

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Electrónica
Licenciatura en Electrónica



**Metodología para Segmentación de Cariotipos
Utilizando Campos Aleatorios de Markov**

TESIS

**Que para obtener el título de:
Licenciado en Electrónica**

PRESENTA:

C. Abraham Gilberto Díaz Nayotl

Directores de Tesis: **Dra. María Monserrat Morín Castillo (FCE - BUAP)**
Dr. José Rubén Conde Sánchez (FCFM - BUAP)
Dra. Gladys Denisse Salgado Suárez (B5M-BUAP)

Junio 2023. Puebla, Pue. Méx.

Agradecimientos

A mis padres y hermanos por el apoyo incondicional, cariño y palabras de aliento.

Al Dr. José Rubén Conde Sánchez, Dra. María Monserrat Morín Castillo y Dra. Gladys Denisse Salgado Suárez por su apoyo, conocimiento, orientación y confianza en el desarrollo de este trabajo de tesis.

A la Dra. Laura Daniel Mora, Dr. Marcelino Taxis Taxis y Mtro. Ricardo Álvarez González por el seguimiento y el apoyo durante el proceso del presente trabajo.

A la Facultad de Ciencias de la Electrónica y profesores por contribuir a mi formación académica.

A mi novia y amigos por su confianza y apoyo.

Dedicatoria

A mi familia.

Índice general

Resumen	3
Introducción	5
1. Conceptos Básicos	7
1.1. Antecedentes	7
1.2. Cromosomas	7
1.3. Cariotipo	9
1.3.1. Ejemplos de anomalías cromosómicas	11
1.4. Procesamiento digital de imágenes y visión por computadora	14
1.4.1. Definiciones	16
2. Planteamiento del problema	19
2.1. Preguntas de investigación	21
2.2. Objetivos	21
2.2.1. Objetivo general	21
2.2.2. Objetivos particulares	21
2.3. Hipótesis	21
2.4. Metodología	22
2.5. Motivación y justificación	22
3. Marco teórico	25
3.1. Sistemas de vecindades y cliques	25
3.2. Problemas de etiquetado	27
3.2.1. Etiquetas	28
3.3. Transformaciones realizadas a una imagen	29
3.3.1. Transformaciones individuales	29
3.3.2. Transformaciones de vecindad	29
3.3.3. Operaciones de vecindad	29
3.3.4. Operadores de extracción de bordes	30
3.4. Segmentación de imágenes	35
3.5. Campos aleatorios de Markov	35
3.6. Pynq Z2	39
3.6.1. Tarjeta de desarrollo Pynq Z2	39

3.6.2. Especificaciones de la tarjeta	40
4. Desarrollo y resultados	43
4.1. Pseudo código CAM	43
4.2. Pruebas con imágenes conocidas	46
4.3. Implementación algoritmo en FPGA Pynq Z2	48
4.4. Resultados	51
Conclusiones	57
Bibliografía	59
A. Código	63

Índice de figuras

1.1. Estructura de un cromosoma.	8
1.2. Clasificación morfológica de los cromosomas humanos.	8
1.3. Microfotografías de los cromosomas.	10
1.4. Comparación de cromosoma 11.	11
1.5. Cariotipo.	11
1.6. Cariotipo humano: a) Cariotipo humano sano, b) Cariotipo con una trisomía en el par 21.	12
1.7. Cariotipo humano: a) Cariotipo humano sano, b) Cariotipo con una trisomía en el par 13.	13
1.8. Cariotipo humano: a) Cariotipo humano sano, b) Cariotipo con ausencia de cromosoma en el par sexual.	14
1.9. Etapas en el procesamiento de imágenes.	15
1.10. Partes de una imagen.	17
2.1. Pasos para la obtención de un cariotipo, cultivo celular.	19
2.2. Pasos para la obtención de un cariotipo, ordenamiento.	20
2.3. Diagrama de la metodología propuesta.	23
3.1. Vecindades. a) Sistema de vecinos de primer orden, b) Sistemas de vecinos de segundo orden, c) Sistemas de vecinos de orden $n = 1, 2, \dots, 5$	27
3.2. Cliques de primer y segundo orden.	28
3.3. Vecindad cuatro.	30
3.4. Comparación: a) Fotografía cromosomas, b) Filtro Prewitt.	32
3.5. Comparación: a) Fotografía cromosoma, b) Filtro Prewitt.	32
3.6. Comparación: a) Fotografía cromosomas, b) Filtro de Sobel.	33
3.7. Comparación: a) Fotografía cromosoma, b) Filtro Sobel.	33
3.8. Cromosoma humano: a) Fotografía de cromosomas, b) Aplicando filtro Frei-Chen.	34
3.9. Cromosoma humano: a) Fotografía cromosoma, b) Aplicando filtro Frei-Chen.	35
3.10. Tarjeta de desarrollo Pynq Z2.	39
3.11. El SoC tiene la parte PS y la parte PL.	40

4.1. Aplicación CAM a una pseudo figura.	46
4.2. Aplicación CAM a un cromosoma.	47
4.3. Aplicación CAM a círculos de colores.	48
4.4. Inicialización sistema operativo Pynq Z2, leds encendidos.	49
4.5. Conexión Pynq-Z2 a computadora.	49
4.6. Entorno de trabajo Pynq Z2.	50
4.7. Algoritmo implementado en la Pynq Z2.	50
4.8. Aplicación CAM a un Cariotipo antes de ser ordenado.	51
4.9. Aplicación CAM a un cromosoma en baja resolución.	52
4.10. Aplicación CAM a un par de cromosomas. Imágenes tomadas [30].	53
4.11. Aplicación CAM a cromosomas. Imágenes de [30].	54
4.12. Aplicación CAM a cromosomas.	55
4.13. Aplicación CAM a cromosomas.	56

Resumen

La Citogenética es una rama de la genética que se enfoca en el estudio de los cromosomas. Para aquellos que trabajan en esta disciplina, el análisis de los cromosomas humanos es especialmente valioso, ya que permite detectar enfermedades cromosómicas en etapas tempranas de la vida: como en fetos, recién nacidos o niñas/os pequeñas/os. Algunas de las enfermedades detectables mediante este análisis incluyen el síndrome de Down, síndrome de Turner, síndrome de Patau, labio leporino, entre otros. Una de las técnicas más empleadas en el análisis es a través del estudio del cariotipo, que se basa en el ordenamiento y clasificación de los cromosomas. A menudo, a pesar de contar con algunos laboratorios especializados, el estudio del cariotipo se lleva a cabo de forma manual. Esta metodología involucra la obtención de muestras, la impresión de las imágenes de los cromosomas y su organización por comparación, apoyándose de un ideograma. De esta forma, se recortan los cromosomas, en forma de pares, se ordenan según su tamaño y forma, como si se tratara de un rompecabezas. Generalmente, de hacerse por los métodos tradicionales, el proceso de análisis es tardado llevando días en completarse, pues las imágenes obtenidas no son como se esperan o contienen mucho ruido, lo que puede complicar el análisis. Existen técnicas de segmentación de imágenes, que permiten un mejoramiento de estas imágenes aplicado en técnicas de delimitación de bordes. Este trabajo de tesis contribuye en el desarrollo de una metodología de delimitación de cromosomas a partir de imágenes, la cual se realiza a través de la segmentación de cromosomas empleando los Campos Aleatorios de Markov. La delimitación de los cromosomas sienta las bases para el ordenamiento del cariotipo.

Palabras Clave: Cariotipo, Segmentación, Campos Aleatorios de Markov.

Introducción

La citogenética se encarga del estudio, organización y distribución del material genético en las células, así como de la identificación de alteraciones cromosómicas que puedan tener implicaciones en la salud y por ende asociadas a ciertas enfermedades [10]. Este campo de la genética tiene aplicaciones importantes en el diagnóstico y tratamiento de enfermedades genéticas, el análisis forense y la investigación en biología evolutiva y de la reproducción.

El estudio de los cromosomas en humanos es de gran utilidad, pues permite diagnosticar enfermedades genéticas en un feto, un bebé o un niño pequeño, los médicos pueden identificar algunos trastornos genéticos como: síndrome de Down, síndrome de Turner, síndrome de Patau, labio leporino, entre otros. Existen varias técnicas para el estudio de los cromosomas [29], a saber:

- Estudio cromosómico de bandeado extendido, el cual consiste en el estudio de los cromosomas utilizando una resolución más alta que la del análisis cromosómico estándar, ahí se observa que los cromosomas están dispuestos de manera tal que se alargan, por lo que se pueden observar más bandas, lo que permite detectar anomalías cromosómicas estructurales más pequeñas, que generalmente no pueden ser observadas en estudios usuales.
- Hibridación fluorescente in situ (por sus siglas en inglés FISH), esta técnica consiste en determinar cuántas copias de un segmento específico de ADN existen en una célula, identificar cromosomas con estructuras anómalas,
- Análisis de microarreglo cromosómico (CMA, por sus siglas en inglés). El CMA es un análisis que se utiliza para detectar desequilibrios cromosómicos a una resolución mayor que las técnicas cromosómicas estándar actuales o las técnicas FISH y el cariotipo que consiste en teñir cromosomas para poder observarlos con un microscopio. Cuando se los tiñe, los cromosomas parecen tiras con "bandas" claras y oscuras.
- Cariotipo, el cual es una de las técnicas más empleadas, debido a las condiciones y simplicidad para realizarlo, consiste en obtener y clasificar los 46 cromosomas humanos a partir de microfotografías por medio de un microscopio electrónico.

La presente tesis se centra en el estudio de la citogenética y la aplicación de técnicas de procesamiento de imágenes para la segmentación y análisis de cromosomas. Se espera que los resultados obtenidos contribuyan al avance y comprensión de este campo de la genética y puedan ser aplicados en el diagnóstico y tratamiento de enfermedades genéticas.

La investigación se estructura de la siguiente manera:

En Capítulo 1 se presentan los conceptos básicos necesarios para comprender el desarrollo de este trabajo, incluyendo algunos métodos de procesamiento de imágenes.

El Capítulo 2 aborda el planteamiento del problema, los objetivos que se persiguen, la pregunta e hipótesis de investigación, así como la justificación del estudio.

El Capítulo 3 del presente trabajo proporciona los conocimientos teóricos necesarios para comprender y desarrollar el algoritmo. Se abordan los campos aleatorios de Markov, el algoritmo *K-means* y se destaca el papel de la tarjeta de desarrollo Pynq Z2 en el proceso de implementación.

En el Capítulo 4 se describe el desarrollo, validación de los resultados, la implementación del algoritmo en la tarjeta de desarrollo Pynq Z2 y los resultados obtenidos de la misma.

Finalmente, se presentarán las conclusiones derivadas de este trabajo de investigación.

Capítulo 1

Conceptos Básicos

1.1. Antecedentes

Las primeras aplicaciones de análisis digital en Biología se enfocaron en el estudio de los cromosomas humanos y en la cuantificación de su densidad a través de microscopios o microfotografías. El análisis del cariotipo fue una de las primeras pruebas en las que se emplearon técnicas de análisis por computadora, y el estudio de los cromosomas humanos fue una de las primeras aplicaciones asistidas por computadora, donde se capturaban directamente a través del microscopio o de microfotografías [17]. En la segmentación de los cromosomas se delimita su forma y tamaño a partir de microfotografías, esto aporta información sobre determinar si presenta malformaciones, carencia o exceso de cromosomas. Esto permite a los genetistas o médicos establecer un tratamiento y realizar pruebas más profundas únicamente a quienes lo necesiten, evitando costos innecesarios.

En este capítulo se revisarán conceptos preliminares para la comprensión del tema de cariotipo y el procesamiento digital de imágenes y visión por computadora.

1.2. Cromosomas

Los cromosomas son estructuras que se encuentran en el interior de las células y contienen la información genética de un organismo. Están compuestos de ADN y proteínas. Cada cromosoma se compone de dos brazos llamados telómeros y un centro denominado centrómero. El brazo superior se identifica como "p" y el brazo inferior como "q". A su vez, cada brazo se divide en diferentes regiones, y estas regiones se subdividen en bandas distintas [26]. La estructura general de un cromosoma se ilustra en la Figura 1.1.

El ser humano cuenta con 46 cromosomas, los cuales se agrupan en pares y su vez se

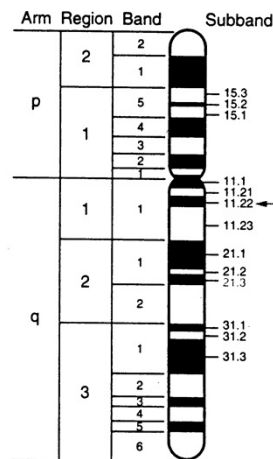


Figura 1.1: Estructura de un cromosoma.

clasifican en grupos alfabéticos, desde la letra A a la G. Además, se pueden ordenar según su tamaño, de mayor a menor. Para facilitar el análisis de los cromosomas, es importante tener conocimiento de su morfología. A continuación, se describen las características de los cromosomas:

- Metacéntricos: donde los telómeros p y q son del mismo tamaño.
- Submetacéntricos: los telómeros p son más pequeños que los telómeros q .
- Acrocéntricos: donde casi no existe el telómero p .

En la Figura 1.2 muestra la clasificación morfológica de los cromosomas humanos.

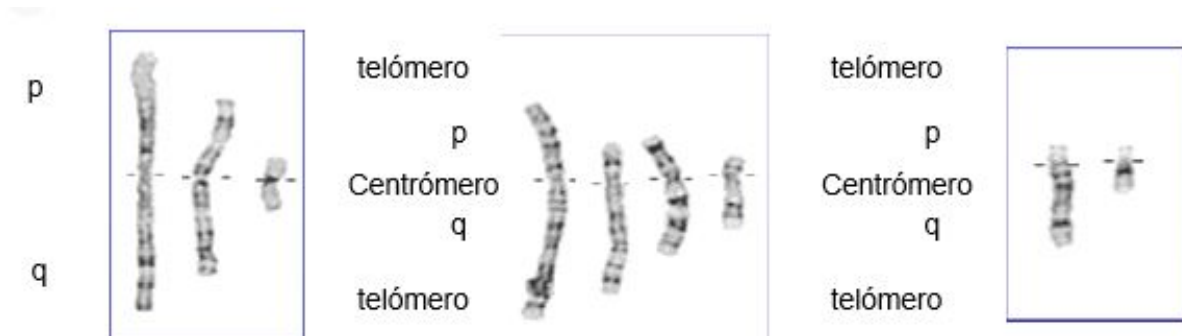


Figura 1.2: Clasificación morfológica de los cromosomas humanos.

A continuación se describen los grupos alfabéticos en los que se clasifican los cromosomas [25]:

- Grupo A: Pares cromosómicos 1, 2 y 3. Se destacan por ser los cromosomas de mayor tamaño. Específicamente par 1 y 3 metacéntricos; y par 2 submetacéntricos.
- Grupo B: Pares cromosómicos 4 y 5. Son de tamaño submetacéntrico.
- Grupo C: Pares de cromosomas 6, 7, 8, 9, 10, 11, 12, X. Son de tamaño mediano, submetacéntricos.
- Grupo D: Pares de cromosomas 13, 14 y 15. Caracterizados por ser de tamaño mediano, sub telocéntricos y telocéntricos.
- Grupo E: Pares de cromosomas 16, 17 y 18. Son de tamaño pequeño, el par 16 es metacéntrico y los pares 17 y 18 son submetacéntrico.
- Grupo F: Pares de cromosomas 19 y 20. Son de tamaño pequeño y metacéntricos.
- Grupo G: Pares de cromosomas 21, 22, Y. Estos cromosomas son caracterizados por ser de menor tamaño y sub telocéntricos.

1.3. Cariotipo

El cariotipo es una técnica para el análisis de los cromosomas con el fin de detectar alguna anomalía cromosómica. Se han realizado cariotipos en una amplia variedad de organismos, incluyendo animales, plantas y otras especies, sin embargo, para fines de esta tesis se considerarán solamente cariotipos humanos.

La realización e interpretación adecuada de esta prueba requiere la experiencia de profesionales especializados. La prueba más común se realiza en un feto en desarrollo y consiste en el estudio cromosómico del líquido amniótico. En otros casos, se puede realizar utilizando los linfocitos, un tipo de leucocito, obtenidos a partir de una muestra de sangre o médula ósea.

Para la detección de las anomalías en los cromosomas, se debe poner especial atención en ver si en la muestra se tienen más o menos de 46 cromosomas, u observar algo inusual en el tamaño o la forma de sus cromosomas, alguna de estas alteraciones puede significar que tiene una enfermedad genética.

La realización de la prueba sigue los siguientes pasos:

1. Se toma muestra de tejidos (sangre, médula ósea). En caso de un embrión se toma del líquido amniótico.

2. Se realiza un cultivo celular, de esta manera las células se multiplican y después de un tiempo se le agrega colchicina con el fin de detener la mitosis (división celular), en este proceso es donde se aprecian los cromosomas.
3. Se aíslan los cromosomas del núcleo celular y se trasladan a un portaobjetos para tratarlos con una tinción específica.
4. Por medio de un gotero se toman pequeñas muestras, y se dejan caer hechuras de láminas para obtener pequeñas gotas, conocidas como botones, se realizan alrededor de 15 botones por prueba aproximadamente.
5. Se toman microfotografías (fotografías a escala de micras) de los cromosomas por medio de un microscopio.
6. A modo de rompecabezas, se agrupan en pares las imágenes obtenidas de los cromosomas de las fotografías reales comparándolas con un ideograma (imagen de referencia para ordenar los cromosomas), se ordenan por tamaño – de mayor a menor- numerándolos del 1 al 22, con el par 23 correspondientes a los cromosomas sexuales.
7. Una vez ordenados los cromosomas se obtiene el cariotipo, por consiguiente el especialista de laboratorio evalúa los pares cromosómicos identificando cualquier posible anomalía.

