0,589	0,740
0,259	0,741	0,719	0,782	0,861	0,598	0,749
0,256	0,744	0,723	0,783	0,862	0,602	0,752
0,258	0,742	0,732	0,761	0,851	0,604	0,746
0,248	0,752	0,740	0,774	0,859	0,615	0,757
0,250	0,750	0,747	0,755	0,850	0,617	0,751

Tabla 56. Índices de desempeño para el clasificador de Mahalanobis, utilizando el conjunto de rasgos 4.

er	cr	se	sp	pp	pn	h_mean
0,071	0,929	0,889	1,000	1,000	0,833	0,941
0,069	0,931	0,895	1,000	1,000	0,833	0,944
0,047	0,953	0,929	1,000	1,000	0,882	0,963
0,036	0,964	0,946	1,000	1,000	0,905	0,972
0,029	0,971	0,957	1,000	1,000	0,923	0,978
0,024	0,976	0,964	1,000	1,000	0,935	0,981
0,020	0,980	0,969	1,000	1,000	0,944	0,984
0,027	0,973	0,959	1,000	1,000	0,929	0,979
0,032	0,968	0,951	1,000	1,000	0,917	0,975
0,029	0,971	0,956	1,000	1,000	0,925	0,978
0,033	0,967	0,950	1,000	1,000	0,914	0,974
0,030	0,970	0,955	1,000	1,000	0,921	0,977
0,033	0,967	0,950	1,000	1,000	0,913	0,974
0,036	0,964	0,945	1,000	1,000	0,907	0,972
0,033	0,967	0,949	1,000	1,000	0,913	0,974
0,031	0,969	0,952	1,000	1,000	0,918	0,975
0,034	0,966	0,955	0,988	0,993	0,921	0,971
0,032	0,968	0,957	0,989	0,994	0,926	0,973
0,034	0,966	0,954	0,989	0,994	0,920	0,971
0,032	0,968	0,956	0,990	0,994	0,924	0,973

Tabla 57. Índices de desempeño para el clasificador SVM, utilizando el conjunto de rasgos 1.

er	cr	se	sp	pp	pn	h_mean
0,143	0,857	0,889	0,800	0,889	0,800	0,842
0,103	0,897	0,895	0,900	0,944	0,818	0,897
0,070	0,930	0,929	0,933	0,963	0,875	0,931
0,054	0,946	0,946	0,947	0,972	0,900	0,947
0,057	0,943	0,957	0,917	0,957	0,917	0,936
0,060	0,940	0,964	0,897	0,946	0,929	0,929
0,051	0,949	0,969	0,912	0,954	0,939	0,939
0,045	0,955	0,973	0,923	0,959	0,947	0,947
0,048	0,952	0,963	0,932	0,963	0,932	0,947
0,043	0,957	0,967	0,939	0,967	0,939	0,953
0,046	0,954	0,970	0,925	0,960	0,942	0,947
0,042	0,958	0,973	0,931	0,964	0,947	0,951
0,049	0,951	0,958	0,937	0,966	0,922	0,947
0,051	0,949	0,961	0,926	0,961	0,926	0,943
0,048	0,952	0,964	0,932	0,964	0,932	0,947
0,045	0,955	0,966	0,936	0,966	0,936	0,951
0,042	0,958	0,968	0,940	0,968	0,940	0,954
0,040	0,960	0,970	0,943	0,970	0,943	0,956
0,041	0,959	0,965	0,946	0,971	0,936	0,956
0,043	0,957	0,967	0,939	0,967	0,939	0,953

Tabla 58. Índices de desempeño para el clasificador SVM, utilizando el conjunto de rasgos 2.

er	cr	se	sp	pp	pn	h_mean
0,071	0,929	0,889	1,000	1,000	0,833	0,941
0,138	0,862	0,842	0,900	0,941	0,750	0,870
0,093	0,907	0,893	0,933	0,962	0,824	0,913
0,071	0,929	0,919	0,947	0,971	0,857	0,933
0,071	0,929	0,935	0,917	0,956	0,880	0,926
0,083	0,917	0,945	0,862	0,929	0,893	0,902
0,071	0,929	0,953	0,882	0,938	0,909	0,916
0,063	0,938	0,959	0,897	0,946	0,921	0,927
0,063	0,937	0,951	0,909	0,951	0,909	0,930
0,057	0,943	0,956	0,918	0,956	0,918	0,937
0,059	0,941	0,960	0,906	0,950	0,923	0,932
0,054	0,946	0,964	0,914	0,955	0,930	0,938
0,066	0,934	0,941	0,921	0,957	0,892	0,931
0,066	0,934	0,945	0,912	0,953	0,899	0,928
0,062	0,938	0,949	0,918	0,956	0,905	0,933
0,058	0,942	0,952	0,923	0,959	0,911	0,937
0,055	0,945	0,955	0,928	0,961	0,917	0,941
0,056	0,944	0,951	0,932	0,963	0,911	0,941
0,056	0,944	0,948	0,935	0,965	0,906	0,942
0,057	0,943	0,951	0,929	0,961	0,910	0,939

Tabla 59. Índices de desempeño para el clasificador SVM, utilizando el conjunto de rasgos 3.

er	cr	se	sp	pp	pn	h_mean
0,143	0,857	1,000	0,600	0,818	1,000	0,750
0,276	0,724	0,895	0,400	0,739	0,667	0,553
0,209	0,791	0,929	0,533	0,788	0,800	0,678
0,196	0,804	0,919	0,579	0,810	0,786	0,710
0,186	0,814	0,935	0,583	0,811	0,824	0,718
0,167	0,833	0,945	0,621	0,825	0,857	0,749
0,163	0,837	0,938	0,647	0,833	0,846	0,766
0,179	0,821	0,932	0,615	0,819	0,828	0,741
0,175	0,825	0,927	0,636	0,826	0,824	0,755
0,171	0,829	0,934	0,633	0,825	0,838	0,754
0,170	0,830	0,940	0,623	0,825	0,846	0,749
0,161	0,839	0,945	0,638	0,832	0,860	0,762
0,154	0,846	0,950	0,651	0,837	0,872	0,772
0,158	0,842	0,945	0,647	0,834	0,863	0,768
0,171	0,829	0,949	0,603	0,818	0,863	0,737
0,174	0,826	0,952	0,590	0,813	0,868	0,728
0,172	0,828	0,955	0,590	0,813	0,875	0,730
0,183	0,817	0,951	0,568	0,804	0,862	0,711
0,177	0,823	0,954	0,581	0,809	0,871	0,722
0,182	0,818	0,956	0,561	0,802	0,873	0,707

Tabla 60. Índices de desempeño para el clasificador SVM, utilizando el conjunto de rasgos 4.

er	cr	se	sp	pp	pn	h_mean
0,071	0,929	1,000	0,800	0,900	1,000	0,889
0,069	0,931	0,947	0,900	0,947	0,900	0,923
0,047	0,953	0,964	0,933	0,964	0,933	0,949
0,036	0,964	0,973	0,947	0,973	0,947	0,960
0,043	0,957	0,957	0,958	0,978	0,920	0,957
0,071	0,929	0,945	0,897	0,945	0,897	0,920
0,061	0,939	0,953	0,912	0,953	0,912	0,932
0,054	0,946	0,959	0,923	0,959	0,923	0,941
0,048	0,952	0,963	0,932	0,963	0,932	0,947
0,050	0,950	0,956	0,939	0,967	0,920	0,947
0,046	0,954	0,960	0,943	0,970	0,926	0,952
0,042	0,958	0,964	0,948	0,972	0,932	0,956
0,038	0,962	0,966	0,952	0,975	0,938	0,959
0,046	0,954	0,953	0,956	0,976	0,915	0,955
0,057	0,943	0,956	0,918	0,956	0,918	0,937
0,054	0,946	0,959	0,923	0,959	0,923	0,941
0,059	0,941	0,955	0,916	0,955	0,916	0,935
0,056	0,944	0,957	0,920	0,957	0,920	0,939
0,060	0,940	0,954	0,914	0,954	0,914	0,933
0,061	0,939	0,956	0,908	0,951	0,918	0,931



Tabla 61. Índices de desempeño para el clasificador lineal, utilizando el conjunto de rasgos 1.

er	cr	se	sp	pp	pn	h_mean
0,071	0,929	1,000	0,800	0,900	1,000	0,889
0,172	0,828	0,947	0,600	0,818	0,857	0,735
0,163	0,837	0,929	0,667	0,839	0,833	0,776
0,143	0,857	0,919	0,737	0,872	0,824	0,818
0,143	0,857	0,913	0,750	0,875	0,818	0,824
0,131	0,869	0,927	0,759	0,879	0,846	0,835
0,112	0,888	0,938	0,794	0,896	0,871	0,860
0,116	0,884	0,932	0,795	0,895	0,861	0,858
0,111	0,889	0,927	0,818	0,905	0,857	0,869
0,107	0,893	0,934	0,816	0,904	0,870	0,871
0,111	0,889	0,940	0,792	0,895	0,875	0,860
0,101	0,899	0,945	0,810	0,904	0,887	0,873
0,104	0,896	0,933	0,825	0,910	0,867	0,876
0,102	0,898	0,938	0,824	0,909	0,875	0,877
0,105	0,895	0,934	0,822	0,908	0,870	0,875
0,103	0,897	0,932	0,833	0,913	0,867	0,880
0,097	0,903	0,935	0,843	0,918	0,875	0,887
0,095	0,905	0,933	0,852	0,922	0,872	0,891
0,094	0,906	0,931	0,860	0,925	0,870	0,894
0,093	0,907	0,934	0,857	0,924	0,875	0,894

Tabla 62. Índices de desempeño para el clasificador lineal, utilizando el conjunto de rasgos 2.

