



UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

Detección de Dientes y Paladar en Fotografías Intrabucales de Vista Palatina

Por:
Kelwin Fernández

Realizado con la asesoría de:
Carolina Chang

PROYECTO DE GRADO
Presentado ante la Ilustre Universidad Simón Bolívar
como requisito parcial para optar al título de
Ingeniero en Computación

Sartenejas, Junio de 2012



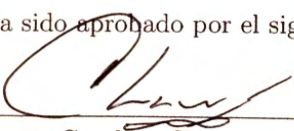
UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

ACTA FINAL PROYECTO DE GRADO

Detección de Dientes y Paladar en Fotografías Intrabucales de Vista Palatina

Presentado por:
Kelwin Fernández

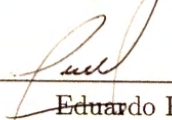
Este Proyecto de Grado ha sido aprobado por el siguiente jurado examinador:



Carolina Chang



Minaya Villasana



Eduardo Roa

Sartenejas, (17/07/12)

Resumen

El presente trabajo tiene por objetivo la detección de dientes y paladar en imágenes a color de vista palatina de la mandíbula superior (maxilar superior), es decir, segmentar las regiones de la imagen pertenecientes a dientes y paladar del resto de la imagen. Éste constituirá un primer paso en la detección de enfermedades en dicha zona. La importancia de detectar enfermedades orales radica en el hecho de que alrededor del 70 % de las personas con VIH padecen de alguna enfermedad oral, que puede indicar la presencia o el avance de este virus en SIDA.

Para lograr este objetivo se hizo uso de conceptos y técnicas de tres áreas importantes de la computación, como lo son: visión de computadoras, inteligencia artificial y algoritmos de optimización heurística, también conocidos como metaheurísticas.

Tanto la herramienta desarrollada como su presentación se descompone en dos fases. La primera consiste en la detección de dientes, en la cual la herramienta fue capaz de predecir con exactitud la pertenencia de píxeles a dientes en el 90.28 % de los casos. En esta fase se utiliza una Red Neural para la clasificación de los píxeles, en conjunto con pasos de preprocesamiento para mejorar la calidad de las imágenes de entrada y de postprocesamiento para limpiar los píxeles mal clasificados. Posteriormente, ocurre una fase de detección del paladar, la cual cuenta con una precisión de 88.66 % de píxeles bien clasificados. En esta fase se utilizan algoritmos capaces de ajustar una curva al arco definido por los dientes y un algoritmo que detecta el ancho de los dientes, con el fin de delimitar la frontera entre los dientes y el paladar. En el estudio de la solución a estos problemas se contrastó, adicionalmente al uso de Redes Neurales, el uso de Máquinas de Soporte de Vectores.

Adicionalmente, se plantean dos extensiones al proyecto. La primera extensión determina si una imagen contiene o no un paladar superior y, en caso positivo, delimitar la zona en la cual se encuentra. La segunda extensión busca detectar la frontera entre cada par de dientes, de tal forma que la descripción generada de la imagen sea más completa.

Este trabajo será presentado en el “*5th Workshop on Artificial Neural Networks in Pattern Recognition*” en la ciudad de Trento, Italia, para su posterior publicación en la serie “*Lecture Notes in Artificial Intelligence*” de la editorial Springer.

A Alan Turing,
padre de la computación y de la inteligencia artificial,
a cien años de su nacimiento.

Agradecimientos

El presente trabajo se llevó a cabo a lo largo de un año, tiempo en el cual no sólo una persona formó parte de su desarrollo. La mención de cada una de esas personas en la presente sección no se plantea por orden de importancia.

En primer lugar quiero agradecer a mis familiares por el apoyo brindado no sólo este sino durante todos mis años de formación académica.

En segundo lugar, quiero agradecer a la Universidad Simón Bolívar, por todo lo que me ha brindado, ayudándome en la formación como un buen profesional. Esperando que este trabajo cierre un ciclo que en ella se inició y sea digno de llevar su nombre.

Además, quisiera hacer llegar mi agradecimiento a la profesora Carolina Chang, por su ayuda constante en cada uno de los pasos que he dado en esta y otras investigaciones. Hacer por hacer no nos lleva a ningún lugar, debemos hacer por causar un impacto positivo en la humanidad.

Muy enfáticamente me gustaría agradecer a la Dra. Vilma Tovar del “Centro de Atención a Personas con Enfermedades Infectocontagiosas Dra Elsa La Corte”, de la Universidad Central de Venezuela. Quien proporcionó imágenes acompañadas de sus diagnósticos para el desarrollo de este trabajo.

Finalmente, dar gracias a Dios, por darme la fortaleza de alcanzar cada meta propuesta a lo largo de mi vida.

Índice general

Índice general	VII
Índice de cuadros	IX
Índice de figuras	X
1. Introducción	1
2. Marco teórico	5
2.1. Visión de Computadoras	5
2.1.1. Imágenes	6
2.1.2. Filtrado de Imágenes	7
2.1.2.1. Equalización de Histograma	7
2.1.2.2. Resaltado de Color	8
2.1.2.3. Filtro de Mediana	9
2.1.2.4. Suavizado/Resaltado de Bordes	10
2.1.2.5. Filtrado por Componentes	11
2.2. Metaheurísticas	12
2.2.1. Definición de Términos	13
2.2.2. Búsqueda Local	13
2.2.3. Búsqueda en Vecindades Variables	16
2.2.4. Búsqueda Local Iterada	17
2.3. Aprendizaje de Máquina	19
2.3.1. Máquinas de Soporte de Vectores	19
2.3.2. Redes Neurales Artificiales	20
2.3.2.1. Neurona Artificial	21

2.3.2.2.	Redes Multicapa	21
2.3.2.3.	Algoritmo “Backpropagation”	22
2.3.3.	Detección de Datos Anómalos	23
3.	Segmentación de Dientes y Paladar	25
3.1.	Segmentación de Dientes	26
3.1.1.	Preprocesamiento	27
3.1.2.	Clasificación de Píxeles	28
3.1.3.	Experimentos	29
3.2.	Segmentación de Paladar	33
3.2.1.	Ajuste de Curva	33
3.2.2.	Experimentos de Ajuste de Curva	36
3.2.3.	Decisión de Ancho de Banda	39
3.2.4.	Resultados Experimentales para la Selección del Ancho de Banda . .	40
4.	Detección de Paladar	43
5.	Segmentación Interdental	48
5.1.	Preprocesamiento	49
5.2.	Identificación de Fronteras	51
5.3.	Resultados Experimentales	51
6.	Conclusiones y recomendaciones	53
	Bibliografía	57
A.	Otros Filtros de Imágenes	60
A.1.	Filtro Homomórfico	61
A.2.	Filtro de Umbral	62
B.	Descripción General de la Implementación	65
C.	Sistema de Clasificación de Imágenes	67

Índice de cuadros

3.1. Máquina de Pruebas	29
3.2. Clasificación de píxeles con SVM y ANN	30
3.3. Clasificación de píxeles sobre formatos combinados con SVM y ANN	31
3.4. Filtrado de Componentes	32
3.5. Ajuste de Curvas sobre Imágenes Clasificadas de Forma óptima	36
3.6. Ajuste de Curvas sobre Imágenes Clasificadas por la Red Neural	37
3.7. Limpieza Iterativa de Píxeles Ruidosos. Número de iteraciones promedio y máximo hasta alcanzar punto fijo.	38
3.8. Limpieza Iterativa de Píxeles Ruidosos	38
3.9. Mejora general mediante el ajuste de curva	39
3.10. Variación en la clasificación de dientes por la selección del ancho de banda .	41
4.1. Detección de Paladar	44
4.2. Clasificación de Píxeles con Detección de Paladar	47
5.1. Segmentación Interdental	52

Índice de figuras

1.1. Imágenes con Enfermedades	2
2.1. Algoritmo de Equalización de Histograma	7
2.2. Equalización de Histograma	8
2.3. Filtro de Color	9
2.4. Filtro de Mediana	10
2.5. Suavizado/Realzado de Bordes	11
2.6. Filtrado de Componentes	12
2.7. Algoritmo de Búsqueda Local	14
2.8. Representación Gráfica del Algoritmo de Búsqueda Local	15
2.9. Representación Gráfica del Algoritmo de Búsqueda en Vecindades Variables .	16
2.10. Algoritmo Búsqueda en Vecindades Variables	17
2.11. Representación Gráfica del Algoritmo de Búsqueda Local Iterada	18
2.12. Algoritmo Búsqueda Local Iterada	19
2.13. Red Neural	22
2.14. Algoritmo Backpropagation	23
2.15. Algoritmo de Detección de Datos Anómalos	24
3.1. Fotografías de Interés	25
3.2. Detección óptima de Dientes	26
3.3. Aplicación de Filtros de Color	27
3.4. Detección de Dientes usando Redes Neurales	29
3.5. Selección de Umbral para Clasificación de Píxeles	32
3.6. Ajuste óptimo de Curva	33
3.7. Zona de Error entre un Par de Curvas	36
3.8. Ajuste de Curvas	38

3.9. Selección de Ancho de Banda.	41
3.10. Selección de Ancho de Banda y Segmentación Diente/Paladar	42
4.1. Imágenes sin Paladar	44
4.2. Falsos Negativos en Detección de Paladar	45
4.3. Falsos Positivos en Detección de Paladar	45
4.4. Imágenes con Paladar en Menor Escala	46
5.1. Aplanado de Curva	49
5.2. Problema con Aplanado de Curva	50
5.3. Dientes Segmentados	52
A.1. Filtro homomórfico	62
A.2. Porcentaje de Clasificación utilizando distintos umbrales	62
A.3. Porcentaje de Clasificación utilizando distintos umbrales	63
B.1. Logotipo de OpenCV	65
B.2. Diagrama de Flujo de la Tarea de Clasificación de Píxeles como Diente/No Diente	66
B.3. Diagrama de Flujo de la Tarea de Segmentación del Paladar	66
C.1. Herramientas Utilizadas. Python (izquierda), Django (centro), PostgreSQL (derecha).	67
C.2. Diagnóstico de Imágenes (Arriba). Carga de Nuevas Imágenes (Abajo). . . .	68

Capítulo 1

Introducción

Treinta y cuatro millones de personas tenían Virus de Inmunodeficiencia Humana (VIH) a fines del año 2010 de acuerdo a la Organización Mundial de la Salud [WHO]. Un 93.5 % de esas personas residen en países en vías del desarrollo y sólo un 47 % de pacientes en este subgrupo reciben el tratamiento antiretroviral que necesitan.

La evidencia sugiere que alrededor del 70 % de estas personas con VIH tienen enfermedades orales, incluyendo Leucoplasia Velloso, Sarcoma de Kaposi y Candidiasis [C.E88]. El diagnóstico de estas enfermedades es muy importante porque algunas de ellas pueden indicar la evolución del VIH hacia el Síndrome de Inmunodeficiencia Adquirida (SIDA) [AT98]. Tristemente, el número de centros de atención odontológica que tratan enfermedades infectocontagiosas en países en vías de desarrollo es limitado, en muchos casos, insuficiente.

Debido a sus características, es posible inferir que algunas enfermedades orales que indican la presencia de VIH/SIDA pueden ser diagnosticadas automáticamente. Una herramienta computacional de bajo costo para detectar enfermedades orales infectocontagiosas puede conducir a una mejora en el cuidado de la salud, especialmente en países de bajos y medianos ingresos.

Este trabajo se enfoca en el problema de segmentar los dientes y el paladar superior en fotografías de vista palatina de la mandíbula superior (maxilar superior) como un paso inicial hacia el diagnóstico de enfermedades orales. El paladar es un área importante y de gran tamaño de la boca donde diversas enfermedades pueden ser observadas. Por ejemplo, en la figura 1.1 se muestran casos de pacientes con enfermedades orales tales como Candidiasis, Hepatitis y Sarcoma de Kaposi.

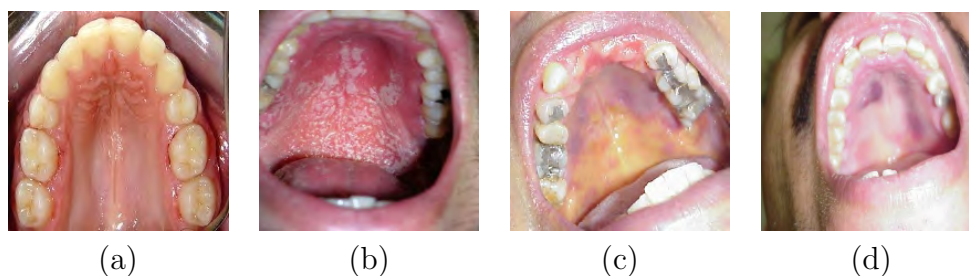


Figura 1.1: (a) Paladar saludable. Enfermedades orales: (b) Candidiasis, (c) Hepatitis, (d) Sarcoma de Kaposi.