En la Figura 2.1 se observa un esquema sobre la prueba del cariotipo de los pasos del 1 - 3.

En la Figura 1.3 se observa un ejemplo de las microfotografías de los cromosomas en forma de botones, como se menciona en el paso 4.

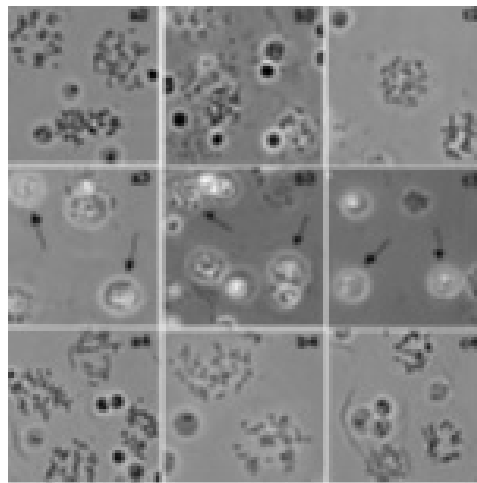


Figura 1.3: Microfotografías de los cromosomas.

Un ejemplo del paso 5 se observa en la Figura 1.4 donde se agrupan de manera manual cromosoma por cromosoma y así obtener el cariotipo.

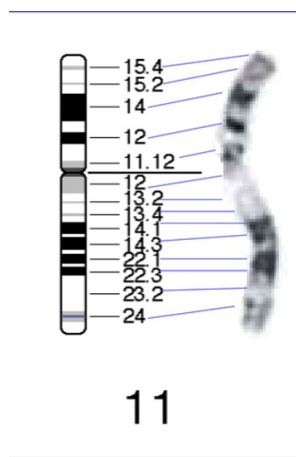


Figura 1.4: Comparación de cromosoma 11.

La Figura 1.5 es la representación del paso 7, durante este paso, el especialista examina los cromosomas previamente ordenados para detectar cualquier deformidad, falta o exceso.

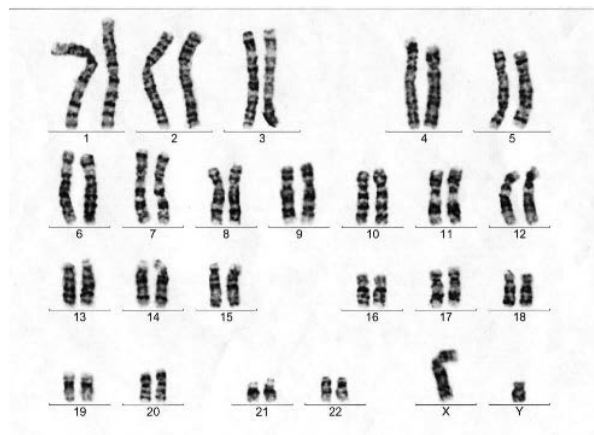


Figura 1.5: Cariotipo.

1.3.1. Ejemplos de anomalías cromosómicas

Las enfermedades cromosómicas o enfermedades congénitas son alteraciones que pueden observarse a partir de alguna alteración de ellos, como pueden ser alteraciones en cantidad, tamaño o forma en los cromosomas. Algunas de las enfermedades que pueden detectarse son:

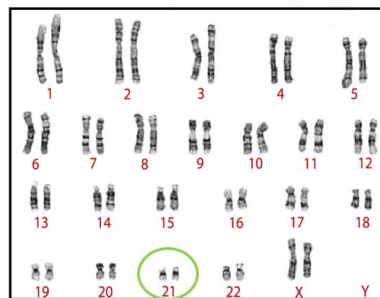
- Síndrome de Down.
- Síndrome de Patau.
- Síndrome de Turner.

A continuación se describen las alteraciones principales de cada una de ellas:

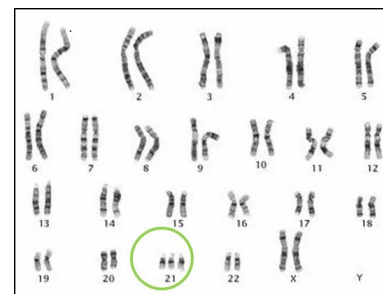
El síndrome de Down es una de las enfermedades congénitas más conocidas, pues contiene un cromosoma de más en el par 21, una copia extra de algún cromosoma es conocido como "trisomía". Algunas características físicas que se presentan debido a una alteración del par 21 son las siguientes:

- Cuello corto.
- Orejas cortas.
- Ojos en forma almendrada, rasgos hacia arriba.
- Manos y pies pequeños.

En la Figura 1.6 se compara un cariotipo de una persona saludable con sus 46 cromosomas respecto al de una persona con síndrome de Down, un cromosoma extra en el par 21.



(a) Cariotipo humano sano.



(b) Cariotipo de una persona con síndrome de Down.

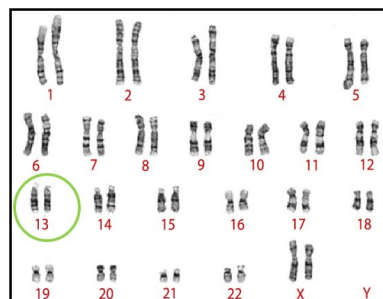
Figura 1.6: Cariotipo humano: a) Cariotipo humano sano, b) Cariotipo con una trisomía en el par 21.

El síndrome de Patau es otra enfermedad congénita, caracterizada por contener 47 cromosomas, el cromosoma extra, se encuentra específicamente en el par 13. Si un individuo presenta esta alteración, generalmente no supera un año de vida [24], alguna de las características visibles en individuos con síndrome de Patau son:

- Labio leporino.

- Malformaciones craneales.
- Malformaciones cardiacas y renales.
- Retraso en el crecimiento.
- Profundo retraso mental.

En la Figura 1.7 se puede observar el cromosoma extra en el par 13.



(a) Cariotipo humano sano.



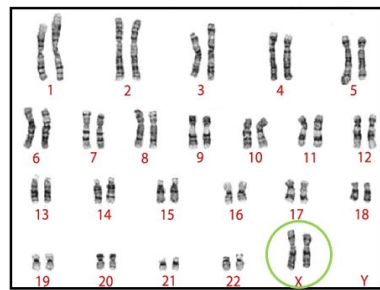
(b) Cariotipo de una persona con síndrome de Patau.

Figura 1.7: Cariotipo humano: a) Cariotipo humano sano, b) Cariotipo con una trisomía en el par 13.

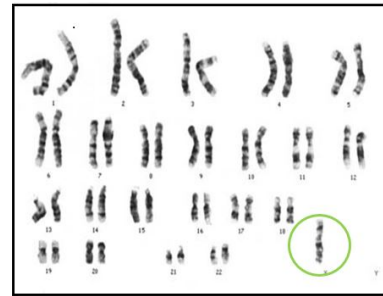
El síndrome de Turner, se caracteriza por la ausencia de un cromosoma, en este caso en vez de tener 46 cromosomas se tiene 45, el cromosoma faltante corresponde al par de la sexualidad, esta enfermedad congénita se da en mujeres. Algunas de las características que destacan son:

- Cuello corto con "pliegues" que van desde la parte superior de los hombros hasta los lados del cuello.
- Aparición de cabello en la parte inferior de la espalda
- Baja ubicación de las orejas.
- Manos y pies inflamados.

En la Figura 1.8 se puede observar: a) Cariotipo de una persona saludable, b) el Cariotipo de una persona con síndrome de Turner.



(a) Cariotipo humano sano.



(b) Cariotipo de una persona con síndrome de Turner.

Figura 1.8: Cariotipo humano: a) Cariotipo humano sano, b) Cariotipo con ausencia de cromosoma en el par sexual.

1.4. Procesamiento digital de imágenes y visión por computadora

Una de las técnicas para el mejoramiento de las imágenes, restauración de imágenes, relieves, entre otras y los procesos para resaltar información de una imagen digital es el uso del procesamiento digital de imágenes (PDI) [28], [19]. El procesamiento digital de imágenes toma una imagen, en este caso los cromosomas y produce una versión modificada de la misma, mejorándola o extrayendo los bordes. La visión por computadora consiste en el desarrollo de algoritmos para simular por visión un proceso biológico [8].

Existen tres niveles de procesamiento en visión por computadora:

Visión de bajo nivel: Este nivel utiliza operaciones de preprocesamiento de imágenes, como la reducción de ruido, mejora de contraste y filtrado, así como la detección de bordes. Estas operaciones se aplican para mejorar la calidad y la claridad de la imagen antes de realizar análisis más avanzados.

Visión de nivel intermedio: En este nivel se realiza la segmentación y clasificación de objetos individuales presentes en las imágenes. Esto implica la extracción de contornos y bordes de los objetos para identificar y delimitar las regiones de interés en la imagen.

Visión de alto nivel: En este nivel se lleva a cabo el reconocimiento de objetos específicos utilizando la información proporcionada por el usuario y la interpretación. Aquí se asigna significado a un conjunto de objetos reconocidos y se realizan tareas más complejas, como la identificación de objetos específicos en una imagen.

Por otro lado, la visión artificial es conocida como al proceso de la obtención, caracterización e interpretaron de la información de imágenes tomadas. Estos procesos a

1.4. PROCESAMIENTO DIGITAL DE IMÁGENES Y VISIÓN POR COMPUTADORA¹⁵

su vez están subdivididos en cinco áreas principales, como se observa la Figura 1.9. Las etapas de procesamiento que realiza se explicará a continuación:

1. Adquisición de la imagen, esto se realiza con la ayuda de un sensor (cámara fotográfica, celular, escáner, etc), el cual captura imágenes y las digitaliza.
2. Preprocesado, es el responsable de detectar y eliminar las fallas que puedan existir en la imagen con el fin de poder mejorarla. Una de las técnicas más utilizadas son: a) mejora del contraste, b) eliminación del ruido, y c) restauración, entre otras.
3. Segmentación, es el proceso que divide una imagen en partes de interés. Algunas de las técnicas básicas son: a) segmentación mediante umbralización, b) segmentación basada en bordes, y c) segmentación semántica, d) segmentación por agrupamiento, entre otros.
4. Obtención de la descripción, es la extracción de las características convenientes para diferenciar un objeto de otro.
5. Reconocimiento de la imagen, es el encargado de asignar una etiqueta a un objeto basándose en la información proporcionada por los usuarios.
6. Finalmente, la etapa Base de Conocimiento, controla la interacción entre dichos módulos.

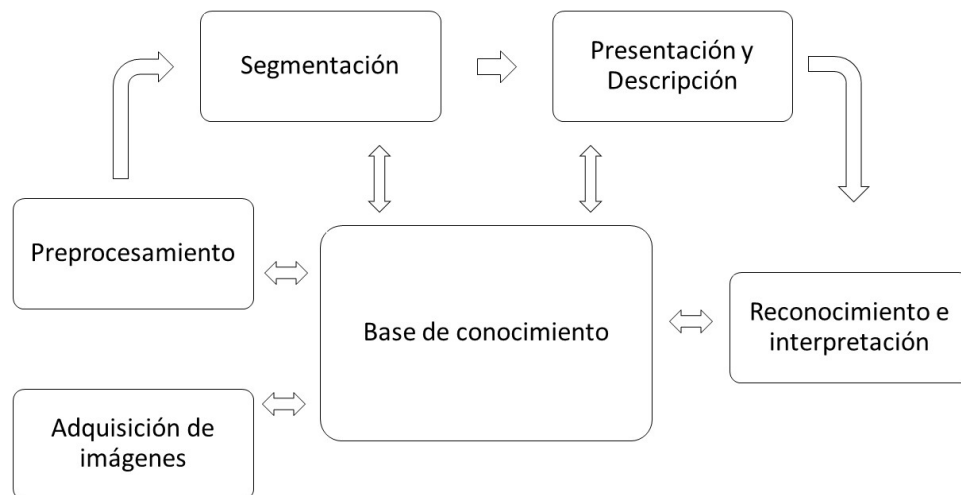


Figura 1.9: Etapas en el procesamiento de imágenes.

En este trabajo se implementarán los tres primeros pasos, pues en el tercer paso de la segmentación se aplicará la técnica de agrupamiento mediante el algoritmo *K-means* y

los Campos Aleatorios de Markov en los cromosomas. Es importante tener en cuenta que la imagen obtenida de los cromosomas se encuentre separados individualmente, ya que al estar unidos o traslapados tiene como consecuencia el tener dificultades para analizarlo.

1.4.1. Definiciones

A continuación, se presentan los conceptos básicos que son usados en el procesamiento digital de imágenes [9].

Imagen

Una imagen es una representación visual de un objeto del mundo real en dos dimensiones. Para la obtención de una imagen digital, se requiere una captura, muestreo, cuantificación y codificación. Se indica por $f(x, y)$. Una imagen está formada por pequeñas muestras, las cuales son llamadas píxeles. Para hacer referencia a un píxel individual se emplearán letras minúsculas, como p y q y un subconjunto de píxeles de la imagen se indicará mediante letras mayúsculas.

Píxel

Un píxel es un elemento direccionable muy pequeño que se encuentra en monitores, pantallas celulares, entre otros equipos; son muestras o porciones de la misma imagen de la cual se está representando o visualizando, que al combinar todos estos, nos dan la imagen completa. Entre mayor sea la cantidad de píxeles en una imagen, será de mejor calidad la imagen.

Resolución

La resolución de una imagen es la cantidad de píxeles que la describen. A mayor número de píxeles, mejor calidad. Y una medida típica es en términos de “píxeles por pulgada” (PPI).

Es importante tener en cuenta que la resolución de una imagen digitalizada es fija, lo que significa que la cantidad de píxeles que la componen no cambia. Sin embargo, al aumentar el tamaño de la imagen, se distribuyen los mismos píxeles en un área mayor, lo que resulta en una reducción de la resolución. Por otro lado, si se reduce el tamaño de la imagen, los mismos píxeles se comprimen en un área más pequeña, lo que puede aumentar la resolución aparente.

Los conceptos anteriores se pueden observar en la Figura 1.10, es una fotografía en escala de grises que muestra la parte superior de un área terrestre. Esta fotografía es una imagen (las imágenes son matrices que contienen elementos discretos), donde cada punto más pequeño de una imagen se denomina píxel. Estos píxeles contienen una escala

de tonalidades de grises compuesta por 256 niveles diferentes, representado en 8 bits; los píxeles en negro significa ausencia total de color representado por "0", mientras un píxel blanco es la presencia total del color representado por "255".

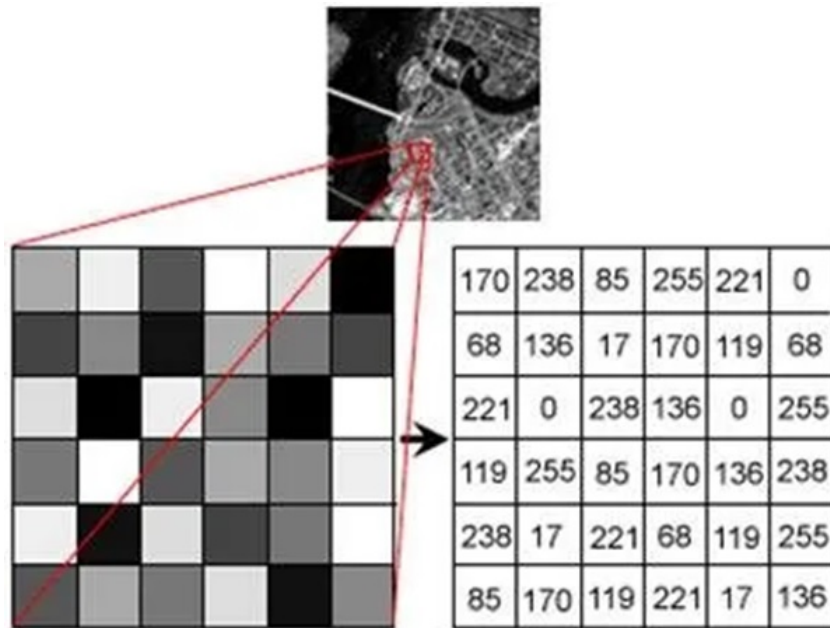


Figura 1.10: Partes de una imagen.

Segmentación

La segmentación de imágenes consiste en dividir una imagen digital en múltiples regiones distintas que contienen cada píxel (conjuntos de píxeles, también conocidos como superpíxeles) con atributos similares, con el fin de separar las regiones de interés del resto de la imagen [19]. Los algoritmos de segmentación de imágenes monocromáticas se basan en dos propiedades principales de los valores a escala de grises: Similitud y discontinuidad [9]:

- Similitud: Consiste en dividir una imagen basándose en el cambio brusco de los niveles de grises de la imagen, como puede ser la detección de bordes de una imagen.
- Discontinuidad: Se basa en dividir la imagen en regiones que son similares de acuerdo a criterios predefinidos, algunos ejemplos son la umbralización, crecimiento de regiones y fusión de regiones.

Los enfoques principales de la segunda categoría se basan en dividir una imagen en regiones que son similares de acuerdo con un conjunto de criterios predefinidos.

Entre las técnicas de segmentación se encuentra la técnica de agrupamiento mediante el algoritmo *K-means* y los Campos Aleatorios de Markov (CAM). Estos modelos se basan en asignar etiquetas a los píxeles de una imagen para indicar la región o clase a la que pertenecen. El algoritmo *K-means* se centra en el agrupamiento de píxeles o regiones similares, mientras que el CAM utiliza probabilidad para determinar a qué clase pertenece cada píxel con la idea de píxeles vecinos tendrán alta probabilidad de pertenecer a la misma clase.

El algoritmo *K-means* puede usarse como un paso previo, de preparación para el CAM. El proceso de segmentación a través de CAM utiliza una etapa de entrenamiento y otra de inferencia, lo que requiere de mayor costo computacional a comparación de otros métodos, sin embargo, dado que es capaz de capturar información tanto de los píxeles individuales como de sus vecinos permite obtener segmentaciones más precisas y coherentes en comparación con otras técnicas más simples, por lo que se convierte en una herramienta valiosa.

