



Universidad Veracruzana

Universidad Veracruzana

Instituto de Investigaciones en Inteligencia Artificial

**Construcción automática de
características para segmentación de
imágenes de pulmones afectados por
COVID-19 utilizando programación
genética**

por

David Herrera Sánchez

Tesis de maestría

Director de tesis:

Dr. Héctor Gabriel Acosta Mesa

Codirector:

Dr. Efrén Mezura Montes

2021

Resumen

En este documento se presenta el trabajo realizado para la solución del problema de segmentación de imágenes de tomografías computarizadas de pulmones afectados por COVID-19 causado por el virus SARS-CoV-2. En la actualidad la manera de identificar el área afectada en los pulmones es mediante la ayuda de un experto, donde él a través de su experiencia y conocimiento a priori determina y señala el área afectada. En determinado momento se vuelve necesaria la ayuda de un sistema computacional con el cual el experto pueda tener un apoyo para realizar un diagnóstico con más seguridad, es por ello que en este trabajo se plantea una solución para realizar la segmentación de estas imágenes. La extracción de características de una imagen es una tarea difícil de realizar, pues de manera típica este proceso de extracción se realiza de manera manual y con la ayuda de un experto. Dada la complejidad que pueda existir en realizar este proceso, se recurre al cómputo evolutivo, de manera específica al paradigma de programación genética. De esta manera se realizan estrategias de búsqueda para obtener posibles soluciones y así lograr una extracción y selección de características aceptables con las que sea permisible realizar la segmentación. La metodología propuesta hace uso de programación genética para evolucionar funciones, en las cuales operadores aritméticos básicos y funciones propias del área de procesamiento de imágenes se combinan con el fin de encontrar el programa o función que sea capaz de realizar la extracción de las características más representativas en la imagen, y así segmentar el área afectada en los pulmones. Cabe mencionar que no se trabaja en un espacio de color, pues las imágenes con las que se trabaja se encuentran en escala de grises. Por otro lado, la manera en cómo se evalúa el desempeño de la segmentación de la imagen a partir de la solución obtenida es utilizando la métrica *DICE*. En esta métrica son comparadas dos imágenes; la primera, es la imagen segmentada por el experto y es la imagen a la cual se busca igualar; la segunda, es la imagen resultante después de haberle aplicado la solución final obtenida por el proceso de búsqueda a la imagen original. Los resultados obtenidos muestran competitividad al ser comparadas de manera visual, y de acuerdo a la métrica se obtienen valores mayores respecto a otros métodos del estado del arte con los que fueron comparados.

Agradecimientos

En primer lugar, quiero agradecer a mis asesores Dr. Héctor Gabriel Acosta Mesa y Dr. Efrén Mezura Montes, por su tiempo, sus consejos y su excelente mentoría a lo largo de este trabajo.

A mis padres, María Modesta e Ildefonso, por su apoyo incondicional en todo momento a lo largo de mi vida y por motivarme a alcanzar nuevas metas.

A mi hermana Karen y su familia, por alentarme en seguir superándome.

A mi novia Greysil, por acompañarme en los días malos y los días llenos de éxito, en los días de vela y de descanso.

A mis amigos de IIE y del IIIA, pues a pesar de la circunstancias de confinamiento siempre seguimos en contacto y apoyándonos.

Índice general

Resumen	I
Agradecimientos	II
Índice de figuras	VI
Índice de cuadros	VIII
1. Introducción	1
1.1. Definición del problema	2
1.2. Motivación	4
1.3. Justificación	5
1.4. Hipótesis	5
1.5. Objetivos	6
1.5.1. Objetivo general	6
1.5.2. Objetivos específicos	6
1.6. Organización del documento	6
2. Marco teórico	8
2.1. Visión artificial y procesamiento de imágenes	8
2.1.1. Representación	10
2.1.2. Valores de los píxeles	12
2.1.2.1. Imágenes en escala de gris	12
2.1.2.2. Imágenes binarias	12
2.1.3. Funciones de operaciones puntuales	13
2.1.4. Operadores morfológicos	15
2.1.4.1. Erosión y dilatación	15
2.1.4.2. Opening y closing	16
2.1.5. Segmentación	17
2.1.6. Métricas de segmentación	18
2.1.6.1. Dice	18
2.1.6.2. Sensitividad y especificidad	19
2.1.6.3. Hausdorff Distance	19
2.2. Cómputo evolutivo	19

2.2.1.	Programación genética	21
2.2.2.	Representación	21
2.2.3.	Inicialización y población	22
2.2.4.	Mecanismos de selección	23
2.2.5.	Operadores de variación	23
2.2.6.	Función de evaluación	24
2.2.7.	Condición de paro	25
3.	Marco referencial	27
4.	Propuesta	31
4.1.	GP-IFC	32
4.1.1.	Configuración de GP-IFC	33
4.1.1.1.	Representación	33
4.1.1.2.	Inicialización, población y condición de paro	34
4.1.1.3.	Mecanismos de selección y operadores de variación	34
4.1.1.4.	Función de evaluación	35
4.1.1.5.	Conjunto de funciones	35
	Función Máscara/umbral	35
4.1.2.	Segmentación	36
4.2.	Resumen	37
5.	Experimentos y resultados	38
5.1.	Experimento 1	39
5.1.1.	Diseño Experimental	39
5.1.1.1.	Parámetros de PG	40
5.1.1.2.	Función de aptitud	40
5.1.2.	Discusión y resultados	40
5.1.2.1.	DICE	42
5.1.2.2.	Sensitividad	44
5.1.2.3.	Especificidad	45
5.1.2.4.	Hausdorff Distance	47
5.1.3.	Resumen	50
5.2.	Experimento 2	51
5.2.1.	Diseño experimental	51
5.2.2.	Discusión y resultados	52
5.2.3.	Resumen	55
5.3.	Experimento 3	56
5.3.1.	Diseño experimental	56
5.3.2.	Discusión y resultados	58
5.3.2.1.	Binarización con clasificador ID3	58
5.3.2.2.	Binarización con clasificador SVM	61
5.3.3.	Resumen	63
5.4.	Experimento 4	64
5.4.1.	Diseño experimental	64

5.4.2.	Discusión y resultados	65
5.4.3.	Resumen	70
5.5.	Comparación con métodos del estado del arte	70
5.5.1.	Resumen	73
5.6.	Resumen	74
6.	Conclusiones y trabajo futuro	75
	Apéndices	77
A.	Imágenes y soluciones resultantes por cada experimento	79
A.1.	Experimento 1	79
A.1.1.	DICE	79
A.1.2.	Especificidad	80
A.1.3.	Senstividad	82
A.1.4.	Hausdorff Distance	83
A.2.	Experimento 2	85
A.3.	Experimento 3	86
A.3.1.	ID3	87
A.3.1.1.	COVID-19	87
A.3.1.2.	Skin lesson	89
A.3.1.3.	Retine Vessel	92
A.3.1.4.	Brain CT images with Intracranial Hemorrhage	96
A.4.	Experimento 4	99
A.4.1.	COVID-19	100
A.4.2.	Skin Lesson	103
A.4.3.	Retine Vessel	106
A.4.4.	Brain CT images with Intracranial Hemorrhage	109
B.	Comparación de imágenes resultantes con métodos del Estado del Arte	112
B.1.	Ejemplo de comparación en los cuatro conjuntos de imágenes	112
	Bibliografía	114

Índice de figuras

1.1.	Imágenes de tomografías computarizadas de pulmones [2].	2
1.2.	Ejemplo de diferentes umbrales aplicados a una imagen [2].	4
2.1.	Representación de una imagen en una matriz	11
2.2.	Comparación entre imagen en escala de grises e imagen binaria.	13
2.3.	Representación de las operaciones píxel.	14
2.4.	Operaciones píxel aplicadas a una imagen [2].	14
2.5.	Dilatación en escala de grises	15
2.6.	Erosión en escala de grises	16
2.7.	Operaciones morfológicas aplicadas a una imagen [2].	17
2.8.	Representación de un programa en árbol	22
2.9.	Representación de la crucea en programación genética	24
2.10.	Representación de la mutación en programación genética	24
2.11.	Diagrama de algoritmo de programación genética [14]	26
4.1.	Ejemplo de representación	34
4.2.	Función máscara.	36
5.1.	Gráfica de convergencia de la mejor ejecución del algoritmo utilizando la métrica DICE.	42
5.2.	Ejemplo de imagen resultante de GP-IFC utilizando la métrica DICE como función de aptitud.	43
5.3.	Gráfica de convergencia de la mejor ejecución del algoritmo utilizando la métrica Sensitividad.	44
5.4.	Ejemplo de imagen resultante de GP-IFC utilizando la métrica Sensitividad como función de aptitud.	45
5.5.	Gráfica de convergencia de la mejor ejecución del algoritmo utilizando la métrica Especificidad.	46
5.6.	Ejemplo de imagen resultante de GP-IFC utilizando la métrica Especificidad como función de aptitud.	47
5.7.	Gráfica de convergencia de la mejor ejecución del algoritmo utilizando la métrica Hausdorff Distance.	48
5.8.	Ejemplo de imagen resultante de GP-IFC utilizando la métrica Hausdorff Distance como función de aptitud.	48
5.9.	Gráfica de convergencia de la Ejecución 5.	50
5.10.	Gráfica de convergencia de la mejor ejecución.	52
5.11.	Ejemplo 1. Binarización con el método Otsu.	53

5.12. Ejemplo 2. Binarización con el método Otsu.	54
5.13. Gráfica de convergencia de la mejor ejecución utilizando ID3 como método de umbralización.	59
5.14. Ejemplo 1. Binarización con clasificador ID3.	60
5.15. Gráfica de convergencia de la mejor ejecución utilizando SVM como método de umbralización.	61
5.16. Ejemplo 1. Binarización con clasificador SVM.	62
5.17. Gráfica de convergencia de la mejor ejecución. Experimento 4 - COVID-19.	66
5.18. Ejemplo de resultados COVID-19.	67
A.1. Solución encontrada por GP-IFC. DICE.	80
A.2. Solución encontrada por GP-IFC. Especificidad.	82
A.3. Solución encontrada por GP-IFC. Sensitividad	83
A.4. Solución encontrada por GPI-FC. Hausdorff Distance.	84
A.5. Solución encontrada por GPI-FC. Etapa 2.	86
A.6. Solución encontrada por GPI-FC. COVID-19	88
A.7. Solución clasificador ID3. COVID-19	89
A.8. Solución encontrada por GPI-FC. Skin Lesson	91
A.9. Solución clasificador ID3. Skin Lesson	92
A.10. Solución encontrada por GPI-FC. Retine Vessel	94
A.11. Solución clasificador ID3. Retine Vessel	95
A.12. Solución encontrada por GPI-FC. Brain CT images with Intracranial Hemorrhage	97
A.13. Solución clasificador ID3. Brain CT images with Intracranial Hemorrhage	98
A.14. Solución clasificador ID3. COVID-19.	102
A.15. Solución clasificador ID3. Skin lesson.	105
A.16. Solución clasificador ID3. Retine Vessel.	108
A.17. Solución clasificador ID3. Brain CT images with Intracranial Hemorrhage.	111

Índice de cuadros

2.1. Imágenes en escala de grises y aplicaciones [9]	12
2.2. Operaciones píxel aritméticas.	14
4.1. Conjunto de funciones iniciales.	35
5.1. Resultados obtenidos para la configuración de parámetros.	41
5.2. Parámetros para ejecución de PG.	41
5.3. Resultados de las 10 ejecuciones utilizando los parámetros y métrica determinadas.	49
5.4. Conjunto de funciones. Experimento 2.	51
5.5. Resultados y estadísticas de los experimentos 1 y 2.	55
5.6. Parámetros para ejecución de PG. Experimento 3.	56
5.7. Conjunto de funciones. Experimento 3.	57
5.8. Resultados y estadísticas de los experimentos 2 y 3 (ID3).	60
5.9. Resultados y estadísticas de los experimentos 2 y 3 (SVM).	63
5.10. Estadísticas de los experimentos 3 (ID3) y 4.	69
5.11. Comparación con métodos del estado del arte	72
A.1. Imágenes resultantes y soluciones encontradas por GP-IFC. DICE.	79
A.2. Imágenes resultantes y soluciones encontradas por GP-IFC. Especificidad.	81
A.3. Imágenes resultantes y soluciones encontradas por GP-IFC. Sensitivi- dad.	82
A.4. Imágenes resultantes y soluciones encontradas por GP-IFC. Hausdorff Distance.	84
A.5. Imágenes resultantes y soluciones encontradas por GP-IFC. Etapa 2.	85
A.6. Imágenes resultantes y soluciones encontradas por GP-IFC. COVID- 19 [2]	87
A.7. Imágenes resultantes y soluciones encontradas por GP-IFC. Skin Les- son [4]	90
A.8. Imágenes resultantes y soluciones encontradas por GP-IFC. Retine Ves- sel [3]	93
A.9. Imágenes resultantes y soluciones encontradas por GP-IFC. Brain CT images with Intracranial Hemorrhage [5].	96
A.10. Resultados de GP-IFC. COVID-19 [2].	100
A.11. Soluciones encontradas por cada característica construida.COVID-19	101
A.12. Resultados de GP-IFC. Skin lesson [4].	103
A.13. Soluciones encontradas por cada característica construida. Skin Lesson	104

A.14. Resultados de GP-IFC. Retine Vessel [3].	106
A.15. Soluciones encontradas por cada característica construida. Retine Vessel.	107
A.16. Imágenes resultantes de Brain CT images with Intracranial Hemorrhage [5].	109
A.17. Soluciones encontradas por cada característica construida. Brain CT images with Intracranial Hemorrhage.	110
B.1. Comparación con métodos del estado del arte en diferentes conjuntos de imágenes. [2], [4], [3], [5]	113

Capítulo 1

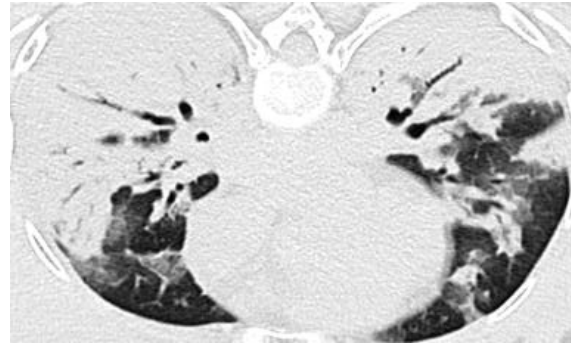
Introducción

Dadas las circunstancias por las que atravesamos actualmente respecto a la pandemia provocada por el virus SARS-CoV-2 causando COVID-19 a la población, la comunidad científica en el ámbito médico ha compartido repositorios de casos de pacientes que han sido contagiados por el virus, en estos repositorios [2] se encuentran tanto diagnósticos médicos como imágenes de Rayos X e imágenes de tomografías computarizadas de los pulmones de dichos pacientes. Estos repositorios contienen las imágenes originales y sus respectivas imágenes donde se muestra el área afectada en los pulmones, la cual es señalada por un experto. Para contrarrestar el trabajo que se realiza a mano por parte del experto en identificar el área afectada en los pulmones, se plantea el uso de programación genética para construir las principales características de las imágenes con el fin de poder determinar el área afectada. Dado que la segmentación de imágenes es una tarea básica para una correcta interpretación y análisis de una cierta escena capturada, sea cual sea el área de aplicación o investigación es muy común que la segmentación sea utilizada como una etapa de preprocesamiento. La segmentación de una imagen permite que ciertas regiones en dicha imagen sean identificadas, o sobresalgan respecto al fondo. La manera en cómo una región de la imagen es determinada como un objeto es bajo un cierto criterio ya sea, por ejemplo, la intensidad de un píxel o textura de una cierta región de la imagen. Es un desafío segmentar una imagen de manera adecuada debido a las variaciones que pueden existir en la imagen, incluso en algunas imágenes existen similitudes entre el fondo y el área de interés, por lo que se requiere de un procesamiento adecuado para poder diferenciar entre la región de fondo y la región del objeto de interés.

Por ejemplo, en el conjunto de imágenes con que se trabaja existen variaciones en las imágenes. En la Figura 1.1(a) y 1.1(b) existe una gran diferencia.



(a) Tomografía computarizada. Ejemplo 1.



(b) Tomografía computarizada. Ejemplo 2.

FIGURA 1.1: Imágenes de tomografías computarizadas de pulmones [2].

En la Figura 1.1(a) se observa de manera clara que los pulmones se encuentran delimitados (color más oscuro), incluso se logran diferenciar del fondo de la imagen (color casi blanco). Por el contrario, en la Figura 1.1(b), la tonalidad que se encuentra en los pulmones se logra mezclar con la tonalidad del fondo. Esta mezcla entre el fondo y el área o región de interés aumentan la complejidad para poder determinar, en primer lugar el área que corresponde a cada pulmón y por consiguiente, determinar el daño que existe en cada uno. Debido a estas variaciones se da lugar a que un experto determine de manera manual cuál método utilizar para encontrar e identificar el área de interés. Estas técnicas que se realizan a mano dan origen al término en inglés *Hand craft feature*. En la actualidad, existen diferentes métodos para realizar el procesamiento de segmentación en una imagen, puede ser mediante umbrales, histogramas, agrupamientos basados en umbrales o entropías [10], pero estos métodos, como fue mencionado, se llevan a cabo de manera manual, es decir, son métodos guiados por un experto(a), y el(ella) es quién decide, con base en su experiencia, cuál método utilizar y qué funciones son las más adecuadas para realizar la segmentación y obtener como resultado el objeto de interés del resto de la imagen.

1.1. Definición del problema

Dado el lineamiento emitido por el gobierno de México [1], la prueba competente para determinar si un paciente está infectado es mediante la realización de la prueba

de PCR. Una desventaja de esta prueba es el tiempo que demora en obtener los resultados el laboratorio, pues desde que es tomada la muestra hasta el momento en que se entregan los resultados, toma aproximadamente 48 hrs. Además, se es requerido un cuidado y transporte especializado tal y como se indica en los lineamientos [1]. Debido a este tipo de situaciones y complicaciones, se recurre a otras técnicas para evaluar al paciente, que en algunos casos ya presenta síntomas. En estas otras técnicas se hace uso de sistemas computarizados, mediante los cuales el personal médico especializado las toma como herramientas, pues son de gran ayuda y facilitan el trabajo para realizar una evaluación, tal es el caso de los *Rayos X* o las *Tomografías Computarizadas*. En ambas técnicas las imágenes obtenidas son analizadas y evaluadas por expertos.

El problema radica en la alta demanda de expertos para evaluar las imágenes, pues debido a la sobrecarga de trabajo el personal tiende a trabajar largas jornadas y sin descanso. Este tipo de situación contribuye a que a la hora de evaluar las imágenes exista un cansancio visual, el cual puede conducir al experto a realizar una mala evaluación e interpretación de dichas imágenes, incluso pasar por alto pequeños detalles que pueden tener repercusiones. Como apoyo fundamental para el personal médico especializado se toman en consideración técnicas utilizadas mediante sistemas computarizados con el fin de ayudar a determinar y evaluar estas imágenes de manera automática. Este procesamiento es la segmentación de las imágenes, en donde se determina el área de interés dentro de una imagen, que en este caso en particular es identificar el área afectada en los pulmones de los pacientes. Como fue mencionado en 1, la tarea de segmentar una imagen se vuelve compleja dada la naturaleza de las imágenes que se desean procesar, ya que las imágenes contenidas en la base de datos varían de manera considerable. En algunos casos, podría decirse que en “*los mejores*”, en la imagen se logra observar de manera definida el área de los pulmones, logrando diferenciar entre el fondo de la imagen y el pulmón. Por otro lado, existen casos donde el pulmón no se logra diferenciar respecto al fondo de la imagen y por ende el área afectada es más difícil de percibir, incluso por el personal médico especializado.

En la Figura 1.2 se muestra el resultado de aplicar diferentes umbrales a una sola imagen 1.2(a) y como se puede observar las imágenes resultantes 1.2(b) y 1.2(c) son bastante diferentes, pues en ambas sobresalen áreas diferentes. La percepción visual cambia, es aquí donde se muestra la manera en como se vuelve compleja la tarea de segmentar una imagen. Esto da pie al término de *Hand Craft Feature* debido a que se tiene que llevar a cabo una búsqueda para obtener los parámetros adecuados. Esto se debe llevar a cabo no solo para segmentar una imagen, sino para segmentar un

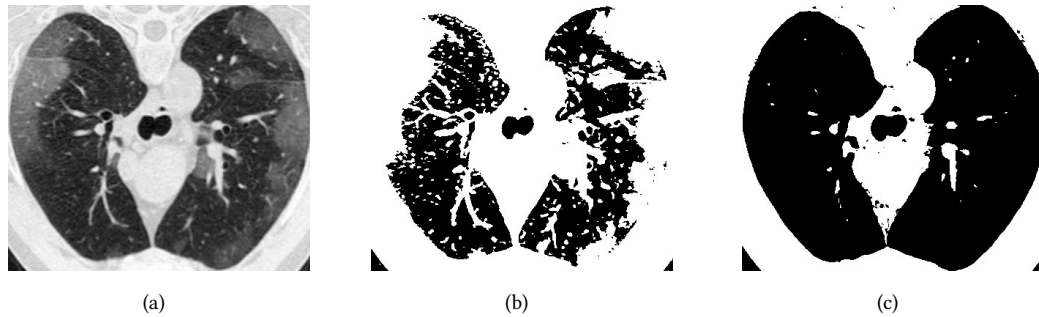


FIGURA 1.2: Ejemplo de diferentes umbrales aplicados a una imagen [2].

conjunto de imágenes, con el fin de que el área de interés se muestre o resalte del resto de la imagen.

1.2. Motivación

Debido a que la segmentación tiende a ser una tarea compleja en una base de datos donde las imágenes contenidas varían mucho, encontrar de manera manual las funciones y métodos adecuados para obtener el área de interés en todas las imágenes es una tarea difícil. De hecho, esta tarea se convierte en un proceso de búsqueda, pues lo que se realiza es la búsqueda de un conjunto de funciones y métodos para lograr identificar el área que es de interés. Dado la naturaleza del problema, para enfrentar este reto se recurre al uso de programación genética, uno de los paradigmas del cómputo evolutivo, el cual permite evolucionar funciones/programas computacionales para resolver un problema en específico. Por mencionar algunas de las bondades y ventajas que tiene este paradigma es que ha sido una herramienta competitiva para la solución de problemas en diferentes áreas debido a la flexibilidad en su representación, el desempeño competitivo que tiene para realizar una búsqueda y también por una mejor interpretación de los resultados que se obtienen al final del proceso. Parte de la motivación también está basada en que la segmentación de imágenes médicas utilizando programación genética es un área poco explorada. Además, trabajos relacionados donde se realiza la segmentación de imágenes utilizan un número considerable de funciones [17][6].

1.3. Justificación

Con el paso del tiempo la tecnología continúa avanzando, facilitando la realización de tareas que solían ser difíciles de realizar. En la actualidad, en el área médica es más sencillo adquirir datos de los pacientes de diversas maneras. Una de ellas es mediante el uso de tomografías computarizadas, las cuales han permitido evaluar al paciente en diferentes áreas del cuerpo, esto es gracias a la gran cantidad de información que se puede obtener mediante este procedimiento. De manera puntual, se puede utilizar para determinar si existen tumores o lesiones dentro del cuerpo, también se pueden obtener imágenes de la cabeza para determinar si existen anomalías como un coágulo o lesiones severas [8]. También ha sido útil para determinar fracturas de huesos, tendones e incluso articulaciones. Además, sirve para obtener información de las vías respiratorias al ser capturadas imágenes de los pulmones. Dada esta facilidad de extracción de información del cuerpo mediante imágenes, es necesario el desarrollo de métodos capaces de realizar una segmentación adecuada con el fin de ayudar a determinar lesiones o daños causados en los pacientes. En nuestro caso y dadas las circunstancias actuales son necesarias estas técnicas para que sean capaces de ayudar o dar soporte al experto y así poder realizar una evaluación con el mínimo error.

Por otra parte, el uso de programación genética y su claridad en la interpretación de resultados, nos permite conocer el procedimiento que se está realizando para llegar al resultado, que en nuestro caso nos muestra las funciones utilizadas y el orden en cómo son aplicadas al conjunto de imágenes para llegar al resultado final.

1.4. Hipótesis

Mediante la construcción automática de características por programación genética utilizando solo las funciones propuestas, se logrará obtener una mejor segmentación del área afectada en los pulmones causada por COVID-19 respecto a los métodos de programación genética más actuales del estado del arte.

1.5. Objetivos

1.5.1. Objetivo general

Mediante el uso de programación genética y bajo el enfoque de representación de árbol crear una función capaz de construir características para segmentar de manera competitiva imágenes de tomografías computarizadas de pulmones y así determinar el área afectada por COVID-19.

1.5.2. Objetivos específicos

- Mediante programación genética evolucionar un conjunto de funciones con operadores aritméticos básicos y funciones propias para procesamiento de imágenes para así aplicarlas en las imágenes en escala de grises del repositorio *COVID-19* [2].
- Segmentar dicho conjunto de imágenes y medir el desempeño de la función utilizando las métricas DICE, especificidad, sensibilidad y Hausdorff distance.
- Resolver el problema de segmentación utilizando los parámetros adecuados de calibración para la programación genética.
- Comparar el desempeño de construcción de características del algoritmo respecto a métodos del estado del arte utilizando las bases de datos de imágenes *COVID-19* [2], *Skin lesion* [4], *Retine Vessel* [3] y *Brain CT images with Intracranial Hemorrhage* [5].

1.6. Organización del documento

El presente documento se encuentra organizado de la siguiente manera:

- **Capítulo 2.** Se presenta el marco teórico referente al procesamiento de imágenes y el cómputo evolutivo centrándose en el paradigma de programación genética.
- **Capítulo 3.** Se presenta el marco referencial donde se citan trabajos previos relacionados a la segmentación de imágenes utilizando programación genética.

- **Capítulo 4.** Se muestra de manera detallada la propuesta para realizar la segmentación en el conjunto de imágenes.
- **Capítulo 5** Se presenta la parte experimental de la implementación y los resultados obtenidos a partir de dicha experimentación, así como la comparación con métodos del estado del arte.
- **Capítulo 6.** En este apartado, tras la finalización de la experimentación, se muestran las conclusiones en las cuales culminó este trabajo de investigación.

Capítulo 2

Marco teórico

2.1. Visión artificial y procesamiento de imágenes

El sentido de la visión puede ser considerado como un proceso, mediante el cual nos permite a los seres humanos y a la mayoría de los animales obtener gran información respecto al entorno que nos rodea. Este sentido es de gran ayuda, pues nos permite desenvolvernó en nuestro entorno e interactuar con él y con los demás seres vivos. Percibimos el entorno de manera tridimensional y es por ello que podemos determinar si algún objeto es grande, pequeño, si se encuentra lejos o cerca de nosotros. También podemos determinar la forma, el color, incluso a través del reconocimiento de patrones relacionados a un objeto podemos diferenciar un objeto de manera específica respecto al resto de la escena que se está observando. Bajo nuestra experiencia podemos decir que los ojos son sensores muy efectivos para reconocer objetos o patrones, nos permiten navegar, evitar obstáculos y también movernos en nuestro entorno permitiéndonos interactuar con nuestros semejantes. Por otro lado nuestra visión nos permite, al momento de interactuar con más personas, reconocer ciertas emociones de acuerdo con las gesticulaciones que se expresan en la cara. La visión engloba demasiada información que nos es útil para determinar nuestras acciones y sensaciones, incluso el simple hecho de observar desencadena una serie de pensamientos en donde se asocia más información. Con el paso del tiempo y tras el desarrollo de nuevas tecnologías se ha podido mimetizar la función de un ojo, un claro ejemplo son las cámaras digitales, que en la actualidad están presentes hasta en un celular inteligente. Este gran avance ha

permitido el desarrollo de nuevas herramientas para facilitar tareas que en cierto momento eran, ya sea complejas de realizar o requerían de un tiempo considerable para poder realizarse.

En un sistema de visión por computadora se trata de describir los momentos capturados del entorno mediante ciertos algoritmos, los cuales en algún momento al realizar una interpretación pueden contener fallas. Además, el proceso para realizar la interpretación en cuestión computacional requiere de un proceso de información bastante moderado. En cambio, lo que los humanos hacemos es totalmente diferente, ya que nuestro sistema de visión realiza una interpretación de manera rápida y sencilla, que además este procesamiento es paralelo con otras funciones del cuerpo, ya que aparte de realizar una interpretación se encuentran involucrados los demás sentidos, los cuales están realizando otras acciones.

La visión por computadora tiene como objetivo proporcionar sistemas informáticos con la capacidad de poder simular la visión humana y así generar una representación numérica o simbólica del mundo real. El uso de sistemas de procesamiento de imágenes permite realizar tareas como una clasificación en un conjunto de imágenes, algunas aplicaciones realizan detección de objetos, reconstrucción de ciertas escenas, segmentación, entre muchas otras. El conjunto de operaciones y técnicas comúnmente aplicadas a las imágenes digitales son conocidas como procesamiento de imágenes. Desde un enfoque computacional, los sistemas de percepción visual están contruidos en cinco módulos [18]:

- **Generación de la imagen.** Es la parte en donde la escena es capturada mediante un sensor especializado. Por ejemplo una cámara digital, la cual se encarga de generar un representación electrónica del mundo.
- **Preprocesamiento.** Las imágenes que ya fueron capturados en el paso anterior son modificadas para realizar un mejoramiento en la calidad de la imagen. Por ejemplo puede ser corrección de color o incluso un compresión para minimizar el espacio utilizado para almacenar dicha imagen.
- **Extracción de características.** Las imágenes son procesadas para poder extraer elementos dentro de la imagen, por ejemplo objetos, bordes, o regiones.
- **Segmentación.** Importantes características de la imagen son seleccionadas para realizar una identificación de un objeto o región de interés.

- **Entendimiento de la imagen.** Este es el nivel más alto del procesamiento de imágenes, aquí el conocimiento y el significado puede ser inferido desde la imagen.

El procesamiento digital de imágenes se puede definir como la operación o modificación de la imagen utilizando programas computacionales. Existe una diferencia importante entre visión por computadora y procesamiento de imágenes, de manera específica las imágenes que se trabajan en procesamiento de imágenes no provienen únicamente del espectro visible con el que es semejado la percepción visual humana, debido a que las imágenes se pueden generar mediante la captación de cualquier área del espectro electromagnético, un claro ejemplo son los sistemas de procesamiento de imagen que trabajan con imágenes generadas a partir de rayos X, rayos gamma, resonancia magnética, entre otras [27].