er	cr	se	sp	pp	pn	h_mean
0,143	0,857	0,889	0,800	0,889	0,800	0,842
0,207	0,793	0,789	0,800	0,882	0,667	0,795
0,163	0,837	0,821	0,867	0,920	0,722	0,843
0,143	0,857	0,838	0,895	0,939	0,739	0,865
0,129	0,871	0,870	0,875	0,930	0,778	0,872
0,119	0,881	0,891	0,862	0,925	0,806	0,876
0,102	0,898	0,906	0,882	0,935	0,833	0,894
0,125	0,875	0,877	0,872	0,928	0,791	0,874
0,135	0,865	0,854	0,886	0,933	0,765	0,870
0,121	0,879	0,868	0,898	0,940	0,786	0,883
0,111	0,889	0,880	0,906	0,946	0,800	0,893
0,107	0,893	0,882	0,914	0,951	0,803	0,898
0,115	0,885	0,866	0,921	0,954	0,784	0,892
0,117	0,883	0,859	0,926	0,957	0,778	0,892
0,110	0,890	0,869	0,932	0,960	0,791	0,899
0,107	0,893	0,877	0,923	0,955	0,800	0,899
0,105	0,895	0,877	0,928	0,958	0,802	0,902
0,103	0,897	0,878	0,932	0,960	0,804	0,904
0,109	0,891	0,867	0,935	0,962	0,791	0,900
0,111	0,889	0,863	0,939	0,963	0,786	0,899

Tabla 63. Índices de desempeño para el clasificador lineal, utilizando el conjunto de rasgos 3.

er	cr	se	sp	pp	pn	h_mean
0,214	0,786	0,889	0,600	0,800	0,750	0,716
0,241	0,759	0,789	0,700	0,833	0,636	0,742
0,233	0,767	0,821	0,667	0,821	0,667	0,736
0,250	0,750	0,811	0,632	0,811	0,632	0,710
0,271	0,729	0,761	0,667	0,814	0,593	0,711
0,274	0,726	0,745	0,690	0,820	0,588	0,716
0,255	0,745	0,781	0,676	0,820	0,622	0,725
0,241	0,759	0,808	0,667	0,819	0,650	0,731
0,246	0,754	0,793	0,682	0,823	0,638	0,733
0,250	0,750	0,780	0,694	0,826	0,630	0,735
0,248	0,752	0,770	0,717	0,837	0,623	0,743
0,238	0,762	0,782	0,724	0,843	0,636	0,752
0,231	0,769	0,790	0,730	0,847	0,648	0,759
0,230	0,770	0,789	0,735	0,849	0,649	0,761
0,229	0,771	0,803	0,712	0,840	0,658	0,755
0,228	0,772	0,801	0,718	0,842	0,659	0,757
0,223	0,777	0,813	0,711	0,840	0,670	0,758
0,230	0,770	0,817	0,682	0,827	0,667	0,743
0,218	0,782	0,827	0,699	0,836	0,684	0,757
0,218	0,782	0,835	0,684	0,831	0,691	0,752

Tabla 64. Índices de desempeño para el clasificador lineal, utilizando el conjunto de rasgos 4.

er	cr	se	sp	pp	pn	h_mean
0,071	0,929	1,000	0,800	0,900	1,000	0,889
0,138	0,862	0,947	0,700	0,857	0,875	0,805
0,093	0,907	0,964	0,800	0,900	0,923	0,874
0,089	0,911	0,973	0,789	0,900	0,938	0,872
0,100	0,900	0,957	0,792	0,898	0,905	0,866
0,083	0,917	0,964	0,828	0,914	0,923	0,890
0,071	0,929	0,969	0,853	0,925	0,935	0,907
0,080	0,920	0,959	0,846	0,921	0,917	0,899
0,087	0,913	0,939	0,864	0,928	0,884	0,900
0,079	0,921	0,945	0,878	0,935	0,896	0,910
0,072	0,928	0,950	0,887	0,941	0,904	0,917
0,065	0,935	0,955	0,897	0,946	0,912	0,925
0,066	0,934	0,950	0,905	0,950	0,905	0,927
0,066	0,934	0,945	0,912	0,953	0,899	0,928
0,071	0,929	0,949	0,890	0,942	0,903	0,919
0,067	0,933	0,952	0,897	0,946	0,909	0,924
0,067	0,933	0,955	0,892	0,943	0,914	0,922
0,067	0,933	0,951	0,898	0,945	0,908	0,924
0,071	0,929	0,948	0,892	0,943	0,902	0,919
0,068	0,932	0,951	0,898	0,945	0,907	0,924

Tabla 65. Índices de desempeño para el clasificador KNN, utilizando el conjunto de rasgos 1.

er	cr	se	sp	pp	pn	h_mean
0,143	0,857	0,889	0,800	0,889	0,800	0,842
0,241	0,759	0,842	0,600	0,800	0,667	0,701
0,256	0,744	0,821	0,600	0,793	0,643	0,693
0,196	0,804	0,865	0,684	0,842	0,722	0,764
0,200	0,800	0,891	0,625	0,820	0,750	0,735
0,179	0,821	0,909	0,655	0,833	0,792	0,762
0,163	0,837	0,922	0,676	0,843	0,821	0,780
0,170	0,830	0,932	0,641	0,829	0,833	0,759
0,167	0,833	0,915	0,682	0,843	0,811	0,781
0,164	0,836	0,923	0,673	0,840	0,825	0,779
0,157	0,843	0,930	0,679	0,845	0,837	0,785
0,149	0,851	0,936	0,690	0,851	0,851	0,794
0,148	0,852	0,933	0,698	0,854	0,846	0,799
0,138	0,862	0,938	0,721	0,863	0,860	0,815
0,133	0,867	0,942	0,726	0,866	0,869	0,820
0,125	0,875	0,945	0,744	0,873	0,879	0,832
0,118	0,882	0,948	0,759	0,880	0,887	0,843
0,111	0,889	0,951	0,773	0,886	0,895	0,853
0,113	0,887	0,942	0,785	0,891	0,880	0,856
0,114	0,886	0,945	0,776	0,887	0,884	0,852

Tabla 66. Índices de desempeño para el clasificador KNN, utilizando el conjunto de rasgos 2.

er	cr	se	sp	pp	pn	h_mean
0,143	0,857	0,889	0,800	0,889	0,800	0,842
0,207	0,793	0,842	0,700	0,842	0,700	0,765
0,233	0,767	0,821	0,667	0,821	0,667	0,736
0,179	0,821	0,865	0,737	0,865	0,737	0,796
0,186	0,814	0,891	0,667	0,837	0,762	0,763
0,167	0,833	0,909	0,690	0,847	0,800	0,784
0,153	0,847	0,922	0,706	0,855	0,828	0,800
0,161	0,839	0,932	0,667	0,840	0,839	0,777
0,159	0,841	0,915	0,705	0,852	0,816	0,796
0,157	0,843	0,923	0,694	0,848	0,829	0,792
0,150	0,850	0,930	0,698	0,853	0,841	0,798
0,143	0,857	0,936	0,707	0,858	0,854	0,806
0,143	0,857	0,933	0,714	0,860	0,849	0,809
0,133	0,867	0,938	0,735	0,870	0,862	0,824
0,129	0,871	0,942	0,740	0,872	0,871	0,829
0,121	0,879	0,945	0,756	0,879	0,881	0,840
0,113	0,887	0,948	0,771	0,886	0,889	0,851
0,107	0,893	0,951	0,784	0,891	0,896	0,860
0,109	0,891	0,942	0,796	0,896	0,881	0,863
0,111	0,889	0,945	0,786	0,891	0,885	0,858

Tabla 67. Índices de desempeño para el clasificador KNN, utilizando el conjunto de rasgos 3.

er	cr	se	sp	pp	pn	h_mean
0,286	0,714	0,889	0,400	0,727	0,667	0,552
0,310	0,690	0,895	0,300	0,708	0,600	0,449
0,302	0,698	0,857	0,400	0,727	0,600	0,545
0,268	0,732	0,892	0,421	0,750	0,667	0,572
0,257	0,743	0,870	0,500	0,769	0,667	0,635
0,286	0,714	0,836	0,483	0,754	0,609	0,612
0,265	0,735	0,844	0,529	0,771	0,643	0,651
0,277	0,723	0,836	0,513	0,763	0,625	0,636
0,294	0,706	0,817	0,500	0,753	0,595	0,620
0,293	0,707	0,824	0,490	0,750	0,600	0,614
0,301	0,699	0,820	0,472	0,745	0,581	0,599
0,315	0,685	0,809	0,448	0,736	0,553	0,577
0,302	0,698	0,824	0,460	0,742	0,580	0,591
0,311	0,689	0,820	0,441	0,734	0,566	0,574
0,310	0,690	0,825	0,438	0,734	0,571	0,572
0,295	0,705	0,836	0,462	0,744	0,600	0,595
0,298	0,702	0,839	0,446	0,739	0,597	0,582
0,306	0,694	0,829	0,443	0,735	0,582	0,578
0,305	0,695	0,827	0,452	0,737	0,583	0,584
0,314	0,686	0,819	0,439	0,730	0,566	0,571

Tabla 68. Índices de desempeño para el clasificador KNN, utilizando el conjunto de rasgos 4.