Capítulo 2

Planteamiento del problema

El cariotipo es un método tradicional para el estudio de los cromosomas humanos, que consiste en acomodar en una hoja los 46 cromosomas de una persona. Estos cromosomas se ordenan según su tamaño, forma y características específicas. Para la obtención de los cromosomas se necesita un tejido vivo que contenga células, en este caso unos mililitros sangre, posteriormente se realiza un cultivo celular con el fin de multiplicar las células, se le agrega colchicina¹. Una vez detenida la división celular, se toman pequeñas muestras de las células y se utilizan goteros para depositar estas muestras en una lámina de vidrio. Al dejar caer las muestras sobre la lámina, se generan pequeñas gotas que contienen los cromosomas. Los pasos que describen este procedimiento se puede apreciar en la Figura 2.1.

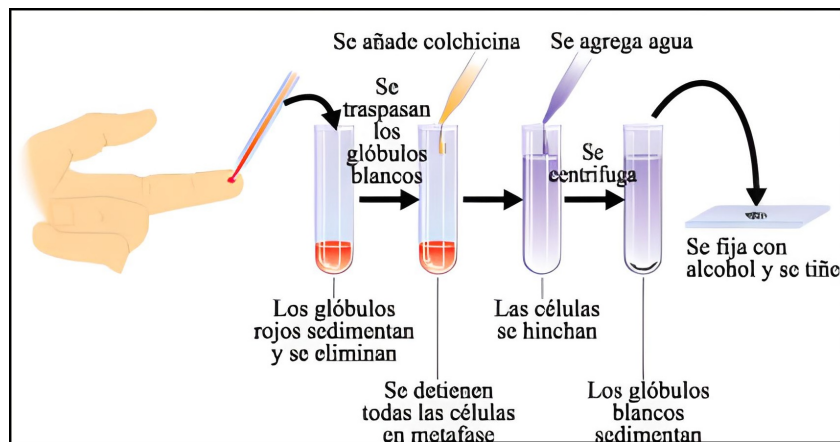


Figura 2.1: Pasos para la obtención de un cariotipo, cultivo celular.

Posteriormente, como se observa en la Figura 2.2, los botones² son observados mediante un microscopio y se toman microfotografías de cada uno de ellos, como paso

¹La colchicina es un fármaco natural, permite detener la división celular en la fase metafásica.

²Pequeña gota que contiene los cromosomas extendidos y fijados sobre una lámina de vidrio.

siguiente se imprimen estas imágenes y se recortan de manera individual, en teoría se espera obtener alrededor de 46 cromosomas en cada botón, posteriormente con ayuda de un ideograma como referencia visual³ se busca emparejar los cromosomas, por tamaño, forma y estructura. Este análisis y clasificación permite ordenar los cromosomas y determinar si existen anomalías genéticas o alteraciones cromosómicas en la muestra [2]. Dicho procedimiento usualmente se realiza de manera manual, lo que implica dedicar demasiado tiempo y esfuerzo que repercute en la obtención temprana del diagnóstico y es propenso a errores humanos.

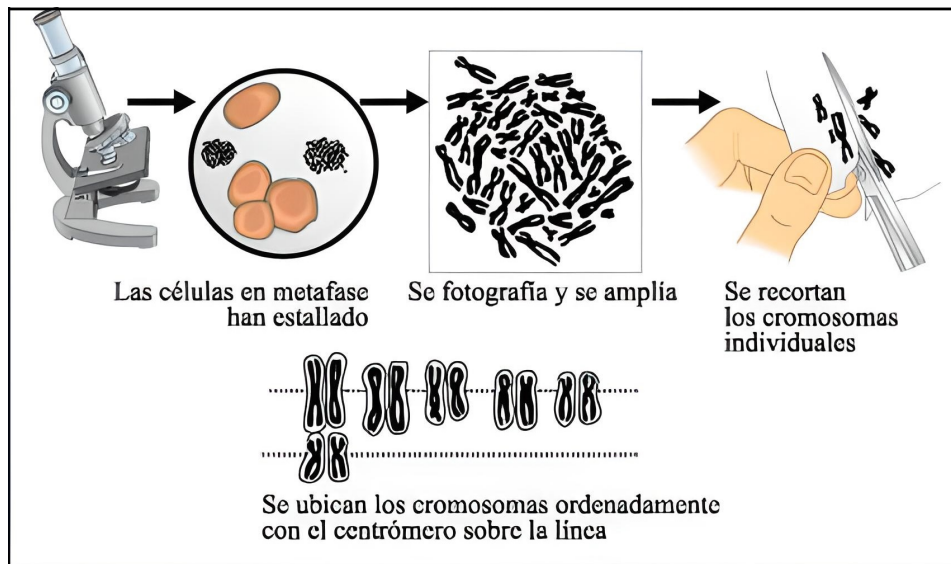


Figura 2.2: Pasos para la obtención de un cariotipo, ordenamiento.

Por otro lado, el campo del procesamiento digital de imágenes puede aportar al proceso del cariotipo a través de la segmentación. Realizar la segmentación en el análisis de cromosomas permite preparar la imagen de manera específica para que los cromosomas puedan ser distinguidos claramente del fondo y analizados de forma individual.

La segmentación correcta de la imagen nos permite separar los cromosomas del resto de la imagen, lo que a su vez nos facilita reconocimiento y análisis de los mismos. Esto nos brinda información sobre su morfología, ubicación de su centrómero, la organización de sus bandas y el número de cromosomas presentes, lo cual es esencial en el estudio de la genética y la detección de posibles anomalías genéticas.

Es fundamental que los cromosomas no estén entrelazados y que se encuentren de forma individual, ya que esto garantiza que se puedan distinguir claramente de otros

³Imagen de referencia con cromosomas ordenados y clasificados por tamaño, forma y característica ideal.

elementos presentes en la muestra, lo que facilita su identificación y análisis. La correcta segmentación de la imagen también contribuye a la eliminación de artefactos y la reducción de errores en el análisis de los cromosomas [7], [15].

De lo anterior, se plantean las siguientes preguntas de investigación.

2.1. Preguntas de investigación

¿Se puede desarrollar un método viable para la segmentación de cariotipo?

¿El procesamiento digital de imágenes proporciona información confiable y suficiente en el armado del cariotipo?

2.2. Objetivos

2.2.1. Objetivo general

Desarrollar una metodología basada en los Campos Aleatorios de Markov para la segmentación de cariotipos, con el fin de mejorar la calidad de los cromosomas y lograr una delimitación precisa, que pueda ser utilizada como una herramienta eficiente en el diagnóstico temprano de enfermedades cromosómicas.

2.2.2. Objetivos particulares

1. Realizar una revisión de la literatura científica y técnica relacionada con la segmentación de cariotipos y los Campos Aleatorios de Markov, con el fin de obtener un amplio conocimiento sobre los avances y las técnicas existentes en el campo.
2. Diseñar y desarrollar un algoritmo de segmentación de cariotipos basado en los Campos Aleatorios de Markov, que permita una delimitación precisa y eficiente de los cromosomas, considerando las características específicas de las imágenes de cariotipos.
3. Comparar los resultados obtenidos mediante la metodología propuesta con un banco de imágenes de laboratorio previamente clasificadas, con el objetivo de determinar la calidad y confiabilidad de la segmentación realizada y verificar la consistencia de los resultados.

2.3. Hipótesis

La aplicación de la metodología de segmentación de cariotipos utilizando los Campos Aleatorios de Markov en el análisis cromosómico permitirá mejorar el diagnóstico tem-

prano de enfermedades cromosómicas, al proporcionar una delimitación más precisa de los cromosomas y facilitar el ordenamiento del cariotipo.

2.4. Metodología

Para lograr el objetivo planteado se proponen una metodología de 3 fases:

1. **Preprocesamiento:** Esta fase se encarga de preparar la imagen de entrada antes de la segmentación. El objetivo es realizar operaciones como la binarización para convertir la imagen en blanco y negro, lo cual facilita el procesamiento posterior.
2. **Segmentación:** En esta fase se utiliza el algoritmo K-means para dividir la imagen en grupos o clusters, lo que permite separar las diferentes regiones de interés. Posteriormente, se aplica la técnica de Campos Aleatorios de Markov (CAM) para mejorar la precisión de la segmentación. Los CAM capturan la información contextual de los píxeles y ayudan a delimitar de manera más precisa los contornos de los cromosomas.
3. **Postprocesamiento:** Finalmente, se obtiene una imagen de salida con las delimitaciones mejoradas. En esta fase se pueden aplicar técnicas adicionales, como el suavizado de bordes o la eliminación de pequeños detalles no deseados, con el objetivo de obtener una imagen final más clara y legible.

La metodología propuesta se enfoca en el desarrollo de una herramienta computacional utilizando un enfoque metodológico cuantitativo; esta metodología implica el uso de instrumentos cuantitativos, como modelos matemáticos, en este caso los Campos Aleatorios de Markov es una herramienta estadísticas que nos permite modelar la relación y dependencia entre píxeles vecinos para mejorar la delimitación de los cromosomas en un cariotipo y mejorar la nitidez en los resultados.

En resumen, la metodología anteriormente mencionada comienza con el preprocesamiento para acondicionar la imagen como es transformación de color a escalas de grises, seguido de la segmentación utilizando los Campos Aleatorios de Markov y K-means para la clasificación y etiquetado de datos de la imagen, y finaliza con el postprocesamiento para obtener una imagen final mejorada. La metodología propuesta se puede observar en la Figura 2.3.

2.5. Motivación y justificación

El hecho de que el proceso de análisis de cariotipos manual sea tardado, complejo y propenso a errores subraya la necesidad de buscar enfoques más avanzados y automatizados. La metodología propuesta, que se basa en la segmentación de cromosomas

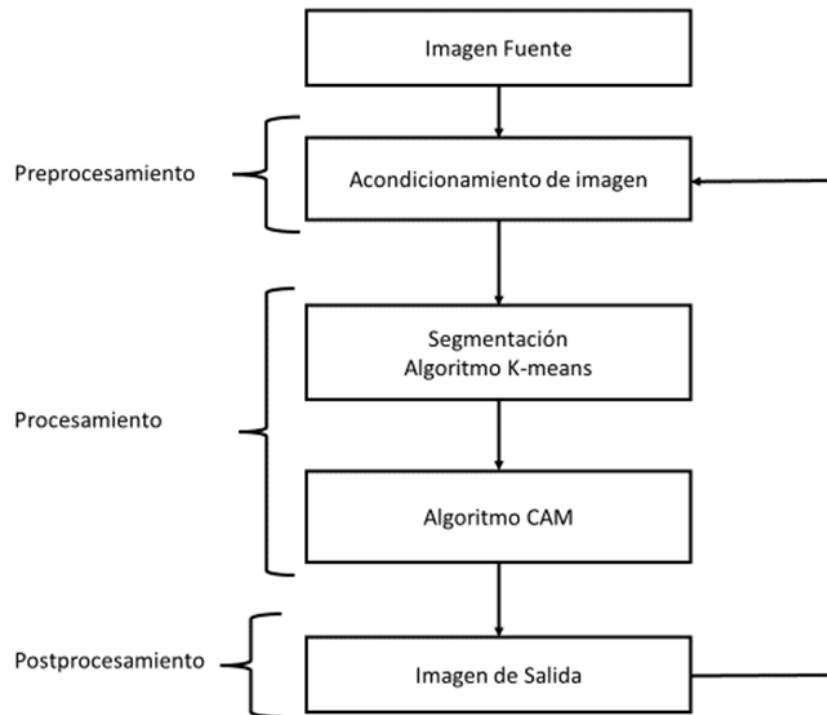


Figura 2.3: Diagrama de la metodología propuesta.

utilizando los Campos Aleatorios de Markov, nos da la oportunidad de mejorar la precisión y eficiencia en este proceso. Esto permitiría obtener resultados más confiables en menos tiempo, lo que a su vez puede llevar a una atención médica más oportuna y efectiva para los pacientes.

Además, el desarrollo de esta metodología podría contribuir al sector público al ofrecer no solo una herramienta sino también diagnósticos en menor tiempo, de fácil acceso, de manera portátil, sin importar su ubicación geográfica o su capacidad económica por otro lado al sector de la salud puede ayudar a los profesionales a identificar y analizar alteraciones cromosómicas asociadas a enfermedades genéticas; esto puede facilitar el diagnóstico temprano, la comprensión de los mecanismos subyacentes de la enfermedad y la selección de tratamientos más adecuados y en el ámbito bioético el uso del algoritmo en la delimitación de cromosomas plantea importantes consideraciones pues es fundamental asegurar el consentimiento informado de los pacientes, la privacidad y confidencialidad de los datos genéticos, y garantizar que el uso de la información genética esté en línea con los principios éticos establecidos.

Este algoritmo contribuye al avance científico y tecnológico en el campo de la citogenética y la ingeniería electrónica, al utilizar herramientas y técnicas innovadoras para abordar desafíos específicos en el análisis de imágenes cromosómicas. Esto abriría

nuevas posibilidades de investigación y colaboración multidisciplinaria, lo que a su vez podría generar un impacto positivo tanto en la comunidad científica como en la sociedad en general.

La justificación para el desarrollo de la tesis “Metodología para Segmentación de Cariotipos utilizando los campos aleatorios de Markov” se basa en la necesidad de mejorar el proceso de diagnóstico temprano de enfermedades a través del cariotipo. Aunque el cariotipo es una técnica ampliamente utilizada, su análisis manual puede ser laborioso y propenso a errores humanos. Además, la interpretación de los resultados puede ser subjetiva y requerir un alto nivel de experiencia por parte de los especialistas.

La propuesta de utilizar los Campos Aleatorios de Markov para la segmentación de cariotipos surge como una solución innovadora y prometedora. Los CAM son modelos probabilísticos que permiten incorporar información contextual y espacial en el proceso de segmentación de imágenes. Al aplicar esta metodología al análisis de cariotipos, se busca mejorar la precisión y eficiencia en la identificación y clasificación de los cromosomas, facilitando así la detección de anomalías cromosómicas y el establecimiento de un diagnóstico más preciso.

Entre estas técnicas, se encuentran la detección de bordes, que permite identificar los límites entre distintas regiones de la imagen, la erosión, que permite eliminar pequeñas irregularidades de la imagen y suavizar los contornos. Por otro lado, la utilización de técnicas de texturizado y suavizado pueden ayudar a mejorar la calidad de la imagen y hacer más visible la información relevante.

Además, la segmentación se utiliza para separar distintas regiones de la imagen en función de sus características, lo que puede ser muy útil para el análisis del cariotipo, así como para identificar regiones anómalas, lo que es esencial en el diagnóstico de enfermedades genéticas.

En definitiva, el uso de técnicas de procesamiento digital de imágenes en el estudio del cariotipo puede mejorar la precisión y rapidez del análisis, permitiendo una mejor comprensión de las características y anomalías. La combinación de estas con la experiencia de los profesionales de la genética puede llevar a avances significativos en el diagnóstico y tratamiento de enfermedades genéticas en el futuro.

La colaboración multidisciplinaria entre genética, medicina clínica, bioinformática e ingeniería electrónica es fundamental para abordar los desafíos en el diagnóstico temprano de enfermedades. La integración de la ingeniería electrónica y las técnicas de procesamiento de imágenes, como los CAM, en el análisis de cariotipos brinda la oportunidad de aprovechar tecnologías avanzadas para mejorar la calidad y velocidad del diagnóstico [23].

Capítulo 3

Marco teórico

3.1. Sistemas de vecindades y cliques

La conectividad entre píxeles es comúnmente utilizado para la detección de regiones u objetos en determinada imagen. Respecto a la vecindad de un píxel, se le conoce como la relación que tiene un píxel respecto a los píxeles cercanos que contiene alrededor de él.

Sea

$$S = \{(i, j) \mid 1 \leq i \leq m, 1 \leq j \leq n\} \quad (3.1)$$

el conjunto de estados de una rejilla rectangular que corresponde a los puntos de una imagen en 2D de tamaño $m \times n$. Los estados en S están relacionados entre sí a través de un sistema de vecindad. Un sistema de vecindad para S , se define como:

$$\mathcal{N} = \{\mathcal{N}_{(i,j)} \mid \forall (i, j) \in S\} \quad (3.2)$$

donde $\mathcal{N}_{(i,j)}$ es el conjunto de los estados vecinos de (i, j) que cumple las siguientes propiedades:

1. Un estado no es vecino de sí mismo: $(i, j) \notin \mathcal{N}_{(i,j)}$
2. Las relaciones de vecindad son recíprocas: $(i, j) \in \mathcal{N}_{(i,j)'} \iff (i, j)' \in \mathcal{N}_{(i,j)}$

Para una rejilla regular S , el conjunto de vecinos de (i, j) está definido como el conjunto de estados próximos dentro de un radio \sqrt{r} de (i, j) , es decir,

$$\mathcal{N}_{(i,j)} = \{(i, j)' \in S \mid \text{dist}(\text{pixel}_{(i,j)'}, \text{pixel}_{(i,j)})^2 \leq r, (i, j)' \neq (i, j)\} \quad (3.3)$$

donde $\text{dist}(A, B)$ denota la distancia euclídea entre A y B , y r es un número entero el cual representa el radio que delimita el conjunto de vecinos de i [3].

