Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Electrónica

Maestría en Ingeniería Electrónica, opción Instrumentación Electrónica

Desarrollo de un sistema automatizado de análisis de imágenes para la construcción de cariogramas humanos

Tesis presentada para obtener el grado de:

Maestría en Ingeniería Electrónica

Presenta:

Lic. Abraham Gilberto Díaz Nayotl

Directora de Tesis:
Dra. María Monserrat Morín Castillo (FCE - BUAP)

Directora de Tesis:
Dra. Bolivia Teresa Cuevas Otahola (FCE - BUAP)

Director de Tesis:
Dr. José Rubén Conde Sánchez (FCFM - BUAP)

Becario CONAHCYT*

Enero 2026, Puebla, Méx.

Agradecimientos

En primer lugar, agradezco al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) por el apoyo económico brindado a lo largo del programa de maestría.

A la Benemérita Universidad Autónoma de Puebla (BUAP), institución en la que realicé mis estudios, y a la Facultad de Ciencias de la Electrónica, junto con todos sus profesores, por contribuir de manera significativa a mi formación académica.

Agradezco también al Instituto Mexicano del Seguro Social (IMSS) HGZ No. 20, institución en la que realicé mi estancia, por las facilidades brindadas y el aprendizaje adquirido durante ese periodo.

Expreso mi más sincero reconocimiento a mis asesores de tesis: la Dra. María Monserrat Morín Castillo, la Dra. Bolivia Teresa Cuevas Otahola y el Dr. José Rubén Conde Sánchez, por su apoyo, conocimiento, orientación y confianza durante el desarrollo de este trabajo.

Asimismo, agradezco a la Mtra. Ana María Rodríguez Domínguez, al Dr. Arnulfo Luis Ramos y al Dr. José Elogio Moisés Gutiérrez Arias, por su seguimiento durante todo el proceso.

De manera especial, extiendo mi agradecimiento a la Dra. Daniela Juárez Melchor por su apoyo, tiempo y asesoría en el área genética, así como a la Dra. Laura Daniel Mora por su paciencia, apoyo y orientación en la metodología y en la presentación del trabajo.

A mis padres y hermanos, por su apoyo incondicional, cariño y palabras de aliento que me impulsaron a seguir adelante.

A mi novia, por su confianza, compañía y respaldo en cada etapa de este camino.

Y finalmente, a mis amigos, por su apoyo y motivación constante.

Dedicatoria

A mi familia.

Índice general

Índice de figuras

Resumen

La elaboración del cariograma humano es una etapa fundamental en el análisis citogenético para la detección de anomalías cromosómicas numéricas y estructurales. Sin embargo, este proceso se realiza tradicionalmente de forma manual, lo que lo convierte en una tarea compleja, repetitiva y susceptible a errores humanos. Ante esta problemática, en la presente tesis se desarrolla un sistema automatizado de análisis de imágenes para la segmentación, discriminación, clasificación y ordenamiento de cromosomas humanos a partir de imágenes de células en metafase, con el objetivo de optimizar el proceso de construcción del cariograma. El sistema propuesto se estructura en dos etapas principales. En la primera etapa se lleva a cabo el preprocesamiento y segmentación de las imágenes cromosómicas, donde se comparan dos métodos de binarización: el método de Otsu y un enfoque basado en K-means combinado con Campos Aleatorios de Markov (K-CAM). Posteriormente, se implementa un proceso de reducción de ruido mediante operaciones morfológicas y filtrado por área. Para la discriminación entre cromosomas individuales y agrupados, se emplea un esquema de tres filtros complementarios basado en la solidez del objeto, la excentricidad de una elipse envolvente y la esqueletización con detección de puntos terminales, lo que permite una identificación de los cromosomas candidatos a clasificación. En la segunda etapa se desarrolla un modelo de red neuronal convolucional (CNN) para la clasificación automática de cromosomas individuales en 24 clases correspondientes a los 22 pares autosómicos y los cromosomas sexuales X y Y. El modelo es entrenado con imágenes previamente normalizadas y acondicionadas, alcanzando una alta exactitud durante el entrenamiento y un desempeño competitivo en validación, considerando el tamaño reducido de la base de datos utilizada. A partir de las predicciones del modelo, se implementa un algoritmo de ordenamiento automático que permite el ordenamiento de los cromosomas, el cual es presentado mediante una interfaz gráfica de usuario diseñada para facilitar la interacción con el sistema. El sistema completo es implementado y validado en una tarjeta de desarrollo Jetson Nano, demostrando la capacidad de ejecutar algoritmos de visión por computadora e inteligencia artificial en plataformas embebidas con recursos limitados. Los resultados obtenidos muestran que el sistema propuesto reduce el tiempo de análisis a menos de cinco minutos por muestra, manteniendo métricas superiores al 90 % en la discriminación de cromosomas y una clasificación confiable.

Capítulo 1

Introducción

La genética es una de las ramas de la biología que se encarga del estudio y análisis de los genes[1], los cuales contienen la información que los organismos trasmiten en su descendencia, herencia y variación. Esta disciplina se enfoca en la composición y organización del ADN, que es la estructura en los genes.

Además, la genética investiga cómo los genes se condensan en **cromosomas** y cómo se transmite la herencia de características biológicas de una generación a otra. Las alteraciones y enfermedades relacionadas con los cromosomas son objeto de estudio de la citogenética. Los cromosomas son estructuras delgadas y alargadas que contienen material genético y se localizan en el núcleo de las células.

En la actualidad, la citogenética se apoya en los avances tecnológicos que permiten realizar diagnósticos más prontos y eficientes en el ámbito de la salud. A través de diversas técnicas de análisis, la citogenética permite la identificación de anomalías cromosómicas que pueden estar relacionadas con enfermedades transmitidas por medio de los genes.
Estas anomalías en humanos se clasifican en dos grupos: numéricas y estructurales. Las anomalías numéricas se refieren a cambios en el total de cromosomas, que a su vez son:

1. Monosomía, ausencia de un cromosoma.

2. Trisomía, presencia de un cromosoma adicional.

En cuanto a las anomalías estructurales, estas se refieren a modificaciones en la forma o la organización de los cromosomas, abarcando fenómenos como:

1. Deleciones, pérdida de segmentos.

2. Duplicaciones, copias adicionales de segmentos.

3. Inversiones, reordenamiento de segmentos.

4. Translocaciones, intercambio de segmentos entre cromosomas.

[1]Mínima unidad de información que puede ser heredable.

Todas las especies con células poseen un número específico de cromosomas; en el caso de los humanos, estos se encuentran en las células somáticas[2]. Los humanos presentan 46 cromosomas organizados en 23 pares. De estos pares, uno corresponde a los cromosomas sexuales, mientras que los otros 22 pares son conocidos como autosomas y no están relacionados con la determinación del sexo [1], [2].

El proceso más utilizado para el análisis cromosómico es el cariotipo. Este proceso permite analizar y evaluar tanto el número como la estructura de los cromosomas, con el objetivo de detectar posibles anomalías. El proceso implica la ordenación y clasificación de los cromosomas, lo que facilita la identificación de alteraciones que puedan afectar el estado del individuo [3].

Para realizar el proceso del cariotipo, es necesario obtener imágenes de los cromosomas. El especialista sigue los siguientes pasos:

1. Se toma una muestra de algún tejido vivo, como una pequeña muestra de sangre.

2. Posteriormente se realiza un proceso de cultivo celular con el objetivo de realizar la mitosis el cual consiste en dividir las células.

3. Una vez que las células alcanzan la metafase[3], se extraen pequeñas muestras utilizando un gotero y se dejan caer desde cierta altura sobre un portaobjetos de vidrio. Esta acción provoca la ruptura de las células, lo que facilita la dispersión de los cromosomas y su posterior visualización al microscopio. Posteriormente, se aplica una técnica de teñido —como el bandeo G, Q, R o C— que permite resaltar patrones específicos en los cromosomas y facilita su análisis estructural.

Todo este proceso se muestra en la Figura 1.1.

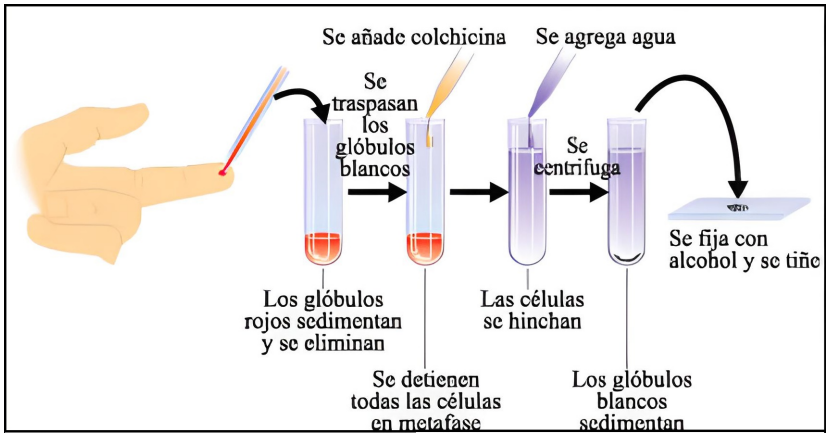


Figura 1.1: Preparación de la toma de muestra.

[2]Todas las células del cuerpo humano, excepto óvulos y espermatozoides.

[3]Una de las fases de la mitosis en la que los cromosomas están más condensados y alineados.

4. El portaobjetos se coloca bajo el microscopio electrónico para realizar un mapeo y capturar microfotografías. En este paso, se obtienen un gran número de microfotografías de las células en metafase, tanto intactas o con ruptura mostrando a los cromosomas.

5. Se arma el banco de imágenes capturadas por el microscopio para almacenarlas en una computadora

6. Posteriormente, el especialista selecciona una imagen del banco capturada en metafase y, mediante un software especializado, realiza de forma manual la ordenación y clasificación de los cromosomas. De esta manera, se construye el **cariograma** del individuo, que representa el conjunto de cromosomas organizados por pares y según su morfología [4], [5].

7. El cariograma se debe de realizar las veces necesarias para tener un diagnóstico certero.

En la Figura 1.2 muestra el seguimiento a la construcción del cariotipo.

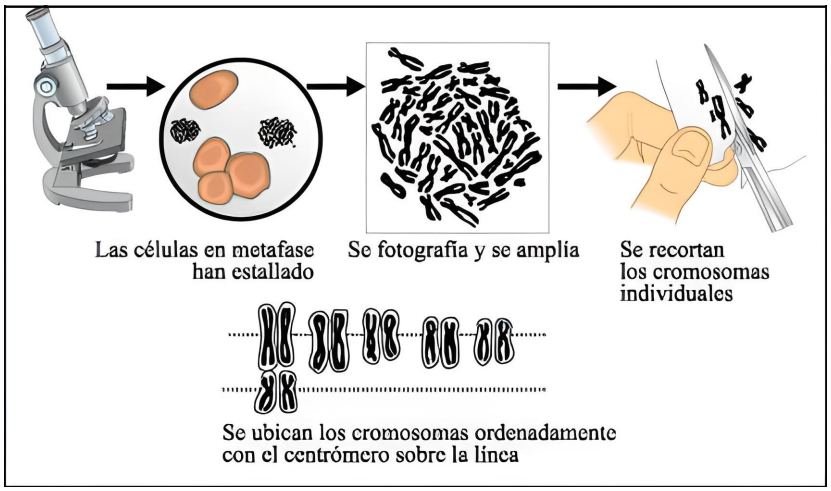


Figura 1.2: Obtención del cariograma.

En el proceso de análisis del cariotipo, la elaboración de cariogramas es una herramienta fundamental para identificar anomalías cromosómicas, como malformaciones, duplicaciones o ausencias, lo que permite diagnosticar trastornos genéticos. No obstante, esta tarea resulta compleja, tediosa y repetitiva. Un especialista puede invertir entre seis y ocho horas en el análisis completo de un solo individuo, lo que implica revisar entre 30 y 50 metafases; en casos de anomalías, este número puede ascender hasta 100 metafases para lograr un diagnóstico preciso [5]. Además, las condiciones de las imágenes suelen dificultar la visualización de los cromosomas, lo que incrementa aún más el tiempo requerido para el diagnóstico [6], [7], [8].

En la Figura 1.3 se muestra el proceso de obtención del cariotipo mediante un diagrama de bloques compuesto por ocho módulos. En el primer módulo se obtiene una muestra de tejido vivo, generalmente sangre o líquido amniótico. Los módulos 2 a 4 comprenden el procedimiento para obtener células en la etapa de metafase, donde los cromosomas pueden observarse con claridad. El módulo 5 consiste en la obtención de microfotografías de distintas células en metafase que muestren los cromosomas. Los módulos 6 y 7 se enfocan en extraer, ordenar y clasificar los cromosomas. Finalmente, en el último módulo se realiza el análisis del cariograma con el fin de establecer el diagnóstico

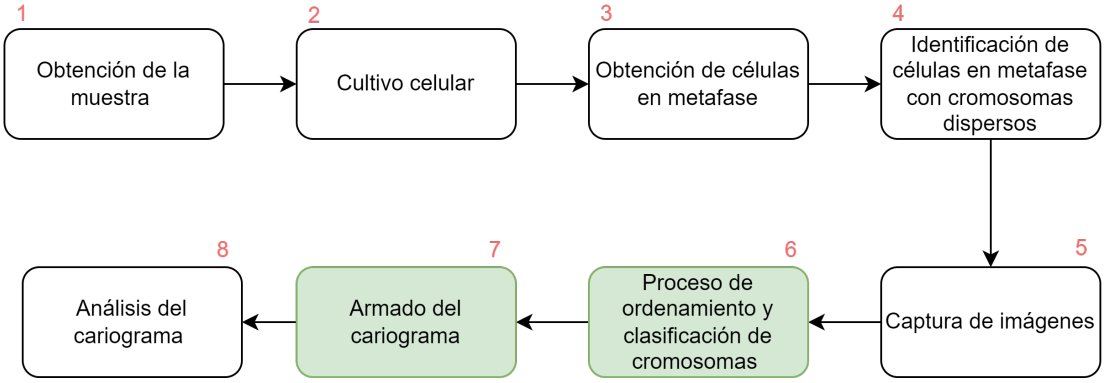


Figura 1.3: Proceso del cariotipo.

En esta tesis se plantea realizar los módulos 6 y 7, partiendo de una base de datos de imágenes de los cromosomas en etapa metafase para el armado del cariograma. Por lo tanto se propone desarrollar un sistema utilizando técnicas de procesamiento de imágenes e inteligencia artificial para la segmentación, extracción, identificación y ordenamiento de los cromosomas.

1.1. Antecedentes

En la rama de la genética y la biología molecular yacen los cromosomas, estructuras que contienen la información genética vital para la vida. Desde el descubrimiento de los cromosomas y su rol en la herencia, la identificación y caracterización de estos elementos se ha convertido en un objetivo en la investigación científica.

A lo largo del tiempo, el diagnóstico genético ha experimentado una evolución tecnológica, lo que ha permitido mejorar la resolución de las imágenes cromosómicas y facilitar su identificación. Para lograr una segmentación, separación y clasificación precisa de los cromosomas, se han desarrollado diversos métodos que han ido perfeccionándose con los avances en procesamiento de imágenes y técnicas computacionales. En este trabajo se presentan algunos de los métodos más relevantes utilizados históricamente, los cuales se describen a continuación. Además, se incluyen tablas comparativas —A.1, A.2, A.3 y

A.4— que pueden consultarse en el Apéndice A, donde se detallan las características y diferencias entre cada enfoque.

- Gady Adam [9] (1997) desarrolló un método para la separación de los cromosomas que están superpuestos y unidos en algún punto, este método se basa en la formulación de hipótesis y utiliza un algoritmo que identifica puntos cóncavos en los contornos de los cromosomas.

- Posteriormente Petros Karvelis [10] (2005) desarrolló un algoritmo para la separación de cromosomas adyacentes, utilizando la transformación Watershed. Este enfoque se basa en la segmentación de los cromosomas mediante teselados, logrando así una precisión del 92 % en la identificación de las estructuras cromosómicas.

- Wacharapong Srisang et al. [11] (2006) utilizó geometría computacional para la separación de los cromosomas traslapados, el método consistía en combinar los diagramas de Voroni con las triangulaciones Delaunary (DT) obteniendo un 85 % de precisión.

- Erinco Grisan et al. [12] (2007) utilizó un esquema de umbralización con variantes en la separación de los cromosomas traslapados y adyacentes, obteniendo una precisión del 90 % con cromosomas traslapados y un 92 % con cromosomas adyacentes.

- Unos años más tarde Petros Karvelis et al. [13] (2010) desarrollaron una mejora en su algoritmo hecho en 2005, el cual consistía en separar cromosomas traslapados y cromosomas adyacentes. Teniendo una precisión de 80.4 % en la separación de cromosomas traslapados y 90.6 % de cromosomas adyacentes.

- Mousami V. Munot et al. [14] (2011) desarrollaron un algoritmo modificado de tipo Snake que incorpora un enfoque Greedy, centrado en la separación de cromosomas adyacentes. Clasificaron las imágenes en dos grupos: el primero consistía en pares de cromosomas que se tocaban en un solo punto, mientras que el segundo incluía agrupaciones de tres y cuatro cromosomas unidos. Los resultados mostraron una precisión del 100 % en la separación de los pares de cromosomas, y del 95 % en el caso de los grupos de tres y cuatro cromosomas.

- Mukol A. Joshi et al. [15] (2012) trabajaron con geometría computacional tratando de separar a los cromosomas traslapados y probando con imágenes sintetizadas, separando en dos los resultados, por un lado, agrupa los cromosomas traslapados en 1 & 2 con una precisión del 100 % y por otro lado los cromosomas traslapados en 3 & 4 con un 88 % de precisión.

- Mousami V. Munot et al. [16] (2013) utiliza geometría computacional para trabajar con los cromosomas traslapados con tinción M-FISH, teniendo un 98 % de precisión.

- Devaraj Somasundaram [17] (2014) desarrolló un algoritmo de separación geométrica para la separación de cromosomas traslapados. Este algoritmo utiliza una hipótesis de separación para identificar puntos y líneas de corte en los contornos de los cromosomas. Además, la identificación de cromosomas homólogos[4] se realiza mediante la posición del centrómero, lo que permite una clasificación más precisa y eficiente de las imágenes cromosómicas.

A continuación, se describe los métodos para la clasificación de los cromosomas en el transcurso del tiempo:

- Boaz Lerner et al. [18] (1998) propuso un perceptrón multicapa con retropropagación para la clasificación de los cromosomas basándose en dos características, en el largo del cromosoma y la posición del centrómero.

- John M Conroy et al. [19] (2000) realiza una comparación de 4 clasificadores para determinar cual tiene mayor precisión descomposición de valores singulares (SVD), análisis de componentes principales (PCA), análisis discriminante de Fisher (FDA) y modelos ocultos de Markov (HMM). De todos los modelos mencionados anteriormente el que tuvo mejor clasificación fueron los modelos ocultos de Markov donde se obtuvo una precisión del 97 %.

- Zhenzen Kou. [20] (2002) utilizó el algoritmo de Máquina de Vectores de Soporte, reduciendo las salidas de las neuronas, de 24 salidas pasaron a 5 salidas, siendo de esta manera la reducción de dimensión, número de datos de entrenamiento y el tiempo de entrenamiento obteniendo un 90 % de precisión.

- Petros Karvelis et al. [21] (2006) se basaron en un clasificador de Bayes para la clasificación de imágenes de cromosomas. Su objetivo inició con la segmentación de las imágenes utilizando el método de Watershed morfológico aplicado a la intensidad del gradiente. Este proceso permitió descomponer la imagen en regiones homogéneas, lo que facilitó al clasificador lograr una mayor precisión. Al final, el clasificador alcanzó una precisión del 89 %.

- Xingwei Wang et al. [22] (2008) utiliza Árboles de decisión y redes neuronales artificiales (ANN) para la clasificación de los cromosomas, pero primero haciendo un proceso de preprocesamiento de las imágenes y obteniendo el 8 % de precisión.

- En el mismo año Petros Karvelis et al. [23] (2008) presentó un clasificador utilizando Bayes donde toma como entrada las regiones espectrales de las imágenes de los cromosomas teniendo un 82.4 % de precisión.

[4]Dicho de una parte de un ser vivo, especialmente de un órgano: Semejante a otra, por tener el mismo origen o estructura, aunque sus funciones puedan ser diferentes.

- Yaser Rahimi et al. [24] (2008) propuso una mejoría a la red neuronal de perceptrón multicapa con retroalimentación Feedforward basándose en las superficies de los cromosomas y la delimitación de los cromosomas teniendo un 73 % de precisión.

- Bonoit Legrand et al. [25] (2008) utilizó el algoritmo de Alineamiento Temporal Dinámico (DTW) utilizando las características de longitud y el perfil de densidad obteniendo el 82 % de precisión.

- Sunthorn Rungruangbaiyok and Pornchai Phukpattaranont. [26] (2010) utilizaron una red neuronal probabilística utilizando como características el área y perímetro de los cromosomas. Además de separar las imágenes en muestras femeninas y masculinas teniendo un 68.19 % y 61.3 % de precisión respectivamente.

- Enea Poletti et al. [27] (2012) desarrollaron un sistema de clasificación automática de cromosomas que se basa en redes neuronales. Utilizan un el eje medio de los cromosomas y la polarización para la extracción de características. Posteriormente, se emplean redes neuronales junto con un algoritmo de reasignación de clases, logrando una precisión promedio del 94 % en la clasificación de cromosomas.

1.2. Estado del arte

En los últimos años, la mayoría de los sistemas empleados para el armado del cariograma siguen utilizando métodos tradicionales de segmentación. Por ejemplo, los métodos tradicionales más utilizados de segmentación de cromosomas son *threshold-based*, *watershed-based*, *fuzzy-clustering-based*, *geometric feature-bases*, entre otros [3], [28].

Existen aplicaciones disponibles tanto gratuitas como de pago para el análisis cromosómico. Altinordu et al. [29] desarrollaron "KaryoType", un software especializado en el análisis citogenético en botánica, permitiendo mediciones y generación de gráficos como cariogramas e ideogramas, además de exportación a Excel. Kirov et al. [30] diseñaron "DRAWID", una herramienta en Java para estudios citológicos, citogenéticos y filogenéticos, con exportaciones en .jpeg, .png y Excel. Mahmoudi et al. [31] crearon un software "IdeoKar" para simplificar el análisis cromosómico, generando gráficos en SVG y PDF. Aunque son gratuitos, algunas páginas para su descarga no están disponibles.

Cabe mencionar que estas herramientas son semi-automáticas, pues se requiere tanto la intervención del usuario como la configuración de algunos parámetros, revisión de los resultados o el ajuste manual de algunos detalles del análisis. En contraste, el software comercial suele estar integrado en los equipos de laboratorio especializados en citogenética. Los más utilizados se encuentran ArgusSoft, HiBand, Ikaros Karyotyping Software, MetaClass y GenASis, entre otros. Estos programas ofrecen funcionalidades avanzadas y mayor grado de automatización, aunque su adquisición implica costos significativos.

Sin embargo, con la aparición de los modelos de Deep Learning (DL), se han propuesto y producido una nueva generación de modelos de segmentación de imágenes con mejoras

de rendimientos en la última década [32]. Las redes neuronales convolucionales (CNN) son una clase de modelos dentro del aprendizaje profundo (Machine Learning), donde cada kernel de convolución actúa como un extractor de características. Varios modelos clásicos de CNN han alcanzado un alto rendimiento en conjuntos de datos [33], [34], [28].

- En 2019, Xie et al. [6] desarrollaron un método para el análisis estadístico de cariogramas utilizando una combinación de Mask R-CNN para la segmentación y ResNet para la clasificación de cromosomas. Su enfoque incluyó la extracción de características mediante técnicas geométricas y de optimización. El método logró una exactitud entre el 82.00 % y el 95.70 %, dependiendo del tipo de cromosoma y las condiciones de la imagen.

- En 2020, Wang et al. [33] propusieron un enfoque para la clasificación de cromosomas basado en una ResNet extendida y vectores de características de etiquetas. Este método utilizó la distancia de Hausdorff para mejorar la exactitud en la clasificación de cromosomas individuales. Su método alcanzó una exactitud del 94.72 % en la clasificación de cromosomas en 24 clases.

- En 2022, Wei et al. [32] desarrolló un método basado en una red neuronal convolucional probabilística, denominado IAPP-CNN, que logró superar el rendimiento de algunas técnicas en la clasificación de cromosomas. El sistema alcanzó una exactitud del 99.20 %.

- En 2022, Liu et al. [35] propusieron un algoritmo de segmentación de imágenes de cromosomas que combina la umbralización basada en Otsu con el crecimiento de regiones. Compararon la efectividad de su método con otros enfoques de umbralización tradicionales. Los resultados mostraron que su algoritmo logró un margen de error promedio de 0.096, significativamente inferior al 0.171 del algoritmo de crecimiento de regiones (RGA) y al 0.193 del método Otsu convencional. Esta mejora la exactitud y resalta la eficacia del enfoque propuesto para la segmentación de imágenes cromosómicas.

- Ese mismo año, You et al. crearon una gran base de datos de cromosomas manualmente etiquetada y densamente anotada para la segmentación de instancias de cromosomas. Utilizaron diferentes redes neuronales para probar su base de datos, obteniendo una exactitud del 83.3 %, 82.7 %, 90.3 %, 93.0 %, 92.4 %, 92.5 % y 89.7 % dependiendo de las redes utilizadas.

- En 2023, Yang et al. [8] propusieron una DCNN de 11 capas para la clasificación de cromosomas con defectos estructurales. Su método alcanzó una exactitud del 91.75 % en la clasificación de 24 tipos de cromosomas y del 87.76 % en la clasificación de 32 tipos.

- También en 2023, Devaraj et al. [28] implementaron un sistema basado en geometría computacional para la separación de cromosomas adyacentes y traslapados. El

método logró una exactitud del 99.68 %, utilizando algoritmos geométricos como la triangulación de Delaunay y la interpolación por splines.

- En 2024, Chang et al. [36] propusieron un enfoque progresivo de segmentación de cromosomas utilizando una combinación de redes neuronales (CNI-Net) y procesamiento de imágenes tradicional. Su método alcanzó una exactitud del 94.60 % en la identificación de agrupaciones cromosómicas y del 99.15 % en la segmentación.
- Ese mismo año, Chen et al. [3] desarrollaron KaryoXpert, un marco preciso para la segmentación y clasificación de cromosomas sin la necesidad de etiquetas manuales. Utilizaron una combinación de ResNet50 y Yolov7 en diferentes etapas, alcanzando una exactitud del 92.3 % y optimizando el uso de memoria de la GPU.
- Finalmente, en 2024, Zhang et al. presentaron un método de clasificación y enderezamiento de cromosomas basado en una HRNet y una red neuronal convolucional para la detección, orientación y polaridad de cromosomas doblados. Su enfoque alcanzó una exactitud del 98.1 % en la clasificación y del 99.8 % en la polaridad de los cromosomas.

1.3. Problemática

Dado que la elaboración del cariograma es un proceso complejo, tedioso, repetitivo y propenso a errores humanos, ha sido un tema de interés para su automatización. Por ello, el presente trabajo se enfoca en desarrollar un sistema de clasificación de cromosomas de manera confiable con el fin de reducir el tiempo en el análisis por parte del especialista y mejorar la precisión diagnóstica.

1.4. Objetivos

1.4.1. Objetivo general

Desarrollar un sistema utilizando técnicas de segmentación e inteligencia artificial para el ordenamiento de cromosomas.

1.4.2. Objetivos específicos

1. Analizar técnicas de visión por computadora y CNN para la segmentación de los cromosomas.
2. Determinar técnicas de visión por computadora para la segmentación y discriminación de los cromosomas.
3. Desarrollar un algoritmo utilizando CNN para la identificación de los cromosomas.

4. Implementar el algoritmo en una tarjeta de desarrollo de propósito específico.

5. Evaluar el algoritmo en base a los datos de libre acceso.

1.5. Justificación

La elaboración del cariograma es un proceso importante para el diagnóstico de enfermedades genéticas, pero su complejidad y el tiempo que demanda lo convierten en una tarea difícil. Ante esta problemática, resulta proponer soluciones que optimicen el trabajo de los especialistas.

Dentro de este trabajo se presentan dos aportaciones principales: una al conocimiento humano y otra al impacto social. En cuanto al aporte al conocimiento humano, el desarrollo de un sistema automatizado de segmentación e identificación de cromosomas, basado en técnicas de visión por computadora y redes neuronales convolucionales (CNN), permite reducir y optimizar el tiempo requerido en la construcción del cariograma, facilitando la emisión de diagnósticos en menor tiempo. Además, la implementación en una tarjeta de desarrollo de propósito específico (Jetson Nano) demuestra la viabilidad del sistema en dispositivos con recursos limitados, ampliando su aplicabilidad. Respecto al impacto social, la implementación de esta herramienta en un entorno educativo contribuye al aprendizaje de los estudiantes, mientras que su aplicación en el ámbito clínico favorece la reducción del tiempo de diagnóstico y disminuye la dependencia de instancias externas. Es importante destacar que este sistema se concibe como una herramienta complementaria y de apoyo para los especialistas, sin pretender sustituir su experiencia ni su juicio clínico.