2.1.1. Representación

El material principal con el que trabaja tanto la visión artificial y el procesamiento de imágenes son las imágenes, y como se mencionó con anterioridad, estas son consideradas como una representación del mundo real. De la misma manera en ellas se encuentra información, la cual es captada mediante un dispositivo. Para obtener imágenes es necesario llevar a cabo un proceso de toma o muestreo, después se realiza una cuantificación y posteriormente una codificación [11]. Una imagen puede definirse como una función bidimensional donde es cuantificada la intensidad de la luz que en la mayoría de los casos es el espectro visible.

Una imagen de manera típica es representada como $I(x,y)$ donde el valor de la intensidad está contenido en las coordenadas x y y . En el mismo sentido, el modelo en como es representada una imagen es mediante una matriz [23], tal y como se muestra a continuación.

$$\mathbf{I(x,y)} = \begin{bmatrix} I(1,1) & I(2,1) & \cdots & I(N,1) \\ I(1,2) & I(2,2) & \cdots & I(N,2) \\ \vdots & \vdots & \ddots & \vdots \\ I(1,M) & I(2,M) & \cdots & I(N,M) \end{bmatrix} \quad (2.1)$$

Como ejemplo se muestra la representación de una imagen en forma matricial en la Figura 2.1. En 2.1(a) se muestra la imagen original de una tomografía computarizada

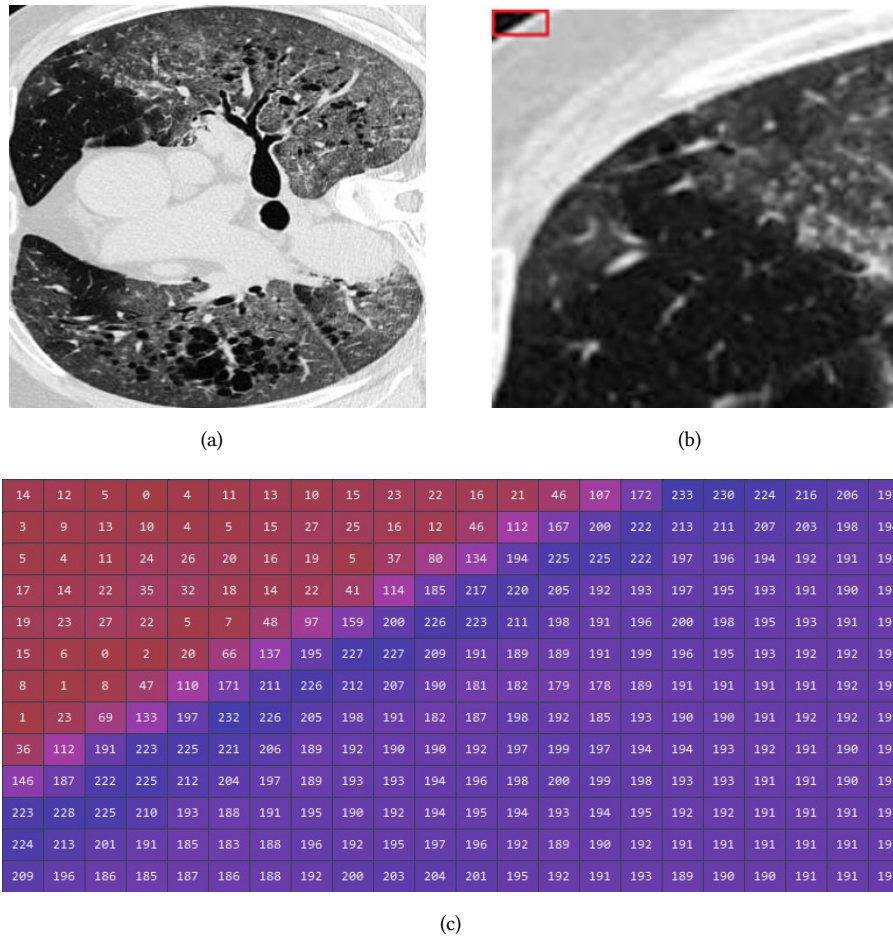


FIGURA 2.1: Representación de una imagen en una matriz

de unos pulmones afectados por COVID-19. En 2.1(b) se muestra en la parte superior izquierda en un recuadro rojo el área que es tomada de la imagen y es representada en 2.1(c) de manera matricial tal y como está en (2.1). El tamaño de la matriz original es equivalente al tamaño de la imagen original, es decir, si la imagen en 2.1(a) tiene dimensiones de 358 píxeles de ancho \times 346 píxeles de alto, la matriz tendrá una dimensión de 358×346 . En 2.1(c) se logra percibir un color rojo en donde están contenidos valores muy pequeños dentro de la matriz que rondan valores entre 1 y 40, en cambio en donde está un tono de color morado - azul se encuentran valores que rondan entre 150 a 200. Estos valores numéricos hacen referencia al valor que toma cada uno de los píxeles de la imagen de acuerdo con la intensidad que está en la imagen original 2.1(a).

2.1.2. Valores de los píxeles

Dentro de una imagen se tiene cierta información, que como ya vimos en la representación matricial toma un valor reflejando la intensidad. Cada píxel o posición en la matriz en las coordenadas x,y contiene diferentes valores y por ende información. Este píxel puede representar 2^n valores diferentes. El exponente n es llamado la profundidad del bit en la imagen. El valor que pueda llegar a tomar n en cada píxel depende totalmente del tipo de imagen con la que se esté trabajando, ya sean imágenes binarias, imágenes en escala de gris o imágenes a color, donde el espacio de color más utilizado es el *RGB*.

2.1.2.1. Imágenes en escala de gris

La información en una imagen en escala de grises consiste en un solo canal, este canal representa la intensidad, brillo o densidad de la imagen. Como se puede observar en la Figura 2.1(a) la imagen se encuentra en escala de grises, y los valores altos representan la intensidad de luz. El rango de esta imagen se encuentra entre los valores de 0 y 255, donde el valor de 0 representa la menor intensidad y se torna color negro, por el contrario el valor máximo que es 255 se torna color blanco. El conjunto de imágenes con las que se realiza este trabajo se encuentran en escala de grises, y la profundidad del píxel es de 2^8 , por ende los valores contenidos en la imagen se encontraran en el rango $[0,255]$.

CUADRO 2.1: Imágenes en escala de grises y aplicaciones [9]

Canales	Rango	Aplicaciones
1	[0,1]	Documentos, fax, ilustraciones. (Imágenes binarias)
1	[0,255]	Medicina, fotografías, escáneres, impresiones. (Universal)
1	[0,4095]	Medicina, fotografías, escáneres, impresiones. (Alta calidad)
1	[0,16383]	Medicina, fotografías, escáneres, impresiones. (Profesional)
1	[0,65535]	Medicina, impresiones. (Muy alta calidad)

2.1.2.2. Imágenes binarias

Las imágenes binarias, de acuerdo con la tabla 2.1 los valores que pueden tomar los píxeles son solo 1 y 0, blanco o negro respectivamente. Este tipo de imágenes son

de gran ayuda, principalmente para la tarea de segmentación, puesto que el área que no es de interés en la imagen se le asigna el valor de 0 y la región que si es de interés le es asignado el valor de 1. El conjunto de imágenes que se utilizan para realizar este trabajo contiene imágenes binarias. Estas imágenes binarias son las que nos muestran el verdadero daño que se tiene en los pulmones. Como se puede ver en la Figura 2.2 en 2.2(a) se encuentra la imagen original de una tomografía computarizada donde se muestran los pulmones, esta imagen es la que analiza un experto y él es quien señala el área afectada por COVID-19, creando así una imagen binaria donde solo es marcada la región de interés, dando como resultado la imagen 2.2(b).

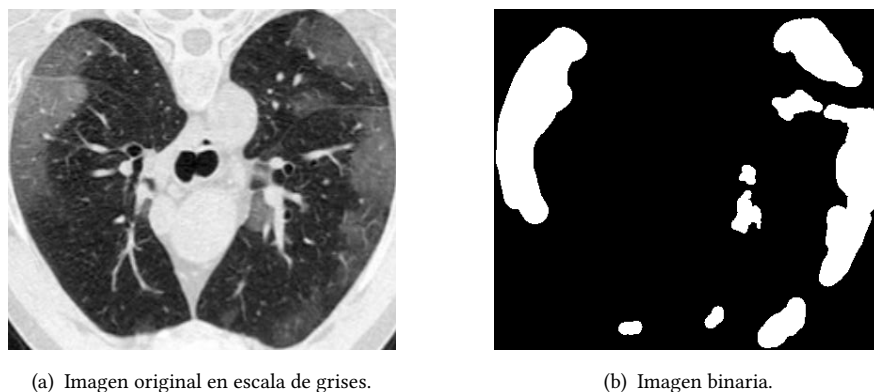


FIGURA 2.2: Comparación entre imagen en escala de grises e imagen binaria.

2.1.3. Funciones de operaciones puntuales

Nos referimos a operaciones píxel a aquellas operaciones realizadas sobre la imagen en donde solo se toma en cuenta el valor del píxel de la imagen. Cada uno de los nuevos valores calculados $Valor' = I'(x, y)$ es dependiente del valor del píxel original $Valor = I(x, y)$ en la misma posición sin importar el valor de los valores vecinos a ese píxel. Este nuevo valor es determinado mediante una función $f[I(x, y)] \rightarrow I'(x, y)$. Si esta función es independiente de las coordenadas de la imagen la función es llamada homogénea [11].

Los operadores aritméticos tales como la suma, resta, división y multiplicación son utilizadas como funciones de operación píxel para modificar los valores contenidos en la imagen. Al realizar este tipo de operaciones la información contenida en la imagen sufre cambios y por ende en la percepción visual se ve reflejada. Una ventaja de las operaciones píxel es que se pueden involucrar más de una imagen a la vez, es decir podemos sumar una imagen con otra, se pueden restar, dividir o multiplicar entre ellas.

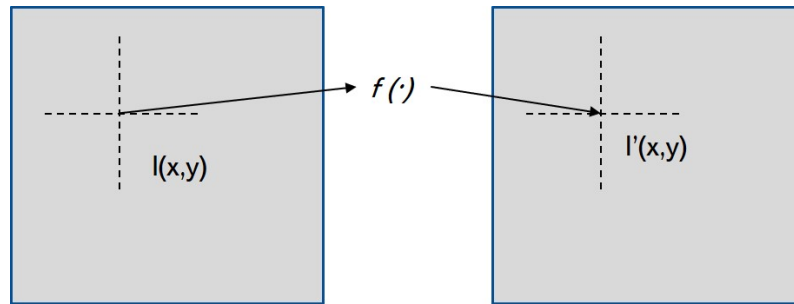
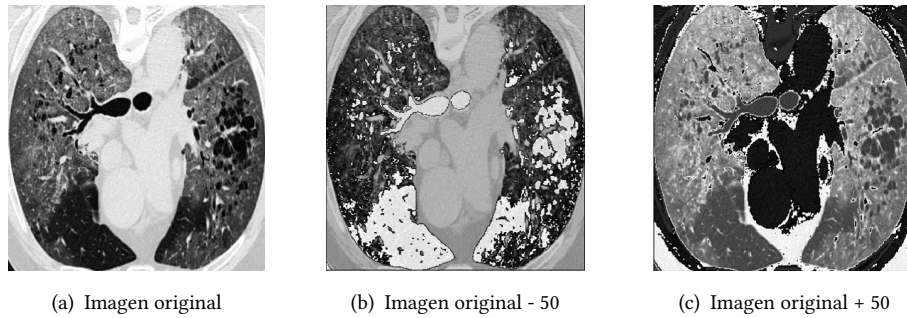


FIGURA 2.3: Representación de las operaciones píxel.

Cabe mencionar que los valores que puede tomar cada uno de los píxeles en la imagen siempre se van a encontrar dentro del rango en el que se esté trabajando. Para nuestro caso, al aplicar estas operaciones en nuestro conjunto de imágenes, los valores que pueden llegar a tomar los píxeles están entre 0 y 255.



(a) Imagen original

(b) Imagen original - 50

(c) Imagen original + 50

FIGURA 2.4: Operaciones píxel aplicadas a una imagen [2].

CUADRO 2.2: Operaciones píxel aritméticas.

Operación	Rango
Suma	$I_{\text{resultado}}(x, y) \leftarrow I_1(x, y) + I_1(x, y)$ $I_{\text{resultado}}(x, y) \leftarrow I_1(x, y) + k$
Resta	$I_{\text{resultado}}(x, y) \leftarrow I_1(x, y) - I_1(x, y)$ $I_{\text{resultado}}(x, y) \leftarrow I_1(x, y) - k$
Multiplicación	$I_{\text{resultado}}(x, y) \leftarrow I_1(x, y) \times I_1(x, y)$ $I_{\text{resultado}}(x, y) \leftarrow I_1(x, y) \times k$
División	$I_{\text{resultado}}(x, y) \leftarrow I_1(x, y) \div I_1(x, y)$ $I_{\text{resultado}}(x, y) \leftarrow I_1(x, y) \div k$

2.1.4. Operadores morfológicos

Los operadores morfológicos fueron diseñados y aplicados originalmente en imágenes binarias. En las operaciones morfológicas binarias se utilizan las funciones AND y OR. Para aplicar estos operadores a imágenes en escala de grises son remplazados por los operadores aritméticos MIN y MAX, respectivamente.

2.1.4.1. Erosión y dilatación

La operación de dilatación en escala de grises $I \oplus H$ está definido como MAX de los valores en H sumados a los valores de la subimagen de I, esto es:

$$(I \oplus H)(x, y) = \max_{(x,y) \in H} (I(x + i, y + j) + H(i, j)) \quad (2.2)$$

El ejemplo de dilatación en escala de grises mostrado en la Figura 2.5 nos dice lo siguiente: tenemos una imagen I y una estructura H. La estructura H de 3×3 elementos se coloca en la imagen I de manera alineada en la parte superior izquierda (recuadro verde). Cada valor de H se suma al elemento correspondiente de I. El resultado de $(H + I)$ se muestra abajo. Ahora bien, el máximo valor que en este caso es 8 (señalado en rojo) es añadido en la imagen resultante $I \oplus H$ y se coloca en la posición actual del origen de H (marco recuadro azul).

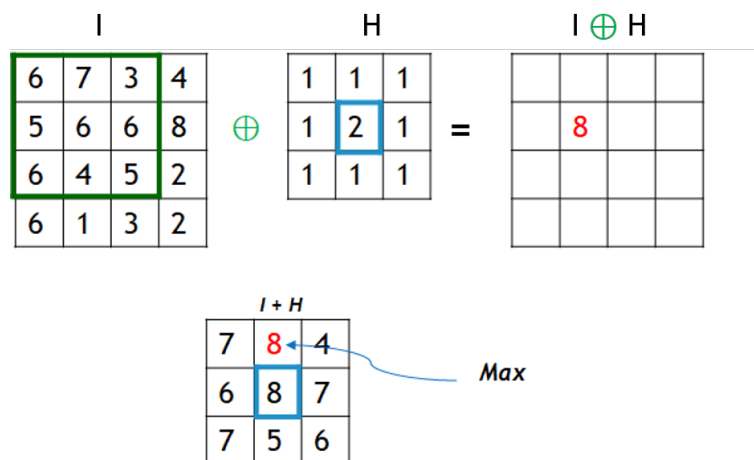


FIGURA 2.5: Dilatación en escala de grises

De manera similar a la dilatación, la operación de erosión queda de la siguiente manera:

$$(I \ominus H)(x, y) = \min_{(x,y) \in H} (I(x + i, y + j) - H(i, j)) \quad (2.3)$$

El ejemplo correspondiente para la erosión se muestra en la Figura 2.6 y va de la siguiente manera: La estructura H de 3×3 elementos se coloca en la imagen I de manera alineada en la parte superior izquierda (recuadro verde). Cada valor de H se resta al elemento correspondiente de I . El resultado de $(H - I)$ se muestra abajo. Ahora bien, el mínimo valor que en este caso es 2 (señalado en rojo) es añadido en la imagen resultante $I \ominus H$ y se coloca en la posición actual del origen de H (marco recuadro azul).

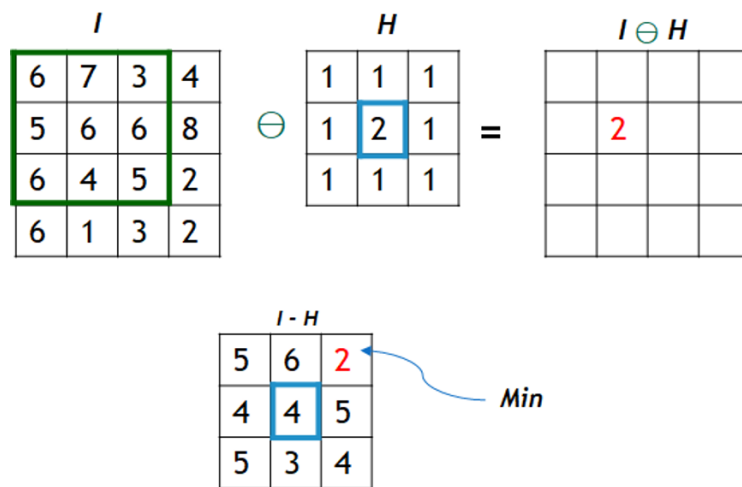


FIGURA 2.6: Erosión en escala de grises

2.1.4.2. Opening y closing

Debido a que *opening* y *closing* son operaciones compuestas a partir de la dilatación y erosión quedan de la siguiente manera [23]. Para opening se tiene:

$$I \circ H = (I \ominus H) \oplus H. \quad (2.4)$$

donde $I \circ H$ denota una erosión seguida por una dilatación con la misma estructura de H . El principal efecto del operador opening es que en estructuras que son más pequeñas que H en el primer plano se eliminan tras el paso de la erosión, y así las estructuras restantes se suavizan por el siguiente paso que es la dilatación. Estas estructuras restantes vuelven a su tamaño original.

Se puede decir que la operación de *closing* es la contraparte de *opening*. Para la operación de *closing* se tiene:

$$I \bullet H = (I \oplus H) \ominus H. \quad (2.5)$$

Closing realiza una limpieza al cerrar orificios. Visualmente se observa partes oscuras tienden a crecer siempre y cuando la estructura H sea mayor que la subimagen de I .

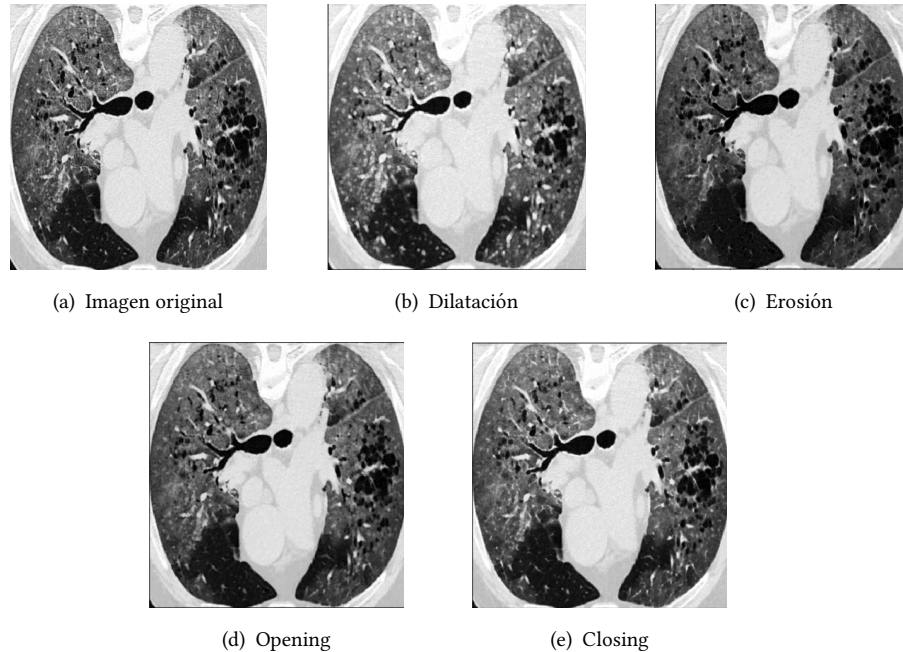


FIGURA 2.7: Operaciones morfológicas aplicadas a una imagen [2].

2.1.5. Segmentación

Formalmente, el problema de segmentación se establece como el proceso de separar una imagen en regiones representativas de acuerdo con al menos un criterio compartido dentro de una región. En la mayoría de los sistemas de visión por computadora se requiere segmentación. Una correcta segmentación puede mejorar el rendimiento de todo un sistema en general, ya que es común que otros procesos dependan de los resultados de esa segmentación.

La división de una imagen en regiones permite analizar y procesar solo las regiones de interés. Existen diferentes técnicas y métodos para la segmentación de imágenes [18]. Por ejemplo, la técnica de *Thresholding* utiliza un histograma en la escala de grises para seleccionar un valor de umbral que está entre los valores de 0 y 255. También está

el *Método basado en bordes* donde los bordes son utilizados para identificar los límites de una región que separan dos o más clases adyacentes. Por otro lado, *Clustering* clasifica cada grupo de píxeles disponibles de una imagen de acuerdo con ciertas características. Unos de las técnicas que muestran resultados sobresalientes son las *Redes neuronales* que dada su naturaleza son aproximadores universales que pueden ser entrenados para predecir o clasificar series de datos, que en este caso son imágenes. Todos estos métodos atacan el mismo problema, la segmentación. Es claro que cada uno de ellos tiene un sesgo hacia un área en específico para lo cuál fue creado o propuesto, es por ello que ninguno puede realizar la misma tarea en diferentes áreas. Como fue mencionado anteriormente, esto problema se puede interpretar como un problema de búsqueda, donde es necesario encontrar ciertas características con las cuáles la segmentación se realiza de manera más sencilla, y claro, también dependerá del área donde sea aplicada.

2.1.6. Métricas de segmentación

La manera en cómo es evaluado el método o la técnica de segmentación es mediante la comparación entre la imagen a la que queremos llegar o imagen objetivo con respecto a la imagen segmentada por la función o método creado.

2.1.6.1. Dice

El coeficiente de DICE también llamado el índice de solapamiento es una de las métricas más utilizadas para validar la segmentación de imágenes médicas [28]. Es común utilizar esta métrica ya que se realiza una comparación directa entre el área generada de manera automática y el área verdadera, midiendo prácticamente la reproducibilidad. Esta métrica está definida por:

$$DICE = \frac{2|S_g^1 \cap S_t^1|}{|S_g^1| + |S_t^1|} = \frac{2VP}{2VP + FP + FN} \quad (2.6)$$

Donde **VP** son los Verdaderos Positivos, **FP** son Falsos Positivos y **FN** son los Falsos Negativos. Cabe destacar que esta métrica es la que mejor se adapta para evaluar el modelo que se genera en este trabajo.

2.1.6.2. Sensitividad y especificidad

Tasa de verdaderos positivos (True Positive Rate) también conocido como sensitividad y *Recall* mide la porción de los píxeles positivos en la imagen original que también están contenidos en la segmentación que está siendo evaluada. De manera antagónica la especificidad mide la porción de píxeles negativos en la segmentación con respecto a la imagen original. Estas dos medidas no son muy utilizadas debido a que penalizan más en los errores pequeños, es decir en pequeñas segmentaciones estas aumentan la penalización que respecto con segmentaciones grandes[28]. Estas métricas están definidas por:

$$\text{Sensitividad} = \frac{VP}{VP + FN} \quad (2.7)$$

$$\text{Especificidad} = \frac{VN}{VN + FP} \quad (2.8)$$

Donde **VP** son los Verdaderos Positivos, **VN** son Verdaderos Negativos, **FP** son Falsos Positivos y **FN** son los Falsos Negativos.

2.1.6.3. Hausdorff Distance

Esta métrica es generalmente sensible a valores atípicos. La distancia de Hausdorff entre dos puntos finitos (A,B) están definidos por [28]:

$$HD(A, B) = \max(h(A, B), h(B, A)) \quad (2.9)$$

donde $h(A, B)$ es llamada *La distancia directa de Hausdorff* y está dada por:

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \quad (2.10)$$

donde $\|a - b\|$ es la distancia entre a y b . Esta distancia puede ser por ejemplo la distancia euclidiana [28].

2.2. Cómputo evolutivo

El cómputo evolutivo está inspirado por la teoría de la evolución propuesta por Charles Darwin, donde dado un ambiente se encuentra una población de individuos

que hacen lo posible para sobrevivir. Entre ellos luchan para aprovechar los recursos y el individuo más apto es quién logra sobrevivir. Entre los individuos de la población se generan descendencias asegurando la supervivencia de la especie, de la misma manera se generan algunos cambios en ciertos individuos para poder adaptarse al ambiente. Basándose en esta metáfora, el cómputo evolutivo es utilizado para resolver problemas en diferentes áreas. Los individuos de la población representan las posibles soluciones al problema. Los individuos tienen una aptitud la cual representa la calidad de la solución respecto al problema a resolver y el ambiente está representado por dicho problema.

Dentro del cómputo evolutivo hay cuatro paradigmas que han servido como bases para el área dando paso a mantener una gran actividad en el área. Estos paradigmas son [12]:

- Algoritmos genéticos concebidos por Jonh Holland.
- Programación genética por John R. Koza.
- Estrategias evolutivas por Rechenberg y Schwefel.
- Programación evolutiva por Fogel.

Las principales diferencias entre estos cuatro paradigmas recae en la manera en como son representados las soluciones / individuos, los métodos de selección y los operadores que se utilizan en cada uno de ellos.

Los componentes más importantes en los algoritmos evolutivos son los siguientes [12]:

- **Representación.** Esta es la primera parte, pues se debe definir una manera de relacionar el algoritmo con el mundo real para resolver un problema.
- **Población.** El objetivo de tener población es para mantener posibles soluciones. Así mismo, es la unidad de evolución mediante la cuál con el paso de las generaciones se obtendrá un individuo apto para la solución del problema.
- **Función de evaluación.** Es la que evalúa el desempeño o la aptitud de un individuo o posible solución. Es mediante la cual se determina si un individuo tiene un desempeño bueno o malo. De la misma manera, abre paso para realizar una selección y así facilitar las mejoras.

- **Mecanismos de selección.** Los mecanismos de selección nos permiten realizar dos cosas. Primero, estos nos permiten elegir a los padres de un individuo de acuerdo con su aptitud. Segundo, es que estos mecanismos toman el rol para determinar que individuo es el sobreviviente, basándose también en la aptitud.
- **Operadores de variación.** La mutación y la cruce nos permiten crear nuevos individuos, por ende se crean nuevas soluciones para un problema dado.
- **Inicialización.** Aunque en la mayoría de los casos la inicialización de la población suele ser de manera aleatoria, también puede ser determinada, en este caso dependerá del problema a resolver.
- **Condición de paro.** Existen diferentes condiciones de paro debido a que los algoritmos evolutivos no garantizan la mejor solución. Bajo un cierto criterio se puede determinar que la solución es óptima o aceptable. Las condiciones de paro más utilizadas son:
 - Tiempo transcurrido en el CPU.
 - El número de evaluaciones llega a un límite específico.
 - Número de generaciones.
 - La diversidad de la población es baja.

2.2.1. Programación genética

La programación genética creada por John R. Koza, ha dado paso a crear programas que crean programas. Enfocándonos en los componentes importantes de los algoritmos evolutivos, a continuación se detallan cada uno de ellos aplicados al paradigma de programación genética.

2.2.2. Representación

En la programación genética los programas son representados de manera típica como árboles. Por ejemplo, en la Figura 2.8 está representado el programa $(2 * y) + ((x + 3) - y)$. Dentro del árbol se encuentran funciones, variables y constantes. Los nodos hoja siempre son constantes o variables. Como se puede observar, en la Figura 2.8 el valor de 2 y 3 son constantes y x y y son variables. Estos, son llamados nodos terminales, mientras que $+$, \times , $-$ son llamados nodos de funciones. El conjunto de funciones

utilizados para la creación de un árbol pueden ser tanto aritméticas como funciones previamente definidas por el programador. Por ejemplo, en el caso particular de este trabajo, funciones para el procesamiento de imágenes son definidas previamente para que sean incluidas al momento de inicializar los árboles.

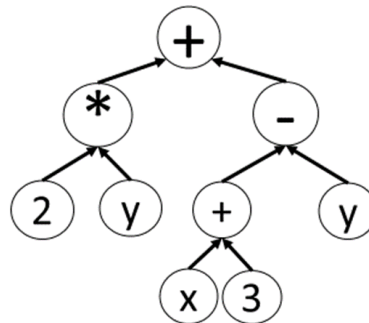


FIGURA 2.8: Representación de un programa en árbol

2.2.3. Inicialización y población

En la programación genética y dada la representación, la manera de generar una población es muy particular. En primera instancia es debido a que nos encontramos con los primeros parámetros de calibración para nuestro algoritmo evolutivo:

- Individuos en la población.
- Máxima profundidad de los árboles generados.
- Mínima profundidad de los árboles generados.
- Método de generación de los árboles.

El número de individuos en la población inicial representa el número de soluciones posibles que se crean al inicio del proceso de búsqueda, esta creación de individuos se realiza de manera aleatoria. La máxima y mínima profundidad de cada árbol que se genera debe ser previamente definida. Para la creación de árboles existen tres métodos. El primer método es *full*, donde las hojas del árbol tienen la misma profundidad, el segundo método es *grow*, el cual permite crear árboles de formas y tamaños variados. Y por último, en [14] se propone la mezcla de los dos anteriores en un método llamado *Ramped half-and-half*, donde la mitad de la población es creada con *full* y la otra

mitad es creada con *grow*. De esta manera se asegura que en la población existan gran variedad de árboles de diferentes tamaños y formas [14].

2.2.4. Mecanismos de selección

Como la mayoría de los algoritmos evolutivos, en programación genética también son utilizados los mecanismos de selección. El mecanismo más utilizado es la selección por torneo. En este mecanismo se realiza un torneo en donde un número de individuos de la población son seleccionados de manera aleatoria para participar en dicho torneo. El tamaño del torneo es otro parámetro que se debe definir, con tamaño me refiero a cuantos individuos son los que participan en el torneo. Para elegir al mejor se realiza una comparación entre cada uno de ellos de acuerdo a su aptitud y el ganador es quién tenga la mejor aptitud, ya sea para ser padre o para pasar a la siguiente generación. Cuando el número de individuos que participan en el torneo es más de 3 el torneo tiende a ser una selección fuerte ya que no es lo mismo competir contra uno a competir con más de dos. En este método de selección los individuos que contienen malas soluciones pueden ser seleccionados, por ejemplo, en una población de 50 individuos, en el torneo entran los 5 individuos menos aptos y uno de ellos es seleccionado.

2.2.5. Operadores de variación

En programación genética son interesantes los operadores aplicados a los árboles, tanto la cruce como la mutación. Para el operador de cruce el método más utilizado es cruce de sub-árbol. Este método se entiende de la siguiente manera. Se tienen dos padres y en este método se escoge un nodo el cuál es llamado punto de cruce. Este punto de cruce es seleccionado en diferentes nodos en ambos padres. Posteriormente se crea el hijo reemplazando el sub-árbol en el nodo que fue seleccionado con el sub-árbol del segundo padre. En la Figura 2.9 se muestra gráficamente el operador de cruce.