Los trabajos previos en el uso de imágenes bucales con fines médicos están principalmente basados en el uso de radiografías. Por ejemplo, se tienen los trabajos descritos a continuación, que tienen como objetivo la segmentación interdental en radiografías dentales para identificación postmortem [Nas06] [LH09].

En [Nas06] un método basado en redes neurales es propuesto para generar un sistema de identificación postmortem mediante el pareamiento de características de la imagen extraída de radiografías dentales. El sistema intenta hacer coincidir radiografías post-mortem y ante-mortem de una persona, proponiendo el uso de características aprendibles inherentes de imágenes dentales para comparar diente por diente.

En [LH09] es propuesto un sistema de clasificación y etiquetado dental que segmenta, clasifica e identifica cada diente en radiografías de mordida. Las radiografías son previamente mejoradas para aislar las regiones de interés usando filtros de imágenes. Una vez que los

dientes son aislados, una Máquina de Soporte de Vectores [CZ01, Vap98] clasifica cada diente como molar o premolar.

Ambos sistemas están basados en radiografías, en contraposición se tiene en [LC10] el problema de segmentación de labios en fotografías de color. Manejado mediante el ajuste de un modelo Gaussiano a través de la información de los segmentos de los labios en imágenes. Dicho trabajo se realiza como paso previo para la generación de un sistema de lectura de labios.

Las radiografías tienen la ventaja de ser tomadas bajo estándares comunes, con máquinas de baja variabilidad pero proveen menos información acerca de enfermedades comparadas con fotografías de vista palatina. Por otro lado, las fotografías son altamente sensitivas a fenómenos tales como iluminación, calidad de la imagen, punto de vista, falta de estándares, entre otros. Comparando los costos de obtención, las fotografías son de mayor acceso para el público en general, además de poder ser tomadas por usuarios no relacionados directamente al campo médico, expandiendo las posibilidades de implantación del proyecto en sectores con bajos recursos y alta población. Por estos motivos, en el presente proyecto se trabaja con fotografías a color, teniendo como precondition que cumplen con ser fotografías de vista palatina, tales como las mostradas en la figura 1.1.

El objetivo general de este proyecto es brindar una descripción general de una imagen de vista palatina, con el fin de servir de apoyo a sistemas de detección de enfermedades. El interés en este aspecto es la contextualización de la imagen, permitiendo así un análisis más profundo de ésta, conociendo de antemano las estructuras presentes en ella.

Con el fin de alcanzar el objetivo general de este proyecto se plantean los siguientes objetivos específicos:

1. Segmentar las zonas de la imagen en la que se encuentran los dientes y aquellas en las que se encuentra el paladar.

2. Generar una solución preferiblemente de bajos costos y de fácil acceso con el fin de subsanar los problemas de movilización de los pacientes y de sus costos generados.
3. Evaluar los resultados obtenidos por el sistema, para poder brindar a las próximas fases información acerca de la precisión del sistema.

El presente trabajo se encuentra dividido en 6 capítulos, incluida la introducción. El capítulo 2 expone las definiciones necesarias para comprender los aspectos fundamentales de la implementación dada, la cual se presenta en el capítulo 3. Los capítulos 4 y 5 muestran dos extensiones al presente trabajo, en el capítulo 4 se muestra un procedimiento para romper con la precondition establecida, en la cual se exige que las imágenes de entrada sean fotografías de vista palatina y en el capítulo 5 se presenta un método para segmentación diente a diente en este tipo de imágenes. Finalmente, el capítulo 6 contiene las conclusiones y recomendaciones, así como los trabajos futuros a ser realizados con base en el siguiente trabajo.

Capítulo 2

Marco teórico

Este capítulo presenta los conceptos fundamentales necesarios para el desarrollo de la presente investigación. El capítulo inicia con la definición del área de Visión de Computadoras, abarcando las técnicas y conceptos de interés para este trabajo. Posteriormente, se describen los algoritmos utilizados en la implementación de la presente investigación, siendo éstos separados en dos bloques: algoritmos metaheurísticos y algoritmos de aprendizaje de máquina. Del primer bloque, se describen los algoritmos de Búsqueda Local, Búsqueda Local Iterada y Búsqueda en Vecindades Variables, todos ellos como técnicas de mejoramiento de soluciones. Del segundo bloque, correspondiente al Aprendizaje de Máquina, se describen los enfoques de Redes Neuronales y de Máquinas de Soporte de Vectores, así como un algoritmo especializado en la detección de datos anómalos en una muestra.

2.1. Visión de Computadoras

Se considera *Visión de Computadoras* (CV por sus siglas en inglés) como el campo de la computación cuyo objetivo consiste en realizar decisiones útiles sobre objetos físicos reales y

escenas basadas en imágenes censadas [SS01]. Para esto, es necesario extraer información de la imagen que permita construir una descripción de la misma.

Por otro lado, muchos investigadores dicen que el objetivo fundamental de la visión de computadoras es construir una descripción de una escena proveniente de imágenes. En esta última definición el proceso previo de obtención de las imágenes en cuestión no es considerado parte fundamental del campo. Autores como Batchelor y Waltz [BW01] consideran esto como una parte ajena a la visión de computadoras, incluyéndolo en contraposición al campo de la Visión de Máquinas.

Si bien el aspecto de la recolección de las imágenes es importante y tiene por sí mismo problemas propios, el presente trabajo se aloja en el campo de la visión de computadoras mas que en el de máquinas. Aún cuando el proceso de captura de datos fue contemplada para su desarrollo.

2.1.1. Imágenes

Se puede definir una imagen F como una matriz con un número fijo de filas (m) y columnas (n). Sea $0 \leq i < m$ y $0 \leq j < n$, se define $F(i, j)$ como la información de una posición en F denominada *píxel* [BW01].

El tipo de retorno de $F(i, j)$ dependerá del formato de la imagen. De esta forma un píxel puede tener un valor escalar, regularmente entre 0 y 255 (por su representación en un byte) correspondiente a la escala de grises o una tupla correspondiente a cada una de las componentes del formato respectivo.

En general, se utilizará en el futuro la notación $g(i, j)$ para expresar la aplicación de la función g al píxel ubicado en la posición (i, j) de la imagen.

2.1.2. Filtrado de Imágenes

Tanto en fases posteriores como en fases previas del procesamiento de imágenes, éstas deben ser mejoradas. En esta mejora, es posible reducir el ruido y resaltar o suprimir estructuras de interés para el problema particular.

En general, se puede decir que un operador de mejora de imágenes facilita la detección de detalles o estructuras importantes en éstas para un hombre o máquina [SS01].

A continuación se define una serie de operadores de filtrado de imágenes.

2.1.2.1. Equalización de Histograma

Sea una imagen F en escala de grises, es decir, cada píxel es un escalar comprendido en el rango entre 0 y 255, ambos extremos inclusive. Si se define un nivel de gris x como cada uno de los valores en el rango $[0 \dots 255]$, donde el número de píxeles de dicho nivel se define como la cantidad de píxeles (i, j) que cumplen con que $F(i, j) = x$, un operador de Equalización de Histograma busca que la imagen resultante F' contenga todos los niveles disponibles de gris y que cada nivel tenga aproximadamente igual número de píxeles que los otros [SS01].

El algoritmo utilizado en el presente trabajo para la equalización de histograma es propuesto por la librería de Visión de Computadoras *OpenCV* [OCV], el cual se puede observar en la figura 2.1:

<pre> 1 Calcular el histograma H de F. 2 Normalizar el histograma para que la suma de sus elementos sea 255. 3 Calcular la integral del histograma: $H'(i) = \sum_{0 \leq k < i} H(k)$ 4 Transformar la imagen como $F'(i, j) = H'(F(i, j))$. </pre>

Figura 2.1: Algoritmo de Equalización de Histograma

Dicha transformación tiene como resultado colateral el incremento del brillo y del contraste de las imágenes. Es importante destacar que el operador es monádico, es decir, la

transformación de cada píxel depende únicamente de su información, sin considerar la de sus vecinos.

En la figura 2.2 se puede apreciar el resultado de aplicar este filtro sobre una imagen. En la imagen resultante, el contraste se ha incrementado considerablemente, esto permite resaltar estructuras que en la imagen original no se distinguían. Además, se puede apreciar como el histograma es mucho más homogéneo que el original.

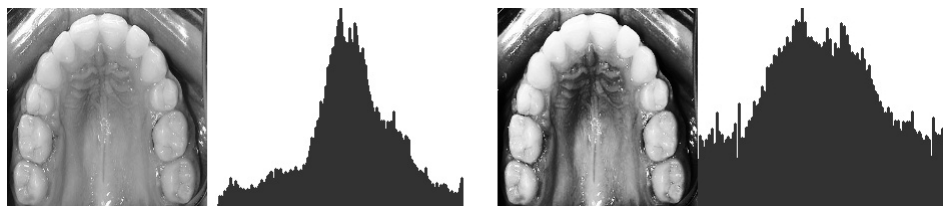


Figura 2.2: Izquierda: imagen original con su histograma. Derecha: imagen equalizada con su histograma.

2.1.2.2. Resaltado de Color

El color es un factor determinante en la detección de ciertas estructuras en imágenes. Sin embargo, es posible que los colores de una imagen no sean fácilmente diferenciables, por lo que es necesario resaltarlos.

Comúnmente, las imágenes en color vienen dadas en formato RGB. Al aplicar los operadores clásicos de imágenes en escala de grises, tales como equalización de histograma y ajuste de contraste, sobre cada componente del formato por separado (rojo, verde y azul), es posible que nuevos colores no deseados se generen en la imagen [XO07]. Por lo tanto, se transforman las imágenes a formato HSV. Dicho formato tiene la ventaja de separar la información de color en la componente H, de la información de saturación y valor de luminosidad, los cuales se encuentran en las componentes S y V respectivamente.

Xiao&Ohya propusieron un filtro de incremento del contraste en imágenes de color [XO07]

basado en el razonamiento pasado. Ellos manejan el problema de aparición de colores inválidos en la imagen resultante manteniendo la componente del matiz (H) inmutable. La componente de luminosidad es mejorada y a la componente de saturación se le aplica una equalización de su histograma.

Finalmente, las imágenes son restituidas al formato RGB. Este filtro resulta especialmente útil en imágenes opacas donde los colores son difícilmente distinguibles. La figura 2.3 muestra el resultado de aplicar este filtro sobre una imagen. Note como diversas estructuras del paladar terminan siendo más fácilmente identificables en la imagen resultante.



Figura 2.3: Izquierda: imagen original. Derecha: imagen con filtro de color propuesto por Xiao&Ohya.

2.1.2.3. Filtro de Mediana

Comúnmente, una imagen I tiene ruido, proveniente de diversos medios, que analizados individualmente pueden llevar a tomar decisiones incorrectas. Un mejor estimado de un píxel $I(x, y)$ es el valor de la mediana de su vecindad [SS01]. Definiendo vecindad como una ventana local de tamaño fijo alrededor de éste, la forma de la ventana es libre, sin embargo, regularmente se utilizan ventanas cuadradas o circulares.

En la figura 2.4 se puede ver el resultado de aplicar un filtro de mediana sobre una imagen en escala de grises. Mientras mayor es la escala, el desenfoque de la imagen es mayor.

Si la vecindad es muy grande, computar el valor de la mediana en ésta puede ser compu-

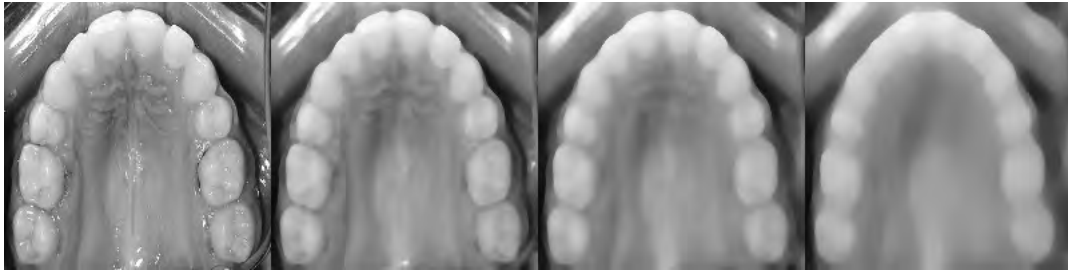


Figura 2.4: Aplicación de filtro de mediana con ventanas 1, 5, 9 y 17 respectivamente.

tacionalmente costoso, aún los algoritmos lineales tienen constantes elevadas, por lo que se suelen utilizar ventanas pequeñas. En caso de ser necesaria la utilización de ventanas de mayor tamaño, en vez de utilizar como estimador la mediana, se puede utilizar la media si se aplica sobre una vecindad homogénea [SS01].