La distancia euclidiana entre dos puntos $A = (pixel_{1'}, pixel_{2'})$ y $B = (pixel_1, pixel_2)$ es

$$\|A - B\| = [(pixel_{1'} - pixel_1)^2 + (pixel_{2'} - pixel_2)^2]^{1/2} \quad (3.4)$$

Algunos tipos de vecindad son: *sistema de primer orden* llamada vecindad de 4 porque cada estado (interior) posee un conjunto de 4 vecinos; *sistema de segundo orden* llamada vecindad de 8; y, *sistema de orden n* o vecindad de n , en donde n indica los conjuntos de vecinos más alejados como se puede ver en la Figura 3.1 (c). Esto es, una vecindad de 4 se constituye como:

$$\mathcal{N}_{(i,j)} = \{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\} \quad (3.5)$$

que son los que se encuentran arriba, abajo, derecha e izquierda como se puede ver en la Figura 3.1 (a), observe que los estados frontera tienen tres y los estados de las esquinas tienen dos.

El sistema de segundo orden, vecindad de 8 incluye la vecindad de 4 más $(i+1, j+1)$, $(i+1, j-1)$, $(i-1, j+1)$ y $(i-1, j-1)$ como se puede observar en la Figura 3.1 (b). Además se puede ver en [8], [14], [21], [11], por lo que:

$$\begin{aligned} \mathcal{N}_{(i,j)} = \{ & (i-1, j), (i+1, j), (i, j-1), (i, j+1), \\ & (i+1, j+1), (i+1, j-1), (i-1, j+1), (i-1, j-1) \}. \end{aligned}$$

Un clique C para $\{S, \mathcal{N}\}$ se define como un subconjunto de S tal que C consiste en un único estado $C = (i, j)$ o un par de estados vecinos $C = (i, j), (i, j)'$, o tres estados vecinos $C = (i, j), (i, j)', (i, j)''$, y así sucesivamente. Se usarán las siguientes notaciones C_1 , C_2 y C_3 , para referirnos a cliques de un único elemento, un par de elementos y tres elementos, respectivamente donde:

$$C_1 = \{(i, j) | (i, j) \in S\}, \quad (3.6)$$

$$C_2 = \{\{(i, j), (j, k)\} | (j, k) \in \mathcal{N}_{(i,j)}, (i, j) \in S\}, \text{ y} \quad (3.7)$$

$$C_3 = \{\{(i, j), (i, j)', (i, j)''\} | (i, j), (i, j)', (i, j)'' \in S \text{ son vecinos unos de otros}\}. \quad (3.8)$$

Los estados en un clique están ordenados, así que $\{(i, j), (i, j)'\}$ no es igual a $\{(i, j)', (i, j)\}$. El conjunto de todos los cliques para $\{S, \mathcal{N}\}$

$$C = C_1 \cup C_2 \cup C_3, \dots, \cup C_n \quad (3.9)$$

donde n son los posibles conjuntos de cliques más grandes.

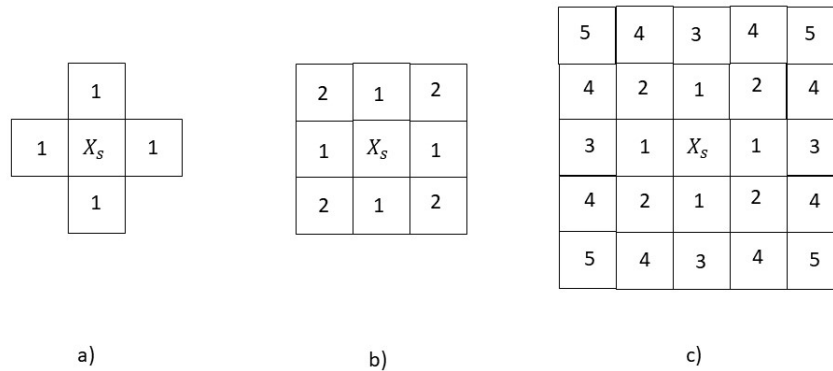


Figura 3.1: Vecindades. a) Sistema de vecinos de primer orden, b) Sistemas de vecinos de segundo orden, c) Sistemas de vecinos de orden $n = 1, 2, \dots, 5$.

El tipo de clique para $\{S, \mathcal{N}\}$ de una rejilla regular viene determinado por su tamaño, dimensión y orientación. La Figura 3.2 muestra los distintos tipos de cliques para sistemas de vecinos de primer y segundo orden de una rejilla. El inciso a) y b) indica cliques de primer orden haciendo combinaciones individuales, pares en forma horizontal y vertical. Para el clique de vecindad de segundo orden incluye las combinaciones del inciso a) y b), el inciso c) que muestra combinaciones de pares en forma diagonal, inciso d) combinaciones en conjuntos de tres e inciso e) combinaciones en conjunto de 4. A medida que aumenta el orden del sistema de vecinos, el número de cliques crece rápidamente y como consecuencia también el coste computacional.

3.2. Problemas de etiquetado

La mayoría de los problemas de visión artificial se consideran como problemas de etiquetado, usando restricciones, debido al conocimiento a priori y al de las observaciones.

Hay muchos problemas para el análisis e interpretaciones de imágenes en el etiquetado ya que el problema esta en el etiquetado de los píxeles o en las características de la imagen.

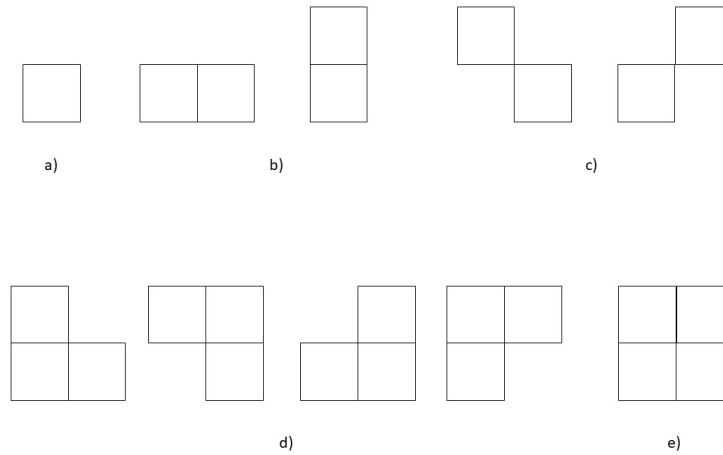


Figura 3.2: Cliques de primer y segundo orden.

3.2.1. Etiquetas

Un problema de etiquetado se especifica en términos de un conjunto de sitios y un conjunto de etiquetas. Sea S como en (3.1) el conjunto de sitios.

Una etiqueta es un evento que puede pasar a un sitio. Sea \mathcal{L} un conjunto de etiquetas que puede ser continuo o discreto. En el caso continuo, el conjunto de etiquetas corresponde la línea real \mathbb{R} o un intervalo compacto de \mathbb{R} ,

$$\mathcal{L} = [x_l, x_h] \subset \mathbb{R}. \quad (3.10)$$

En caso discreto, las etiquetas se asumen dentro de M valores discretos,

$$\mathcal{L} = \{l, \dots, l_M\}. \quad (3.11)$$

En detección de bordes, se asigna a la etiqueta $\mathcal{L} = \{\text{borde}, \text{noborde}\}$.

El problema de etiquetado consiste en asignar una etiqueta del conjunto de etiquetas \mathcal{L} a cada estado de S ; por ejemplo, en detección de bordes de una imagen se le asigna una etiqueta $f_{(i,j)}$ dentro del conjunto $\mathcal{L} = \{\text{borde}, \text{noborde}\}$ para cada estado $(i, j) \in S$, donde los elementos en S indican los píxeles de la imagen. En otras palabras, llamaremos etiquetado a

$$f = \{f_{(1,1)}, \dots, f_{(n,m)}\} \quad (3.12)$$

donde a cada estado $(i, j) \in S$ se le asigna una única etiqueta $l \in \mathcal{L}$ a este conjunto se le conoce como *configuración* en términos de campos aleatorios.

3.3. Transformaciones realizadas a una imagen

Las transformaciones en las que se genera un nuevo píxel, se obtienen mediante la función del valor de intensidad, y se clasifican en dos tipos:

1. Operaciones individuales, y
2. Operaciones de vecindad.

La generación del nuevo píxel depende de cada píxel, o del valor de los píxeles cercanos a lo que se le conoce como vecindad de píxel.

3.3.1. Transformaciones individuales

Consiste en obtener el valor del píxel de una localización dada en la imagen, para así modificarlo con un operador, lo cual puede ser lineal o no lineal, después de realizar dicha operación el valor del nuevo píxel se coloca en la correspondiente localización de la nueva imagen. La salida no depende de los píxeles adyacentes al píxel original. Uno de los operadores es *operador umbral* el cual consiste en crear una imagen de salida binaria a partir de una escala de grises, donde el nivel de transición está dada por el parámetro de entrada p_1 y la función de transformación es la siguiente:

$$q = \begin{cases} 0 & \text{si } p \leq p_1 \\ 255 & \text{si } p > p_1 \end{cases} . \quad (3.13)$$

3.3.2. Transformaciones de vecindad

Son las transformaciones para modificar una imagen de entrada en otra imagen distinta, o en una matriz de valores transformados. Se dice que todo píxel p de coordenadas (i, j) , tienen cuatro vecinos horizontales y verticales cuyas coordenadas vienen dadas por:

$$(i + 1, j); (i - 1, j); (i, j + 1); (i, j - 1)$$

hay que considerar excepciones cuando el píxel (i, j) es un punto de frontera o esquina de la imagen. Como se puede observar en la Figura 3.3.

3.3.3. Operaciones de vecindad

Transforma el valor de un píxel p en la posición (i, j) , teniendo en cuenta los valores de los píxeles vecinos. Por ejemplo, si se considera una vecindad de segundo orden $\mathcal{N}_{(i,j)}$, se realiza una suma ponderada con los valores de los ocho vecinos, y el resultado de dicha suma es el valor del nuevo píxel q de la imagen de salida de la misma posición (i, j) .

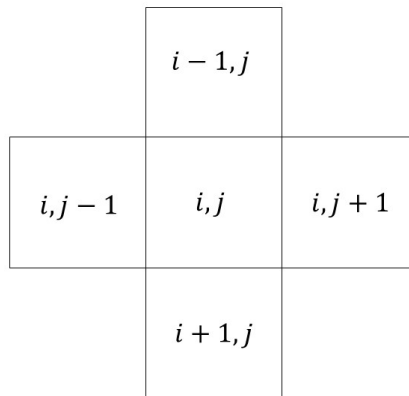


Figura 3.3: Vecindad cuatro.

Finalmente, se establecen valores de ponderación, definiendo una máscara con valores constantes. Esta máscara es un filtro. Por ejemplo, se define la siguiente máscara:

$$A = \begin{bmatrix} 1.0 & 2.0 & 1.0 \\ 0.0 & 1.2 & 0.0 \\ -1.0 & -2.0 & -1.0 \end{bmatrix}.$$

El valor del píxel $q(i, j)$ vendría dado por la suma ponderada, con los factores de ponderación definidos por la máscara:

$$\begin{aligned} q(i, j) = & 1 * p(i - 1, j) + 2 * p(i, j - 1) + 1 * p(i + 1, j - 1) \\ & + 0 * p(i - 1, j) + 1.2 * p(i, j) + 0 * p(i + 1, j) \\ & - 1 * p(i - 1, j + 1) - 2 * p(i, j + 1) - 1 * p(i + 1, j + 1). \end{aligned}$$

Se realiza esta transformación sobre todos los píxeles de la imagen original con la misma máscara, se obtendrá una nueva imagen de salida de dimensión inferior a la original.

En la figura se muestra el resultado de aplicar la operación de la máscara sobre la imagen original, con la máscara dada. Como se puede observar, el resultado es un relieve donde se marcan los bordes.

3.3.4. Operadores de extracción de bordes

Las características de una imagen, como son los bordes y contornos, son detectados mediante los cambios de intensidad o de color, ya que es importante en la interpretación

de la imagen. Los bordes llegan a considerarse como puntos en una imagen y su intensidad cambia drásticamente dependiendo de la dirección determinada, a continuación se mostrara algunos operadores utilizados como es operador Roberts, operador Prewitt, operador Sobel y operador Frei-Chen; cada uno de ellos utiliza una matriz o máscara para calcular la magnitud y la dirección del gradiente de la imagen en cada punto, lo que permite resaltar los bordes. A continuación se describen los operadores, cabe mencionar que unos son similares a otros:

1. Operador Roberts

Es uno de los filtros más antiguos utilizados para la localización de bordes de imágenes, ya que es simple y fácil de calcular, sin embargo, los resultados no serán los mejores. Es muy pequeño, ya que utiliza una matriz de coeficientes de 2×2 , para determinar el gradiente en sus dos diferentes direcciones a lo largo de sus diagonales, en las matrices mostradas de la Ecuación 3.14 donde H_1^R determina el gradiente de diferencial derecha y H_2^R determina el gradiente diferencial izquierda. El filtro está definido por:

$$H_1^R = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}; \text{ y, } H_2^R = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (3.14)$$

Este operador reacciona específicamente a los bordes de la imagen que tienen una dirección diagonal, por esta razón el filtro es poco selectivo en dirección a los gradientes, particularmente cuando se calcula sobre regiones con la misma orientación constante, como una línea vertical u horizontal.

2. Operador Prewitt

Se obtiene mayores resultados que el filtro Roberts al utilizar como filtro una matriz de coeficientes de 3×3 una máscara más grande que el filtro anterior, por lo que puede tener un costo computacional más grande, una de las ventajas es fácil de configurar para no ser tan vulnerable al ruido propio de la imagen. El filtro está definido por:

$$H_i^P = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}; \text{ y, } H_j^P = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}. \quad (3.15)$$

Las matrices de la Ecuación 3.15 donde la matriz H_i^P sirve para calcular las diferencias horizontales y la matriz H_j^P sirve para calcular las diferencias horizontales, esto hace que tenga una gran sensibilidad al detectar bordes en dirección vertical y horizontal, sin embargo, es muy propenso a la sensibilidad del ruido. Como se comenta a continuación:

En la Figura 3.4 se observa los bordes de la imagen de los cromosomas, sin embargo, al aplicar el mismo filtro a la Figura 3.5 de un cromosoma se obtiene el realce de los bordes y ruido.

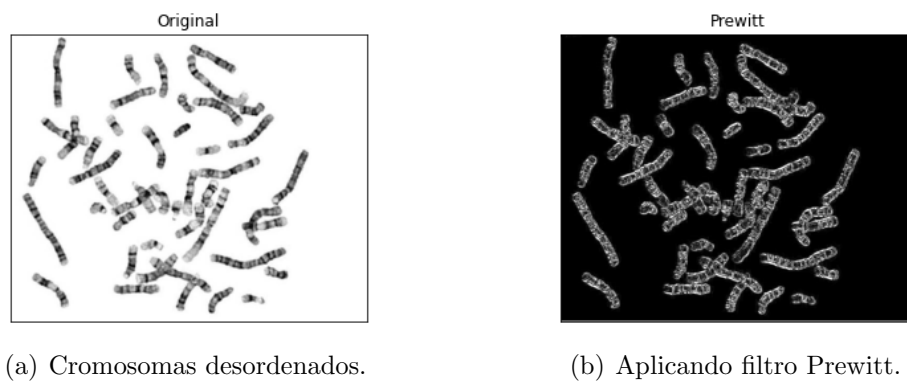


Figura 3.4: Comparación: a) Fotografía cromosomas, b) Filtro Prewitt.

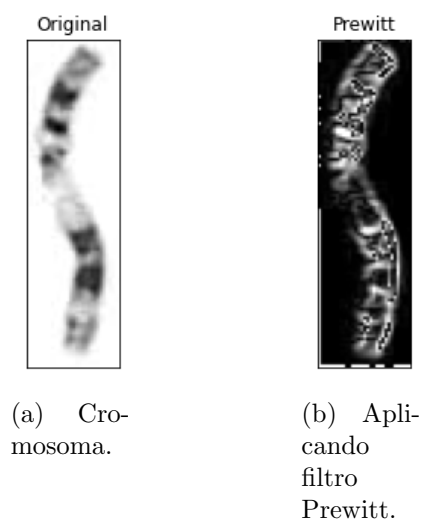


Figura 3.5: Comparación: a) Fotografía cromosoma, b) Filtro Prewitt.

3. Operador de Sobel

El filtro utilizado en el operador de Sobel también ocupa una matriz de 3×3 es idéntico al filtro de Prewitt, con la diferencia que a este filtro se le da un peso mayor al renglón o columna central del filtro, diseñado para proporcionar una mayor sensibilidad a los bordes diagonales, muy útil en casos donde los bordes puedan tener una dirección inclinada. La matriz del filtro queda de la siguiente manera:

$$H_i^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}; \text{ y, } H_j^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \quad (3.16)$$

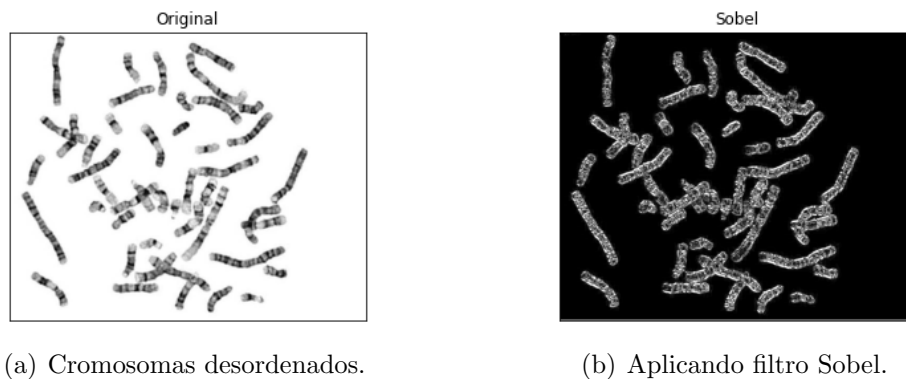


Figura 3.6: Comparación: a) Fotografía cromosomas, b) Filtro de Sobel.