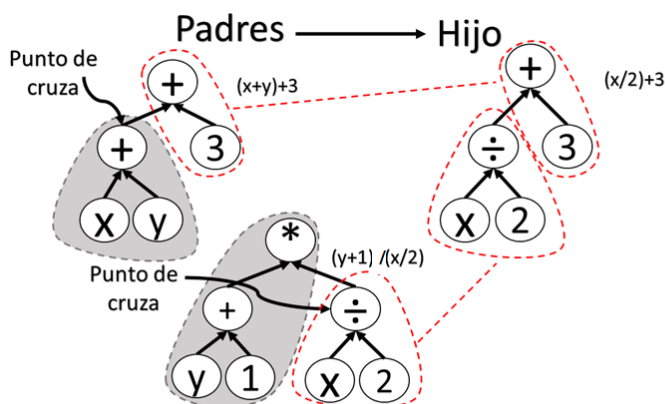


FIGURA 2.9: Representación de la crucea en programación genética

El operador de mutación más utilizado es donde un nodo es elegido de manera aleatoria y un sub-árbol es añadido en ese nodo seleccionado. La operación de mutación se muestra de manera gráfica en la Figura 2.10.

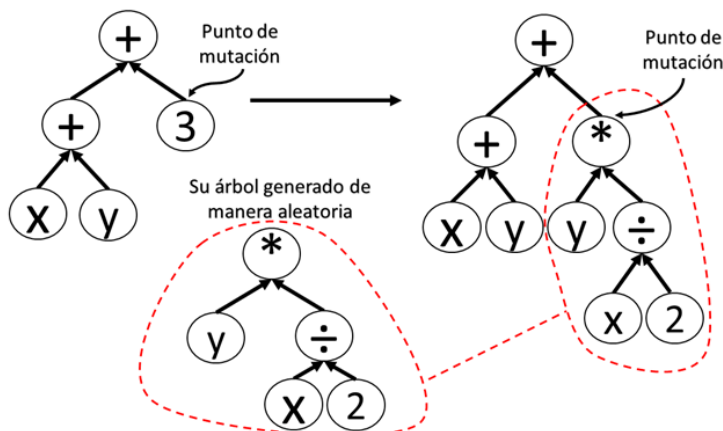


FIGURA 2.10: Representación de la mutación en programación genética

2.2.6. Función de evaluación

La función de evaluación dependerá del problema que se esté resolviendo, esta función es la encargada de asignarle una aptitud a cada uno de los individuos. La aptitud se puede obtener, ya sea mediante métricas utilizadas para evaluar un problema en específico o simplemente con evaluar el error de la función o programa que se esté construyendo. Debido a la naturaleza del paradigma, los individuos pueden ser

programas computacionales y por consiguiente puede llegar a suceder que el tiempo computacional requerido para evaluar una solución tome demasiado tiempo.

2.2.7. Condición de paro

La condición de paro que es más utilizada dentro de programación genética es el número de generaciones transcurridas en el proceso de evolución. En algunas ocasiones, dada la complejidad que pueda a llegar a tener una solución, el criterio de paro es encontrar una aptitud aceptable para la solución del problema planteado.

En la Figura 2.11 se encuentra el diagrama de flujo en general del proceso evolutivo de la programación genética. En dicho proceso, se parte de una población inicial que posteriormente ingresa al proceso de evolución. Entre los individuos de la población se aplican los operadores de cruce y mutación, con el fin de crear nuevos individuos, asegurando así la supervivencia de la especie. En este caso, la descendencia son las soluciones modificadas esperando a que el desempeño mejore de acuerdo al problema a resolver. Posteriormente, se realiza una selección de individuos de la población para continuar el proceso de evolución, dando paso a una nueva generación. Este proceso se repite hasta que el criterio de paro es alcanzado. Como resultado final del proceso evolutivo se obtiene una población diferente a la inicial. Dado que es una búsqueda por una mejor solución, se espera que la población final sea un conjunto de individuos con una mejor aptitud respecto a la población de la generación cero o población inicial.

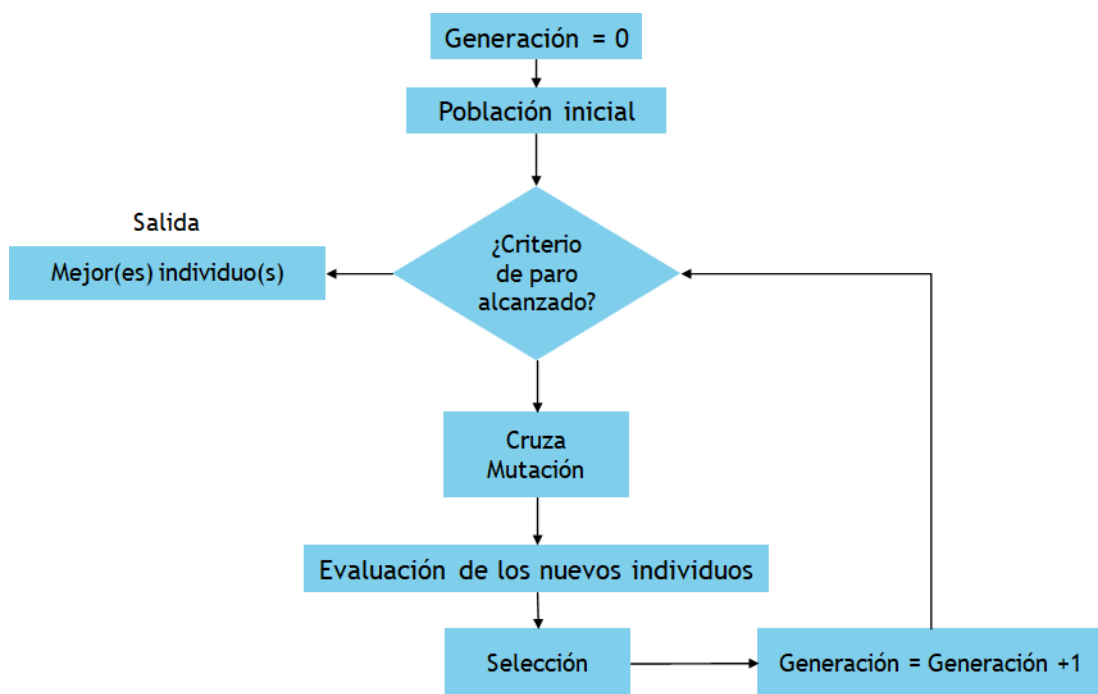


FIGURA 2.11: Diagrama de algoritmo de programación genética [14]

Capítulo 3

Marco referencial

La segmentación de imágenes es una herramienta muy útil independientemente el área en donde sea utilizada. Trabajos realizados atacan el problema de la segmentación de imágenes en diferentes áreas, pero debido a la complejidad que puede llegar a tener este proceso y de acuerdo a cada una de las áreas de aplicación han sido propuestos diferentes métodos y técnicas. Trabajos donde la programación genética se hace presente para realizar segmentación de imágenes han demostrado ser competitivos en diferentes aspectos.

Uno de los primeros trabajos en realizar segmentación de imágenes médicas utilizando programación genética fue realizado por Riccardo Poli [22], el trabajo se centra en imágenes de resonancia magnética en donde la tarea principal es segmentar el cerebro. La representación que utiliza para crear las soluciones es mediante árboles. A juzgar por las funciones utilizadas en su implementación los resultados son competitivos, incluso el método que él propone es comparado con una red neuronal y toma ventaja sobre ella de manera significativa. Prácticamente este trabajo fue el que abrió la brecha de investigación para realizar tareas de procesamiento de imágenes utilizando el paradigma de programación genética. La limitante que tuvo en su momento fue que para evaluar cada una de las soluciones hubo una gran demanda computacional por parte del método propuesto.

Trabajos más actuales como [21] utiliza de igual manera programación genética pero bajo el enfoque lineal. El trabajo se centra en crear operadores de imágenes de manera automática, en este caso crea filtros lineales y no lineales. Estos filtros logran realizar las suficientes operaciones en la imagen de tal manera que se logra realizar

una segmentación. Pedrino solamente utiliza operadores morfológicos en su implementación y estos mismos son modificados hasta encontrar la combinación necesaria para realizar dicha tarea de segmentación. Las imágenes utilizadas no provienen de un área en específico o algún conjunto de datos popular, su método solo es probado con imágenes tomadas de manera aleatoria.

La programación genética se ha ido utilizando de diferentes maneras, en este caso en [25] se realiza una clasificación de imágenes vía multiobjetivo. Lo interesante que se presenta en este trabajo es la manera en como se realiza una clasificación buscando los operadores necesarios para realizar el procesamiento de imagen correcto y posteriormente clasificarla. El conjunto de funciones en la programación genética sobrepasa las 30 funciones. Debido al uso de un número considerable de funciones el otro objetivo que atacan es el tamaño de la solución, pues a mayor número de funciones sean incluidas mayor será el espacio de búsqueda. Las imágenes que se utilizan en este trabajo son conocidas, sin embargo el método propuesto no es probado con imágenes del área médica.

Así mismo, en [26] se realiza una construcción de características para realizar la clasificación de imágenes médicas. Específicamente se clasifican imágenes de lesiones dermatológicas. La construcción de características está basado en utilizar las características de Haralick, con el fin de describir las texturas en las imágenes. Otro trabajo interesante donde realizan una segmentación de objetos es en [6]. En este trabajo se realizan segmentaciones en las imágenes como un paso previo al proceso evolutivo, con el fin de obtener diferentes objetos de cada una de las imágenes. Posteriormente, son introducidas al algoritmo de programación genética y así, de esta manera los objetos salientes que se obtuvieron desde el inicio son mezclados hasta obtener solo el objeto deseado.

Dos trabajos que destacan son [17] y [16]. En ambos trabajos son utilizadas bases de datos de imágenes conocidas como Pascal que contiene imágenes de aviones y Weizmann en la cual se encuentran imágenes de caballos. Ambas bases contienen imágenes a color. En esta aplicación tampoco son incluidas imágenes médicas para verificar el desempeño del método propuesto. Yuyu Liang en [17], bajo un enfoque multiobjetivo propone una función de penalidad para evitar que el tamaño de la solución crezca durante el proceso evolutivo. Esta función de penalidad recibe el nombre de *Penalidad de crecimiento por suma ponderada*. Algo bastante interesante es que no incluyó funciones propias del área de procesamiento de imágenes para realizar la segmentación. En [16] si se incluyen funciones para el procesamiento de imágenes las cuales son operadores

aritméticos básicos y filtros de Gabor, cabe mencionar que fueron 43 filtros utilizados. La manera en como se realiza la segmentación es utilizando clasificadores tradicionales. Lo sobresaliente es que sus resultados tienden a generar soluciones con un número considerable de nodos.

En [6], uno de los trabajos más actuales y con un enfoque interesante utiliza la programación genética para seleccionar diferentes niveles en escala de grises como variables para que posteriormente sean modificadas. Además, se realiza una mezcla con diferentes imágenes resultantes de los operadores que se utilizan dentro de la programación genética y así lograr la detección de objetos sobresalientes dentro de la imagen. El enfoque es comparado con métodos donde un experto es quien realiza la tarea de segmentación para identificar esos objetos en las imágenes. El trabajo no toma en consideración evaluar el método propuesto con imágenes pertenecientes al área médica.

Por último, uno de los trabajos donde se proponen diferentes enfoques para realizar la segmentación es [15]. Los métodos propuestos son evaluados con imágenes en bases de datos como Weizman Y Pascal. Son utilizadas un número considerable de funciones propias del procesamiento de imágenes, como filtros, operaciones morfológicas, entre algunas otras. En el primer enfoque es *CoevoGP*, en el cual se realiza una coevolución donde la población es dividida y diferentes operaciones se utilizan en cada una de las poblaciones generadas. El procesamiento en escala de grises y el de color es separado del procesamiento en imágenes binarias. El segundo enfoque es *TwostageGP*, en el cual el proceso de segmentación es dividido en dos etapas. En la primera etapa se realiza procesamiento en escala de grises y color para que la imagen resultante pase por un entrenamiento y prueba para convertirla en binaria, y la segunda etapa es modificar esa imagen resultante y volver a evaluar después de haberle aplicado operadores morfológicos donde ahora solo son aplicadas en las imágenes binarias. Por último, el enfoque llamado *StronglyGP*, aquí las funciones reciben un tipo de dato específico. Por ejemplo, no permiten el paso a imágenes que se encuentran en escala de grises o color a operaciones o funciones que solo trabajan con imágenes binarias. Este último enfoque demuestra competitividad respecto con los otros métodos propuestos. Específicamente este enfoque se utiliza como método de comparación para la propuesta planteada en este trabajo.

En la mayoría de los trabajos no se toman en consideración imágenes pertenecientes al área médica para evaluar el método o técnica que se está proponiendo, dando pauta a que el área es poco explorada para la programación genética. Además, es

notable que en algunos trabajos el uso de funciones es excesivo, creando así un modelo complejo que probablemente sea computacionalmente costoso. Por otro lado, el problema de segmentación en imágenes afectados por COVID-19 es, hasta ahora, atacado mayormente por redes neuronales. El trabajo se centra en obtener un modelo menos complejo respecto al estado del arte para llevar a cabo la construcción de características y la segmentación del conjunto de imágenes. Además, se evita el uso excesivo de funciones, pues en un determinado momento el exceso de funciones puede llegar a generar un sesgo al proceso de búsqueda para una óptima solución.

Capítulo 4

Propuesta

Tomando en consideración la importancia de detectar el daño causado por COVID-19 en los pulmones de los pacientes mediante el uso de tomografías computarizadas, en esta tesis se propone el uso de programación genética, con la cual se construye un programa computacional de manera automática. Este programa computacional se encarga de construir las características necesarias de las imágenes para realizar una segmentación competitiva y por ende, determinar el área afectada por el virus.

Como fue mencionado con anterioridad, el paradigma de programación genética ofrece la posibilidad de crear programas computacionales mediante la representación de árboles. Además, permite al programador implementar sus propias funciones de acuerdo al problema a resolver. Para el caso de este trabajo, las funciones que se implementan son funciones que realizan las operaciones de procesamiento de imágenes previamente mencionadas en las secciones 2.1.3 y 2.1.4. La programación genética toma ventaja gracias a la representación de árbol, pues como resultado final del proceso evolutivo se obtiene claridad de lo que se está realizando para llegar a la solución final, es decir, en la solución final se muestra el conjunto de funciones que son utilizadas y el orden en que fueron aplicadas para llegar a la imagen resultante. Queda claro que los programas o funciones para realizar el procesamiento de imágenes modifican la información contenida en la imagen, por ende, los valores numéricos contenidos en las matrices son modificados, esto da pie a que la percepción visual de lo que está contenido en las imágenes cambie, destacando una característica o área en específico dentro de la imagen.

4.1. GP-IFC

El algoritmo propuesto es GP-IFC (Genetic Programming - Image Feature Construction), el cual está basado en programación genética. En dicho algoritmo se realiza la construcción automática de un programa computacional en donde están involucrados programas computacionales para realizar un procesamiento de imágenes. Como se mencionó anteriormente, el algoritmo evolutivo base utiliza los parámetros de población (número de individuos), el número de generaciones, los operadores de cruce, mutación y selección, así como el método de inicialización de los árboles en la población inicial y por último, la profundidades mínima y máxima en los árboles (individuos) generados. Además, es necesario el criterio de paro, pues con este el proceso de evolución se detiene en el momento que se llega a un número de generaciones determinado.

El algoritmo está diseñado como una herramienta para realizar la construcción de características y con ellas poder realizar una segmentación a un conjunto de imágenes. Para utilizar esta herramienta es necesario tener la imagen original y la imagen a la cual se requiere llegar. Ésta imagen objetivo es una imagen binaria en donde se muestra el área segmentada por el experto. La imagen segmentada por el experto también es conocida como Ground Truth (GT). Cabe mencionar que no es necesario que las imágenes tengan una dimensión en específico o que todas las imágenes tengan las mismas dimensiones. Además, si el conjunto de imágenes son a color, estas se convierten a imágenes en escala de grises, pues la propuesta trabaja con este tipo de imágenes.

Algoritmo 1: GP-IFC

Entrada: Conjunto de imágenes (Reales y GT).

Salida: Función constructora de características

Inicialización de población;

Evaluación de la población;

mientras $Generaciones < Max\ Generaciones$ **hacer**

Selección de padres;

Creación de nuevos individuos mediante los operadores de cruce y mutación;

Evaluación de los nuevos individuos;

Selección de individuos sobrevivientes para la siguiente generación;

fin

El Algoritmo 1 muestra la implementación de la propuesta. El algoritmo recibe como entrada el conjunto de imágenes, el cual debe contener la imagen real y su correspondiente imagen objetivo.

Antes de comenzar el proceso de evolución los individuos de la población son generados de manera aleatoria. Después de que hayan sido creados, se evalúan mediante la función de aptitud, es decir, cada individuo (programa) se ejecuta para realizar la construcción de características de las imágenes, y mediante la métrica seleccionada cada individuo obtiene una aptitud. Una vez que los individuos son evaluados el proceso de evolución comienza, y este termina hasta que el criterio de paro sea alcanzado. Durante el proceso de evolución se realiza una selección de padres en cada generación. Esta consta en seleccionar individuos dentro de la población para que puedan crear una descendencia. De acuerdo con los parámetros de cruce y mutación estos nuevos individuos son creados y cada uno de ellos contendrá características similares a sus padres. Estos nuevos individuos son llamados hijos o descendencia. De la misma manera que los padres, los descendientes son evaluados para asignarles su aptitud. Después, estos nuevos individuos son agregados a la población. Así, posteriormente se realiza de nuevo una selección de individuos con el fin de seleccionar a los sobrevivientes para la siguiente generación. Esta selección se realiza con el objetivo de mantener fijo el tamaño de la población durante todo el proceso de evolución. Este algoritmo evolutivo, en donde los nuevos individuos son agregados a la población general y a partir de ahí se realiza la selección para la siguiente generación se le conoce como $(\mu + \lambda)$, donde μ es el número de individuos a seleccionar para la siguiente generación y λ es el número de hijos a producir en cada generación.

4.1.1. Configuración de GP-IFC

En la sección 2.2.1 fueron descritos los principales componentes de la programación genética. A continuación, se describe la configuración de ellos. Cabe mencionar que durante la experimentación los parámetros de población, el conjunto de funciones, el número de generaciones y la función de aptitud fueron cambiando de acuerdo al comportamiento del algoritmo y respecto a la solución que se estaba obteniendo.

4.1.1.1. Representación

Cada uno de los individuos son representados mediante árboles. Por ejemplo, en la Figura 4.1, en los nodos internos (color azul) se encuentran los programas que se encargan de realizar el procesamiento de imágenes y en los nodos hoja (color verde) se encuentra la imagen original en escala de gris, sin ninguna modificación.

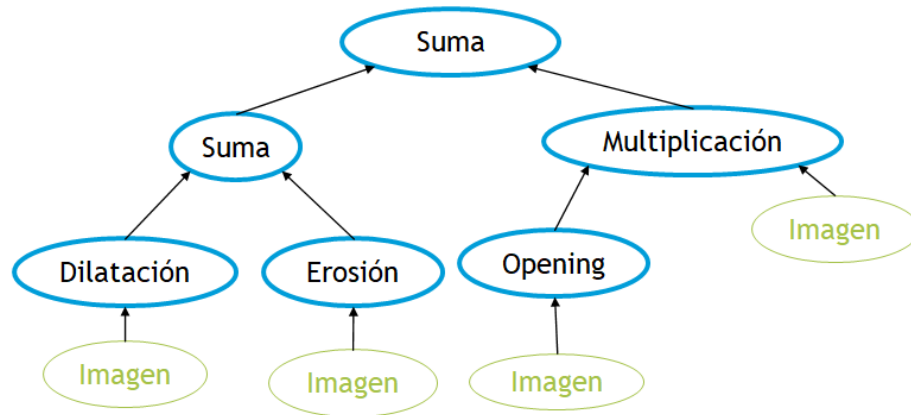


FIGURA 4.1: Ejemplo de representación

4.1.1.2. Inicialización, población y condición de paro

La población inicial es generada de manera aleatoria. Los árboles son creados mediante el método *Full*. De manera inicial, la profundidad mínima y máxima fueron configurados en 2 y 7 respectivamente. La condición de paro del proceso evolutivo para esta propuesta es mediante el número de generaciones. De manera inicial, el número de generaciones es configurado en 200.

4.1.1.3. Mecanismos de selección y operadores de variación

El proceso de selección de padres y la selección de sobrevivientes durante el proceso evolutivo es mediante el mecanismo de selección de torneo. El tamaño del torneo es configurado a 5 individuos, por ende es una selección fuerte.

El operador de variación de cruza es cruza de sub-árbol y la mutación es mediante la selección aleatoria de un nodo, ambos operadores se encuentran descritos en la sección 2.2.5. En programación genética existe un fenómeno llamado *Bloat*. Este fenómeno ocurre en el proceso de evolución cuando son aplicados los operadores de mutación y de cruza. Existe un momento en que los nodos seleccionados para realizar la cruza o la mutación son los de hasta abajo, y el árbol que es insertado en ese nodo tiene una cierta profundidad. Con el paso de las generaciones el individuo tiende a crecer, creando un individuo con más de 30 nodos de profundidad. Para evitar este fenómeno, es configurado un control de profundidad con el fin de tener el control y limitar el tamaño de la solución generada. Este parámetro se configura con base en lo propuesto por Koza [14].

4.1.1.4. Función de evaluación

La métrica utilizada para evaluar la aptitud de cada uno de los individuos de la población es cambiada durante la experimentación, pues parte de la metodología es determinar cuál métrica guía de manera eficiente el proceso de búsqueda. Las métricas utilizadas en la experimentación están descritas en la sección 2.1.6.

4.1.1.5. Conjunto de funciones

Las funciones o programas computacionales para realizar el procesamiento de imágenes están descritas a detalle en la sección 2.1.

En el cuadro 4.1 se encuentran el conjunto de funciones utilizadas para los nodos internos y nodos terminales para la creación de arboles. Cabe destacar que el tipo de entrada en cada una de las funciones es una imagen en 2-D, es decir, es una imagen en escala de gris.

CUADRO 4.1: Conjunto de funciones iniciales.

Nombre de función	Definición
Suma	$(x + y)$
Resta	$(x - y)$
Multiplicación	$(x \times y)$
División(protegida)	$(x \div y)$
Erosión	Operación morfológica de erosión
Dilatación	Operación morfológica de dilatación
Opening	Operación morfológica de <i>opening</i>
Closing	Operación morfológica de <i>closing</i>
Máscara/Umbral	Operación pixel
Terminales	
Im	Imagen en 2-D

Función Máscara/umbral

La función propuesta realiza el procesamiento en la imagen de la siguiente manera. Recibe como entrada la imagen en escala de gris. Después, se selecciona un número entre 0 y 255 de manera aleatoria al cual se le llama umbral. Este número se comparara

a nivel píxel con la imagen de entrada, es decir, en la posición $I(x,y)$ se verifica si el valor es menor que el umbral. La matriz máscara tiene el mismo tamaño que la imagen de entrada y es en donde son asignados los valores de **Verdadero** y **Falso** de acuerdo a la condición de umbralización. Posteriormente, se realiza la operación **AND** entre la imagen de entrada con la máscara. En la posición en donde se encuentre un **Verdadero** en la matriz de máscara el valor de la imagen de entrada no cambia. Por el contrario, si se encuentra con un **Falso**, el valor de ese píxel es igual a cero. El Algoritmo 2 muestra de manera resumida el proceso.

Algoritmo 2: Función máscara/umbral

Entrada: Imagen en escala de grises

Salida: Imagen es escala de grises

umbral = random(0,255) ;

mascara = Imagen_entrada <= umbral;

imagen_salida = Imagen_entrada(x,y) **AND** mascara

Para clarificar el funcionamiento de la función propuesta en la Figura 4.2 se encuentra el ejemplo de la operación que se realiza.

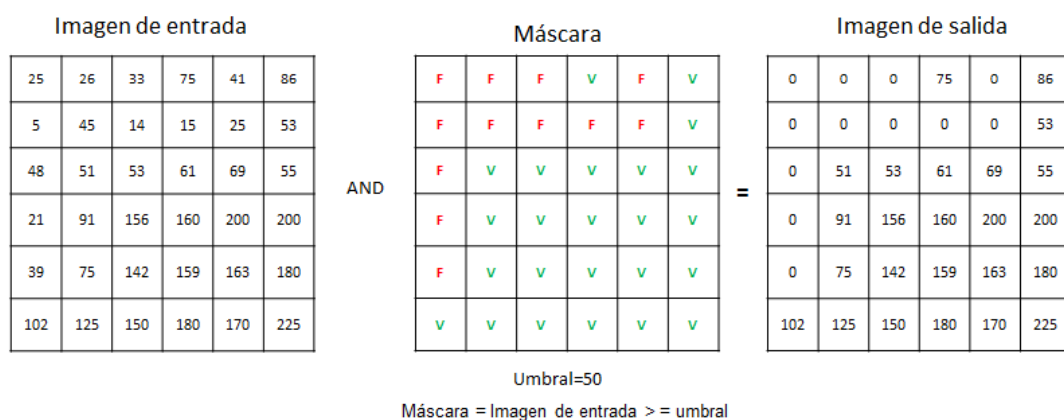


FIGURA 4.2: Función máscara.

4.1.2. Segmentación

Como fue mencionado en la sección 2.1.5, el proceso de segmentación puede realizarse de diferentes maneras. Recordando que el resultado o salida de la programación genética es un programa el cuál contiene funciones para construir características significativas de la imagen. En este proceso se busca alcanzar un nivel de similitud mediante las métricas. Durante la experimentación se utilizan diferentes métodos para realizar

la umbralización de las imágenes resultantes para pasar de imágenes en escala de gris a imágenes binarias. Al umbral se le asigna el valor de 128, pues recordando las imágenes con las que se está trabajando se encuentran en el rango de $[0,255]$. De esta manera los valores de los píxeles se comparan con el valor del umbral, y si son menores o iguales a 128 ese píxel toma el valor de 0, en caso contrario el valor del píxel toma el valor de 1. El valor de 0 representa el área que no es de interés en la imagen y el 1 representa el área de interés. Otro método utilizado es el método de Otsu [19], el cuál selecciona el umbral de manera automática partiendo de un histograma de niveles de gris de la imagen. Por último, se propone el uso de clasificadores para realizar la segmentación. El problema de segmentación es atacado por aprendizaje automático, pues en términos generales se realiza una clasificación a nivel píxel, el cual contiene 2 clases, el área de interés y el área que no es de interés. Los clasificadores utilizados para realizar la segmentación son ID3 y SVM (Support Vector Machine).

4.2. Resumen

En este capítulo se mostraron detalles del algoritmo propuesto para realizar la construcción de características de manera automática. Se detalló la configuración y las características principales del algoritmo como lo son la representación utilizada, el método de inicialización de la población, el criterio de paro y los mecanismos de selección utilizados. De la misma manera, se detallan las funciones que se encuentran de manera inicial en el conjunto de funciones, así como la función máscara propuesta. Además, se especifican los métodos de segmentación/binarización tomados en consideración para la parte de experimentación con el fin de determinar cuál es el que mejor lo realiza. Así mismo, se detallan cuáles métricas son candidatas para la función de aptitud del algoritmo.

Capítulo 5

Experimentos y resultados

La experimentación de la propuesta se divide en cuatro etapas, pues en cada una de ellas se muestran resultados de los experimentos que van mejorando de manera gradual. Los experimentos realizados se enfocan en evaluar el desempeño de la propuesta en usar un número moderado de funciones de procesamiento de imágenes así como la función de máscara propuesta.

El conjunto de imágenes con los que se trabajan fueron tomados de diferentes repositorios. Las bases de datos son *COVID-19* [2], *Skin lesion* [4], *Retine Vessel* [3] y *Brain CT images with Intracranial Hemorrhage* [5]. Todas las bases de datos contienen la imagen original y su respectiva imagen binaria previamente segmentada por el experto (GT).

En todos los experimentos se utilizan la base de datos de *COVID-19* [2]. En los experimentos 3 y 4 se utilizan las otras 3 bases de datos previamente mencionadas. Para la parte experimental los conjuntos de imágenes que se utiliza son divididos en entrenamiento y validación. De cada repositorio son tomadas 100 imágenes con sus respectivas imágenes objetivo, a excepción de *Retine Vessel* [3], pues solo contiene un total de 20 imágenes con sus respectivas imágenes objetivo. En la etapa de entrenamiento se utilizan el 70 % de las imágenes para crear la función constructora de características y el 30 % de imágenes restantes son utilizadas para verificar y validar el desempeño del algoritmo. Cabe mencionar que los parámetros del método propuesto no fueron calibrados mediante bibliotecas que realizan de manera automática el ajuste de parámetros, esto es debido a que el tiempo que toma una ejecución del algoritmo se encuentra dentro del intervalo que va de 12 a 15 horas. Debido a la demanda de tiempo

que requiere el algoritmo, el ajuste de parámetros fue realizado de manera experimental. Dicho ajuste constó en realizar pruebas con diferentes valores en los parámetros con el fin de determinar cuáles conducían a un mejor desempeño.

La implementación de los experimentos propuestos en este trabajo están realizados en el lenguaje de programación Python 3. Además, el algoritmo de programación genética se realizó utilizando la librería DEAP [13]. Así mismo, son utilizadas las librerías Skimage [29] y OpenCV [7] para realizar el procesamiento de imágenes, también se utilizó Sklearn [20] como herramienta, pues contiene diferentes algoritmos de aprendizaje supervisado, los cuales son utilizados en la cuarta etapa de experimentación. El equipo utilizado para realizar los experimentos 1, 2 y 3 consta de un procesador Intel Core i7-8750H @2.20GHZ y memoria RAM de 16GB en el Sistema Operativo Windows 10. Para el experimento 4 se realizan las pruebas en un servidor con 64 procesadores Intel Xeon Silver 4216 @2.10GHZ, memoria RAM de 264GB y sistema operativo LINUX.

5.1. Experimento 1

5.1.1. Diseño Experimental

En la primera etapa de experimentación se realiza la calibración de parámetros para la programación genética. Como se menciona en 4.1.1, los parámetros que se configuran son el tamaño de la población, el número de generaciones, probabilidades de los operadores de mutación y cruza, así como el método de selección de individuos y la configuración inicial para la generación de los individuos. De la misma manera el conjunto de funciones utilizadas se encuentra en el Cuadro 4.1, pues estas fueron las funciones que se plantearon en un inicio.