1.6. Metodología

En la presente tesis se propone la siguiente metodología (Figura 1.4), la cual presenta de manera general la estructura para alcanzar los objetivos planteados. El proceso inicia con la disponibilidad de un banco de imágenes con las siguientes características: imágenes de los cromosomas desordenados (metafase) y cromosomas previamente clasificados o en su defecto el cariograma de cada metafase. A partir del banco de imágenes, se selecciona una imagen de cromosomas en metafase, la cual se somete a un proceso de segmentación. Este proceso inicia con la Etapa I, donde se comparan dos técnicas distintas de binarización. Posteriormente, se lleva a cabo la discriminación entre cromosomas individuales y agrupados, considerando únicamente los cromosomas individuales para la etapa posterior del análisis. La Etapa II corresponde a la clasificación de los cromosomas. En esta fase se realizan diversos procesos, entre los cuales se plantea el uso de una red neuronal convolucional (CNN) como modelo principal para la clasificación. Este modelo se entrena y valida utilizando una base de datos de cromosomas individuales previamente clasificados. Para ello, se lleva a cabo un preprocesamiento de los datos de entrada, que

incluye la normalización y el redimensionamiento de las imágenes, con el fin de adaptarlas a la arquitectura de la CNN tanto los datos para entrenamiento y validación como los datos de la etapa I. Una vez completado el entrenamiento y la validación, se emplean estos cromosomas para llevar a cabo el ordenamiento y la clasificación cromosómica.

Una vez completadas las etapas anteriores, se procede a la implementación del sistema en una tarjeta de desarrollo de propósito específico, con el objetivo de optimizar el código y permitir su ejecución en dispositivos con recursos computacionales limitados. Esta implementación busca validar la eficiencia del sistema en entornos reales.

Finalmente, se realiza una evaluación del desempeño del sistema, con el propósito de medir la eficacia del algoritmo propuesto y analizar su viabilidad para tareas de clasificación y ordenamiento cromosómico. Esta evaluación permitirá identificar fortalezas, limitaciones y posibles áreas de mejora para futuras versiones del sistema.

1.7. Organización de la tesis

En este trabajo de tesis esta organizada en cinco capítulos, a continuación se describe cada capitulo.

- Capítulo 1: *Introducción*. En este apartado se presenta el planteamiento del problema, el objetivo general, los objetivos específicos, la justificación del trabajo y una descripción de la metodología propuesta.

- Capítulo 2: *Marco teórico*. Se hace una descripción de los conceptos de los cromosomas, el cariograma y el cariotipo. Además, se abordan los fundamentos de procesamiento digital de imágenes y el aprendizaje profundo (Deep Learning).

- Capítulo 3: *Metodología*. Se describe la base de datos utilizada, el preprocesamiento de las imágenes, el algoritmo de la discriminación de los cromosomas y la arquitectura empleada para la clasificación de los mismos.

- Capítulo 4: *Resultados*. En este capítulo se presentan los resultados obtenidos durante las etapas de preprocesamiento, procesamiento, discriminación de los cromosomas, la clasificación de los cromosomas y la implementación del algoritmo en la tarjeta de desarrollo de propósito específico.

- Capítulo 5: *Conclusiones*. Se describe las conclusiones de los resultados obtenidos en las etapas de la discriminación de los cromosomas, clasificación y la implementación del algoritmo en la tarjeta de desarrollo de propósito específico.

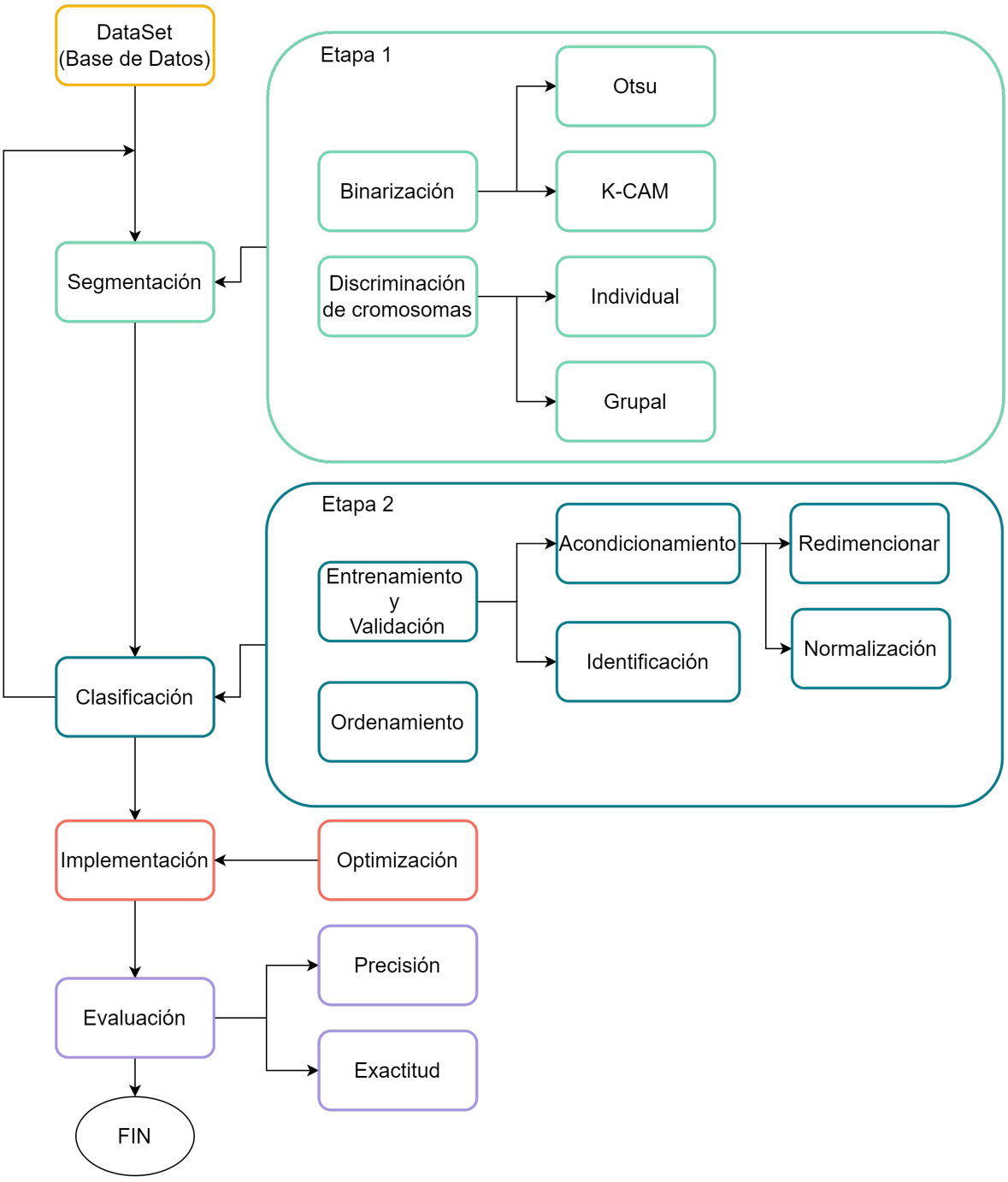


Figura 1.4: Diagrama de bloques.

Capítulo 2

Marco teórico

2.1. Fundamentos

2.1.1. Cariotipo

Antes de continuar, es fundamental aclarar algunos conceptos, ya que los términos **cariotipo**, **cariograma** e **ideograma** suelen confundirse, aunque tienen significados distintos. Un cariograma es una representación ordenada de los cromosomas, que puede ser realizada mediante dibujo, fotografía o imagen digital, y muestra los cromosomas de una célula. Por otro lado, el cariotipo se refiere al estudio del conjunto de cariogramas de un individuo, ya sea normal o anormal. Finalmente, un ideograma es una representación gráfica de cada cromosoma para el armado del cariograma [37].

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. Los cromosomas están divididos en dos secciones por el centrómero, que es el punto de unión entre los dos brazos del cromosoma; el brazo corto "p"(de la palabra francesa petite) y un brazo largo "q"(de queue, que significa cola). Los extremos de los cromosomas se conocen como telómeros, estructuras que protegen la integridad del material genético. Cada brazo se subdivide en regiones y bandas, las cuales pueden distinguirse mediante técnicas de tinción cromosómica, como se ilustra en la Figura 2.1 [38].

El ser humano cuenta con 46 cromosomas, los cuales se agrupan en pares y su vez se clasifican en 7 grupos alfabéticos, desde la letra A a la G. Además, pueden ser ordenados según su tamaño, de mayor a menor. Para facilitar el análisis de los cromosomas, es importante tener conocimiento de su morfología, por lo que esta clasificación, se basa en la posición del centrómero. A continuación se describen parte de la morfología de los cromosomas:

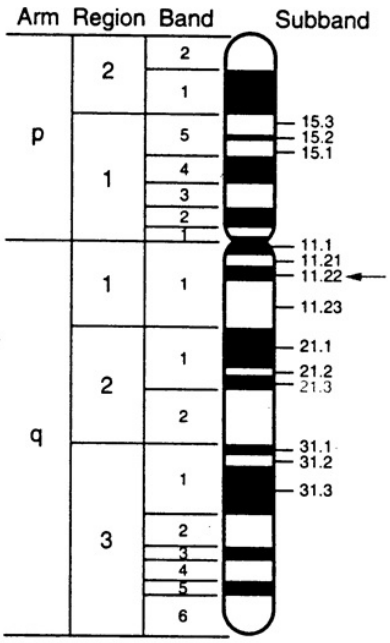


Figura 2.1: Estructura de un cromosoma [37].

- Metacéntricos: donde brazos (p) y (q) son del mismo tamaño.
- Submetacéntricos: brazos (p) son más pequeños que los brazos (q).
- Acrocéntricos: brazos (q) son largos respecto brazos (p), el centrómero se encuentra mas cercano al brazo (p).

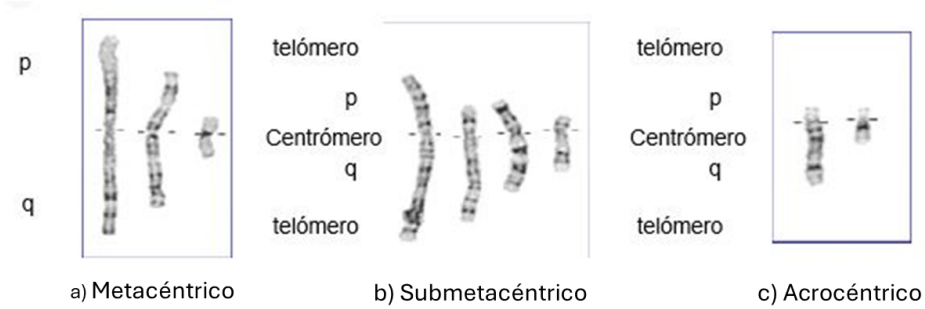


Figura 2.2: Clasificación morfológica de los cromosomas humanos.

En la Figura 2.2 muestra la clasificación morfológica de los cromosomas humanos. A la izquierda se encuentran los cromosomas metacéntricos, en medio los submetacéntricos y del lado derecho los acrocéntricos.

En la Tabla 2.1 se describen los grupos de los cromosomas y una pequeña descripción de los mismos [39] [38].

Grupo	Concepto
Grupo A $(1-3)$	Los cromosomas metacéntricos grandes se distinguen fácilmente entre sí por su tamaño y la posición del centrómero. Específicamente par 1 y 3 metacéntricos; y par 2 submetacéntricos.
Grupo B $(4-6)$	Cromosomas submetacéntricos son grandes.
Grupo C $(6-12, X)$	Cromosomas de tamaño medio metacéntrico o submetacéntrico. El cromosoma X se parece a los cromosomas más largos de este grupo.
Grupo D $(13-15)$	Cromosomas acrocéntricos de tamaño medio.
Grupo E $(16-18)$	Son de tamaño pequeño, el par 16 es metacéntrico y los pares 17 y 18 son submetacéntrico.
Grupo F $(19-20)$	Cromosomas metacéntricos cortos y pequeños.
Grupo G $(21-22, Y)$	Cromosomas de menor tamaño y acrocéntricos.

Tabla 2.1: Clasificación de cromosomas.

Los laboratorios de citogenética evalúan la resolución cromosómica mediante la identificación y puntuación de puntos de referencia visibles en las imágenes de los cromosomas. Para determinar la resolución a nivel de bandas, se deben cumplir al menos tres criterios de referencia, conforme a lo establecido en el estándar ISCN 2020 [37]. En la Figura 2.3, se presenta el cromosoma 1 como ejemplo, mostrando distintas resoluciones. De izquierda a derecha, se observa un incremento progresivo en la resolución, donde una mayor resolución permite visualizar un mayor número de bandas cromosómicas, mientras que una menor resolución muestra menos detalles, dificultando la identificación precisa de las estructuras internas del cromosoma [40].

Técnicas de tinción y bandeo

En los cromosomas en metafase, las técnicas de bandeo producen una serie de puntos de referencia que permiten examinar cromosomas individuales e identificar segmentos específicos de cada cromosoma. Estos puntos de referencia permiten evaluar cromosomas normales e identificar puntos de ruptura en caso de que exista una anomalía. Por lo cual se han desarrollado diferentes técnicas de bandeo con propiedades y aplicaciones específicas. Estos patrones de bandeo suelen ser un código de barras que los citogenetistas pueden reconocer para identificar anomalías numéricas y estructurales, como translocaciones, inversiones, deleciones, duplicaciones, sitios frágiles y otras características más complejas, además de precisar los puntos de ruptura [40].

Las técnicas de bandeo se dividen en dos grupos:

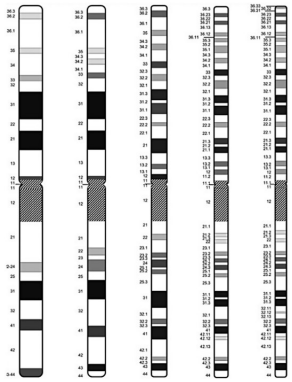


Figura 2.3: Ideograma del patrón de bandeo G del cromosoma 1, mostrando los diferentes tipos de resolución de izquierda a derecha en aproximadamente 300, 400, 550, 700 y 850 bandas [40].

- Bandas distribuidas a lo largo de todo el cromosoma:
  - Bandeo con Giemsa (bandeo G, GTG)
  - Bandeo con quinacrina (bandeo Q, QFQ)
  - Bandeo inverso (bandeo R, RHG)
- Bandas que tiñen estructuras específicas del cromosoma
  - Tinción de heterocromatina centromérica (bandeo C, CBG)
  - Tinción de la región organizadora del nucléolo (tinción NOR)
  - Bandeo T
  - FISH (hibridación in situ con fluorescencia)

Cariotipo

El cariotipo es un proceso utilizado para el análisis estructural y numérico de los cromosomas, con el objetivo de detectar posibles anomalías cromosómicas, tales como deleciones [1], duplicaciones [2], translocaciones [3] o alteraciones en el número de cromosomas. Esta prueba ha sido aplicada en una amplia variedad de organismos vivos como: animales, plantas, seres humanos, entre otras especies.

Para la detección de anomalías cromosómicas, se debe verificar que la muestra contenga el número correcto de cromosomas, es decir, 46 cromosomas en individuos sin alteraciones genéticas. Además, se debe observar si existen posibles irregularidades en el tamaño, forma o estructura de los cromosomas.

[1] Es la pérdida de un segmento del ADN dentro de un cromosoma.
[2] Ocurre cuando una región del cromosoma se copia una o más veces.
[3] Intercambio de segmentos entre dos cromosomas no homólogos.

A continuación, se describen las etapas principales del proceso:

1. Se toma la muestra de algún tejido vivo (sangre, médula ósea) y en caso de mujeres embarazadas 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, un agente que detiene la división celular en la fase metafase. Esta etapa los cromosomas se encuentran más condensados y visibles, donde se les aplica una tinción específica que permite resaltar sus estructuras

3. Mediante un gotero, se toman pequeñas muestras que se dejan caer desde una altura controlada sobre un portaobjetos. Este impacto rompe las células que contienen los cromosomas, lo que provoca que, al estallar, los cromosomas queden expuestos. Estas células rotas se conocen comúnmente como "botones".

4. Se realiza un mapeo al portaobjetos y se toman microfotografías (fotografías a escala de micras), con el fin de identificar las áreas donde se encuentran las células en metafase.

5. Una vez completado el mapeo, se seleccionan aquellas imágenes que presentan los cromosomas con mayor claridad.

6. Se selecciona una imagen previamente capturada de una célula en metafase, en la cual los cromosomas son visibles. A partir de esta imagen, los cromosomas se extraen individualmente mediante técnicas de recorte digital. Posteriormente, se procede a agruparlos en pares, en forma de rompecabezas, en el que cada cromosoma es comparado con un ideograma de referencia. Este ideograma sirve como guía para identificar y ordenar los cromosomas según su tamaño (de mayor a menor), asignándolos del 1 al 22, correspondiente a los autosomas, y un par adicional (par 23) que representa los cromosomas sexuales.

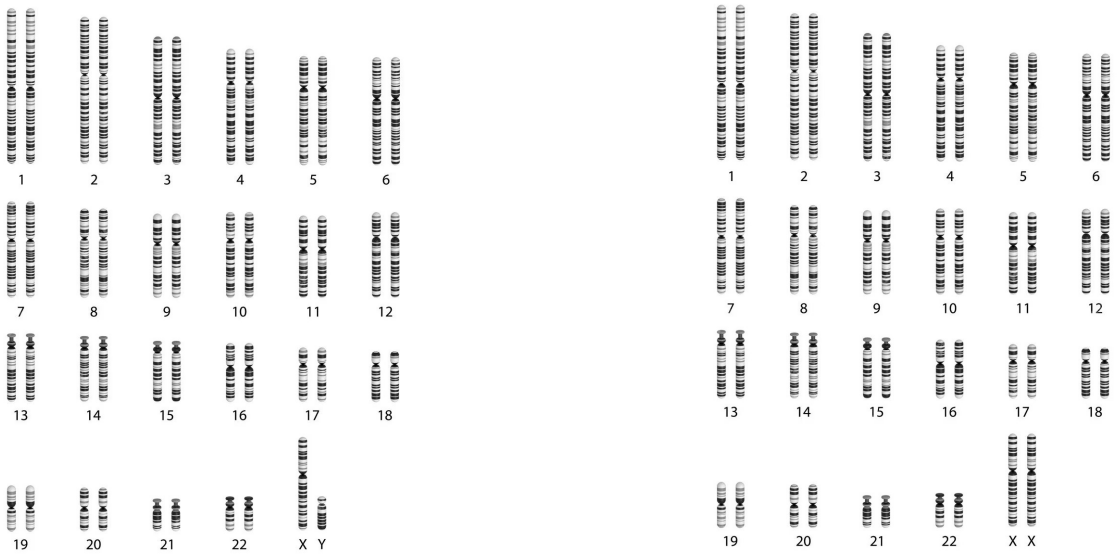
7. Una vez que los cromosomas han sido clasificados y ordenados, se obtiene el cariograma, una representación gráfica que organiza los cromosomas en pares homólogos de acuerdo con su tamaño, forma y patrón de bandas. Este cariograma es evaluado por un especialista en citogenética, quien examina cada par cromosómico con el propósito de identificar posibles anomalías estructurales o numéricas.

8. Para garantizar un diagnóstico confiable, es necesario analizar entre 30 y 50 células en metafase de un mismo individuo. No obstante, cuando se detecta alguna alteración cromosómica, el estudio debe ampliarse hasta 100 células, con el fin de confirmar la anomalía y asegurar la precisión del resultado.

En la Figura 2.4 se presenta un ideograma[4] que sirve como referencia para el ordenamiento y clasificación de los cromosomas en hombres y mujeres. En este ideograma,

[4]Un ideograma es una representación esquemática que ilustra el tamaño, la forma y el patrón de bandas de todo el complemento cromosómico.

muestra la ubicación del centrómero, así como las diferencias en los tamaños de los brazos y las bandas cromosómicas. La diferencia entre el cariograma masculino y femenino se realiza a través del par de cromosomas sexuales: los hombres poseen un par compuesto por un cromosoma X y uno Y $(X,\ Y)$, mientras que las mujeres tienen un par formado por dos cromosomas X $(X,\ X)$.



(a) Cariograma de un hombre.

(b) Cariograma de una mujer.

Figura 2.4: Ejemplo del armado del cariograma.

2.2. Segmentación de los cromosomas

La extracción, clasificación y el ordenamiento de cromosomas son tareas repetitivas, tediosas y requieren mucho tiempo. Esta situación ha impulsado el desarrollo de soluciones orientadas a la automatización. Un Sistema de Cariotipo Automático (Automatic Karyotyping System, AKS) debe integrar varias etapas como: a) adquisición de la imagen de una célula, b) segmentación de la imagen para localizar los diferentes cromosomas, c) caracterización de cada cromosoma, y d) ordenamiento y clasificación de los cromosomas. Cada una de estas etapas presenta diversos desafíos técnicos que deben ser superados para lograr una automatización eficiente y precisa del proceso [39]. En la etapa de segmentación (Figura 2.5, inciso b)), es crucial, ya que permite identificar y aislar cada cromosoma. Este proceso (inciso c)) requiere un análisis detallado para separar los objetos candidatos a cromosomas del fondo, seguido de una evaluación para distinguir entre cromosomas individuales y aquellos que se encuentran agrupados. Una vez identificados, los cromosomas agrupados son separados mediante técnicas de procesamiento de imagen, quedando listos para la extracción de características, así como su ordenamiento y clasificación, como se muestra en el inciso d).

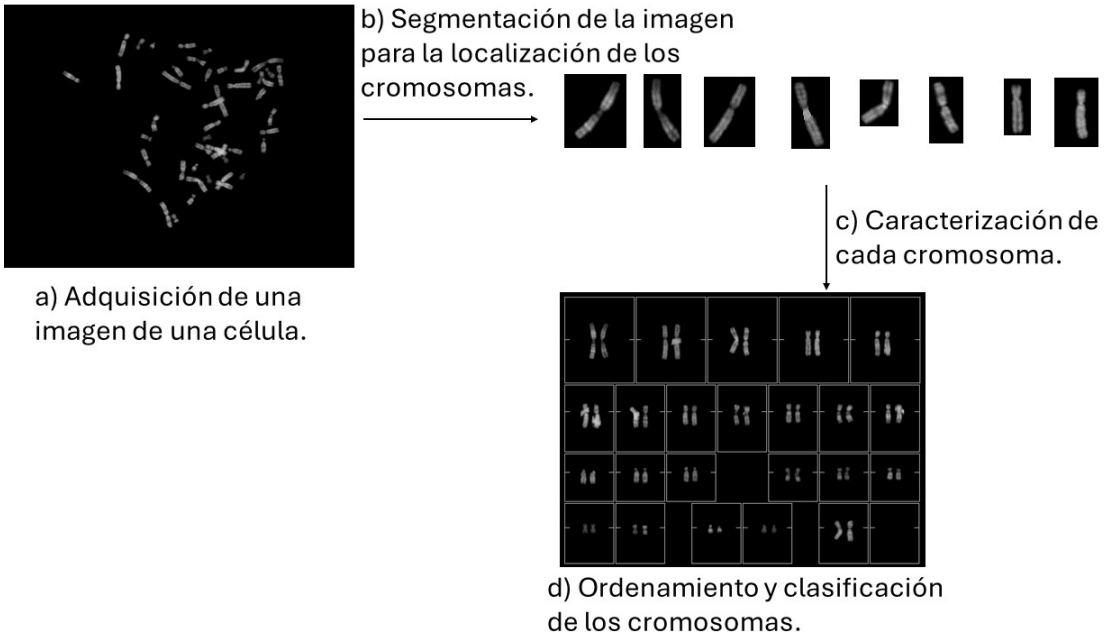


Figura 2.5: Etapas del AKS.

A continuación, se describen los métodos propuestos para el desarrollo de las tres etapas principales del sistema AKS de la Figura 2.5: b) preprocesamiento y segmentación, c)extracción de características y d) clasificación.

Preprocesamiento y segmentación

La segmentación es un paso importante en los sistemas automáticos de análisis de cromosomas (AKS). En las imágenes capturadas durante la metafase, es común encontrar cromosomas individuales, objetos no deseados, así como cromosomas parcialmente unidos o traslapados. Estas últimas situaciones complican el análisis, representando un desafío para los especialistas al momento de ordenar y clasificar los cromosomas correctamente.

En la mayoría de los casos, las imágenes se capturan en color; sin embargo, para realizar la segmentación, es necesario un preprocesamiento en el que la imagen se convierte a escala de grises, se ajusta el tamaño de la imagen; posteriormente, este paso permite llevar a cabo una limpieza, detección y segmentación de los cromosomas. La limpieza se refiere a la reducción del ruido, donde se considera ruido cualquier objeto presente en la imagen que no sea un cromosoma, como el núcleo celular, restos de suciedad, entre otros. La detección se encarga de identificar los cromosomas individuales o grupos de cromosomas que pueden estar parcialmente unidos o traslapados.

Extracción de objetos

El primer paso en el análisis de las imágenes es la separación de los cromosomas del fondo. Para lograrlo, es común convertir la imagen original I en una imagen binaria, donde los objetos claros (cromosomas) aparecen sobre un fondo oscuro. En este caso, los píxeles del objeto y del fondo tienen intensidades agrupadas en dos tonos dominantes.

La segmentación binaria se basa en la elección de un umbral de gris T, que permita distinguir entre los píxeles pertenecientes a los cromosomas y aquellos que corresponden al fondo. La imagen I se representa como una función $f(x, y)$, donde (x, y) indica la posición de cada píxel, y $f(x, y)$ su nivel de intensidad. De este modo, un píxel (x, y) con $f(x, y) > T$ será clasificado como parte del objeto (p_1), de lo contrario, se considerará parte del fondo (p_0).

La binarización de cada píxel en la imagen, se realiza de la siguiente forma:

$$I_b(x, y) = \begin{cases} p_0 & \text{si } f(x, y) < T \\ p_1 & \text{si } f(x, y) \geq T \end{cases} \tag{2.1}$$

Donde:

- $I_b(x, y)$ es la imagen binarizada resultante
- p_0 es el pixel que se le asigna el valor 0 por debajo del umbral T
- p_1 es el pixel que se le asigna el valor 1 por arriba del umbral T

Uno de los métodos más utilizado para determinar un umbral global óptimo es la método de Otsu [41], que se basa en buscar un valor de umbral (T) global que maximiza la varianza entre clases (fondo y objeto), lo cual indica la mejor separación posible entre ambos grupos.

En imágenes de 8 bits (niveles de gris de 0 a 255), el número total de niveles de intensidad es $L = 2^8 = 256$, y un umbral dado T, se calculan las siguientes probabilidades:

$$p_0(T) = \sum_{i=0}^{T} p_i, \quad p_1(T) = \sum_{i=T+1}^{L-1} p_i$$

Donde

- $p_0(T)$: es la probabilidad acumulada del fondo (niveles de 0 a T)
- $p_1(T)$: la probabilidad acumulada del objeto (niveles de $T+1$ a $L-1$)
- p_i: representa la probabilidad del nivel de gris i, obtenida del histograma de la imagen

Las intensidades promedio del fondo (m_0), el objeto (m_1) y la imagen completa (m) se calculan mediante:

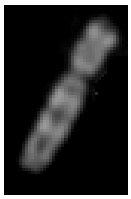
$$m_0 = \sum_{i=0}^{T} ip_i, \quad m_1 = \sum_{i=T+1}^{L-1} ip_i, \quad m = \sum_{i=0}^{L-1} ip_i$$

La varianza entre los grupos, el cual representa la separación entre el fondo y el objetos, se define como:

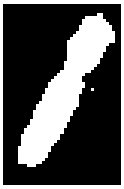
$$v_{\text{intermedio}}(T) = p_0(T) \cdot p_1(T) \cdot (m_0 - m_1)^2 \tag{2.2}$$

Este proceso se repite para diversos valores de umbral (T), seleccionando aquel que maximiza $v_{intermedio}$, lo cual indica una mejor separación entre el fondo y el objeto. Los píxeles con intensidades inferiores al umbral se asignan un valor de cero, mientras que aquellos con intensidades superiores reciben un valor de uno [42].

Un ejemplo de binarización se muestra en la Figura 2.6, donde (a) presenta la imagen de un cromosoma antes de aplicar la umbralización, y (b) muestra la imagen tras ser binarizada.



(a) Imagen en escala de grises



(b) Imagen binarizada

Figura 2.6: Imagen de un cromosoma en escala de grises antes y después de aplicar umbralización.

El algoritmo *K-means* es uno de los métodos más utilizados para el agrupamiento no supervisado. Su objetivo es particionar un conjunto de datos en K clústeres minimizando la función de costo:

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2, \tag{2.3}$$

donde μ_k representa el centroide del clúster C_k. Este procedimiento se basa en un proceso iterativo que alterna entre la asignación de cada punto al clúster más cercano y la actualización de los centroides. La calidad del resultado depende de tres factores principales: la inicialización de los centroides, el número de clústeres K y la métrica de distancia, siendo la euclidiana la más común.

Para abordar estas limitaciones, se emplean modelos probabilísticos como los *Campos Aleatorios de Markov* (CAM), utilizados en visión por computadora. Un CAM se define sobre un grafo no dirigido donde cada nodo representa una variable aleatoria y las aristas codifican relaciones de vecindad. Cumple la propiedad de Markovianidad:

$$P(f_i | f_{S-\{i\}}) = P(f_i | f_{\mathcal{N}_i}), \tag{2.4}$$

lo que implica que la probabilidad de un nodo depende únicamente de sus vecinos directos. Esta característica permite modelar dependencias espaciales y reducir la incertidumbre en la asignación de etiquetas. Según el teorema de Hammersley-Clifford, todo CAM es equivalente a un *Campo Aleatorio de Gibbs*, cuya distribución se expresa como:

$$P(f) = \frac{1}{Z} \exp\left[-\frac{U(f)}{T}\right], \tag{2.5}$$

donde $U(f)$ es la función de energía compuesta por potenciales de cliques y Z es la función de partición que normaliza la distribución.