2.1.2.4. Suavizado/Resaltado de Bordes

En muchos problemas, puede resultar de interés suavizar o resaltar los bordes de una imagen. Sea I la imagen sobre la cual se quiere aplicar el filtro e I' la imagen resultante. La ecuación 2.1 define el proceso de suavizado de bordes, mientras que en la ecuación 2.2, se muestra el proceso inverso (resaltado de bordes) [GW07].

$$I' = I + (\mathcal{M}(I, w) - I) \quad (2.1)$$

$$I' = I + (I - \mathcal{M}(I, w)) \quad (2.2)$$

En ambas ecuaciones, $\mathcal{M}(I, w)$ denota el resultado de aplicar el filtro de mediana expuesto en 2.1.2.3, sobre la imagen I , con una ventana de tamaño w .

En la figura 2.5 se puede observar el resultado de aplicar ambos filtros sobre una imagen. En el caso del suavizado de bordes, se aprecia cómo los bordes de dientes y relieves del paladar

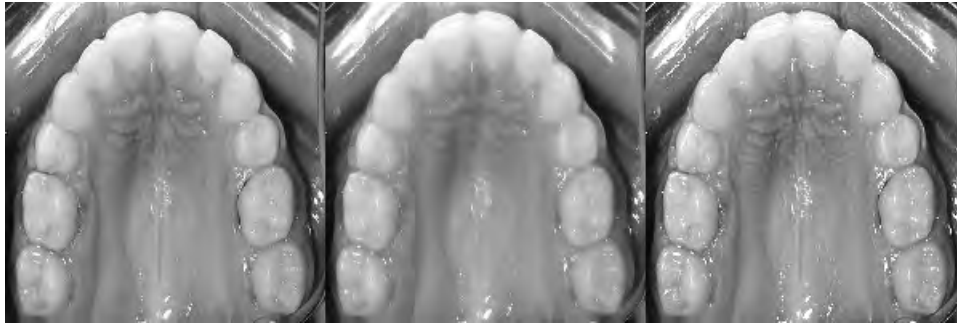


Figura 2.5: De izquierda a derecha: imagen original, suavizado de bordes, realzado de bordes.

se suprimen mientras que en el resultado de aplicar resaltado de bordes se exacerban estas características de la imagen.

2.1.2.5. Filtrado por Componentes

Sea I una imagen binaria, es decir, el valor de cualquier píxel es 0 o 1. Dicha imagen puede ser el resultado de un proceso de clasificación de cada píxel en la imagen. Es posible que dicha imagen tenga puntos de ruido en ella que deban ser eliminados. Una forma de eliminarlos es utilizando el filtro de mediana estudiado anteriormente. Sin embargo, si se conoce la estructura del objeto a evaluar en la imagen, es posible descartar grupos de píxeles adyacentes de pequeño tamaño. Dichos grupos de píxeles se conocen también como componentes conexas de la imagen o como “huecos” [SS01].

Una forma de filtrar los pequeños grupos de píxeles que se encuentran en una imagen es intercambiar su color. Si en una imagen existe un pequeño grupo de píxeles con valor 1 rodeado de un gran grupo con valor 0, se puede asignar a cada píxeles del primer grupo el valor 0. Análogamente, si se consigue un pequeño grupo con valor 0, es posible asignarle 1. Este algoritmo puede ser generalizado para n colores, sin embargo, se considera la versión binaria por ser de interés en este trabajo. El tamaño de una componente se define por el número de píxeles que pertenecen a él, una forma de decidir la frontera entre grupos pequeños y grupos

grandes es establecer un umbral.

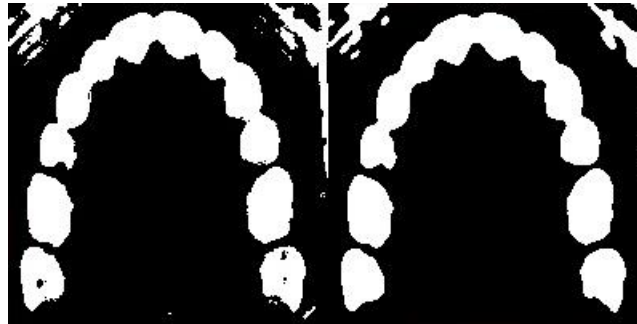


Figura 2.6: Izquierda: imagen original, Derecha: filtrado de componentes con umbral 10.

En la figura 2.6 se puede apreciar el resultado de aplicar este algoritmo, note la reducción del ruido en la imagen derecha, con respecto a la imagen original a la izquierda, especialmente en la zona superior, donde se encontraban inicialmente pequeñas estructuras.

2.2. Metaheurísticas

Las metaheurísticas son métodos que optimizan soluciones a un problema, regularmente de naturaleza discreta, buscando mejorar iterativamente una solución candidata con la ayuda de una medida de calidad [GPJ10].

Las metaheurísticas combinan técnicas de mejora local y estrategias de diversificación que buscan escapar de óptimos locales y explorar el espacio de búsqueda [GPJ10]. Éstas, buscan siempre mantener el mecanismo simple y general, de tal forma que, a diferencia de las heurísticas, puedan ser aplicadas a una gran variedad de problemas.

Si bien no existe seguridad de alcanzar la solución óptima en la mayoría de las metaheurísticas bajo una función de calidad cualquiera, resolver el problema usando un mecanismo exacto que garantice optimalidad suele ser infactible debido al tiempo que toma encontrar dicha solución.

Generalmente se clasifican según su comportamiento en tres tipos: de trayectoria, constructivas y poblacionales. Para el problema a resolver en este trabajo, son de especial interés las de trayectoria debido a su comportamiento. Las metaheurísticas de trayectoria parten de una solución inicial y buscan optimizarla aplicando sobre ésta variaciones.

2.2.1. Definición de Términos

Sea Ω el espacio de soluciones de un problema discreto. Sea $f : \Omega \rightarrow \mathbb{R}$ una función objetivo que determina la calidad de una solución [GPJ10] [AL97]. Se supone sin pérdida de generalidad, que f tiene la propiedad de que es menor mientras la solución sea de mayor calidad, es decir, esté más cercana al óptimo.

El objetivo es conseguir la solución $w^* \in \Omega$, tal que $f(w^*) \leq f(w) \forall w \in \Omega$. Se define una función de vecindad de una solución ($N(w) \subseteq \Omega$). Dicha vecindad, define el conjunto de posibles soluciones resultantes de realizar un paso de optimización a una solución w [GPJ10] [AL97].

2.2.2. Búsqueda Local (Local Search - LS)

Se puede definir la Búsqueda Local (LS) como un mecanismo iterativo, que parte de una solución inicial w . En cada paso, explora la vecindad $N(w)$ y bajo cierto criterio de aceptación se mueve hacia alguno de los vecinos [GPJ10]. Regularmente el criterio de aceptación viene dado por un algoritmo voraz en la forma de escoger el mejor de toda la vecindad (mejor-mejor) o el primer mejor al candidato de solución actual (primer mejor). Este proceso se repite hasta cumplirse un criterio de parada, el cual puede venir dado por un número máximo de iteraciones permitidas o hasta converger. En la figura 2.7 se muestra el pseudocódigo de este algoritmo, dejando cuatro aspectos libres: obtención de la solución inicial, criterio de parada, exploración de la vecindad y el criterio de aceptación. El primer y el tercer punto se dejan

abiertos para brindar poder de adaptación a distintos problemas. Variaciones del criterio de parada y de aceptación llevarán a distintas versiones de este algoritmo.

```

1 S := solucion_inicial()
2 S* := S
3 while( !criterio_de_parada() ){
4     S_vecina := explorar_vecindad(S)
5     if ( criterio_aceptacion(S, S_vecina) ){
6         S := S_vecina
7     } else {
8         break
9     }
10
11     if ( f(S) < f(S*) ){
12         S* := S
13     }
14 }
15 return S

```

Figura 2.7: Pseudocódigo del algoritmo Búsqueda Local

- **Solución Inicial:** dado que LS se comporta como una metaheurística de trayectoria, debe partir de una solución base. Esta solución puede ser dada aleatoriamente, con algún método voraz que genere soluciones iniciales buenas o por algún otro algoritmo, posiblemente una metaheurística.
- **Criterio de Parada:** dado que, en general, no es posible determinar en qué momento se ha encontrado un óptimo global, es necesario definir un criterio de parada, regularmente se establece un máximo número de iteraciones o verificación de convergencia.
- **Exploración de la Vecindad:** diversas formas de explorar la vecindad generan distintas alternativas de LS. Dos alternativas comúnmente utilizadas es seleccionar el mejor de toda la vecindad y otra es seleccionar el primer vecino mejor a la solución actual.
- **Criterio de Aceptación:** define bajo qué circunstancias la solución vecina es aceptada y se procederá a intensificar la solución a través de ella. Generalmente se acepta

si la solución vecina tiene función objetivo menor (o igual) a la solución actual. Otra alternativa es aceptar bajo cierta probabilidad una solución de menor calidad, sin embargo, esto conduce a mecanismos de diversificación que no son utilizados en las formas básicas de búsqueda local.

Por ejemplo, en la figura 2.8 se muestra la trayectoria que efectúa LS en un proceso de optimización. Tal como se definió, se supone que se busca minimizar la función objetivo.

La solución inicial se denota S_0 , mientras que la solución alcanzada después del proceso de búsqueda local se denota S^* . Se puede observar que si bien el valor de $f(S^*)$ es inferior a $f(S_0)$, éste no es el óptimo.

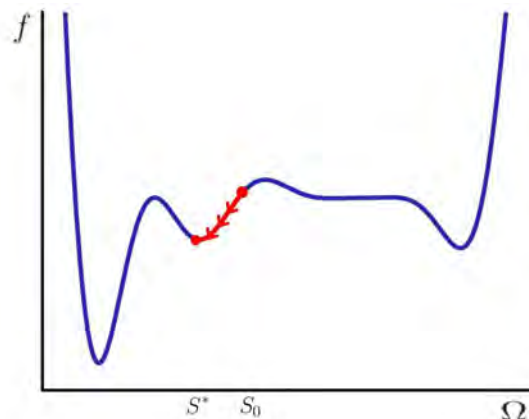


Figura 2.8: Representación gráfica del funcionamiento del algoritmo de Búsqueda Local.

LS presenta el problema de poder quedarse fácilmente atascado (converger) en óptimos locales [AL97], también denominados valles. En la figura 2.8 el algoritmo no será capaz de salir de dicho punto, dado que busca siempre mejorar la solución actual, por lo que muchos autores consideran que LS no entra en la definición de metaheurística al no incluir mecanismos de diversificación. Se considera en la presente sección dado que es utilizado como un subproceso de optimización de muchas metaheurísticas.

2.2.3. Búsqueda en Vecindades Variables (Variable Neighborhood Search - VNS)

Búsqueda en Vecindades Variables es una metaheurística para resolver problemas combinatorios de optimización cuya idea es alternar sistemáticamente la vecindad sobre la cual se explora el espacio de búsqueda [GPJ10].

Un aspecto fundamental en la consideración de diversas funciones de vecindad es, que si bien el mínimo global es un mínimo en cada vecindad, distintas funciones no tienen por qué compartir los mismos mínimos locales [GPJ10]. Como consecuencia de esto, una vez que el algoritmo se ha quedado estancado en un valle, se establece como mecanismo de diversificación el paso a otra vecindad. La figura 2.9 muestra la fundamentación de este algoritmo, se supone que la línea azul representa la función de vecindad original mientras que la línea punteada representa otra posible función de vecindad. Nótese que en el punto S^* , es un óptimo local de la primera función mas no así de la segunda, por lo que el algoritmo puede encontrar una salida a través de esta.

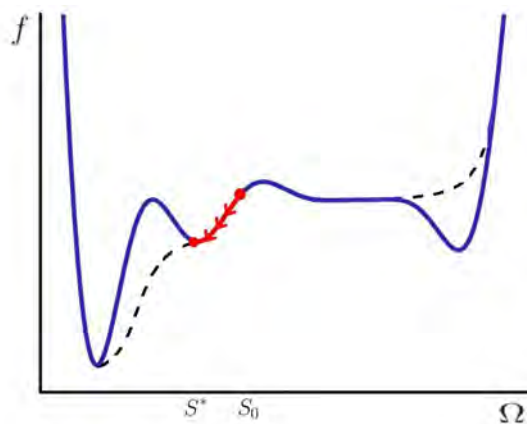


Figura 2.9: Representación gráfica del funcionamiento del algoritmo de Búsqueda en Vecindades Variables.

En la figura 2.10 se muestra el algoritmo detalladamente.

```

1  S := solucion_inicial()
2  S* := S
3  MAX := numero de vecindades
4  while( !criterio_de_parada() ){
5      k := 1
6      while ( k < MAX ){
7          S_vecina := explorar_vecindad(k,S)
8
9          if ( criterio_aceptacion(S, S_vecina) ){
10             S := S_vecina
11             k := 1
12          } else {
13             k := k+1
14          }
15
16          if ( f(S) < f(S*) ){
17             S* := S
18          }
19      }
20 }
21 return S

```

Figura 2.10: Pseudocódigo del algoritmo Búsqueda en Vecindades Variables

2.2.4. Búsqueda Local Iterada (Iterated Local Search - ILS)

La idea principal de la Búsqueda Local Iterada (ILS) consiste en construir iterativamente una secuencia de soluciones generadas por una heurística integrada. ILS posee dos puntos de interés:

- Debe existir una cadena de soluciones a seguir.
- El paso de mejora de cada solución se efectúa sobre el espacio reducido de salidas de una heurística integrada. Generalmente se utiliza búsqueda local como mecanismo interno, sin embargo esto no es estrictamente necesario.