Por otro lado, la segmentación se realizó mediante la configuración de un umbral. Este umbral fue configurado con un valor de 128. Esto quiere decir que en la imagen los píxeles que tienen un valor menor o igual a 128 (recordando que las imágenes resultantes van de 0 a 255) toman el valor de 0 y en caso contrario, el valor que toma el píxel es de 1. De esta manera, se obtiene como resultado la imagen binaria para que sea comparada con la imagen objetivo y así poder realizar la evaluación de la segmentación.

5.1.1.1. Parámetros de PG

Los parámetros de probabilidad de cruza, mutación y selección en esta etapa de experimentación son tomados de [14]. En cuanto a profundidad máxima y mínima fueron tomados de [15]. En cuanto a población y número de generaciones se determinaron tras ejecutar 5 veces el algoritmo probando con diferentes valores. Para determinar cuáles parámetros seleccionar tras las ejecuciones del algoritmo se realizó una comparación entre los resultados obtenidos. Cabe destacar que solo se realizaron 5 ejecuciones por cada uno de los parámetros debido al tiempo que requiere una ejecución.

5.1.1.2. Función de aptitud

Uno de los objetivos de esta primera etapa de experimentación es determinar cuál función de aptitud es la que mejor guía el proceso de búsqueda, pues como se mencionó en 2.1.6, las métricas utilizadas para cuantificar el desempeño de segmentación en la imagen son *DICE*, *Sensitividad*, *Especificidad* y *Hausdorff Distance*.

5.1.2. Discusión y resultados

A partir de las ejecuciones realizadas para obtener y seleccionar los parámetros adecuados se obtuvieron los valores mostrados en el Cuadro 5.2. Además, en esta etapa el algoritmo fue guiado con la métrica DICE tomando como referencia [15].

El primer parámetro que se obtuvo fue el tamaño de la población, pues fue clave en un inicio debido al tiempo computacional que requiere evaluar cada uno de los individuos. El mejor parámetro para el número de individuos en la población fue de 100 individuos. Aunque una mayor población representa una mayor exploración en el espacio de búsqueda, para este caso bastó un población de 100 individuos.

Se realizó la prueba no paramétrica Krustal - Waillis con el fin de realizar una comparación entre los valores alcanzados entre cada una de las configuraciones del tamaño de población. La prueba se ajustó con un nivel de confianza de 95 %. El resultado de la prueba no paramétrica arroja un valor de $\rho = 0,08046$, descartando la existencia de una diferencia significativa entre cada configuración de población probada. Debido a este resultado, se opta por configurar el parámetro de población a 100, pues beneficia

al proceso de evolución en cuestión de tiempo. Además, en las cinco ejecuciones realizadas con esta configuración de población con 100 individuos, se encuentra la solución más alta respecto a las demás.

Posterior a obtener la población fija, se realizaron las pruebas del algoritmo para encontrar el número de generaciones necesario con el fin de obtener un buen desempeño. Siguiendo los mismos pasos para obtener el tamaño de la población se obtuvo el número de generaciones para el proceso evolutivo. Aplicando la misma prueba no paramétrica y el mismo nivel de confianza al conjunto de resultados obtenidos de las 5 ejecuciones, se obtiene un valor de $\rho = 0,48191$. El resultado muestra que tampoco existe una diferencia significativa. Sin embargo, con el parámetro configurado a 200 generaciones se alcanzó un mejor desempeño. En esta primera etapa, se elige este parámetro como el adecuado, pues también se toma en consideración el comportamiento de la solución que se está generando. Establecer un considerable número de generaciones permite al algoritmo seguir buscando soluciones que tengan el mismo desempeño pero con una estructura de árbol más pequeña, es decir que menos programas o funciones sean ejecutados en la solución.

CUADRO 5.1: Resultados obtenidos para la configuración de parámetros.

Estadística (DICE)	Parámetros					
	Población			Generaciones		
	100	150	200	100	150	200
Mejor	0.51221	0.49336	0.45140	0.49995	0.50256	0.51002
Peor	0.45070	0.36565	0.35801	0.35151	0.35465	0.39545
Media	0.48046	0.43089	0.41920	0.42346	0.43388	0.46339
Mediana	0.48874	0.41515	0.42333	0.40231	0.43163	0.46874
Desv. Std.	0.02822	0.05631	0.03786	0.05845	0.05629	0.04608

CUADRO 5.2: Parámetros para ejecución de PG.

Parámetro	Valor	Parámetro	Valor
Población	100	P. de cruza	0.8
Generaciones	200	P. de mutación	0.2
Profundidad mínima	2	Profundidad máxima	7
μ	100	λ	Población/2
Tipo de selección	Torneo (5)	Inicialización	Full

Teniendo ya definido el conjunto de parámetros para el algoritmo, se procedió a realizar pruebas para determinar cuál métrica es la que tiene un mejor desempeño para guiar la búsqueda. Cabe mencionar que el análisis para determinar cuál métrica es la adecuada, se toma en consideración la percepción visual de las imágenes resultantes. Esto es debido a que las métricas penalizan de manera diferente los errores y los aciertos. Para elegir la métrica se realizaron 5 ejecuciones para cada una de ellas, los parámetros utilizados fueron los ya encontrados previamente, mostrados en el Cuadro 5.2.

5.1.2.1. DICE

Dado que esta métrica es una de las más utilizadas para evaluar la segmentación, era de esperarse que las imágenes resultantes destacaran por la percepción visual al compararlas con las imágenes objetivo. Esto es debido a que la métrica involucra la precisión y la sensibilidad generando un balance entre los píxeles correctamente asignados a su valor de acuerdo con la imagen objetivo. A continuación, en la Figura 5.1 se muestra el comportamiento del algoritmo de la mejor corrida de las 5 realizadas.

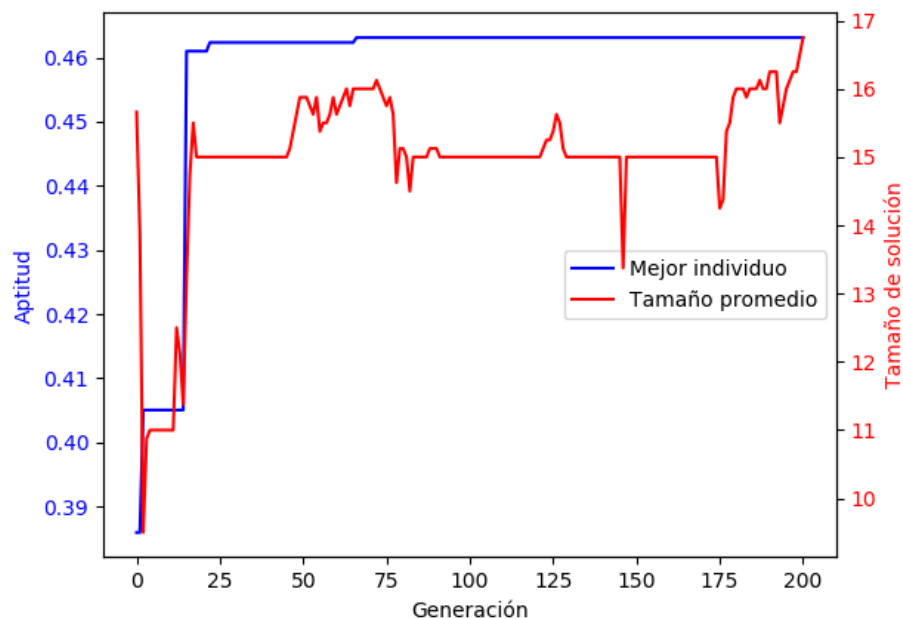


FIGURA 5.1: Gráfica de convergencia de la mejor ejecución del algoritmo utilizando la métrica DICE.

Como puede observarse existe una convergencia prematura en el proceso de búsqueda, pues valores máximos son alcanzados en las primeras 50 generaciones. En las generaciones posteriores existe un mejoramiento en aproximadamente en la generación 70 - 75 y a partir de ahí se mantiene estancado. Cabe mencionar que el comportamiento respecto al tamaño promedio en los individuos se mantiene estable, sin embargo en el último cuarto del proceso evolutivo se muestra que existe un crecimiento promedio del tamaño de la soluciones que se encuentran en la población.

La convergencia prematura que ocurre durante el proceso motiva a agregar funciones y realizar cambios en el conjunto de parámetros del algoritmo. Estos cambios se ven reflejados en los experimentos 2, 3 y 4.

En la Figura 5.2 se muestra una de las imágenes resultantes después de utilizar esta métrica en el proceso de búsqueda. La percepción visual para determinar el área afectada de la imagen binarizada en comparación con la imagen objetivo es favorable, pues comparando visualmente ambas imágenes tienen un cierto grado de similitud, aunque la métrica obtenga un valor no tan alto (menor a 0.6) se logra identificar un área en específico.

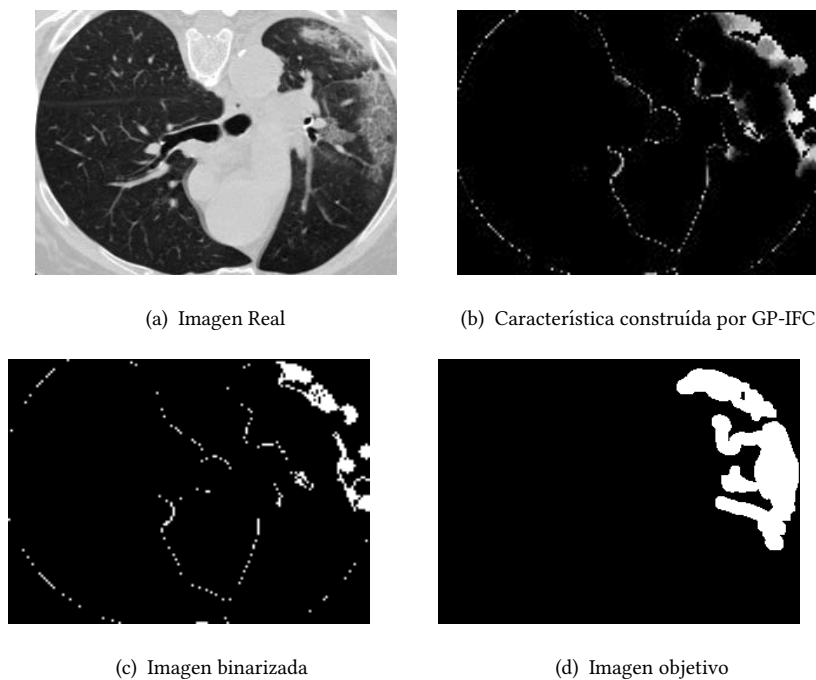


FIGURA 5.2: Ejemplo de imagen resultante de GP-IFC utilizando la métrica DICE como función de aptitud.

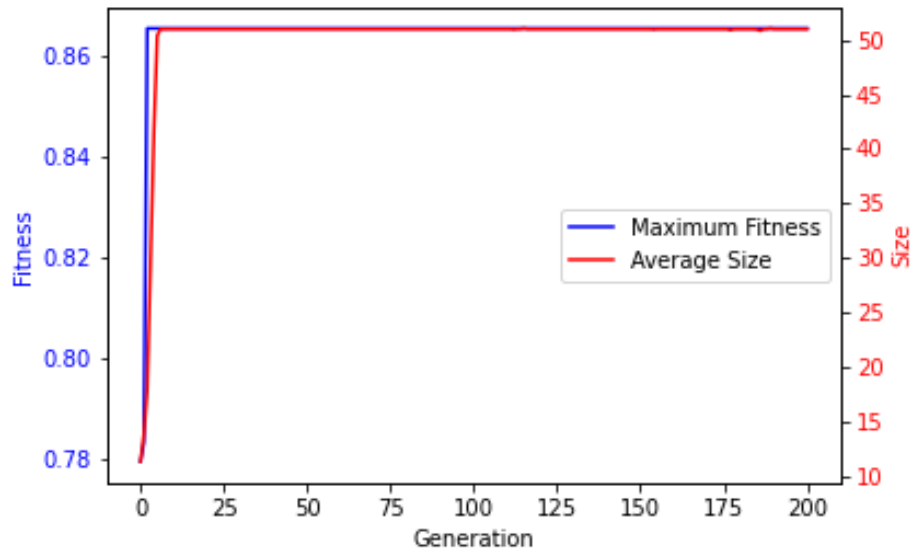


FIGURA 5.3: Gráfica de convergencia de la mejor ejecución del algoritmo utilizando la métrica Sensitividad.

5.1.2.2. Sensitividad

Los resultados obtenidos con esta métrica son bastante interesantes, pues a diferencia de Especificidad y Hausdorff Distance esta métrica logra definir área de los pulmones y se logra apreciar en la imagen. Aunque la solución en la imagen no muestra el área objetivo, se logra identificar los pulmones a la hora de realizar la binarización de la característica construida.

El comportamiento de la búsqueda utilizando esta métrica como función de aptitud mostrada en la Figura 5.3 destaca de manera similar a DICE y Especificidad una convergencia prematura. Por otro lado el comportamiento del promedio del tamaño de las soluciones en la población incrementa de manera significativa. Pues comparando con las otras métricas el tamaño promedio de las soluciones se triplica.

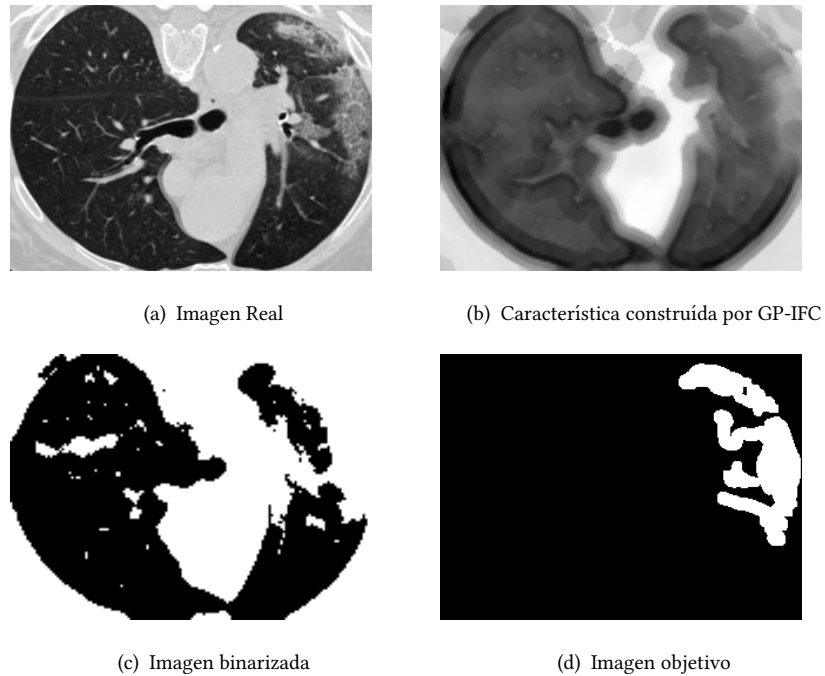


FIGURA 5.4: Ejemplo de imagen resultante de GP-IFC utilizando la métrica Sensitividad como función de aptitud.

5.1.2.3. Especificidad

Esta métrica demostró un desempeño bastante deficiente en cuanto a la percepción visual de la imagen resultante, pues la imagen prácticamente la volvió negra. Claramente el área que se requiere trabajar e identificar no es perceptible. Comparando de manera visual la imagen resultante con la imagen objetivo queda completamente descartada esta métrica, pues cuando se realiza la binarización no se determina un área en específico. En esta métrica influye el valor de los verdaderos negativos, es decir, los píxeles correctamente puestos en oscuro son los que sesgan el resultado de esta métrica, pues entre más oscura esté la imagen mejor valor obtendrá.

En la Figura 5.5 se muestra el comportamiento de la ejecución del algoritmo que obtuvo el mejor desempeño.

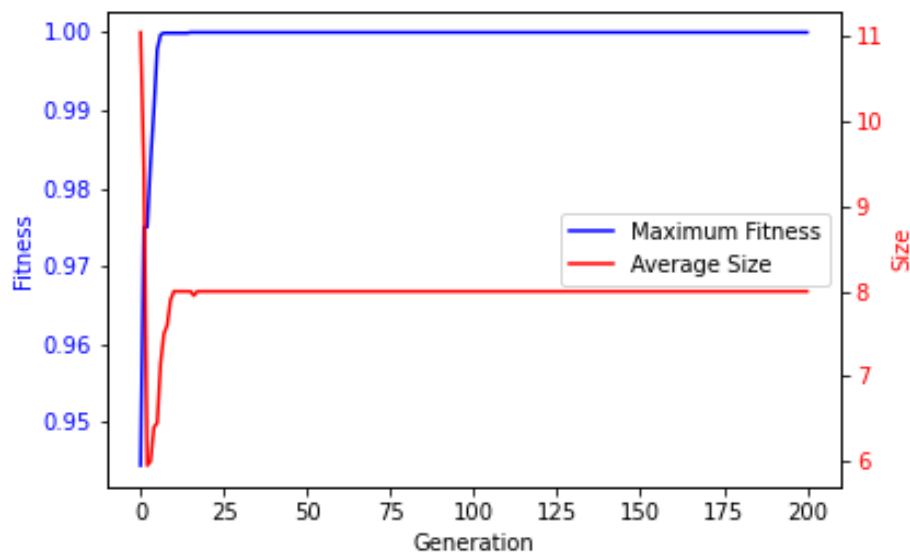


FIGURA 5.5: Gráfica de convergencia de la mejor ejecución del algoritmo utilizando la métrica Especificidad.

En la gráfica se puede observar que existe una convergencia prematura en proceso de evolución. Alcanza el valor máximo antes de la generación número 25 del proceso. Aunque la métrica determina que se alcanza un valor mayor a 0.98 las imágenes resultantes muestran un resultado visual deficiente. Una de las imágenes resultantes se observa en la Figura 5.6

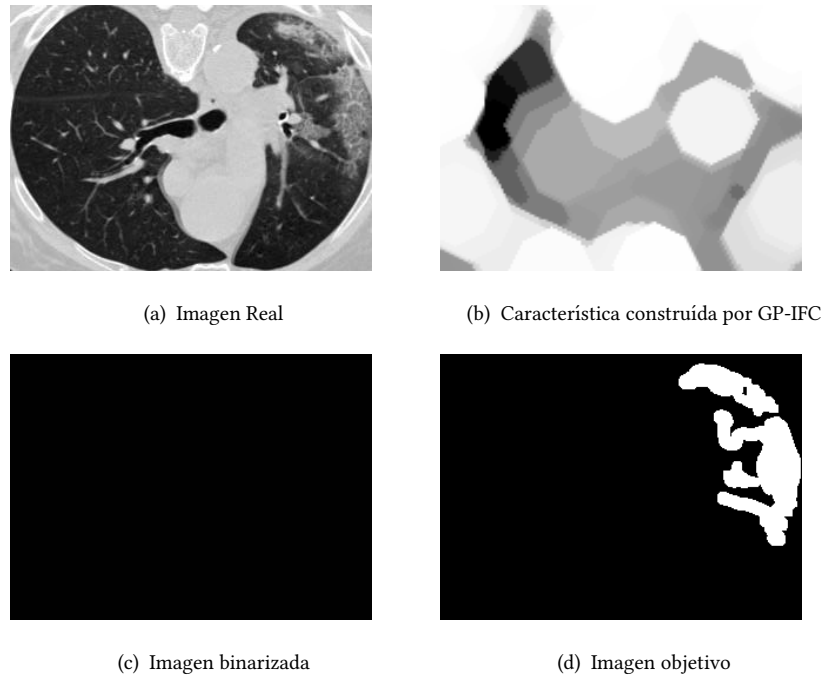


FIGURA 5.6: Ejemplo de imagen resultante de GP-IFC utilizando la métrica Especificidad como función de aptitud.

5.1.2.4. Hausdorff Distance

Al utilizar esta métrica se debe tomar en consideración que entre menor distancia existe mejor percepción visual de la imagen resultante debe existir. Es por ello que para esta métrica se debe realizar una minimización. Los resultados al utilizarla no fueron favorables debido a que la imagen resultante no ofrece una percepción visual clara del área afectada.

En la Figura 5.7 se muestra el comportamiento que tuvo la mejor corrida del algoritmo. Utilizando esta métrica no se observó una convergencia prematura, por ende el proceso evolutivo en cuanto a exploración de posibles soluciones tiene un comportamiento adecuado para la búsqueda. De la misma manera, conforme avanza el proceso evolutivo se observa que el promedio del tamaño de las soluciones en la población se mantiene estable y además disminuye conforme avanza el proceso de evolución.

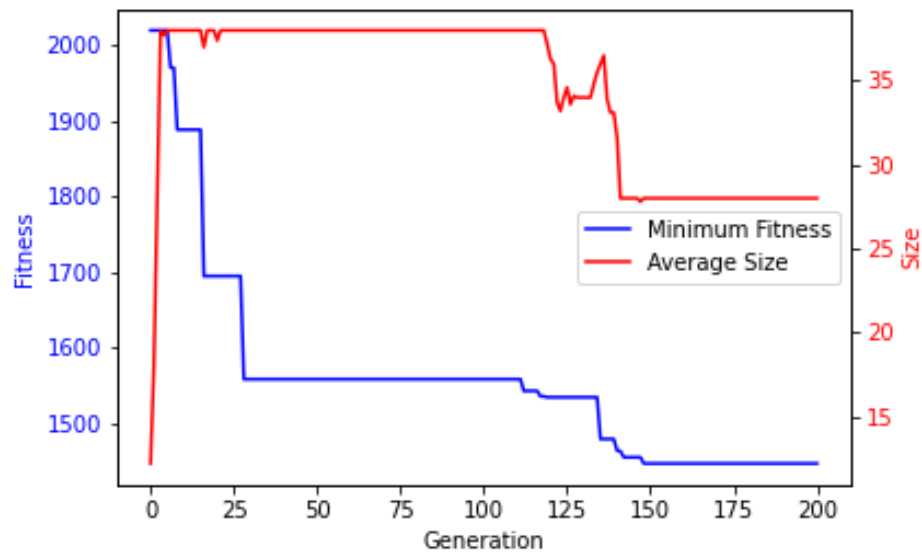
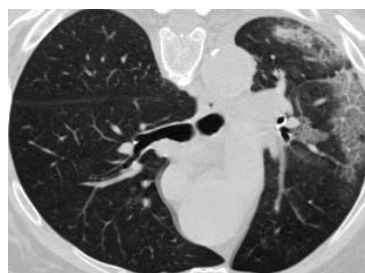
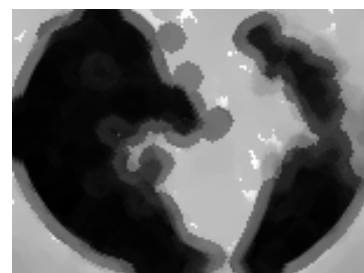


FIGURA 5.7: Gráfica de convergencia de la mejor ejecución del algoritmo utilizando la métrica Hausdorff Distance.

En la Figura 5.8, se muestra una de las imágenes resultantes después de utilizar esta métrica en el proceso de búsqueda. Como se puede observar, la imagen binarizada no muestra un área de interés y queda completamente descartada el área objetivo.



(a) Imagen Real



(b) Característica construída por GP-IFC



(c) Imagen binarizada



(d) Imagen objetivo

FIGURA 5.8: Ejemplo de imagen resultante de GP-IFC utilizando la métrica Hausdorff Distance como función de aptitud.

Tras haber determinado los parámetros para el proceso de búsqueda así como la métrica que mejor desempeño tiene para identificar el área objetivo en las imágenes, se realizaron 10 ejecuciones posteriores obteniendo las estadísticas mostradas en el Cuadro 5.3.

Ejecución	Aptitud	Tamaño de solución
1	0.45412 ± 0.25	6.13 ± 0.73
2	0.37047 ± 0.16	9.13 ± 5.73
3	0.46636 ± 0.11	4.06 ± 1.253
4	0.53227 ± 0.10	14.97 ± 0.50
5	0.5912 ± 0.13	7.54 ± 0.77
6	0.45859 ± 0.25	6.13 ± 0.73
7	0.38749 ± 0.11	18.19 ± 1.84
8	0.34591 ± 0.11	21.16 ± 2.69
9	0.41363 ± 0.13	6.13 ± 0.73
10	0.51022 ± 0.13	6.20 ± 1.66

CUADRO 5.3: Resultados de las 10 ejecuciones utilizando los parámetros y métrica determinadas.

En la Figura 5.9 se muestra la gráfica del comportamiento de la ejecución 5. Como se puede observar, a pesar de haber configurado los parámetros se continua teniendo una convergencia prematura y un estancamiento en la solución encontrada, pues tras el paso de las generaciones no existe algún cambio o incremento en la aptitud de algún individuo de la población. Por otro lado, se observó que las funciones de multiplicación y división no aportan en la construcción de la característica, pues como resultado la imagen se volvía completamente negra o blanca, realizando un sesgo en las imágenes resultantes, además que en las soluciones mejores obtenidas no eran tomadas en consideración.

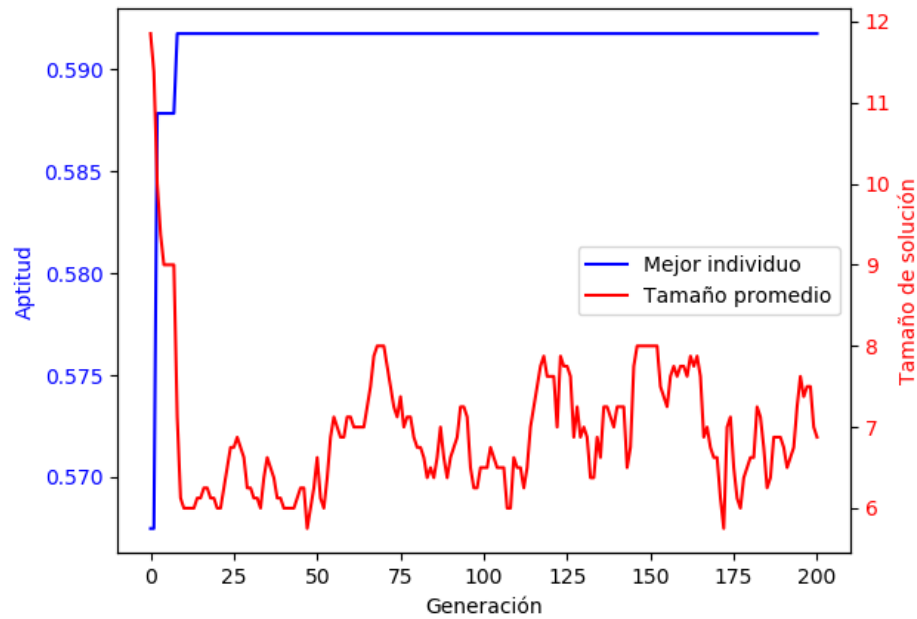


FIGURA 5.9: Gráfica de convergencia de la Ejecución 5.

5.1.3. Resumen

En esta etapa de experimentación se definieron los parámetros adecuados para la configuración de algoritmo. Además, se determinó cuál métrica es la que guía mejor el proceso de evolución y que a su vez los resultados en la imagen fueran favorables para poder determinar el área de interés. La métrica DICE es la más adecuada y se determina como función de aptitud. Quedan descartadas las métricas de Hausdorff Distance, Especificidad y Sensitividad. Por otro lado, el comportamiento del algoritmo en la búsqueda de la solución tiene un convergencia prematura. Dado este comportamiento se planteó la modificación de algunos parámetros en el algoritmo con el fin de obtener un mejor desempeño en la métrica seleccionada, así como generar un comportamiento diferente en el proceso de búsqueda. Además, se planteó agregar funciones al banco de funciones así como se descartaron aquellas que no aportan en la construcción de la característica.

5.2. Experimento 2

5.2.1. Diseño experimental

En esta segunda etapa de experimentación ya se tienen definidos los parámetros adecuados así como la función de aptitud a utilizar para guiar la búsqueda. Dados los resultados obtenidos en la primera etapa de experimentación se procede a realizar cambios con el fin de mejorar el desempeño y a su vez mejorar de manera visual la identificación del área de interés dentro de la imagen.

El primer cambio que se realizó fue el método con el cual se realiza la binarización. En la etapa de experimentación 1, la binarización se realizaba mediante una umbralización con un umbral configurado a 128. El cambio radica en cómo se realiza este proceso, pues en esta etapa se realiza utilizando el método de Otsu [19]. Además, el banco de funciones que se utiliza para realizar la búsqueda es modificado. Se retiran las funciones que no aparecen en las soluciones encontradas y otras 3 funciones nuevas son agregadas. Las funciones que dejan de ser tomadas en consideración son multiplicación y división. Las funciones que se agregan al banco son Filtro Gaussiano, Detector de Bordes (Cany) y una función la cual realiza una operación píxel en donde los valores son invertidos, es decir, si el valor de un píxel es de 255, al realizar la inversión ahora su valor es de 0. Prácticamente se realiza una resta $|255 - I(x,y)|$. El conjunto de funciones utilizadas en esta etapa de experimentación se muestra en el Cuadro 5.4.

CUADRO 5.4: Conjunto de funciones. Experimento 2.

Nombre de función	Definición
Suma	$(x + y)$
Resta	$(x - y)$
Erosión	Operación morfológica de erosión
Dilatación	Operación morfológica de dilatación
Opening	Operación morfológica de <i>opening</i>
Closing	Operación morfológica de <i>closing</i>
Máscara/Umbral	Operación píxel
Filtro Gaussiano	Filtro
Detector de bordes Cany	Filtro
Invertir Valores	$ 255 - I(x,y) $
Terminales	
Im	Imagen en escala de grises (2-D)

5.2.2. Discusión y resultados

Tras realizar las modificaciones previamente mencionadas se procedió a realizar la ejecución del algoritmo, el cual fue ejecutado 10 veces obteniendo un desempeño favorable, tanto de manera visual como de aptitud en cuanto a la métrica de segmentación. Además, las imágenes resultantes mejoraron visualmente, pues se logra identificar el área de interés con mayor facilidad.

En la Figura 5.10 se muestra el comportamiento del algoritmo de la corrida que obtuvo la mejor aptitud. Como se puede observar en la gráfica aún existe una convergencia prematura en el proceso de búsqueda, alcanzando la mejor aptitud en el primer cuarto del proceso de evolución. Por otro lado el tamaño de la solución se mantiene un tanto estable después de que se alcanza el valor máximo de aptitud dentro de la población. Debido al comportamiento de la aptitud durante el proceso evolutivo en la siguiente etapa de experimentación se realizan ajustes en los parámetros del algoritmo con el fin de incrementar la aptitud de la solución y mejorar el comportamiento de búsqueda del algoritmo.