La combinación de ambos enfoques ha dado lugar a algoritmos híbridos como *K-CAM*, que integra la agrupación inicial mediante *K-means* con la regularización espacial de los CAM. Este método clasifica píxeles en regiones homogéneas (binarización) y ajusta las fronteras considerando la vecindad, lo que mejora la segmentación en imágenes cromosómicas con ruido o estructuras parcialmente traslapadas [43].

Después de extraer los cromosomas del fondo mediante la umbralización, se aplican técnicas adicionales para identificar y reducir objetos no deseados, como manchas o restos. Estas técnicas pueden ser morfológicas, geométricas (basadas en tamaño y forma) o de nivel de gris. De esta manera, se identifican los cromosomas, sin importar si están aislados o agrupados.

Discriminación de cromosomas

La etapa de discriminación suele combinarse con la de separación entre cromosomas unidos y agrupados. Es importante seleccionar el método que se utilizará para identificar

los objetos que forman parte de los cromosomas y asignarlos al proceso de separación correspondiente.

Calzada [39] considera lo siguiente para la discriminación:

- Esqueleto: Reduce el objeto a una línea delgada en el centro.
- Ramas y puntos de cruce del esqueleto: Son las partes donde el esqueleto se divide o donde varias ramas se conectan.
- Distancia Euclidianas de los píxeles del esqueleto: Esta medida calcula qué tan lejos están los puntos del esqueleto (como las ramas y cruces) de otras áreas de la imagen, lo que ayuda a saber si los cromosomas están correctamente separados.
- Polígono convexo: Se utiliza el contorno del cromosoma para generar su envolvente convexa, que es el polígono más pequeño sin concavidades que lo contiene completamente. Esta envolvente permite identificar los espacios entre el contorno real del cromosoma y su forma convexa ideal.
- Elipse: Permite obtener la excentricidad, que indica el grado de elongación del objeto. Valores cercanos a 1 sugieren cromosomas individuales, mientras que valores cercanos a 0 pueden indicar agrupamientos.

Características morfológicas

Entre las medidas morfológicas más empleadas se encuentran la longitud, el área y el perímetro convexo:

- Longitud: Se determina mediante el Medial Axis Transform (MAT) [44], que corresponde a la línea central del cromosoma atravesando el centrómero (Figura 2.7 (a)). La longitud del cromosoma se define como la longitud de esta línea [45, 46]. También puede calcularse mediante operaciones morfológicas [39]. Chen et al. [47] proponen una mejora al algoritmo de adelgazamiento para obtener esqueletos de un solo píxel de ancho. La longitud relativa se expresa como la razón entre la longitud del cromosoma y la suma de las longitudes de todos los cromosomas del conjunto. Cho et al. [48] sostienen que esta característica resulta más representativa que la longitud absoluta.
- Área: Se define como el número de píxeles que conforman el cromosoma (Figura 2.7 (b)). El área relativa se calcula como la razón entre el área del cromosoma y la suma de las áreas de todos los cromosomas del conjunto.
- Perímetro convexo: Corresponde al perímetro del menor polígono convexo que puede contener al cromosoma (Figura 2.7 (c)), lo que permite una medición simplificada y estandarizada.

Finalmente, el centrómero constituye la región donde las cromátidas inician su separación durante la división celular (Figura 2.7 (d)). Habitualmente se identifica mediante el perfil de intensidad del medial axis [49]. El índice del centrómero se calcula como la razón entre la longitud del brazo corto (p) y la longitud del brazo largo (q).

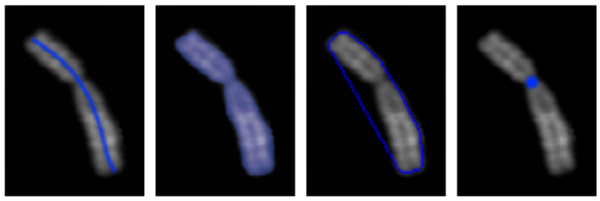


Figura 2.7: Características de un cromosoma: (a) línea central que pasa por el cromosoma, (b) área total del cromosomas, (c) perímetro del menor polígono convexo, (d) centrómero que divide los brazos q y q del cromosoma [50].

2.3. Ordenamiento y clasificación de cromosomas

El ordenamiento y clasificación de cromosomas en este trabajo se realizará mediante técnicas de inteligencia artificial (IA), específicamente utilizando redes neuronales convolucionales (CNN, por sus siglas en inglés). Las CNN han demostrado ser altamente en el procesamiento de imágenes, debido a su capacidad para aprender y extraer características relevantes de manera automática, lo que las convierte en una herramienta ideal para el análisis citogenético.

La IA comprende un conjunto de métodos orientados a desarrollar sistemas capaces de realizar tareas que tradicionalmente requieren inteligencia humana. Dentro de este campo se encuentra el aprendizaje automático (machine learning), que se centra en el diseño de algoritmos capaces de aprender patrones a partir de datos sin necesidad de programación explícita para cada tarea. A su vez, el aprendizaje profundo (deep learning) constituye una subcategoría del machine learning, caracterizada por el uso de arquitecturas con múltiples capas que permiten extraer representaciones jerárquicas y complejas de los datos. Las CNN forman parte de este enfoque y son especialmente eficaces en el análisis de imágenes, ya que pueden identificar patrones espaciales y estructuras visuales con gran precisión. En las siguientes subsecciones se presentan los conceptos fundamentales de inteligencia artificial, machine learning, deep learning y redes neuronales.

Inteligencia Artificial

Desde una perspectiva técnica, la Inteligencia Artificial (IA) se define como la capacidad de una máquina para procesar información de su entorno y utilizarla en la toma de

decisiones con el fin de alcanzar resultados deseados. Este enfoque permite a las máquinas aprender de los datos y aplicar ese conocimiento en la resolución de problemas, imitando ciertos aspectos del comportamiento humano. Dentro de la IA se encuentra el machine learning, que a su vez incluye el deep learning [51].

Machine Learning

El machine learning (aprendizaje automático) es un campo que se centra en la extracción de conocimiento a partir de datos, situándose en la intersección de la estadística, la inteligencia artificial y la informática. Su principio fundamental es que los sistemas pueden aprender de los datos y realizar predicciones o tomar decisiones sin ser programados explícitamente para cada tarea. Existen dos enfoques principales:

- Aprendizaje Supervisado: El algoritmo recibe datos de entrada junto con las salidas deseadas (etiquetas) y aprende a predecir resultados para nuevos datos.
- Aprendizaje No Supervisado: El algoritmo trabaja con datos sin etiquetas, buscando descubrir patrones o estructuras ocultas [52, 53].

Deep Learning

El deep learning es una subcategoría del machine learning que utiliza arquitecturas profundas compuestas por múltiples capas. Cada capa aprende representaciones progresivamente más complejas de los datos, lo que permite resolver problemas de alta dimensionalidad, como el análisis de imágenes. Estas arquitecturas se implementan mediante redes neuronales artificiales, que constituyen la base del aprendizaje profundo [54].

Redes Neuronales Artificiales

Las CNN son un tipo especializado de red neuronal diseñada para procesar datos con estructura de cuadrícula, como imágenes. Su arquitectura incluye capas convolucionales que aplican filtros para detectar características locales (bordes, texturas, formas), seguidas de capas de agrupamiento y capas totalmente conectadas para la clasificación final. En este trabajo, las CNN se emplearán para clasificar cromosomas a partir de imágenes cromosomas en estado metafase, aprovechando su capacidad para aprender características relevantes sin necesidad de diseñar manualmente los descriptores.

Arquitectura de la red neuronal

La arquitectura de una red neuronal (Figura 2.8) se describe de la siguiente manera:

- Capa de entrada: Almacena los datos iniciales.

- Capas ocultas: La salida de cada capa de entrada se convierte en la entrada para todas las capas ocultas, que son responsables de procesar la información y realizar las operaciones matemáticas necesarias.
- Capa de salida: Esta capa es la encargada de realizar las predicciones.

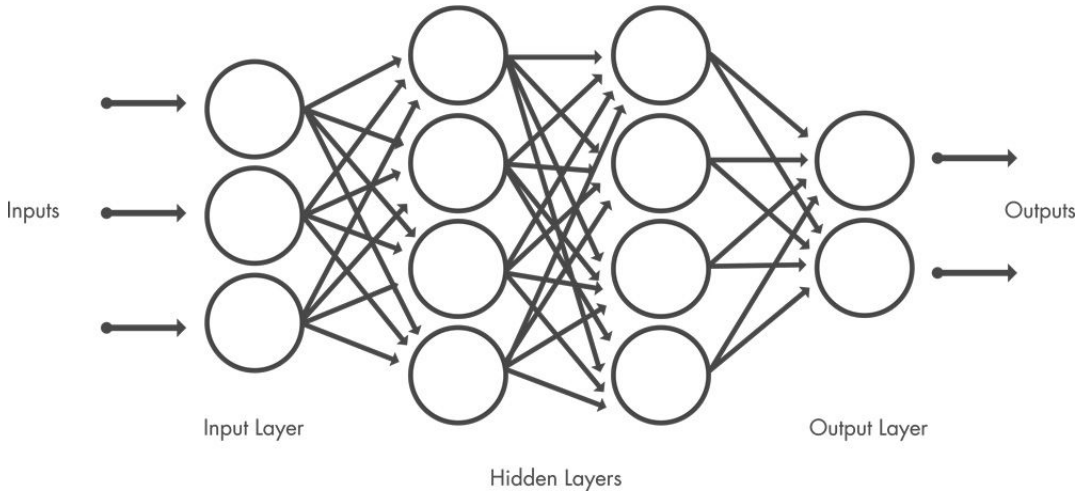


Figura 2.8: Capas de una red neuronal [55].

La red neuronal realiza predicciones, ya sea en forma de regresión o clasificación. Para evaluar la precisión de estas predicciones, la red compara los valores reales con los resultados obtenidos. Esta comparación genera un score o error, conocido como función de pérdida, que indica la calidad de la predicción. Un valor bajo en la función de pérdida sugiere que la predicción es precisa, mientras que un valor alto indica una mala predicción.

Para minimizar el error de la función de pérdida, se utiliza el descenso del gradiente o un optimizador que actualiza los pesos de la red. Después de esta actualización, se realiza una nueva predicción y se compara nuevamente con el valor real, generando así una nueva función de pérdida. Este proceso se repite de manera recursiva, aplicando el gradiente para reducir el error y encontrar el mínimo global, lo que permite obtener predicciones más precisas.

Redes Neuronales Convolucionales (CNN por sus siglas en inglés)

Las redes neuronales convolucionales son un tipo de arquitectura de red neuronal artificial diseñada para procesar datos estructurados en forma de cuadrícula, como las imágenes. Su funcionamiento se basa en múltiples capas interconectadas que extraen características. En las primeras capas, la CNN detecta patrones simples (bordes, colores o texturas). En capas más profundas, combina estas características para identificar estructuras complejas (formas, objetos). Las CNNs son capaces de realizar tareas avanzadas como clasificación de imágenes, detección de objetos o segmentación semántica.

La convolución es una operación matemática que combina dos funciones para producir una tercera. En el procesamiento de imágenes es muy utilizado para extraer características de las cuales se representan como matrices de tamaño $N \times M$. La operación de convolución se define como:

$$S(i,j) = \sum_m \sum_n I(i+m, j+n) \cdot K(m,n)$$

donde:

- $S(i,j)$ es el valor del píxel en la posición (i,j) de la imagen de salida.
- I representa la imagen de entrada.
- K es el kernel o filtro, una pequeña matriz que se desliza sobre la imagen.
- m y n recorren las dimensiones del kernel, permitiendo acceder a cada elemento del kernel.

Posteriormente, estas matrices se envían a una capa de agrupamiento (pooling), cuya función es reducir la complejidad de las imágenes al disminuir su ancho y largo. Esta etapa actúa en seleccionar características relevantes. Entre los métodos más utilizados para realizar esta reducción se encuentran el max pooling (que selecciona el valor máximo de una región) y el average pooling (que calcula el promedio de los valores en esa región). Este proceso de convolución y reducción se repite varias veces, generando imágenes de tamaño cada vez menor.

Una vez finalizado el bloque anterior, se emplea una capa Flatten que convierte el tensor tridimensional resultante en un vector unidimensional. Esta transformación es necesaria para conectar con la siguiente etapa del modelo: una red completamente conectada (Fully-Connected), compuesta por una o más capas densas. Estas capas son responsables de realizar la clasificación final, como se muestra en la Figura 2.9. Para mejorar la capacidad de generalización del modelo, también se puede incluir una capa de Dropout, la cual desactiva aleatoriamente ciertas neuronas durante el entrenamiento, reduciendo el riesgo de sobreajuste (overfitting) al evitar que la red dependa excesivamente de combinaciones específicas de neuronas.

En cada operación de convolución dentro de una CNN, se utiliza un filtro o núcleo (kernel), que es una pequeña matriz que se desliza sobre la imagen desde la esquina superior izquierda hasta la esquina inferior derecha. Durante este recorrido, se realiza una operación matemática que permite resaltar características específicas de la imagen, como bordes, texturas o patrones. Los tamaños más comunes de kernel son 3×3, 5×5 o 7×7 Si el kernel es demasiado grande en relación con la imagen, habría demasiadas características compitiendo dentro del campo receptivo, lo que dificultaría que la capa convolucional aprenda de manera efectiva.

El stride (o paso) se refiere al tamaño del desplazamiento que realiza el kernel al moverse sobre la imagen. Un stride de 1 píxel es el más común, ya que permite un

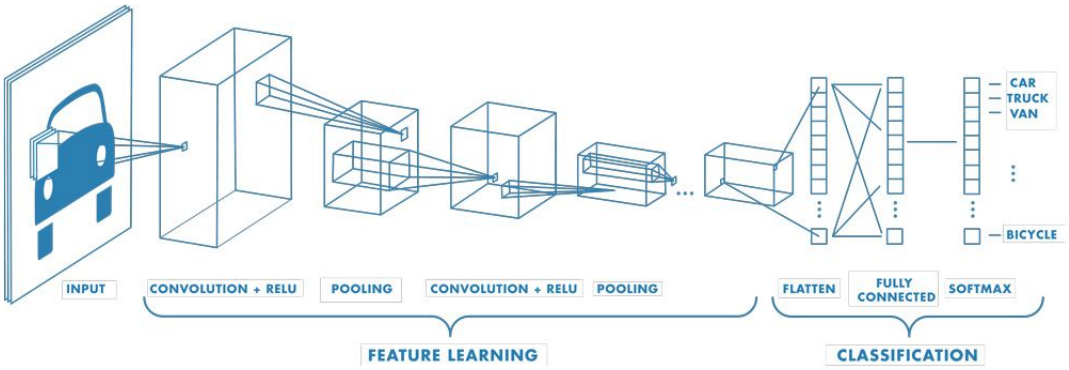


Figura 2.9: Arquitectura de una red neuronal convolucional [55].

análisis detallado. También se utilizan strides de 2 o 3 píxeles para reducir la resolución espacial del mapa de activación, aunque strides mayores pueden hacer que el modelo se salte regiones importantes de la imagen, afectando su capacidad de aprendizaje.

Para evitar que el tamaño de la imagen se reduzca tras cada operación de convolución, se puede aplicar padding (relleno), que consiste en añadir un borde de ceros alrededor de la imagen, que funciona en conjunto con el stride para mantener en orden los cálculos de una capa convolucional. Esta técnica permite conservar información valiosa, especialmente en las regiones cercanas a los bordes, y mantener el tamaño del mapa de activación igual al de la imagen original. Esto puede calcular con la siguiente ecuación:

$$Mapa\ de\ activación = \frac{D - K + 2P}{S} + 1$$

donde:

- D es el tamaño de la imagen (ya sea el ancho o la altura).
- K es el tamaño del filtro.
- P es la cantidad de padding (relleno).
- S es la longitud del stride (paso).

En la Figura 2.10 se muestra cómo una CNN procesa una imagen de entrada en blanco y negro de tamaño 28×28, la cual contiene un solo canal. El primer paso consiste en aplicar una operación de convolución, que utiliza un filtro (kernel) de tamaño 5×5. Como no se utiliza relleno (padding), las dimensiones de la imagen se reducen a 24×24 después de esta operación. La función principal de esta etapa es extraer patrones locales, como bordes o texturas. A continuación, se aplica una capa de Max-Pooling, cuya finalidad es reducir la dimensionalidad de la imagen y conservar las características más importantes. En este caso, se reduce el tamaño de la imagen a 12×12 píxeles. Se realiza nuevamente una convolución utilizando un kernel de 5×5 sin relleno, lo que reduce el tamaño de

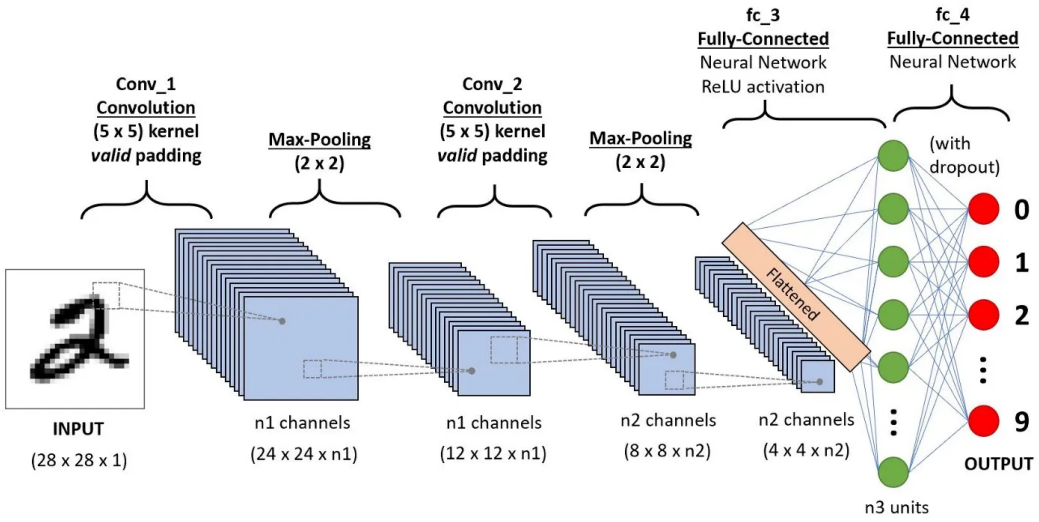


Figura 2.10: Ejemplo de una red neuronal convolucional [56].

la imagen a 8×8. A continuación, se aplica otra capa de Max-Pooling, que incrementa la profundidad de la salida de la capa anterior, mientras que el tamaño de la imagen se reduce a 4×4. Posteriormente, se utiliza la capa Flatten para convertir el tensor en una sola dimensión. Finalmente, se aplica una red Fully-Connected compuesta por capas densas, utilizando la función de activación ReLU, que es especialmente efectiva para las capas ocultas de la red neuronal. La capa densa final se encarga de realizar la predicción.

Métricas de evaluación

La evaluación del sistema propuesto requiere analizar tanto la etapa de segmentación como la etapa de clasificación basada en redes neuronales convolucionales (CNN). Por ello, se consideran métricas específicas para cada caso.

Evaluación de la segmentación

Para determinar la efectividad del algoritmo de segmentación, se utilizan métricas que permiten calcular el porcentaje de éxito en función del número de píxeles analizados. Como referencia se toma la segmentación manual de la imagen original, la cual desempeña como *ground truth* o verdad base. A partir de esta comparación se definen los siguientes valores:

- **Verdadero Positivo (VP):** área de la referencia correctamente identificada por el algoritmo.
- **Verdadero Negativo (VN):** área de fondo correctamente clasificada como tal.

- **Falso Positivo (FP):** área que el algoritmo identificó erróneamente como objeto.
- **Falso Negativo (FN):** área de la referencia que no fue detectada.

Cada una de estas cantidades se define como:

$$VP = |R_{ij} \cap S_{ij}| \tag{2.6}$$

$$VN = |\overline{R_{ij} \cup S_{ij}}| \tag{2.7}$$

$$FP = |R_{ij} \cup S_{ij} - R_{ij}| \tag{2.8}$$

$$FN = |R_{ij} \cup S_{ij} - S_{ij}| \tag{2.9}$$

donde S_{ij} es la imagen segmentada y R_{ij} es la imagen de referencia, ambas binarias. En este contexto, $|I|$ indica el número total de píxeles en la imagen I, mientras que $|\overline{I}|$ se refiere al complemento de la imagen I.

A partir de estos valores se calculan métricas estándar como sensibilidad (recall), precisión, exactitud (accuracy) y F1-score [57, 39]:

Exactitud (Accuracy)

Se refiere a la relación entre la zona de interés y el fondo que han sido identificados correctamente por el método propuesto, Ecuación 2.10.

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.10}$$

Precisión (Precision)

La precisión, representada por la Ecuación 2.11, hace referencia a la frecuencia con la que el algoritmo detecta todas las predicciones correctas.

$$Precision = \frac{VP}{VP + FP} \tag{2.11}$$

Sensibilidad (Recall)

La Ecuación 2.12 describe la frecuencia con la que el algoritmo detecta correctamente todos los casos que son verdaderos objetos a identificar dentro del conjunto de datos.

$$Recall = \frac{VP}{VP + FN} \tag{2.12}$$

F1-score

Es la media armónica entre la precisión (p) y la sensibilidad (r), dada por la siguiente Ecuación (2.13).

$$F1 = \frac{2 \cdot VP}{2 \cdot VP + FP + FN} \tag{2.13}$$

Evaluación de la clasificación con CNN

Una vez obtenidas las regiones de interés, la red neuronal convolucional (CNN) se encarga de clasificar cada cromosoma en su categoría correspondiente. En este caso, la evaluación se realiza a nivel de etiquetas, comparando las predicciones de la red contra las clases reales.

De manera análoga a la segmentación, se consideran los valores de VP, FP, FN y VN, pero ahora aplicados a la correcta o incorrecta clasificación de datos:

- **Verdadero Positivo (VP):** número de datos clasificados en la clase correcta.
- **Verdadero Negativo (VN):** número de datos que no pertenecen a una clase y que fueron correctamente clasificados como no pertenecientes a ella.
- **Falso Positivo (FP):** número de datos clasificados en una clase incorrecta.
- **Falso Negativo (FN):** número de datos de una clase real que no fueron reconocidos por el modelo.

A partir de estos valores se calculan métricas en problemas de clasificación multiclase:

Exactitud (Accuracy)

Mide el porcentaje de clasificaciones correctas respecto al total de datos analizados:

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} = \frac{N_{aciertos}}{N_{total}} \tag{2.14}$$

Precisión (Precision)

Indica la proporción de daros predichos como pertenecientes a una clase que efectivamente pertenecen a ella:

$$Precision = \frac{VP}{VP + FP} \tag{2.15}$$

Sensibilidad (Recall)

Mide la capacidad del modelo para detectar todos los datos de una clase determinada:

$$Recall = \frac{VP}{VP + FN} \tag{2.16}$$

F1-score

Es la media armónica entre la precisión y el recall, útil en casos de clases desbalanceadas:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{2.17}$$

Promedios en problemas multiclase

En clasificación de datos se deben calcular estas métricas para cada clase y luego promediarlas:

- **Promedio macro:** cada clase tiene el mismo peso, sin importar el número de ejemplos.
- **Promedio micro:** las métricas se ponderan según la cantidad de ejemplos en cada clase.

Finalmente, la **matriz de confusión** proporciona un resumen visual del desempeño del clasificador, indicando cuántos cromosomas de cada clase fueron clasificados correctamente y cuántos se confundieron con otras clases, como se muestra en la Tabla 2.2.

		Predicción		
		A	B	C
Real	A	10	2	1
	B	0	6	1
	C	0	3	8

Tabla 2.2: Matriz de confusión para la clasificación en tres clases [58].

Por ejemplo, la primera fila de la matriz indica que existen 13 objetos cuya clase verdadera es A. De ellos, 10 han sido clasificados correctamente como A, mientras que 2 fueron erróneamente clasificados como pertenecientes a la clase B y 1 como clase C.

2.4. Implementación del algoritmo en la Jetson nano

Para la implementación del algoritmo se utiliza la tarjeta de desarrollo de propósito específico Jetson Nano, diseñada por NVIDIA para aplicaciones de inteligencia artificial en dispositivos embebidos. En la Figura 2.11 se muestra la placa de desarrollo NVIDIA Jetson Nano. Este dispositivo cuenta con un procesador Quad-core ARM Cortex-A57 @ 1.43 GHz, una memoria RAM de 4 GB LPDDR4 de 64 bits, y una GPU de 128 núcleos CUDA basada en la arquitectura NVIDIA Maxwell. Además, dispone de interfaces de comunicación como GPIO, I2C, I2S, SPI y UART, y requiere una alimentación de 5 V

Figura 2.11: Jetson Nano.

y 4 A. Jetson Nano es una minicomputadora optimizada para ejecutar redes neuronales, lo que la hace ideal para aplicaciones como visión por computadora, clasificación de imágenes, detección de objetos, segmentación semántica y procesamiento de voz [59], [60].

2.4.1. Sistema Operativo

El sistema operativo utilizado en la Jetson Nano es Ubuntu 18.04 LTS, incluido en la distribución oficial JetPack, que proporciona un entorno completo para el desarrollo de aplicaciones de inteligencia artificial. Ubuntu es ampliamente adoptado en entornos de IA por su estabilidad, soporte comunitario y compatibilidad con herramientas de código abierto, lo que lo convierte en la opción preferida para proyectos de Aprendizaje Automático (Machine Learning) y Aprendizaje Profundo (Deep Learning). El JetPack SDK, en su versión 4.6.6, provee un entorno de escritorio Linux completo basado en Ubuntu 18.04 con gráficos acelerados, soporte para NVIDIA CUDA Toolkit 10.2, y bibliotecas optimizadas como cuDNN 8.2 y TensorRT 8.2. La Figura 2.12 muestra la información del software instalado en la Jetson Nano, incluyendo detalles sobre el sistema operativo y las versiones de las herramientas disponibles. Además, incluye la capacidad de instalar de forma nativa bibliotecas de aprendizaje automático como TensorFlow, PyTorch, Caffe, Keras y MXNet, así como librerías para visión por computadora (OpenCV) y desarrollo de robótica (ROS). Estas herramientas permiten aprovechar al máximo la aceleración por GPU, optimizar modelos para inferencia y desarrollar aplicaciones en áreas como visión artificial, procesamiento de imágenes y sistemas autónomos.

2.4.2. Python

Python es un lenguaje de programación de alto nivel, dinámico e interpretado, lo que significa que no requiere compilación para ejecutar aplicaciones. Está disponible para todos los sistemas operativos y es de código abierto, lo que lo convierte en una

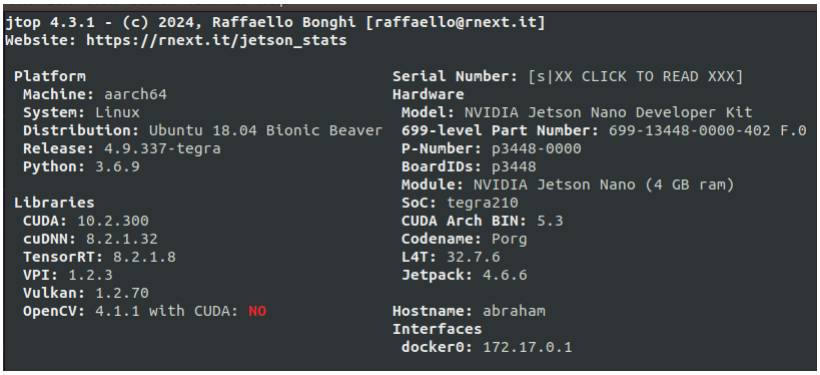


Figura 2.12: Información de la Jetson Nano.

opción ideal para proyectos de inteligencia artificial. Actualmente, Python es uno de los lenguajes más populares a nivel mundial y una de las mejores opciones para el desarrollo de IA, debido a su amplia gama de bibliotecas especializadas, como:

- NumPy: utilizada para manejar datos en matrices N-dimensionales.
- Pandas: proporciona estructuras de datos y herramientas analíticas fáciles de usar.
- Matplotlib: biblioteca para la generación de gráficos 2D.
- Scikit-learn: enfocada en algoritmos de aprendizaje automático.
- PyTorch y TensorFlow: para redes neuronales y aprendizaje profundo.
- OpenCV: para procesamiento de imágenes y visión por computadora.

La combinación de estas librerías y su naturaleza de código abierto, hacen de Python el lenguaje más utilizado en proyectos de IA.

2.4.3. Framework

La placa de desarrollo, cuenta con algunos frameworks para trabajar con los datos y aprovechar la aceleracion de la GPU. Entre ellos se encuentran dos muy utilizados Pytorch y Tensorflow. A continuación, se describen estos frameworks:

- Pytorch: desarrollada por Facebook, PyTorch es una biblioteca de código abierto para machine learning y deep learning. Basada en Python, su principal objetivo es ofrecer una alternativa más rápida a NumPy, facilitando un uso eficiente de las GPUs. PyTorch se destaca por su flexibilidad y velocidad.
- Tensorflow: es una biblioteca de aprendizaje automático desarrollada por Google. Esta biblioteca de código abierto ofrece una amplia gama de APIs para la construcción y entrenamiento de redes neuronales. TensorFlow es versátil, ya que puede ejecutarse tanto en CPU como en GPU.