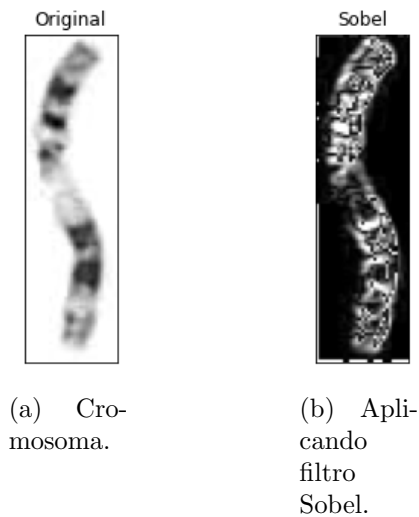


Figura 3.7: Comparación: a) Fotografía cromosoma, b) Filtro Sobel.

Las matrices de la Ecuación 3.16 donde la matriz H_i^S calcula las diferencias horizontales y la matriz H_j^S calcula las diferencias horizontales, muy similar al operador Prewitt, solo que tiene mayor peso al renglón o columna central, esto hace que tenga mayor sensibilidad al momento de detectar bordes diagonales, también es propenso al ruido como se menciona a continuación. En la Figura 3.6 se observa los bordes de la imagen de los cromosomas como en la Figura 3.4 al igual cuando se aplica el filtro a la Figura 3.7 de un cromosoma se obtiene el realce de los bordes y ruido.

4. Operador Frei-Chenn

El filtro utilizado en el operador de Frei-Chenn ocupa una matriz de 3×3 es casi idéntico a los filtros mencionados anteriormente, muy útil en la detección de bordes dando resultados más precisos. La matriz del filtro queda de la siguiente manera:

$$H_i^F = \begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}; \text{ y, } H_j^F = \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}. \quad (3.17)$$

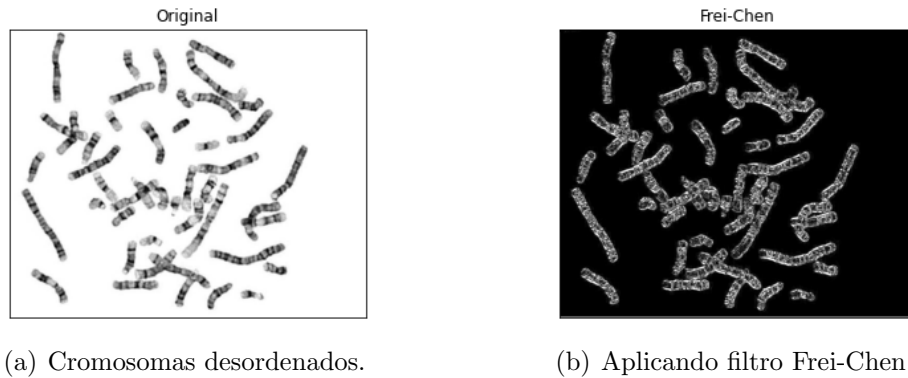


Figura 3.8: Cromosoma humano: a) Fotografía de cromosomas, b) Aplicando filtro Frei-Chen.

Como se observa en la Ecuación 3.17 la matriz H_i^F calcula las diferencias horizontales y la matriz H_j^F calcula las diferencias horizontales, al ser matrices más complejas tiene una mayor detección de bordes de manera diagonal y curva, pero sigue siendo propenso a ruido. En la Figura 3.8 y en la Figura 3.9 se pueden observar resultados similares a los filtros anteriores, se puede notar que en imágenes con poco ruido se aprecian los bordes, en caso contrario, cuando la imagen presenta ruido, este puede interferir en la calidad y claridad de la imagen, dificultando la segmentación adecuada de los cromosomas.

En resumen, el uso de operadores de detección de bordes como Roberts, Sobel, Prewitt y Frei-Chen son adecuados para detectar bordes en imágenes, pero pueden ser sensibles al ruido. Por lo tanto, se busca utilizar métodos de segmentación alternativos que sean más robustos frente al ruido y permitan obtener resultados más precisos.

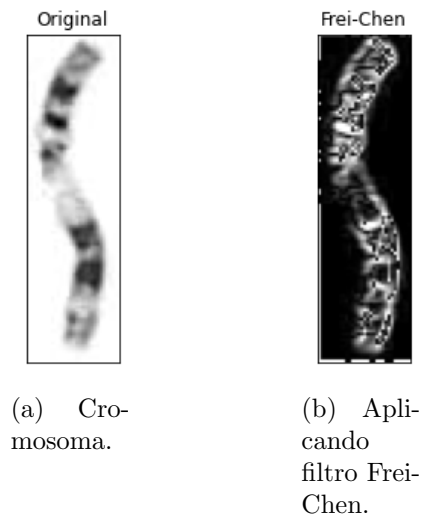


Figura 3.9: Cromosoma humano: a) Fotografía cromosoma, b) Aplicando filtro Frei-Chen.

Los métodos de segmentación alternativos puede ser el uso de Campos Aleatorios de Markov y el algoritmo K-means para separar los objetos de interés del fondo y reducir el impacto del ruido en el resultado final.

3.4. Segmentación de imágenes

La segmentación de imágenes es el proceso de seleccionar y agrupar los píxeles que contienen características visuales y numéricas similares entre sí. Al realizar la agrupación de píxeles o datos que se generan se denominan *clusters*, cada uno contiene un centroide, el cual representa el valor medio de los datos con características similares. Este método de agrupamiento de datos es una técnica para la clasificación de datos que nos permite tener un mejor manejo, la aplicación de esta técnica se utiliza en diversos campos como: la minería de datos, compresión de datos y la segmentación de imágenes.

3.5. Campos aleatorios de Markov

La teoría de campos aleatorios de Markov [18] es una rama de la teoría de la probabilidad que se utiliza para analizar las dependencias espaciales o contextuales en fenómenos físicos, esto quiere decir que, ayuda a comprender cuanto una región está asociada o relacionada con las características de las regiones vecinas. Esta teoría se basa en el proceso de decisión estocástico y tiene aplicaciones en diversos campos como la medicina, biomedicina y robótica, entre otros. Estos campos aleatorios proporcionan una manera conveniente y consistente de modelar entidades que dependen del contexto, como los píxeles en el caso de operar con una imagen, son especialmente exitosos

en la modelación de problemas de visión tales como la restauración de imágenes, la segmentación, la reconstrucción de superficies, el análisis de texturas, el flujo óptico, la integración visual y la delimitación.

Los campos aleatorios de Markov, en el marco de la segmentación de imágenes ayudan a integrar características relevantes, disminuir el ruido y la incertidumbre. Además, estas capacidades contribuyen a mejorar la precisión y la calidad general en la segmentación de imágenes [11].

Sea $F = \{F_{(1,1)}, \dots, F_{(m,n)}\}$ una familia de variables aleatorias definidas en el conjunto S , en donde, cada variable aleatoria $F_{(i,j)}$ toma un valor $f_{(i,j)}$ en \mathcal{L} . La familia de F es llamada *Campo aleatorio*.

En la segmentación de imágenes los estados S corresponden a los píxeles, las variables aleatorias $F_{(i,j)}$ son las características que poseen estos píxeles, los cliques son aristas no dirigidas que se interconectan entre los píxeles vecinos.

Se dice que la familia de variables aleatorias F es un *Campo Aleatorio de Markov* sobre S en un sistema de vecindad \mathcal{N} si y sólo si satisface las siguientes dos condiciones:

$$P(f) > 0, \forall f \in \mathbb{F} \quad \text{positivo, y} \quad (3.18)$$

$$P(f_{(i,j)} | f_{S - \{(i,j)\}}) = P(f_{(i,j)} | f_{\mathcal{N}_{(i,j)}}) \quad \text{Markovianidad} \quad (3.19)$$

donde f es una configuración de F , $P(f)$ mide la probabilidad de ocurrencia de la configuración, $P(f_{(i,j)})$ es la probabilidad de que la variable aleatoria $F_{(i,j)}$ tome el valor de $f_{(i,j)}$, $S - \{(i,j)\}$ es el conjunto de diferencias, $f_{S - \{(i,j)\}}$ denota el conjunto de etiquetas de los sitios en $S - \{(i,j)\}$ y $f_{\mathcal{N}_{(i,j)}} = \{f_{(i,j)'} | (i,j)' \in \mathcal{N}_{(i,j)}\}$ es el lugar para el conjunto de etiquetas en los sitios de vecindad (i,j) .

Un conjunto de variables aleatorias F se dice que son campos aleatorios de Gibbs sobre S con respecto al conjunto de vecindad \mathcal{N} si y sólo si sus configuraciones siguen una *Distribución de Gibbs*.

La distribución de Gibbs toma la siguiente forma:

$$P(f) = \frac{1}{Z} \exp \left[-\frac{1}{T} U(f) \right] \quad (3.20)$$

donde

$$Z = \sum_{f \in \mathbb{F}} \exp \left[-\frac{1}{T} U(f) \right] \quad (3.21)$$

es una constante de normalización llamada *función de partición*, T es una constante de *temperatura*, debe suponerse que es 1 a menos que se indique lo contrario, y $U(f)$ es la *función de energía*. La energía está dada por:

$$U(f) = \sum_{c \in \mathcal{C}} V_c(f) \quad (3.22)$$

es la suma de *potenciales de clique* $V_c(f)$ sobre todos los cliques posibles \mathcal{C} . El valor de $V_c(f)$ depende sobre la configuración local del cliques c .

Cabe mencionar que para una función de probabilidad de ocurrencia $P(f)$, las configuraciones más probables son las que contienen energías $U(f)$ más bajas y la temperatura T es alta, todas las configuraciones tienden a estar igualmente distribuidas; por otro lado, cuando una temperatura se aproxima a cero, la distribución se concentra alrededor de los mínimos globales de energía.

El campo aleatorio de Markov se caracteriza por sus propiedades locales, mientras que el campo aleatorio de Gibbs se caracteriza por sus propiedades globales. El teorema Hammersley-Clifford establece la equivalencia entre estos dos campos.

El teorema de Hammersley-Clifford dice: F es un CAM en los estados de S con respecto a un sistema de vecinos \mathcal{N} si y sólo si F es un campo aleatorio de Gibbs en S con respecto a \mathcal{N} [21].

En el uso de la segmentación de imágenes y en la delimitación se basa en la siguiente función de energía [4]:

$$E(\alpha) = \sum_{i \in V} E_{\text{datos}}(\alpha_i) + \sum_{(i,j) \in \epsilon} E_{\text{suavizado}}(\alpha_i, \alpha_j) \quad (3.23)$$

donde

$$\sum_{i \in V} E_{\text{datos}}(\alpha_i) = \sum \frac{img - \mu}{\sigma} (img - \mu) + \log(\sigma) \quad (3.24)$$

donde μ es la media y σ la covarianza del área segmentada. Esta función representa la medida de ajuste entre la imagen original y las etiquetas asignadas a cada píxel. Esta función captura la información y características específicas de los datos observados, y

$$\sum_{(i,j) \in \epsilon} E_{\text{suavizado}}(\alpha_i, \alpha_j) = \sum V_c. \quad (3.25)$$

Esta función de energía modela la regularidad espacial o suavidad deseada en el área segmentada. Esta función penaliza las configuraciones de etiquetas que presentan

discontinuidades abruptas entre los píxeles vecinos. El objetivo es obtener una segmentación coherente y suave.

K-means

Los campos aleatorios de Markov requieren de una herramienta para etiquetar los datos, y una forma efectiva de clasificar estos datos en grupos es utilizando el algoritmo no supervisado de Clustering llamado *K-means*. Este algoritmo es ampliamente utilizado debido a que toma decisiones basadas en centroides [12]. El Clustering se puede entender como sigue: dado un conjunto de n objetos y una representación de ellos, se pueden formar K grupos basados en sus características y se agrupan en base a su similitud mutua.

Dada $X = \{x_i, i = 1, \dots, n\}$ donde X es un conjunto de objetos que son agrupados en K grupos, $C = \{c_k, k = 1, \dots, K\}$ [13]. Este algoritmo consiste en encontrar una partición tal que el error cuadrático entre la media empírica de un clúster y los puntos en el clúster sea minimizada. Dado μ_k es la media del clúster c_k . El error cuadrático entre μ_k y los puntos dentro del clúster c_k está definido como:

$$J(C_k) = \sum_{x_i \in C_k} \|x_i - \mu_k\|^2. \quad (3.26)$$

El objetivo de *K-means* es minimizar la suma de los errores cuadráticos de todos los clústeres K .

$$J(C_k) = \sum_{k=1}^K \sum_{x_i \in C_k} \|x_i - \mu_k\|^2. \quad (3.27)$$

Este algoritmo empieza con una partición inicial con K clúster y asigna patrones a los clústeres para reducir el error cuadrático [12], [16]. Los pasos principales del algoritmo *K-means* es el siguiente:

1. Definir el número de clústeres K en un conjunto $X = \{x_1, x_2, \dots, x_n\}$ de n – *dimensión*.
2. Inicializar los puntos de partida $v_1, v_2, \dots, v_k, \dots, v_K$, de manera aleatoria.
3. Asignación de datos más cercanos a v_i , usando la distancia Euclidiana métrica.
4. Recalcular v_i como centroides de sus datos cercanos, utilizando la fórmula dada por 3.27.
5. Si v_i no ha sido estabilizado, regresar al paso 3 y repetir el proceso hasta que el error cuadrático medio sea casi 0 o 0.

3.6. Pynq Z2

3.6.1. Tarjeta de desarrollo Pynq Z2

La tarjeta de desarrollo Pynq Z2 [27] (ver la Figura 3.10) es una plataforma de alto desempeño, del fabricante TUL[©], creada bajo el *Xilinx University Program* para el desarrollo de sistemas embebidos, cuenta con la compatibilidad del *Framework PYNQ* (*Python Productivity for Zynq*); aporta la facilidad de programación en lenguaje Python y oportunidad de aprovechar los recursos de los sistemas embebidos de alto rendimiento. Esta tarjeta es basa en un chip Zynq-7000 SoC (*System on Chip*), que combina un procesador ARM dual-core A9 [5] y una FPGA de Xilinx[1].

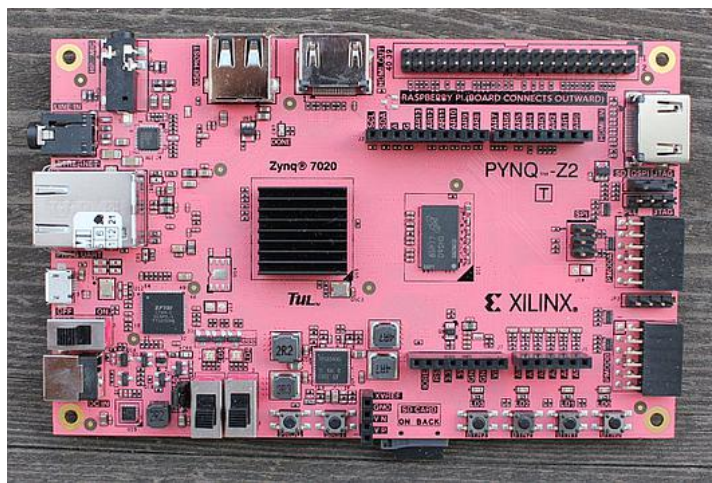


Figura 3.10: Tarjeta de desarrollo Pynq Z2.

Principalmente la arquitectura de un SoC (Figura 3.11) esta constituido por dos partes:

- El Sistema de Procesamiento (PS - *Processing System*): Está constituido por un procesador de doble núcleo ARM Cortex-A9, este es alojado físicamente en el chip, por lo que es considerado un *hard processor*.
- Lógica Programable (PL - *Programing Logic*): Esta sección es la correspondiente a un FPGA y para utilizarse se requiere de las herramientas de diseño como VHDL, HLS, u otros.