ILS busca mejorar los resultados de la heurística interna a través del uso de la iteración. Para esto, sea una solución inicial S_0 , sea S^* el resultado de aplicar búsqueda local sobre la solución inicial. Se tiene que $f(S^*) \leq f(S_0)$. S^* es un óptimo local, por lo que sus vecinos

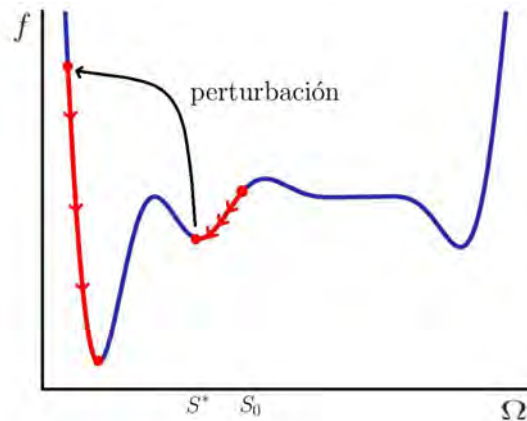


Figura 2.11: Representación gráfica del funcionamiento del algoritmo de Búsqueda Local Iterada.

tienen función de costo no menor. ILS busca mejorar dicha solución moviéndose de alguna forma a una solución cercana, sin la restricción de utilizar una vecindad. Este cambio se conoce como *perturbación*, la cual busca escapar de un óptimo local aplicando variaciones a una solución. Si la perturbación es muy pequeña, no será suficiente para escapar del valle generado por la función de vecindad, si la perturbación es muy grande, de tal forma que ignore la información de la solución anterior, el algoritmo estará manejando una estrategia de búsqueda local con reinicio.

En la figura 2.11 se muestra un paso de perturbación sobre el resultado de la figura 2.8. Generalmente se utilizan perturbaciones aleatorias para evitar ciclar en caso de que la metaheurística integrada devuelva la solución perturbada al mismo óptimo local. Adicionalmente, se pueden utilizar funciones de perturbación con memoria, para agregar la posibilidad de una mayor diversificación. El pseudocódigo del algoritmo ILS se expone en la figura 2.12.

```

1 S := solucion_inicial()
2 S* := busqueda_local(S)
3
4 do {
5     S' := perturbacion(S*, historia)
6     S*' := busqueda_local(S')
7
8     if ( criterio_aceptacion(S*, S*', historia) ){
9         S* := S*'
10    }
11 } while ( !criterio_de_parada() )
12
13 return S*

```

Figura 2.12: Pseudocódigo del algoritmo Búsqueda Local Iterada

2.3. Aprendizaje de Máquina

Se puede definir el área de Aprendizaje de Máquina (ML - Machine Learning) como una rama de la Inteligencia Artificial que estudia el diseño de algoritmos y máquinas capaces de aprender, esto es informalmente, que sean capaces de desempeñarse bien ante casos no vistos anteriormente.

Formalmente, se dice que un programa aprende de una experiencia E con respecto a cierta tarea T y una medida de desempeño P , si su funcionamiento en tareas de T , medidas por P , mejora a través de la experiencia E [Mit97].

En la presente sección, se presentan dos tipos de máquinas de aprendizaje y un algoritmo de detección de datos anómalos.

2.3.1. Máquinas de Soporte de Vectores

Las Máquinas de Soporte de Vectores (SVM) son un tipo de máquinas de aprendizaje supervisado, es decir, en el cual todos los datos de entrenamiento son previamente clasificados por un agente experto. El objetivo del aprendizaje supervisado es predecir la clase o valor de

futuras instancias de datos.

Dos conjunto de datos en dos dimensiones son linealmente separables si pueden ser divididos en su totalidad por una línea. En general, dos conjuntos de puntos n -dimensionales son linealmente separables si éstos pueden ser divididos por un hiperplano [Vap98].

En particular, las SVM se basan en la idea de que un conjunto de datos no separable linealmente, puede describir, en dimensiones mayores, un comportamiento separable. Para esto, se transforma el vector de entrada a otro espacio de dimensiones posiblemente mayores. Luego, sobre el vector resultante, se busca construir un hiperplano que separe las dos clases contenidas [CZ01, Vap98]. La función de transformación se conoce como *kernel* y puede ser una transformación lineal o no. En los casos de interés se utilizan transformaciones polinómicas que convierten el vector aplicando multiplicaciones por si mismo hasta un cierto grado k ; la transformación lineal, que puede ser vista como una transformación polinómica de grado uno y una función de base radial [Vap98].

Como han sido definidos, las SVM permiten establecer la separación de dos clases, sin embargo, esto puede ser generalizado a separar un conjunto de un número arbitrario de clases [CZ01].

2.3.2. Redes Neurales Artificiales

Las Redes Neurales Artificiales (ANN) son un tipo de máquina de aprendizaje bioinspirada en la estructura de la interconexión entre las neuronas naturales. Las redes neurales están interconectadas por un conjunto de unidades simples, donde cada unidad (neurona) recibe como entrada un conjunto de valores reales y produce un valor real. En particular, es posible que la salida de una neurona sea parte de la entrada de otra [Mit97].

Las redes neurales se adaptan bien a los problemas en los que el conjunto de entrenamiento puede tener ruido [Mit97], como es el caso de las imágenes provenientes de una cámara.

En el caso particular de este trabajo, se utilizarán en los Perceptrones Multicapa, los cuales son un tipo de red neural que utiliza aprendizaje supervisado.

2.3.2.1. Neurona Artificial

Una Neurona Artificial es una función matemática que representa el concepto de neurona en una ANN. Cada neurona recibe un conjunto de $n + 1$ entradas, una de las cuales siempre toma el valor de 1, denominado el valor de sesgo. Cada una de las entradas tiene asociado un peso w_i . El resultado de la neurona viene dado por la aplicación de una función sobre el producto punto del vector de entradas por el vector de peso. Dicha función es conocida como *función de activación*. Es decir, sea F la función de activación, el resultado de una neurona viene dado por la fórmula

$$F(\sum_{i=0}^n w_i \cdot x_i) \quad (2.3)$$

Entre las posibles funciones de activación están la función identidad, que da lugar a las llamadas unidades Lineales y la función sigmoide, definida como

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Estas funciones de activación son de especial interés dado que son derivables. Próximamente podrá ver la importancia de este hecho.

2.3.2.2. Redes Multicapa

Se define una red multicapa como una red, cuya topología viene definida por una secuencia de capas, cada una contiene un conjunto de neuronas. Las neuronas de cada nivel están completamente conectadas con las neuronas de la capa anterior, es decir, una neurona en la capa i , recibe como vector de entrada, la salida de cada una de las neuronas de la capa

$i - 1$ (adicionalmente a la entrada de sesgo). Por ejemplo, una posible topología de red neural multicapa se puede apreciar en la figura 2.13.

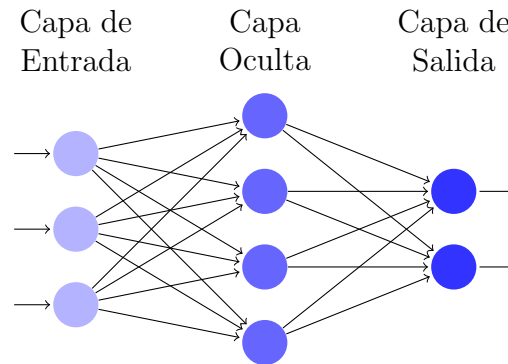


Figura 2.13: Estructura de una red neural multicapa con 3 capas: la capa de entrada con 3 neuronas, una capa oculta con 4 neuronas y una capa de salida con 2 neuronas.

2.3.2.3. Algoritmo “Backpropagation”

El algoritmo de backpropagation busca aprender los pesos internos de cada unidad en una red multicapa con una topología fija. Se trata de un algoritmo de aprendizaje supervisado que, de una asignación inicial de los pesos, frecuentemente aleatoria, y con la ayuda del algoritmo de descenso del gradiente, busca minimizar el error cuadrático entre la salida de la red y la función objetivo [Mit97]. Es por esta razón que es necesario que la función de activación de las unidades sea diferenciable, en el caso de la neurona sigmoideal, su derivada viene dada en función de $\sigma(x) \cdot (1 - \sigma(x))$, por lo que puede ser calculada rápidamente su derivada utilizando simplemente la aplicación de esta misma función. En la figura 2.14 se puede apreciar el pseudocódigo de este algoritmo. Cada ejemplo de entrada es de la forma $\langle x, t \rangle$, donde x es el vector de entradas y t el valor esperado de salida. n es la tasa de aprendizaje. La entrada proveniente de la unidad i a la unidad j es denotada como x_{ji} y el peso de la unidad i a la j es denotado como w_{ji} .

El algoritmo presentado viene dado únicamente para redes con una sola capa oculta, puede


```

1  for each <x,t> in conjunto_entrenamiento, do
2      Propagar la entrada por la red:
3          1. Ingresar x a la red y calcular su salida o_u,
4             para cada unidad u en la capa de salida.
5
6          Propagar los errores hacia atrás en la red:
7              2. Para cada unidad de salida k, calcular su error
8                  $S_k = o_k * (1 - o_k) * (t_k - o_k)$ 
9              3. Para cada unidad oculta h, calcular su error
10                  $S_h = o_h * (1 - o_h) * \sum(w_{kh} * S_k)$  forall k in outputs
11              4. Actualizar cada peso de la red w_ji
12                  $w_{ji} = w_{ji} + n * S_j * x_{ji}$ 
13
14 repeat until ( criterio_parada() )

```

Figura 2.14: Pseudocódigo del algoritmo Backpropagation [Mit97].

ser generalizado para redes con un mayor número de capas, sin embargo, se ha demostrado que para las funciones de interés es suficiente con una capa oculta [Cyb89, Mit97].

2.3.3. Detección de Datos Anómalos

La detección de datos anómalos consiste en, dado un conjunto de datos, detectar si existe algún subconjunto de ellos que no se adapte bien al modelo inducido por la muestra. Los datos anómalos pueden estar presentes en un conjunto de datos por errores en la captura de éstos o porque la naturaleza del problema determina que estos existan.

Dado que el número de datos anómalos es bajo por definición, los algoritmos clásicos de aprendizaje supervisado tienden a no ser capaces de generar un modelo capaz de detectar futuros datos anómalos y pasan a memorizar las instancias ya observadas [Dok].

Por otro lado, el problema de *clustering* es una técnica de aprendizaje no supervisado, en la que se busca descubrir una agrupación de datos mediante la detección de patrones presentes en ellos.

En este trabajo se utilizará la técnica de detección de datos anómalos presentada en [CA]

basada en clustering. La idea fundamental consiste en agrupar los datos en conjuntos según su proximidad, definida por una métrica de interés. Una vez agrupados, aquellos conjuntos muy pequeños vienen dados por datos que no comparten patrones de interés, bajo la métrica dada, con el resto de las instancias del conjunto, por lo que pueden ser considerados anómalos.

```

1 for d in conjunto_datos:
2     if S == 0:
3         S := S U {{d}}           // crear un cluster con d
4     else:
5         C := min_arg( {C0 in S, dist(C0,d)} ) // cluster más cercano a d
6
7         if dist(C,d) <= W:
8             C := C U {d}
9         else:
10            S := S U {{d}}

```

Figura 2.15: Pseudocódigo de la técnica de Detección de Datos Anómalos no Supervisada mediante clustering presentada en [Dok].

Sea M la métrica de interés, $dist(C, d)$ la distancia de un cluster a un dato bajo la métrica M y W el ancho de un cluster, es decir, la distancia máxima permitida entre el representante (centroide) de un conjunto y una instancia perteneciente a él. La figura 2.15 muestra el pseudocódigo de la fase de agrupación del algoritmo. Seguido a esta fase, cada conjunto debe ser detectado como anómalo o no, según su tamaño. El algoritmo supone que los datos se encuentran normalizados.

Capítulo 3

Segmentación de Dientes y Paladar

En este capítulo, se presenta la resolución del siguiente problema: Dada una imagen de vista palatina superior, indicar las regiones de ésta que pertenecen a los dientes y las regiones pertenecientes al paladar. Esto con el fin de dar una descripción general de la imagen, que permita a aplicaciones posteriores determinar aspectos de interés en cada una de estas áreas, como ausencia de dientes, enfermedades o anomalías en el paladar, etc. Se supone en este punto que la imagen de entrada cumple con las precondiciones establecidas, es decir, que se recibe una imagen a color de vista palatina, como las que se pueden apreciar en la figura 3.1



Figura 3.1: Fotografías de vista palatina superior.