er	cr	se	sp	pp	pn	h_mean
0,214	0,786	0,889	0,600	0,800	0,750	0,716
0,276	0,724	0,842	0,500	0,762	0,625	0,627
0,279	0,721	0,821	0,533	0,767	0,615	0,647
0,214	0,786	0,865	0,632	0,821	0,706	0,730
0,214	0,786	0,891	0,583	0,804	0,737	0,705
0,190	0,810	0,909	0,621	0,820	0,783	0,738
0,173	0,827	0,922	0,647	0,831	0,815	0,760
0,188	0,813	0,918	0,615	0,817	0,800	0,737
0,175	0,825	0,915	0,659	0,833	0,806	0,766
0,171	0,829	0,923	0,653	0,832	0,821	0,765
0,163	0,837	0,930	0,660	0,838	0,833	0,772
0,161	0,839	0,927	0,672	0,843	0,830	0,780
0,159	0,841	0,916	0,698	0,852	0,815	0,793
0,153	0,847	0,922	0,706	0,855	0,828	0,800
0,157	0,843	0,920	0,699	0,851	0,823	0,794
0,147	0,853	0,925	0,718	0,860	0,836	0,808
0,139	0,861	0,929	0,735	0,867	0,847	0,821
0,135	0,865	0,927	0,750	0,874	0,846	0,829
0,135	0,865	0,919	0,763	0,878	0,835	0,834
0,139	0,861	0,923	0,745	0,870	0,839	0,824

## Anexo V Programa Matlab para la clasificación de las clases 3 y 4

```

cont=0;
load classbi_R9_C2 % matriz de rasgos y nombres
nombres3y4=nombresbi_R9_C2;
rasgos3y4=matrixprueb_R9;
% A partir del vector nombres_R9, se crea un nuevo vector columna
'groups' que divide
% los datos en dos grupos para clasificarlos: 'normales' y 'anormales'
%%
% Se seleccionan ahora de forma aleatoria los conjuntos de prueba y de
% entrenamiento.
% indices = crossvalind('Kfold',nombres3y4,20);
load indices
posval={'anormal'};% Célula con algún tipo de anomalía.
negval={'normal'};% Celulas sin anomalía
%% SVM con KERNEL rbf == 2
% Inicializando cp, el objeto de evaluación del clasificador.
cp = classperf(nombres3y4,'Positive',posval,'Negative',negval);
for i = 1:20
    test = (indices == i); train = ~test;

```



```

svmStruct =
svmtrain(rasgos3y4(train,:),nombres3y4(train),'KERNEL_FUNCTION','rbf','RBF_Sigma',2); ...
    %utilizando kernel Gaussiano de valor 2(el mejor)
class = svmclassify(svmStruct,rasgos3y4(test,:));
classperf(cp,class,test);
er(i)=cp.ErrorRate;
se(i)=cp.Sensitivity;
cr(i)=cp.CorrectRate;
sp(i)=cp.Specificity;
pp(i)=cp.PositivePredictiveValue;
pn(i)=cp.NegativePredictiveValue;
h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media armónica
end
[ermu2,ersigma2]=normfit(er);
[semu2,sesigma2]=normfit(se);
[crmu2,crsigma2]=normfit(cr);
[spmu2,ppsigma2]=normfit(sp);
[ppmu2,ppsigma2]=normfit(pp);
[pnmu2,pnsigma2]=normfit(pn);
[h_meanmu2,h_meansigma2]=normfit(h_mean);
cont=cont+1;
sensibilidad{cont}=[se]';
pred_negativa{cont}=[pn]';
h_media{cont}=[h_mean]';
    nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'
    tabla_svm_rbf=[er' cr' se' sp' pp' pn' h_mean'];
    [f,c]=size(tabla_svm_rbf); %determino el tamaño de la matriz
    Tabla_svm_rbf=cell(f,c); %creo un array con el tamaño de la matriz
for i=1:f
    for j=1:c
        Tabla_svm_rbf{i,j}=tabla_svm_rbf(i,j);
    end
end
Tabla_SVM_rbf=cat(1,nombres,Tabla_svm_rbf); %uno los nombres y los datos
xlswrite('desempeño_C_3y4_svm_rbf.xls',Tabla_SVM_rbf)
%%
mu_svm_rbf=[ermu2 ersigma2; crmu2 crsigma2; semu2 sesigma2;...
    spmu2 ppsigma2; ppmu2 ppsigma2; pnmu2 pnsigma2;
    h_meanmu2,h_meansigma2];
% en >>mu_lineal<< queda una matriz que por las filas
% contiene er, cr, se, sp, pp, pn , h_mean y por las columnas
% la media y la desviación estándar de estos parámetros.
% Los guardo en un documento excel.
nombres={'media','sigma'};
    nombres2={' '; 'er'; 'cr'; 'se'; 'sp'; 'pp'; 'pn'; 'h_mean'};
[f,c]=size(mu_svm_rbf);
for i=1:f
    for j=1:c
        tabla_SVM_mu{i,j}=mu_svm_rbf(i,j);
    end
end
mu_RBF=cat(1,nombres,tabla_SVM_mu);
mu_svm_lineal=cat(2,nombres2,mu_RBF);
xlswrite('mu_clases3y4_svm_rbf.xls',mu_svm_lineal)
%% Clasificador LINEAL

```

```

cp = classperf(nombres3y4, 'Positive', posval, 'Negative', negval);
for i = 1:20
    test = (indices == i); train = ~test;
    class =
classify(rasgos3y4(test,:), rasgos3y4(train,:), nombres3y4(train,:));
    classperf(cp, class, test);
    er(i)=cp.ErrorRate;
    se(i)=cp.Sensitivity;
    cr(i)=cp.CorrectRate;
    sp(i)=cp.Specificity;
    pp(i)=cp.PositivePredictiveValue;
    pn(i)=cp.NegativePredictiveValue;
    h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
armónica
end
[ermu3, ersigma3]=normfit(er);
[semu3, sesigma3]=normfit(se);
[crmu3, crsigma3]=normfit(cr);
[spmu3, ppsigma3]=normfit(sp);
[ppmu3, ppsigma3]=normfit(pp);
[pnmu3, pnsigma3]=normfit(pn);
[h_meanmu3, h_meansigma3]=normfit(h_mean);
cont=cont+1;
sensibilidad{cont}=[se]';
pred_negativa{cont}=[pn]';
h_media{cont}=[h_mean]';
%%
    nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'
    tabla_lineal=[er' cr' se' sp' pp' pn' h_mean'];
    [f,c]=size(tabla_lineal); %determino el tamaño de la matriz
    tabla_Lineal=cell(f,c); %creo un array con el tamaño de la matriz
for i=1:f
    for j=1:c
        tabla_Lineal{i,j}=tabla_lineal(i,j);
    end
end
Tabla_lineal=cat(1,nombres,tabla_Lineal); %uno los nombres y los datos
xlswrite('desempeño_C_3y4_lineal.xls',Tabla_lineal)
%Se crea un documento excel, en su interior el contenido de >tabla_lineal
%%
mu_lineal=[ermu3 ersigma3;crmu3 crsigma3;semu3 sesigma3; spmu3
ppsigma3;...
ppmu3 ppsigma3;pnmu3 pnsigma3; h_meanmu3,h_meansigma3];
% en >>mu_lineal<< queda una matriz que por las filas
% contiene er, cr, se, sp, pp, pn , h_mean y por las columnas
% la media y la desviación estándar de estos parámetros.
% Los guardo en un documento excel.
nombres={'media', 'sigma'};
nombres2={' ' 'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};
[f,c]=size(mu_lineal);
for i=1:f
    for j=1:c
        tabla_Lineal_mu{i,j}=mu_lineal(i,j);
    end
end
mu_lineal=cat(1,nombres,tabla_Lineal_mu);
mu_lineal=cat(2,nombres2,mu_lineal);

```

```

xlswrite('mu_clases3y4_lineal.xls',mu_lineal)
%% Clasificador utilizando distancia de mahalanobis
cp = classperf(nombres3y4,'Positive',posval,'Negative',negval);
for i = 1:20
    test = (indices == i); train = ~test;
    class =
classify(rasgos3y4(test,:),rasgos3y4(train,:),nombres3y4(train,),'mahala
nobilis');
    classperf(cp,class,test);
    er(i)=cp.ErrorRate;
    se(i)=cp.Sensitivity;
    cr(i)=cp.CorrectRate;
    sp(i)=cp.Specificity;
    pp(i)=cp.PositivePredictiveValue;
    pn(i)=cp.NegativePredictiveValue;
    h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
armónica
end
[ermu4,ersigma4]=normfit(er);
[semu4,sesigma4]=normfit(se);
[crmu4,crsigma4]=normfit(cr);
[spmu4,ppsigma4]=normfit(sp);
[ppmu4,ppsigma4]=normfit(pp);
[pnmu4,pnsigma4]=normfit(pn);
[h_meanmu4,h_meansigma4]=normfit(h_mean);
cont=cont+1;
sensibilidad{cont}=[se]';
pred_negativa{cont}=[pn]';
h_media{cont}=[h_mean]';
%%
    tabla_maha=[er' cr' se' sp' pp' pn' h_mean'];
    nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};
    [f,c]=size(tabla_maha);
    tabla_Maha=cell(f,c);
for i=1:f
    for j=1:c
        tabla_Maha{i,j}=tabla_maha(i,j);
    end
end
Tabla_Maha=cat(1,nombres,tabla_Maha);
xlswrite('desempeño_C_3y4_maha.xls',Tabla_Maha)
%%
mu_maha=[ermu4 ersigma4; crmu4 crsigma4 ; semu4 sesigma4;...
    spmu4 ppsigma4;ppmu4 ppsigma4;pnmu4 pnsigma4;
h_meanmu4,h_meansigma4];
nombres={'media','sigma'};
nombres2={' '; 'er'; 'cr'; 'se'; 'sp'; 'pp'; 'pn'; 'h_mean'};
[f,c]=size(mu_maha);
for i=1:f
    for j=1:c
        tabla_Mahala_mu{i,j}=mu_maha(i,j);
    end
end
mu_Maha=cat(1,nombres,tabla_Mahala_mu);
mu_Mahal=cat(2,nombres2,mu_Maha);
xlswrite('mu_clases3y4_maha.xls',mu_Mahal)
%% k vecinos mas cercanos