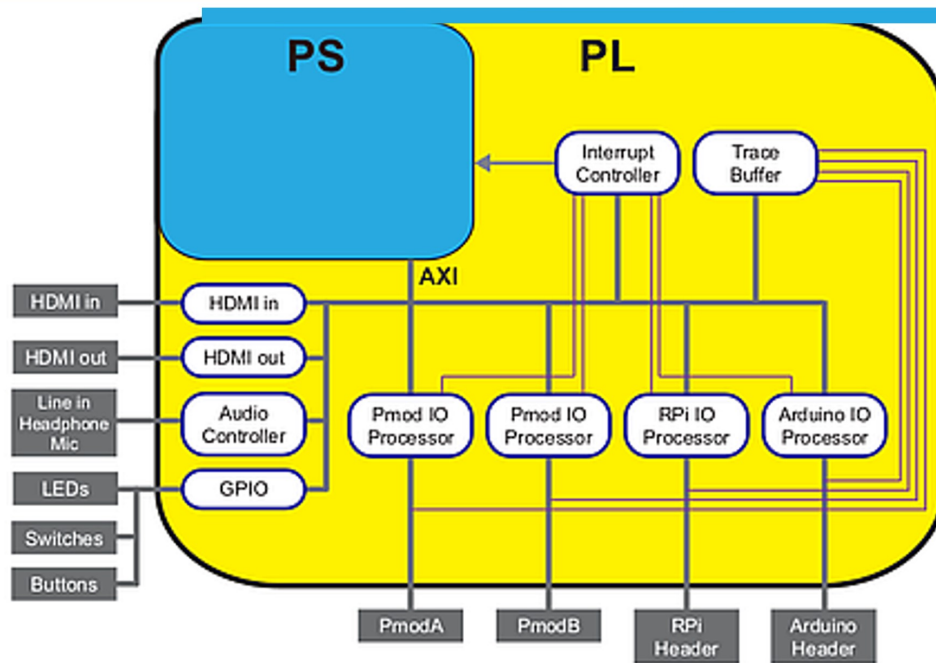
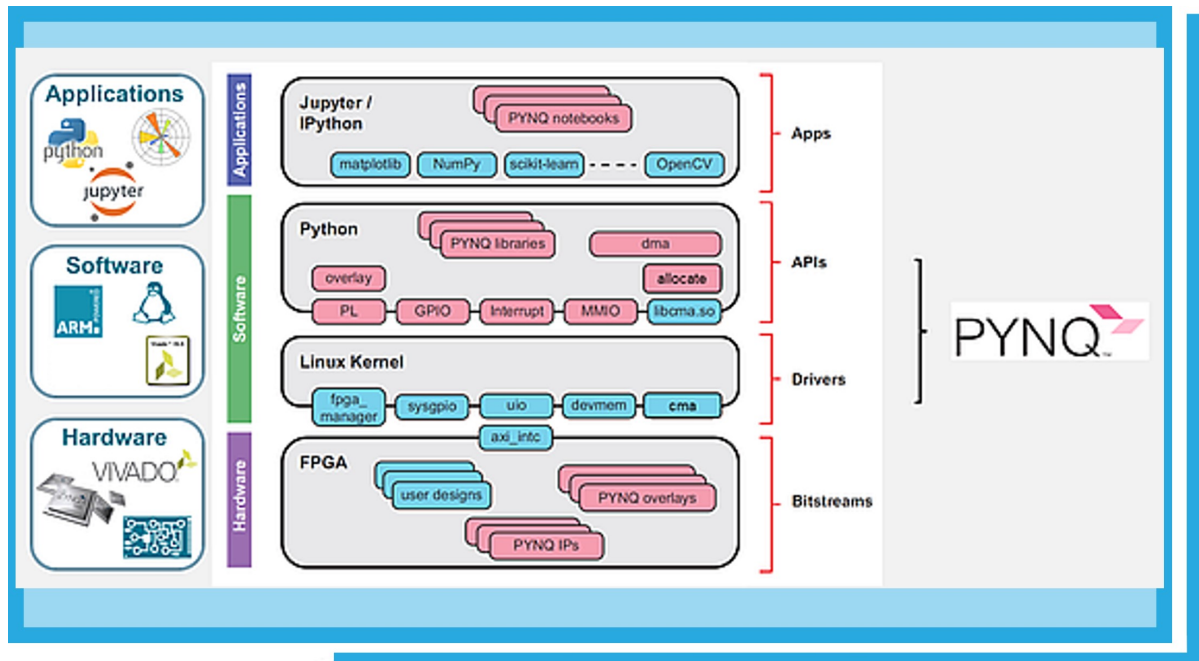


Figura 3.11: El SoC tiene la parte PS y la parte PL.

3.6.2. Especificaciones de la tarjeta

Entre las especificaciones sobresalientes se enuncian las siguientes:

- Es basada en el núcleo SoC XC7Z020-1CLG400C de la familia Zynq-7000.

- Contiene un procesador de doble núcleo ARM Cortex-A9, a 650MHz.
 - Lógica Programable.
 - 13,300 logic slices, cada una con 6 LUTs de 6 entradas y 8 flip-flops.
 - 630 KB de memoria block RAM.
 - 220 bloques DSP.
 - Convertidor analógico-digital a 1 MSPS *on-chip* (XADC) con seis entradas analógicas .
 - Permite la programación a través del JTAG, Quad - SPI flash o tarjeta MicroSD.
- Memoria y almacenamiento
 - Memoria DDR3 de 512MB con ancho de palabra de 16 bits a 1050Mbps.
 - 16MB Quad -SPI Flash con identificador programado de fábrica EUI-48/64© de 48 bits globalmente único.
 - Una ranura de expansión tipo MicroSD.
 - Alimentación
 - A través de puerto USB o por medio de regulador externo de 7V a 15V CD.
 - Conectividad USB y Ethernet.
 - Gigabit Ethernet PHY.
 - Puerto Micro USB para programación a través de JTAG.
 - Puerto Micro USB para comunicación a través de puente UART.
 - Puerto USB 2.0 OTG PHY (soporta modo host únicamente).
 - Switches, Push-Buttons y LEDs
 - 4 push-buttons.
 - 2 switches.
 - 4 LEDs.
 - 2 LEDs RGB.

La Pynq Z2 es una opción para realizar desarrollo de proyectos orientados a nivel licenciatura o posgrados tanto para principiantes como para expertos en sistemas embebidos, ya que permite diferentes niveles de abstracción a través de software especializado como MATLAB-HLS (*High Level Synthesis*)[22], o Labview [20]. Permite diseñar y desarrollar sistemas acelerados de manera intuitiva por hardware utilizando Python y herramientas de programación de nivel alto. Esto permite a los usuarios aprovechar la

flexibilidad y el rendimiento de la FPGA sin adentrarse en el diseño de hardware a un nivel más bajo [27].

Otra característica de la Pynq Z2 es su capacidad de reconfiguración dinámica de la FPGA. Esto significa que el diseño puede ser actualizado en cualquier momento y cambiar la lógica del hardware en tiempo de ejecución, lo que resulta especialmente útil para aplicaciones que requieren adaptabilidad y flexibilidad. Esto permite la implementación de algoritmos y funciones personalizadas de manera eficiente y optimizada [6].

Capítulo 4

Desarrollo y resultados

El desarrollo del algoritmo se realiza en diferentes etapas, como primer paso el algoritmo se implementa en MATLAB posteriormente se traslado a Python, en ambos lenguajes de programación funcionó correctamente el código, finalmente se opto por utilizar Python por ser un lenguaje de programación de libre acceso y finalmente se implementó en el algoritmo en la tarjeta de desarrollo Pynq Z2, también cuenta con librerías para el uso de la segmentación de imágenes.

4.1. Pseudo código CAM

En este apartado se presenta el pseudo algoritmo para el desarrollo del presente trabajo (ver Algoritmo 1), el funcionamiento del algoritmo es el siguiente:

1. Se introduce una imagen de entrada, no importa la dimensión o el tamaño siempre y cuando cumpla con formato .jpg, .jpeg, .bmp, se tiene que considerar que entre más grande el tamaño de la imagen el costo computacional será más grande por las cantidades de operaciones que realizaría el algoritmo.
2. Se necesita un valor de agrupamiento $K \geq 2$
3. Se transforma la imagen a escala de grises.
4. Para utilizar el algoritmo K-means es necesario transformar la matriz de entrada a vector.
5. Se inicializa el algoritmo K-means
 - a) Asignación de centroides, usando la distancia Euclidiana métrica.
 - b) Recalcular los centroides de sus datos cercanos, utilizando la fórmula dada por 3.27.

- c) Si los centroides no ha sido estabilizado, regresar al paso anterior y repetir el proceso hasta que el error cuadrático medio sea casi 0 o 0.
6. Calcular μ (media) y σ (covarianza) de cada grupo por separado.
7. Se calcula la función de energía de los datos para etiquetar los datos de cada grupo.
8. Se calcula la función de energía de suavizado para realizar el delimitado.
9. Se suma las funciones energía calculadas anteriormente
10. Los datos resultantes están en un vector, se regresan a su estado inicial (a una matriz).
11. Salida de la imagen procesada con resultados como son la delimitación, eliminación de ruido y mejor nitidez.

A continuación se ve reflejado el Pseudo-Código:

Algorithm 1 Pseudo-Código segmentación

Require: $K \geq 2$

Require: img

$img_{gray} \leftarrow Transformar(img)$

▷ Imágenes aceptadas .jpg, .jpeg o .bmp

$vector \leftarrow img_{gray}$

▷ Transformar imagen a escala de grises

▷ Transformar matriz a vector

Inicializar el algoritmo K-means

centroides = k-means(vector, K)

asignacion = asignar(vector, centroides)

$nuevos_{centroides} = recalcular(vector, asignacion)$

while $nuevos_{centroides} \neq 0$ **do**

▷ Cálculo de centroides

$nuevos_{centroides} = recalcular(vector, asignacion)$

end while

$media = calcular_{media}(vector)$

$covarianza = calcular_{cov}(vector)$

$suma_{datos} = 0$

for cada asignacion en vector **do**

▷ Función de energía características

$suma_{datos} = (asignacion - media)/cov + \log(cov)$

end for

$suma_{suavizado} = 0$

for cada asignacion en vecindad **do**

▷ Función de energía suavizado

$suma_{suavizado} = valor_{suavizado}$

end for

$energia_{total} = suma_{datos} + suma_{suavizado}$

▷ Suma de funciones de energía

$matriz_{mod} = convertir(vector, dim_{ini})$

▷ Transformación vector a matriz inicial

Salida: imagen modificada

4.2. Pruebas con imágenes conocidas

Se realizaron pruebas para determinar si el algoritmo otorgaba a la salida el resultado esperado. Las primeras pruebas que se desarrollaron fueron con diferentes imágenes, con diferentes formas y colores en el entorno de Python, los resultados se muestran a continuación:

La Figura 4.1 muestra una imagen de una pseudo figura, de baja resolución en tonalidades de grises, al aplicar el algoritmo a diferentes agrupamientos, por ejemplo: considerando a) imagen de entrada; en b) $K = 2$ se obtiene un rectángulo; en c) $K = 3$ el algoritmo, al no tener un grupo de vecinos de píxeles adecuado, no realiza la toma de decisiones correctamente, ya que carece de información; y en d) $K = 4$ al ser, de baja resolución y reducida en tonalidades de grises, el algoritmo entrega una imagen neutra, que carece de sentido, debido a que el algoritmo requiere de mayor información para operar.

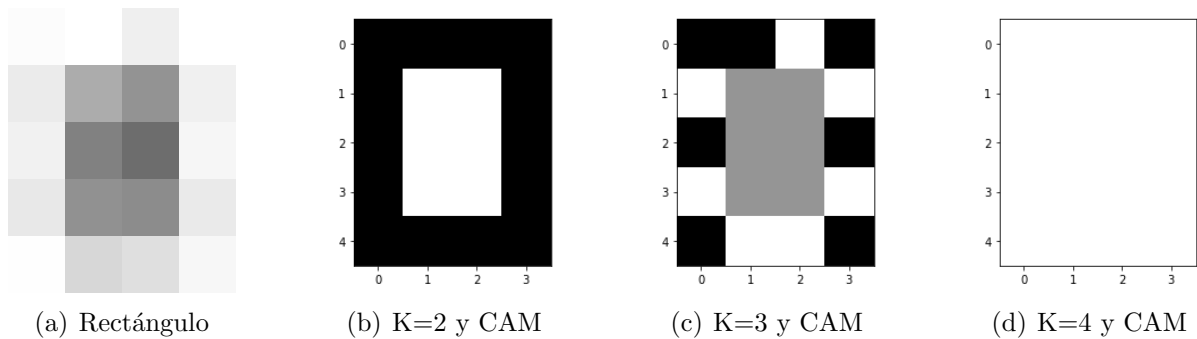


Figura 4.1: Aplicación CAM a una pseudo figura.

En la Figura 4.2 se introduce una imagen con más tamaño de píxeles, es un cromosoma "x" la imagen original a) Imagen de entrada, cuando b) $K = 2$ el algoritmo nos arroja la silueta del cromosoma, cuando c) $K = 3$ se empieza a observar la forma del cromosoma, d) $K = 4$ el algoritmo agrupa en 4 conjuntos la imagen y se observa que el cromosoma tendría la forma de una "X", e) $K = 5$ el cromosoma se observa más delimitado y tiene forma de "X" pero en f) $K = 6$ el cromosoma ya no se observa como una "X" empieza a perder forma.

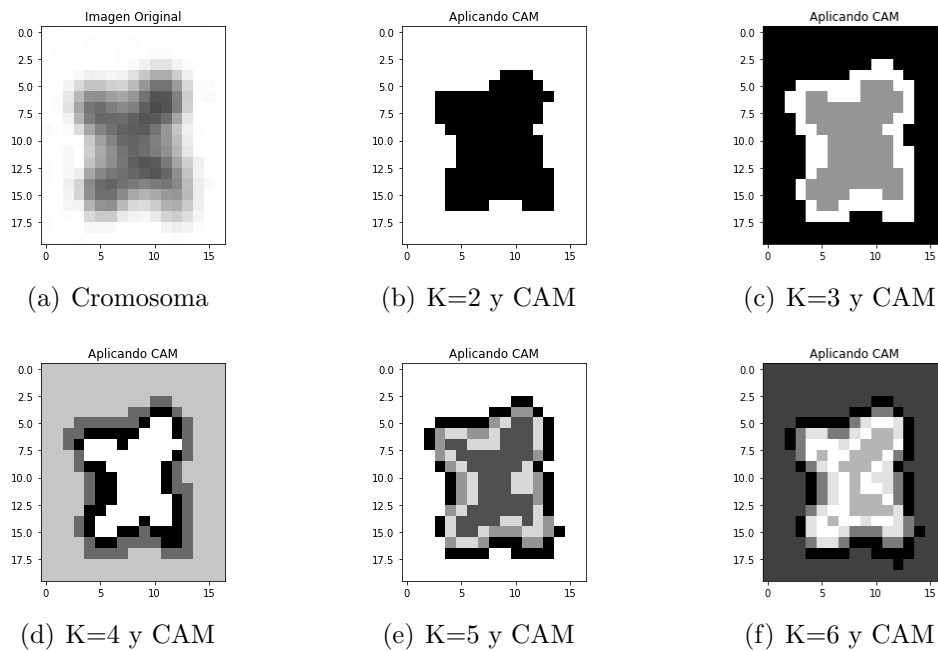


Figura 4.2: Aplicación CAM a un cromosoma.

En la Figura 4.3 se tomó como prueba una imagen con círculos de diferentes tonalidades de colores en esta imagen a) son 6: rojo, naranja, verde, verde claro, azul y amarillo, como la imagen se convierte en escala de grises, cuando b) $k = 2$ el algoritmo solo interpreta dos grupos, un grupo es el fondo y el segundo algunos círculos, cuando c) $K = 3$ se empiezan a observar más círculos, ya que las tonalidades que más se le parezcan al negro se incluirán (ya que la imagen está en escala de grises), cuando d) $K = 4$ el algoritmo interpreta más círculos y así va sucediendo con e) y f), en otras palabras el algoritmo interpreta los distintos colores en este caso depende de K .

Después de observar que los resultados obtenidos con la prueba se corroboró que el algoritmo esta funcionando de manera correcta, por consiguiente el algoritmo se implementó a la Pynq Z2.

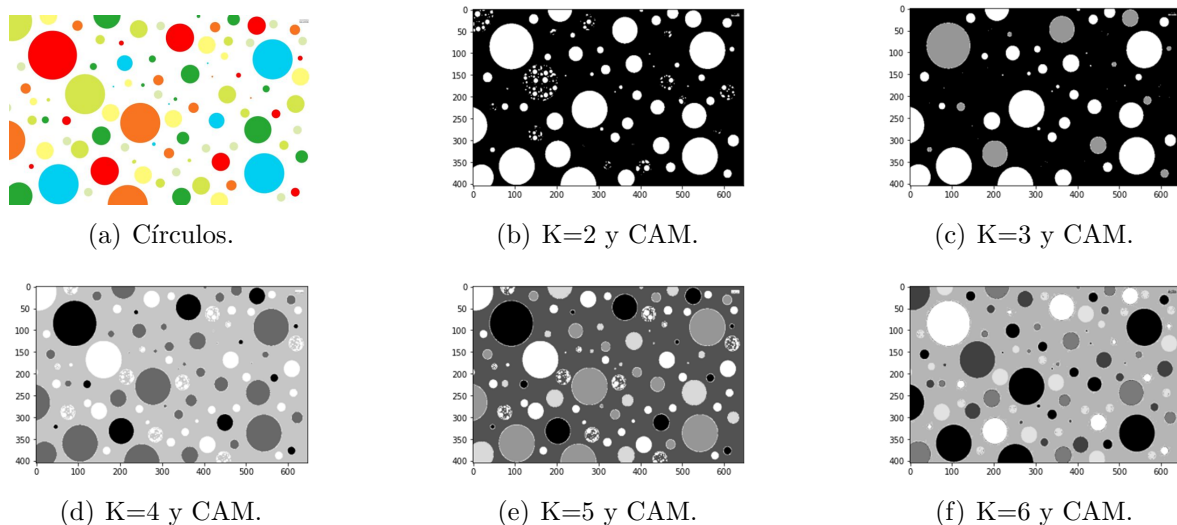


Figura 4.3: Aplicación CAM a círculos de colores.

4.3. Implementación algoritmo en FPGA Pynq Z2

A continuación se describirán los pasos que se siguieron para la implementación en la FPGA Pynq Z2, para utilizar la tarjeta se requiere una memoria micro SD para cargar el sistema operativo y trabajar en el entorno de Jupyter Notebook, posteriormente se conectara la tarjeta mediante un cable RJ45 a una computadora para transmisión de datos y una alimentación por medio de micro-USB para su utilización.

A continuación se mostrará la implementación del código en la Pynq Z2. En la Figura 3.10 se muestra la tarjeta de desarrollo Pynq Z2 del lado izquierdo inferior se encuentra conectado a la fuente de alimentación (5V) entrada micro-usb y del lado izquierdo superior un conector RJ45, se procede a encender la Pynq Z2.

Después de unos minutos de encender la tarjeta se debería de observa dos LED Azules y cuatro LED Verdes parpadear simultáneamente. Los LED Azules luego se encenderá y se apagarán mientras los LED Verdes permanecerán encendido, el sistema operativo ahora esta cargado y listo para su uso, como se observa en la Figura 4.4.