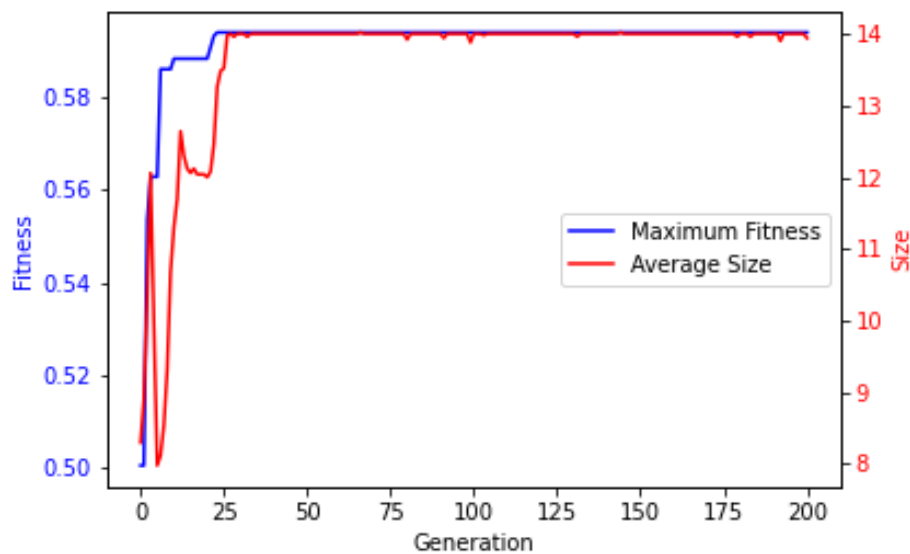


FIGURA 5.10: Gráfica de convergencia de la mejor ejecución.

En la Figura 5.11 se muestra un ejemplo del cambio que sufre la imagen original al realizar el procesamiento con la solución generada por GP-IFC. En 5.11(b) se puede apreciar que la característica construida es notablemente diferente a la imagen original. El área de interés es visualmente diferente al resto de la imagen. Posteriormente, en la

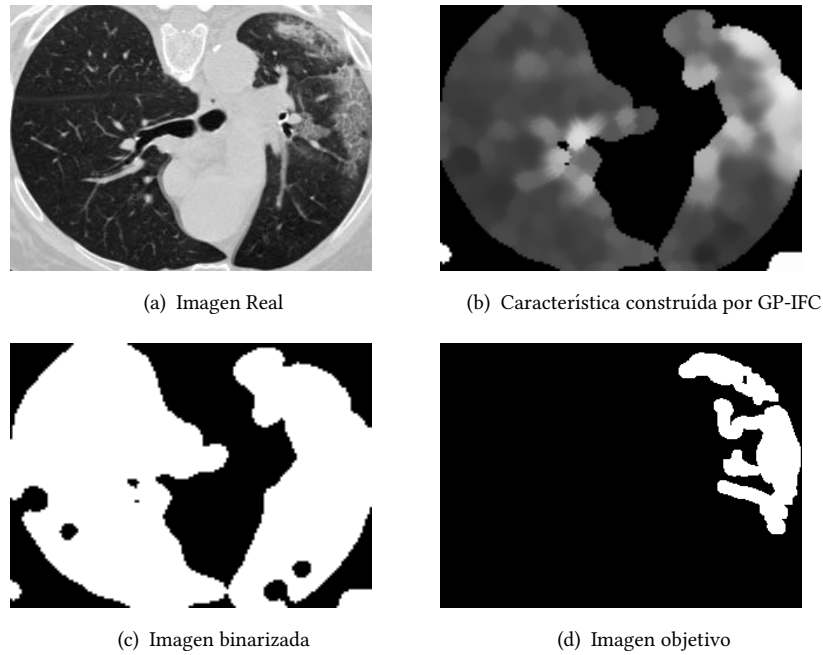


FIGURA 5.11: Ejemplo 1. Binarización con el método Otsu.

umbralización 5.11(c) no da el resultado esperado, pues la imagen resultante segmenta completamente el área de los pulmones, sin tomar en consideración el área de interés respecto a la imagen objetivo.

Por otro lado, en algunas imágenes resultantes tal y como se muestra en la Figura 5.12, la binarización se realiza de manera inversa ya que sobresale el área que no es de los pulmones, tal y como se muestra en la Figura 5.12(c). Aunque dentro del área correspondiente a los pulmones se logran apreciar ciertas regiones blancas, y esta se compara con la imagen objetivo 5.12(d) se aprecia que el área blanca marcada corresponde con el área de interés.

Hasta este punto se observa un comportamiento diferente en la segmentación de las imágenes debido a la variación que existe en cada una de ellas. Las soluciones generadas por GP-IFC generan resultados bastante interesantes en la construcción de la característica de manera general en el conjunto de imágenes.

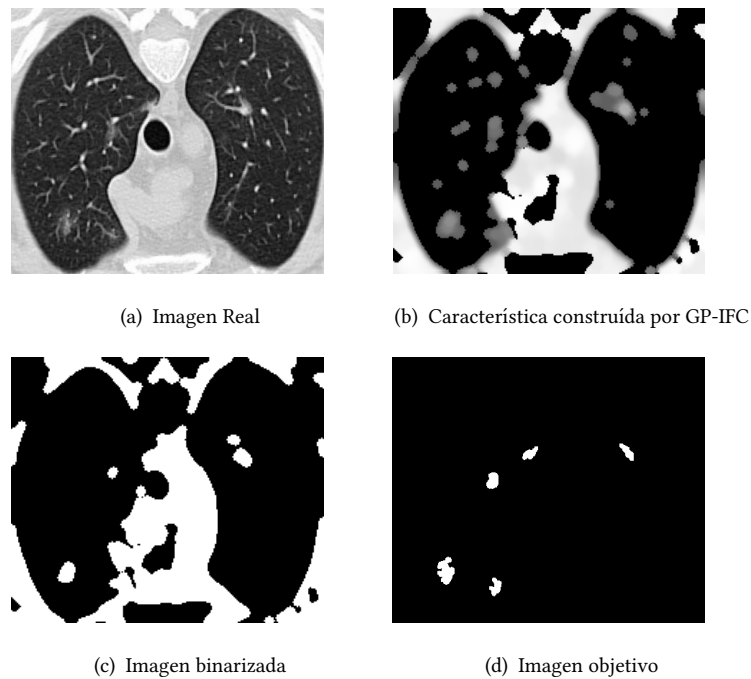


FIGURA 5.12: Ejemplo 2. Binarización con el método Otsu.

Para determinar si existe una diferencia entre los resultados de la etapa de experimentación 1 respecto a la etapa de experimentación 2, se realiza la prueba estadística no paramétrica U de Mann-Whitney con un nivel de confianza de 95 %. En el Cuadro 5.5 se muestran los resultados obtenidos por cada una de las corridas realizadas. Además, se muestran las estadísticas del experimento 1 y experimento 2. Con base en el resultado de la prueba estadística se determina que existe una diferencia significativa entre los dos experimentos, dando como resultado un mejor desempeño en esta segunda etapa de experimentación.

Ejecución	Exp. 1		Exp. 2	
	Aptitud	Tam. Solución	Aptitud	Tam. Solución
1	0.45412 ± 0.25	6.13 ± 0.73	0.56644 ± 0.122	8.08 ± 0.42
2	0.37047 ± 0.16	9.13 ± 5.73	0.55700 ± 0.120	7.92 ± 0.64
3	0.46636 ± 0.11	4.06 ± 1.253	0.54906 ± 0.229	9.48 ± 0.31
4	0.53227 ± 0.10	14.97 ± 0.50	0.54900 ± 0.120	10.02 ± 0.35
5	0.5912 ± 0.13	7.54 ± 0.77	0.56017 ± 0.233	6.14 ± 0.45
6	0.45859 ± 0.25	6.13 ± 0.73	0.58745 ± 0.223	7.42 ± 0.58
7	0.38749 ± 0.11	18.19 ± 1.84	0.49031 ± 0.105	6.14 ± 0.34
8	0.34591 ± 0.11	21.16 ± 2.69	0.53841 ± 0.158	10.46 ± 1.38
9	0.41363 ± 0.13	6.13 ± 0.73	0.59394 ± 0.123	10.58 ± 2.62
10	0.51022 ± 0.13	6.20 ± 1.66	0.54517 ± 0.115	8.36 ± 1.74
Estadística	Aptitud		Aptitud	
Mejor	0.5912		0.59394	
Peor	0.34591		0.4903	
Media	0.45320		0.55369	
Mediana	0.45635		0.55302	
Desv. Std	0.076		0.028	
<i>p</i> -value	0.0036			

CUADRO 5.5: Resultados y estadísticas de los experimentos 1 y 2.

5.2.3. Resumen

En esta segunda etapa se demostró que la binarización utilizando el método de Otsu no mejoró la calidad visual para determinar el área objetivo, sin embargo al agregar más funciones y quitar aquellas que no tomaban relevancia en el proceso de búsqueda contribuyó a realizar un mejor desempeño en términos de segmentación con la métrica DICE. Dados los resultados se destaca que el método con el cual se realiza la binarización también entra en juego, pues es comparando entre las dos etapas, se obtuvo un mejor rendimiento en la segunda. Por otro lado, en las ejecuciones del algoritmo se continúa teniendo una convergencia prematura en el proceso de búsqueda, alcanzando un máximo antes de la mitad de dicho proceso y manteniendo esa aptitud hasta el fin de las generaciones. Al observar este comportamiento repetitivo en las ejecuciones realizadas, en la siguiente etapa se realizan ajustes en los parámetros para cambiar

dicho comportamiento, así como reducir el número de generaciones para realizar una búsqueda menos tardada, pues si existe una convergencia a tempranas generaciones no es necesario realizar un proceso de evolución prolongado.

5.3. Experimento 3

5.3.1. Diseño experimental

Debido al comportamiento del algoritmo en el proceso de búsqueda en la etapa previa de experimentación, se realizan algunos cambios en ciertos parámetros. El parámetro de mutación se ve modificado en un valor de $+0,05$ y para el de cruza $-0,05$. Otro parámetro modificado es el número de generaciones. Esto se realiza con el fin de obtener un proceso no tan prolongado, pues dada la convergencia prematura que ocurre en las previas experimentaciones en alcanzar la solución a tempranas generaciones, es conveniente reducir el número de generaciones para realizar un proceso de búsqueda más corto. Aunado a esto, dadas las estadísticas mostradas en el Cuadro 5.1 y de acuerdo a la prueba no paramétrica realizada, no existe una diferencia significativa en el desempeño si este parámetro es modificado, es por ello que se toma la decisión de reducir el número de generaciones. Los parámetros nuevos y finales se muestran en el Cuadro 5.6.

CUADRO 5.6: Parámetros para ejecución de PG. Experimento 3.

Parámetro	Valor	Parámetro	Valor
Población	100	P. de cruza	0.75
Generaciones	100	P. de mutación	0.25
Profundidad mínima	5	Profundidad máxima	15
μ	100	λ	Población/2
Tipo de selección	Torneo (5)	Inicialización	Full

Además de cambiar los parámetros, en el conjunto de funciones es agregado un nuevo filtro. El filtro de Gabor puede tomar de manera aleatoria uno de los 5 valores para la frecuencia (0.2,0.4,0.6,0.8) y uno de los 5 valores de los grados de orientación de dicho filtro (15°, 30°, 45°, 75°y 90°).

CUADRO 5.7: Conjunto de funciones. Experimento 3.

Nombre de función	Definición
Suma	$(x + y)$
Resta	$(x - y)$
Erosión	Operación morfológica de erosión
Dilatación	Operación morfológica de dilatación
Opening	Operación morfológica de <i>opening</i>
Closing	Operación morfológica de <i>closing</i>
Máscara/Umbral	Operación pixel
Filtro Gaussiano	Filtro
Detector de bordes	Filtro
Invertir Valores	$ 255 - I(x,y) $
Filtro de Gabor	Filtro
Terminales	
Im	Imagen en escala de grises (2-D)

Dados los resultados obtenidos al realizar la binarización con el método de Otsu, en las imágenes resultantes se observa un desempeño que no es favorable, pues visualmente no se logra identificar el área en específico o el área objetivo a la que es requerida llegar. En este sentido, se realizó un cambio en el método utilizado para realizar la binarización, pues en esta etapa se realiza mediante el uso de clasificadores. Específicamente el clasificador ID3 y la Máquina de Soporte Vectorial (Support Vector Machine - SVM) con el fin de atacar la binarización como una tarea de clasificación a nivel píxel. En este sentido, el clasificador es el que se encarga de extraer características representativas a partir de la imagen original y la característica construida por GP-IFC y así realizar una clasificación para determinar si es área de interés o no. Cabe mencionar que es una clasificación binaria en donde se elige el valor de píxel correspondiente de acuerdo a la imagen objetivo y también que es un aprendizaje supervisado, pues a los clasificadores le son proporcionadas las imágenes originales, las imágenes construidas por GP-IFC y las imágenes objetivo. Para realizar la clasificación de los píxeles, las imágenes fueron convertidos a vectores, teniendo así las características (imagen real + imágenes de las características construidas) y sus etiquetas correspondientes (imágenes objetivo).

Para realizar la experimentación de binarización con el clasificador, el ID3 fue configurado para generar una solución con una profundidad máxima de 5 y así evitar un sobre ajuste. En cuanto a la Máquina de Soporte de Vectores se utilizaron las configuraciones establecidas por defecto de la librería. Por cada método de binarización se

realizaron 10 ejecuciones. En esta etapa de experimentación el algoritmo es probado con las cuadro bases de datos.

5.3.2. Discusión y resultados

5.3.2.1. Binarización con clasificador ID3

Después de implementar el clasificador para realizar la binarización durante el proceso se obtuvieron resultados sobresalientes, pues visualmente el área de interés se puede identificar con mayor facilidad y al comparar con la imagen objetivo muestra alto grado de similitud. Además, al medir el desempeño en términos de la métrica se obtienen mejores resultados, alcanzando en algunas imágenes resultantes valores mayores a 0.8 e incrementando la media a mas de 0.6.

La Figura 5.13 muestra el comportamiento del algoritmo de la mejor ejecución llevada a cabo utilizando la binarización mediante el clasificador ID3. Como se puede observar en la gráfica el comportamiento del algoritmo no reporta una convergencia prematura. A diferencia de las experimentaciones anteriores se está cumpliendo con el objetivo de mejorar el desempeño de dicha búsqueda, pues se alcanza un desempeño favorable en el primer cuarto del proceso y en generaciones posteriores continúa obteniendo una mejor aptitud. En cuanto a la aptitud del individuo es notable la diferencia, pues con base en la métrica DICE se obtienen valores superiores a 0.6. Además, la percepción visual de las imágenes resultantes es bastante favorable, logrando diferenciar el área de interés dentro de la imagen.

Por otro lado, analizando el comportamiento del tamaño de la solución que se genera a lo largo del proceso, se puede observar que dicha solución continúa creciendo y decreciendo. Esto es un indicador de que el algoritmo continúa realizando la búsqueda por una estructura diferente, dando pauta a que durante el proceso existe diversidad dentro de la población.

Las imágenes resultantes de esta etapa marcan una diferencia al momento de realizar la binarización, logrando un desempeño significativo. Esto a razón del desempeño de la segmentación pues al medirlo se alcanzan valores mayores a 0.8 en algunas imágenes. Además, se logra incrementar la media en cuanto a la métrica y la percepción de las imágenes es bastante clara.

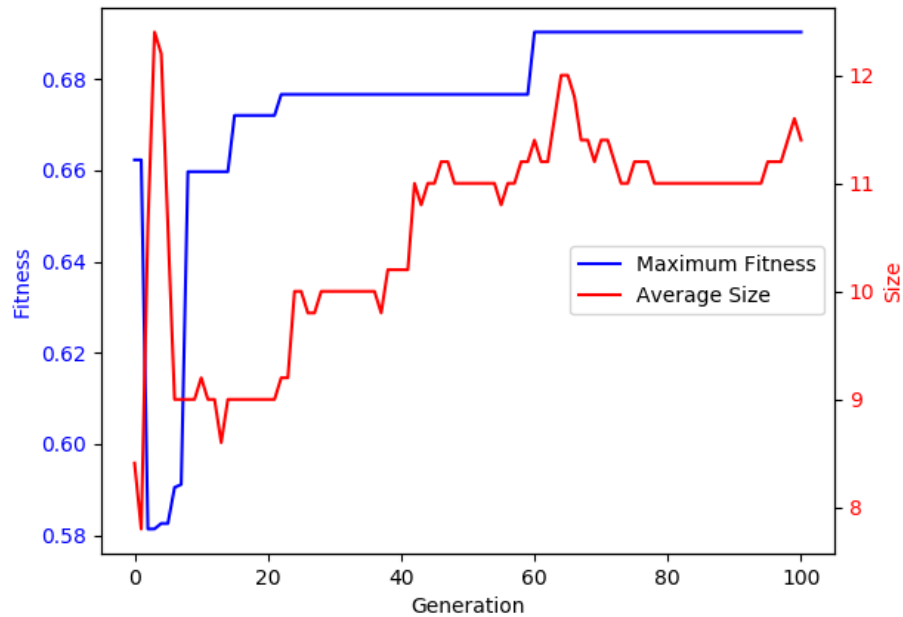


FIGURA 5.13: Gráfica de convergencia de la mejor ejecución utilizando ID3 como método de umbralización.

En la Figura 5.14 se muestra el desempeño que se obtiene tras realizar la binarización con el clasificador. Claramente se puede identificar el área de interés, pues al compararla de manera visual con la imagen objetivo se determina un gran similitud. Además, esta imagen alcanza un desempeño de segmentación en la métrica DICE de 0.779.

Para verificar que existe una diferencia significativa entre este método de binarización y el realizado en la etapa 2 de experimentación se realizan la prueba no paramétrica de U de Mann-Whitney con un nivel de confianza de 95 %. En el Cuadro 5.8 se muestran las estadísticas de las 10 ejecuciones realizadas en este experimento. De acuerdo al resultado de la prueba estadística realizada se demuestra que existe una diferencia significativa, determinando que el algoritmo obtuvo un mejor desempeño en cuanto a la aptitud al utilizar este método de umbralización.

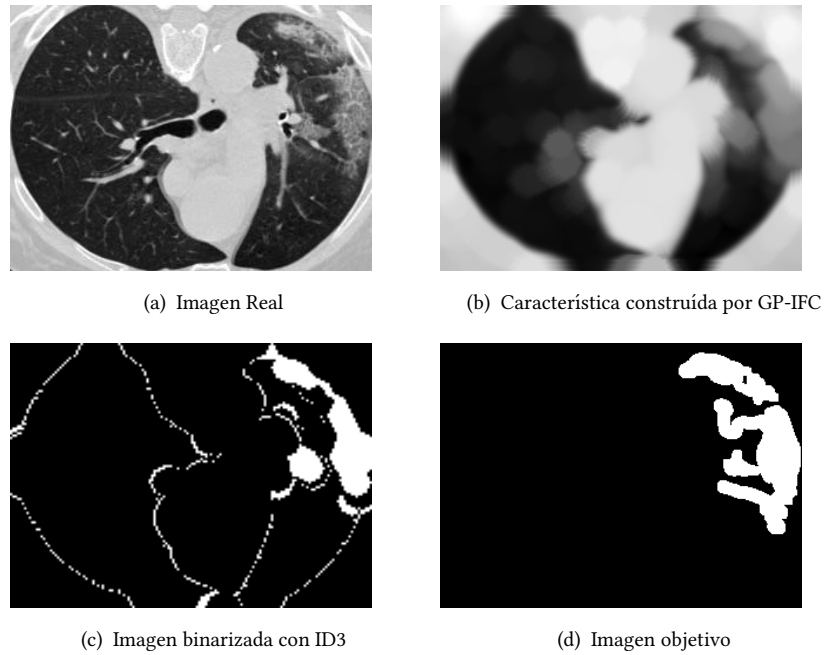


FIGURA 5.14: Ejemplo 1. Binarización con clasificador ID3.

Ejecución	Exp. 2		Exp. 3 (ID3)	
	Aptitud	Tam. Solución	Aptitud	Tam. Solución
1	0.56644 ± 0.122	8.08 ± 0.42	0.59890 ± 0.283	9.6 ± 3.41
2	0.55700 ± 0.120	7.92 ± 0.64	0.60214 ± 0.263	22.2 ± 4.81
3	0.54906 ± 0.229	9.48 ± 0.31	0.60277 ± 0.253	21.8 ± 3.76
4	0.54900 ± 0.120	10.02 ± 0.35	0.68752 ± 0.196	16.3 ± 7.53
5	0.56017 ± 1.233	6.14 ± 0.45	0.64142 ± 0.262	18.36 ± 5.76
6	0.58745 ± 0.223	7.42 ± 0.58	0.58694 ± 0.196	7.5 ± 4.27
7	0.49031 ± 0.105	6.14 ± 0.34	0.65915 ± 0.315	9.0 ± 0.0
8	0.53841 ± 0.158	10.46 ± 1.38	0.66841 ± 0.215	15.8 ± 1.83
9	0.59394 ± 0.123	10.58 ± 2.62	0.67151 ± 0.381	15.7 ± 1.84
10	0.54517 ± 0.115	8.36 ± 1.74	0.52612 ± 0.415	8.6 ± 1.35
Estadística	Aptitud		Aptitud	
Mejor	0.59394		0.68752	
Peor	0.4903		0.52612	
Media	0.55369		0.62449	
Mediana	0.55302		0.62210	
Desv. Std	0.028		0.0497	
p-value	0.0036			

CUADRO 5.8: Resultados y estadísticas de los experimentos 2 y 3 (ID3).

5.3.2.2. Binarización con clasificador SVM

Realizar la binarización con este clasificador no fue tan competitivo respecto al ID3, debido a que las imágenes resultantes muestran ciertas áreas que no son de interés. El desempeño en el conjunto de imágenes respecto a la métrica utilizada no alcanza una media mayor a 0.5 en las 10 ejecuciones realizadas.

En la Figura 5.15 se muestra el comportamiento de la mejor ejecución utilizando SVM como método de binarización. El comportamiento del algoritmo muestra un desempeño de búsqueda adecuado, pues no existe una convergencia prematura. Los mejores valores de aptitud son alcanzados antes de la mitad del total de generaciones. En la segunda mitad del proceso, es decir de la generación 50 en adelante el comportamiento se mantiene con un valor de aptitud alto hasta encontrar uno mejor y terminar el proceso de evolución. En cuanto al tamaño de las soluciones promedio en la población crece en el ultimo cuarto del proceso, este incremento va acorde al incremento en la aptitud. Cabe destacar que utilizando este método de umbralización el tamaño promedio de las soluciones contenidas en la población es mayor, pues sobrepasan una profundidad de 30 en el árbol de solución creado.

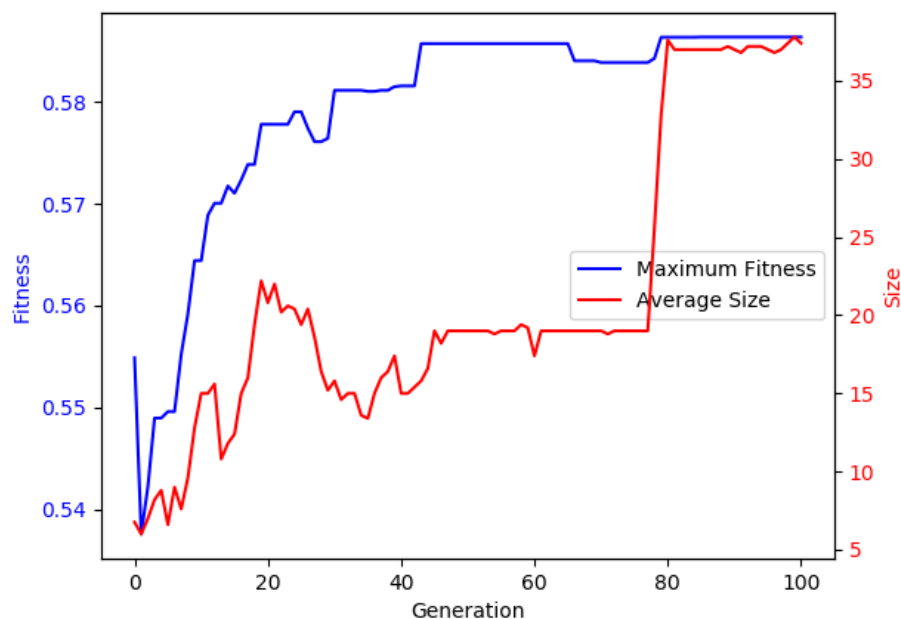


FIGURA 5.15: Gráfica de convergencia de la mejor ejecución utilizando SVM como método de umbralización.

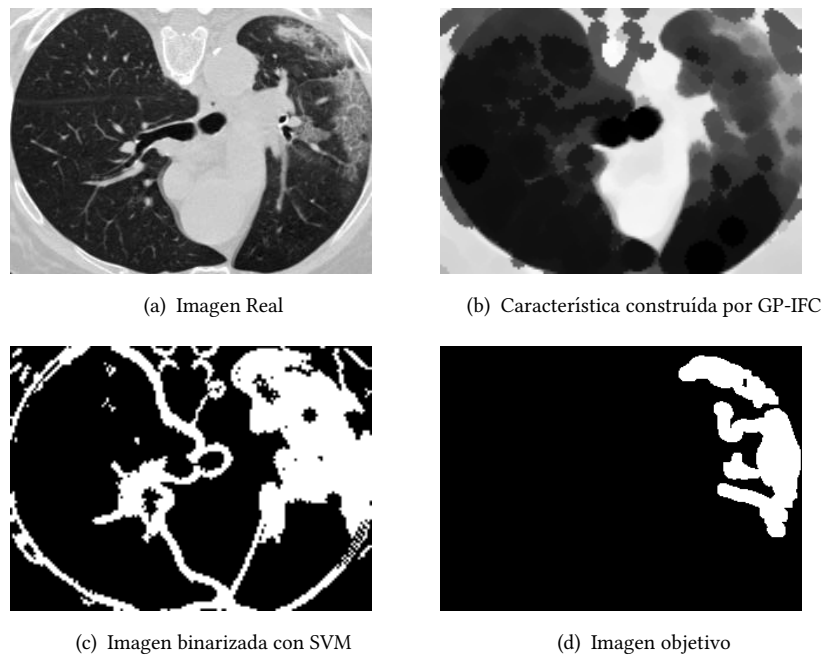


FIGURA 5.16: Ejemplo 1. Binarización con clasificador SVM.

En cuanto a las imágenes resultantes de este proceso, en la Figura 5.16 se muestra el resultado obtenido de binarizar la imagen con SVM. Como se puede observar en 5.16(c) la región de interés si es marcada cuando se realiza dicha binarización. Sin embargo, además de mostrar el área afectada se muestran otras regiones de la imagen como si fueran áreas de interés. Es interesante el resultado pues se muestra un contorno claro alrededor de los pulmones incluyendo el área afectada. En comparación con la imagen objetivo, de manera visual se logra un alto valor de similitud, sin embargo la métrica dice lo contrario, pues la imagen tiene un valor de DICE de 0.3561.

Para verificar que existe una diferencia significativa entre este método de binarización utilizando SVM y el realizado en la etapa 2 con el método de OTSU, se realiza la prueba no paramétrica de U de Mann-Whitney con un nivel de confianza de 95%. En el Cuadro 5.9 se muestran las estadísticas de las 10 ejecuciones realizadas en este experimento, comparándose con los resultados de la experimentación de la etapa 2. De acuerdo al resultado de la prueba estadística realizada se determina que se tiene un mejor desempeño con el algoritmo de la etapa de experimentación 2. A pesar de que visualmente pareciera que con la binarización realizada con SVM es eficiente, la prueba determina lo contrario.

Ejecución	Exp. 2		Exp. 3 (SVM)	
	Aptitud	Tam. Solución	Aptitud	Tam. Solución
1	0.56644 ± 0.122	8.08 ± 0.42	0.39054 ± 0.308	20.3 ± 7.53
2	0.55700 ± 0.120	7.92 ± 0.64	0.42013 ± 0.260	19.1 ± 8.129
3	0.54906 ± 0.229	9.48 ± 0.31	0.58720 ± 0.244	27.2 ± 1.46
4	0.54900 ± 0.120	10.02 ± 0.35	0.45777 ± 0.133	27.4 ± 2.20
5	0.56017 ± 0.233	6.14 ± 0.45	0.55058 ± 0.108	29.6 ± 2.93
6	0.58745 ± 0.233	7.42 ± 0.58	0.44644 ± 0.257	25.4 ± 4.47
7	0.49031 ± 0.105	6.14 ± 0.34	0.54680 ± 0.254	21.8 ± 0.40
8	0.53841 ± 0.158	10.46 ± 1.38	0.43212 ± 0.152	22.7 ± 2.19
9	0.59394 ± 0.123	10.58 ± 2.62	0.50718 ± 0.235	26.3 ± 2.10
10	0.54517 ± 0.115	8.36 ± 1.74	0.49577 ± 0.154	23.8 ± 1.46
Estadística	Aptitud		Aptitud	
Mejor	0.59394		0.59720	
Peor	0.4903		0.39054	
Media	0.55369		0.48445	
Mediana	0.55302		0.47677	
Desv. Std	0.028		0.066	
<i>p</i> -value	0.00058			

CUADRO 5.9: Resultados y estadísticas de los experimentos 2 y 3 (SVM).

5.3.3. Resumen

En esta etapa de experimentación se realizaron cambios en los parámetros del algoritmo los cuales fueron el número de generaciones así como la probabilidad de mutación y de cruza. También se agregó un filtro al conjunto de funciones, sumando un total de 11 funciones utilizadas para la construcción de características. Por otro lado, los parámetros de profundidad mínima y máxima se modificaron con el fin de que las soluciones tuvieran mayor profundidad, realizando más procesamiento en las imágenes y de esta manera construir una característica representativa.

Queda claro que el método de binarización también aportó para alcanzar un desempeño competitivo, pues el área de interés se puede identificar con mayor facilidad y respecto a la métrica DICE se determina que la segmentación se realiza de manera competitiva. De los cuatro métodos propuestos para realizar la binarización desde el

umbral, método Otsu, ID3 y SVM, el que mejor obtuvo desempeño fue con el clasificador ID3. En cuanto al clasificador SVM, el desempeño no fue competitivo de acuerdo a los valores obtenidos al medir la segmentación con la métrica DICE, así como el resultado obtenido mediante la prueba estadística en donde se comprueba que el método propuesto de binarización con SVM no tiene el desempeño adecuado.

5.4. Experimento 4

5.4.1. Diseño experimental

Hasta este punto ya se determinaron las funciones utilizadas en la programación genética para construir la característica a partir de la imagen original. Además, los parámetros tienen el ajuste adecuado para realizar un proceso de búsqueda y con ellos se llega a una solución competitiva, alcanzando valores mayores a 0.6 respecto a la métrica de DICE y también se tiene seleccionado método a utilizar para realizar la binarización.