```

```

cp = classperf(nombres3y4, 'Positive', posval, 'Negative', negval);
for i = 1:20
    test = (indices == i); train = ~test;
    class =
knnclassify(rasgos3y4(test,:), rasgos3y4(train,:), nombres3y4(train,:), 3, 'c
ityblock');
    classperf(cp, class, test);
    er(i)=cp.ErrorRate;
    se(i)=cp.Sensitivity;
    cr(i)=cp.CorrectRate;
    sp(i)=cp.Specificity;
    pp(i)=cp.PositivePredictiveValue;
    pn(i)=cp.NegativePredictiveValue;
    h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
armónica
end
[ermu5, ersigma5]=normfit(er);
[semu5, sesigma5]=normfit(se);
[crmu5, crsigma5]=normfit(cr);
[spmu5, ppsigma5]=normfit(sp);
[ppmu5, ppsigma5]=normfit(pp);
[pnmu5, pnsigma5]=normfit(pn);
[h_meanmu5, h_meansigma5]=normfit(h_mean);
cont=cont+1;
sensibilidad{cont}=[se]';
pred_negativa{cont}=[pn]';
h_media{cont}=[h_mean]';
%%
    tabla_Kveci=[er' cr' se' sp' pp' pn' h_mean]';
    nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};
    [f,c]=size(tabla_Kveci);
    tabla_Kvecinos=cell(f,c);
for i=1:f
    for j=1:c
        tabla_Kvecinos{i,j}=tabla_Kveci(i,j);
    end
end
Tabla_Kvecinos=cat(1,nombres,tabla_Kvecinos);
xlswrite('desempeño_C_3y4_Kvecinos.xls',Tabla_Kvecinos)
%%
mu_Kvec=[ermu5 ersigma5; crmu5 crsigma5; semu5 sesigma5;...
    spmu5 ppsigma5; ppmu5 ppsigma5; pnmu5 pnsigma5;
    h_meanmu5, h_meansigma5];
nombres={'media', 'sigma'};
nombres2={''; 'er'; 'cr'; 'se'; 'sp'; 'pp'; 'pn'; 'h_mean'};

[f,c]=size(mu_Kvec);
for i=1:f
    for j=1:c
        tabla_Kvecinos_mu{i,j}=mu_Kvec(i,j);
    end
end
mu_Kvec=cat(1,nombres,tabla_Kvecinos_mu);
mu_Kvec=cat(2,nombres2,mu_Kvec);
xlswrite('mu_clases_3y4_Kvecinos.xls',mu_Kvec)
%% verificando medias
medias_por_columnas= [mean(sensibilidad{1,1}) mean(sensibilidad{1,2})...]

```

```

    mean(sensibilidad{1,3}) mean(sensibilidad{1,4}))%
    disp('svm_rbf==2 cla_lineal mahalan k_veci')
%% Guardando los datos
% 1 SVM lineal
% 2 SVM rbf==2
% 3 clasificador lineal
% 4 distancia de mahalanobis
% 5 k vecinos mas cercanos
nombres={'SVM_rbf' 'C_LINEAL' 'C_MAHALANOBIS' 'C_knn'};
matrix=[]; %reservo espacio para la matriz que luego convertiré en *.xls
for b=1:length(sensibilidad); %convirtiendo del array a matriz double
para poder guardar
    matrix(:,b)=sensibilidad{1,b};
end
%convirtiendo 'matrix' en un array de (fxc)
% hacemos esto para poder guardar en un excel y poder
% ponerle en la primera fila los nombres de los clasificadores
[f,c]=size(matrix);
sensi=cell(f,c);
for i=1:f
    for j=1:c
        sensi{i,j}=matrix(i,j);
    end
end
%exportando para el excel
todo1=cat(1,nombres,sensi);
sensibilidad_exel= xlswrite('Sensibilidad_C_3y4.xls',todo1);
%convirtiendo a *.xls
%% lo mismo para la predictividad negativa
matrix2=[]; %reservo espacio para la matriz que luego convertiré en *.xls
for b=1:length(pred_negativa); %convirtiendo del array a matriz double
para poder guardar
    matrix2(:,b)=pred_negativa{1,b};
end
% hacemos esto para poder guardar en un excel y poder
% ponerle en la primera fila los nombres de los clasificadores
[f,c]=size(matrix2);
Pre_Nega=cell(f,c);
for i=1:f
    for j=1:c
        Pre_Nega{i,j}=matrix2(i,j);
    end
end
todo2=cat(1,nombres,Pre_Nega); %exportando para el excel
predictividad_negativa_exel=
xlswrite('predictividad_negativa_C_3y4.xls',todo2); %convirtiendo a *.xls
%% guardando los datos de H_MEDIA en excel
matrix3=[]; %reservo espacio para la matriz que luego convertiré en *.xls
for b=1:length(h_media); %convirtiendo del array a matriz double para
para poder guardar
    matrix3(:,b)=h_media{1,b};
end
%convirtiendo 'matrix' en un array de (fxc)
% hacemos esto para poder guardar en un excel y poder
% ponerle en la primera fila los nombres de los clasificadores
[f,c]=size(matrix3);
H_MEDIA=cell(f,c);

```

```

for i=1:f
    for j=1:c
        H_MEDIA{i,j}=matrix3(i,j);
    end
end
% exportando para el excel
todo3=cat(1,nombres,H_MEDIA);
H_medi= xlswrite('H_MEDIA_C_3y4.xls',todo3); %convirtiendo a *.xls
%% Rasgos Geométricos con reducción de la dimensionalidad (RD=7) PCA
%% SVM con KERNEL rbf == 2
cp = classperf(nombres3y4,'Positive',posval,'Negative',negval);
for i = 1:20
    test = (indices == i); train = ~test;
    options=[];
options.ReducedDim=7;
[eigvector,eigvalue] = PCA(rasgos3y4(train,:),options);
measpcal = rasgos3y4(train,:)*eigvector;
[eigvector2,eigvalue2] = PCA(rasgos3y4(train,:),options);
measpcal2 = rasgos3y4(test,:)*eigvector2;
svmStruct =
svmtrain(measpcal,nombres3y4(train),'KERNEL_FUNCTION','rbf','RBF_Sigma',2
);
    class = svmclassify(svmStruct,measpcal2);
    classperf(cp,class,test);
    er(i)=cp.ErrorRate;
    se(i)=cp.Sensitivity;
    cr(i)=cp.CorrectRate;
    sp(i)=cp.Specificity;
    pp(i)=cp.PositivePredictiveValue;
    pn(i)=cp.NegativePredictiveValue;
    h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
armónica
end
[ermu2,ersigma2]=normfit(er);
[semu2,sesigma2]=normfit(se);
[crmu2,crsigma2]=normfit(cr);
[spmu2,ppsigma2]=normfit(sp);
[ppmu2,ppsigma2]=normfit(pp);
[pnmu2,pnsigma2]=normfit(pn);
[h_meanmu2,h_meansigma2]=normfit(h_mean);
cont=cont+1;
sensibilidad{cont}=[se]';
pred_negativa{cont}=[pn]';
h_media{cont}=[h_mean]';
%%
    nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};%creo un array
    tabla_svm_rbf=[er' cr' se' sp' pp' pn' h_mean'];
    [f,c]=size(tabla_svm_rbf); %determino el tamaño de la matriz
>tabla_lineal<
    Tabla_svm_rbf=cell(f,c); %creo un array con el tamaño de la matriz
>tabla_lineal<
for i=1:f
    for j=1:c
        Tabla_svm_rbf{i,j}=tabla_svm_rbf(i,j);
    end
end
Tabla_SVM_rbf=cat(1,nombres,Tabla_svm_rbf); %uno los nombres y los datos

```

```

    xlswrite('desempeño_C_3y4_svm_rbf_RedDimen.xls', Tabla_SVM_rbf)
%%
mu_svm_rbf=[ermu2 ersigma2; crmu2 crsigma2; semu2 sesigma2;...
    spmu2 ppsigma2; ppmu2 ppsigma2; pnmu2 pnsigma2;
    h_meanmu2, h_meansigma2];
nombres={'media', 'sigma'};
nombres2={'er'; 'cr'; 'se'; 'sp'; 'pp'; 'pn'; 'h_mean'};
[f,c]=size(mu_svm_rbf);
for i=1:f
    for j=1:c
        tabla_SVM_mu{i,j}=mu_svm_rbf(i,j);
    end
end
mu_RBF=cat(1,nombres,tabla_SVM_mu);
mu_svm_lineal=cat(2,nombres2,mu_RBF);
xlswrite('mu_clases3y4_svm_rbf_RedDimen.xls',mu_svm_lineal)
%% Clasificador LINEAL
cp = classperf(nombres3y4,'Positive',posval,'Negative',negval);
for i = 1:20
    test = (indices == i); train = ~test;
    options=[];
    options.ReducedDim=7;
    [eigvector,eigvalue] = PCA(rasgos3y4(train,:),options);
    measpcal = rasgos3y4(train,:)*eigvector;
    [eigvector2,eigvalue2] = PCA(rasgos3y4(train,:),options);
    measpca2 = rasgos3y4(test,:)*eigvector2;
    class = classify(measpca2,measpcal,nombres3y4(train,:));
    classperf(cp,class,test);
    er(i)=cp.ErrorRate;
    se(i)=cp.Sensitivity;
    cr(i)=cp.CorrectRate;
    sp(i)=cp.Specificity;
    pp(i)=cp.PositivePredictiveValue;
    pn(i)=cp.NegativePredictiveValue;
    h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
    armónica
end
[ermu3,ersigma3]=normfit(er);
[semu3,sesigma3]=normfit(se);
[crmu3,crsigma3]=normfit(cr);
[spmu3,ppsigma3]=normfit(sp);
[ppmu3,ppsigma3]=normfit(pp);
[pnmu3,pnsigma3]=normfit(pn);
[h_meanmu3,h_meansigma3]=normfit(h_mean);
cont=cont+1;
sensibilidad{cont}=[se]';
pred_negativa{cont}=[pn]';
h_media{cont}=[h_mean]';
%%
nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};%creo un array
tabla_lineal=[er' cr' se' sp' pp' pn' h_mean]';
[f,c]=size(tabla_lineal); %determino el tamaño de la matriz
>tabla_lineal<
    tabla_Lineal=cell(f,c); %creo un array con el tamaño de la matriz
>tabla_lineal<
for i=1:f
    for j=1:c