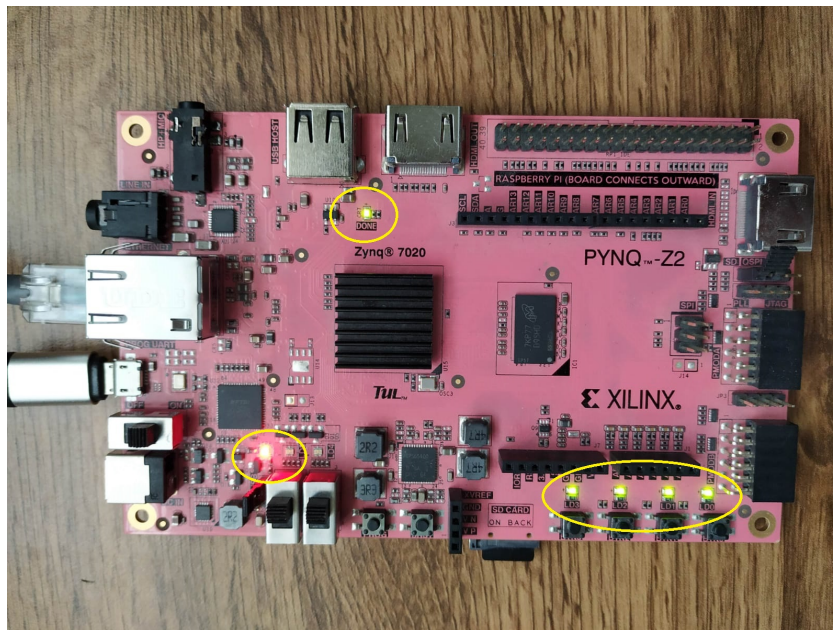


Figura 4.4: Inicialización sistema operativo Pynq Z2, leds encendidos.

En la Figura 4.5 se muestra la conexión de la Pynq-Z2 el cable de Ethernet conectado a la computadora, y se establece una conexión de red de manera directa para entrar al entorno de Jupyter Notebook mediante un navegador web de apoyo, en este trabajo se utilizo Firefox.

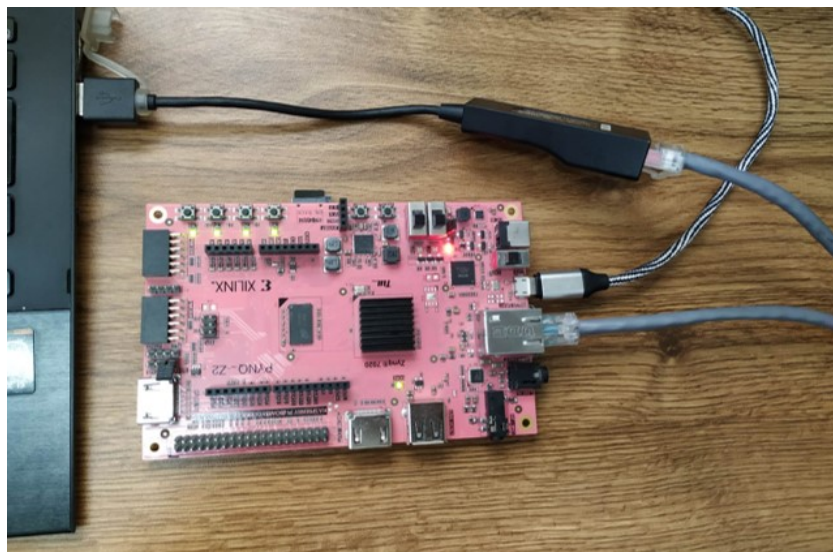


Figura 4.5: Conexión Pynq-Z2 a computadora.

En la Figura 4.6 se observa el entorno de trabajo de la tarjeta en la computadora,

es aquí donde se ha establecido una comunicación entre la Pynq Z2 y la computadora mediante el conector RJ45, después de tener una correcta conexión se carga el archivo del algoritmo.

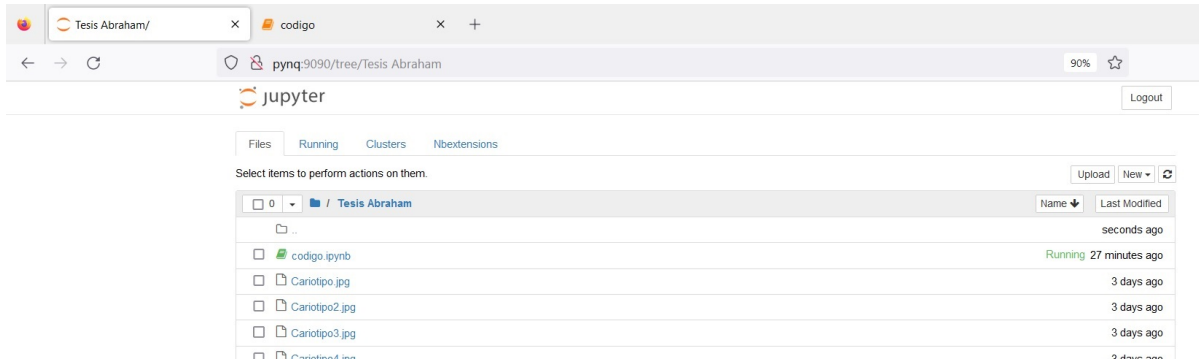


Figura 4.6: Entorno de trabajo Pynq Z2.

Después de cargar los archivos a la tarjeta de desarrollo se procede a ejecutar el programa con formato .py y las imágenes a utilizar, cabe mencionar que el algoritmo solo trabaja con una imagen a la vez, en la Figura 4.7 se puede observar el resultado de algoritmo realizado en el entorno de trabajo de la FPGA Pynq Z2.

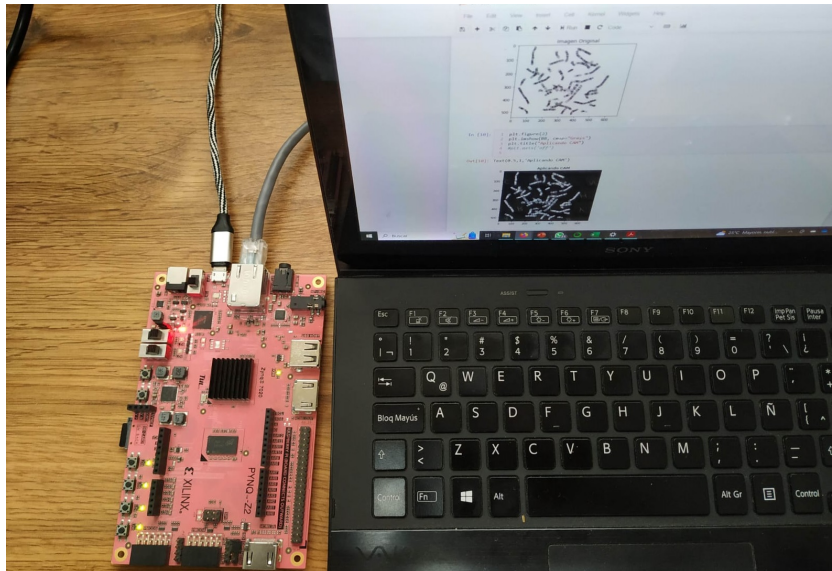


Figura 4.7: Algoritmo implementado en la Pynq Z2.

4.4. Resultados

Una vez finalizada la implementación de las ecuaciones para el CAM y las pruebas correspondientes en Python, ahora es posible obtener los resultados de un conjunto de imágenes cromosómicas obtenidas de un laboratorio de genética, las cuales fueron adquiridas siguiendo condiciones de confiabilidad permanentes cabe mencionar que los resultados obtenidos se realizaron en una computadora y la Pynq Z2. Se presentarán los resultados obtenidos del algoritmo propuesto tras realizar las operaciones con las imágenes de los cromosomas.

En la Figura 4.8 se puede observar que al aplicar la metodología propuesta a la imagen de cromosomas, b) donde $K = 2$ resalta el fondo de los cromosomas, c) donde $K = 3$ se empieza a apreciar las bandas de los cromosomas y mejoramiento de la imagen d) donde $K = 4$ resaltan puntos entre las bandas de los cromosomas e) cuando $K = 5$ empieza a mostrar aún más detalles, pero ya se empieza a observar el ruido de la imagen y no se aprecia del todo finalmente e) donde $K = 6$ las líneas que anteriormente pareciera que se unen y no se puede analizar del todo la imagen.

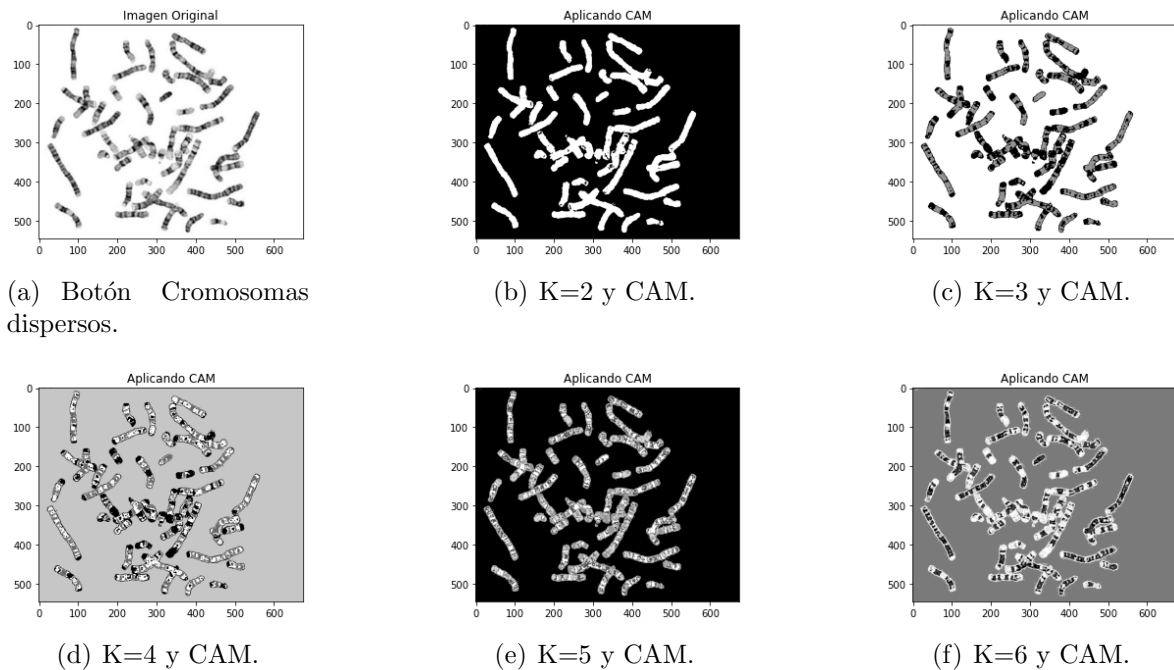


Figura 4.8: Aplicación CAM a un Cariotipo antes de ser ordenado.

En la Figura 4.9 a) es la imagen original de un cromosoma, sin modificación alguna, b) donde $K = 2$ se puede observar la diferencia del fondo al del objeto a percibir, en este caso el cromosoma, c) cuando $K = 3$ se empieza a notar las bandas del cromosoma d) cuando $K = 4$ las bandas del cromosoma aparecen mucho más definidas e) cuando $K = 5$ se delimita aún mejor las bandas, se observan más detalles, f) sin embargo, cuando $K = 6$ la imagen se empiezan a observar con ruido.

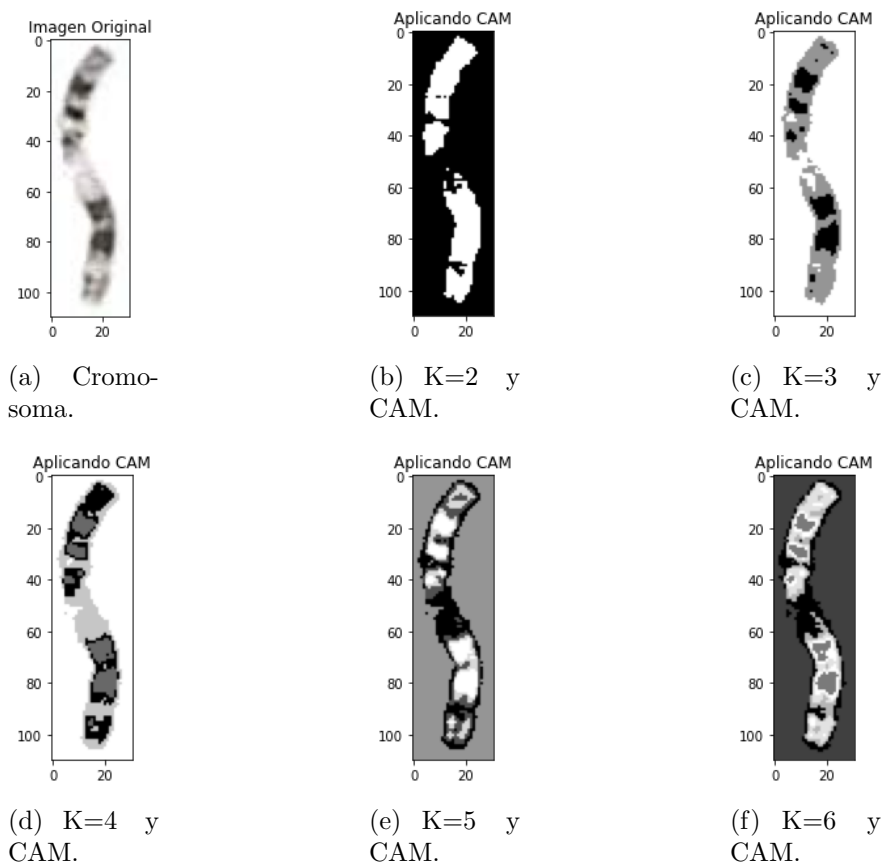
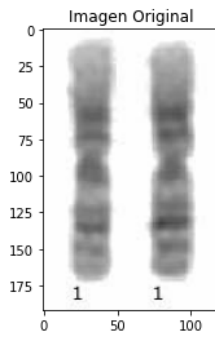


Figura 4.9: Aplicación CAM a un cromosoma en baja resolución.

En la Figura 4.10 se muestra un par de cromosomas, b) cuando $K = 2$ solo se puede destacar los dos cromosomas del fondo, sin detalle alguno, c) cuando $K = 3$ no se muestra buen detalle de bandas, d) cuando $K = 4$ las bandas aún quedan entrelazadas, no se encuentran detalles sobre las bandas, e) cuando $K = 5$ las formas de las bandas empiezan a tener más forma y se pueden apreciar más delimitadas, f) cuando $K = 6$ se nota que las bandas tienen un poco de ruido.



(a) Par Cromosomas.

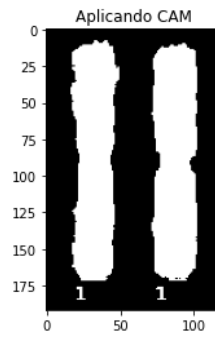
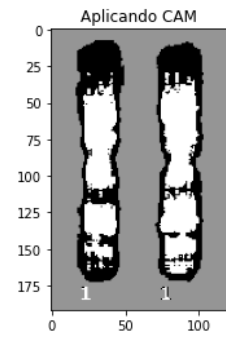
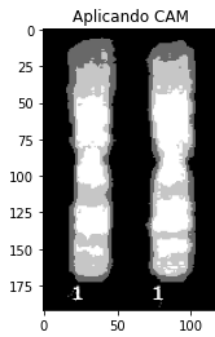
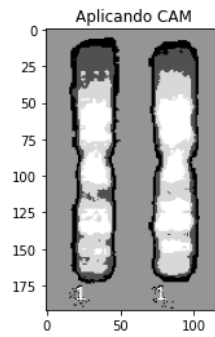
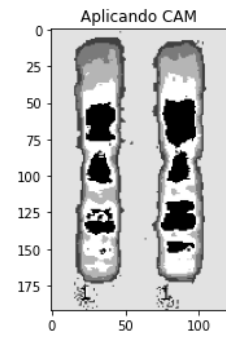
(b) $K=2$ y CAM.(c) $K=3$ y CAM.(d) $K=4$ y CAM.(e) $K=5$ y CAM.(f) $K=6$ y CAM.

Figura 4.10: Aplicación CAM a un par de cromosomas. Imágenes tomadas [30].

En la Figura 4.11 a) se observa pocos cromosomas y la imagen se encuentra desenfocada, b) cuando $K = 2$ se observa el fondo y las siluetas de los cromosomas, c) cuando $K = 3$ se empieza a definir mejor el borde de los cromosomas, d) cuando $K = 4$ las bandas de los cromosomas se observan, e) cuando $K = 5$ las bandas de los cromosomas se traslapa las bandas, f) cuando $K = 6$ se nota mejoría en bordes, sin embargo, no se puede llevar un análisis en las bandas de los cromosomas.

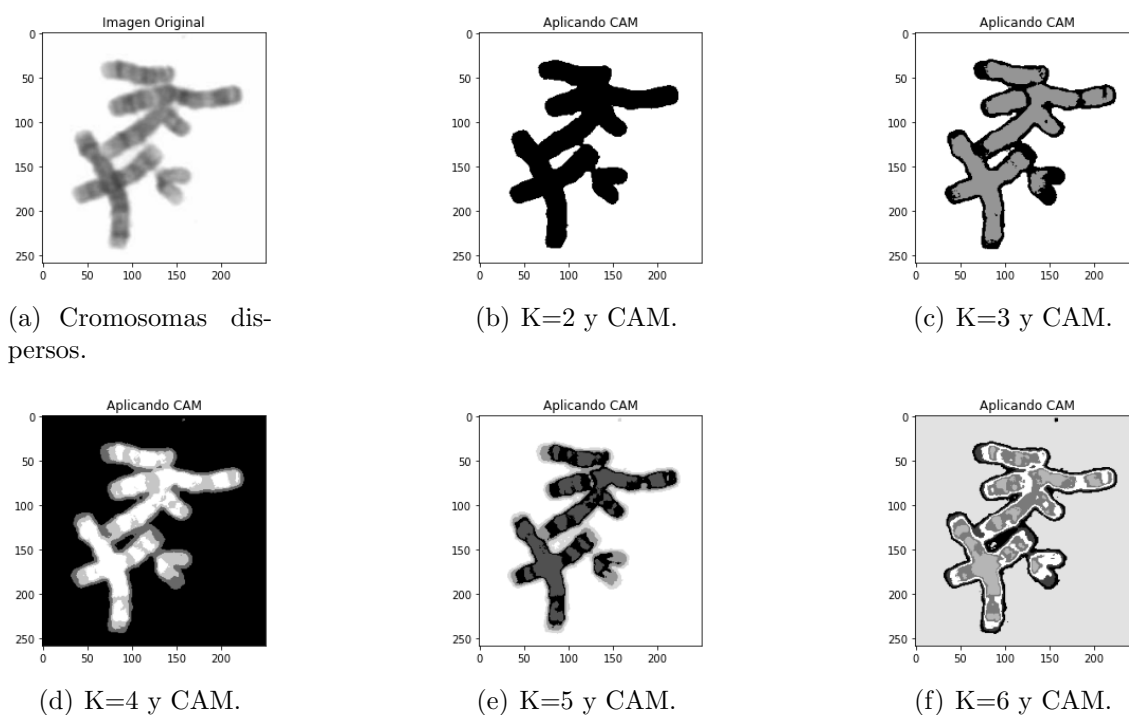


Figura 4.11: Aplicación CAM a cromosomas. Imágenes de [30].