El capítulo se divide en varias secciones, en las cuales se explorará la segmentación de dientes, mediante la clasificación de los píxeles entre factibles candidatos y los negativos. Posteriormente, se muestra el ajuste de una curva al arco definido por los dientes en la imagen y finalmente la segmentación del paladar.

3.1. Segmentación de Dientes

El objetivo de esta fase del sistema es generar una imagen binaria donde se separen los píxeles pertenecientes a dientes de aquellos pertenecientes a otras regiones de la imagen. La figura 3.2 muestra el conjunto de imágenes originales y el resultado esperado de esta fase para cada una de ellas.



Figura 3.2: Detección óptima de dientes. Arriba: imágenes originales. Abajo: clasificación dada por usuario experto.

En la resolución de este problema se plantearon diversas alternativas basadas en las técnicas comúnmente utilizadas. Inicialmente se probó con el uso de filtros que apuntan a resaltar las estructuras de interés para luego utilizar un umbral capaz de determinar la frontera entre los dientes y el resto, sin embargo, estas estrategias no dieron buenos resultados debido a la alta variabilidad de las imágenes manejadas. Estos experimentos se pueden encontrar en el apéndice A. Como solución final, se propuso el uso de algoritmos de aprendizaje de máquina, en conjunto con fases de pre y postprocesamiento, los cuales dieron resultados satisfactorios. A continuación se describen cada una de las fases involucradas en este proceso.

3.1.1. Preprocesamiento

El color es un factor determinante en la segmentación de dientes. Sin embargo, no todas las imágenes tienen buena calidad de color, por lo tanto, podría ser necesario un preprocesamiento que mejore la calidad de las imágenes.

Se propone un filtro basado en el método propuesto por Xiao&Ohya [XO07] expuesto en la sección 2.1.2.2.

La variación propuesta en este trabajo tiene el mismo enfoque en las componentes de color y luminosidad, sin embargo, la componente de saturación es desplazada de tal forma que la saturación media sea la misma en todas las imágenes. Esta variación busca mantener la diferencia de la saturación de cada píxel a la vez que genera valores de saturación buenos para que los colores sean distinguibles.

La figura 3.3 muestra el resultado de aplicar ambos filtros sobre seis imágenes. Más adelante se realizan experimentos que contrastan la mejora obtenida con el uso de estos filtros.



Figura 3.3: Preprocesamiento de las imágenes. Arriba: imágenes originales. Medio: Imágenes con filtro de Xiao&Ohya. Abajo: Imágenes con el filtro propuesto.

Además, tanto dientes como paladar contienen pequeñas estructuras con bordes irregulares que resulta de interés eliminar para simplificar el problema de detección de dientes. Por lo tanto, se aplica el filtro de suavizado de bordes expuesto anteriormente.

3.1.2. Clasificación de Píxeles

Se entrenaron Máquinas de Soporte de Vectores (SVM) y perceptrones para clasificar los píxeles de las imágenes entre regiones de dientes y no dientes. Los kernels utilizados en las SVM incluyen funciones lineales, polinómicas y de base radial (RBF). Para la decisión de los parámetros asociados al entrenamiento de cada tipo de SVM se utilizó una búsqueda logarítmica en el espacio de posibles valores. El algoritmo de entrenamiento para los perceptrones multicapa es el algoritmo aleatorio secuencial clásico backpropagation con una función sigmoide estándar como función de activación [LM98]. Las topologías utilizadas en las redes neurales tuvieron una capa oculta con entre 5 y 100 neuronas escondidas. Los resultados de los experimentos se muestran en la sección 3.1.3.

Como se puede apreciar en la imagen 3.4, en la hilera central de imágenes se aprecian bordes irregulares y puntos dispersos de ruido. Por lo tanto, es necesario antes de pasar a la siguiente fase un paso de postprocesamiento que permita depurar un poco la imagen. Con este fin se aplica un filtro de mediana y un filtro de componentes pequeñas de la imagen de salida de la red [SS01]. El resultado se puede apreciar en la tercera hilera de la imagen.

Es importante notar que después de esta fase siguen quedando zonas erróneas en la clasificación, por lo que las fases posteriores deben ser capaces de lidiar con esta situación y de ser posible, rectificarla. Si se observa por ejemplo, las fotografías 4 y 6 de la figura 3.4, se aprecia cómo algunas regiones de piel son clasificadas como regiones de dientes.



Figura 3.4: Detección de dientes. Arriba: Imágenes Originales. Medio: Salida de la red neural. Abajo: Filtro de componentes pequeñas.

3.1.3. Experimentos y Resultados de Segmentación de Dientes

Para los experimentos expuestos a continuación y, en general, para los futuros experimentos en donde no se especifique lo contrario, se utilizó un conjunto de 100 imágenes. El conjunto de entrenamiento y validación cruzada tiene un total de 20 imágenes, mientras que el conjunto de pruebas tiene un total de 80. Cada imagen tiene una resolución de 200×200 píxeles, dejando así un total de 800,000 píxeles para el conjunto de entrenamiento y 3,200,000 píxeles para el de pruebas.

Todos los experimentos fueron ejecutados en una máquina con las especificaciones definidas en la figura 3.1.

Procesador	Intel Core2Duo 2.4 GHz
Memoria	4 GB
Sistema Operativo	Ubuntu Linux 11.04

Cuadro 3.1: Especificaciones de la Máquina de Pruebas

Args	SVM					ANN				
	Kernel	Train	Cr-Val	Test	T _{seg}	Top.	Train	Cr-Val	Test	T _{seg}
RGB	Poly	81.37	81.43	81.23	1.06	20	81.16	83.28	78.14	0.08
eRGB'	Poly	81.62	81.61	81.57	1.10	25	85.10	85.24	82.21	0.10
eRGB	Poly	39.40	39.36	39.00	1.08	30	82.56	82.51	81.72	0.12
HSV	Poly	81.66	81.64	81.77	0.81	50	83.15	83.30	79.10	0.11
eHSV'	Poly	67.78	68.06	67.73	0.48	10	76.31	82.40	79.29	0.09
eHSV	Poly	75.73	75.83	77.12	0.52	5	82.78	82.76	81.87	0.08
GRAY	Poly	81.31	81.34	81.22	0.89	50	81.56	81.56	81.79	0.09
eGRAY	Poly	78.82	79.14	79.19	0.82	30	81.56	81.84	81.90	0.07

Cuadro 3.2: Clasificación de píxeles con SVM y ANN

El objetivo de los siguientes experimentos es determinar qué formato de imagen está mejor adaptado para la tarea de segmentación. Por consiguiente, se entrenaron distintas máquinas de tipo SVM y ANN usando cada uno de los siguientes formatos. RGB, HSV y GRAY se refieren respectivamente a los formatos RGB, HSV y escala de grises de la imagen de entrada. eRGB y eHSV son los formatos análogos pero aplicando el filtro de mejora de Xiao&Ohya's, mientras que eRGB' y eHSV' se refieren al filtro propuesto en el presente trabajo. eGRAY es la imagen en escala de grises aplicando equalización de histograma.

El cuadro 3.2 muestra que las redes Neurales se desenvuelven mejor en este problema que las Máquinas de Soporte de Vectores. Para cada formato de color, la mejor ANN encontrada supera la mejor SVM encontrada, exceptuando el caso de formato RGB sin aplicación de filtro, donde la SVM encontrada superó a la red neural en el conjunto de entrenamiento y de pruebas. Al igual que en el formato HSV puro, donde la SVM superó a la ANN en el conjunto de pruebas. Adicionalmente, el tiempo de clasificación de cada imagen es significativamente mejor utilizando redes neurales, alrededor de un orden de magnitud inferior.

Para explorar en mayor profundidad el impacto de los formatos de imagen, se realizaron experimentos combinándolos, los cuales se muestran en el cuadro 3.3.

En el caso de las SVM, no se apreció una mejora significativa en los resultados. Ninguna de las combinaciones logró superar el porcentaje de clasificación de este tipo de máquinas

con el formato HSV puro. Sin embargo, nótese que es determinante la inclusión de este formato al manejar estas máquinas, dado que en la mayoría de las combinaciones en las que se encontró presente el formato los resultados superaron a aquellas combinaciones que no contaban con éste. La mejor SVM encontrada clasifica un porcentaje correcto de 81.64 % sobre el conjunto de validación cruzada.

Argumentos	SVM					ANN				
	Kernel	Train	Cr-Val	Test	T _{seg}	Top.	Train	Cr-Val	Test	T _{seg}
eRGB' + eHSV'	Linear	68.63	68.68	67.07	0.68	100	85.42	85.64	83.94	0.15
eRGB + eHSV	RBF	78.40	78.64	75.72	0.80	10	81.98	85.19	83.38	0.10
eRGB' + eGRAY	Linear	37.93	37.87	39.16	0.90	5	82.13	81.95	81.66	0.08
eRGB + eGRAY	RBF	40.19	40.16	40.83	1.13	30	82.47	82.75	78.56	0.10
eHSV' + eGRAY	Poly	81.56	81.56	81.79	0.43	50	81.98	82.33	80.20	0.12
eHSV + eGRAY	Poly	81.56	81.56	81.79	0.32	20	82.66	83.10	82.36	0.09
eR' + eH' + eG	Linear	51.95	51.84	51.91	0.83	10	83.86	89.80	87.89	0.10
eR + eH + eG	Poly	79.62	79.79	80.44	0.96	50	89.32	89.93	87.71	0.13

Cuadro 3.3: Clasificación de píxeles sobre formatos combinados con SVM y ANN

La combinación de formatos de imagen mejoraron, en general, los porcentajes de clasificación de las Redes Neurales. El mejor porcentaje de clasificación fue de 89.93 % para el conjunto de validación. Este resultado fue obtenido por una red con 50 neuronas ocultas, que combina los formatos RGB y HSV con el filtro propuesto por Xiao&Ohya con la imagen mejorada en escala de grises. Sin embargo, al comparar esta combinación con el porcentaje de clasificación de la combinación análoga con el filtro propuesto en el presente trabajo, se determina que la diferencia entre ambos es pequeña, adicionalmente, la red entrenada con el filtro propuesto es de menor tamaño, incluyendo apenas 10 neuronas ocultas, esto se ve reflejado en el tiempo promedio de clasificación.

De los resultados se puede concluir que las ANN funcionan mejor que los SVM tanto en porcentaje de clasificación como en tiempo.

Para la mejor red encontrada, se puede apreciar en la figura 3.5 la variación del porcentaje de clasificación correcta para los conjuntos de validación cruzada (línea continua) y pruebas

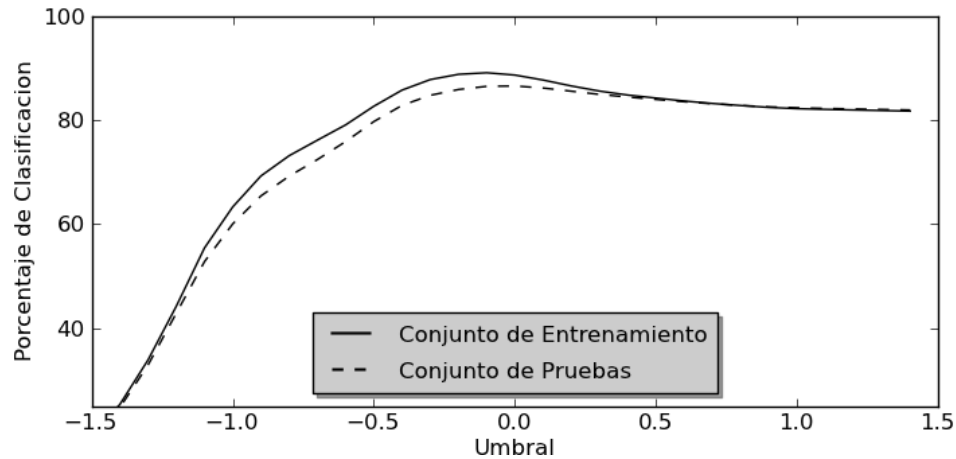


Figura 3.5: Comparación porcentaje de clasificación con diferentes umbrales de aceptación para la mejor ANN encontrada.

(línea punteada). Dicho gráfico sugiere una selección de umbral de aceptación de los resultados de la red de -0.1. Se puede observar que este resultado está generalizando patrones encontrados en los datos correctamente al comparar el valor de clasificación con el del conjunto de pruebas, el cual no es mucho menor.

Fase	Porcentaje(%)	Falsos Positivos (%)	Falsos Negativos (%)
ANN	87.8964	5.3436	6.7600
Filtrado por Componentes	88.3527	5.3857	6.2616

Cuadro 3.4: Porcentaje de Clasificación después de Remover Componentes Pequeñas

El cuadro 3.4 muestra el efecto de la eliminación de componentes pequeñas de la salida de la Red Neural utilizando el formato eRGB'+ eHSV'+ eGRAY. Este filtro logra obtener una mejora de 0.45 %, logrando decrementar el número de falsos negativos en 0.50 %. Si bien el resultado no parece ser significativo, véase en la figura 3.4 el efecto que produce este filtro sobre la imagen, si se comparan la hilera media e inferior de ésta, se puede apreciar una reducción importante de ruidos pequeños alrededor de la imagen.