```

```

        tabla_Lineal{i,j}=tabla_lineal(i,j);
end
end
Tabla_lineal=cat(1,nombres,tabla_Lineal); %uno los nombres y los datos
xlswrite('desempeño_C_3y4_lineal_RedDimen.xls',Tabla_lineal)
%%
mu_lineal=[ermu3 ersigma3;crmu3 crsigma3;semu3 sesigma3; spmu3
ppsigma3;...
ppmu3 ppsigma3;pnmu3 pnsigma3; h_meanmu3,h_meansigma3];
nombres={'media','sigma'};
nombres2={' '; 'er'; 'cr'; 'se'; 'sp'; 'pp'; 'pn'; 'h_mean'};
[f,c]=size(mu_lineal);
for i=1:f
    for j=1:c
        tabla_Lineal_mu{i,j}=mu_lineal(i,j);
    end
end
mu_lineal=cat(1,nombres,tabla_Lineal_mu);
mu_lineal=cat(2,nombres2,mu_lineal);
xlswrite('mu_clases3y4_lineal_RedDimen.xls',mu_lineal)
%% Clasificador de Mahalanobis
cp = classperf(nombres3y4,'Positive',posval,'Negative',negval);
for i = 1:20
    test = (indices == i); train = ~test;
    test = (indices == i); train = ~test;
    options=[];
    options.ReducedDim=7;
    [eigvector,eigvalue] = PCA(rasgos3y4(train,:),options);
    measpcal = rasgos3y4(train,:)*eigvector;
    [eigvector2,eigvalue2] = PCA(rasgos3y4(train,:),options);
    measpcal2 = rasgos3y4(test,:)*eigvector2;
    class =
    classify(measpcal2,measpcal,nombres3y4(train,:), 'mahalanobis');
    classperf(cp,class,test);
    er(i)=cp.ErrorRate;
    se(i)=cp.Sensitivity;
    cr(i)=cp.CorrectRate;
    sp(i)=cp.Specificity;
    pp(i)=cp.PositivePredictiveValue;
    pn(i)=cp.NegativePredictiveValue;
    h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
armónica
end
[ermu4,ersigma4]=normfit(er);
[semu4,sesigma4]=normfit(se);
[crmu4,crsigma4]=normfit(cr);
[spmu4,ppsigma4]=normfit(sp);
[ppmu4,ppsigma4]=normfit(pp);
[pnmu4,pnsigma4]=normfit(pn);
[h_meanmu4,h_meansigma4]=normfit(h_mean);
cont=cont+1;
sensibilidad{cont}=[se]';
pred_negativa{cont}=[pn]';
h_media{cont}=[h_mean]';

%%
nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};%creo un array

```



```

    tabla_maha=[er' cr' se' sp' pp' pn' h_mean'];
    [f,c]=size(tabla_maha); %determino el tamaño de la matriz
>tabla_lineal<
    tabla_Maha=cell(f,c); %creo un array con el tamaño de la matriz
>tabla_lineal<
for i=1:f
    for j=1:c
        tabla_Maha{i,j}=tabla_maha(i,j);
    end
end
Tabla_Maha=cat(1,nombres,tabla_Maha); %uno los nombres y los datos
    xlswrite('desempeño_C_3y4_Maha_RedDimen.xls',Tabla_Maha)
%%
mu_maha=[ermu4 ersigma4;crmu4 crsigma4;semu4 sesigma4; spmu4 ppsigma4;...
    ppmu4 ppsigma4;pnmu4 pnsigma4; h_meanmu4,h_meansigma4];
nombres={'media','sigma'};
    nombres2={'er';'cr';'se';'sp';'pp';'pn';'h_mean'};
[f,c]=size(mu_maha);
for i=1:f
    for j=1:c
        tabla_maha_mu{i,j}=mu_maha(i,j);
    end
end
mu_Mahal=cat(1,nombres,tabla_maha_mu);
mu_Mahal1=cat(2,nombres2,mu_Mahal);
xlswrite('mu_clases3y4_maha_RedDimen.xls',mu_Mahal1)
%% k vecinos mas cercanos
cp = classperf(nombres3y4,'Positive',posval,'Negative',negval);
tic
for i = 1:20
    test = (indices == i); train = ~test;
    options=[];
options.ReducedDim=7;
[eigvector,eigvalue] = PCA(rasgos3y4(train,:),options);
measpca1 = rasgos3y4(train,:)*eigvector;
[eigvector2,eigvalue2] = PCA(rasgos3y4(train,:),options);
measpca2 = rasgos3y4(test,:)*eigvector2;
    class =
knnclassify(measpca2,measpca1,nombres3y4(train,:),3,'cityblock');
    classperf(cp,class,test);
    er(i)=cp.ErrorRate;
    se(i)=cp.Sensitivity;
    cr(i)=cp.CorrectRate;
    sp(i)=cp.Specificity;
    pp(i)=cp.PositivePredictiveValue;
    pn(i)=cp.NegativePredictiveValue;
    h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
armónica
end
knn=toc;
[ermu5,ersigma5]=normfit(er);
[semu5,sesigma5]=normfit(se);
[crmu5,crsigma5]=normfit(cr);
[spmu5,ppsigma5]=normfit(sp);
[ppmu5,ppsigma5]=normfit(pp);
[pnmu5,pnsigma5]=normfit(pn);
[h_meanmu5,h_meansigma5]=normfit(h_mean);

```

```

cont=cont+1;
sensibilidad{cont}=[se]';
pred_negativa{cont}=[pn]';
h_medio{cont}=[h_mean]';
%%
    tabla_Kveci=[er' cr' se' sp' pp' pn' h_mean'];
    nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};
    [f,c]=size(tabla_Kveci);
    tabla_Kvecinos=cell(f,c);
for i=1:f
    for j=1:c
        tabla_Kvecinos{i,j}=tabla_Kveci(i,j);
    end
end
Tabla_Kvecinos=cat(1,nombres,tabla_Kvecinos);
xlswrite('desempeño_C_3y4_Kvecinos_RedDimen.xls',Tabla_Kvecinos)
%%
mu_Kvec=[ermu5 ersigma5; crmu5 crsigma5 ; semu5 sesigma5;...
    spmu5 ppsigma5;ppmu5 ppsigma5;pnmu5 pnsigma5;
h_meanmu5,h_meansigma5];
nombres={'media','sigma'};
nombres2={' ','er';'cr';'se';'sp';'pp';'pn';'h_mean'};

[f,c]=size(mu_Kvec);
for i=1:f
    for j=1:c
        tabla_Kvecinos_mu{i,j}=mu_Kvec(i,j);
    end
end
mu_Kvec=cat(1,nombres,tabla_Kvecinos_mu);
mu_Kvec=cat(2,nombres2,mu_Kvec);
xlswrite('mu_clases_3y4_Kvecinos_RedDimen.xls',mu_Kvec)
%% verificando medias
medias_por_columnas= [mean(sensibilidad{1,5}) mean(sensibilidad{1,6})...
    mean(sensibilidad{1,7}) mean(sensibilidad{1,8})]%
mean(sensibilidad{1,9})]
disp('svm_rbf==2 cla_lineal mahalanobis k_nn')
%% Guardando los datos
nombres={'SVM_rbf' 'C_LINEAL' 'Mahalanobis' 'C_knn'};
j=1;
matrix=[]; %reservo espacio para la matriz que luego convertiré en *.xls
for b=5:length(sensibilidad); %convirtiendo del array a matriz double
    para poder guardar
        matrix(:,j)=sensibilidad{1,b};
        j=j+1;
    end
end
[f,c]=size(matrix);
sensi=cell(f,c);
for i=1:f
    for j=1:c
        sensi{i,j}=matrix(i,j);
    end
end
todol=cat(1,nombres,sensi);
sensibilidad_exel= xlswrite('Sensibilidad_C_3y4_RedDimen.xls',todol);
%convirtiendo a *.xls
%% lo mismo para la predictividad negativa