Los siguientes resultados de un banco de imágenes¹: en la Figura 4.12 a) es la imagen original mostrando un conjunto de cromosomas, b) cuando $K = 2$ se observa el fondo y las siluetas de los cromosomas, c) cuando $K = 3$ se empieza a definir mejor el borde de los cromosomas, d) cuando $K = 4$ las bandas de los cromosomas se observan y los centrómeros, e) Si $K = 5$ se observan más definidas las delimitaciones de los cromosomas del fondo, f) Con $K = 6$ se nota mejoría en bordes y en bandas.

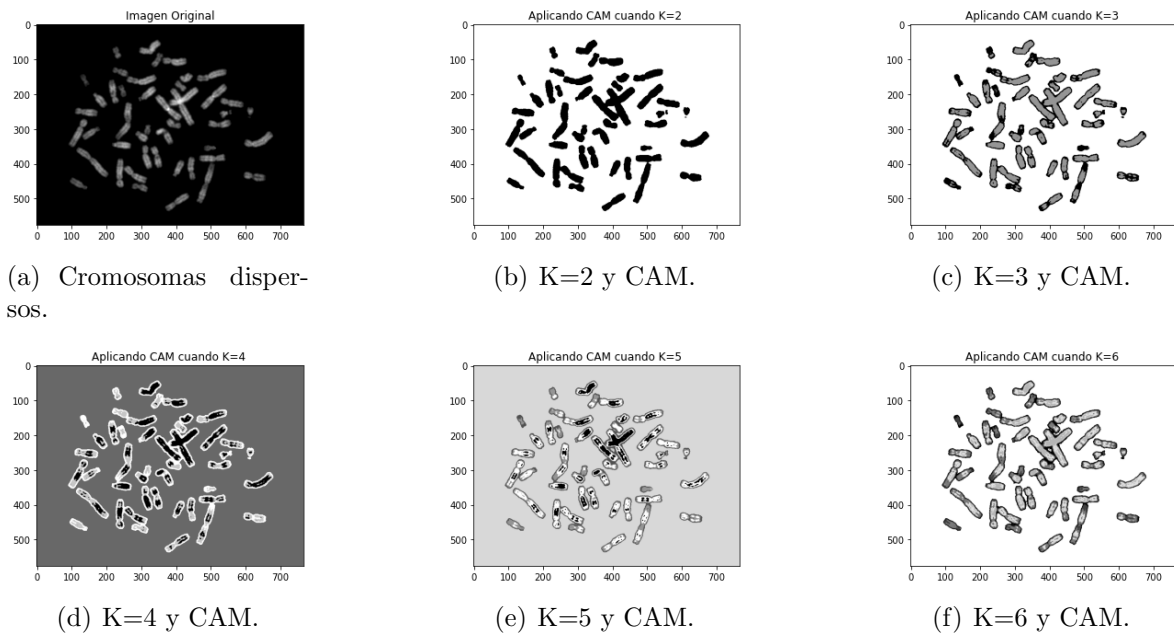


Figura 4.12: Aplicación CAM a cromosomas.

¹El conjunto de datos está a disponible al público para su descarga en <https://bioimlab.dei.unipd.it/>

En la Figura 4.13 a) se muestra la imagen original, b) cuando $K = 2$ se puede observar el fondo y las siluetas de los cromosomas, c) cuando $K = 3$ se empieza a definir las delimitaciones de los cromosomas, d) cuando $K = 4$ las bandas de los cromosomas y los centrómeros se empiezan a definir, e) cuando $K = 5$ se observan más definidas las delimitaciones de los cromosomas del fondo, f) cuando $K = 6$ no se ve mucha diferencia respecto al inciso e).

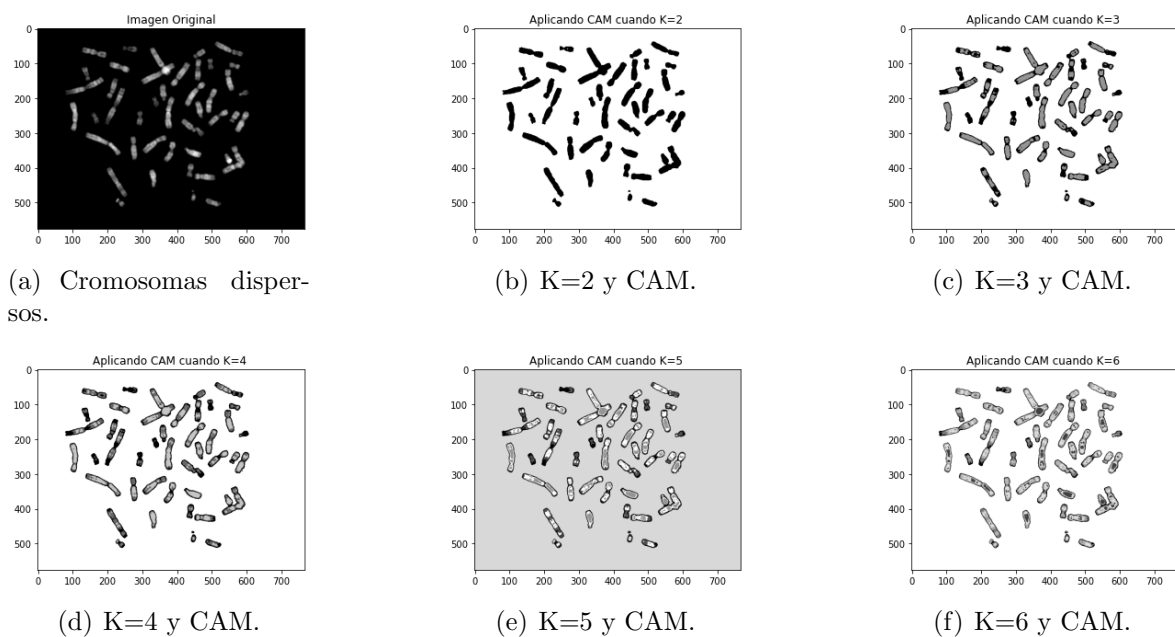


Figura 4.13: Aplicación CAM a cromosomas.

Conclusiones

Se propuso e implementó una metodología basada en Campos Aleatorios de Markov para abordar la segmentación de cromosomas. Con el fin de lograrlo, se propuso utilizar la segmentación por agrupación apoyándose en el algoritmo K-means.

Con esta técnica se obtuvieron resultados satisfactorios que demostraron una mejora en la calidad de las imágenes procesadas. La capacidad de ajustar los parámetros de entrada permite modificar la salida de la imagen, lo que resalta de mejor forma las siluetas y las bandas cromosómicas.

Se ha implementado con éxito el algoritmo en la tarjeta Pynq Z2 y los resultados coinciden con los obtenidos en Python ejecutado en una computadora, dicho esto la tarjeta de desarrollo es viable para la implementación del algoritmo. Por otro lado, su flexibilidad permite la mejora y optimizar aún más el rendimiento.

La implementación de esta metodología ofrece una herramienta orientada hacia su uso en el análisis de cariotipos. Esto podría contribuir a una mejor comprensión de las características genéticas y al diagnóstico de enfermedades relacionadas con alteraciones cromosómicas, lo que lo hace un método de gran impacto en el área médica.

Bibliografía

- [1] *AMD together we advance_{AI}*. 2020. URL: <https://www.amd.com/en.html>.
- [2] Sandra Díaz Barriga Arceo y Rosalba Bonilla Sánchez. *Citogenética humana*. Universidad Nacional Autónoma de México.
- [3] Adrian Baddeley y col. *Spatial Point Patterns Spatial Point Patterns Methodology and Applications with R Spatial Point Patterns*. 1.^a ed. Chapman y Hall/CRC, 2015.
- [4] *COMPUTER VISION FOR VISUAL EFFECTS*. Cambridge University Press, dic. de 2012.
- [5] Cortex-A9. *Cortex-A9*. 2016. URL: <https://developer.arm.com/Processors/Cortex-A9>.
- [6] Louise H. Crockett y col. *The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc*. Glasgow, GBR: Strathclyde Academic Media, 2014. ISBN: 099297870X.
- [7] Luis Enrique y Ledezma Fuentes. “Reconocimiento de patrones para estructuras cromosómicas”. En: (2013).
- [8] Cuevas Jiménez Erik, Zaldívar Navarro Daniel y Pérez Cisneros Marco Antonio. *Procesamiento digital de imágenes usando MatLab Simulink*. 1. ed. México, D.F.: Alfaomega, 2010, pág. 815.
- [9] Rafael C. Gonzalez y Richard E. Woods. *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008, págs. –. ISBN: 9780131687288 013168728X 9780135052679 013505267X. URL: http://www.worldcat.org/search?qt=worldcat_org_all&q=013505267X.
- [10] Oscar Gualdrón-González y col. “Obtención de Cariotipos Procesamiento Digital de Imágenes Aplicaciones al Área de la Citogenética”. En: *Energía y Computación* 11 (oct. de 2011), págs. 22-26.
- [11] Osvaldo Gutiérrez. “Segmentación Robusta de Imágenes Mediante Campos Aleatorios de Markov y Estimación de la Entropía”. Universidad Autónoma de Zacatecas, dic. de 2012.

- [12] Anil K. Jain. “Data clustering: 50 years beyond K-means”. En: *Pattern Recognition Letters* 31 (8 jun. de 2010), págs. 651-666. ISSN: 01678655. DOI: 10.1016/j.patrec.2009.09.011.
- [13] Anil K. Jain. “Data clustering: 50 years beyond K-means”. En: *Pattern Recognition Letters* 31.8 (2010). Award winning papers from the 19th International Conference on Pattern Recognition (ICPR), págs. 651-666. ISSN: 0167-8655. DOI: <https://doi.org/10.1016/j.patrec.2009.09.011>. URL: <https://www.sciencedirect.com/science/article/pii/S0167865509002323>.
- [14] Ramesh Jain, Rangachar Kasturi y Brian G. Schunck. *Machine Vision*. USA: McGraw-Hill, Inc., 1995. ISBN: 0070320187.
- [15] Carla Maria Alonso Jané, Goar Orúe Sánchez Miriela Escobedo Nicot y Alexeis Galano Compte. “Comparativa de métodos de segmentación para el análisis automático de cromosomas humanos”. En: *Orange Journal* 3 (5 nov. de 2021), págs. 53-62. DOI: 10.46502/issn.2710-995x/2021.5.06.
- [16] M Jordan, J Kleinberg y B Schölkopf. *Pattern Recognition and Machine Learning*.
- [17] Nicolás Jouve, Angeles Cuadrado y Pilar Rubio. “Aplicaciones genéticas de las técnicas de análisis de imagen”. En: (1994).
- [18] Zoltan Kato y Josiane Zerubia. “Markov random fields in image segmentation”. En: *Foundations and Trends in Signal Processing* 5 (1-2 2011), págs. 1-155. ISSN: 19328346. DOI: 10.1561/20000000035.
- [19] Nora La Serna Palomino y Ulises Norberto Román Concha. “Técnicas de Segmentación en Procesamiento Digital de Imágenes”. En: *Revista de investigación de Sistemas e Informática* 6.2 (dic. de 2009), págs. 9-16. URL: <https://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/3299>.
- [20] LabiewWeb. *¿Qué son los procesos metacognitivos?* 2019. URL: <https://www.ceupe.com/blog/que-son-los-procesos-metacognitivos.html>.
- [21] Stan Z. Li. *Markov Random Field Modeling in Image Analysis*. 3rd. Springer Publishing Company, Incorporated, 2009. ISBN: 9781848002784.
- [22] Matlabweb. *High-Level Synthesis - MATLAB Simulink*. 2016. URL: <https://www.mathworks.com/discovery/high-level-synthesis.html>.
- [23] Rubén Medina y Jesús Bellera. *Bases del Procesamiento de Imágenes Médicas*. Universidad de los Andes, 2009. URL: <http://www.ing.ula.ve/~gibula/>.
- [24] Robert F Mueller. *Genética médica*. 575.1 MUE. Young, Ian D., 2001.
- [25] Viridiana Rubí Calzada Navarrete. “Segmentación Automática de Imágenes de Cromosomas”. Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, nov. de 2012.

- [26] Myriam Ortega, Jose Torres y Julialba Ángel. *Fundamentos de citogenética humana y animal*. 1.^a ed. Sep. de 2020, págs. 34-64. URL: <https://www.omim.org/>.
- [27] pynqweb. *PYNQ Introduction — Python productivity for Zynq (Pynq)*. URL: <https://pynq.readthedocs.io/en/v2.5/index.html>.
- [28] Alejandro Domínguez Torres. “Procesamiento Digital de Imágenes”. En: *Perfiles Educativos* (abr. de 1996). ISSN: 0185-2698.
- [29] Peter Turnpenny y Sian Ellard. *Emery’s Elements of Medical Genetics*. 15.^a ed. Elsevier, 2018.
- [30] Test Video. *VIDEOTEST-KARYO 3.1*. Saint-Petersburg: VideoTest,ltd, 1999. URL: <http://www.videotest.ru>.

Apéndice A

Código

En Listing A.1 se muestra el código final en el lenguaje de programación Python.

Listing A.1: Código.

```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt

K = 6
i = 1
potencial = 0.5
imagen = cv.imread('original_12a.bmp',1)
gray = cv.cvtColor(imagen, cv.COLOR_BGR2GRAY)
(w, h) = np.shape(gray)
grayf = np.float32(gray)

n = w*h
m = w*h
seq = np.reshape(grayf, [n,1])
criteria = (cv.TERM_CRITERIA_EPS +
cv.TERM_CRITERIA_MAX_ITER, 10, 1.0)

ret, label, center=cv.kmeans(seq, K, None,
criteria, 10, cv.KMEANS_RANDOMCENTERS)

### Modelo Guassiano Mixto
mu = np.zeros((K,1))
mu = center
```

```

sigma = np.zeros((1,1,K))
for i in range(K):
    mu_i=mu[i,]
    Im_i = seq[label==i,]
    Ki = np.shape(Im_i)
    eje = np.ones((Ki))
    mu_arr = np.kron(eje, mu_i)
    Im_i = Im_i - mu_arr
    Kir = np.double(Ki)
    sig = (np.dot(Im_i.T, Im_i))/(Kir-1)
    sigma[... , i] = sig

### Campo de energia de características

E = np.zeros((n,K))
for i in range(K):
    mu_i=mu[i,]
    sigma_i = sigma[... , i]
    ej = np.ones((n))
    museq = np.kron(ej, mu_i)
    museq = np.float32([museq])
    diff_i = seq - museq.T
    s_inv = np.linalg.inv(sigma_i)
    s_det = np.linalg.det(sigma_i)
    log_s = np.log(s_det)
    mul = diff_i*s_inv*diff_i
    E[:, i]=mul.sum(axis=1)+log_s

### Vecindad

(m,n) = np.shape(label)
segmentacion = np.reshape(label, [w,h])
X = segmentacion
[s, t] = np.shape(X)
#### X Arriba Izquierda
XAI = np.zeros([s, t])
XAI[2:s, 2:t,] = X[1:s-1, 1:t-1,]
#### X Arriba
XA = np.zeros([s, t])
XA[2:s, ...] = X[1:s-1, ...]
#### X Arriba Derecha
XARD = np.zeros([s, t])

```

```

XARD[2:s,1:t-1,] = X[1:s-1,2:t,]
##### X Derecha
XD = np.zeros([s,t])
XD[:,1:t-1,] = X[:,2:t,]
##### X Abajo Derecha
XABD = np.zeros([s,t])
XABD[1:s-1,1:t-1,] = X[2:s,2:t,]
##### X Abajo
XAB = np.zeros([s,t])
XAB[1:s-1,...] = X[2:s,...]
##### X Abajo Izquierda
XABI = np.zeros([s,t])
XABI[1:s-1,2:t,] = X[2:s,1:t-1,]
##### X Izquierda
XI = np.zeros([s,t])
XI[:,2:t,] = X[:,1:t-1,]

Z = np.array((XAI,XA,XARD,XD,XABD,XAB,XABI,XI))

Vec8 = np.reshape(Z,[w*h,8])

E2 = np.zeros([m,K])
for i in range(K):
    E2[:,i] = np.sum(Vec8!=i,1)

E3 = E2*potencial
Et = E + E3

label2 = np.min(Et,1)
Etmin = Et.argmax(1)

B = Etmin.reshape([w, h])
B8 = np.uint8(B)

plt.figure(1)
plt.imshow(imagen, cmap="Greys")
plt.axis('off')
plt.title("Imagen_Original")
plt.figure(2)

```

```
plt.imshow(B8, cmap="Greys")  
#plt.axis('off')  
plt.title(f'Aplicando CAM cuando K={K}')
```