En esta etapa de experimentación se plantea la construcción de más características. Este planteamiento surge debido a que para realizar la binarización con el clasificador solo se utilizan dos imágenes, la imagen original y la imagen que se construye, y a partir de ellas se realiza el entrenamiento del modelo para realizar una clasificación. Con el fin de que el modelo obtenga más información y de esta manera lograr un mejor desempeño en clasificación a nivel píxel, se plantea la construcción de múltiples características. Para llevar a cabo esto, en vez de generar solo un árbol como solución final, se propone realizar generar más mediante la creación de un bosque. Dicho bosque contiene 5 árboles, los cuales son evolucionados y en conjunto construyen 5 características con el objetivo de agregarlas al clasificador. En este sentido, el bosque es quien toma el rol de individuo y cada bosque tiene su propia aptitud. Dicha aptitud está en términos de la métrica de DICE. En cada bosque de manera interna se aplican los operadores tal y como se realiza en los experimentos 1,2 y 3.

Como resultado de este proceso de evolución se obtiene el bosque con la mejor aptitud el cual contiene 5 árboles o 5 programas los cuales se encargan de realizar la construcción de características a partir de la imagen original. Dichas características le son proporcionadas al clasificador al momento de realizar la binarización. El clasificador que se utiliza en esta etapa es el ID3, pues de acuerdo a la etapa previa es el

que tiene un mejor desempeño para realizar la tarea de binarización. Para este enfoque se utiliza una población de bosques reducida con el fin de igualar el número de árboles generados respecto a las previas experimentaciones. En este sentido se generan 20 bosques con 5 árboles cada uno. Los demás parámetros continúan tal y como se muestra en la Tabla 5.6.

5.4.2. Discusión y resultados

Tras realizar 10 veces la ejecución del algoritmo, el desempeño al momento de realizar la construcción de características es bastante competitivo, pues los resultados en términos de segmentación incrementan de manera considerable. Además, visualmente se puede apreciar de manera clara el área afectada que está presente en los pulmones. Cuando se realiza la comparación de la imagen binarizada respecto a la imagen objetivo se observa una alta similitud.

El comportamiento del algoritmo en esta implementación es bastante diferente, pues en algunas ejecuciones presenta una convergencia prematura. Cabe mencionar que a comparación de las gráficas de convergencia de los experimentos anteriores, en la Figura 5.17 no se muestra el comportamiento del tamaño promedio de la solución que se encuentra en la población. Esto se debe a un conflicto en la herencia de clases de la librería utilizada, pues en la implementación ahora se toma en consideración que el bosque evolucionado es el individuo principal y el árbol que se encuentra en el bosque es un individuo secundario, al cual no se le puede extraer la propiedad de profundidad durante el proceso de evolución. Como se puede observar, la mejor solución alcanza un valor significativo a comparación de las pruebas anteriores, pues se llega a un valor mayor de 0.75 en la métrica de segmentación.

En cuanto a los resultados visuales, la imagen resultante destaca por identificar el área de manera adecuada, aunque en algunas imágenes resultantes se muestran pequeñas áreas que no son marcadas por la imagen objetivo. Sin embargo, el área objetivo sí es marcada con claridad.

En la Figura 5.18 se muestra un ejemplo de los resultados obtenidos con el enfoque propuesto, en donde se realiza la construcción de 5 características 5.18(a) - 5.18(f). Estas 5 características construidas son agregadas al clasificador con el fin de incrementar el desempeño de la binarización. De esta manera el clasificador ahora tiene más información para determinar cuál característica y qué valores son los que más aportan al

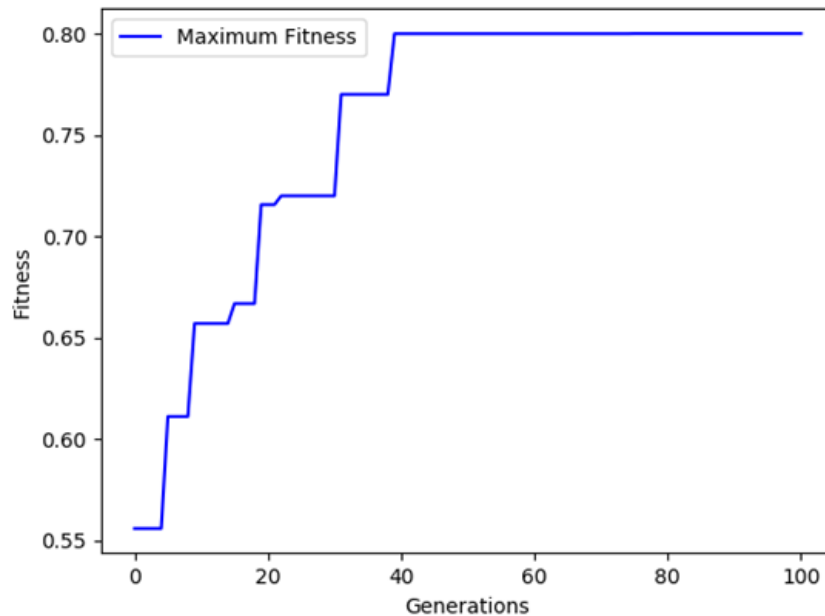


FIGURA 5.17: Gráfica de convergencia de la mejor ejecución. Experimento 4 - COVID-19.

momento de realizar la binarización. El resultado obtenido es visualmente competitivo, pues el área de interés es bastante notoria y el desempeño sobresale del resto de los experimentos.

Como en cada etapa de experimentación, en esta, se realiza una comparación del desempeño del algoritmo con respecto a la etapa anterior con el fin de determinar si el algoritmo realiza un mejor desempeño tanto en cuestión de segmentación con la métrica, así como la percepción visual.

De la misma manera, los algoritmos son probados con los otros 3 conjuntos conjuntos de imágenes, pues se pretende que ambos realicen un desempeño favorable al momento de realizar la segmentación y determinar el área de interés. Por cada conjunto de imágenes se realiza la prueba estadística U de Mann-Whitney con un nivel de confianza de 95 % para determinar si existe una diferencia significativa en cuanto al desempeño de ambos algoritmos. También, los resultados se utilizan como referencia para determinar en cuál conjunto de imágenes es favorable agregar más características al clasificador para realizar la binarización.

El primer conjunto de imágenes en donde son probados y comparados es el de COVID-19. En este, ambos muestran un desempeño favorable, pues de manera visual ambos algoritmos logran determinar el área afectada. Sin embargo, en cuanto a los

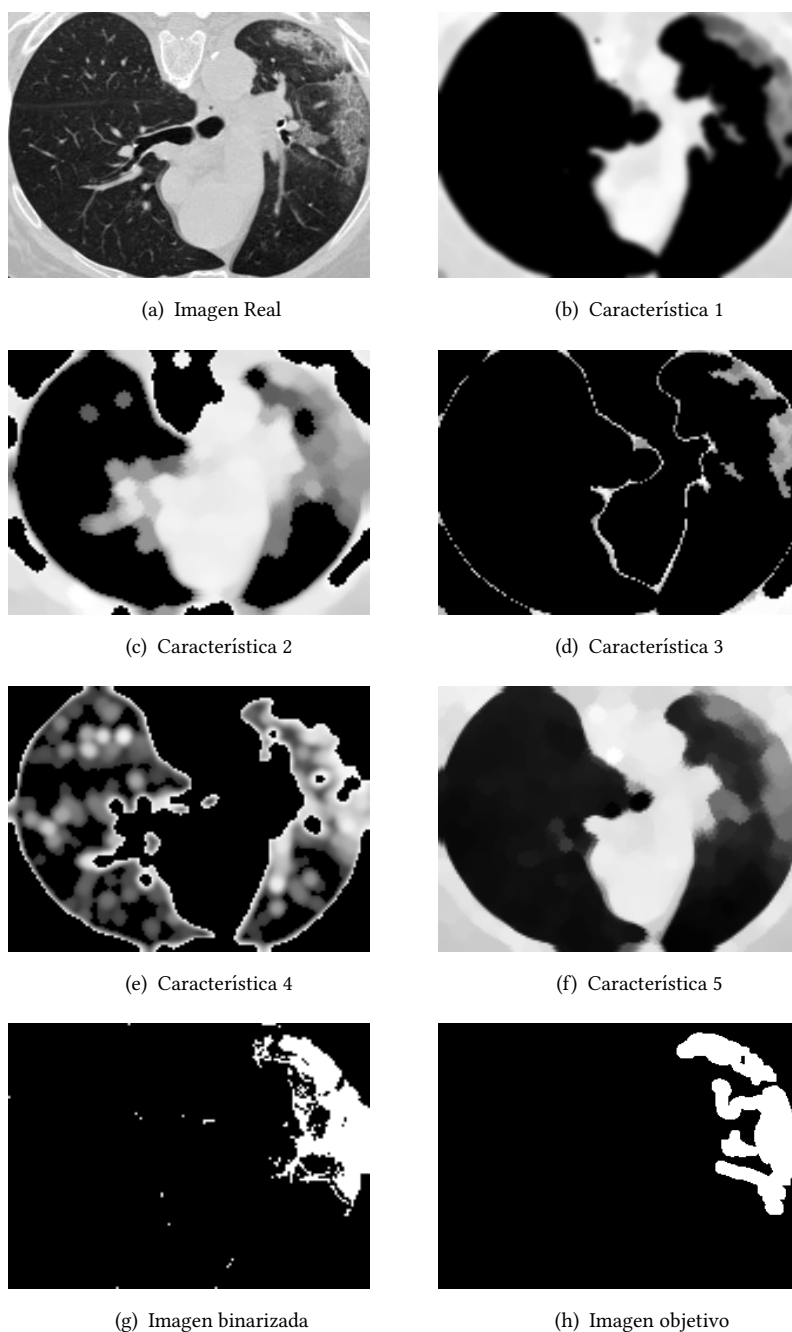


FIGURA 5.18: Ejemplo de resultados COVID-19.

valores alcanzados por la métrica de DICE y de acuerdo a los resultados obtenidos por la prueba estadística se determina que si existe una diferencia significativa, en donde el algoritmo de la etapa de experimentación 4 tiene mejor desempeño que el algoritmo de la etapa 3. Además, la media incrementó en un 0.126 unidades respecto a la métrica de DICE. En el Apéndice A.4 se muestran algunas comparaciones de las imágenes resultantes de ambos algoritmos.

El segundo conjunto de imágenes utilizado fue *Skin lesson*. El desempeño para ambos algoritmos con estas imágenes se observó competitivo en ambos, pues en algunas ejecuciones del algoritmo de la etapa 3 alcanzaron un valor mayor 0.7 en la métrica de dice. De igual manera, para el algoritmo de la etapa 4 se alcanzaron valores mayores a 0.7, así como un notable incremento en la media de 0.185 unidades en la métrica de DICE. Sin embargo, con la prueba estadística se determinó que existe una diferencia significativa en ambos algoritmos. De esta manera se determina que el algoritmo de la etapa 4 obtiene un mejor desempeño. Además, se determina que el uso de más características para realizar la binarización fue favorable permitiendo un mejor desempeño.

El tercer conjunto de imágenes probado fue *Retine Vessel*. Al realizar las pruebas con el algoritmo de la etapa 3 se obtuvieron resultados que destacaron, pues se logra una métrica mayor a 0.7 y la percepción visual es favorable para determinar el área de interés. Por otro lado, en cuanto el algoritmo de la etapa 4 el desempeño no fue el mismo, pues los resultados de algunas ejecuciones no se logra determinar con claridad el área de interés. Al realizar la prueba estadística se determinó que no existe una diferencia significativa, lo cual indica que el algoritmo de la etapa 4 tiene el mismo desempeño que el de la etapa 3. En el mismo sentido, se determinó que al agregar más características para este conjunto de imágenes no favoreció al desempeño de la segmentación para identificar el área de interés.

Por ultimo, se realizó la prueba en el conjunto de imágenes *Brain CT images with Intracranial Hemorrhage*. Los resultados en este conjunto de imágenes mejoraron en cuanto a la métrica, pues de manera visual no se observa una diferencia clara como lo fue en los demás conjuntos de imágenes. La media en el algoritmo de la etapa 4 aumentó en 0.133 unidades en la métrica de DICE respecto al algoritmo de la etapa 3. De acuerdo con el resultado de la métrica estadística realizada se determinó que si existe una diferencia significativa, además el resultado demuestra que agregar más características al clasificador favorece al desempeño. Los resultados de las pruebas realizadas se encuentran mostrados en el Cuadro 5.10, en el cual se encuentran las 10 ejecuciones de ambos algoritmos por cada uno de los conjuntos de imágenes en los que fueron probados. Los valores marcados en negritas muestran las ejecuciones que mejor desempeño alcanzaron. El símbolo \uparrow corresponde al algoritmo que obtuvo el mejor desempeño con base a los resultados obtenidos de la prueba estadística realizada.

Ejecución		Desempeño de segmentación (DICE)											
		COVID-19			Skin lesson			Retine Vessel			Brain CT Images		
		Exp. 3 (ID3)	Exp. 4 ↑	Exp. 4 ↑	Exp. 3 (ID3)	Exp. 4 ↑	Exp. 4 ↑	Exp. 3 (ID3)	Exp. 4	Exp. 4	Exp. 3 (ID3)	Exp. 4 ↑	
1	0.59890 ± 0.283	0.77662 ± 0.134	0.74762 ± 0.198	0.68100 ± 0.237	0.74762 ± 0.198	0.72180 ± 0.027	0.27097 ± 0.115	0.41420 ± 0.178	0.55591 ± 0.176				
2	0.60214 ± 0.263	0.74069 ± 0.161	0.77122 ± 0.167	0.58253 ± 0.223	0.77122 ± 0.167	0.72189 ± 0.027	0.24671 ± 0.112	0.38506 ± 0.295	0.37785 ± 0.247				
3	0.60277 ± 0.253	0.73149 ± 0.161	0.78020 ± 0.122	0.71655 ± 0.135	0.78020 ± 0.122	0.60165 ± 0.052	0.55044 ± 0.232	0.18052 ± 0.133	0.30801 ± 0.264				
4	0.68752 ± 0.196	0.74812 ± 0.159	0.72262 ± 0.215	0.76490 ± 0.143	0.72262 ± 0.215	0.19235 ± 0.153	0.74769 ± 0.025	0.30850 ± 0.177	0.56400 ± 0.190				
5	0.64142 ± 0.262	0.74278 ± 0.182	0.77905 ± 0.169	0.32590 ± 0.341	0.77905 ± 0.169	0.36017 ± 0.147	0.72860 ± 0.008	0.27822 ± 0.167	0.31666 ± 0.226				
6	0.58694 ± 0.196	0.75051 ± 0.165	0.77491 ± 0.165	0.55145 ± 0.244	0.77491 ± 0.165	0.71471 ± 0.024	0.66228 ± 0.028	0.33984 ± 0.188	0.50176 ± 0.242				
7	0.65915 ± 0.315	0.75278 ± 0.151	0.78152 ± 0.189	0.49898 ± 0.278	0.78152 ± 0.189	0.59615 ± 0.086	0.73090 ± 0.024	0.23748 ± 0.178	0.42440 ± 0.227				
8	0.66841 ± 0.215	0.73820 ± 0.134	0.76184 ± 0.186	0.57996 ± 0.278	0.76184 ± 0.186	0.62354 ± 0.122	0.68931 ± 0.059	0.32124 ± 0.200	0.38559 ± 0.282				
9	0.67151 ± 0.381	0.81540 ± 0.216	0.75004 ± 0.212	0.39476 ± 0.358	0.75004 ± 0.212	0.70152 ± 0.021	0.65552 ± 0.054	0.29462 ± 0.180	0.53050 ± 0.232				
10	0.52612 ± 0.415	0.71014 ± 0.172	0.71997 ± 0.205	0.63469 ± 0.226	0.71997 ± 0.205	0.69421 ± 0.038	0.55260 ± 0.153	0.31459 ± 0.167	0.44868 ± 0.212				
Estadísticas													
Mejor	0.68752	0.81549	0.78152	0.76490	0.78152	0.72189	0.74769	0.41420	0.56400				
Peor	0.52612	0.71042	0.71997	0.32590	0.71997	0.19235	0.24670	0.18052	0.30801				
Media	0.62449	0.75071	0.75890	0.57306	0.75890	0.59280	0.58350	0.30743	0.44134				
Mediana	0.62210	0.74545	0.76653	0.58125	0.76653	0.65888	0.65890	0.31155	0.43654				
Desv. Std	0.0497	0.028	0.023	0.138	0.023	0.178	0.184	0.067	0.094				
p-value	0.00018			0.00078			0.9681			0.00578			

CUADRO 5.10: Estadísticas de los experimentos 3 (ID3) y 4.

5.4.3. Resumen

En esta última etapa de experimentación se realizaron pruebas del algoritmo con 4 diferentes conjuntos de imágenes para determinar el desempeño del mismo. Además, se realizaron comparaciones entre el algoritmo de la etapa de experimentación 3 y el algoritmo de esta última etapa. Los resultados alcanzados por el algoritmo de esta etapa destacaron en su desempeño tanto de manera visual y en cuanto a la métrica de segmentación. Los valores numéricos de las medias incrementaron de manera considerable en los conjuntos de imágenes *COVID-19*, *Skin lesson* y *Brain CT images with Intracranial Hemorrhage*. De la misma manera se determinó que en esos conjuntos de datos es favorable el uso de más características para que el clasificador desempeñe la binarización y se alcance una mejor segmentación.

En cambio, específicamente para el conjunto de imágenes de *Retine Vessel*, este enfoque no fue favorable. Con base en la prueba estadística realizada, el desempeño de segmentación no obtuvo una diferencia significativa en comparación del algoritmo de la etapa 3. Así mismo, en el aspecto visual las imágenes resultantes no se destacan diferencias puntuales para mostrar con más claridad el área de interés.

Los resultados obtenidos en 3 de los 4 conjuntos de imágenes demuestran competitividad del algoritmo propuesto, tanto en el desempeño de segmentación como en la percepción visual del área de interés, dando como resultado un algoritmo con la capacidad de realizar la tarea de construcción de características de manera automática en diferentes conjuntos de imágenes del área médica para poder llevar a cabo una segmentación competitiva, lo cual conlleva una mejor identificación del área de interés.

5.5. Comparación con métodos del estado del arte

Tras las experimentaciones realizadas y dados los resultados competitivos obtenidos en cuanto a nivel de segmentación, se realizó la comparación del algoritmo propuesto con otros métodos del estado del arte encargados de realizar segmentación de imágenes. Dos de ellos son bajo el enfoque de programación genética, los cuales son StronglyGP y TwostageGP, ambos tomados de [15]. Estos fueron tomados en consideración para realizar la comparación debido a que son métodos actuales y además,

fueron probados con diferentes conjuntos de imágenes obteniendo resultados competitivos. Cabe mencionar que ambos algoritmos fueron configurados con los parámetros reportados en [15].

Por otro lado, también se realiza la comparación con otros dos métodos conocidos, los cuales se encargan de realizar la segmentación en imágenes médicas. Estas son las CNN's (Convolutional Neural Network's) UNet [24] y UNet++ [30].

En el Cuadro 5.11 se reportan las medianas de cada uno de los métodos comparados. Por cada uno de ellos se realizaron 10 ejecuciones. En cuanto a métricas DICE, los valores marcados con \uparrow representan el mejor valor obtenido de acuerdo con la prueba estadística U de Mann-Whitney con un nivel de confianza de 95 %. Además, se reportan los resultados de métricas Especificidad (SPC), Sensitividad (SEN) y Hausdorff Distance. En negritas se muestran los mejores valores obtenidos en la mediana de cada método.

Para el primer conjunto de imágenes *COVID-19*, se obtuvo un resultado favorable para GP-IFC en cuanto al desempeño de segmentación. Se puede observar en la tabla 5.11 que los valores alcanzados son bastante sobresalientes respecto a los métodos con los que se está comparando. En cuanto al método de StronglyGP, el desempeño para realizar la segmentación no favorece pues utiliza como método de binarización el método Otsu, dando como resultado una segmentación no tan competitiva en un porcentaje de imágenes. El utilizar este método para la binarización quedó descartado en la etapa de experimentación 2. Respecto a TwostageGP logra alcanzar un mejor valor que StronglyGP. Por otro lado, el desempeño con las CNN's es notorio, pues Unet++ alcanza un valor de 0.54038. UNet++ alcanza un nivel mayor respecto a UNet, TwostageGP y StronglyGP pero no mayor que GP-IFC. Visualmente los resultados no muestran con claridad el área afectada en los pulmones, pues en algunas imágenes finales son completamente oscuras.

En el conjunto de imágenes de *Skin lesson* es notorio el desempeño obtenido por el algoritmo propuesto, superando por mucho a los algoritmos con los cuales se están comparando. El segundo método que obtuvo mejores valores en cuanto a desempeño fue TwostageGP. Esto es debido a que en la primera etapa de dicho algoritmo se realizan ciertas operaciones y filtros en la imagen, posteriormente en su segunda etapa, comienza a realizar operaciones pero en la imagen resultante de la primera etapa, la cual ya está completamente binarizada. Para el caso de Unet y Unet++ el desempeño

Método	Base de datos de imágenes											
	COVID-19					Skin lesson						
	DICE	SPC	SEN	HD	DICE	SPC	SEN	HD	DICE	SPC	SEN	HD
GP-IFC	0.74545 ↑	0.95487	0.80138	7.9085	0.76653 ↑	0.90350	0.85314	126.39				
StronglyGP	0.40169	0.96501	0.33086	2045.26	0.23269	0.63669	0.19702	1680.96				
TwostageGP	0.45691	0.89115	0.48671	16233.66	0.45576	0.81370	0.40159	1851.13				
Unet	0.43925	0.91819	0.63041	2300.48	0.19923	0.78806	0.27397	948.09				
Unet++	0.54038	0.95916	0.71819	2091.99	0.32248	0.90480	0.34495	2157.25				
Método	Base de datos de imágenes											
	Retine Vessel					Brain CT Images						
	DICE	SPC	SEN	HD	DICE	SPC	SEN	HD	DICE	SPC	SEN	HD
GP-IFC	0.65890 ↑	0.95547	0.87072	9.0123	0.43654 ↑	0.99034	0.81399	3.9328				
StronglyGP	0.54219	0.930281	0.53984	2087.34	0.13070	0.99238	0.07866	2650.29				
TwostageGP	0.44372	0.96210	0.31310	2697.37	0.16728	0.98849	0.11846	2542.92				
Unet	0.10856	0.91677	0.51323	1152.01	0.10282	0.99500	0.05872	2337.89				
Unet++	0.15381	0.91888	0.66411	980.59	0.10028	0.99601	0.05576	2396.68				

CUADRO 5.11: Comparación con métodos del estado del arte

no fue favorable, aunque visualmente se observa que en algunas imágenes resultantes se logra asemejar con la imagen objetivo, la métrica de DICE determina lo contrario.

El tercer conjunto de imágenes en donde se compara la propuesta es *Retine Vessel*. Para GP-IFC el desempeño fue favorable, tanto visual como el nivel de segmentación. El método alcanza un valor mayor a 0.60 en la métrica de DICE y sobrepasa los métodos con los cuales se compara. Una observación interesante es que los métodos de

basados en programación genética se desempeñan de manera más competitiva que las CNN's. Para este conjunto de imágenes el algoritmo StronglyGP se desempeña mejor que TwostageGP. Los resultados por parte de ambas CNN's no son favorables ni en desempeño de segmentación ni en la percepción visual de las imágenes resultantes.

Por último, en el conjunto de imágenes *Brain CT images with Intracranial Hemorrhage* los métodos con los que se realiza la comparación no alcanzan un valor de 0.2 en términos de la métrica DICE. Incluso, el desempeño de GP-IFC no fue tan favorable respecto a los otros conjuntos de imágenes. Además, dada la naturaleza de las imágenes el determinar con exactitud el área donde existe la hemorragia es bastante complicado, pues es una pequeña área respecto a la imagen completa. Apesar de ser un conjunto de imágenes en donde es complicado segmentar el área de interés, GP-IFC alcanza un nivel significativo al compararse con los demás métodos. Pues visualmente en algunas imágenes resultantes es posible visibilizar el área. Caso contrario para los demás métodos, pues no logran acercarse a la imagen objetivo.

Dadas los resultados y las comparaciones entre cada uno de los métodos se comprueba que el algoritmo propuesto GP-IFC tiene un desempeño competitivo con respecto a los métodos del estado del arte. Además, en la percepción visual de las imágenes resultantes se puede determinar con facilidad el área de interés, dando como resultado que es una herramienta capaz de construir las características necesarias para poder llevar a cabo la segmentación de manera adecuada.

En la Figura B.1 se muestran algunos resultados de cada uno de los métodos por cada conjunto de imágenes en los que fueron probados.

5.5.1. Resumen

Para evaluar el desempeño de segmentación del método propuesto se realizaron comparaciones con cuatro métodos del estado del arte. *StronglyGP* y *TwostageGP* ambos están basados en programación genética y utilizan diferentes operadores y funciones de procesamiento de imágenes respecto al método propuesto. Los otros dos son las Redes Neuronales Convolucionales (CNN's) UNet y UNet++, las cuales destacan dentro del área del procesamiento de imágenes por desempeño que tienen para realizar la segmentación en imágenes médicas. El método propuesto destacó de manera puntual al competir con cada uno de ellos en diferentes conjunto de imágenes, pues se logró

alcanzar un mejor nivel que los métodos antes mencionados. Además, al comparar visualmente las imágenes resultantes con los métodos, se observó una diferencia notable, pues es fácil identificar el área de interés ya sea en el área afectada en los pulmones, el área de lesión en la piel, las venas en las retina o incluso el área de afectación por una hemorragia interna en el cráneo.

5.6. Resumen

En este capítulo se realizaron cuatro etapas de experimentación y una comparación del algoritmo propuesto con métodos actuales del estado del arte. En la primera etapa de experimentación se realizaron pruebas para determinar los parámetros adecuados a utilizar en la configuración del algoritmo. También, se realizaron pruebas para definir cuál medida de evaluación de segmentación es la más adecuada para utilizar como función de aptitud del algoritmo. Se descartaron por completo las métricas de Sensitividad, Especificidad y Hausdorff Distance. Se determinó que la mejor métrica para evaluar el desempeño de las soluciones generadas por el algoritmo es DICE, pues fue la única que permitió establecer e identificar el área de interés en las imágenes de manera clara. En la segunda etapa de experimentación se realizaron ajustes en algunos parámetros, pues existía un comportamiento no esperado del algoritmo durante el proceso de búsqueda. Así mismo, se encontró un mejor conjunto de funciones, las cuales mejoraron el desempeño del algoritmo. Además, se determinó que el método de binarización Otsu no era el adecuado, pues se lograba señalar el área de interés de manera clara. En la tercera etapa de experimentación, de acuerdo a los resultados estadísticos se determinó que el mejor método para realizar la binarización es mediante el uso de un clasificador. En la cuarta y última etapa de experimentación, se implementó un enfoque en donde se realiza la creación de bosques. En cada uno de los bosques se encuentran los árboles/solución que realizan la construcción de características. Por cada bosque se construyen 5 características, las cuales son proporcionadas al clasificador con el fin que de este obtenga más información para realizar la binarización y con ello mejorar el desempeño de segmentación, así como la percepción visual de las imágenes resultantes. Posterior a la experimentación, se realizó la comparación de GP-IFC con métodos del estado del arte. En dichas comparaciones el algoritmo propuesto destacó pues realizó un mejor desempeño en la segmentación y en la percepción visual de las imágenes resultantes.

Capítulo 6

Conclusiones y trabajo futuro

Basado en los resultados obtenidos en este trabajo, queda claro que la capacidad de los algoritmos evolutivos logra alcanzar resultados competitivos y, aunque no dan la mejor solución, nos ofrecen una o varias soluciones capaces de resolver el problema en cuestión. Para este trabajo, la programación genética nos brindó la posibilidad de crear soluciones utilizando programas y funciones de procesamiento de imágenes para realizar el proceso de construcción de características. Esto a su vez permitió realizar la segmentación del área de interés de manera adecuada, mostrando competitividad respecto a diferentes algoritmos actuales que atacan el problema de segmentación.

En este trabajo se presentó el diseño del algoritmo GP-IFC el cual realiza la construcción automática de características con el objetivo de realizar una segmentación competitiva e identificar con claridad el área de interés en las imágenes. El desarrollo del algoritmo propuesto se realizó en 4 etapas de experimentación con el objetivo de ir resolviendo un problema a la vez.

En la primera etapa de experimentación se definieron los parámetros adecuados para realizar la configuración del algoritmo, los cuales fueron población y el criterio de paro del algoritmo, que en este caso fue mediante el número de generaciones. Además, se determinó la métrica de DICE como la función de aptitud que mejor guía el proceso de búsqueda, pues de las 4 propuestas solo una obtuvo un desempeño favorable en cuanto al nivel de segmentación alcanzado y la percepción visual resultante en las imágenes.

En la segunda etapa de experimentación se modificó el conjunto de funciones propuestas a utilizar dentro de la programación genética. Se quitaron aquellas que no

toman relevancia en la solución y que a causa de ellas la solución se vio afectada, estas funciones fueron división y multiplicación. Por otro lado, se incluyeron 3 funciones, entre ellas 2 filtros y una función que realiza la inversión de valores en la imagen. Adicionalmente, se probó el método Otsu para realizar la binarización de las características construidas con el fin de observar el resultado. En esta etapa de experimentación se alcanzaron mejores resultados Sin embargo, se determinó que el realizar la binarización con dicho método no mejoraba la percepción visual de las imágenes resultantes. Dados estos resultados, se plantearon cambios en la propuesta y se procedió a cambiar la binarización en la etapa 3 de experimentación.

En la tercera etapa, se comprueba que utilizar un clasificador como método de binarización mejora considerablemente el desempeño de la segmentación. De los dos clasificadores utilizados y probados (SVM e ID3), se determinó que el que obtiene un mejor desempeño es el algoritmo ID3. De la misma manera, se obtiene una mejora significativa en la percepción visual de las imágenes resultantes. Además, en esta etapa se determinó el reajuste de parámetros con base en las pruebas realizadas en la etapa 1 con el fin de reducir el costo computacional en el proceso de búsqueda. Cabe mencionar que se agregó otra función filtro al banco de funciones utilizadas. En esta etapa, los resultados obtuvieron un nivel significativo, pues el desempeño de segmentación obtenido fue mayor a 0.6 unidades en la métrica de DICE.

En la última etapa de experimentación, la propuesta realizada alcanza valores competitivos, pues el desempeño de segmentación alcanzó valores mayores a 0.7 en términos de DICE. Además, se determina el área de interés de manera clara y concisa. Con los resultados obtenidos se realizaron comparaciones con otros métodos del estado del arte que realizan la tarea de segmentación. Dos de ellos basados en programación genética y otros dos, los cuales realizan la segmentación utilizando redes neuronales convolucionales.