```

---

```

matrix2=[]; %reservo espacio para la matriz que luego convertiré en *.xls
j=1;
for b=5:length(pred_negativa); %convirtiendo del array a matriz double
para poder guardar
    matrix2(:,j)=pred_negativa{1,b};
    j=j+1;
end
[f,c]=size(matrix2);
Pre_Nega=cell(f,c);
for i=1:f
    for j=1:c
        Pre_Nega{i,j}=matrix2(i,j);
    end
end
todo2=cat(1,nombres,Pre_Nega);
predictividad_negativa_excel=
xlswrite('predictividad_negativa_C_3y4_RedDimen.xls',todo2);
%convirtiendo a *.xls
%% guardando los datos de H_MEDIA en excel
matrix3=[]; %reservo espacio para la matriz que luego convertiré en *.xls
j=1;
for b=5:length(h_media); %convirtiendo del array a matriz double para
poder guardar
    matrix3(:,j)=h_media{1,b};
    j=j+1;
end
[f,c]=size(matrix3);
H_MEDIA=cell(f,c);
for i=1:f
    for j=1:c
        H_MEDIA{i,j}=matrix3(i,j);
    end
end
% exportando para el excel
todo3=cat(1,nombres,H_MEDIA);
H_media= xlswrite('H_MEDIA_C_3y4_RedDimen',todo3); %convirtiendo a *.xls
%% Utilizando los rasgos de TEXTURA
% reducción de la dimensionalidad (RD = 17) PCA
cont=0;
load unio_max_of1
rasgos3y4=unio_max_of1;
%% SVM con KERNEL rbf == 2
cp = classperf(nombres3y4,'Positive',posval,'Negative',negval);
for i = 1:20
    test = (indices == i); train = ~test;
    options=[];
    options.ReducedDim=17;
    [eigvector,eigvalue] = PCA(rasgos3y4(train,:),options);
    measpca1 = rasgos3y4(train,:)*eigvector;
    [eigvector2,eigvalue2] = PCA(rasgos3y4(train,:),options);
    measpca2 = rasgos3y4(test,:)*eigvector2;
    svmStruct =
    svmtrain(measpca1,nombres3y4(train),'KERNEL_FUNCTION','rbf','RBF_Sigma',2
); ...
    %utilizando kernel Gaussiano de valor 2(el mejor)
    class = svmclassify(svmStruct,measpca2);
    classperf(cp,class,test);

```

```

    er(i)=cp.ErrorRate;
    se(i)=cp.Sensitivity;
    cr(i)=cp.CorrectRate;
    sp(i)=cp.Specificity;
    pp(i)=cp.PositivePredictiveValue;
    pn(i)=cp.NegativePredictiveValue;
    h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
armónica
end
[ermu2,ersigma2]=normfit(er);
[semu2,sesigma2]=normfit(se);
[crmu2,crsigma2]=normfit(cr);
[spmu2,ppsigma2]=normfit(sp);
[ppmu2,ppsigma2]=normfit(pp);
[pnmu2,pnsigma2]=normfit(pn);
[h_meanmu2,h_meansigma2]=normfit(h_mean);
cont=cont+1;
sensibilidad1{cont}=[se]';
pred_negativa1{cont}=[pn]';
h_medial{cont}=[h_mean]';
%%
    nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};
    tabla_svm_rbf=[er' cr' se' sp' pp' pn' h_mean'];
    [f,c]=size(tabla_svm_rbf);
    Tabla_svm_rbf=cell(f,c);
for i=1:f
    for j=1:c
        Tabla_svm_rbf{i,j}=tabla_svm_rbf(i,j);
    end
end
Tabla_SVM_rbf=cat(1,nombres,Tabla_svm_rbf);
xlswrite('desempeño_C_3y4_svm_rbf_RedDimen_TEXURA.xls',Tabla_SVM_rbf)
%%
mu_svm_rbf=[ermu2 ersigma2; crmu2 crsigma2; semu2 sesigma2;...
    spmu2 ppsigma2; ppmu2 ppsigma2; pnmu2 pnsigma2;
    h_meanmu2,h_meansigma2];
nombres={'media','sigma'};
nombres2={' ','er'; 'cr'; 'se'; 'sp'; 'pp'; 'pn'; 'h_mean'};
[f,c]=size(mu_svm_rbf);
for i=1:f
    for j=1:c
        tabla_SVM_mu{i,j}=mu_svm_rbf(i,j);
    end
end
mu_RBF=cat(1,nombres,tabla_SVM_mu);
mu_svm_lineal=cat(2,nombres2,mu_RBF);
xlswrite('mu_clases3y4_svm_rbf_RedDimen_TEXURA.xls',mu_svm_lineal)
%% Clasificador LINEAL
cp = classperf(nombres3y4,'Positive',posval,'Negative',negval);
for i = 1:20
    test = (indices == i); train = ~test;
    options=[];
    options.ReducedDim=17;
    [eigvector,eigvalue] = PCA(rasgos3y4(train,:),options);
    measpcal = rasgos3y4(train,:)*eigvector;
    [eigvector2,eigvalue2] = PCA(rasgos3y4(train,:),options);
    measpcal2 = rasgos3y4(test,:)*eigvector2;

```

```

class = classify(measpc2,measpc1,nombres3y4(train,:));
classperf(cp,class,test);
er(i)=cp.ErrorRate;
se(i)=cp.Sensitivity;
cr(i)=cp.CorrectRate;
sp(i)=cp.Specificity;
pp(i)=cp.PositivePredictiveValue;
pn(i)=cp.NegativePredictiveValue;
h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
armónica
end
[ermu3,ersigma3]=normfit(er);
[semu3,sesigma3]=normfit(se);
[crmu3,crsigma3]=normfit(cr);
[spmu3,ppsigma3]=normfit(sp);
[ppmu3,ppsigma3]=normfit(pp);
[pnmu3,pnsigma3]=normfit(pn);
[h_meanmu3,h_meansigma3]=normfit(h_mean);
cont=cont+1;
sensibilidad1{cont}=[se]';
pred_negativa1{cont}=[pn]';
h_medial{cont}=[h_mean]';
%%
nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};
tabla_lineal=[er' cr' se' sp' pp' pn' h_mean'];
[f,c]=size(tabla_lineal); %determino el tamaño de la matriz
>tabla_lineal<
tabla_Lineal=cell(f,c);
for i=1:f
    for j=1:c
        tabla_Lineal{i,j}=tabla_lineal(i,j);
    end
end
Tabla_lineal=cat(1,nombres,tabla_Lineal); %uno los nombres y los datos
xlswrite('desempeño_C_3y4_lineal_RedDimen_TEXURA.xls',Tabla_lineal)
%%
mu_lineal=[ermu3 ersigma3;crmu3 crsigma3;semu3 sesigma3; spmu3
ppsigma3;...
ppmu3 ppsigma3;pnmu3 pnsigma3; h_meanmu3,h_meansigma3];
nombres={'media','sigma'};
nombres2={' ' ;'er' ;'cr' ;'se' ;'sp' ;'pp' ;'pn' ;'h_mean'};
[f,c]=size(mu_lineal);
for i=1:f
    for j=1:c
        tabla_Lineal_mu{i,j}=mu_lineal(i,j);
    end
end
mu_lineal=cat(1,nombres,tabla_Lineal_mu);
mu_lineal=cat(2,nombres2,mu_lineal);
xlswrite('mu_clases3y4_lineal_RedDimen_TEXURA.xls',mu_lineal)
%% clasificador Mahalanobis
cp = classperf(nombres3y4,'Positive',posval,'Negative',negval);
for i = 1:20
    test = (indices == i); train = ~test;
    test = (indices == i); train = ~test;
    options=[];
options.ReducedDim=17;

```

```

[eigvector,eigvalue] = PCA(rasgos3y4(train,:),options);
measpcal = rasgos3y4(train,:)*eigvector;
[eigvector2,eigvalue2] = PCA(rasgos3y4(train,:),options);
measpc2 = rasgos3y4(test,:)*eigvector2;
class =
classify(measpc2,measpcal,nombres3y4(train,:), 'mahalanobis');
classperf(cp,class,test);
er(i)=cp.ErrorRate;
se(i)=cp.Sensitivity;
cr(i)=cp.CorrectRate;
sp(i)=cp.Specificity;
pp(i)=cp.PositivePredictiveValue;
pn(i)=cp.NegativePredictiveValue;
h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
armónica
end
[ermul,ersignal]=normfit(er);
[semul,sesignal]=normfit(se);
[crmul,crsignal]=normfit(cr);
[spmul,ppsignal]=normfit(sp);
[ppmul,ppsignal]=normfit(pp);
[pnmul,pnsignal]=normfit(pn);
[h_meanmul,h_meansignal]=normfit(h_mean);
cont=cont+1;
sensibilidad1{cont}=[se]';
pred_negativa1{cont}=[pn]';
h_medial{cont}=[h_mean]';

%%
nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};%creo un array
tabla_maha=[er' cr' se' sp' pp' pn' h_mean'];
[f,c]=size(tabla_maha); %determino el tamaño de la matriz
>tabla_lineal<
tabla_Maha=cell(f,c); %creo un array con el tamaño de la matriz
>tabla_lineal<
for i=1:f
    for j=1:c
        tabla_Maha{i,j}=tabla_maha(i,j);
    end
end
Tabla_Maha=cat(1,nombres,tabla_Maha); %uno los nombres y los datos
xlswrite('desempeño_C_3y4_Maha_RedDimen_tex.xls',Tabla_Maha)

%%
mu_maha=[ermul ersignal;crmul crsignal;semul sesignal; spmul ppsignal;...
ppmul ppsignal;pnmul pnsignal; h_meanmul,h_meansignal];
nombres={'media','sigma'};
nombres2={' ' 'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};
[f,c]=size(mu_maha);
for i=1:f
    for j=1:c
        tabla_maha_mu{i,j}=mu_maha(i,j);
    end
end
mu_Mahal=cat(1,nombres,tabla_maha_mu);
mu_Mahal1=cat(2,nombres2,mu_Mahal);
xlswrite('mu_clases3y4_maha_RedDimen_tex.xls',mu_Mahal1)
%% k vecinos mas cercanos