Capítulo 3

Desarrollo de algoritmos para el procesamiento de imágenes

Este capítulo describe los algoritmos seleccionados para el desarrollo del sistema propuesta. El trabajo se dividirá en dos etapas. La primera etapa se centrará en la extracción de los cromosomas y en la separación de aquellos que se encuentren agrupados. En la segunda etapa, se llevará a cabo el enderezamiento de los cromosomas, así como su clasificación, validación e implementación del algoritmo en la tarjeta de desarrollo Jetson Nano. En la primera etapa, se llevará a cabo la extracción de los cromosomas, para lo cual será necesario realizar un preprocesamiento y segmentación.

3.1. Preprocesamiento y segmentación

Para el preprocesamiento y segmentación de los cromosomas se propone lo siguientes tres procesos:

⋆ Separación de objetos del fondo: En esta etapa consiste en separar los objetos del fondo mediante técnicas de binarización. En este trabajo se comparan dos métodos: Otsu y K-CAMS, con el fin de seleccionar el más adecuado para continuar con la siguiente etapa.

⋆ Reducción de objetos no deseados: Consiste en reducir los pequeños objetos presentes en la imagen que no corresponden a cromosomas. Estos elementos suelen ser ruido visual o fragmentos de material celular que pueden interferir con el análisis y la segmentación de los cromosomas. Para esta tarea, se aplican técnicas de filtrado y análisis morfológico que permiten conservar los elementos de mayor tamaño y reducir los que no son relevantes.

⋆ Discriminación de cromosomas individuales y agrupados: Una vez que se han segmentado los cromosomas del fondo, es necesario distinguir entre aquellos que se

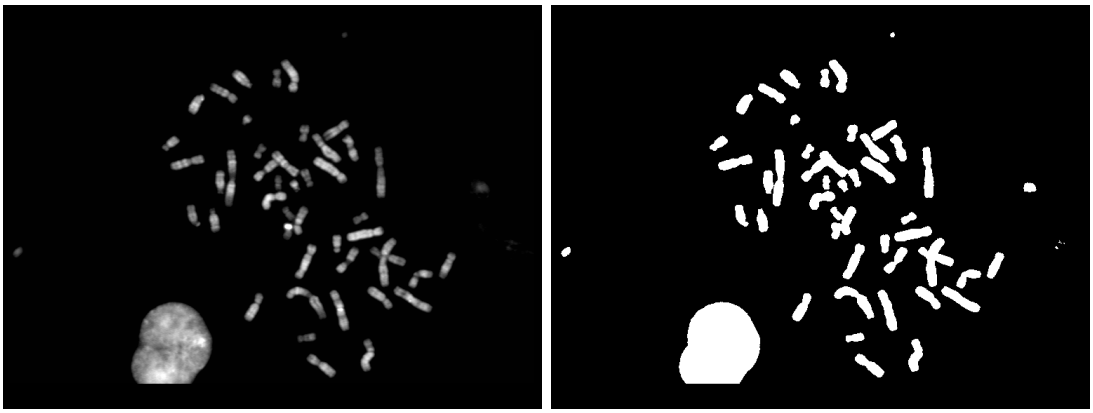
encuentran de forma individual y los que están en grupos (adyacentes o traslapados). Para esta clasificación, se aplican tres métodos complementarios:

- *Solidez*: basada en el análisis del menor polígono convexo que envuelve el objeto.
- *Excentricidad*: se ajusta una elipse a la figura y se analiza su alargamiento.
- *Adelgazamiento y detección de extremos*: se utiliza un algoritmo de esqueletización para reducir las figuras a sus esqueletos y detectar puntos extremos.

⋆ Identificación y etiquetado de cromosomas: En esta etapa se realiza un etiquetado entre cromosomas individuales y aquellos que se encuentran en grupo, ya sea en posiciones adyacentes o traslapados.

3.1.1. Separación de objetos del fondo

Consiste primero en convertir la imagen a escala de grises y, posteriormente, a una imagen binaria, como se muestra en la Figura 3.1. Para llevar a cabo esta binarización, se ha implementado un enfoque basado en Campos Aleatorios de Markov (Markov Random Fields) combinado con el algoritmo de K-means [43]. El uso de CAM junto con K-means permite una clasificación de los datos de la imagen en "k" grupos; en este caso, se requieren dos grupos para distinguir entre el fondo y los objetos de interés (cromosomas). Al aplicar esta técnica, el algoritmo agrupa los píxeles en función de la intensidad, generando una imagen binaria en la que los cromosomas y los objetos no deseados quedan resaltados respecto al fondo.



(a) Imagen de cromosomas desordenados. (b) Imagen de cromosomas binarizada

Figura 3.1: Imagen de cromosomas aplicando binarización.

3.1.2. Reducción de objetos no deseados

Para reducir parte del ruido, se suelen emplear operaciones morfológicas, que se utilizan para extraer estructuras relevantes o reducir objetos irrelevantes. En este caso, se aplicará la operación morfológica de apertura, que consiste en realizar primero una erosión y luego una dilatación, como se muestra en la Figura 3.2. La erosión se define de la siguiente manera:

$$I_B \otimes B = \mathbf{b} \in E^2 : \mathbf{d} + \mathbf{b} \in I_B \text{ para cada } \mathbf{b} \in B$$

La erosión de I_B por B produce un conjunto de puntos $\mathbf{d}$ para los cuales todos los puntos posibles de la forma $\mathbf{d} + \mathbf{b}$ pertenecen al conjunto I_B. Esta operación permite eliminar pequeños objetos y reducir el tamaño de estructuras indeseadas en la imagen.

Y la dilatación esta dada por

$$I_B \oplus B = \mathbf{d} \in E^2 : \mathbf{d} = i_B + \mathbf{b} \text{ para cada } i_B \in I_B \text{ y } \mathbf{b} \in B$$

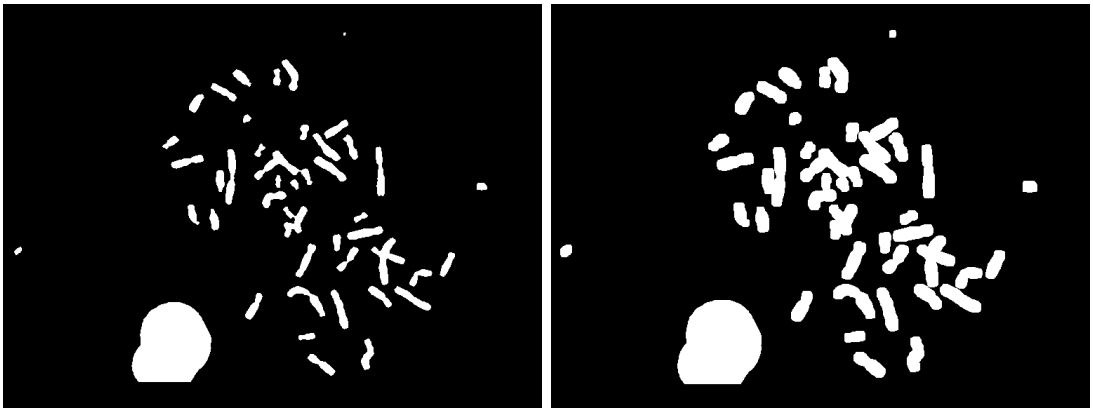
La dilatación de I_B por el elemento estructurante B genera un conjunto de puntos $\mathbf{d}$ tal que, para cada punto $\mathbf{i_B}$ perteneciente a I_B, existe un desplazamiento $\mathbf{b}$ en B. Este proceso expande el conjunto original al añadir una capa alrededor de sus bordes. La dilatación es especialmente útil para cerrar pequeños huecos en las estructuras, unir componentes cercanos y resaltar características en la imagen, ya que "amplía" los bordes de los objetos. Para que las operaciones morfológicas sean efectivas, es fundamental definir un elemento estructurante (B), el cual determina cómo se aplican las transformaciones sobre la imagen. Este elemento influye en las regiones de la imagen binaria que se expanden o se contraen durante el proceso.

En este trabajo, el elemento estructurante utilizado es un kernel de 3×3 compuesto por una matriz de unos, empleado en la operación morfológica de apertura. Dichas operaciones permiten modificar el número de píxeles en cada objeto de la imagen, reduciendo o aumentando su área según el caso. Es importante destacar que estas transformaciones son no invertibles: si se aplica erosión seguida de dilatación, la imagen original no se recupera completamente. En este contexto, el objetivo es reducir el ruido y aumentar el área de los objetos para facilitar su posterior extracción.

Para reducir la mayoría de objetos grandes y pequeños que no corresponden a cromosomas, se aplica un filtro de área, definiendo un rango de área mínima $Area_{min}$ y máxima $Area_{max}$. En este caso, al trabajar con imágenes de resolución $(576, 768)$ píxeles.

- Se ha propuesto un área mínima de 250 píxeles, lo que significa que cualquier objeto más pequeño que este tamaño será excluido del análisis, ya que probablemente no corresponde a un cromosoma.

- Se ha propuesto una área máxima de 3500 píxeles, o que implica que cualquier objeto más grande que este tamaño también será excluido, ya que podría ser ruido o una estructura no deseada que no representa un cromosoma.

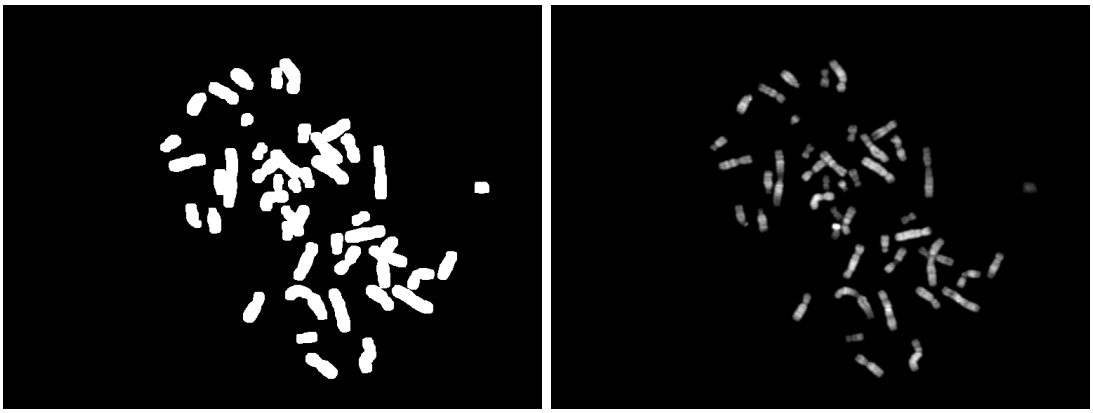


(a) Erosión. (b) Dilatación.

Figura 3.2: Aplicando operaciones morfológicas.

- Cabe mencionar que estos parámetros pueden ajustarse según las características de cada imagen y obtener mejores resultados.

En la Figura 3.3 se muestra (a) la imagen binarizada y (b) la imagen resultante tras aplicar el filtro, obteniendo los objetos de interés dentro de la área mínima y máxima.

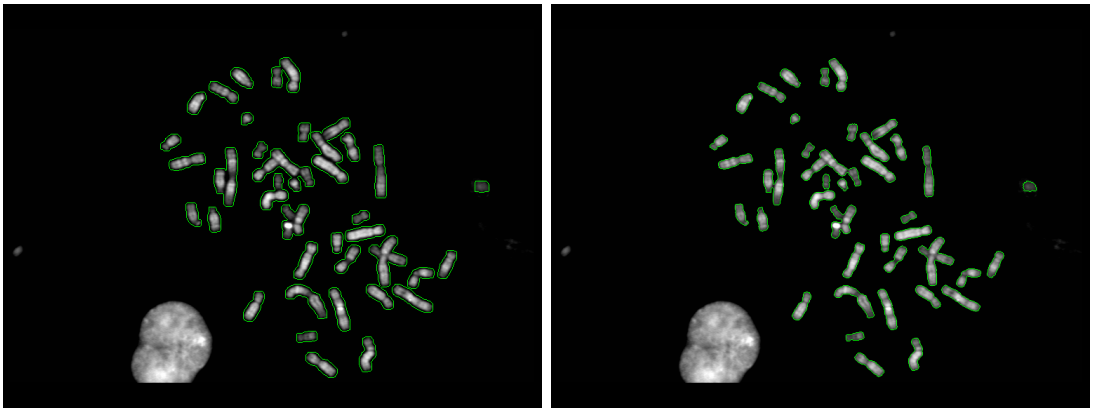


(a) Imagen binarizada filtrada. (b) Imagen filtrada.

Figura 3.3: Imagen filtrada.

Una vez obtenidos los objetos de interés, se procede a aplicar los contornos para delimitar los cromosomas. Sin embargo, como se observa en la Figura 3.4 (a), los contornos iniciales no encierran completamente los cromosomas. Para mejorar esta delimitación, se realiza una nueva binarización de la máscara, lo que permite obtener los bordes más cercanos a los cromosomas. Luego de este proceso de binarización, los contornos se extraen y se superponen a la imagen original, como se muestra en el inciso (b).

En la Figura 3.5 se presenta el diagrama de flujo para la reducción de objetos no deseados mediante operaciones morfológicas y un rango de área específico. Este algoritmo



(a) Imagen original con contornos. (b) Imagen filtrada con contornos.

Figura 3.4: Antes y después de aplicar contornos.

permite visualizar los contornos de los objetos de interés en color verde. La técnica binarizar utiliza la función K-CAM [43] , es un método diseñado para agrupar los datos de la imagen en escala de grises. En esta parte, se utiliza la binarización para resaltar y obtener los contornos de los objetos relevantes. El procedimiento se describe en el pseudocódigo correspondiente, incluido en los Anexos (véase Algoritmo 1), donde se mandan a llamar las operaciones morfológicas y el filtrado por área para la extracción de objetos de interés.

3.1.3. Identificación de cromosomas individuales y agrupados

Una vez extraídos los posibles objetos de interés candidatos a cromosomas, es necesario evaluarlos para distinguir entre cromosomas individuales y aquellos que se encuentran agrupados, parcialmente unidos o traslapados.

Como se mencionó anteriormente, para facilitar la identificación y clasificación, se utiliza un ideograma de referencia, como se muestra en la Figura 2.2. Los cromosomas presentes en la imagen pueden variar en tamaño, forma y orientación, lo que complica su análisis. Generalmente, los cromosomas individuales adoptan distintas configuraciones morfológicas, como se ilustra en la Figura 3.6: (a) forma de S, (b) forma de C, (c) forma de L. Estas variaciones suelen ser comunes en cromosomas individuales, mientras que los agrupados suelen presentarse en las posiciones (d)-(f), donde se observan adyacencias o superpuestos entre sí.

Para discriminar entre cromosomas individuales y agrupados, es necesario disponer de características que permitan diferenciarlos. Algunos autores, como en [39] y [50], emplean características geométricas y morfológicas, como el área del objeto, el área del polígono convexo, la excentricidad y el esqueleto del cromosoma que proporcionan información relevante de los cromosomas. Utilizan tres filtros que se basan en la geometría de los mismos. Los cuales son:

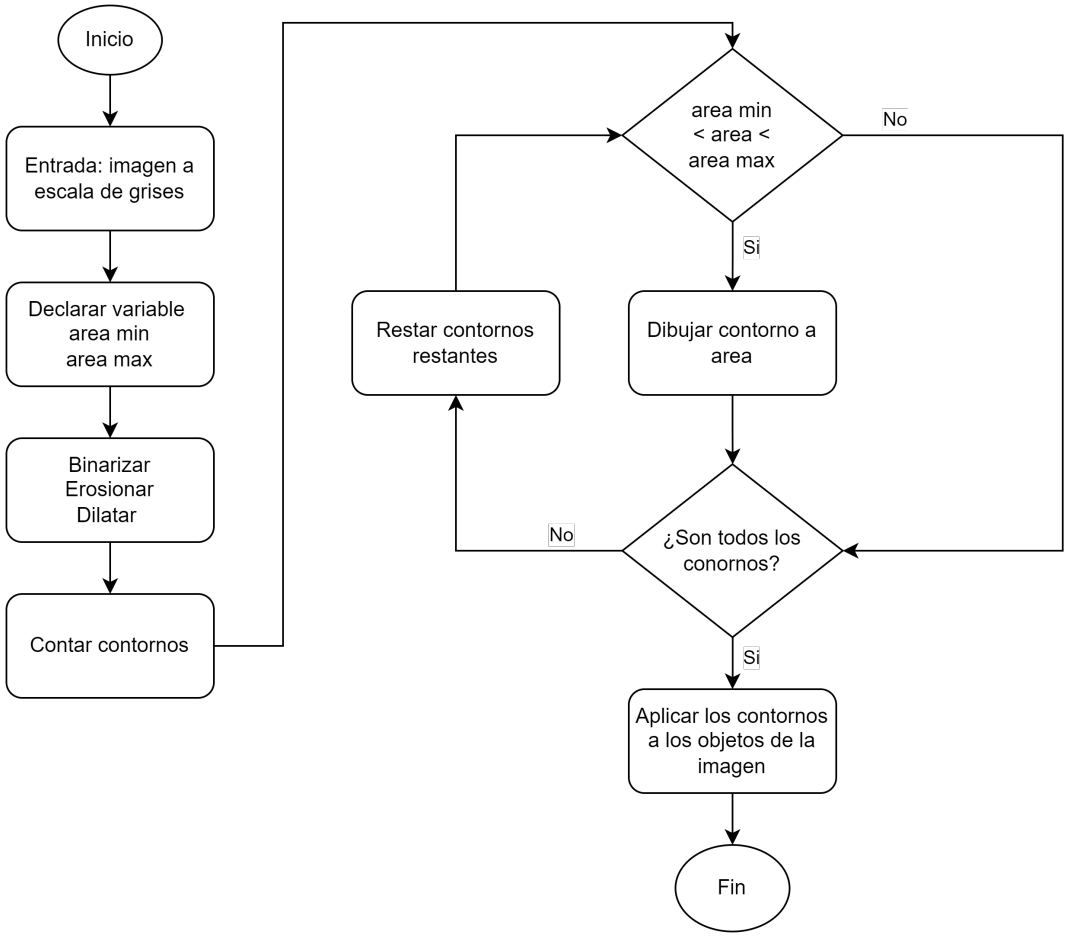


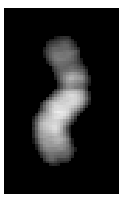
Figura 3.5: Reducción de objetos no deseados.

- Método del menor polígono convexo
- Método de la elipse que los rodea
- Método del esqueleto

Cada forma pasa a través de estos filtros y si satisface ciertos criterios entonces es considerada como cromosomas en grupo, de lo contrario es considerado un cromosoma individual.

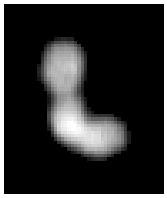
Método del menor polígono convexo

Este método se basa en el cálculo de la solidez del objeto. La solidez de un objeto puede calcularse como la relación entre su área y el área de su envolvente convexa, lo cual permite obtener el porcentaje de píxeles en el menor polígono convexo que también

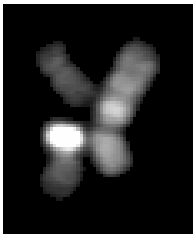


(a) Cromosoma en forma de S.

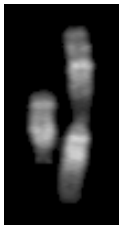
(b) Cromosoma en forma de C.



(c) Cromosoma en forma de L.



(d) Cromosomas adyacentes.



(e) Parcialmente adyacentes.

(f) Cromosomas traslapados.

Figura 3.6: Posibles formas de cromosomas individuales y en grupo.

forman parte del área del objeto. Esta métrica es útil para diferenciar entre cromosomas individuales y agrupaciones de cromosomas, ya que los cromosomas aislados tienden a ser convexos y tendrán una solidez cercana a uno, mientras que los grupos de cromosomas, debido a espacios entre los elementos, presentan una menor solidez.

La solidez S está dada por la siguiente expresión:

$$S = \frac{A_{obj}}{A_{mpc}}$$

donde:

- A_{obj} es el área del objeto, que representa el número de píxeles en la región de interés. Para un objeto binarizado, el área A se calcula como:

$$A_{obj} = \sum_{i=1}^{n} \sum_{j=1}^{m} obj[i, j]$$

- A_{mpc} Es el área del menor polígono convexo que rodea al objeto.

Los cromosomas individuales suelen ser aproximadamente convexos, lo que implica que el área del polígono convexo que los contiene es similar a su área real, dando una solidez cercana a uno. Por otro lado, cuando hay grupos de cromosomas, el área de la envolvente convexa será mayor que la suma de las áreas de los cromosomas individuales, reduciendo la solidez. Para detectar agrupaciones de cromosomas, se puede establecer un umbral en la solidez. Si la solidez del objeto es menor que este umbral, se clasifica el objeto como un grupo de cromosomas, lo cual permite pasar a una fase posterior de procesamiento para segmentar los elementos individuales dentro del grupo.

En Figura 3.7 se presenta el diagrama de flujo para obtener los contornos del menor polígono convexo de los cromosomas agrupados y los que están aislados. Este algoritmo recibe como entrada una imagen en escala de grises que contiene los objetos de interés y la lista de contornos. Posteriormente, se crea una lista vacía destinada a almacenar la solidez de cada objeto. Se usa un ciclo *for* para calcular el área de cada cromosoma, su polígono convexo y el área de dicho polígono. Se establece una condición para evitar la división por cero, y se calcula la solidez de cada objeto, almacenando los resultados en **solidez**. Finalmente, se procede a realizar una comparación de la solidez respecto a un umbral para la separación entre cromosomas individuales o agrupados. El procedimiento se presenta en el pseudocódigo del Algoritmo 2, incluido en los Anexos, donde se especifican las operaciones para la obtención del polígono convexo y el cálculo de la razón de solidez haciendo uso de algunas bibliotecas de python.

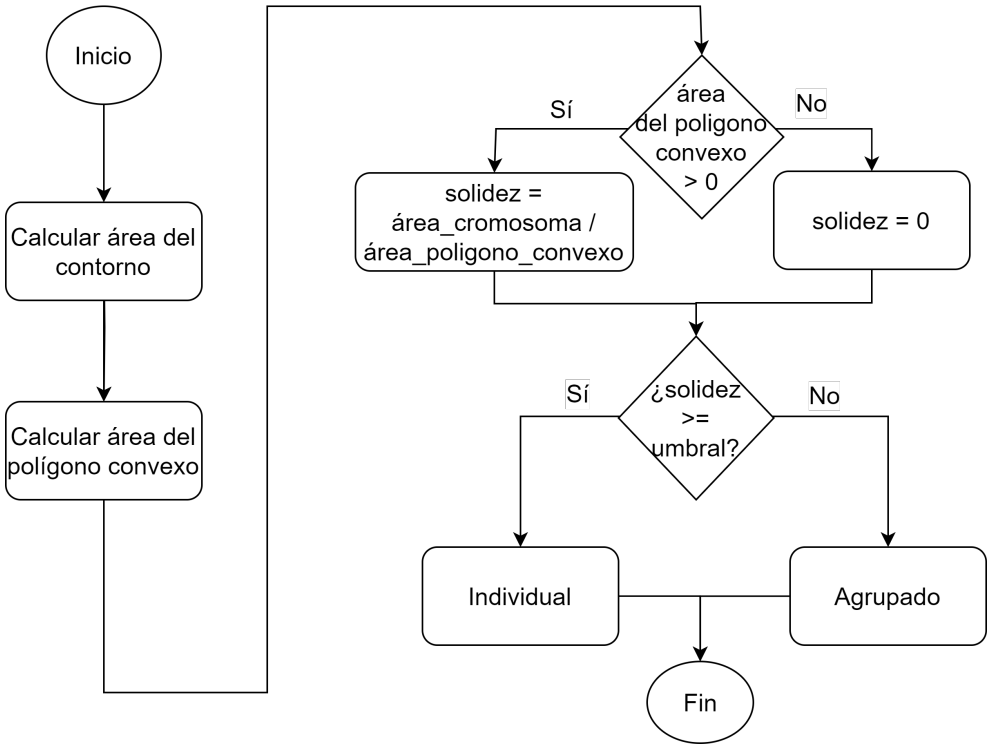


Figura 3.7: Método del menor polígono convexo.

En la Figura 3.8 se muestran cromosomas contenidos dentro de un polígono convexo, el cual se puede observar la solidez de cada objeto envuelto: los cromosomas individuales tienen una solidez cercana a 1, ya que su contorno se ajusta casi por completo al polígono que los envuelve. En cambio, los cromosomas que están unidos, curvados o traslapados presentan una solidez menor, pues el polígono convexo incluye áreas vacías alrededor de los objetos, lo que reduce la solidez.

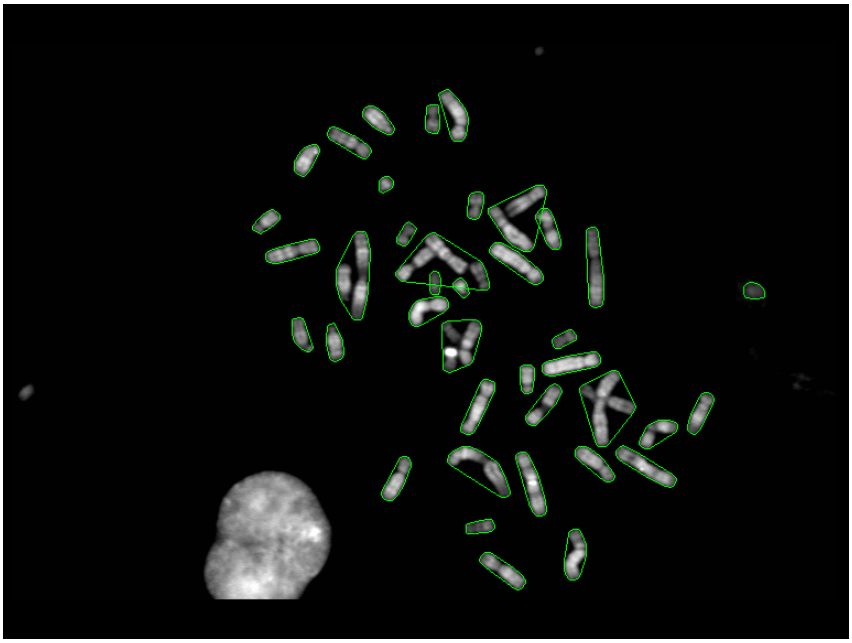


Figura 3.8: Cromosomas envueltos con el menor polígono convexo marcado de color verde.

Método de la elipse que los rodea

Por lo general, los cromosomas tienen una forma alargada y delgada, excepto algunos pares, como el 20, 21 y 22, que tienden a ser más cortos y compactos. Esta morfología hace que la elipse que rodea a un cromosoma individual sea alargada. Cuando dos o más cromosomas están traslapados o se encuentran unidos, la elipse que los envuelve tiende a acercarse a una forma circular. Basándose en esta idea, es posible detectar agrupaciones de cromosomas mediante la excentricidad de la elipse que rodea al objeto.

La excentricidad E se define como la relación entre el eje menor y el eje mayor de la elipse que envuelve al objeto:

$$E = \frac{eje\ menor}{eje\ mayor}$$

o bien

$$0 < \sqrt{1 - \frac{(eje\_menor)^2}{(eje\_mayor)^2}} < 1$$

La excentricidad, que varía entre 0 y 1, ayuda a diferenciar cromosomas individuales de grupos. Un valor de excentricidad próximo a 1 indica una forma circular, caracterizando un grupo de cromosomas traslapados o unidos. En cambio, un valor bajo indica un cromosoma individual. En la Figura 3.9 se presenta en diagrama de flujo para encerrar en elipses a los cromosomas que no están agrupados o superpuestos, mientras que aquellos que sí lo están se encierran en círculos. El proceso comienza con la imagen filtrada y

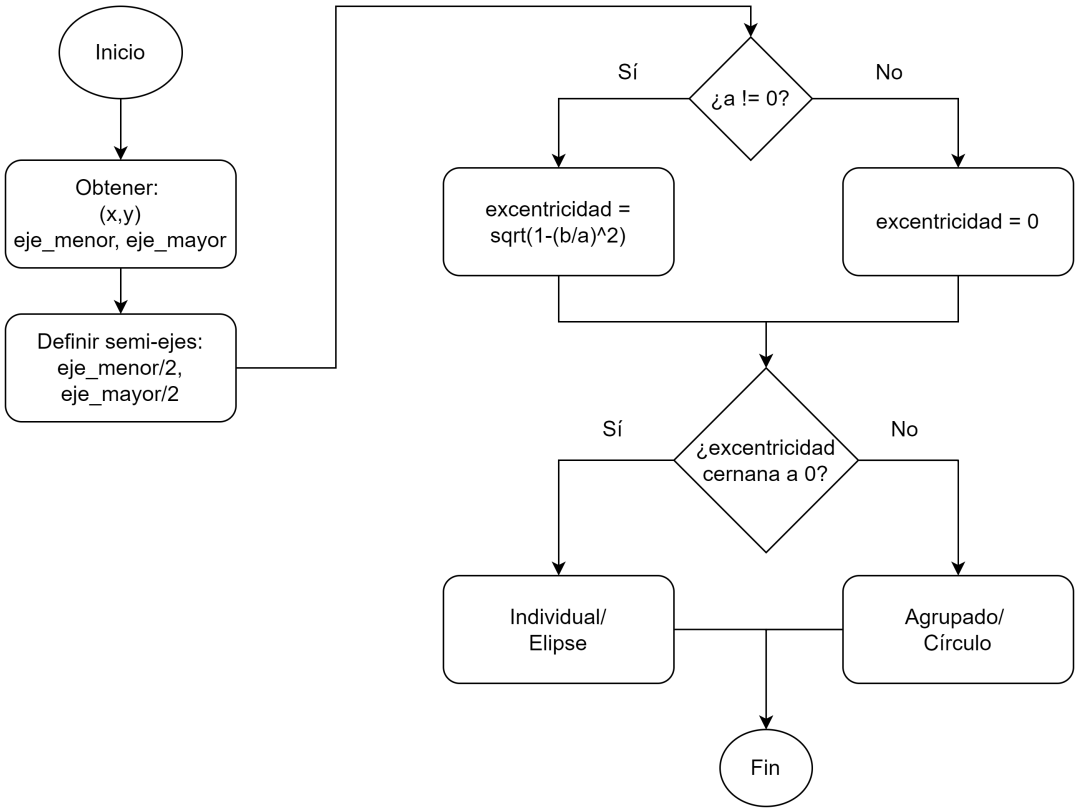


Figura 3.9: Método de la elipse que los rodea.

los contornos identificados. Se inicializan dos listas: una para almacenar la razón de los ejes de la elipse de cada cromosoma y otra para guardar las elipses correspondientes a cada cromosoma. A continuación, para cada contorno de los cromosomas, se extraen los datos necesarios para determinar las posibles elipses. Se calcula un umbral promedio que permite filtrar los cromosomas individuales de los grupales. Finalmente, se dibuja una elipse de color verde alrededor de cada cromosoma individual y un círculo de color rojo alrededor de cada grupo de cromosomas detectado. Dentro del procedimiento se presenta en el pseudocódigo del Algoritmo 3, puede verse en los Anexos, donde se aplican operaciones para el cálculo de excentricidad y la clasificación visual mediante elipses y círculos haciendo uso de las bibliotecas de python.