Figura 3.6: Imágenes con los dientes segmentados y sendas curvas ajustadas por un humano.

3.2. Segmentación de Paladar

Una vez que las regiones correspondientes a los dientes son encontradas es más fácil determinar la región en la que se encuentra el paladar. La presente sección se divide en dos bloques, el primero explica el inicio del proceso de segmentación de paladar, que parte con base en la imagen con los píxeles clasificados y busca ajustar una curva al arco definido por los dientes. El segundo bloque se centra en la decisión final de la frontera diente/paladar. Por claridad, los experimentos relacionados a cada una de las fases se encuentran acompañando a su explicación.

3.2.1. Ajuste de Curva

El problema a resolver en primera instancia es: dada una imagen binaria donde los píxeles con valor 1 corresponden con píxeles de dientes y los píxeles con valor 0 al resto de las regiones, se desea ajustar una curva que pase por el centro del arco definido por los dientes. La figura 3.6 muestra un conjunto de imágenes clasificadas y una curva ajustada a las regiones correspondientes a los dientes.

Para esto, se ajustará una curva a las regiones en las que se encuentran los dientes mediante los algoritmos de optimización estudiados en la sección 2.2.

Estos algoritmos manejan problemas discretos, por lo que no se contemplarán curvas continuas sino curvas discretas definidas por una secuencia finita de puntos. Dichas curvas discretizadas definirán el *conjunto de soluciones* posibles a este problema. Es importante

notar la cardinalidad del conjunto de soluciones que definen el espacio de búsqueda, el cual, para una imagen de r filas y c columnas, con curvas discretizadas en k puntos, es de $(r \times c)^k$. Para el caso particular que se está manejando de imágenes de tamaño 200×200 y de curvas con el orden de 10 puntos, se tiene alrededor de 10^{46} posibles soluciones. Es necesaria entonces la intervención de algoritmos aproximados como los descritos en la sección 2.2.

Se experimentó con un total de 3 tipos de funciones de *calidad de una solución*. En primera instancia se probó con la distancia promedio de los píxeles clasificados como dientes a la curva (d), lo cual es un indicador de qué tan lejos está la curva del arco definido por los dientes. Sin embargo, esta aproximación llevó a resultados en los cuales la curva resultante perdía características de un paladar, por lo que se añadió más información a la métrica. Ésta lleva al segundo tipo de función de adaptación estudiada, se define como qué tan distorsionados se encuentran los puntos pertenecientes a la curva con respecto a la curva original (obviando traslaciones, escalas y rotaciones) y se denotará dtr . Sería posible establecer como función

$$d \cdot dtr$$

sin embargo, la distorsión, medida en el intervalo $[0 \dots 1]$, podría asignar como soluciones óptimas aquellas que sólo apliquen traslaciones, rotaciones y escalas sobre la solución inicial, por no poseer distorsiones. Por lo tanto, se desplazará el intervalo al $[0,5 \dots 1]$, quedando así la función definida como

$$d \cdot \frac{(1 + dtr)}{2}$$

Sea p , un punto perteneciente a la curva, que cumple con ser el más cercano a un píxel x clasificado como diente. Es posible asumir que p es un punto de interés en la solución dado que cubre un sector del arco. Sin embargo, si un punto p no ejerce influencia sobre ningún píxel, se considera inútil en la solución. A partir de esto, se plantean dos alternativas: eliminar

aquellos puntos que no ejerzan influencia o añadir a la función de calidad esta información, que se denotará (*useless*). La segunda opción es más interesante dado que no se resta libertad a la curva, permitiendo, en futuras iteraciones, una mejor adaptación a las nubes de píxeles. El tercer tipo de función de calidad agrega este parámetro. Al igual que en el caso anterior, es necesario desplazar el intervalo

$$d \cdot \frac{(1 + dtr)}{2} \cdot \frac{(1 + useless)}{2} = d \cdot \frac{(1 + dtr) \cdot (1 + useless)}{4}$$

Cada una de estas funciones serán denominadas en los experimentos como adaptación 0, 1 y 2 respectivamente. Sólo queda por definir la *función de vecindad* utilizada por cada algoritmo. Se utilizaron como funciones de vecindad la transformaciones por escalado, multiplicando ancho y largo de la curva por coeficientes de 1.0, 0.8 y 1.3, cuyas combinaciones dan un total de 8 vecinos; combinaciones de desplazamientos en ambos ejes y sentidos, para un total de 8 soluciones adyacentes; cuatro rotaciones en sentido horario y cuatro en sentido antihorario, añadiendo 8 soluciones más y finalmente el acercamiento de alguno de sus puntos al centro de masa de los píxeles activos más cercanos a éste, dando como resultado k vecinos más, para una curva discretizada en k puntos. En total, se tiene que todas las vecindades en conjunto tienen cardinalidad $24 + k$.

Cada metaheurística internamente utiliza la función descrita anteriormente como función de bondad, sin embargo, esta función no tiene una semántica clara y objetiva en el resultado final del ajuste, por lo tanto, el criterio de escogencia entre cada algoritmo es la distancia media entre la curva obtenida y una curva dada por un usuario experto. La distancia media entre dos curvas discretas se puede definir como el área que existe entre ellas cuando se abstraen como polígonos no necesariamente simples (que pueden intersectarse a si mismos) cerrados. La región sombreada en gris en la figura 3.7 muestra visualmente el error entre un par de curvas.



Figura 3.7: Curvas discretizadas (líneas azul y negra). Zona de error entre las curvas (gris).

3.2.2. Experimentos de Ajuste de Curva

En el cuadro 3.5 se muestran los resultados de aplicar cada una de las metaheurísticas expuestas sobre las imágenes clasificadas por un usuario experto. El error de la curva inicial en este caso es de 45.77 %.

	Adaptación 0		Adaptación 1		Adaptación 2	
Metaheurística	Error (%)	T_{seg}	Error (%)	T_{seg}	Error (%)	T_{seg}
LS	18.3319	0.17	18.1747	0.14	13.8655	0.27
VNS	19.2894	0.16	17.9496	0.15	13.8677	0.23
ILS	15.6552	0.85	15.7836	0.77	12.6823	0.75

Cuadro 3.5: Ajuste de Curvas sobre Imágenes Clasificadas de Forma óptima

Como se puede apreciar, a medida que la función de adaptación tiene más información, ésta obtiene soluciones con menor error porcentual medio. Sin embargo, el tiempo de cómputo es mayor, debido a que las funciones de adaptación más informadas permiten una mayor exploración del espacio de soluciones, adicionalmente, el tiempo necesario para calcularlas es mayor.

El mejor resultado se obtuvo utilizando la metaheurística ILS sobre la función de adaptación más informada (Adaptación 2). Sin embargo, esta opción tomó considerablemente más tiempo que los otros algoritmos bajo la misma métrica.

Buscando profundizar más en este análisis se realizaron experimentos utilizando las imágenes clasificadas por la red neural, en vez de utilizar las imágenes clasificadas por un humano. Los resultados de este experimento se muestran en el cuadro 3.6.

	Adaptación 0		Adaptación 1		Adaptación 2	
Metaheurística	Error (%)	T _{seg}	Error (%)	T _{seg}	Error (%)	T _{seg}
LS	33.3755	0.34	33.5197	0.17	32.3707	0.33
VNS	32.4951	0.37	32.8097	0.22	33.1590	0.24
ILS	32.5534	0.88	31.6030	0.62	30.0798	0.74

Cuadro 3.6: Ajuste de Curvas sobre Imágenes Clasificadas por la Red Neural

Como se puede apreciar, el mejor resultado es encontrado por la metaheurística de Búsqueda Local Iterada (ILS), utilizando como heurística interna LS, esta alcanzó reducir el error inicial en 15.7%. Sin embargo, también fue la variación que más tiempo necesitó para culminar. La comparación en tiempos de búsqueda local y búsqueda en vecindades variables no fue significativa.

Todas las estrategias utilizadas tuvieron comportamientos similares sobre los datos clasificados automáticamente. La diferencia principal entre ellas fue de tiempo de ejecución, en la cual la diferencia entre ILS y LS es de más del doble.

Con respecto a los resultados utilizando los datos clasificados de forma óptima se ve una reducción importante en la efectividad de las estrategias, de 17.39% para el caso de ILS con la función de adaptación más informada. En estos casos, en los cuales la imagen contiene zonas ruidosas, dado que la métrica incluye información de distancia a los puntos clasificados como dientes, los distintos algoritmos intentan alcanzar un punto estable entre los puntos de ruido, caracterizados por lo general por puntos ubicados en la piel. Se propone un mecanismo de limpieza iterativa del ruido situado por fuera de la curva, correspondientes por lo general a píxeles de piel. El proceso finaliza al alcanzar punto fijo de eliminación de píxeles. Dado que el número de píxeles en una imagen es finito, este algoritmo converge, el número máximo de iteraciones viene dado por el tamaño de la imagen, sin embargo, el cuadro 3.7 muestra que el método converge en pocas iteraciones para los tres algoritmos.

El cuadro 3.8 muestra el porcentaje de error de la curva generada por cada uno de los algoritmos, utilizando la función de adaptación más informada sobre los datos clasificados

Metaheurística	Iteraciones(promedio)	Iteraciones(máximo)
LS	2.49	7
VNS	3.5	9
ILS+LS	3.53	10

Cuadro 3.7: Limpieza Iterativa de Píxeles Ruidosos. Número de iteraciones promedio y máximo hasta alcanzar punto fijo.

Metaheurística	Error (%)	T_{seg}
LS	29.1108	0.77
VNS	29.1108	0.8
ILS+LS	28.3821	3.39

Cuadro 3.8: Limpieza Iterativa de Píxeles Ruidosos

por la red neural. Se puede apreciar una mejora de alrededor de un 3 % para los distintos algoritmos. Sólo ILS presenta tiempos considerablemente peores con respecto a la versión original, sin embargo, es la que muestra mejores resultados. Nuevamente, dependiendo si la prioridad es tiempo o fiabilidad, una u otra estrategia podría ser la más indicada.

Estos experimentos resultaron útiles para la estimación de la bondad del ajuste de las curvas con respecto a una curva dada por un usuario, sin embargo, esta medida es abstracta en el problema de segmentación de dientes y paladar. Pensando ahora en la variación sobre el porcentaje de clasificación de píxeles, el cuadro 3.9 muestra la diferencia entre los resultados previos a la realización del ajuste y los resultados posteriores. Como se puede apreciar, se



Figura 3.8: Ajuste de curvas y segmentación de posterior al filtro de píxeles fuera de la curva. Arriba: Curva Inicial. Abajo: Curva ajustada al arco del paladar.

observa una mejora de 1.36 %, con una reducción considerable de los falsos positivos (casi 2 %), generada por la limpieza iterativa de los píxeles fuera de la curva.

Fase	Porcentaje(%)	Falsos Positivos (%)	Falsos Negativos (%)
Clasificación Previa	88.3527	5.3857	6.2616
Ajuste de Curva	89.7204	3.6843	6.5953

Cuadro 3.9: Mejora general mediante el ajuste de curva

Esta fase no sólo permitió dar a conocer la ubicación del paladar en la imagen, sino que además logró mejorar las fases previas del sistema.

3.2.3. Decisión de Ancho de Banda

Como se puede observar en la figura 3.8, las curvas tienden a ajustarse en el centro de los dientes. Para decidir el límite alrededor de la curva el cual contiene píxeles de dientes se consideró inicialmente fijar un valor máximo de distancia. Aquellos puntos cuya distancia a la curva sea menor o igual al máximo permitido se clasifican como dientes y aquellos puntos más alejados se clasifican de forma negativa. Mediante el análisis de los casos de estudio se concluyó que diversos problemas tales como la escala de la imagen, diferencias anatómicas entre individuos y ruido en los datos clasificados por la red neural pueden derivar en tamaños diferentes de dientes. Es necesario por lo tanto detectar individualmente, para cada imagen de entrada, el ancho de banda alrededor de la curva en el que éstos están incluidos. Dadas estas características del problema, no es beneficioso realizar un aprendizaje previo, en cambio, el aprendizaje no supervisado da la ventaja de poder aplicarse sobre cada instancia por separado sin depender de la experiencia.

Se utilizó el algoritmo de aprendizaje no supervisado expuesto en la sección 2.3.3 y presentado por Chimphlee&Abdullah en [Dok]. Este algoritmo se propuso originalmente para detectar datos anómalos en una muestra, se realizó una adaptación en el problema a manejar con el fin de poderlo adaptar a este algoritmo.

Se define la muestra como la distancia a la curva de cada uno de los píxeles ubicados en la frontera de las regiones diente/no diente. Teniendo esto, se espera que los píxeles en la frontera se encuentren en un rango de distancias siguiendo un patrón, mientras que aquellos píxeles de ruido estén ubicados espacialmente de forma distinta.