Dados los resultados obtenidos tras probar el algoritmo en diferentes conjuntos de imágenes pertenecientes al ámbito médico y realizar la comparación con otros métodos del estado del arte, se confirma la hipótesis planteada en donde se afirma que se logrará obtener una mejor identificación del área afectada en los pulmones causada por COVID-19 mediante la construcción automática de características solo con las 11 funciones propuestas. De la misma manera, se valida que el método propuesto realiza una mejor identificación del área afectada en los pulmones causada por COVID-19. Incluso, el desempeño de GP-IFC estadísticamente realizó mejor identificación del área de interés en los 4 conjuntos de imágenes probados.

Acorde a lo previamente descrito se concluye que mediante el uso de programación genética se pueden generar soluciones de manera automática utilizando programas computacionales obteniendo resultados competitivos, tal y como los obtuvo GP-IFC, alcanzando un desempeño favorable en cuatro conjuntos de imágenes y determinando el área de interés de manera adecuada.

Por otro lado, el desarrollo de esta investigación abrió nuevas brechas en términos de investigación.

En la primera brecha se propone realizar una etapa previa a la construcción de características. En dicha etapa se propone realizar un mejoramiento de imágenes en cuestiones de calidad, en la cual se realice un preprocesamiento en donde se apliquen algoritmos completos, por ejemplo el mejoramiento de contraste.

Por otro lado, se propone investigar en una configuración del algoritmo para que se realice una búsqueda multiobjetivo. Así, de esta manera se realiza la búsqueda del programa que construye las características y por otro lado se busca el método que realice de mejor manera la binarización de las imágenes, pues como se determinó en este trabajo, al utilizar diferentes métodos de binarización los resultados varían en demasía.

Por último, otro enfoque multiobjetivo donde se realice la búsqueda de la mejor función que construya las características y se esté evaluando respecto a dos métricas, pues como se observó en este trabajo, el uso de diferentes métricas para la evaluación de segmentación guían a resultados distintos.

Apéndices

Apéndice A




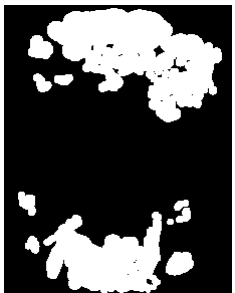
Imágenes y soluciones resultantes por cada experimento

A.1. Experimento 1

Imágenes resultantes tomadas aleatoriamente tras la primera etapa de experimentación en donde se encuentran las imágenes por cada una de las métricas probadas para guiar el proceso de búsqueda.

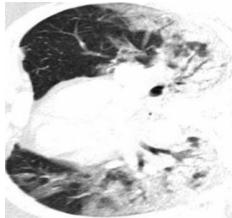
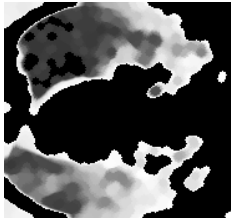


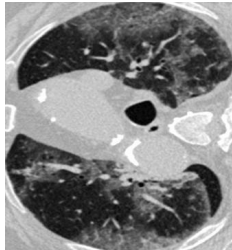



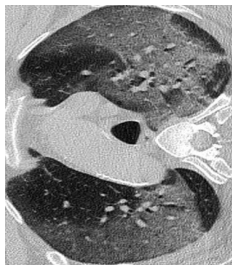



A.1.1. DICE

CUADRO A.1: Imágenes resultantes y soluciones encontradas por GP-IFC. DICE.

I. Original	GP-IFC	I. Segmentada	G.T
			

Continúa en la siguiente página...

Cuadro A.1: Imágenes resultantes y soluciones (continuación...)

I. Original	GP-IFC	I. Segmentada	G.T
			
			
			

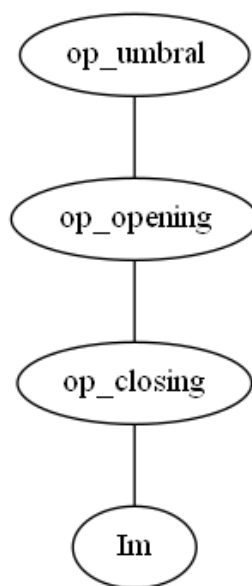




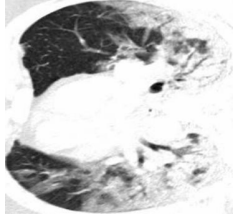



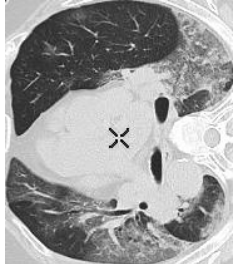




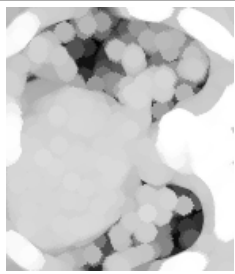




FIGURA A.1: Solución encontrada por GP-IFC. DICE.

A.1.2. Especificidad

CUADRO A.2: Imágenes resultantes y soluciones encontradas por GP-IFC. Especificidad.

I. Original	GP-IFC	I. Segmentada	G.T
			
			
			
			

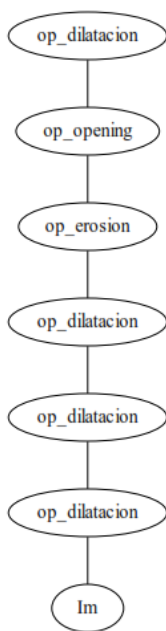


FIGURA A.2: Solución encontrada por GP-IFC. Especificidad.


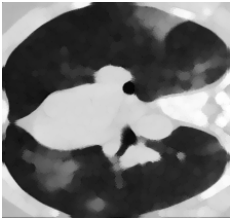


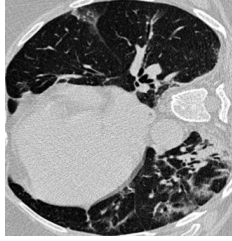
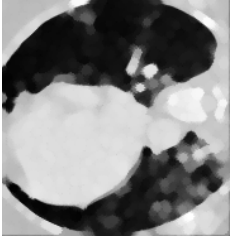


A.1.3. Sensitividad

CUADRO A.3: Imágenes resultantes y soluciones encontradas por GP-IFC. Sensitividad.

I. Original	GP-IFC	I. Segmentada	G.T

Continúa en la siguiente página...

Cuadro A.3: Imágenes resultantes y soluciones (continuación...)

I. Original	GP-IFC	I. Segmentada	G.T
			
			

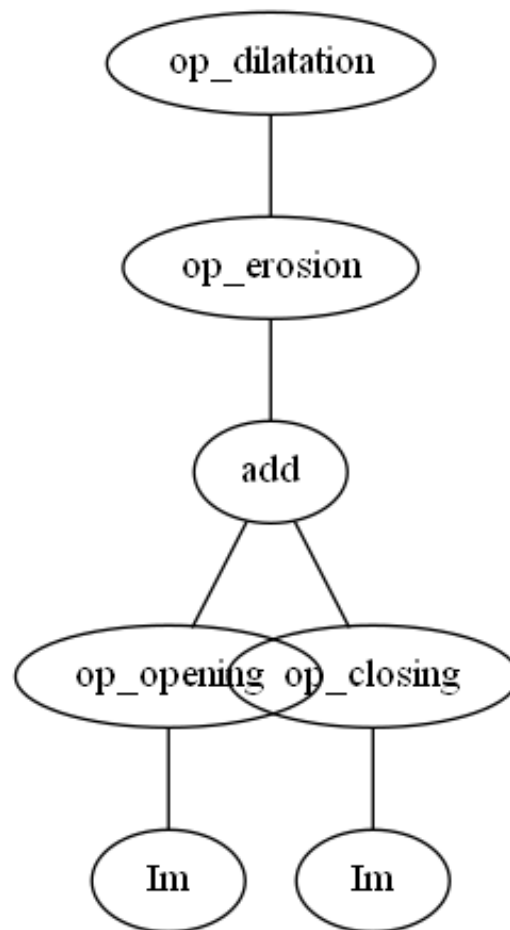


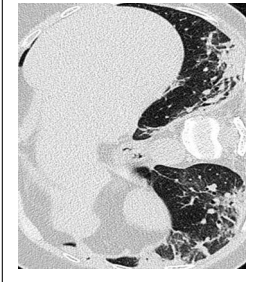
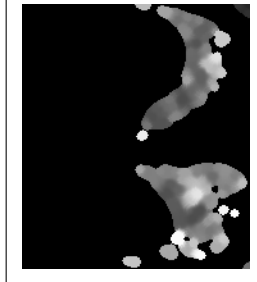
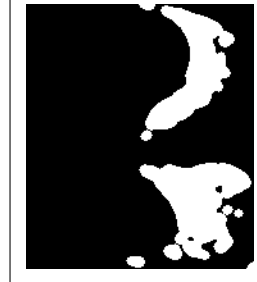

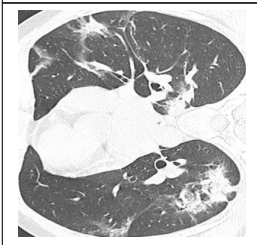
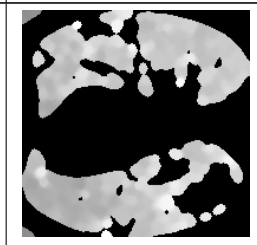

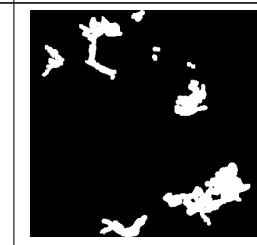
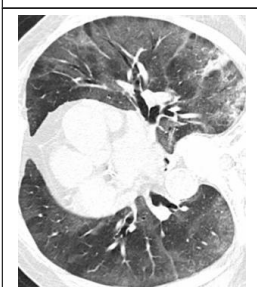
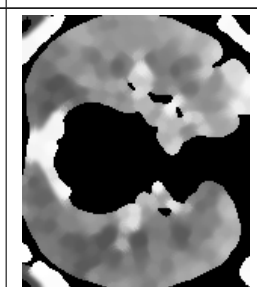

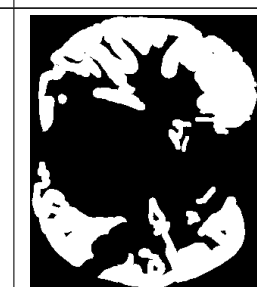

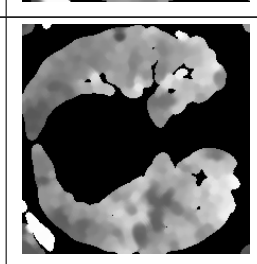
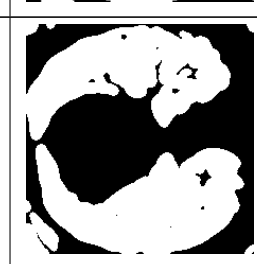
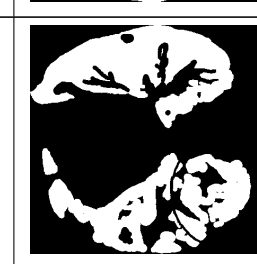
FIGURA A.3: Solución encontrada por GP-IFC. Sensitividad

A.1.4. Hausdorff Distance

A.2. Experimento 2

Imágenes resultantes de la segunda etapa de experimentación. En ellas se comienza a notar el desempeño del algoritmo de manera visual, pues los resultados asemejan a las imágenes objetivo.

CUADRO A.5: Imágenes resultantes y soluciones encontradas por GP-IFC. Etapa 2.

I. Original	GP-IFC	I. Segmentada	G.T
			
			
			
			

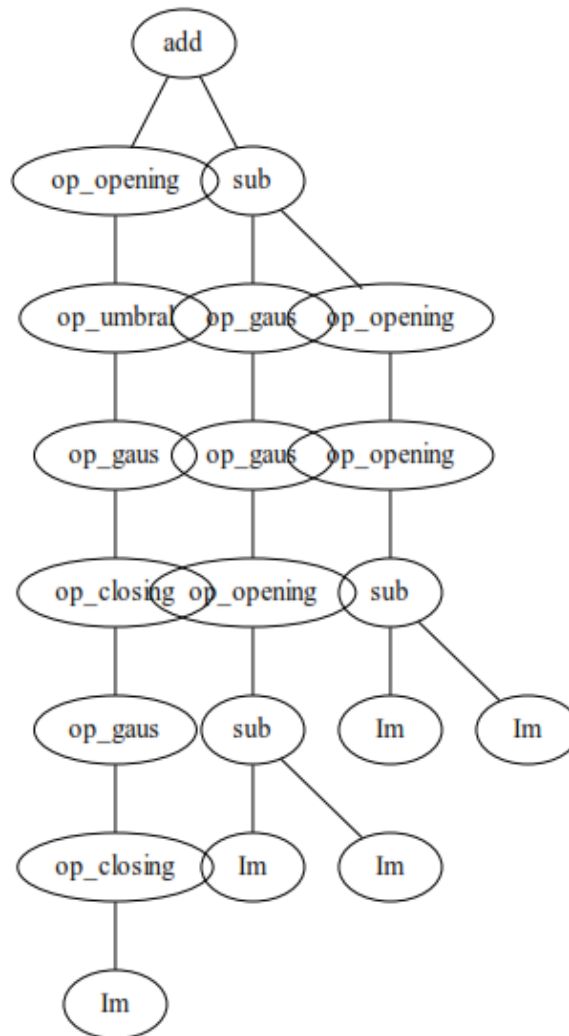


FIGURA A.5: Solución encontrada por GPI-FC. Etapa 2.

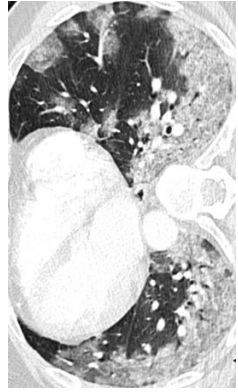
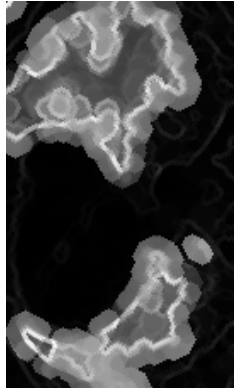


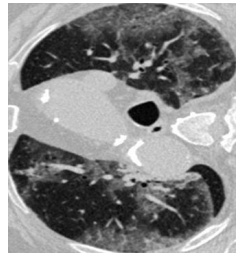
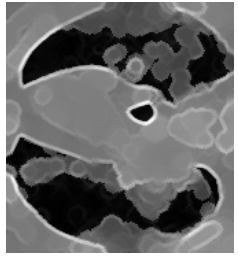


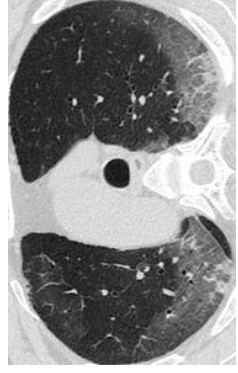
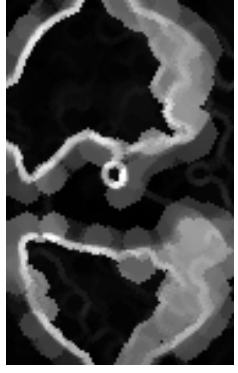


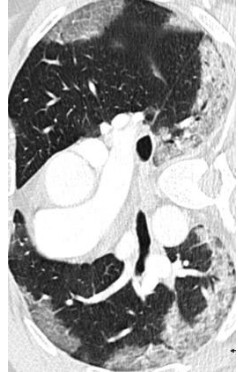
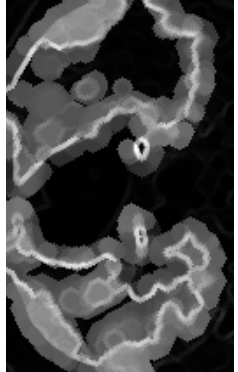
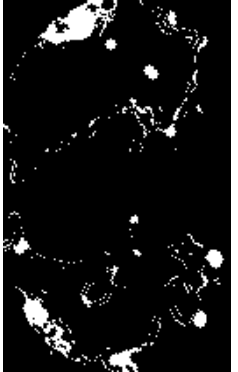

A.3. Experimento 3

Imágenes resultantes de la tercera etapa de experimentación. Visualmente el resultado sobresa a comparación de los experimentos previos, sin embargo en algunas imágenes aún se continúa determinando áreas diferentes a las de la imagen objetivo. También, se muestran las soluciones generadas por GP-IFC así como el árbol generado por el clasificador ID3.

A.3.1. ID3

A.3.1.1. COVID-19

CUADRO A.6: Imágenes resultantes y soluciones encontradas por GP-IFC. COVID-19 [2]

I. Original	GP-IFC	I. Segmentada	G.T
			
			
			
			

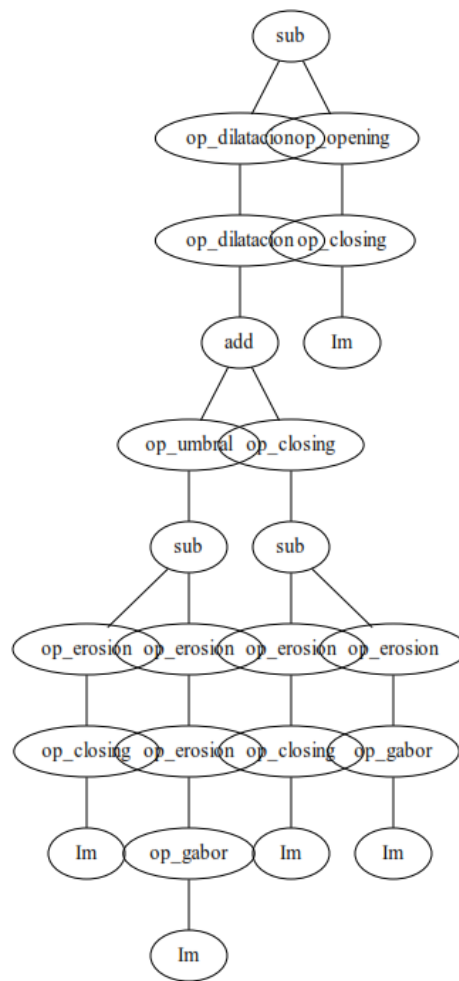


FIGURA A.6: Solución encontrada por GPI-FC. COVID-19

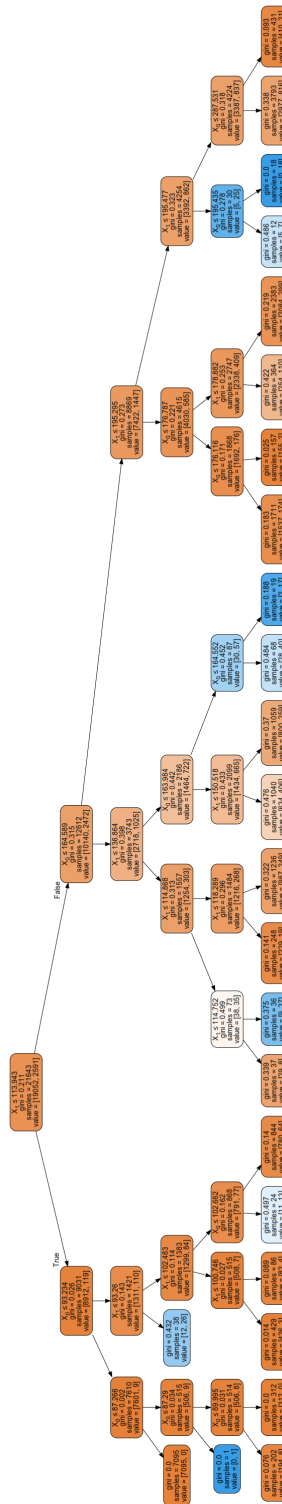
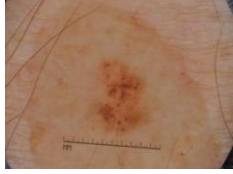



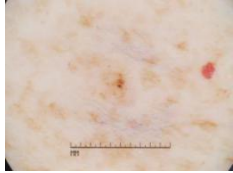



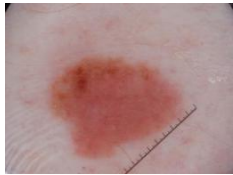
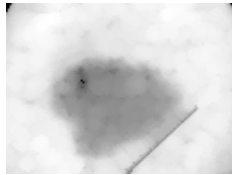


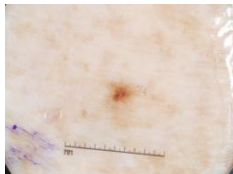
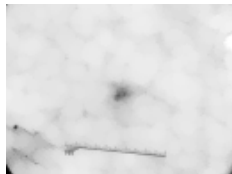




FIGURA A.7: Solución clasificador ID3. COVID-19

A.3.1.2. Skin lesson

CUADRO A.7: Imágenes resultantes y soluciones encontradas por GP-IFC. Skin Lesson [4].

I. Original	GP-IFC	I. Segmentada	G.T
			
			
			
			

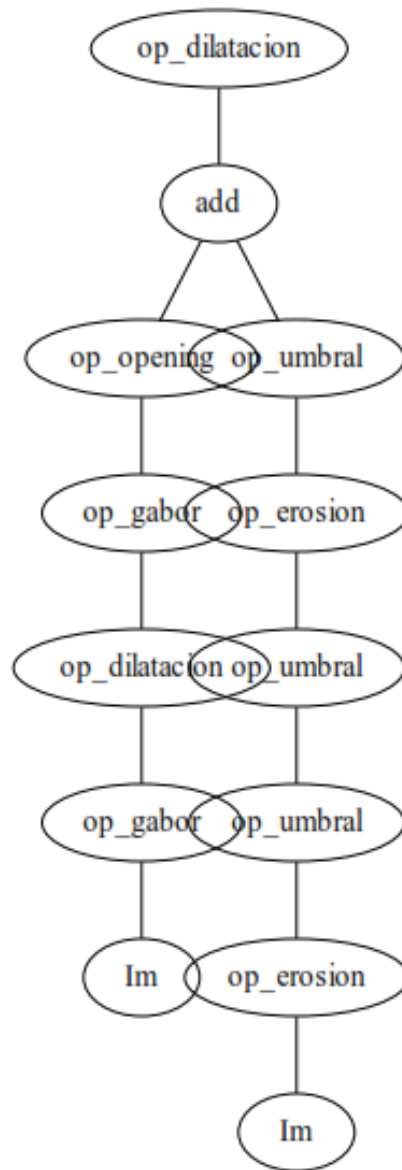


FIGURA A.8: Solución encontrada por GPI-FC. Skin Lesson

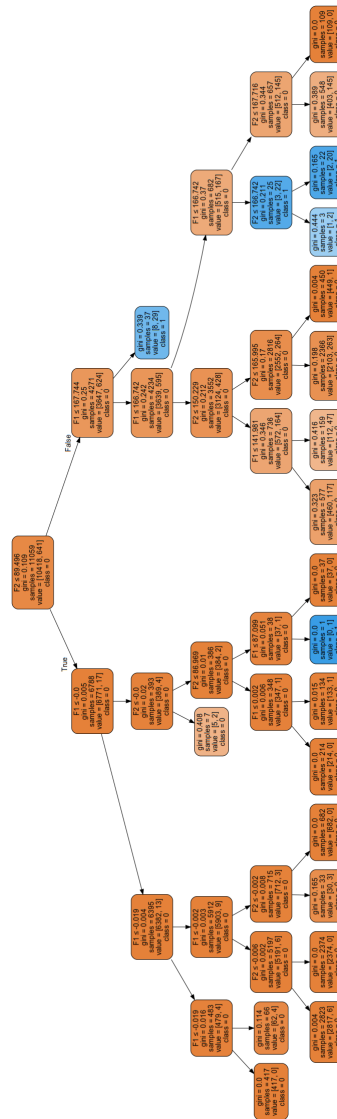

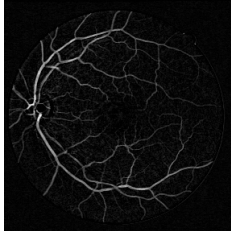
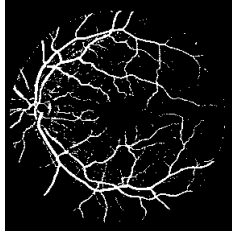
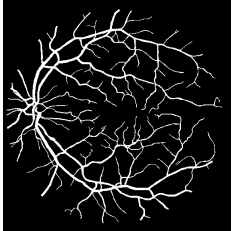

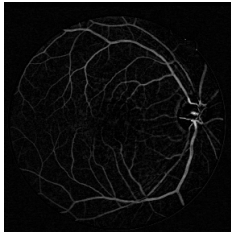
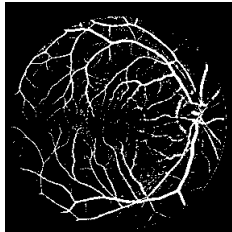
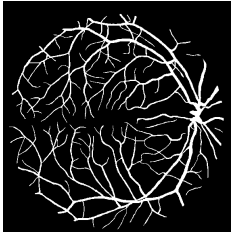


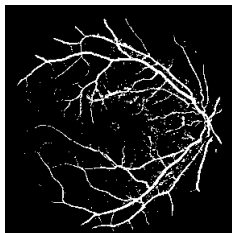
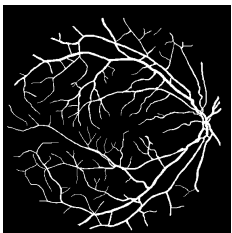

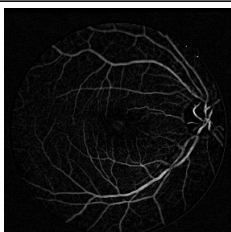
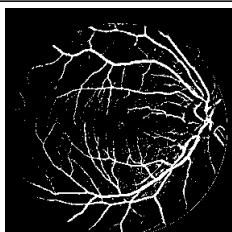
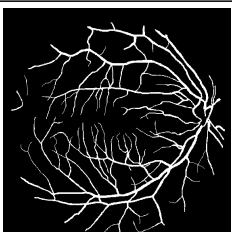


FIGURA A.9: Solución clasificador ID3. Skin Lesson

A.3.1.3. Retine Vessel

CUADRO A.8: Imágenes resultantes y soluciones encontradas por GP-IFC. Retine Ves-
sel [3] .

I. Original	GP-IFC	I. Segmentada	G.T
			
			
			
			

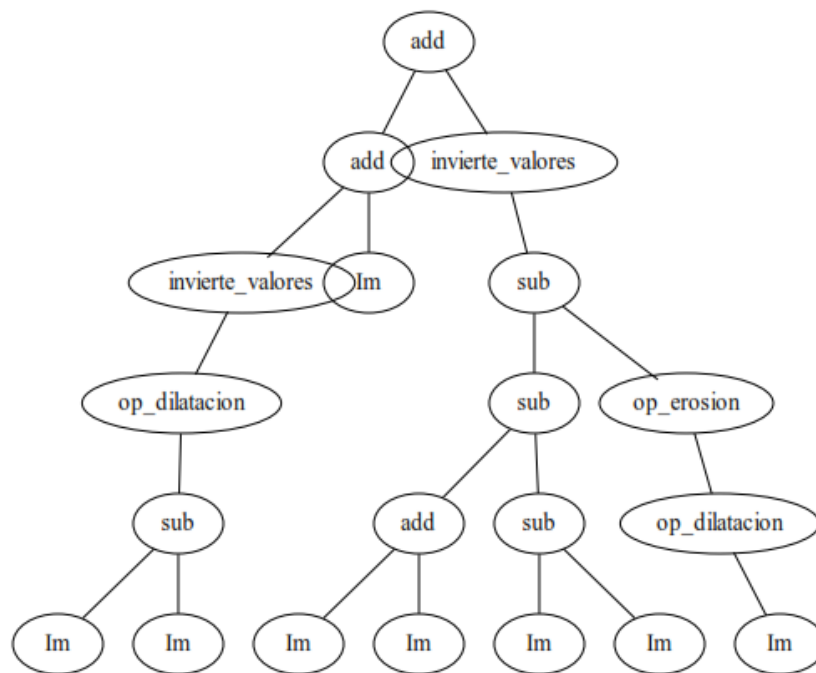


FIGURA A.10: Solución encontrada por GPI-FC. Retine Vessel

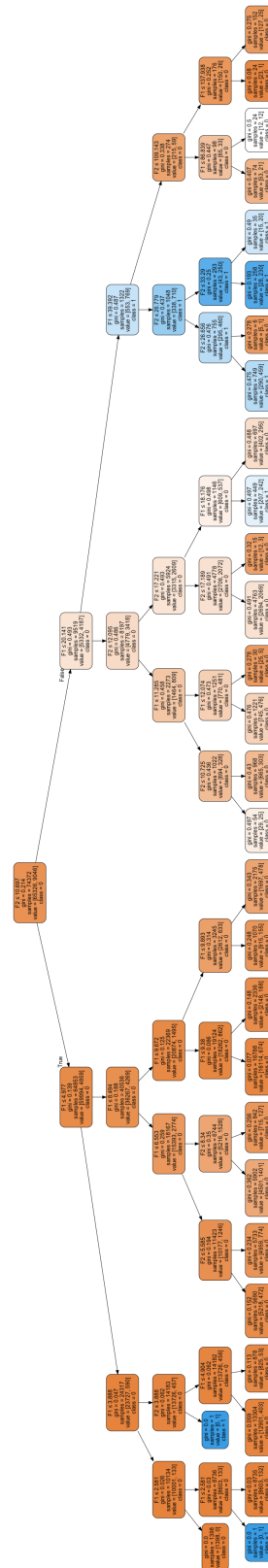
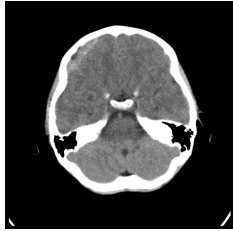

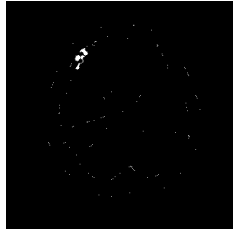

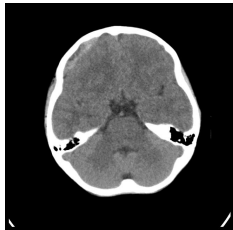
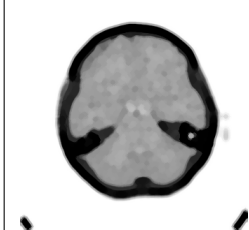



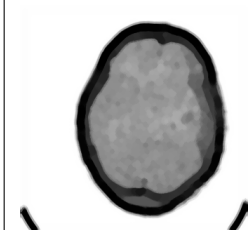


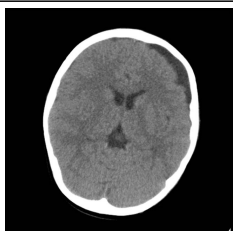
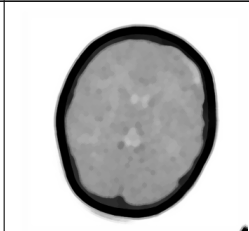

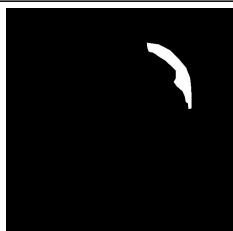


FIGURA A.11: Solución clasificador ID3. Retine Vessel

A.3.1.4. Brain CT images with Intracranial Hemorrhage

CUADRO A.9: Imágenes resultantes y soluciones encontradas por GP-IFC. Brain CT images with Intracranial Hemorrhage [5].