En la Figura 3.10 se muestran los cromosomas envueltos por una elipse. Los cromosomas individuales, que presentan una forma más alargada, están rodeados por una elipse de color verde. Por otro lado, los cromosomas agrupados o traslapados están rodeados por un círculo rojo. Sin embargo, algunos cromosomas individuales también aparecen en rojo debido a su pequeño tamaño o forma curva, que los aleja de una elipse alargada.

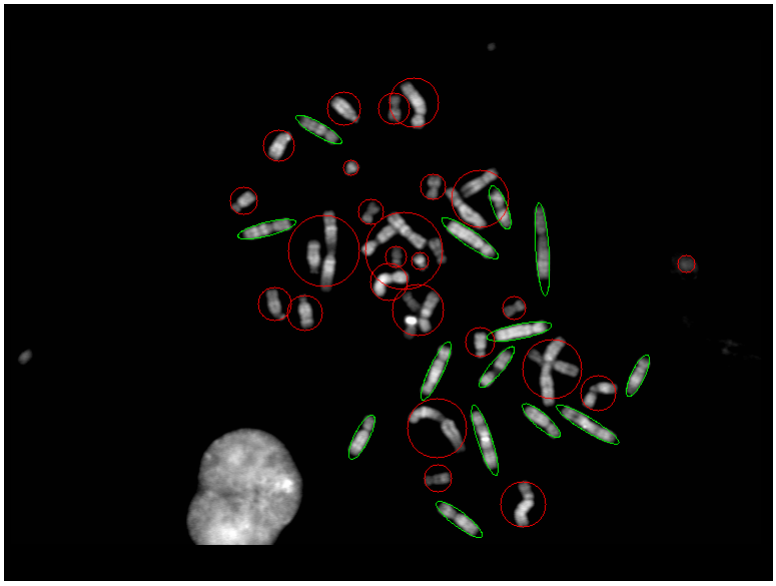


Figura 3.10: Los cromosomas están envueltos mediante el método de elipse. Los cromosomas individuales están rodeados por una elipse de color verde, mientras que los agrupados, unidos, traslapados o curvados se muestran rodeados por un círculo de color rojo.

Es por esto que se pueden presentar algunas limitaciones:

- Cromosomas pequeños: Estos suelen ser identificados correctamente mediante el criterio del menor polígono convexo, pero en este caso el método de la elipse lo envuelve en un circulo.

- Cromosomas curvados: La forma curvada puede afectar el cálculo de la elipse acercarse a un círculo.

Esqueletos de los cromosomas

El tercer filtro consiste en identificar y contar los puntos terminales presentes en el esqueleto de cada cromosoma. Para lograr el esqueleto, suelen emplear operaciones morfológicas como la erosión y la apertura, que permiten adelgazar un objeto hasta convertirlo en una línea o secuencia de píxeles que conserva la estructura del objeto original [39], [50]. Sin embargo, existen otros enfoques para el adelgazamiento de objetos, como el algoritmo de Zhang-Suen [61]. Este algoritmo se enfoca en extraer el esqueleto del objeto, manteniendo la conectividad y la topología de las estructuras originales. No obstante, presenta limitaciones, ya que no garantiza que el esqueleto resultante tenga un ancho de un solo píxel y puede generar segmentos redundantes. Wei Chen et al. [47] proponen una mejora al algoritmo de Zhang-Suen que aborda estas limitaciones,

asegurando que el esqueleto obtenido tenga un ancho de un solo píxel y eliminando los segmentos redundantes.

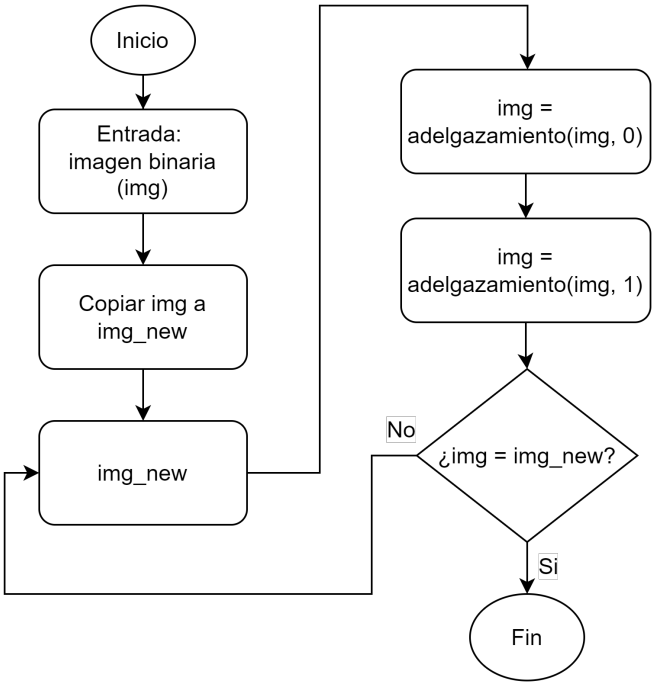


Figura 3.11: Algoritmo para obtener el esqueleto de los cromosomas.

Dentro del bucle principal se implementa la función de adelgazamiento, ilustrada en la Figura 3.12. Esta función recibe la imagen binaria y un valor de iteración (0 o 1), iniciando una búsqueda en la vecindad de ocho píxeles alrededor del píxel central. Para cada píxel, se calcula la suma de vecinos (A) y el número de transiciones $0 \rightarrow 1$ (C) en sentido horario, con el objetivo de identificar trayectorias y descartar bordes. Según la subiteración, se aplican condiciones geométricas que evitan la eliminación de puntos críticos y preservan la conectividad. Los píxeles que cumplen simultáneamente los criterios ($img = 1$, $2 \leq A \leq 6$, $C = 1$ y las restricciones de producto) se marcan para borrado diferido, es decir, se eliminan en bloque al final de cada pasada. Este proceso se repite hasta que no existan cambios entre iteraciones, obteniendo finalmente el esqueleto con grosor unitario, como se detalla en el pseudocódigo del Algoritmo 4. El algoritmo utiliza dos métricas para evaluar cada píxel en su vecindad:

Para cada píxel P_1 de primer plano, se consideran sus ocho vecinos $P_2, \ldots, P_9$ en sentido horario dentro de una ventana 3×3. Se definen las siguientes métricas:

$$A(P_1) = \sum_{k=2}^{9} P_k \tag{3.1}$$

donde $A(P_1)$ representa el número de vecinos de primer plano, y

$$C(P_1) = \sum_{k=2}^{9} \big[(P_k = 0) \wedge (P_{k+1} = 1)\big], \quad P_{10} = P_2 \tag{3.2}$$

donde $C(P_1)$ indica el número de transiciones $0 \rightarrow 1$ en la secuencia circular de vecinos.

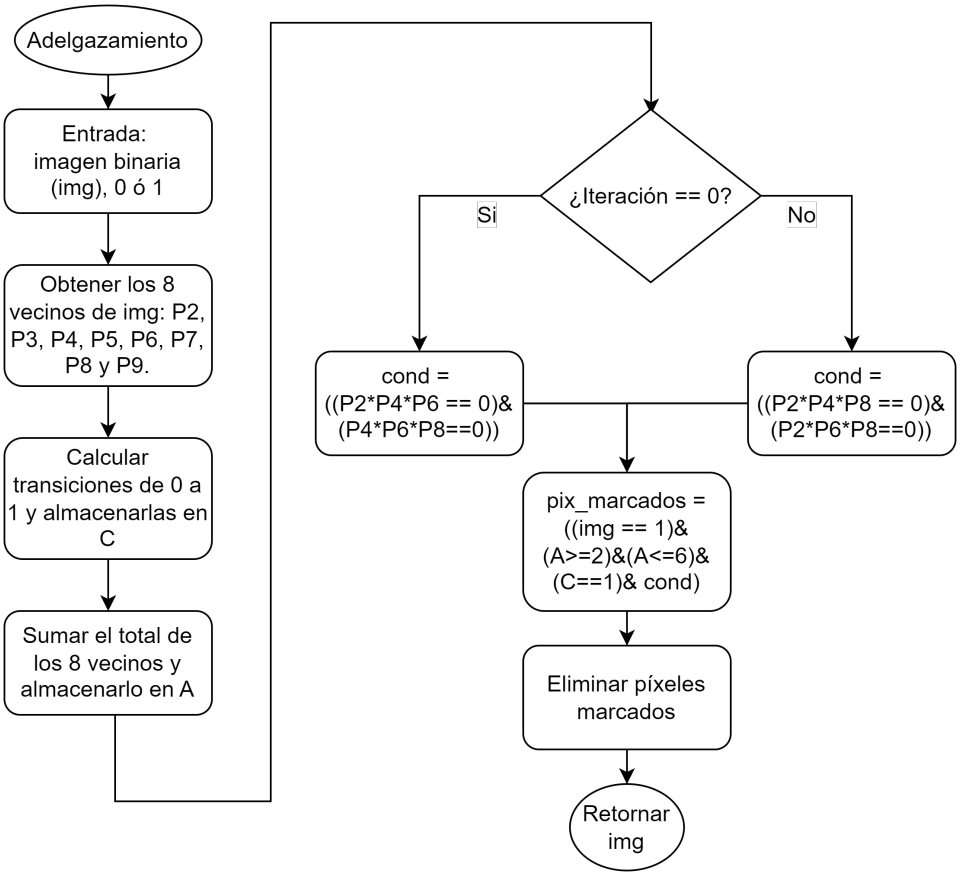


Figura 3.12: Algoritmo para adelgazar los objetos.

La identificación y conteo de puntos terminales en el esqueleto de los cromosomas es útil para discriminar cromosomas individuales de aquellos que se encuentran agrupados, ya que los cromosomas aislados suelen presentar dos extremos, mientras que los agrupados pueden mostrar múltiples ramificaciones.

En la Figura 3.13 el proceso comienza con la imagen binaria previamente segmentada, sobre la cual se genera el esqueleto mediante el algoritmo de adelgazamiento. Una vez obtenido el esqueleto, se calcula la cantidad de vecinos para cada píxel utilizando una convolución con una máscara 3×3 (excluyendo el centro). Los píxeles que pertenecen

al esqueleto y tienen exactamente un vecino son considerados puntos terminales. Estos puntos se almacenan en una lista y se contabilizan para obtener el número total de extremos.

Finalmente, se genera una imagen combinada que muestra el esqueleto en color verde y los puntos terminales resaltados con círculos rojos, lo que facilita la interpretación visual del resultado. El procedimiento se detalla en el pseudocódigo del Algoritmo 5, incluido en los Anexos.

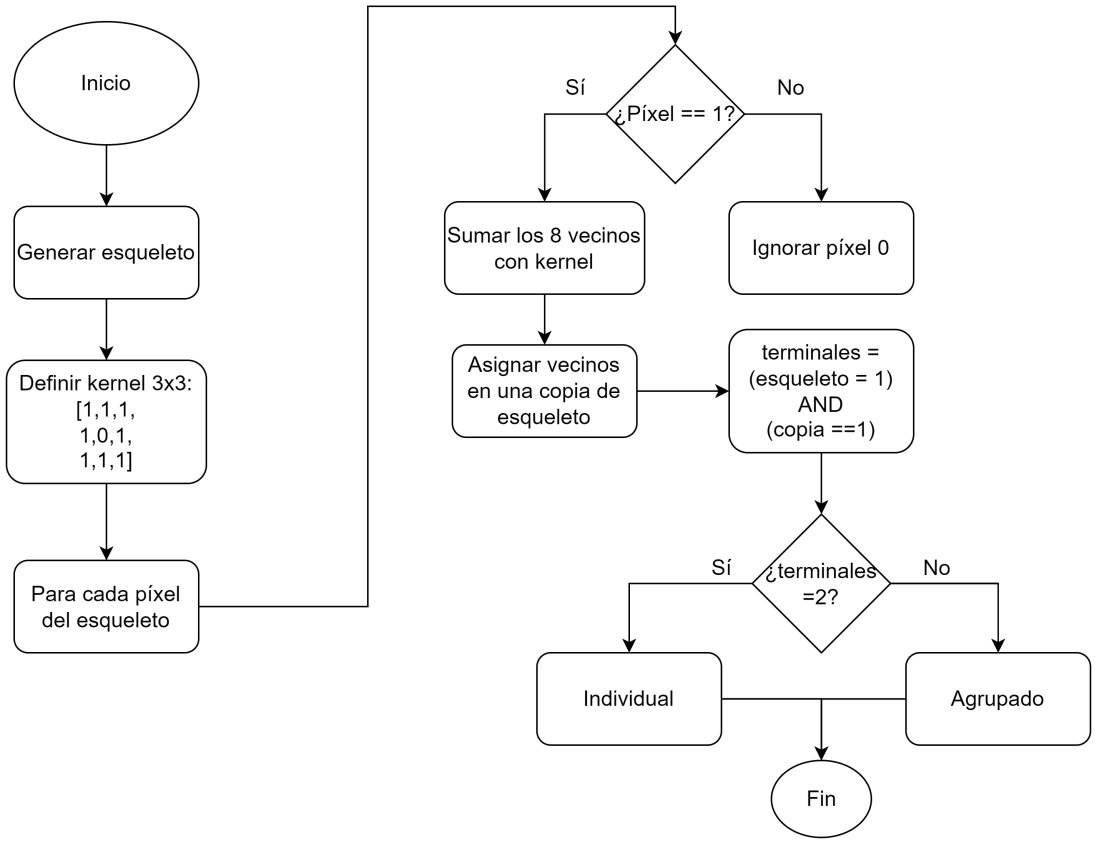
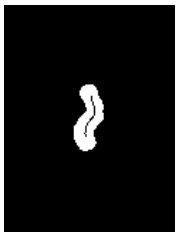


Figura 3.13: Algoritmo para la obtención de la puntas.

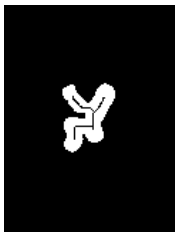
En la Figura 3.14 se presentan los esqueletos de los cromosomas que han sido extraídos, mostrando su estructura. Los puntos terminales del esqueleto son píxeles que permiten el análisis de las ramificaciones del esqueleto del objeto. La existencia de más de dos puntos terminales sugiere la presencia de dos o más cromosomas, se puede observar en la Figura 3.15. Para identificar los puntos terminales, se analiza la vecindad de cada uno de los píxeles que componen el esqueleto del objeto. Si el valor de cualquiera de los ocho píxeles adyacentes al punto en cuestión coincide con el del punto analizado, se identifica como un punto terminal.



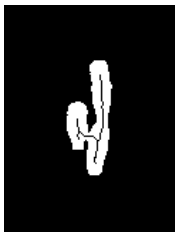
(a) Cromosoma en forma de S.

(b) Cromosoma en forma de C.

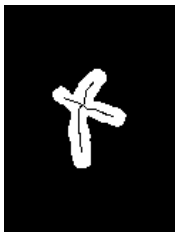
(c) Cromosoma en forma de L.



(d) Cromosomas adyacentes.

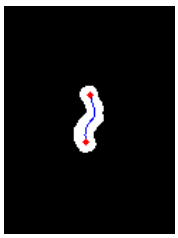


(e) Parcialmente adyacentes.

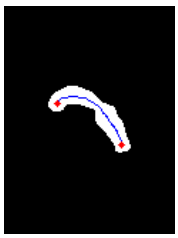


(f) Cromosomas traslapados.

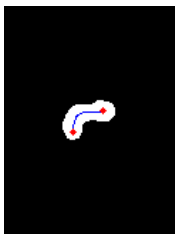
Figura 3.14: Esqueletos de algunos cromosomas.



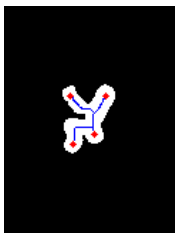
(a) Cromosoma en forma de S.



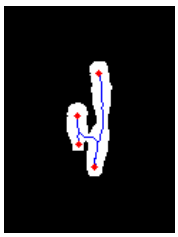
(b) Cromosoma en forma de C.



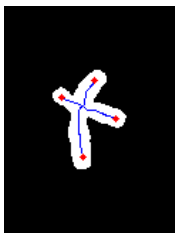
(c) Cromosoma en forma de L.



(d) Cromosomas adyacentes.



(e) Parcialmente adyacentes.



(f) Cromosomas traslapados.

Figura 3.15: Puntas terminales de algunos de los esqueletos de los cromosomas.

Organización de los cromosomas en carpetas

A continuación, se describe de manera general el flujo seguido para organizar la discriminación entre cromosomas agrupados e individuales:

1. **Segmentación inicial**
   - Convertir la imagen a escala de grises.
   - Aplicar binarización para separar cromosomas del fondo.
2. **Reducción de objetos no deseados**
   - Aplicar operaciones morfológicas (erosión y dilatación).
   - Filtrar por área mínima y máxima para excluir ruido y fragmentos.
3. **Discriminación entre cromosomas individuales y agrupados**
   - **Filtro 1 (Solidez):** Calcular el área y el polígono convexo; si la solidez $<$ umbral, clasificar como agrupado.
   - **Filtro 2 (Excentricidad):** Ajustar una elipse; si la relación de ejes indica forma circular, mantener como agrupado; si es alargada, reclasificar como individual.
   - **Filtro 3 (Esqueletización):** Generar el esqueleto y contar puntas; dos puntas $\rightarrow$ individual, más de tres $\rightarrow$ agrupado.
4. **Reasignación y organización**
   - Reubicar cromosomas según los resultados de los filtros.
   - Crear dos carpetas:
     - `individuales/` $\rightarrow$ cromosomas para clasificación.
     - `agrupados/` $\rightarrow$ cromosomas almacenados.
5. **Exportación y trazabilidad**
   - Guardar imágenes con sus respectivos identificadores.

Lo que se espera al final de la Etapa 1 es la obtención de un conjunto organizado de carpetas que contienen los cromosomas individuales y los agrupados, resultado de aplicar los filtros mencionados anteriormente (solidez, excentricidad y conteo de puntas). Cada cromosoma individual se almacena en su carpeta correspondiente para la fase de clasificación, mientras que los agrupados se almacenan en otra carpeta denominada "agrupados".

3.2. Clasificación

Una vez que se finaliza la etapa anterior, se procede a la fase de clasificación de aquellos cromosomas que han sido segmentados individualmente. Para esta tarea, se diseñó una red neuronal convolucional (CNN) utilizando la biblioteca Keras, propia de TensorFlow. Esta herramienta permite construir, entrenar y gestionar modelos de aprendizaje profundo, además de ofrecer la posibilidad de guardar y cargar modelos mediante archivos con extensión `.h5`, lo que facilita la reutilización y el despliegue del modelo entrenado. El modelo se desarrolló con el objetivo de aprender patrones espaciales y estructurales de los cromosomas a partir de imágenes en escala de grises con una resolución de 128×64 píxeles. Estas imágenes constituyen la entrada de la red neuronal convolucional, mientras que la capa de salida está compuesta por 24 neuronas. Estas neuronas representan las 24 clases del cariograma humano: los 22 pares autosómicos (clases 1-22), el cromosoma X (clase 23) y el cromosoma Y (clase 24).

3.2.1. Entrenamiento y validación

El modelo CNN consta de las siguientes capas como se muestra en la Figura 3.16:

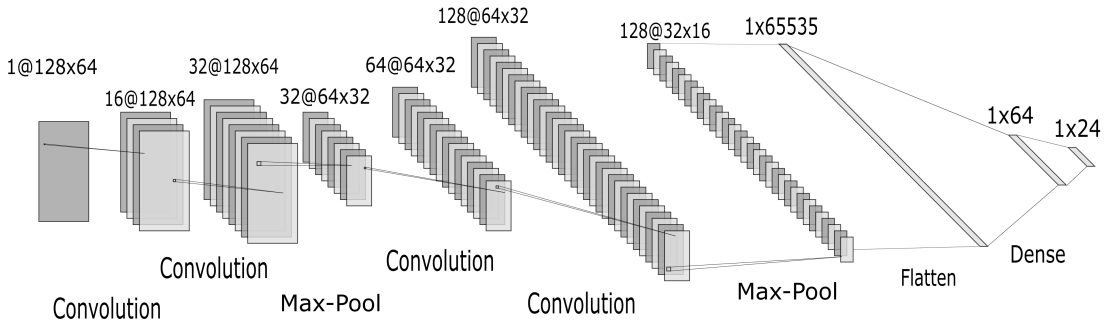


Figura 3.16: Arquitectura de la red propuesta.

- Entrada: Imágenes en escala de grises con resolución de 128×64 píxeles.
- Primera capa convolucional: 16 filtros de tamaño 3×3, activación ReLU y padding *same*, preservando la resolución espacial.
- Segunda capa convolucional: 32 filtros con kernel 5×5, seguida de una capa de *Batch Normalization* para estabilizar el aprendizaje.
- MaxPooling y Dropout: Reducción de resolución mediante *max pooling* 2×2 y *dropout* del 10 % para prevenir sobreajuste.
- Tercera capa convolucional: 64 filtros de tamaño 3×3, seguida de una cuarta capa con 128 filtros de tamaño 5×5, ambas con activación ReLU y normalización por lotes.

- Segunda etapa de MaxPooling y Dropout: Se aplica nuevamente *max pooling* 2×2 y *dropout* del 20 %.
- Global Average Pooling: Para reducir la dimensionalidad y mantener la información más relevante de las características extraídas.
- Capas densas: Una capa densa con 64 neuronas y activación ReLU, seguida de un *dropout* del 30 %. Finalmente, una capa de salida con 24 neuronas con activación *softmax*.
- Entrenamiento: Se utilizó el optimizador Adam con tasa de aprendizaje de 0.0005, 24 épocas, *class weights* para balancear las clases.
- Resultados: Precisión en entrenamiento: 98.92 %, precisión en validación: 81.25 %.

Preprocesamiento de datos

Para el entrenamiento del modelo de clasificación se utilizó la base de datos [62], la cual contiene un conjunto de imágenes correspondientes a los 23 pares de cromosomas humanos: 22 pares autosómicos y el par sexual (X, Y). El conjunto total de datos se dividió en dos subconjuntos: 80 % para entrenamiento y 20 % para validación.

Base de datos

El conjunto total de datos consta de 5,474 imágenes, cromosomas individuales, divididas en:

- 80 % para entrenamiento (4,370 imágenes)
- 20 % para validación (1,104 imágenes)

→ Carpeta de entrenamiento:

- Contiene 4,370 imágenes distribuidas en 24 clases.
- Las clases del 1 al 22 tienen 190 imágenes cada una.
- La clase 23 tiene 154 imágenes.
- La clase 24 cuenta con 36 imágenes.

→ Carpeta de validación:

- Contiene 1,104 imágenes.
- Las clases del 1 al 22 tienen 48 imágenes cada una.
- La clase 23 tiene 39 imágenes.
- La clase 24 tiene 9 imágenes.

Acondicionamiento

Durante la etapa de entrenamiento y validación de la red neuronal convolucional (CNN) propuesta, se realizó un acondicionamiento previo de los datos con el objetivo de mejorar la calidad del aprendizaje y reducir el riesgo de sobreajuste. Este acondicionamiento incluye:

- Conversión de formato de imagen: Las imágenes originales en formato BMP son convertidas a PNG, lo que reduce el tamaño del archivo y mejora la compatibilidad con librerías de procesamiento.
- Normalización de los valores de intensidad: Ajuste de los valores de los píxeles al rango [0,1], lo que permite una convergencia más estable durante el entrenamiento.
- Centrado del cromosoma en el recuadro: Cada cromosoma es reposicionado en el centro de la imagen para garantizar que la red neuronal reciba patrones consistentes, reduciendo la variabilidad espacial.
- Aplicación de *padding*: Se añade relleno para unificar el tamaño de todas las imágenes, evitando distorsiones y asegurando que la arquitectura de la CNN procese entradas homogéneas.
- Redimensionamiento uniforme: Todas las imágenes son ajustadas a una resolución estándar para la entrada de la red (128×64 píxeles), lo que facilita el procesamiento y reduce la carga computacional.
- Balanceo de clases: Se incorporan técnicas para compensar el desbalance entre clases, como el uso de *class weights* durante el entrenamiento, evitando que el modelo favorezca clases mayoritarias.

Estos pasos no solo mejoran la calidad del conjunto de datos, sino que también dan estabilidad en el entrenamiento, incrementan la capacidad de generalización del modelo y disminuyen el riesgo de sobreajuste. En conjunto, el acondicionamiento previo se traduce en una mejora significativa en la precisión y robustez del sistema, sentando las bases para un proceso de clasificación más confiable y eficiente. En la Figura 3.17 se muestra el cromosoma 1 en distintas situaciones. Algunos cromosomas se encuentran muy ajustado al borde del recuadro, otras se encuentra descentrado o mal posicionado, lo que puede afectar el rendimiento del modelo. Además, las imágenes se encontraban originalmente en formato BMP, lo que requería su conversión a un formato PNG. Este tipo de variaciones puede introducir ruido en el entrenamiento y dificultar la capacidad de la CNN para aprender patrones.

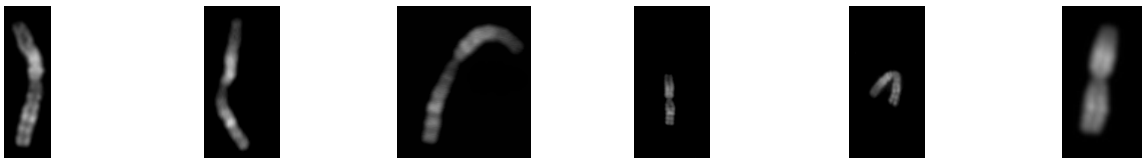


Figura 3.17: Cromosomas antes del acondicionamiento.

3.2.2. Ordenamiento

Una vez completado el entrenamiento y validación del modelo de la red neuronal convolucional (CNN) propuesta, se procede a la etapa de identificación y ordenamiento de los cromosomas. Para ello, se retoman los cromosomas individuales previamente discriminados durante la etapa de segmentación. Antes de ser procesados por la CNN, las imágenes deben pasar por un preprocesamiento adicional, conversión a escala de grises, normalización de los valores de intensidad en un intervalo de 0 a 1, centrado del cromosoma dentro de un recuadro de tamaño 120×160 píxeles, y posteriormente redimencionarlo a la entrada de la red a un tamaño de 64×128.

Para cada imagen, se aplica el preprocesamiento, se obtiene la predicción y se asigna la clase correspondiente. Este algoritmo recorre todas las imágenes segmentadas, aplica el preprocesamiento y utiliza el modelo CNN para predecir la clase. La salida es una lista con la ruta de cada imagen y su clase asignada, que servirá para el ordenamiento.

Una vez obtenidas las predicciones, el siguiente algoritmo organiza los cromosomas en una malla, generando la imagen final del ordenamiento y clasificación de los cromosomas. Toma las predicciones y organiza los cromosomas en una malla de 6×8. Cada cromosoma se coloca en su posición correspondiente y se genera la imagen final de los cromosomas organizados.

Los pseudocódigos que explican el proceso anteriormente mencionado se encuentran en los Anexos 6 y 7. A continuación, se describe de manera gráfica el algoritmo que implementa esta etapa. Una vez que las imágenes han sido preprocesadas, se carga el modelo entrenado (archivo con extensión `.h5`) y se realiza la identificación automática de cada cromosoma. Posteriormente, los cromosomas clasificados se ordenan según su categoría, lo que permite construir el cariograma final. En la Figura 3.18 se muestra el diagrama que representa el flujo del algoritmo encargado de la identificación y el ordenamiento de los cromosomas.