Se utilizará como función de distancia de un punto a un cluster la distancia euclidiana entre el primero y el centro de masa del segundo.

Cuando todos los puntos son asignados, el algoritmo clasifica cada cluster como normal o anómalo basado en el tamaño de la agrupación. El ancho de banda seleccionado por el algoritmo corresponde con el centro de masa del cluster con datos normales más alejado de la curva, de esta forma se permite que toda la frontera catalogada como normal sea aceptada en el rango de los dientes.

El método descrito es esencialmente no supervisado, pero hay dos variables que deben ser establecidas, la holgura de aceptación permitida para un cluster (W) y el umbral de clasificación entre datos anómalos y datos normales. Se generó para esto un método supervisado de búsqueda exhaustiva que busca minimizar la distancia entre el ancho de banda real (establecido por un usuario experto) y el seleccionado por el algoritmo no supervisado.

3.2.4. Resultados Experimentales para la Selección del Ancho de Banda

La figura 3.9 muestra el error relativo medio de la aplicación del algoritmo de selección de ancho de banda variando el parámetro W . Para cada valor de W , el umbral de clasificación de clusters como normal o anómalo es seleccionado de tal forma que se minimice el error medio, los mejores resultados encontrados en el conjunto de entrenamiento fueron de 12,40 y 0,1 para W y el umbral de anomalía respectivamente. Con esta selección, el error medio relativo entre el ancho de banda seleccionado y el real es de 0.26 para el conjunto de entrenamiento

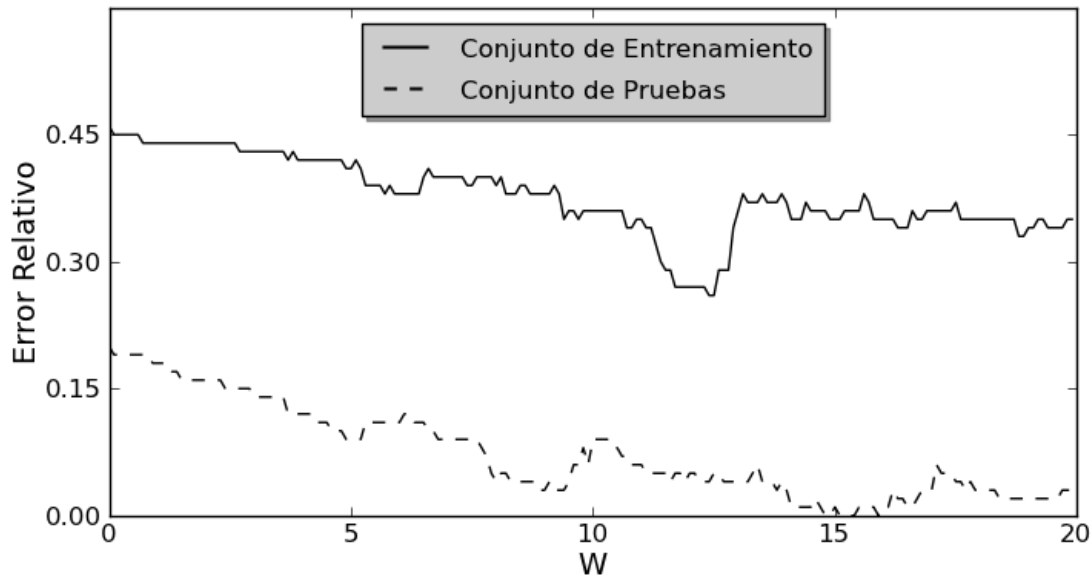


Figura 3.9: Selección de Ancho de Banda.

Fase	Porcentaje (%)	Falsos Positivos (%)	Falsos Negativos (%)
Ajuste de Curva	89.7204	3.6843	6.5953
Ancho de Banda	89.7594	2.2674	7.9732
Recuperación de Píxeles	89.8878	2.3557	7.7564

Cuadro 3.10: Variación en la clasificación de dientes por la selección del ancho de banda

y 0.04 para el de pruebas.

El cuadro 3.10 muestra la variación en la calidad de la solución en términos de la clasificación de los dientes. Recordando el cuadro 3.4 donde las redes neurales lograron un porcentaje de clasificación correcto de 87.89 %. Las fases de eliminación de componentes pequeñas, ajuste de curva y selección de ancho de banda mejoraron el porcentaje a 89.75 %. Identificada ya la región a la cual pertenecen los dientes, es posible reintegrar los píxeles que fueron descartados en etapas tales como el ajuste de curva y que, finalmente, terminaron perteneciendo a la región seleccionada como positiva. Este paso de recuperación de píxeles busca decrementar el número de falsos negativos, resultado que se puede apreciar en el descenso de 7.97 % a 7.75 %, dejando como resultado final un porcentaje de 89.88 % píxeles correctamente clasificados.

Es importante notar que la fase de detección del ancho de banda produce un incremento en el número de falsos negativos, dado que existen regiones positivas que podrían no ser cubiertas por la curva del paladar, como se puede apreciar en la banda superior de la figura 3.10. Sin embargo, es un paso crucial en la determinación de la ubicación del paladar.

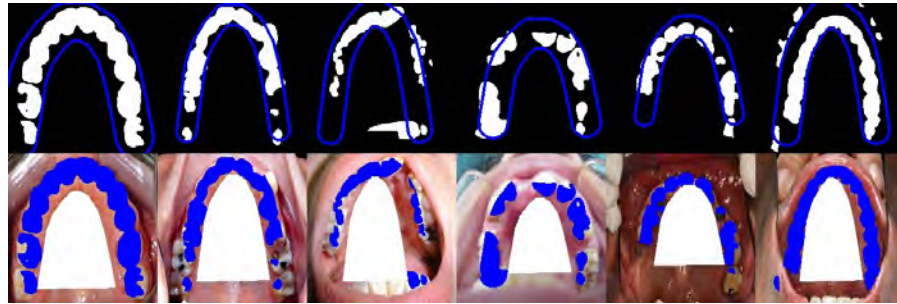


Figura 3.10: Selección de Ancho de Banda y Segmentación Diente/Paladar

Una vez que el ancho de banda es establecido, se tiene suficiente información para segmentar los dientes y el paladar del resto de la imagen. Los puntos clasificados como positivos entre la curva y la banda permitida se denotan como dientes. Los puntos dentro del polígono generado por la curva cerrada se denotan como parte del paladar, la salida de esta fase se muestra en la hilera inferior de la figura 3.10.

Finalmente, el porcentaje de clasificación de los píxeles como paladar es de 87.6501 %, teniendo el mayor porcentaje de datos erróneos en falsos positivos (9.0765 %).

Capítulo 4

Detección de Paladar

El presente trabajo tenía como objetivo la segmentación de dientes y paladar en fotografías de vista palatina, tal como se especificó anteriormente. En el capítulo 3 se mostró el mecanismo utilizado para este fin, siempre bajo la restricción de que las imágenes contenían un paladar.

Sin embargo, dado que el resultado de este trabajo pretende poder ser utilizado por usuarios no expertos, es posible que imágenes que no cumplan con esta precondition intenten ser evaluadas, por lo que se realizó un esfuerzo posterior en romper con esta barrera.

Para ello se entrenó una red neural encargada de clasificar cada imagen como válida o inválida. Análogamente a los experimentos anteriores el tipo de red de interés fueron los perceptrones multicapa con una capa oculta, adicional a las capas de entrada y salida.

El banco de imágenes utilizadas cuenta con un total de 2063 imágenes negativas (sin paladares) y un total de 400 imágenes positivas (imágenes de paladares). El conjunto de imágenes positivas se generó aplicando perturbaciones al conjunto inicial de 100 imágenes, con el fin de no generar un desbalance tan grande entre ambas clases (positivas y negativas). Para el otro conjunto, se utilizó el banco proporcionado por Naotoshi Seo en [Seo]. Ambos conjuntos cuentan con imágenes en escala de grises, por restricciones del conjunto de datos

negativos.

La ventaja de este conjunto es que posee una gran diversidad de imágenes, incluyendo fotografías de distintos temas. En la figura 4.1 se pueden apreciar algunas de las imágenes que se pueden conseguir en este conjunto.



Figura 4.1: Imágenes sin Paladar

Todas las imágenes se escalaron a un tamaño de 64×64 píxeles, de tal forma que todas las redes pudiesen tener el mismo número de entradas. Los argumentos pasados a la red se corresponden con la linealización de la matriz de la imagen. Se probó con dos tipos de preprocesamiento: imágenes originales e imágenes con equalización de histograma. El conjunto de entrenamiento cuenta con un 20 % del total, al igual que el conjunto de validación cruzada, el conjunto de pruebas cuenta con 60 % de los datos.

	Imagen Original				Equalización de Histograma			
Neuronas	Train	Cr-Val	Test	T_{mseg}	Train	Cr-Val	Test	T_{mseg}
5	99.28	94.76	94.91	66	99.88	96.31	96.42	83
10	98.81	93.33	93.56	99	99.76	95.83	96.18	115
15	99.40	95.60	95.87	127	100.00	97.50	98.05	143
20	99.76	96.31	96.10	134	100.00	96.90	96.62	147
25	100.00	95.00	94.71	165	100.00	96.90	97.66	188
30	99.76	95.00	95.63	192	99.88	96.79	96.62	209
35	99.40	92.74	93.48	227	100.00	94.29	94.16	240
40	99.64	94.64	94.91	231	100.00	96.43	96.54	246
45	98.33	93.57	93.96	274	100.00	97.02	97.46	294

Cuadro 4.1: Detección de Paladar

En el cuadro 4.1 se muestran los resultados de este experimento para clasificación binaria (positivo o negativo). Como se puede apreciar, los resultados obtenidos utilizando el preprocesamiento de equalización de histograma superan en todos los casos a sus análogos con la información original, bajo un costo agregado pequeño (en el orden de las 20 milésimas de segundo). Los mejores resultados para los conjuntos de validación cruzada y pruebas para las redes que entrenaron con los datos originales fue de 96.31 % de clasificación y 96.10 % respectivamente, mientras que los mejores resultados para las redes entrenadas bajo el preprocesamiento de las imágenes fue de 97.5 % y 98.09 % respectivamente.

El porcentaje de clasificación aleatoria, determinado por la relación entre el número de instancias positivas y negativas es de 90.62 %, con lo cual se demuestra la importancia del aprendizaje en este problema.



Figura 4.2: Falsos Negativos en Detección de Paladar



Figura 4.3: Falsos Positivos en Detección de Paladar

De todos los conjuntos, sólo cinco imágenes resultaron siendo falsos negativos, las cuales se pueden apreciar en la figura 4.2, mientras que un total de 14 imágenes fueron falsos positivos, se muestran las más representativas en la figura 4.3. De los falsos positivos resaltan

la primera y tercera figura, donde se evidencia un arco claro sobre un centro opaco, similar al arco inducido por los dientes y el paladar. Es importante recordar que las redes reciben estas imágenes con un menor tamaño a fin de evitar aprender patrones específicos.

Ahora que se tiene una máquina capaz de decidir con suficiente confianza cuando una imagen puede ser procesada por el sistema, cabe pensar en el caso de qué sucede con las imágenes que tienen un paladar pero en menor escala. Imágenes como estas se encuentran incluso en el conjunto de datos original, la franja superior de la figura 4.4 muestra ejemplos de estos casos.

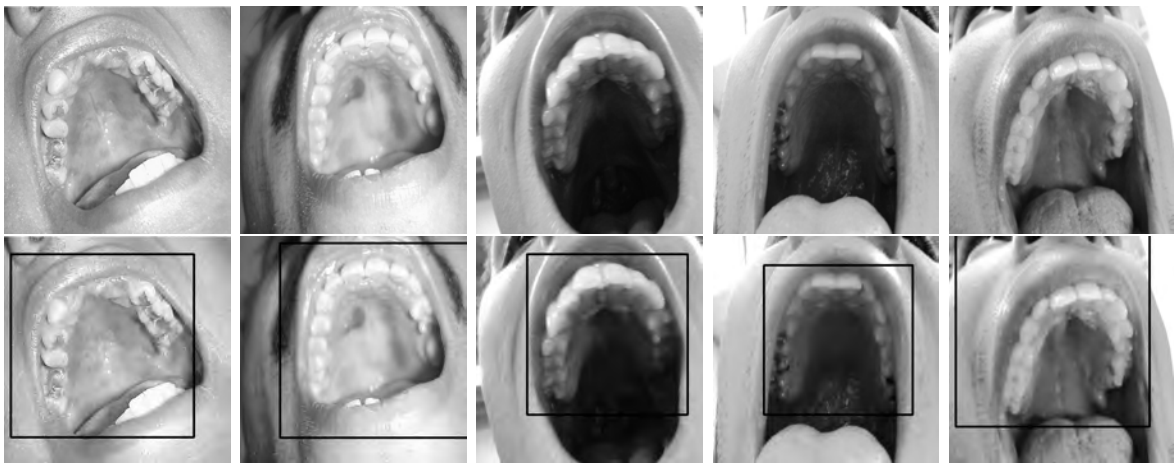


Figura 4.4: Imágenes con Paladar en Menor Escala

Para poder lidiar con casos en los que estas imágenes puedan ser clasificadas como negativas, se implementó un algoritmo basado en ventanas deslizantes. Este algoritmo busca la presencia de un paladar en la imagen, para esto, utiliza una región de interés de ésta, localizado en una ventana o recuadro de la imagen. Se realizan barridos de la imagen con ventanas de distintos tamaños hasta encontrar una ventana en la cual la red neural dé resultado positivo o hasta agotar el espacio de búsqueda. La franja inferior de la figura 4.4 muestra el resultado de aplicar el algoritmo de ventana deslizante sobre el análogo de cada imagen en la franja superior de la misma figura.