```

---

```

% cp = classperf(nombres3y4);
cp = classperf(nombres3y4, 'Positive', posval, 'Negative', negval);
for i = 1:20
    test = (indices == i); train = ~test;
    options=[];
options.ReducedDim=17;
[eigvector,eigvalue] = PCA(rasgos3y4(train,:),options);
measpcal = rasgos3y4(train,:)*eigvector;
[eigvector2,eigvalue2] = PCA(rasgos3y4(train,:),options);
measpc2 = rasgos3y4(test,:)*eigvector2;
class =
knnclassify(measpc2,measpcal,nombres3y4(train,:),3,'cityblock');
classperf(cp,class,test);
er(i)=cp.ErrorRate;
se(i)=cp.Sensitivity;
cr(i)=cp.CorrectRate;
sp(i)=cp.Specificity;
pp(i)=cp.PositivePredictiveValue;
pn(i)=cp.NegativePredictiveValue;
h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
armónica
end
[ermu5,ersigma5]=normfit(er);
[semu5,sesigma5]=normfit(se);
[crmu5,crsigma5]=normfit(cr);
[spmu5,ppsigma5]=normfit(sp);
[ppmu5,ppsigma5]=normfit(pp);
[pnmu5,pnsigma5]=normfit(pn);
[h_meanmu5,h_meansigma5]=normfit(h_mean);
cont=cont+1;
sensibilidad1{cont}=[se]';
pred_negativa1{cont}=[pn]';
h_medial{cont}=[h_mean]';
%%
    tabla_Kveci=[er' cr' se' sp' pp' pn' h_mean'];
    nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};
    [f,c]=size(tabla_Kveci);
    tabla_Kvecinos=cell(f,c);
for i=1:f
    for j=1:c
        tabla_Kvecinos{i,j}=tabla_Kveci(i,j);
    end
end
Tabla_Kvecinos=cat(1,nombres,tabla_Kvecinos);

xlswrite('desempeño_C_3y4_Kvecinos_RedDimen_TEXURA.xls',Tabla_Kvecinos)
%%
mu_Kvec=[ermu5 ersigma5; crmu5 crsigma5 ; semu5 sesigma5;...
    spmu5 ppsigma5;ppmu5 ppsigma5;pnmu5 pnsigma5;
    h_meanmu5,h_meansigma5];
nombres={'media','sigma'};
    nombres2={' ' 'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};
    [f,c]=size(mu_Kvec);
for i=1:f
    for j=1:c
        tabla_Kvecinos_mu{i,j}=mu_Kvec(i,j);
    end
end

```

```

end
mu_Kvec=cat(1,nombres,tabla_Kvecinos_mu);
mu_Kvec=cat(2,nombres2,mu_Kvec);
xlswrite('mu_clases_3y4_Kvecinos_RedDimen_TEXURA.xls',mu_Kvec)
%% verificando medias
medias_por_columnas= [mean(sensibilidad1{1,1})
mean(sensibilidad1{1,2})...
mean(sensibilidad1{1,3}) mean(sensibilidad1{1,4})]%
mean(sensibilidad1{1,4})]
disp('svm_rbf==2 cla_lineal Mahalanobis k_veci')
%% Guardando los datos
nombres={'SVM_rbf' 'C_LINEAL' 'Mahalanobis' 'C_knn'};
j=1;
matrix=[]; %reservo espacio para la matriz que luego convertiré en *.xls
for b=1:length(sensibilidad1); %convirtiendo del array a matriz double
para poder guardar
matrix(:,j)=sensibilidad1{1,b};
j=j+1;
end
[f,c]=size(matrix);
sensi=cell(f,c);
for i=1:f
for j=1:c
sensi{i,j}=matrix(i,j);
end
end
%exportando para el excel
todo1=cat(1,nombres,sensi);
sensibilidad_exel=
xlswrite('Sensibilidad_C_3y4_RedDimen_TEXURA.xls',todo1); %convirtiendo a
*.xls
%% lo mismo para la predictividad negativa
matrix2=[]; %reservo espacio para la matriz que luego convertiré en *.xls
j=1;
for b=1:length(pred_negativa1); %convirtiendo del array a matriz double
para poder guardar
matrix2(:,j)=pred_negativa1{1,b};
j=j+1;
end
[f,c]=size(matrix2);
Pre_Nega=cell(f,c);
for i=1:f
for j=1:c
Pre_Nega{i,j}=matrix2(i,j);
end
end
todo2=cat(1,nombres,Pre_Nega);
predictividad_negativa_exel=
xlswrite('predictividad_negativa_C_3y4_RedDimen_TEXURA.xls',todo2);
%convirtiendo a *.xls
%% guardando los datos de H_MEDIA en excel
matrix3=[];
j=1;
for b=1:length(h_medial1); %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
matrix3(:,j)=h_medial1{1,b};
j=j+1;
end

```



```

[f,c]=size(matrix3);
H_MEDIA=cell(f,c);
for i=1:f
    for j=1:c
        H_MEDIA{i,j}=matrix3(i,j);
    end
end
todo3=cat(1,nombres,H_MEDIA);
H_medi= xlswrite('H_MEDIA_C_3y4_RedDimen_TEXURA.xls',todo3);
%convirtiendo a *.xls
%% uniendo los dos conjuntos de rasgos el geométrico y el de textura
%con reducción de dimensionalidad RD=17
load GyT_max_of1
cont=0;
rasgos3y4=GyT_max_of1;
%% SVM con KERNEL rbf == 2
% cp = classperf(nombres3y4); % Inicializando cp, el objeto de evaluación
del clasificador.
cp = classperf(nombres3y4,'Positive',posval,'Negative',negval);
tic
for i = 1:20
    test = (indices == i); train = ~test;
    options=[];
options.ReducedDim=17;
[eigvector,eigvalue] = PCA(rasgos3y4(train,:),options);
measpca1 = rasgos3y4(train,:)*eigvector;
[eigvector2,eigvalue2] = PCA(rasgos3y4(train,:),options);
measpca2 = rasgos3y4(test,:)*eigvector2;
svmStruct =
svmtrain(measpca1,nombres3y4(train),'KERNEL_FUNCTION','rbf','RBF_Sigma',2
); ...
    %utilizando kernel Gaussiano de valor 2(el mejor)
    class = svmclassify(svmStruct,measpca2);
    classperf(cp,class,test);
    er(i)=cp.ErrorRate;
    se(i)=cp.Sensitivity;
    cr(i)=cp.CorrectRate;
    sp(i)=cp.Specificity;
    pp(i)=cp.PositivePredictiveValue;
    pn(i)=cp.NegativePredictiveValue;
    h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
armónica
end
SVM=toc;
[ermu2,ersigma2]=normfit(er);
[semu2,sesigma2]=normfit(se);
[crmu2,crsigma2]=normfit(cr);
[spmu2,ppsigma2]=normfit(sp);
[ppmu2,ppsigma2]=normfit(pp);
[pnmu2,pnsigma2]=normfit(pn);
[h_meanmu2,h_meansigma2]=normfit(h_mean);
cont=cont+1;
sensibilidad2{cont}=[se]';
pred_negativa2{cont}=[pn]';
h_media2{cont}=[h_mean]';
%%
nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};%creo un array

```

```

    tabla_svm_rbf=[er' cr' se' sp' pp' pn' h_mean'];
    [f,c]=size(tabla_svm_rbf); %determino el tamaño de la matriz
>tabla_lineal<
    Tabla_svm_rbf=cell(f,c); %creo un array con el tamaño de la matriz
>tabla_lineal<
for i=1:f
    for j=1:c
        Tabla_svm_rbf{i,j}=tabla_svm_rbf(i,j);
    end
end
Tabla_SVM_rbf=cat(1,nombres,Tabla_svm_rbf); %uno los nombres y los datos
xlswrite('desempeño_C_3y4_svm_rbf_RedDimenGyT.xls',Tabla_SVM_rbf)
%%
mu_svm_rbf=[ermu2 ersigma2; crmu2 crsigma2; semu2 sesigma2;...
    spmu2 ppsigma2; ppmu2 ppsigma2; pnmu2 pnsigma2;
    h_meanmu2,h_meansigma2];
nombres={'media','sigma'};
nombres2={' '; 'er'; 'cr'; 'se'; 'sp'; 'pp'; 'pn'; 'h_mean'};
[f,c]=size(mu_svm_rbf);
for i=1:f
    for j=1:c
        tabla_SVM_mu{i,j}=mu_svm_rbf(i,j);
    end
end
mu_RBF=cat(1,nombres,tabla_SVM_mu);
mu_svm_lineal=cat(2,nombres2,mu_RBF);
xlswrite('mu_clases3y4_svm_rbf_RedDimenGyT.xls',mu_svm_lineal)
%% Clasificador LINEAL
cp = classperf(nombres3y4, 'Positive', posval, 'Negative', negval);
tic
for i = 1:20
    test = (indices == i); train = ~test;
    options=[];
options.ReducedDim=17;
[eigvector,eigvalue] = PCA(rasgos3y4(train,:),options);
measpcal = rasgos3y4(train,:)*eigvector;
[eigvector2,eigvalue2] = PCA(rasgos3y4(train,:),options);
measpcal2 = rasgos3y4(test,:)*eigvector2;
    class = classify(measpcal2,measpcal,nombres3y4(train,:));
    classperf(cp,class,test);
    er(i)=cp.ErrorRate;
    se(i)=cp.Sensitivity;
    cr(i)=cp.CorrectRate;
    sp(i)=cp.Specificity;
    pp(i)=cp.PositivePredictiveValue;
    pn(i)=cp.NegativePredictiveValue;
    h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
armónica
end
lineal=toc;
[ermu3,ersigma3]=normfit(er);
[semu3,sesigma3]=normfit(se);
[crmu3,crsigma3]=normfit(cr);
[spmu3,ppsigma3]=normfit(sp);
[ppmu3,ppsigma3]=normfit(pp);
[pnmu3,pnsigma3]=normfit(pn);
[h_meanmu3,h_meansigma3]=normfit(h_mean);