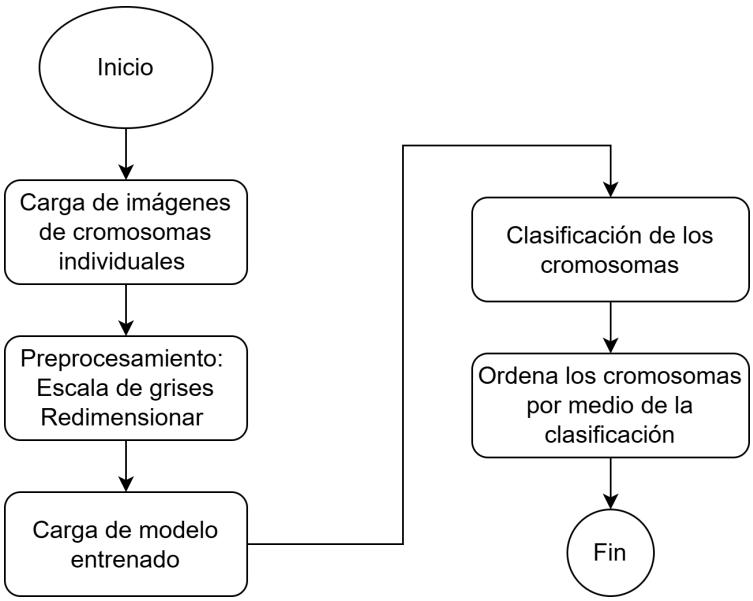


Figura 3.18: Algoritmo para la obtención de la puntas.

3.3. Implementación

La implementación del algoritmo se realizó en la tarjeta de desarrollo NVIDIA Jetson Nano, un dispositivo optimizado para aplicaciones de inteligencia artificial en entornos embebidos. Esta plataforma utiliza el sistema operativo Ubuntu 18.04 LTS, incluido en la distribución JetPack, que proporciona soporte para bibliotecas aceleradas por GPU como CUDA, cuDNN y TensorRT, esenciales para la ejecución eficiente de redes neuronales convolucionales (CNN). El entorno de desarrollo se basa en Python 3.6.9, aprovechando librerías ampliamente utilizadas en visión por computadora y aprendizaje profundo, tales como:

- **OpenCV**: Para operaciones de procesamiento de imágenes, segmentación y filtrado.
- **NumPy**: Para manipulación de matrices y operaciones numéricas.
- **Matplotlib**: Para la visualización de resultados y métricas.
- **TensorFlow/Keras**: Para la construcción, entrenamiento y despliegue del modelo CNN.
- **Scikit-learn**: Para cálculos de métricas y validación del rendimiento.

El algoritmo es desarrollado en Python, aplicando técnicas de visión por computadora para la segmentación y redes neuronales convolucionales para la clasificación. La Jetson Nano, por su GPU integrada con 128 núcleos CUDA, permite acelerar las operaciones

de inferencia. Además de su capacidad de cómputo, la Jetson Nano ofrece conectividad y flexibilidad mediante:

- Salida de video por HDMI o DisplayPort para visualización directa.
- Cuatro puertos USB para conexión de periféricos como cámaras y dispositivos de almacenamiento.
- Puerto microUSB para comunicación y alimentación.
- Conector dedicado para fuente de energía, garantizando estabilidad en aplicaciones de alto consumo.
- Interfaces GPIO, I^2C, SPI y UART para integración con sensores y dispositivos externos.

Estas características permiten integrar el sistema con dispositivos externos y se visualizan en la Figura 3.19.

Figura 3.19: Periféricos Jetson nano.

Para optimizar el rendimiento, se realizaron ajustes en la configuración del entorno, incluyendo la instalación de controladores CUDA, que permiten reducir el tiempo de inferencia mediante optimización del modelo. Asimismo, se implementaron técnicas de reducción de dependencias y funciones personalizadas para garantizar la compatibilidad con la versión de Python disponible en la Jetson Nano.

Capítulo 4

Resultados

En este capítulo se presentarán los resultados obtenidos en el transcurso del trabajo, donde se ha llevado a cabo parte del preprocesamiento de las imágenes, la segmentación de los cromosomas, la clasificación y el ordenamiento.

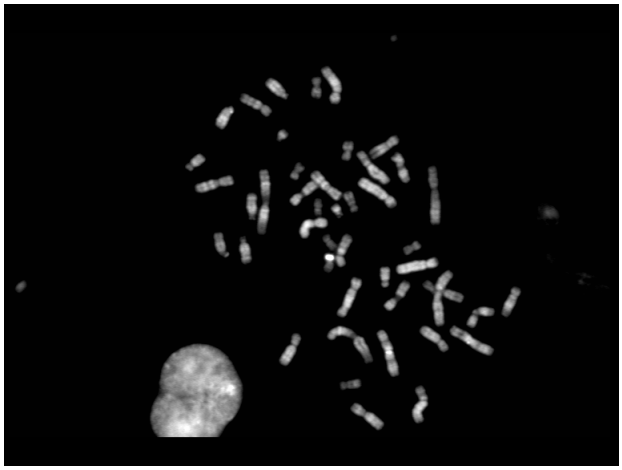
A continuación se explicaran las características de los conjuntos de datos:

- Q-banding: esta base de datos [1], se ha tomado como principal referencia para el desarrollo de este trabajo. Se trata de un conjunto de 108 imágenes en estado metafase con una resolución de 768×576 píxeles, con un aproximado de 6,683 cromosomas. Además, cuenta con su cariograma, pues cada imagen ha sido ordenada y clasificada por un experto. La base de datos también proporciona un "ground truth"que actúa como referencia en el análisis de la segmentación [62], [63].

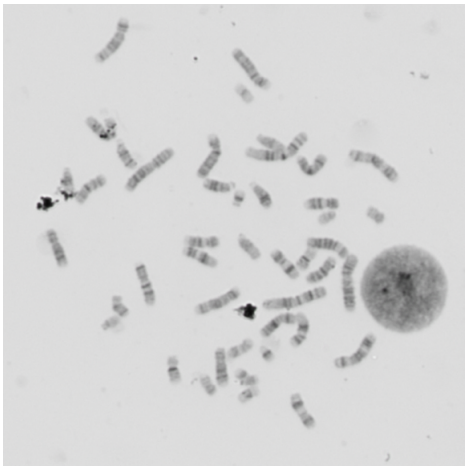
- G-banding: AutoKary2022 es una base de datos que está compuesta por un conjunto de 612 imágenes en estado metafase, cada una con una resolución de 1600 x 1600 píxeles, incluye aproximadamente 27,000 instancias de cromosomas, que han sido extraídas de muestras de 50 pacientes. Las anotaciones que contiene cada imagen han sido elaboradas por expertos en el campo [7].

En la Figura 4.1 se presentan imágenes correspondientes a dos bases de datos diferentes. La Figura (a) muestra una imagen del banco de datos principal utilizado en este trabajo, mientras que la Figura (b) corresponde a otra base de datos alternativa. Es posible observar diferencias entre ambas imágenes, ya que fueron adquiridas mediante métodos distintos: una fue teñida con tinción tipo Q y la otra con tinción tipo G, lo que influye en el contraste, la textura, características y apariencia de los cromosomas.

[1]El conjunto de datos ha sido recuperado del siguiente link:`http://bioimlab.dei.unipd.it`



(a) Imagen banco principal.

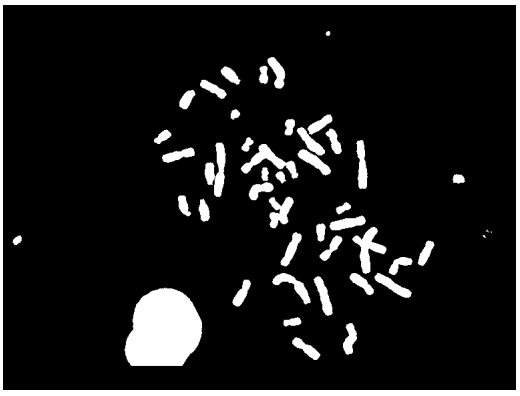


(b) Imagen banco secundario.

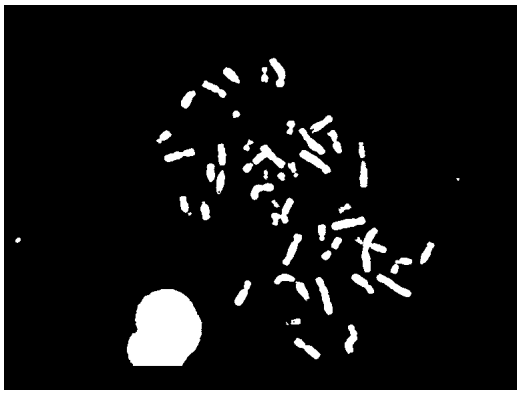
Figura 4.1: Imágenes de los bancos de imágenes.

4.1. Preprocesamiento

En esta sección se presentan los resultados obtenidos durante la etapa de preprocesamiento de las imágenes cromosómicas. Para ello, se evaluaron distintos métodos de binarización con el fin de determinar cuál ofrece una mejor binarización de las imágenes. En la Figura 4.2 se presenta una comparación de la binarización de la misma imagen utilizando dos métodos diferentes: (a) K-menas junto con Campos Aleatorios de Markov (K-CAM) y (b) el método de Otsu. Se puede observar que la binarización realizada con K-CAM produce una silueta más definida en comparación con el método de Otsu, ya que este último muestra que algunos cromosomas han sido separados.



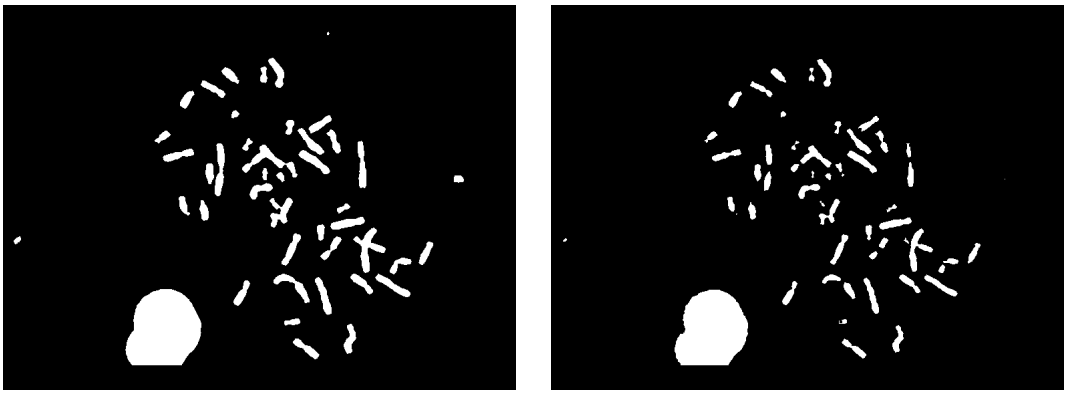
(a) Imagen binarizada CAM.



(b) Imagen binzarizada OTSU.

Figura 4.2: Comparación de binarización entre método OTSU y CAM.

En las Figuras 4.3 y 4.4, se lleva a cabo la operación morfológica de apertura, que consiste en aplicar primero una erosión seguida de una dilatación. Este proceso tiene como objetivo principal el reducir objetos muy pequeños que podrían interferir con la correcta identificación de los cromosomas. La erosión reduce el tamaño de los objetos en la imagen, excluyendo aquellos elementos que no cumplen con un tamaño mínimo, lo que ayuda a reducir el ruido y pequeños objetos no deseados. Posteriormente, la dilatación se aplica para restaurar el tamaño de los cromosomas, rellenando espacios que podrían haberse perdido o dividido durante la etapa de binarización.



(a) Imagen erosionada K-CAM. (b) Imagen erosionada Otsu.

Figura 4.3: Imagen erosionada.

(a) Imagen dilatada K-CAM. (b) Imagen dilatada Otsu.

Figura 4.4: Imagen dilatada.

Una vez completado el proceso de binarización y la reducción de objetos no deseados, se procede a realizar una primera evaluación. Wang [63] llevó a cabo un etiquetado manual de los cromosomas, lo cual proporciona una referencia confiable para comparar los resultados obtenidos mediante los algoritmos implementados.

A continuación se presenta en la Tabla 4.1 que compara los resultados utilizando umbral mediante Otsu y umbral mediante K-CAM, aplicados a las 108 imágenes. Las métricas consideradas incluyen precisión, recall, exactitud y F-score. Ambos métodos alcanzan un valor alto en recall (0.96) y exactitud (0.99), indicando una buena identificación para los cromosomas presentes en las imágenes. Sin embargo, el método K-CAM presenta una ligera mejora en precisión (0.87 frente a 0.85) y F-score (0.91 frente a 0.90). Se puede decir que, aunque ambos métodos son eficaces, el enfoque basado en K-CAM proporciona un desempeño ligeramente superior, siendo una alternativa más robusta para la etapa de binarización en las imágenes.

Método	Precision	Recall	Accuracy	F1
Otsu	0.85	0.96	0.99	0.90
CAM	0.87	0.96	0.99	0.91

Tabla 4.1: Evaluación de las 162 imágenes.

Para generar la Tabla 4.1, se desarrolló una interfaz gráfica que permite cargar imágenes, aplicar el proceso de binarización y almacenar los resultados obtenidos en un archivo. Estos datos fueron posteriormente utilizados para calcular los valores presentados en la tabla. La Figura 4.5 muestra la interfaz diseñada para este propósito. La interfaz desarrollada cuenta con varios elementos interactivos que permiten al usuario controlar el flujo del preprocesamiento. Entre estos se encuentran barras deslizantes que permiten ajustar los umbrales de tamaño mínimo y máximo con el fin de excluir los objetos que no cumplan con el tamaño propuesto en las imágenes. Además, se incluyen botones para cargar imágenes individuales o carpetas completas, aplicar el procesamiento de binarización, y almacenar los resultados.

4.2. Discriminación de cromosomas individuales y agrupados

Una vez realizadas las operaciones morfológicas y la exclusión de objetos de mayor y menor tamaño que los cromosomas, se procede al primer filtrado. Este consiste en evaluar la solidez de cada objeto mediante el uso del menor polígono convexo que puede contenerlo. Para ello, se superponen dos contornos sobre cada objeto detectado: el contorno real del objeto y el contorno del menor polígono convexo que lo envuelve. La diferencia entre ambos contornos permite identificar los huecos en cada cromosoma, lo cual es clave para distinguir entre cromosomas individuales y agrupados. A través de un umbral, se determina si un objeto corresponde a un cromosoma aislado o a un grupo de cromosomas superpuestos. En la Figura 4.6 se presentan ejemplos de este proceso. En dichas imágenes, el color blanco representa el objeto (cromosoma), el negro corresponde al fondo, y el gris indica el área del hueco entre el cromosoma y su respectivo polígono convexo.

Figura 4.5: Interfaz gráfica para el proceso de binarización.

Figura 4.6: Primer filtrado a cromosomas.

El método de la elipse constituye un filtro adicional que utiliza la excentricidad como criterio para discriminar entre cromosomas aislados y agrupados. Este método complementa al algoritmo del polígono convexo, siendo ambos parte de un conjunto de tres filtros para la discriminación de cromosomas. En la Figura 4.7, se ilustra la aplicación del algoritmo basado en el método de la elipse. Este algoritmo evalúa la excentricidad de cada objeto y, mediante un umbral definido en el rango de 0 a 1, determina si se trata de un cromosoma individual o de un grupo de cromosomas. Los objetos clasificados como agrupados se encierran en un círculo, mientras que los cromosomas aislados se representan mediante una elipse. El algoritmo también tiende a clasificar como agrupados aquellos cromosomas que presentan una curvatura pronunciada o que son de menor tamaño, debido a que sus ejes principales tienden a ser similares, lo que reduce la excentricidad. Esta limitación evidencia la necesidad de incorporar un tercer algoritmo que permita una discriminación robusta entre cromosomas individuales y agrupados.

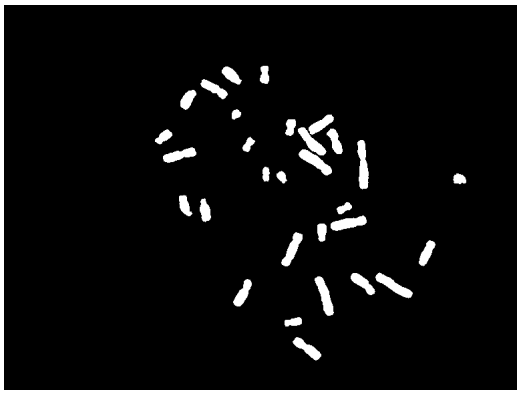
Figura 4.7: Segundo filtrado a cromosomas.

El método de esqueletización, combinado con el conteo de puntos terminales, se utiliza como filtro final para la discriminación de cromosomas. La esqueletización reduce la forma del cromosoma a una línea delgada de un solo píxel de grosor. A su vez, el número de puntas en los extremos del esqueleto permite determinar si un cromosoma se encuentra de manera individual o forma parte de un grupo. Generalmente, los cromosomas individuales presentan dos puntas, mientras que aquellos agrupados o traslapados presentan un número mayor. En la Figura 4.8, se muestra la aplicación de este método. Los cromosomas han sido reducidos a líneas delgadas mediante esqueletización, y los puntos terminales se destacan en color rojo, lo que permite visualizar las diferencias entre estructuras individuales y agrupadas. Esta técnica complementa los métodos anteriores (polígono convexo y elipse), lo que robustece la identificación de cromosomas indviduales.

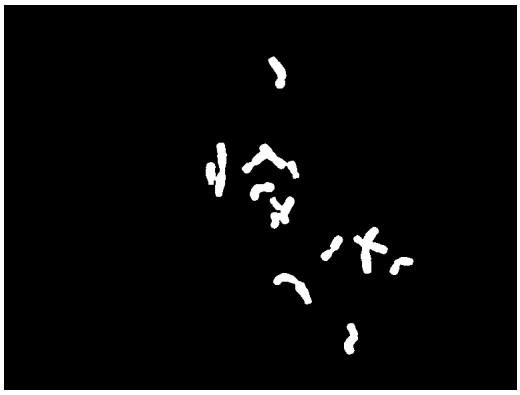
Figura 4.8: Tercer filtrado a cromosomas.

Al aplicar los filtros de discriminación para separar cromosomas individuales de agrupaciones, cada filtro actúa como un paso de refinamiento. En la Figura 4.9 se presentan dos máscaras: en (a) se muestran los cromosomas identificados como individuales por el primer filtro, y en (b) los considerados como grupales. Sin embargo, es posible observar que algunos cromosomas individuales han sido clasificados erróneamente como grupales. Por esta razón, se requiere aplicar un segundo filtro que permita mejorar la precisión de la discriminación.

Una vez aplicado el primer filtro, se procede a utilizar el método basado en el ajuste de elipses, a través del cual se obtiene la excentricidad. Este segundo filtro se aplica sobre la máscara que contiene los cromosomas clasificados como agrupados. Aquellos objetos que correspondan a cromosomas individuales, son reasignados a la máscara de cromosomas individuales. Como se muestra en la Figura 4.10, en (a) se encuentran los cromosomas individuales, mientras que en (b) la cantidad de elementos en la máscara de cromosomas agrupados se ha reducido.

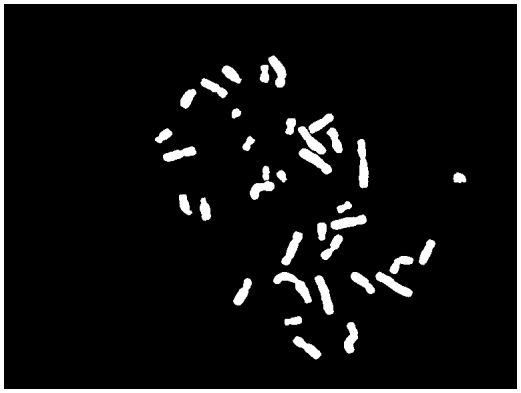


(a) Primer filtro individuales.

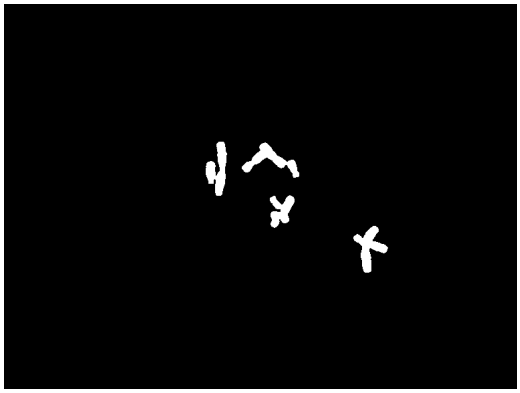


(b) Primer filtro agrupados.

Figura 4.9: Primer filtro aplicado (solidez).

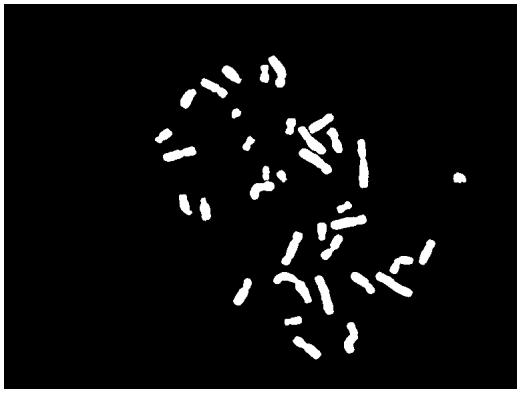


(a) Segundo filtro individuales.

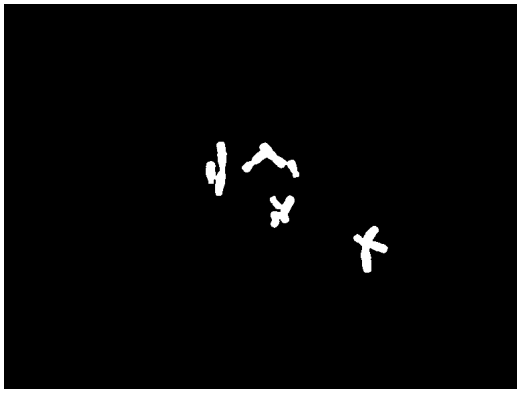


(b) Segundo filtro agrupados.

Figura 4.10: Segundo filtro aplicado (excentricidad).



(a) Tercer filtro individuales.



(b) Tercer filtro agrupados.

Figura 4.11: Tercer filtro aplicado (puntas).

Para reforzar la discriminación entre cromosomas individuales y agrupados, se aplica un tercer filtro, el cual se basa en el número de puntas detectadas en la imagen esqueletizada. Este filtro se aplica a ambas máscaras: si en la máscara de cromosomas individuales se identifican objetos con más de tres puntas, estos se clasifican como cromosomas agrupados. Por el contrario, si en la máscara de cromosomas agrupados se encuentran objetos con dos puntas, se consideran como cromosomas individuales y se trasladan a la máscara de cromosomas individuales. En la Figura 4.11 se muestra la máscara final obtenida tras aplicar los filtros propuestos, donde se distinguen (a) los cromosomas individuales y (b) los cromosomas agrupados. Además, cada conjunto se almacena en sus respectivas carpetas para su posterior procesamiento y análisis.

La Tabla 4.2 presenta los resultados del algoritmo propuesto para la discriminación entre cromosomas individuales y grupales, evaluado sobre un conjunto de 40 imágenes. Las métricas utilizadas para la evaluación fueron: VP (Verdaderos Positivos), que representan los cromosomas individuales correctamente detectados; VN (Verdaderos Negativos), correspondientes a los grupos correctamente identificados; FP (Falsos Positivos), que indican grupos erróneamente clasificados como individuales; y FN (Falsos Negativos), que indican cromosomas individuales erróneamente clasificados como grupo.

Se observa un alto número de VP y VN en la mayoría de los casos, lo que indica una buena capacidad del algoritmo para identificar correctamente tanto los cromosomas individuales como los grupales. Sin embargo, algunas imágenes como la 57, 60 y 64 presentan valores elevados de FP y FN, lo que indica que el algoritmo tiende a confundirse en algunas situaciones. Esta situación es por la forma en que los cromosomas están agrupados: cuando se encuentran muy próximos entre sí y alineados en línea recta, el algoritmo tiende a interpretarlos como un solo objeto.

Para evaluar el rendimiento general del algoritmo de discriminación de cromosomas individuales y grupales, se calcularon métricas como precisión, exactituda, recall y F1 score a partir de los valores acumulados de VP, VN, FP y FN. La precisión obtenida fue de 0.9540, lo que indica que el 95.4 % de las predicciones positivas fueron correctas. La exactitud, que representa el porcentaje total de clasificaciones correctas (tanto positivas como negativas), fue de 0.9473, reflejando el desempeño general del algoritmo. La recall alcanzó un valor de 0.9853, lo que muestra que el algoritmo es capaza de detectar correctamente los cromosomas individuales. Finalmente, el F1 score, que combina precisión y recall, fue de 0.9694, mostrando que el modelo tiene un equilibrio entre la detección correcta y lo mínimos errores presentados.

Nom	Ind	Grupo	VP	VN	FP	FN	Precisión	Exactitud	Recall	F1-Score
1	38	4	38	4	0	0	100 %	100 %	100 %	100 %
9a	18	6	16	6	2	0	89 %	92 %	100 %	90 %
13	26	7	24	5	2	2	92 %	88 %	92 %	90 %
17	31	3	31	3	0	0	100 %	100 %	100 %	100 %
18	34	4	34	4	0	0	100 %	100 %	100 %	100 %
21	30	6	28	6	2	0	93 %	94 %	100 %	94 %
25	39	3	39	1	0	2	100 %	95 %	95 %	98 %
28	43	1	42	1	1	0	98 %	98 %	100 %	98 %
31	32	5	31	4	1	1	97 %	95 %	97 %	96 %
34	33	4	31	3	2	1	94 %	92 %	97 %	93 %
35	33	2	31	2	2	0	94 %	94 %	100 %	94 %
36a	39	3	38	2	1	1	97 %	95 %	97 %	96 %
41	38	5	38	3	0	2	100 %	95 %	95 %	98 %
42	30	6	29	6	1	0	97 %	97 %	100 %	97 %
44	35	2	34	2	1	0	97 %	97 %	100 %	97 %
46	36	4	35	4	1	0	97 %	98 %	100 %	97 %
49	35	2	34	2	1	0	97 %	97 %	100 %	97 %
50	30	4	27	4	3	0	90 %	91 %	100 %	91 %
51	34	3	31	3	3	0	91 %	92 %	100 %	92 %
52	32	6	30	6	2	0	94 %	95 %	100 %	94 %
56	23	7	23	7	0	0	100 %	100 %	100 %	100 %
57	26	4	22	4	4	0	85 %	87 %	100 %	86 %
58	25	9	25	7	0	2	100 %	94 %	93 %	97 %
59	24	7	21	7	3	0	88 %	90 %	100 %	89 %
60	37	4	34	2	3	2	92 %	88 %	94 %	90 %
61	28	7	26	6	2	1	93 %	91 %	96 %	92 %
64	16	7	12	6	4	1	75 %	78 %	92 %	77 %
65	19	4	19	4	0	0	100 %	100 %	100 %	100 %
69	34	4	33	4	1	0	97 %	97 %	100 %	97 %
74	35	3	32	3	3	0	91 %	92 %	100 %	92 %
77	44	2	44	2	0	0	100 %	100 %	100 %	100 %
83	37	4	36	3	1	1	97 %	95 %	97 %	96 %
88	37	4	36	3	1	1	98 %	96 %	98 %	97 %
89	32	3	30	3	2	0	94 %	94 %	100 %	94 %
93	29	5	28	5	1	0	97 %	97 %	100 %	97 %
94	27	6	24	6	3	0	89 %	91 %	100 %	90 %
95	33	4	33	4	0	0	100 %	100 %	100 %	100 %
96	26	6	23	6	3	0	88 %	91 %	100 %	90 %
100	34	4	33	4	1	0	97 %	97 %	100 %	97 %
103	30	6	29	5	1	1	97 %	94 %	97 %	96 %

Tabla 4.2: Desempeño del método de discriminación de cromosomas individuales y grupales.

4.3. Clasificación (Segunda etapa)

La etapa de clasificación comprende cuatro componentes: pre-procesamiento de los datos, entrenamiento y validación del modelo, el desarrollo de un algoritmo para el ordenamiento cromosómico, y la implementación de una interfaz de usuario (GUI) que permita la generación del cariograma de forma interactiva.

4.3.1. Preprocesamiento de los datos

Antes de iniciar el entrenamiento de la CNN, es recomendable realizar un preprocesamiento a los datos con el fin de estandarizarlos y garantizar la calidad del aprendizaje. Para ello, se llevaron a cabo las siguientes etapas:

- **Conversión de formato:** Las imágenes originales en formato `.bmp` fueron convertidas a `.png` para reducir el tamaño y mejorar la compatibilidad.
- **Extracción de objetos (cromosomas):** Se segmentaron los cromosomas con las técnicas de Otsu y K-CAM, eliminando el fondo, aislando cada objeto y realizar una comparación de ambas técnicas.
- **Centrado de los objetos:** Cada cromosoma fue reposicionado en el centro de un nuevo fondo.
- **Determinación del tamaño máximo:** Se calculó el ancho y la altura del cromosoma más grande dentro del conjunto de datos.
- **Aplicación de padding:** A partir del tamaño máximo identificado, se aplicó relleno (padding) a todos los cromosomas para unificar el tamaño de los datos.

En la Tabla 4.3 se presenta el porcentaje de exactitud obtenido en la extracción de cada cromosoma, con el objetivo de evaluar la efectividad de los métodos de segmentación utilizados. Este proceso se realizó para automatizar la extracción y el centrado de los cromosomas sobre un nuevo fondo. Los resultados muestran que el método K-CAM superó a Otsu, ya que en la mayoría de los casos no fue necesaria la intervención manual para corregir errores de segmentación.