I. Original	GP-IFC	I. Segmentada	G.T
			
			
			
			

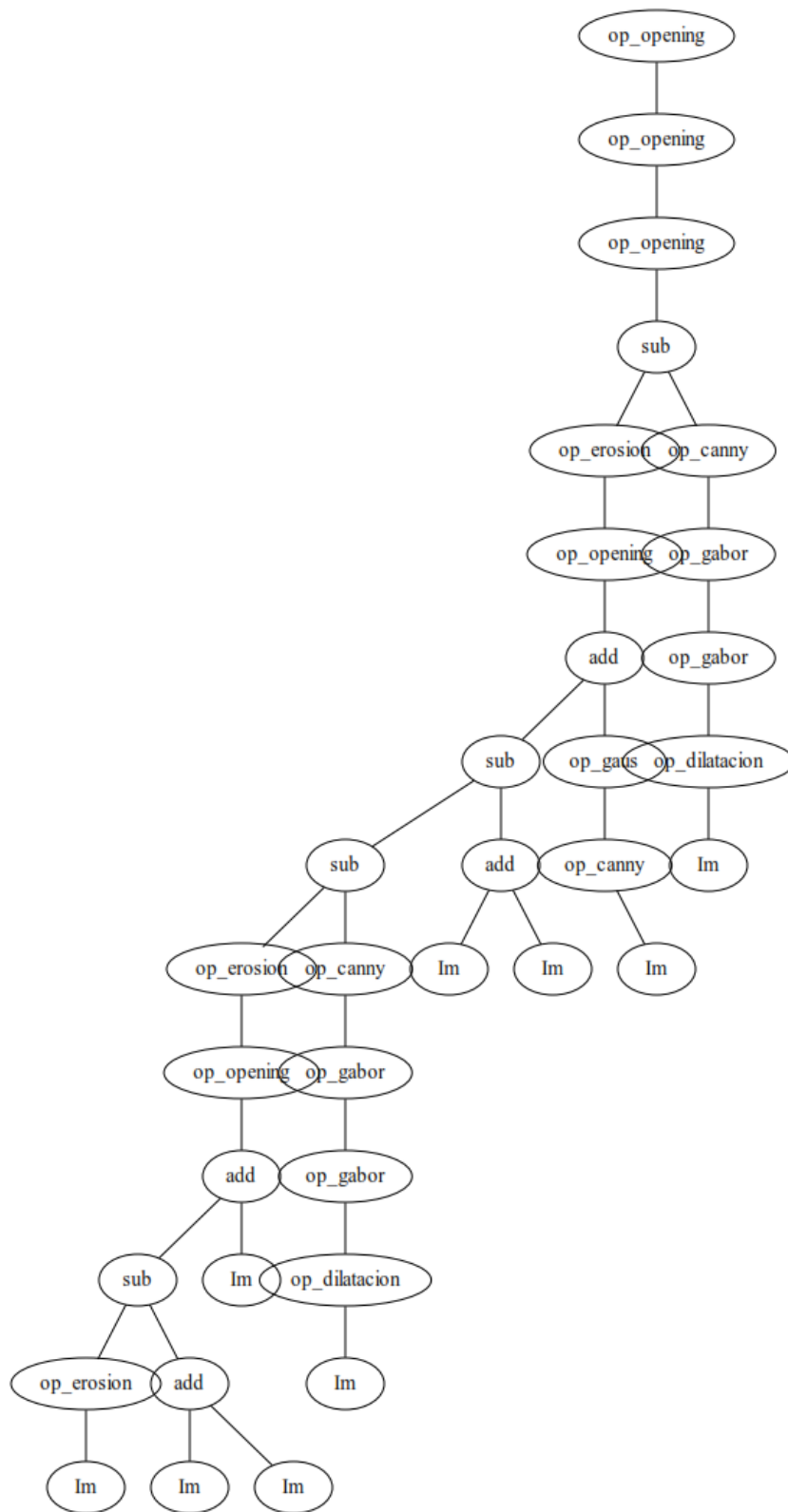


FIGURA A.12: Solución encontrada por GPI-FC. Brain CT images with Intracranial Hemorrhage

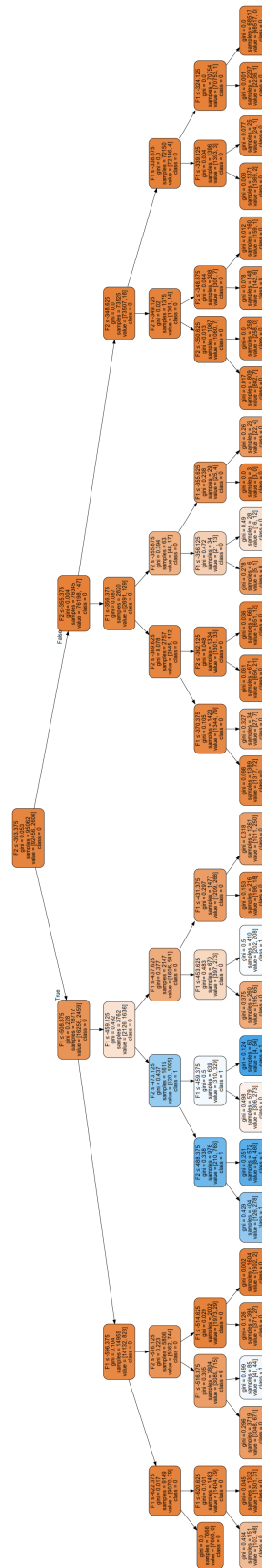


FIGURA A.13: Solución clasificador ID3. Brain CT images with Intracranial Hemorrhage

A.4. Experimento 4

Última etapa de experimentación. Se muestran ejemplos de las imágenes resultantes de cada una de las bases de datos con las que el algoritmo fue probado. También se muestra el árbol generado por PG-IFC como solución para la construcción de las características de las imágenes así como el árbol creado para realizar la binarización mediante el clasificador ID3.

A.4.1. COVID-19

G.T				
I. Segmentada				
C5				
C4				
C3				
C2				
C1				
I. Original				

CUADRO A.10: Resultados de GP-IFC. COVID-19 [2].

Solución 1	Solución 2	Solución 3	Solución 4	Solución 5

CUADRO A.11: Soluciones encontradas por cada característica construida.COVID-19

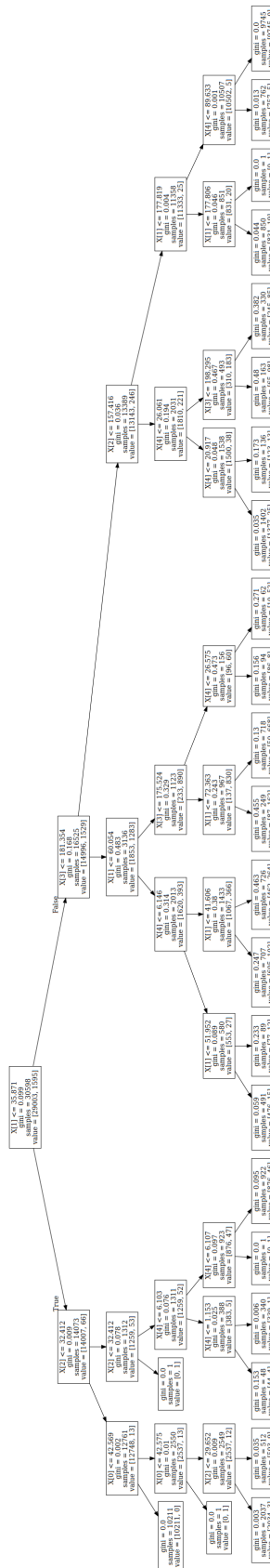
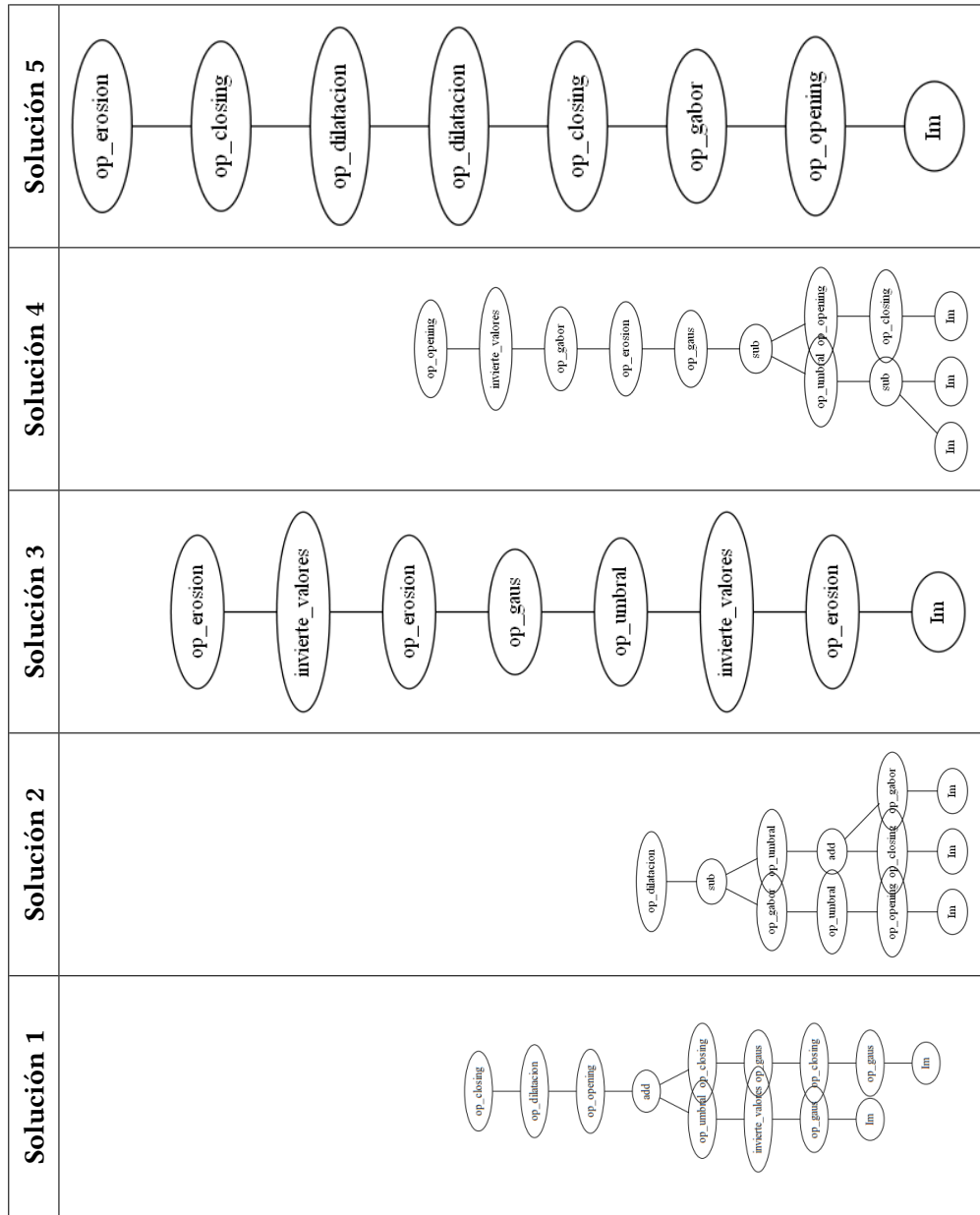


FIGURA A.14: Solución clasificador ID3. COVID-19.

A.4.2. Skin Lesson

G.T				
I. Segmentada				
C5				
C4				
C3				
C2				
C1				
I. Original				

CUADRO A.12: Resultados de GP-IFC. Skin lesson [4].



CUADRO A.13: Soluciones encontradas por cada característica construida. Skin Lesson

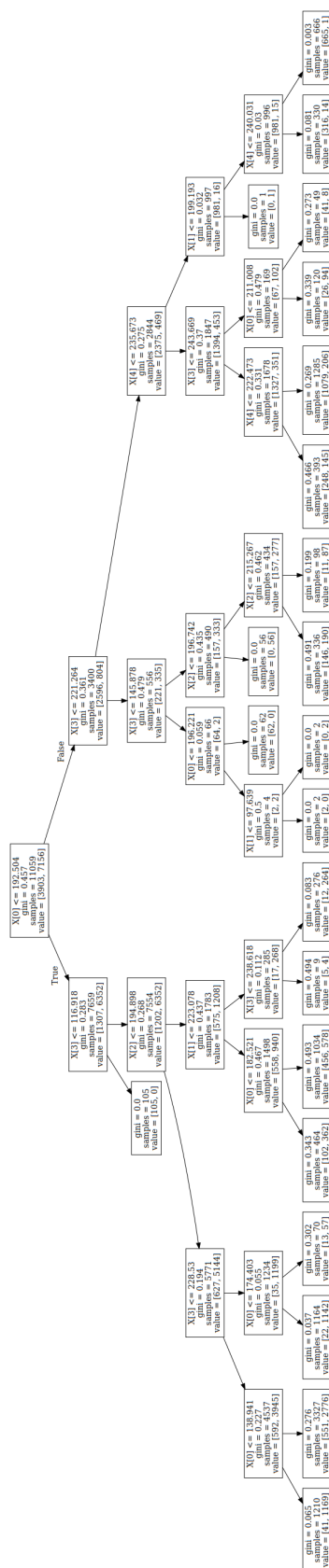
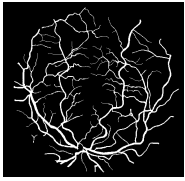
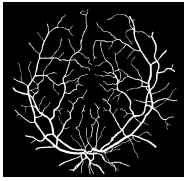
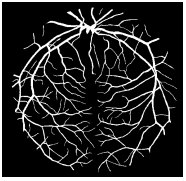
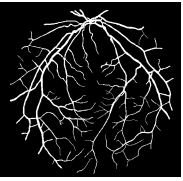
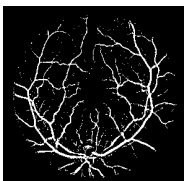
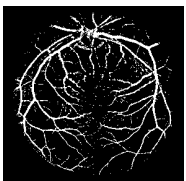
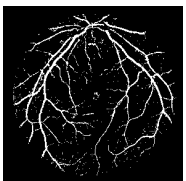
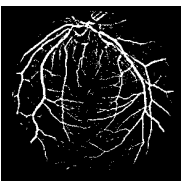

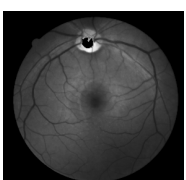

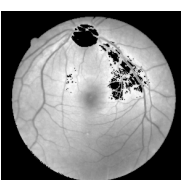
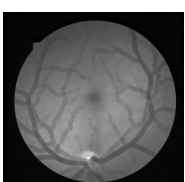
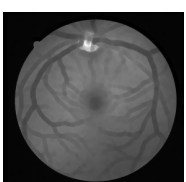

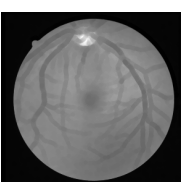
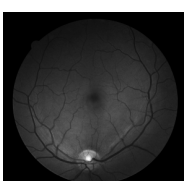
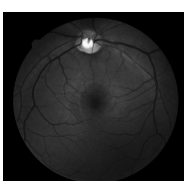

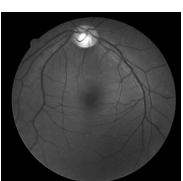
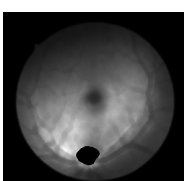
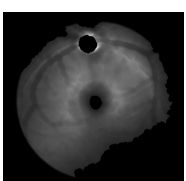
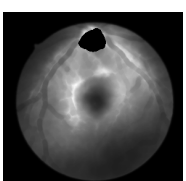
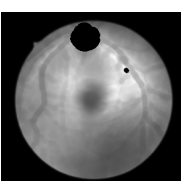
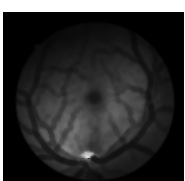
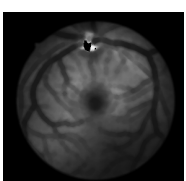
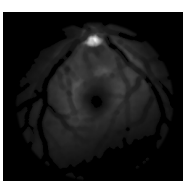
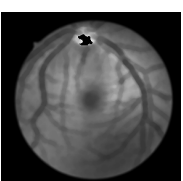






FIGURA A.15: Solución clasificador ID3. Skin lesson.

A.4.3. Retine Vessel

G.T				
I. Segmentada				
C5				
C4				
C3				
C2				
C1				
I. Original				

CUADRO A.14: Resultados de GP-IFC. Retine Vessel [3].

Solución 1	Solución 2	Solución 3	Solución 4	Solución 5

CUADRO A.15: Soluciones encontradas por cada característica construida. Retine Ves-sel.

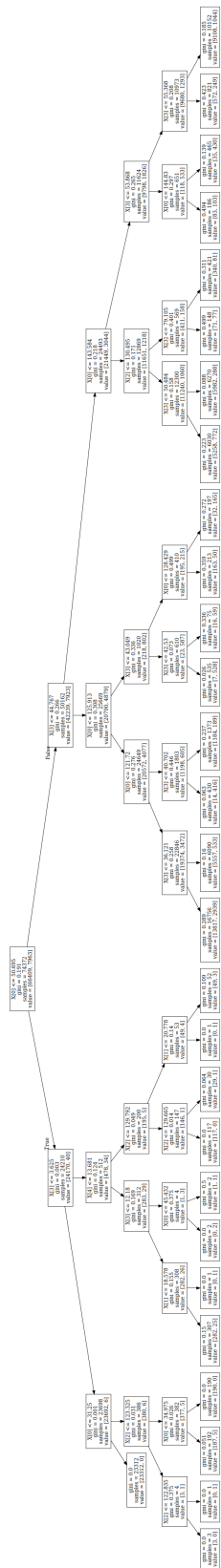
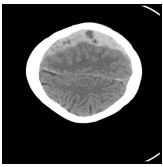
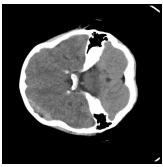
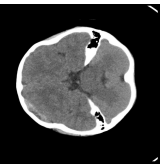
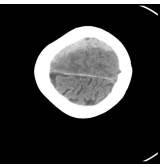

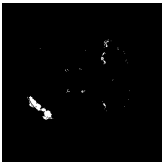
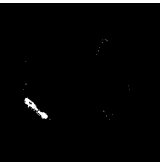

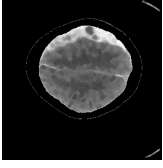
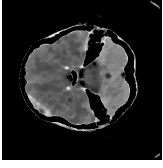
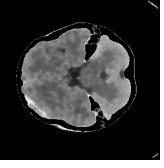
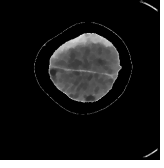
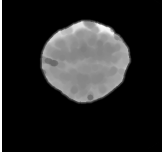
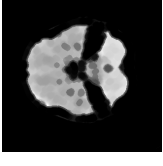
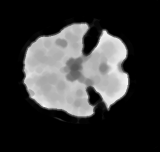
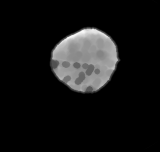
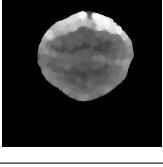
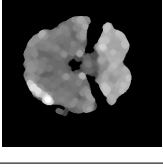
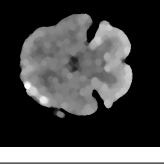
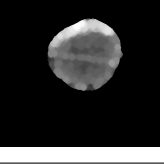
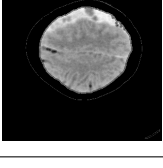
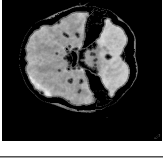

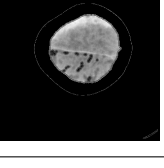
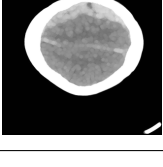


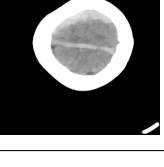
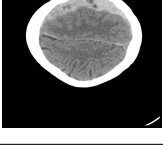

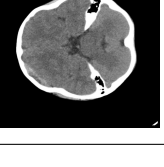
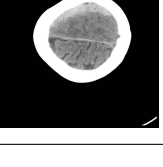
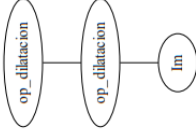
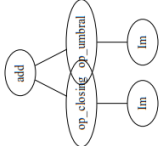
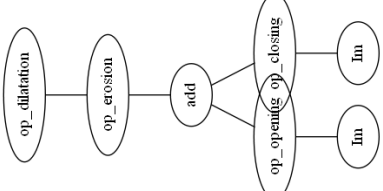
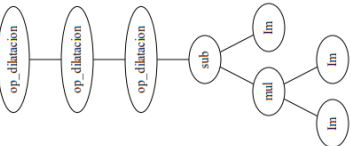
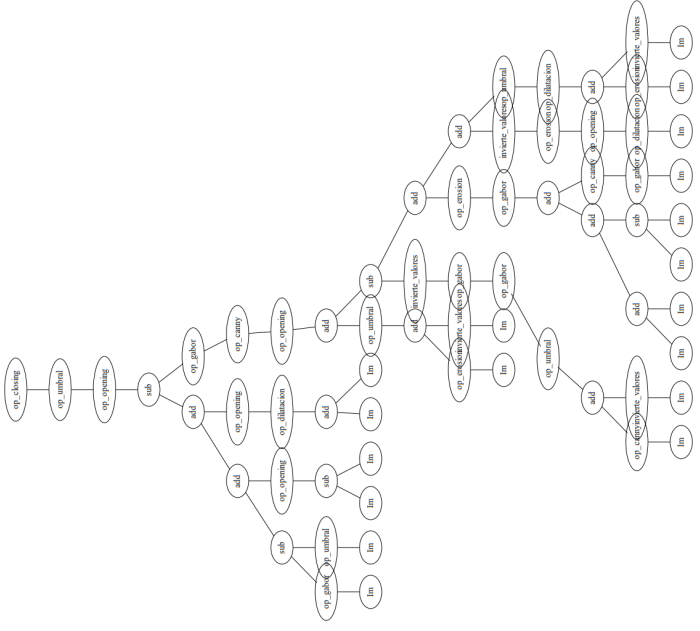


FIGURA A.16: Solución clasificador ID3. Retine Vessel.

A.4.4. Brain CT images with Intracranial Hemorrhage

G.T				
I. Segmentada				
C5				
C4				
C3				
C2				
C1				
I. Original				

CUADRO A.16: Imágenes resultantes de Brain CT images with Intracranial Hemorrhage [5].

Solución 1	Solución 2	Solución 3	Solución 4	Solución 5
				

CUADRO A.17: Soluciones encontradas por cada característica construida. Brain CT images with Intracranial Hemorrhage.

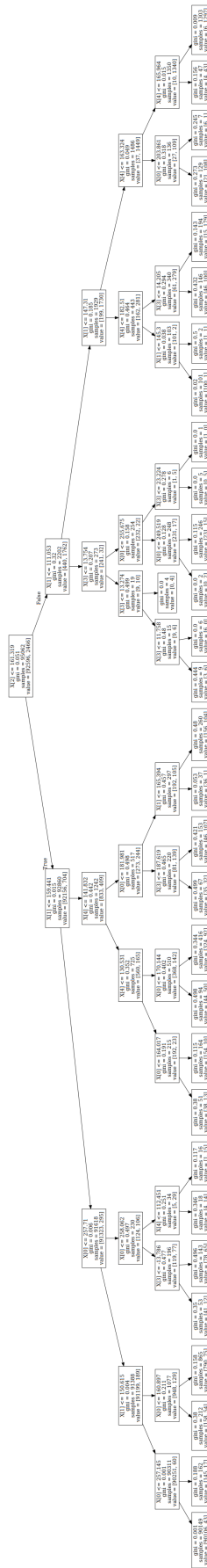

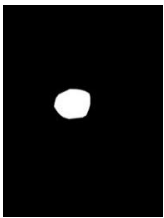
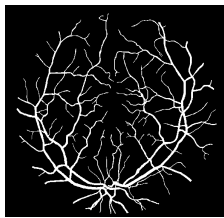
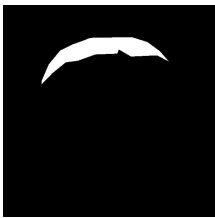




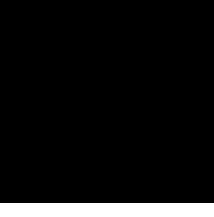



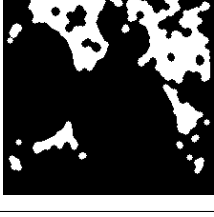


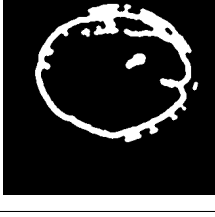




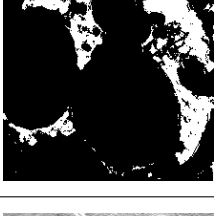

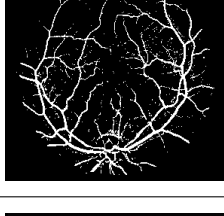

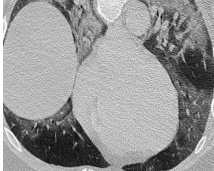
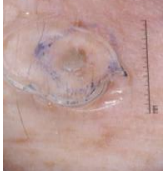




FIGURA A.17: Solución clasificador ID3. Brain CT images with Intracranial Hemorrhage.

Apéndice B

Comparación de imágenes resultantes con métodos del Estado del Arte

B.1. Ejemplo de comparación en los cuatro conjuntos de imágenes

G.T				
U-Net++				
U-Net				
TwostageGP				
StronglyGP				
GP-IFC				
I. Original				

CUADRO B.1: Comparación con métodos del estado del arte en diferentes conjuntos de imágenes. [2], [4], [3], [5]

Bibliografía

- [1] Lineamiento estandarizado para la vigilancia epidemiológica y por laboratorio de la enfermedad respiratoria viral, Sep 2020. [Online; accessed 14. May. 2021].
- [2] COVID-19 - Medical segmentation, May 2021. [Online; accessed 14. May. 2021].
- [3] DRIVE Digital Retinal Images for Vessel Extraction, May 2021. [Online; accessed 14. May. 2021].
- [4] ISIC Archive, May 2021. [Online; accessed 14. May. 2021].
- [5] MRI Hippocampus Segmentation, May 2021. [Online; accessed 14. May. 2021].
- [6] AFZALI, S., AL-SAHAF, H., XUE, B., HOLLITT, C., AND ZHANG, M. Genetic programming for feature selection and feature combination in salient object detection. In *Applications of Evolutionary Computation* (Cham, 2019), P. Kaufmann and P. A. Castillo, Eds., Springer International Publishing, pp. 308–324.
- [7] BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).
- [8] BROOKS, S. L. Computed tomography. *Dental Clinics of North America* 37, 4 (1993), 575–590.
- [9] BURGER, W., AND BURGE, M. *Digital Image Processing: An Algorithmic Introduction Using Java*. Texts in Computer Science. Springer London, 2009.
- [10] CHAKI, J., AND DEY, N. *A Beginner's Guide to Image Pre-processing Techniques*. 06 2018.
- [11] CUEVAS, E., ZALDÍVAR, D., AND PÉREZ-CISNEROS, M. *Procesamiento digital de imágenes usando MatLAB & Simulink*. Alfaomega, 2010.
- [12] EIBEN, A., AND SMITH, J. *Introduction to Evolutionary Computing*. Springer, December 2010.

- [13] FORTIN, F.-A., DE RAINVILLE, F.-M., GARDNER, M.-A., PARIZEAU, M., AND GAGNÉ, C. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research* 13 (jul 2012), 2171–2175.
- [14] KOZA, J. R. *Genetic Programming On the Programming of Computers by Means of Natural Selection*. 1992.
- [15] LIANG, J., WEN, J., WANG, Z., AND WANG, J. Evolving semantic object segmentation methods automatically by genetic programming from images and image processing operators. *Soft Computing* 24 (02 2020).
- [16] LIANG, Y., ZHANG, M., AND BROWNE, W. N. Figure-ground image segmentation using genetic programming and feature selection. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (July 2016), pp. 3839–3846.
- [17] LIANG, Y., ZHANG, M., AND BROWNE, W. N. Multi-objective genetic programming for figure-ground image segmentation. In *Artificial Life and Computational Intelligence* (Cham, 2016), T. Ray, R. Sarker, and X. Li, Eds., Springer International Publishing, pp. 134–146.
- [18] OLIVA, D. *Metaheuristic Algorithms for Image Segmentation : Theory and Applications*. Springer, 2019.
- [19] OTSU, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9, 1 (1979), 62–66.
- [20] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISSEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
- [21] PEDRINO, E. C., SAITO, J. H., KATO, E. R. R., MORANDIN, O., DEL VAL CURA, L. M., RODA, V. O., TRONCO, M. L., AND TSUNAKI, R. H. Automatic construction of image operators using a genetic programming approach. In *2011 11th International Conference on Intelligent Systems Design and Applications* (2011), pp. 636–641.
- [22] POLI, R. Genetic programming for feature detection and image segmentation. 110–125.
- [23] RAFAEL C. GONZALEZ, R. E. W. *Digital Image Processing*, 3rd ed. Prentice Hall, 2008.

- [24] RONNEBERGER, O., FISCHER, P., AND BROX, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (Cham, 2015), N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Springer International Publishing, pp. 234–241.
- [25] SHAO, L., LIU, L., AND LI, X. Feature learning for image classification via multiobjective genetic programming. *IEEE Transactions on Neural Networks and Learning Systems* 25, 7 (2014), 1359–1371.
- [26] STREATER, J. Genetic programming for the automatic construction of features in skin-lesion image classification. *Master’s thesis, University of Edinburgh* (2010).
- [27] SZELISKI, R. *Computer Vision: Algorithms and Applications*. Springer, October 2010.
- [28] TAHA, A. A., AND HANBURY, A. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. In *BMC medical imaging* (August 2015), vol. 15.
- [29] VAN DER WALT, S., SCHÖNBERGER, J. L., NUNEZ-IGLESIAS, J., BOULOGNE, F., WARNER, J. D., YAGER, N., GOULLART, E., AND YU, T. scikit-image: image processing in python. *PeerJ* 2 (2014), e453.
- [30] ZHOU, Z., RAHMAN SIDDIQUEE, M. M., TAJBAKHSI, N., AND LIANG, J. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support* (2018), Springer International Publishing, pp. 3–11.