```

```

cont=cont+1;
sensibilidad2{cont}=[se]';
pred_negativa2{cont}=[pn]';
h_medio2{cont}=[h_mean]';
%%
    nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};
    tabla_lineal=[er' cr' se' sp' pp' pn' h_mean'];
    [f,c]=size(tabla_lineal); %determino el tamaño de la matriz
>tabla_lineal<
    tabla_Lineal=cell(f,c); %creo un array con el tamaño de la matriz
>tabla_lineal<
for i=1:f
    for j=1:c
        tabla_Lineal{i,j}=tabla_lineal(i,j);
    end
end
Tabla_lineal=cat(1,nombres,tabla_Lineal); %uno los nombres y los datos
xlswrite('desempeño_C_3y4_lineal_RedDimenGyT.xls',Tabla_lineal)
%%
mu_lineal=[ermu3 ersigma3; crmu3 crsigma3; semu3 sesigma3; spmu3
ppsigma3; ...
    ppmu3 ppsigma3; pnmu3 pnsigma3; h_meanmu3,h_meansigma3];
nombres={'media','sigma'};
nombres2={' ' 'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};
[f,c]=size(mu_lineal);
for i=1:f
    for j=1:c
        tabla_Lineal_mu{i,j}=mu_lineal(i,j);
    end
end
mu_lineal=cat(1,nombres,tabla_Lineal_mu);
mu_lineal=cat(2,nombres2,mu_lineal);
xlswrite('mu_clases3y4_lineal_RedDimenGyT.xls',mu_lineal)
%% clasificador Mahalanobis
cp = classperf(nombres3y4,'Positive',posval,'Negative',negval);
tic
for i = 1:20
    test = (indices == i); train = ~test;
    test = (indices == i); train = ~test;
    options=[];
options.ReducedDim=17;
[eigvector,eigvalue] = PCA(rasgos3y4(train,:),options);
measpcal = rasgos3y4(train,:)*eigvector;
[eigvector2,eigvalue2] = PCA(rasgos3y4(train,:),options);
measpcal2 = rasgos3y4(test,:)*eigvector2;
class =
classify(measpcal2,measpcal,nombres3y4(train,:), 'mahalanobis');
classperf(cp,class,test);
er(i)=cp.ErrorRate;
se(i)=cp.Sensitivity;
cr(i)=cp.CorrectRate;
sp(i)=cp.Specificity;
pp(i)=cp.PositivePredictiveValue;
pn(i)=cp.NegativePredictiveValue;
h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
armónica
end

```

```

mahala=toc;
[ermul,ersignal]=normfit(er);
[semul,sesignal]=normfit(se);
[crmul,crsignal]=normfit(cr);
[spmul,ppsignal]=normfit(sp);
[ppmul,ppsignal]=normfit(pp);
[pnmul,pnsignal]=normfit(pn);
[h_meanmul,h_meansignal]=normfit(h_mean);
cont=cont+1;
sensibilidad2{cont}=[se]';
pred_negativa2{cont}=[pn]';
h_media2{cont}=[h_mean]';

%%
nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};%creo un array
tabla_maha=[er' cr' se' sp' pp' pn' h_mean]';
[f,c]=size(tabla_maha); %determino el tamaño de la matriz
>tabla_lineal<
tabla_Maha=cell(f,c); %creo un array con el tamaño de la matriz
>tabla_lineal<
for i=1:f
    for j=1:c
        tabla_Maha{i,j}=tabla_maha(i,j);
    end
end
Tabla_Maha=cat(1,nombres,tabla_Maha); %uno los nombres y los datos
xlswrite('desempeño_C_3y4_Maha_unión.xls',Tabla_Maha)

%%
mu_maha=[ermul ersignal;crmul crsignal;semul sesignal; spmul ppsignal;...
ppmul ppsignal;pnmul pnsignal; h_meanmul,h_meansignal];
nombres={'media','sigma'};
nombres2={' ' 'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};
[f,c]=size(mu_maha);
for i=1:f
    for j=1:c
        tabla_maha_mu{i,j}=mu_maha(i,j);
    end
end
mu_Mahal=cat(1,nombres,tabla_maha_mu);
mu_Mahal1=cat(2,nombres2,mu_Mahal);
xlswrite('mu_clases3y4_maha_unión.xls',mu_Mahal1)
%% k vecinos mas cercanos
cp = classperf(nombres3y4, 'Positive', posval, 'Negative', negval);
for i = 1:20
    test = (indices == i); train = ~test;
    options=[];
    options.ReducedDim=17;
    [eigvector,eigvalue] = PCA(rasgos3y4(train,:),options);
    measpcal = rasgos3y4(train,:)*eigvector;
    [eigvector2,eigvalue2] = PCA(rasgos3y4(train,:),options);
    measpca2 = rasgos3y4(test,:)*eigvector2;
    class =
    knnclassify(measpca2,measpcal,nombres3y4(train,:),3,'cityblock');
    classperf(cp,class,test);
    er(i)=cp.ErrorRate;
    se(i)=cp.Sensitivity;
    cr(i)=cp.CorrectRate;
end

```

```

        sp(i)=cp.Specificity;
        pp(i)=cp.PositivePredictiveValue;
        pn(i)=cp.NegativePredictiveValue;
        h_mean(i)=2/((cp.Sensitivity)^(-1)+(cp.Specificity)^(-1)); %media
armónica
end
[ermu5,ersigma5]=normfit(er);
[semu5,sesigma5]=normfit(se);
[crmu5,crsigma5]=normfit(cr);
[spmu5,ppsigma5]=normfit(sp);
[ppmu5,ppsigma5]=normfit(pp);
[pnm5,pnsigma5]=normfit(pn);
[h_meanmu5,h_meansigma5]=normfit(h_mean);
cont=cont+1;
sensibilidad2{cont}=[se]';
pred_negativa2{cont}=[pn]';
h_media2{cont}=[h_mean]';
%%
        tabla_Kveci=[er' cr' se' sp' pp' pn' h_mean'];
        nombres={'er' 'cr' 'se' 'sp' 'pp' 'pn' 'h_mean'};
        [f,c]=size(tabla_Kveci);
        tabla_Kvecinos=cell(f,c);
for i=1:f
    for j=1:c
        tabla_Kvecinos{i,j}=tabla_Kveci(i,j);
    end
end
Tabla_Kvecinos=cat(1,nombres,tabla_Kvecinos);
xlswrite('desempeño_C_3y4_Kvecinos_RedDimenGyT.xls',Tabla_Kvecinos)
%%
mu_Kvec=[ermu5 ersigma5; crmu5 crsigma5 ; semu5 sesigma5;...
        spmu5 ppsigma5;ppmu5 ppsigma5;pnm5 pnsigma5;
        h_meanmu5,h_meansigma5];
nombres={'media','sigma'};
nombres2={'','er';'cr';'se';'sp';'pp';'pn';'h_mean'};
[f,c]=size(mu_Kvec);
for i=1:f
    for j=1:c
        tabla_Kvecinos_mu{i,j}=mu_Kvec(i,j);
    end
end
mu_Kvec=cat(1,nombres,tabla_Kvecinos_mu);
mu_Kvec=cat(2,nombres2,mu_Kvec);
xlswrite('mu_clases_3y4_Kvecinos_RedDimenGyT.xls',mu_Kvec)
%% verificando medias
medias_por_columnas= [mean(sensibilidad2{1,1})
mean(sensibilidad2{1,2})...
mean(sensibilidad2{1,3}) mean(sensibilidad2{1,4})]
mean(sensibilidad2{1,4})]
disp('svm_rbf==2 cla_lineal Mahalanobis k_veci')
%% Guardando los datos
nombres={'SVM_rbf' 'C_LINEAL' 'Mahalanobis' 'C_knn'};
j=1;
matrix=[]; %reservo espacio para la matriz que luego convertiré en *.xls
for b=1:length(sensibilidad2); %convirtiendo del array a matriz double
para poder guardar
    matrix(:,j)=sensibilidad2{1,b};

```

```

        j=j+1;
    end
    [f,c]=size(matrix);
    sensi=cell(f,c);
    for i=1:f
        for j=1:c
            sensi{i,j}=matrix(i,j);
        end
    end
    todo1=cat(1,nombres,sensi);
    sensibilidad_exel= xlswrite('Sensibilidad_C_3y4_RedDimenGyT.xls',todo1);
    %convirtiendo a *.xls

    %% lo mismo para la predictividad negativa
    matrix2=[]; %reservo espacio para la matriz que luego convertiré en *.xls
    j=1;
    for b=1:length(pred_negativa2); %convirtiendo del array a matriz double
        para poder guardar
            matrix2(:,j)=pred_negativa2{1,b};
            j=j+1;
        end
    end
    [f,c]=size(matrix2);
    Pre_Nega=cell(f,c);
    for i=1:f
        for j=1:c
            Pre_Nega{i,j}=matrix2(i,j);
        end
    end
    todo2=cat(1,nombres,Pre_Nega);
    predictividad_negativa_exel=
    xlswrite('predictividad_negativa_C_3y4_RedDimenGyT.xls',todo2);
    %convirtiendo a *.xls
    %% guardando los datos de H_MEDIA en excel
    matrix3=[]; %reservo espacio para la matriz que luego convertiré en *.xls
    j=1;
    for b=1:length(h_media2); %convirtiendo del array a matriz double para
        poder guardar
            matrix3(:,j)=h_media2{1,b};
            j=j+1;
        end
    end
    [f,c]=size(matrix3);
    H_MEDIA=cell(f,c);
    for i=1:f
        for j=1:c
            H_MEDIA{i,j}=matrix3(i,j);
        end
    end
    todo3=cat(1,nombres,H_MEDIA);
    H_medi= xlswrite('H_MEDIA_C_3y4_RedDimenGyT.xls',todo3); %convirtiendo a
    *.xls
    toc

```