Método	Exactitud
Otsu	0.75
CAM	0.99

Tabla 4.3: Evaluación de la segmentación de los datos.

En la Figura 3.15 se muestra el resultado final del preprocesamiento, donde cada imagen presenta dimensiones normalizadas de 120×160 píxeles y los cromosomas están

centrados en su respectivo recuadro. Con las imágenes ya normalizadas, se organiza en carpetas por clase, correspondientes a los pares cromosómicos del 1 al 22, y dos clases adicionales para los cromosomas sexuales (X y Y). Finalmente, se realiza la división del conjunto de datos en dos subconjuntos: 70 % para entrenamiento y 30 % para validación.

Figura 4.12: Cromosomas después del acondicionamiento.

4.3.2. Entrenamiento y validación

Para evaluar el desempeño del sistema propuesto, se realizó una comparación con estudios recientes que emplean redes neuronales convolucionales para la clasificación de cromosomas. La Tabla 4.4 muestra el tamaño del conjunto de datos utilizado por cada autor y la exactitud alcanzada.

Autor	Dataset	Accuracy	Equipo
Xie et al [6]	660000	95.7 %	NVIDIA GTX1080ti
Wang et al [33]	90624	94.72 %	NVIDIA 2080ti
Wei et al [32]	13800	99.2 %	No proporciona
You et al [7]	60000	93 %	NVIDIA RTX3090

Tabla 4.4: Comparación del tamaño del dataset y exactitud en trabajos previos.

El proceso de entrenamiento y validación del modelo se llevó a cabo utilizando diferentes arquitecturas propuestas. Tras una serie de pruebas comparativas, el modelo que presentó el mejor desempeño es el que se describe en la sección de desarrollo.

Modelo	**Entrenamiento**	**Validación**
EfficientNetB0	4.35 %	4.35 %
ResNet50	5.29 %	5.16 %
VGG16	23.00 %	20.00 %
MobileNetV2	61.83 %	44.66 %
InceptionV3	66.84 %	50.36 %
CNN propuesto	98.92 %	81.25 %

Tabla 4.5: Comparación del desempeño entre modelos preentrenados y la CNN propuesta.

Como se muestra en la Tabla 4.5, el modelo de red neuronal convolucional propuesto supera a los modelos preentrenados evaluados. Mientras que EfficientNetB0 y ResNet50 presentan desempeños muy bajos, con apenas un 4.35 % y 5.16 % de exactitud en validación respectivamente, modelos más complejos como InceptionV3 alcanzan un rendimiento moderado (66.84 % en entrenamiento y 50.36 % en validación). Sin embargo, la CNN propuesta logra una exactitud del 98.92 % en entrenamiento y del 81.25 % en validación, lo que muestra una mejor capacidad de aprendizaje y generalización. Esta diferencia sugiere que los modelos preentrenados, diseñados para tareas de clasificación genérica, no capturan adecuadamente las características del conjunto de datos utilizado.

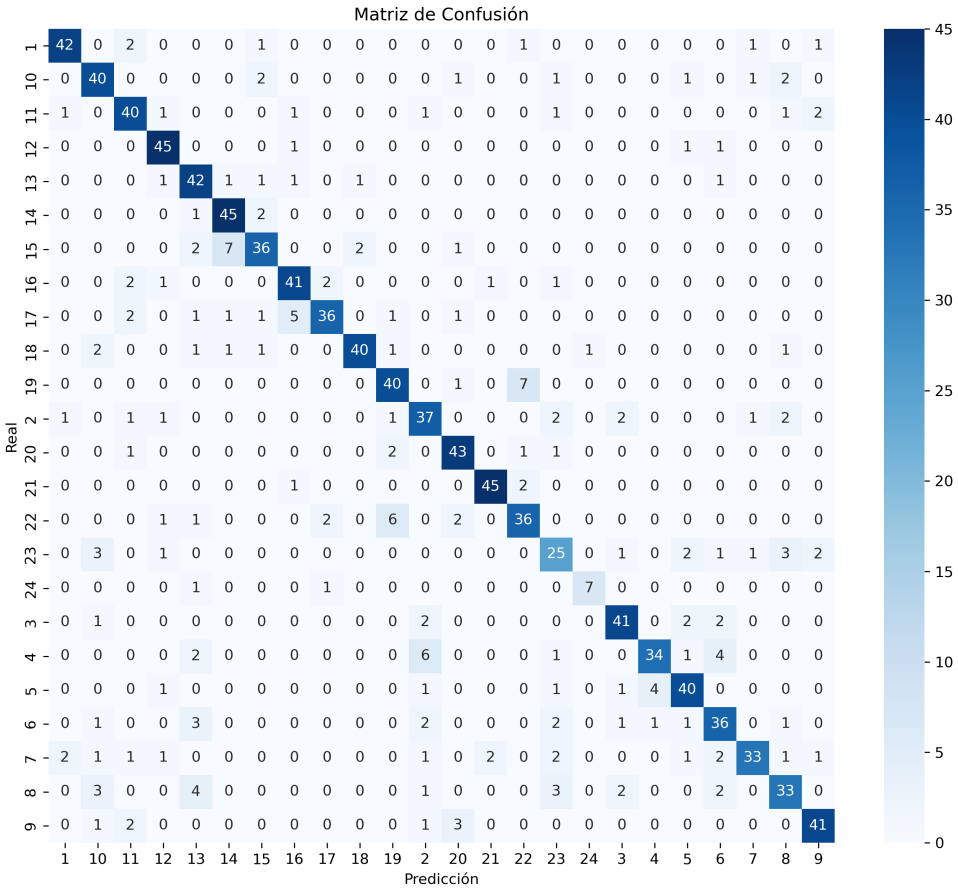


Figura 4.13: Matriz de confusión de la CNN propuesta.

La matriz de confusión presentada en la Figura 4.13 permite evaluar el rendimiento del modelo CNN propuesto en la clasificación de 24 clases correspondientes a los cromo-

somas. Cada fila representa la clase real y cada columna la clase predicha. Los valores en la diagonal principal indican predicciones correctas, mientras que los valores fuera de la diagonal corresponden a errores de clasificación. La mayoría de las clases presentan valores elevados en la diagonal, lo que indica que el modelo logra identificar correctamente la mayoría de las clases. Las clases 1, 3, 9, 12, 13, 14, 20 y 21 muestran una alta precisión, con más de 40 aciertos y muy pocos errores, lo que sugiere que estas clases tienen características diferenciadas. Algunas clases presentan errores altos como: 2, 4, 6, 7, 8, 15, 17 y 22. Esto indica que estas clases comparten características que dificultan la clasificación.

Clase	**Precisión**	**Recall**	**F1-Score**
1	91 %	87 %	89 %
2	71 %	77 %	74 %
3	85 %	85 %	85 %
4	87 %	70 %	78 %
5	81 %	83 %	82 %
6	73 %	75 %	74 %
7	89 %	68 %	77 %
8	75 %	68 %	71 %
9	87 %	85 %	86 %
10	76 %	83 %	80 %
11	78 %	83 %	80 %
12	84 %	93 %	89 %
13	72 %	87 %	79 %
14	81 %	93 %	87 %
15	81 %	75 %	78 %
16	82 %	85 %	83 %
17	87 %	75 %	80 %
18	93 %	83 %	87 %
19	78 %	83 %	80 %
20	82 %	89 %	86 %
21	93 %	93 %	93 %
22	76 %	75 %	75 %
23	62 %	64 %	63 %
24	87 %	77 %	82 %
Promedio Macro	81.8 %	81.1 %	81.2 %
Promedio Micro	81.3 %	81.3 %	81.3 %

Tabla 4.6: Métricas de evaluación por clase y promedios globales del modelo CNN propuesto.

La Tabla 4.6 presenta las métricas de evaluación por clase para el modelo CNN propuesto, la cual incluye precisión, recall y F1 score. Estas métricas permiten analizar el rendimiento del modelo, identificando las clases con un mayor desempeño y aquellas con dificultad para clasificación. Por ejemplo, la clase 21 muestra un F1-score del 93 %, indicando una alta capacidad de predicción, mientras que la clase 23 presenta el valor más bajo (63 %), lo que causa confusión al modelo con clases de características muy similares. También se incluyen los promedios macro y micro para evaluar el rendimiento general del modelo. El promedio macro (Precisión: 81.8 %, Recall: 81.1 %, F1-score: 81.2 %) calcula el rendimiento medio considerando todas las clases con el mismo peso, lo que es una evaluación equilibrada del modelo con clases desbalanceadas. Por otro lado, el promedio micro (81.3 % en las tres métricas) pondera el cálculo según el número total de muestras, reflejando el desempeño general del modelo en el conjunto de datos. La similitud entre ambos promedios indica que el modelo mantiene un comportamiento consistente.

4.3.3. Ordenamiento

Una vez entrenada y validada la CNN propuesta, se procede a utilizar el modelo para el ordenamiento automático de los cromosomas. Para ello, se emplea el entrenamiento previamente guardado, integrándolo en un algoritmo que identifica cada cromosoma y lo organiza en pares homólogos, conformando el cariograma. Con el fin de facilitar la interacción, se desarrolló una interfaz gráfica (GUI) que permite al usuario realizar el proceso de manera intuitiva (Figura 4.14). Esta herramienta ofrece funcionalidades clave, como la exclusión de objetos que no corresponden a cromosomas (por ejemplo, fragmentos demasiado grandes o pequeños), la discriminación precisa de cromosomas y su posterior identificación y ordenamiento.

El algoritmo de ordenamiento tiene como objetivo organizar los cromosomas previamente discriminados. Para ello, utiliza aquellos cromosomas que han sido detectados como individuales. Posteriormente, el algoritmo carga la red neuronal convolucional entrenada para realizar la identificación de cada cromosoma y, finalmente, los ubica en su posición correspondiente. En la figura 4.15 se muestra el resultado final, donde se visualizan los cromosomas identificados y ordenados en su respectivo lugar.

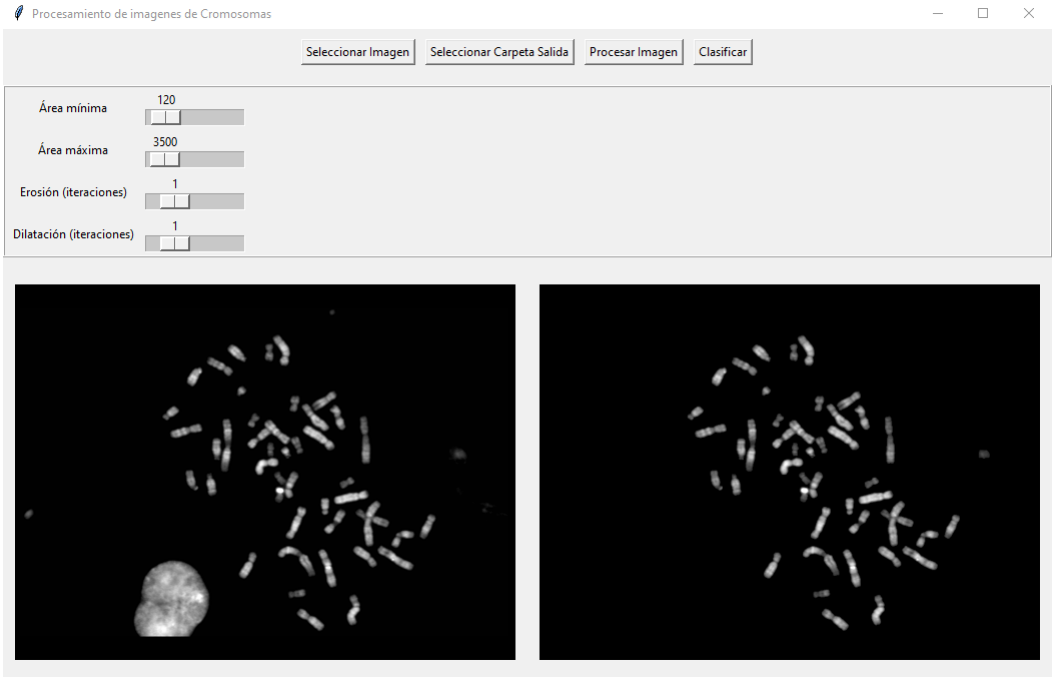


Figura 4.14: Interfaz de usuario para el ordenamiento de los cromosomas.

4.3.4. Implementación

Durante la implementación del algoritmo en la tarjeta de desarrollo Jetson Nano, se presentaron algunas dificultades respecto a la versión de Python en el dispositivo (3.6.9), por otro lado con la versión utilizada durante el desarrollo en un equipo de cómputo (3.12). Esta diferencia generó incompatibilidades con ciertas bibliotecas, lo que obligó a realizar ajustes en el código. Para resolver este inconveniente, se incorporaron funciones compatibles con ambas versiones de Python, minimizando el uso de bibliotecas externas. Entre estas funciones se encuentran la asignación de los puntos terminales de los cromosomas y el proceso de adelgazamiento de los mismos, los cuales fueron implementados de forma manual para asegurar la portabilidad del algoritmo. En la Figura 4.16a se muestra el esquema de conexión y encendido de la Jetson Nano. Una de las principales ventajas de trabajar con esta tarjeta de desarrollo es su sistema operativo basado en Linux, el soporte nativo para Python y la integración de una GPU. En la Figura 4.16b se presenta el algoritmo ejecutándose dentro de la Jetson Nano, sin presentar inconvenientes. Esto demuestra que el algoritmo ha sido optimizado y adaptado para funcionar en diferentes entornos y versiones de Python.

En la Tabla 4.7 se presentan los tiempos promedio obtenidos durante la ejecución del algoritmo, mostrando el tiempo para la segmentación de la imagen y el tiempo a la clasificación. El dispositivo que presentó mayor tiempo de procesamiento fue la Jetson Nano; sin embargo, el algoritmo logra completar ambos procesos en menos de cinco minutos. Este resultado representa una mejora en comparación con el método tradicional,

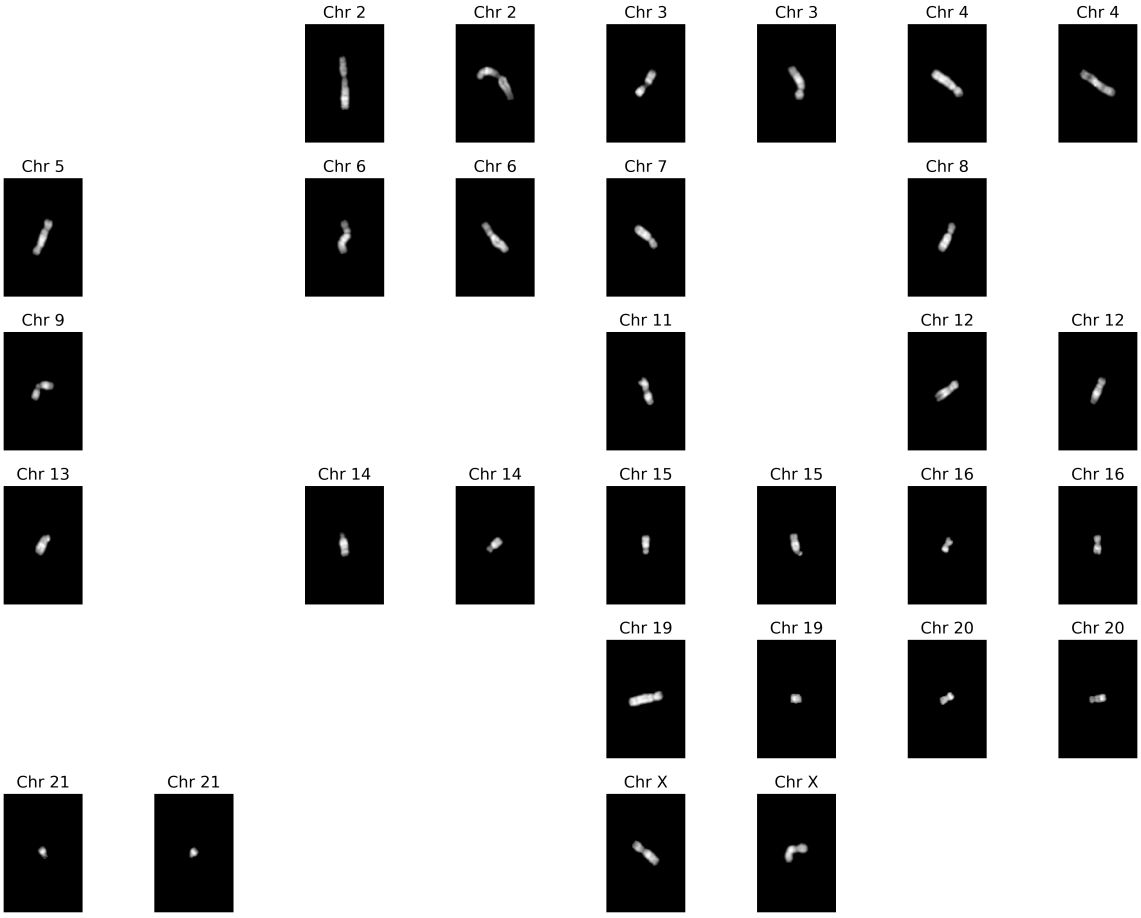


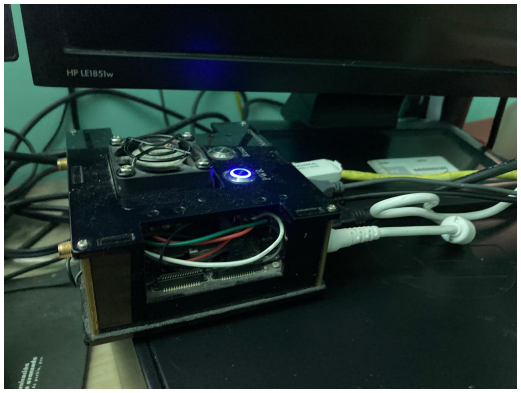
Figura 4.15: Cariograma.

donde la obtención del cariograma suele tardar varias horas. El tiempo de procesamiento se reduce, permitiendo un análisis cromosómico en menos tiempo.

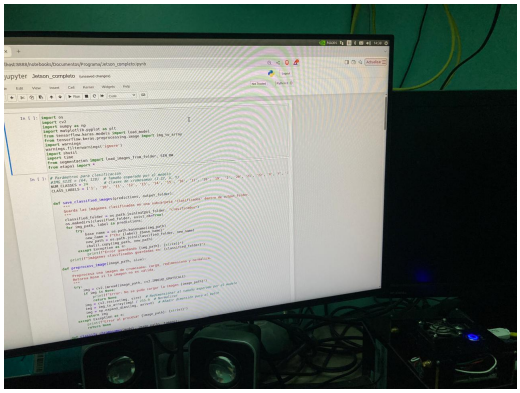
Equipo	**Tiempo segmentación**	**Tiempo clasificación**
Laptop core i7	0.26s	5.07s
CPU core i3	0.50s	10.41s
Jetson nano	0.86s	29.66s

Tabla 4.7: Comparación del tiempo promedio en diferentes equipos.

Aunque el tiempo de procesamiento en la Jetson Nano es el más alto respecto a otros equipos, su implementación fortalece el aporte del sistema por varias razones. Esta tarjeta de desarrollo ofrece portabilidad y permite la operación en entornos con recursos limitados, ejecutando el algoritmo sin depender de estaciones de trabajo de alto rendimiento ni de servicios en la nube. Además, su bajo consumo energético la convierte



(a) Conexión inicial de la Jetson Nano.



(b) Ejecución del algoritmo en la Jetson Nano.

Figura 4.16: Jetson Nano.

en una opción viable para laboratorios académicos y clínicos. Otro aspecto relevante es la disponibilidad de interfaces de comunicación (GPIO, I^2C, SPI, UART), que facilitan la integración con dispositivos externos, como microscopios o sistemas de adquisición, abriendo la posibilidad de evolucionar hacia un sistema completo de análisis automatizado. Por estas razones, el uso de la Jetson Nano aporta escalabilidad y accesibilidad.

Como se muestra en la Tabla 4.8, el sistema propuesto ofrece una sistema automático (segmentación, clasificación y ordenamiento), con tiempos de análisis menores a cinco minutos y bajo costo de implementación, lo que lo distingue de herramientas comerciales (semi-automatización con revisión manual y licencias elevadas) y de métodos recientes que, aunque alcanzan mayores exactitudes sobre grandes bases de datos y GPU dedicadas, no abordan necesariamente el ordenamiento automático ni la portabilidad en hardware embebido.

Característica	**Sistema propuesto**	**Software comercial (Ej: Ikaros, Metafer)**	**Métodos recientes del estado del arte**
Automatización	segmentación y clasificación	segmentación y clasificación	segmentación y clasificación
Tiempo de análisis (por muestra)	$\approx$ **4 min 30 s** (Jetson Nano).	$\approx$ **15-20 min** (modo automatizado con verificación)	$\approx$ **2-8 min** para inferencia en GPU (RTX 3080, entre otros).
Precisión (Clasificación)	**Exactitud: 0.82** Validación en dataset interno de 5000 imágenes.	**Variable** 90-98 %. La precisión operativa depende mucho de la calibración y experiencia del usuario.	**Variable** $>$ 95 % en datasets $>$ 13000.
Costo	**Bajo**. Software de licencia abierta (Python)	**Muy alto**. Licencias anuales: \$10,000-\$50,000 USD, más el costo del servidor/hardware de estación de trabajo.	**Bajo**. Software de licencia abierta (Python)
Accesibilidad	**Alta**. Portátil y diseñada para entornos con recursos limitados. Cuneta con su propio GUI.	**Moderada a Baja**. Limitada a laboratorios o clínicas con presupuesto para equipamiento especializado y personal capacitado, además de su propio GUI.	**Baja**. Requiere conocimiento experto en aprendizaje profundo para adaptar, entrenar, desplegar y mantener los modelos.

Tabla 4.8: Comparación del sistema propuesto con software comercial y métodos del estado del arte para el análisis de imágenes de laboratorio.

Capítulo 5

Conclusiones

Se implementó un proceso de segmentación para la reducción de ruido y de objetos no deseados, aplicando criterios de exclusión basados en el área de los objetos. Para evaluar la calidad del preprocesamiento, se utilizaron métricas específicas que permitieron comparar el desempeño de diferentes métodos.

Se han comparado dos métodos de binarización K-CAM y el método de Otsu; el primero captura la silueta de los cromosomas. En contraste, el segundo tiende a fragmentar algunos cromosomas. Presentando una evaluación ligeramente superior en la precisión.

Se ha encontrado que los algoritmos de polígono convexo y elipse presenta tanto ventajas como limitaciones en la identificación de cromosomas. Ambos algoritmos son efectivos para detectar cromosomas individuales y rectos, así como aquellos que están agrupados. Sus limitaciones se observan cuando se trata de cromosomas más pequeños o de formas curvas, ya que estos pueden confundirse con los cromosomas agrupados. Debido a esto, se implementó un algoritmo adicional que funciona como un filtro complementario, mejorando la discriminación entre cromosomas agrupados y aquellos que están aislados.

A pesar de incorporar un tercer filtro diseñado para robustecer la discriminación entre cromosomas, este método aún presenta ciertas limitaciones. El filtro detecta las puntas de los cromosomas después de aplicar un proceso de esqueletización, determinando el número de extremos. Sin embargo, cuando los cromosomas se encuentran muy próximos entre sí, formando estructuras en cadena, el algoritmo tiende a cometer errores, ya que estas configuraciones no presentan ramificaciones y su esqueleto se reduce a una línea continua.

El método de discriminación de cromosomas muestra un desempeño sólido como se muestra en la Tabla 4.3, con métricas de precisión, exactitud, recall y F1-score superiores al 90 % en la mayoría de los casos, lo que indica una fiabilidad en la discriminación. Se muestra que el recall se mantiene cercano al 100 %, lo que significa que el sistema rara vez omite cromosomas individuales, aunque en algunas imágenes con mayor complejidad (por ejemplo, cromosomas adyacentes en una sola línea) la precisión disminuye ligeramente, evidenciando falsos positivos.

Aunque el diseño de la red neuronal alcanza una exactitud del 98.92 % en el conjunto

de entrenamiento y del 81.25 % en validación, se siguen observando errores de clasificación. Esta diferencia implica la presencia de sobreajuste lo que indica que el modelo no generaliza adecuadamente ante datos nuevos. Para mitigar este problema, se recomienda incrementar el tamaño de la base de datos.

A diferencia de los sistemas existentes, que utilizan volúmenes de datos ($\approx$ 3,707 imágenes por clase) y equipos de alto rendimiento, el sistema propuesto logra resultados competitivos utilizando un conjunto reducido ($\approx$ 190 imágenes por clase) y ejecutándose en una plataforma embebida (Jetson Nano). Esta diferencia muestra la eficiencia del modelo y su aplicabilidad en entornos con recursos limitados frente a herramientas comerciales que implican costos elevados.

La implementación del algoritmo en la Jetson Nano presentó desafíos relacionados con la compatibilidad de las bibliotecas disponibles. Para superar estas limitaciones, se desarrollaron funciones personalizadas que redujeron la dependencia de estas, optimizando el consumo de recursos y ejecutando el modelo en un entorno con las restricciones del hardware.

El tiempo que tardó cada dispositivo (Tabla 4.7) para realizar la segmentación y clasificación de los cromosomas está relacionado con las características de hardware de cada uno. A pesar de estas diferencias, el tiempo total para completar ambas tareas no supera los 5 minutos, lo que representa la optimización en el proceso de armado del cariograma. Esta reducción en el tiempo contribuye a la viabilidad del método en entornos con recursos limitados.

La incorporación de una interfaz gráfica no solo facilita el uso del algoritmo de manera intuitiva, sino que también contribuye a su accesibilidad en diferentes equipos de cómputo. Además, esta interfaz sirvió como herramienta de apoyo para la evaluación de las métricas de los algoritmos y la medición del desempeño general del sistema.

Aportaciones del trabajo

Las contribuciones de esta tesis son las siguientes:

- **Automatización del proceso de armado del cariograma:** Se desarrolló un sistema que realiza la segmentación, discriminación y un clasificador cromosómico de manera automática, reduciendo el tiempo requerido frente al método manual.
- **Optimización en entornos con recursos limitados:** A diferencia de sistemas existentes que requieren grandes volúmenes de datos ($\approx 3,707$ imágenes por clase) y hardware de alto rendimiento, el sistema propuesto logra resultados competitivos utilizando un conjunto reducido ($\approx$ 190 imágenes por clase) y ejecutándose en una tarjeta de desarrollo de propósito específico (Jetson Nano).
- **Desarrollo de algoritmos complementarios:** Se implementaron filtros adicionales para mejorar la discriminación entre cromosomas individuales y agrupados, alcanzando métricas superiores al 90 % en precisión, exactitud, *recall* y F1-score.

- **Interfaz gráfica intuitiva:** Se incorporó una interfaz que facilita la interacción con el sistema, permitiendo su uso en entornos educativos y clínicos sin requerir conocimientos avanzados en programación.

Trabajo a futuro

Como trabajos a futuros se proponen lo siguiente:

- Integración con dispositivos del laboratorio: Aprovechar los pines de comunicación de la Jetson Nano para establecer interacción directa con otros equipos del laboratorio, permitiendo la identificación y clasificación de cromosomas en tiempo real.
- Mejoras en la interfaz gráfica: Incorporar funciones en la interfaz gráfica para la visualización de los resultados. Así como la integración de funciones adicionales que se empleen en el entorno clínico.
- Base de datos: Se propone la creación de una base de datos robusta que contenga al menos 1,000 imágenes por clase de cromosomas obtenidas en entornos clínicos. Esta recomendación se basa en dos aspectos:
  - Mejorar la capacidad de generalización del modelo: El conjunto actual ($\approx$190 imágenes por clase) presenta limitaciones que se reflejan en la diferencia entre la precisión en entrenamiento (98.92 %) y validación (81.25 %), lo que indica sobreajuste.
  - Evidencia en trabajos previos: Estudios reportan que modelos entrenados con grandes volúmenes de datos ($\approx$3,700 imágenes por clase en promedio) alcanzan exactitudes cercanas al 99 %.
- Características diferenciables en el entrenamiento del CNN: Se plantea el uso de resaltar características de cada cromosoma para aumentar el porcentaje de aprendizaje de la red neuronal.
- Mejoras en la CNN: Se busca incrementar la exactitud y la precisión del CNN para optimizar el ordenamiento automático de los cromosomas y garantizar una mayor fiabilidad en la construcción del cariograma.

Bibliografía

[1] Lisa Shaffer y Theisen. "Disorders caused by chromosome abnormalities". En: *The Application of Clinical Genetics* (dic. de 2010), pág. 159. DOI: 10.2147/tacg.s8884.

[2] Delia Aiassa et al. *CITOGENÉTICA Teoría y Práctica*. Centro de estudios de Pablació y Desarrollo, 2015. ISBN: 9789872950231.

[3] Xiaoyu Chen et al. "ChroSegNet: An Attention-Based Model for Chromosome Segmentation with Enhanced Processing". En: *Applied Sciences* 13 (4 feb. de 2023), pág. 2308. ISSN: 2076-3417. DOI: 10.3390/app13042308.

[4] Eisuke Gotoh. "A Basic and Simple Chromosome Preparation Protocol". En: 2023, págs. 1-7. DOI: 10.1007/978-1-0716-2433-3_1.

[5] Elodia Torres. "Ventajas y limitaciones de la citogenética en la medicina actual". En: *Memorias del Instituto de Investigaciones en Ciencias de la Salud* 16 (2 ago. de 2018), págs. 107-112. ISSN: 18174620. DOI: 10.18004/mem.iics/1812-9528/2018.016(02)107-112.

[6] Ning Xie et al. "Statistical Karyotype Analysis Using CNN and Geometric Optimization". En: *IEEE Access* 7 (2019), págs. 179445-179453. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2951723.

[7] Dan You et al. "AutoKary2022: A Large-Scale Densely Annotated Dataset for Chromosome Instance Segmentation". En: *Proceedings - IEEE International Conference on Multimedia and Expo* 2023-July (2023), págs. 1577-1582. ISSN: 1945788X. DOI: 10.1109/ICME55011.2023.00272.

[8] Chuan Yang et al. "Chromosome classification via deep learning and its application to patients with structural abnormalities of chromosomes". En: *Medical Engineering and Physics* 121 (nov. de 2023). ISSN: 18734030. DOI: 10.1016/j.medengphy.2023.104064.

[9] Gady Agam, Student Member y hak Dinstein. "Geometric Separation of Partially Overlapping Nonrigid Objects Applied to Automatic Chromosome Classification". En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (11 nov. de 1997), págs. 1212-222. ISSN: 0162-8828. DOI: 10.1109/34.632981.

[10] Petros Karvelis et al. “Segmentation of chromosome images based on a recursive watershed transform”. En: *The 3rd European Medical and Biological Engineering Conference*. Vol. 18. Nov. de 2005.

[11] Wacharapong Srisang, Krisanadej Jaroensutasinee y Mullica Jaroensutasinee. “Segmentation of Overlapping Chromosome Images Using Computational Geometry”. En: *Walailak J Sci & Tech* 3 (2 2006), págs. 181-194.

[12] Enrico Grisan et al. “Automatic segmentation of chromosomes in Q-band images”. En: *IEEE*. IEEE, ago. de 2007. ISBN: 978-1-4244-0787-3. DOI: 10.1109/IEMBS.2007.4353594.

[13] Petros Karvelis, Aristidis Likas y Dimitrios I. Fotiadis. “Identifying touching and overlapping chromosomes using the watershed transform and gradient paths”. En: *Pattern Recognition Letters* 31 (16 dic. de 2010), págs. 2474-2488. ISSN: 01678655. DOI: 10.1016/j.patrec.2010.08.002.

[14] Mousami V Munot et al. “Automated Karyotyping of Metaphase Cells with Touching Chromosomes”. En: *International Journal of Computer Applications* 29 (12 2011), págs. 975-8887. DOI: 10.5120/3700-5175.

[15] Mukul A. Joshi et al. “Automated Detection of the Cut-points for the Separation of Overlapping Chromosomes”. En: *2012 IEEE-EMBS Conference on Biomedical Engineering and Sciences*. IEEE, dic. de 2012. ISBN: 9781467316668. DOI: 10.1109/IECBES.2012.6498193.

[16] Mousami V. Munot et al. “Automated Detection of Cut-Points for Disentangling Overlapping Chromosomes”. En: *Point-of-Care Healthcare Technologies (PHT)*. IEEE, ene. de 2013. ISBN: 9781467327671. DOI: 10.1109/PHT.2013.6461299.

[17] D. Somasundaram y V. R. Vijay Kumar. “Separation of overlapped chromosomes and pairing of similar chromosomes for karyotyping analysis”. En: *Measurement: Journal of the International Measurement Confederation* 48 (1 feb. de 2014), págs. 274-281. ISSN: 02632241. DOI: 10.1016/j.measurement.2013.11.024.

[18] Boaz Lerner. “Toward A Completely Automatic Neural-Network-Based Human Chromosome Analysis”. En: *CYBERNETICS* 28 (4 ago. de 1998). DOI: 10.1109/3477.704293.

[19] John M Conroy et al. “Chromosome Identification Using Hidden Markov Models: Comparison with Neural Networks, Singular Value Decomposition, Principal Components Analysis, and Fisher Discriminant Analysis”. En: *Laboratory Investigation* 0 (11 nov. de 2000), págs. 1629-1641. DOI: 10.1038/labinvest.3780173.

[20] Zhenzhen Kou, Liang Ji y Xuegong Zhang. “Karyotyping of comparative genomic hybridization human metaphases by using support vector machines”. En: *Cytometry* 47 (1 ene. de 2002), págs. 17-23. ISSN: 01964763. DOI: 10.1002/cyto.10027.

[21] Petros S. Karvelis et al. “A Watershed Based Segmentation Method for Multispectral Chromosome Images Classification”. En: *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, ago. de 2006. ISBN: 1-4244-0032-5. DOI: 10.1109/IEMBS.2006.260682.

[22] Xingwei Wang et al. “A rule-based computer scheme for centromere identification and polarity assignment of metaphase chromosomes”. En: *Computer Methods and Programs in Biomedicine* 89 (1 ene. de 2008), págs. 33-42. ISSN: 01692607. DOI: 10.1016/j.cmpb.2007.10.013.

[23] Petros S. Karvelis y Dimitrios I. Fotiadis. “A region based decorrelation stretching method: Application to multispectral chromosome image classification”. En: *2008 15th IEEE International Conference on Image Processing*. IEEE, sep. de 2008, págs. 1456-1459. ISBN: 9781424417643. DOI: 10.1109/ICIP.2008.4712040.

[24] Yaser Rahimi, Rassoul Amirfattahi y Reza Ghaderi. “Design of a neural network classifier for separation of images with one chromosome from images with several chromosomes”. En: *Proceedings - 3rd International Conference on Broadband Communications, Informatics and Biomedical Applications, BroadCom 2008*. 2008, págs. 186-190. ISBN: 9780769534534. DOI: 10.1109/BROADCOM.2008.9.

[25] Benoît Legrand et al. “Chromosome classification using dynamic time warping”. En: *Pattern Recognition Letters* 29 (3 feb. de 2008), págs. 215-222. ISSN: 01678655. DOI: 10.1016/j.patrec.2007.09.017.

[26] Sunthorn Rungruangbaiyok y Pornchai Phukpattaranont. “Chromosome image classification using a two-step probabilistic neural network”. En: *Songklanakarin J. Sci. Technol* 32 (3 mar. de 2010), págs. 255-262.

[27] Enea Poletti, Enrico Grisan y Alfredo Ruggeri. “A modular framework for the automatic classification of chromosomes in Q-band images”. En: *Computer Methods and Programs in Biomedicine* 105 (2 feb. de 2012), págs. 120-130. ISSN: 01692607. DOI: 10.1016/j.cmpb.2011.07.013.

[28] Devaraj Somasundaram et al. “A Novel Automated Chromosome Analyzer Software Bundle for Karyotyping and Birth Defect Analysis”. En: *IEEE Access* 11 (2023), págs. 145516-145526. ISSN: 21693536. DOI: 10.1109/ACCESS.2023.3344664.

[29] Fahim Altınordu et al. “A tool for the analysis of chromosomes: KaryoType”. En: *Taxon* 65 (3 jun. de 2016), págs. 586-592. ISSN: 19968175. DOI: 10.12705/653.9.

[30] Ilya Kirov et al. “DRAWID: User-friendly java software for chromosome measurements and idiogram drawing”. En: *Comparative Cytogenetics* 11 (4 2017), págs. 747-757. ISSN: 1993078X. DOI: 10.3897/CompCytogen.v11i4.20830.

[31] Shoaeib Mahmoudi y Ghader Mirzaghaderi. “Tools for Drawing Informative Idiograms”. En: *bioRxiv* (oct. de 2021). DOI: 10.1101/2021.09.29.459870. URL: https://doi.org/10.1101/2021.09.29.459870.

[32] Hua Wei et al. "Classification of Giemsa staining chromosome using input-aware deep convolutional neural network with integrated uncertainty estimates". En: *Biomedical Signal Processing and Control* 71 (ene. de 2022). ISSN: 17468108. DOI: 10.1016/j.bspc.2021.103120.

[33] Chengyu Wang et al. "Extended ResNet and label feature vector based chromosome classification". En: *IEEE Access* 8 (2020), págs. 201098-201108. ISSN: 21693536. DOI: 10.1109/ACCESS.2020.3034684.

[34] Shervin Minaee et al. "Image Segmentation Using Deep Learning: A Survey". En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44 (7 jul. de 2022), págs. 3523-3542. ISSN: 19393539. DOI: 10.1109/TPAMI.2021.3059968.

[35] Ganghao Liu et al. "Chromosome Image Segmentation Based on OTSU and Region Growing Algorithm". En: Institute of Electrical y Electronics Engineers Inc., 2022, págs. 1046-1050. ISBN: 9781665499163. DOI: 10.1109/PRAI55851.2022.9904165.

[36] Ling Chang et al. "An automatic progressive chromosome segmentation approach using deep learning with traditional image processing". En: *Medical and Biological Engineering and Computing* 62 (1 ene. de 2024), págs. 207-223. ISSN: 17410444. DOI: 10.1007/s11517-023-02896-x.

[37] J McGowan-Jordan, RJ Hastings y S Moore. *ISCN 2020*. 1.ª ed. KARGER, nov. de 2020. ISBN: 9783318067064. DOI: 10.1159/000510090.

[38] Victoria Del Castillo Ruíz, Rafael Dulijh Uranga Hernández y Gildardo Zafra de la Rosa. *Genética Clínica*. 2.ª ed. ManualModerno, 2019. ISBN: 9786074486971.

[39] Viridiana Rubí Calzada N y César Torres H. "Segmentación Automática de Imágenes de Cromosomas". En: (nov. de 2012).

[40] Inmaculada Campos-Galindo. "Cytogenetics techniques". En: Elsevier, ene. de 2020, págs. 33-48. ISBN: 9780128165614. DOI: 10.1016/B978-0-12-816561-4.00003-X.

[41] Nobuyuki Otsu. "A Threshold Selection Method from Gray-Level Histograms". En: *IEEE Transactions on Systems, Man, and Cybernetics* 9 (ene. de 1979), pág. 62. DOI: 10.1109/TSMC.1979.4310076.

[42] Ravishankar Chityala y Sridevi Pudipeddi. *Image Processing and Acquisition using Python*. eng. Chapman & Hall / CRC Mathematical and Computational Imaging Sciences Series. Hoboken: CRC Press, 2015. ISBN: 978-0367198084.

[43] Abraham Díaz Nayotl. "Metodología para segmentación de cariotipos utilizando campos aleatorios de Markov". Benemérita Universidad Autónoma de Puebla, jun. de 2023.

[44] Harry Blum. "A transformation for extracting new descriptors of shape". En: *Proceedings of the Symposium on Models for Perception of Speech and Visual Form* (1964), págs. 362-380.

[45] Jim Piper y Erik Granum. "On fully automatic feature measurement for banded chromosome classification". En: *Cytometry* 10 (3 mayo de 1989), págs. 242-255. ISSN: 10970320. DOI: 10.1002/cyto.990100303.

[46] Mehdi Moradi y S. Kamaledin Setarehdan. "New features for automatic classification of human chromosomes: A feasibility study". En: *Pattern Recognition Letters* 27 (1 ene. de 2006), págs. 19-28. ISSN: 01678655. DOI: 10.1016/j.patrec.2005.06.011.

[47] Wei Chen et al. "Improved Zhang-Suen thinning algorithm in binary line drawing applications". En: *2012 International Conference on Systems and Informatics (ICSAI2012)*. Yantai, China: IEEE, mayo de 2012, págs. 1947-1950. DOI: 10.1109/ICSAI.2012.6223430. URL: http://ieeexplore.ieee.org/document/6223430/ (visitado 14-01-2025).

[48] J Cho, S. Y Ryu y S. H. Woo. "A study for the hierarchical artificial neural network model for Giemsa-stained human chromosome classification". En: *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* IEEE, sep. de 2004. ISBN: 0-7803-8439-3. DOI: 10.1109/IEMBS.2004.1404272.

[49] Nirmala Madian y K. B. Jayanthi. "Analysis of human chromosome classification using centromere position". En: *Measurement: Journal of the International Measurement Confederation* 47 (1 2014), págs. 287-295. ISSN: 02632241. DOI: 10.1016/j.measurement.2013.08.033.

[50] Liana Beatriz Juliá Roget y Ricardo Fundora Hernández. "Conjunto de herramientas citogenéticas para el trabajo con cromosomas". Universidad de La Habana, jun. de 2016.

[51] Jon Krohn, Grant Beyleveld y Aglaé Bassens. *Deep learning illustrated: a visual, interactive guide to artificial intelligence*. eng. The Pearson Addison-Wesley data & analytics series. Boston Columbus New York San Francisco Amsterdam Cape Town: Addison-Wesley, 2020. ISBN: 978-0-13-511669-2.

[52] Cristopher Bishop. *Pattern Recognition and Machine Learning*. Ed. por M Jorda, J Kleinberg y Bernhad Scholkopf. 1.ª ed. Vol. 1. Spring, ago. de 2006.

[53] Andreas Christian Müller y Sarah Guido. *Introduction to machine learning with Python: a guide for data scientists*. eng. First edition. Beijing Boston Farnham Sebastopol Tokyo: O'Reilly, 2016. ISBN: 978-1-4493-6941-5.

[54] Jorge Molina Lechuga. "Reconocimiento de Objetos en Tiempo Real para un Robot Humanoide usando Técnicas de Inteligencia Artificial". Benemérita Universidad Autónoma de Puebla, mayo de 2023.

[55] MATLAB & Simulink. *¿Qué son las redes neuronales convolucionales?* URL: https://la.mathworks.com/discovery/convolutional-neural-network.html.

[56] Dharmaraj. *Convolutional Neural Networks (CNN) — Architecture Explained — by Dharmaraj — Medium*. Jun. de 2022. URL: https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243.

[57] Marina Sokolova y Guy Lapalme. "A systematic analysis of performance measures for classification tasks". En: *Information Processing and Management* 45 (4 jul. de 2009), págs. 427-437. ISSN: 03064573. DOI: 10.1016/j.ipm.2009.03.002.

[58] Kai Ming Ting. "Confusion Matrix". En: Springer US, 2011, págs. 209-209. DOI: 10.1007/978-0-387-30164-8_157.

[59] NVIDIA. *Jetson Nano Lleva el Poder de la IA Moderna a los Dispositivos en el Edge — NVIDIA*. URL: https://www.nvidia.com/es-la/autonomous-machines/embedded-systems/jetson-nano/product-development/.

[60] Jetson Inference. *Hello AI World guide to deploying deep-learning inference networks and deep vision primitives with TensorRT and NVIDIA Jetson.* URL: https://github.com/dusty-nv/jetson-inference.

[61] T. Y. Zhang y C. Y. Suen. "A fast parallel algorithm for thinning digital patterns". en. En: *Communications of the ACM* 27.3 (mar. de 1984), págs. 236-239. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/357994.358023. URL: https://dl.acm.org/doi/10.1145/357994.358023 (visitado 15-01-2025).

[62] Enrico Grisan, Enea Poletti y Alfredo Ruggeri. "Automatic segmentation and disentangling of chromosomes in Q-band prometaphase images". En: *IEEE Transactions on Information Technology in Biomedicine* 13 (4 feb. de 2009), págs. 575-581. ISSN: 10897771. DOI: 10.1109/TITB.2009.2014464.

[63] Chengyu Wang et al. "Down Syndrome detection with Swin Transformer architecture". En: *Biomedical Signal Processing and Control* 86 (2023), pág. 105199. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2023.105199. URL: https://www.sciencedirect.com/science/article/pii/S1746809423006328.

[64] Devaraj Somasundaram et al. "A Novel Automated Chromosome Analyzer Software Bundle for Karyotyping and Birth Defect Analysis". En: *IEEE Access* 11 (2023), págs. 145516-145526. ISSN: 21693536. DOI: 10.1109/ACCESS.2023.3344664.

Apéndice A

Anexos

A continuación las tablas A.1, A.2, A.3 y A.4 fue rescatada de [64], donde hace una recopilación de los métodos y técnicas que utilizan cierto autores para la separación de los cromosomas cuando están traslapado, adyacentes y métodos para la clasificación de los cromosomas. Incluye tanto algoritmos tradicionales de procesamiento de imágenes como técnicas basadas en inteligencia artificial, lo que proporciona una visión integral de las estrategias utilizadas en el análisis citogenético automatizado.

A.0.1. Algoritmo de segmentación

Algoritmo 1: Eliminación de objetos no deseados y extracción de objetos de interés

Entrada: Imagen en escala de grises (`imagen_gris`), Umbral K para `CAM`, `area_minima`, `area_maxima`
Salida: `imagen_filtrada` y `contornos_filtrada`

```
bin ← CAM(imagen_gris, K = 2)
krn ← kernel 3 × 3
erosionada ← Erosión(bin, krn)
dilatada ← Dilatación(erosionada, krn, iter=3)
contornos ← EncontrarContornos(dilatada)
mascara ← imagen negra del tamaño de imagen_gris
for c en contornos do
    if area_minima < Área(c) < area_maxima then
        LlenarContorno(mascara, c, 255)
imagen_filtrada ← imagen_gris ∧ mascara
bin_final ← CAM(imagen_filtrada, K = 2)
contornos_filtrada ← EncontrarContornos(bin_final)
return imagen_filtrada, contornos_filtrada
```

A.0.2. Algoritmo 2

Algoritmo 2: Primer filtro: Menor polígono convexo

Entrada: `imagen_filtrada`, `contornos_filtrada`
Salida: `imagen_poligono` y `ratio_list` (lista de razones de solidez)

```
imagen_poligono ← Convertir2BGR(imagen_filtrada)
ratio_list ← [ ]
for c en contornos_filtrada do
    area_obj ← Área(c)
    mpc ← MenorPolígonoConvexo(c)
    area_mpc ← Área(hull)
    if area_mpc > 0 then
        ratio ← area_obj / area_mpc
        Agregar(ratio_list, ratio)
        DibujarContorno(imagen_poligono, mpc, (0,255,0), 2)
return imagen_poligono, ratio_list
```

A.0.3. Algoritmo 3

Algoritmo 3: Segundo Filtro: Elipse que los rodea (excentricidad)

Entrada: `imagen_filtrada`, `contornos_filtrada`
Salida: `imagen_elipse`, listas `individuales` y `agrupados`

```
ratios ← [ ]                                          ▷ Relación eje menor / eje mayor
elipses ← [ ]
imagen_elipse ← CONVERTIR2BGR(imagen_filtrada)
for c en contornos_filtrada do
    if LONGITUD(c) ≥ 5 then
        ell ← AJUSTARELIPSE(c)
        major, minor ← ejes de ell
        ratio ← minor / major
        AGREGAR(ratios, ratio)
        AGREGAR(elipses, ell)
umbral ← PROMEDIO(ratios)
individuales ← [ ]; agrupados ← [ ]
for i en 0..LONGITUD(elipses)-1 do
    if ratios[i] ≥ umbral then
        AGREGAR(individuales, elipses[i])
        DIBUJARELIPSE(imagen_elipse, elipses[i], (0,255,0), 2)
    else
        AGREGAR(agrupados, elipses[i])
        center, (major, minor), angle ← elipses[i]
        radio ← max(major, minor)/2
        DIBUJARCÍRCULO(imagen_elipse, center, radio, (0,0,255), 2)
return imagen_elipse, individuales, agrupados
```

A.0.4. Algoritmo 4

Algoritmo 4: Esqueleto

Entrada: `imagen` (Imagen binaria con valores 0/1 o 0/255)
Salida: Esqueleto `S` de un píxel de grosor

```
1: img ← copia de imagen
2: img[img>0] ← 1
3: procedure IteracionAdelgazamiento( img, iter)
4:     Calcular vecinos P_2, P_3, ..., P_9 mediante desplazamiento circular
5:     A ← P_2 + P_3 + P_4 + P_5 + P_6 + P_7 + P_8 + P_9
6:     C ← número de transiciones 0 → 1 en sentido horario
7:     if iter = 0 then
8:         cond ← (P_2 · P_4 · P_6 = 0) ∧ (P_4 · P_6 · P_8 = 0)
9:     else
10:        cond ← (P_2 · P_4 · P_8 = 0) ∧ (P_2 · P_6 · P_8 = 0)
11:    marcar ← (img=1) ∧ (2 ≤ A ≤ 6) ∧ (C = 1) ∧ cond
12:    img[marcar] ← 0
13:    return img
14: repeat
15:    prev ← copia de img
16:    IteracionAdelgazamiento(img, 0)
17:    IteracionAdelgazamiento(img, 1)
18: Hasta que img = prev
19: return img
```

A.0.5. Algoritmo 5

Algoritmo 5: Tercer filtro: Detección y conteo de puntos terminales en el esqueleto

Entrada: Imagen binaria `imagen_bw`, `radio_puntas`=2, `color_puntas`=(0,0,255)
Salida: `num_puntas`, `S`, `imagen_combinada`

```
B ← (imagen_bw > 127)
if ∑B = 0 then
    return 0, ceros, negro, negro
S ← ESQUELETO(B)
vecinos ← convolución 3×3 (centro=0) sobre S
puntas ← (S=1) ∧ (vecinos=1)
num_puntas ← SUMA(puntas)
imagen_combinada ← CONVERTIR2BGR(B × 255)
imagen_combinada[S=1] ← (0,255,0)                    ▷ Esqueleto en verde
DIBUJARCÍRCULOS(imagen_combinada, posiciones de puntas, radio=radio_puntas, color=color_puntas)
return num_puntas, S, imagen_combinada
```

A.0.6. Algoritmo 6

Algoritmo 6: Clasificación de cromosomas usando modelo entrenado

Entrada: `model` (Modelo entrenado (CNN)), `image_paths` (Lista de rutas de imágenes), `tamano` (Tamaño de redimensionado)
Salida: `predictions` = (`ruta`, `clase_predicha`)

```
predictions ← []
for img_path en image_paths do
    img ← PREPROCESARIMAGEN(img_path, tamano)
    if img ≠ nulo then
        pred ← MODEL.PREDICT(img, verbose=0)
        class_idx ← ARGMAX(pred, axis=1)[0]
        class_label ← CLASS_LABELS[class_idx]
        AGREGAR(predictions, (img_path, class_label))
return predictions
```

A.0.7. Algoritmo 7

Algoritmo 7: Asignación y organización de los cromosomas

Entrada: `predictions` = (`ruta`, `clase`) (Lista de predicciones), `output_folder` (Carpeta de salida) `organizado.png`) (Nombre del archivo final)

Salida: `organizado`

`organizado_folder` ← UnirRutas(output_folder, "organizado")
CrearCarpeta(organizado_folder)
`cromosomas_por_clase` ← diccionario vacío con claves `CLASS_LABELS`
for (`ruta`, `label`) **en** `predictions` **do**
    Agregar(`cromosomas_por_clase`[label], ruta)
Definir malla 6×8 para 23 pares + X/Y
`fig, axes` ← CrearSubplots(6, 8, tamaño=(20,15))
Definir posiciones de los cromosomas organizados (Denver):
    Chr 1 → (0,0), 1b → (0,1), ..., X → (5,4), Y → (5,6), etc.
Desactivar ejes en todas las celdas
for `label` **en** `CLASS_LABELS` **do**
    `cromosomas` ← `cromosomas_por_clase`[label]
    for `i` = 0,1 **si** existen cromosomas **do**
        `clave` ← label si $i = 0$, sino label+"b"
        `row, col` ← `class_positions`[clave]
        `img` ← LeerYRedimensionar(cromosomas[i], (80,120))
        MostrarEnEje(axes[row,col], img, escala de grises)
        PonerTítulo(axes[row,col], "Chr"+label, tamaño=15)
        DesactivarEjes(axes[row,col])
GuardarFigura(fig, ruta completa, dpi=300)
CerrarFigura(fig)
Mostrar mensaje:"Organizado guardado en ruta"

Autor/ Año	Método de segmentación/ Accuracy	Comentarios
Gady Adam (1997)	Método basado en hipótesis	▪ Se trabajó con cromosomas traslapados y cromosomas adyacentes. ▪ Identificación de puntos cóncavos, convexos y de líneas de separación.
Petros Karvelis et al. (2005)	Método Watershed Transform (WT) / 92 %	▪ Solo se trabajó con cromosomas adyacentes. ▪ Segmentación por puntos cóncavos. ▪ 940 cromosomas de los cuales 340 cromosomas tocándose, 29 superpuestos y 571 individuales.
Wacharapong Srisang et al. (2006)	Geometría computacional / 85 %	▪ Solo se trabajó con cromosomas traslapados. ▪ Separación de los cromosomas mediante diagrama Voronoi con posibles puntos de corte obtenidos por triangulaciones Delaunary (DT).
Enrico Grisan et al. (2007)	Esquema de umbralización de variante espacial / 90 % cromosomas traslapados y 92 % por cromosomas adyacentes	▪ Se trabajó con cromosomas adyacentes y traslapados. ▪ 1380 imágenes de cromosomas fueron considerados.
Petros Karvelis et al. (2010)	Watershed Transform / 80.4 % en cromosomas traslapados y 90.6 % por cromosomas adyacentes	▪ Se trabajó con imágenes tinción M-FISH con cromosomas traslapados y adyacentes. ▪ 183 imágenes M-FISH fueron tomadas.

Tabla A.1: Métodos de segmentación.

Autor/ Año	Método de segmentación/ Accuracy	Comentarios
Mousami V. Munot et al. (2011)	Algoritmo Snake modificado con enfoce Greedy / 100 % con 3 cromosomas adyacentes y 95 % para 3 & 4 cromosomas adyacentes.	Solo se trabajó con cromosomas adyacentes.
Mukul A. Joshi et al. (2012)	Geometría computacional / 100 % para 1 & 2 cromosomas traslapados y 88 % para 3 & 4 cromosomas traslapados.	▪ Solo se trabajó con cromosomas traslapados. ▪ Probado con imágenes sintetizadas.
Mousami V. Munot et al. (2013)	Geometría computacional / 98 % sólo identificación del punto de corte.	▪ Solo se trabajó con cromosomas traslapados. ▪ Imágenes con tinción M-FISH fueron consideradas.
Tanvi and Renu Dhir (2014)	Geometría computacional / 87.4 %	▪ Solo se trabajó con cromosomas traslapados. ▪ Los puntos de corte se identifican a lo largo del contorno para separar los cromosomas traslapados.
Devaraj Somasundaram (2014)	Algoritmo de separación por geometría / 94 %	▪ Solo se trabajó con cromosomas traslapados. ▪ Se identifican los puntos y las líneas de corte. Se verifican mediante la hipótesis de separación. ▪ La identificación de cromosomas homólogos se realiza por la posición del centrómero.

Tabla A.2: Continuación métodos de segmentación.

Autor/ Año	Método de clasificación/ Accuracy	Comentarios
Boaz Lerner et al. (1998)	MLP (Perceptrón Multicapa) Algoritmo de clasificación por retropropagación entrenado / 90 %	▪ Segmentación de cromosomas parcialmente. ▪ Dos características consideradas para la clasificación: largo de los cromosomas y posición del centrómero.
John M Conroy et al. (2000)	Descomposición de valores singulares (SVD), análisis de componentes principales (PCA), análisis discriminante de Fisher (FDA) y modelos de Markov ocultos (HMM) / 97 %	▪ De todos lo modelos realizados el que tuvo mejores resultados fue los modelos de Markov pues demuestran ser mejores para la clasificación. ▪ Se probó en imágenes de cromosomas de banda G de bases de datos (Filadelfia, Edimburgo y Copenhague).
Zhenzhen Kou (2002)	Maquina de vectores de soporte (SVM) / 90 %	▪ Las salidas de las neuronas han sido reducidas, de ser 24 paso a 5 salidas. ▪ Reduciendo la dimensión de la red, el número de datos de entrenamiento, tiempo de entrenamiento. ▪ Se tomaron 304 imágenes de la base de datos Copenhagen.
Petros Karvelis et al. (2006)	Clasificador Bayes / 89 %	La segmentación por Morphological Watershed es aplicado a la intensidad del gradiente a la imagen, lo que ayuda a descomponer la imagen en un conjunto de regiones homogéneas.
Xingwei Wang et al. (2008)	Árbol de decisión y Redes Neuronales Artificiales (ANN) / 85 %	Probado con 170 imágenes y proceso adaptado con algoritmos de filtrado, umbralización y etiquetado de imágenes.

Tabla A.3: Métodos de clasificación.

Autor/ Año	Método de clasificación/ Accuracy	Comentarios
Petros Karvelis et al. (2008)	Clasificador Bayes / 82.4 %	Las regiones espaciales y espectrales se toman como entrada para el clasificador.
Yaser Rahimi et al. (2008)	Red neuronal Perceptrón Multicapa con retroalimientación Feedforward / 73 %	Las características para la clasificación se basa en la superficie de los cromosomas, se delimitan por los píxels de los cromosomas y seis momentos se consideran.
Benoit Legrand et al. (2008)	Algoritmo Alineamiento Temporal Dinámico (DTW) / 81 %	Las características para la clasificación son la longitud y el perfil de densidad.
Sunthorn Rungruangbaiyok and Pornchai Phukpattaranont. (2010)	Red Neuronal Probabilística / 68.19 % para muestras femeninas y 61.3 % para muestras masculinas.	Las características consideradas para la clasificación de los cromosomas son el área, el perímetro, el área de banda y la descomposición de valores singulares (SVD).
Enea Poletti et al. (2012)	Clasificador por Redes Neuronales / 94 %	▪ La estimación del eje medio y de la polarización se realiza para la extracción de características. ▪ Se utiliza un algoritmo de reasignación de clases junto con el clasificador de red neuronal para aumentar la tasa de probabilidad de clasificación. ▪ Se utilizó 5474 cromosomas para la clasificación.

Tabla A.4: Continuación métodos de clasificación.