Si bien este paso se considera una etapa de preprocesamiento, dado que se desea dar una medida clara de la mejora dada por esta estrategia, el siguiente experimento mostrará cómo afecta la selección de este recuadro al porcentaje de clasificación de píxeles.

	Sin Detección			Con Detección		
Fase	%	F+	F-	%	F+	F-
ANN	87.89	5.34	6.76	87.89	5.34	6.76
Filtrado por Componentes	88.35	5.38	6.26	88.35	5.38	6.26
Detección de Paladar	—	—	—	89.60	3.82	6.56
Ajuste de Curva	89.72	3.68	6.59	90.32	2.88	6.79
Ancho de Banda	89.75	2.26	7.97	90.27	1.97	7.76
Recuperación de Píxeles	89.88	2.35	7.75	90.28	1.98	7.74
Píxeles del Paladar	87.65	9.07	3.27	88.66	8.18	3.15

Cuadro 4.2: Clasificación de Píxeles con Detección de Paladar. Se muestra el porcentaje de clasificación correcta (%), el número de falsos positivos (F+) y el número de falsos negativos (F-).

En el cuadro 4.2 se muestra la mejora sobre el porcentaje de clasificación de los píxeles en cada fase dado que se aplica este método como una etapa más en el proceso. En dicho cuadro se muestra para el proceso completo con y sin detección de paladar la variación en cada una de las etapas. Es importante notar como el porcentaje de píxeles bien clasificados pasó de 89.88 % a 90.28 %, mientras que la clasificación de la región del paladar subió más de un por ciento, pasando de 87.65 % originalmente a 88.66 %.

Capítulo 5

Segmentación Interdental

Como extensión a la información brindada por las fases anteriores se incluye esta fase. El objetivo final de este trabajo radica en ser base en la detección de enfermedades, en el caso de este trabajo del paladar. Sin embargo, como se evidenció en la introducción, existe una fuerte tendencia de trabajos que estudian fenómenos diente a diente [Nas06] [LH09], en los casos particulares presentados, la asociación de un par de radiografías a una persona. Es de interés por lo tanto, no sólo brindar la información de qué sectores de la imagen se corresponden a paladar y qué sectores a dientes sino adicionalmente brindar dónde empieza y dónde termina cada uno de los dientes. Esta información podría servir de base en futuros proyectos relacionados a los ya estudiados en el área, ahora con el uso de fotografías.

Con esto, no se está buscando etiquetar cada diente, sólo se pretende dar la información de ubicación de cada uno de ellos.

El resto de este capítulo se divide de la siguiente manera. Primero se muestra una fase de uso de la información suministrada por el sistema y de preprocesamiento para la presente fase, seguidamente se presenta el problema de detección de fronteras entre dientes y finalmente se exponen los resultados experimentales de esta fase.

5.1. Preprocesamiento

Una vez que los dientes y el paladar se encuentran segmentados del resto de la imagen, el mecanismo brinda lo siguiente: el conjunto de píxeles que pertenecen al paladar, el conjunto de píxeles que pertenecen a los dientes y una curva ajustada al arco inducido por los dientes. De esta información se utilizarán los dos últimos elementos.

Trabajar con esta información para tales fines no es sencillo, por lo que se quiere llevar a otra representación, sobre la cual sea más fácil realizar la detección.

El objetivo a apuntar es transformar este espacio variable, definido por la pendiente de la curva en una forma fija de imagen a imagen. La técnica utilizada es análoga a lo que se busca al obtener una representación plana del globo terrestre. Supóngase por un momento que la curva definida anteriormente, junto con la banda generada alrededor de ella se comportan como una cuerda que se busca tensar, al realizar esto, la figura obtenida se convierte en una recta. Este proceso se denominará aplanado de curva y se muestra en la figura 5.1. Es importante notar que se busca preservar la relación espacial de los puntos originales de la imagen (representados por los puntos rojos ubicados fuera de la curva y los puntos azules dentro de la curva) con respecto a la transformación de la curva.

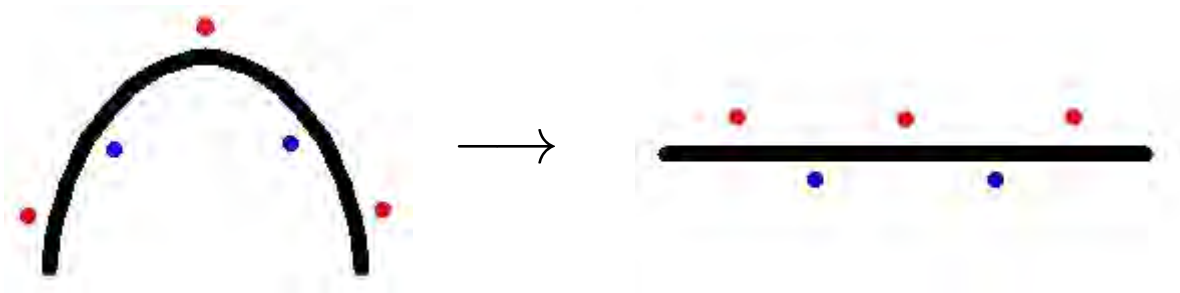


Figura 5.1: Aplanado de Curva

La banda de puntos a considerar alrededor de la curva se determina utilizando el radio generado por el mecanismo de aprendizaje no supervisado mostrado en la sección 2.3.3. Este

algoritmo presenta ciertos puntos problemáticos de la banda que generarán espacios vacíos en la imagen resultante. En la figura 5.2, todos los puntos definidos en la recta roja del lado izquierdo se encuentran equidistantes a ambos segmentos de curva (izquierdo y derecho), por lo tanto, al generar el aplanamiento de ésta, una región sin información se mostrará en el centro de estas dos regiones, tal como se puede apreciar en la figura central.

Este problema se resolvió inundando aquellas zonas de la imagen resultantes en las cuales no se asignaron píxeles de la imagen original con información generada de forma artificial. Esta información se genera mediante un algoritmo de inundación por capas, es decir, en forma de búsqueda en amplitud, asignando a cada píxel vacío el promedio de sus vecinos con información previa.

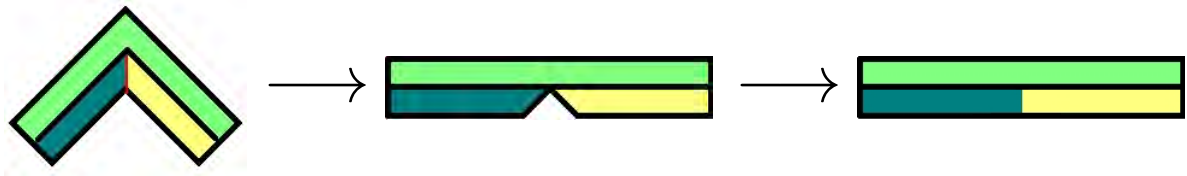


Figura 5.2: Problema con Aplanado de Curva. Izquierda: Curva Original. Medio: Curva Aplanada. Derecha: Curva Aplanada con Relleno

A pesar de que los segmentos izquierdo y derecho parecen ser contrastantes, resultando en una imagen con ruido, se debe recordar que ambas fronteras se encontraban unidas previamente, por lo que en general, tendrán información análoga en ambos extremos.

Una vez que se tiene aplanada la banda, se convierte su formato a escala de grises y finalmente se aplica equalización de histograma con el fin de resaltar el contraste entre dientes y encías.

5.2. Identificación de Fronteras

Se define una frontera como el punto en el que empieza o termina un diente en la secuencia de dientes planteada por la curva. Para la identificación de fronteras se plantea la utilización de una máquina de aprendizaje, en los experimentos de esta sección se utilizaron SVM y ANN, con el fin de clasificar, si en una imagen existe o no una frontera entre un par de dientes.

Seguidamente un algoritmo de ventana deslizante, análogo al planteado en el capítulo 4 busca las fronteras entre los dientes. La ventaja de tener la región de los dientes representada como un rectángulo es que la ventana puede tener un tamaño y forma fija. La complejidad del algoritmo es lineal en el número de ventanas deslizantes por la complejidad en tiempo del clasificador.

Se realizaron experimentos con Máquinas de Soporte de Vectores y Perceptrones Multicapa. Cada ventana utilizada tiene un tamaño de 50×25 píxeles.

5.3. Resultados Experimentales

Para los experimentos se utilizó un conjunto de entrenamiento de 70 imágenes clasificadas como frontera y 150 imágenes clasificadas de forma negativa. Para el conjunto de pruebas se utilizaron conjuntos de tamaño 143 y 313 respectivamente.

Como se muestra en el cuadro 5.1, las Redes Neurales superaron a las SVM en esta tarea. Los mejores resultados en tiempo y calidad fueron alcanzados por este tipo de máquinas. La mejor Red Neural encontrada tiene 15 neuronas en la capa oculta, a pesar de que el mejor resultado se obtuvo utilizando imágenes sin preprocesamiento, la red neural seleccionada trabaja con la imagen modificada. De esta forma, estará mejor adaptada a nuevos casos, porque la entrada es mejorada y estandarizada. Además, su configuración incluye menos

	SVM				ANN			
Argumentos	Kernel	Train	Test	T_{ms}	Neuronas	Train	Test	T_{ms}
Original	Linear	86.82	81.14	149	50	100.00	88.82	85
Mejorada	Poly	92.27	83.99	154	15	99.55	88.82	92

Cuadro 5.1: Segmentación Interdental

neuronas, por lo cual el modelo que adapta es más simple y tiene mayores probabilidades de haber aprendido patrones generales en la imagen en vez de memorizar los casos aprendidos.

La figura 5.3 muestra los resultados de aplicar esta método sobre las imágenes de entrada. La hilera central muestra el resultado de aplanar el ancho de banda de la curva, junto a su mejora y clasificación por parte de la red neural.

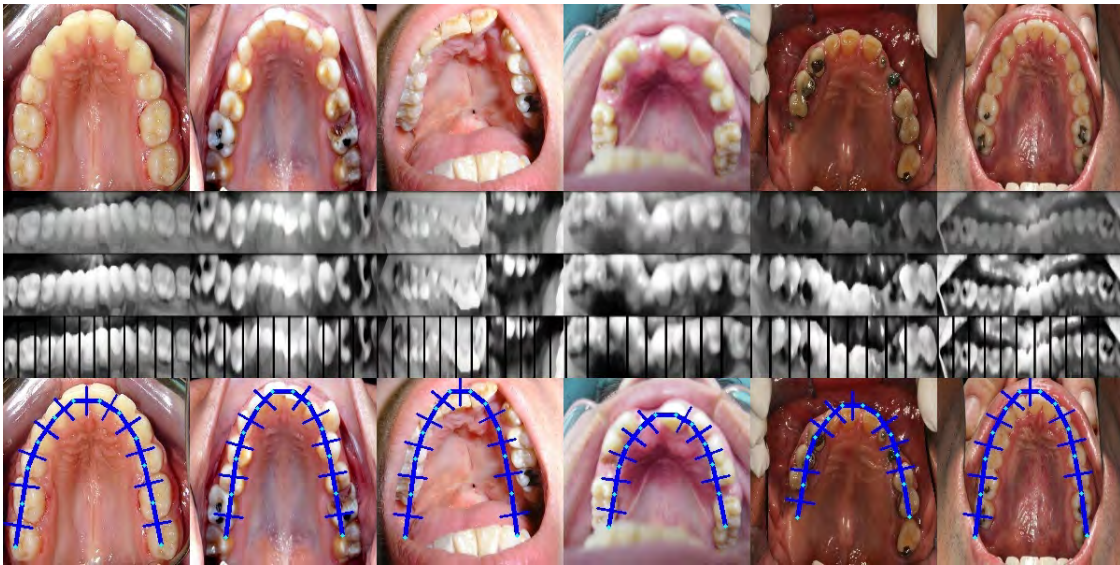


Figura 5.3: Dientes Segmentados. Arriba: Imágenes de Entrada. Medio: Curva Aplanada, mejorada y segmentada. Derecha: Transformación de la curva aplanada a la curva original.

Capítulo 6

Conclusiones y recomendaciones

Como resultado de este proyecto se logró implementar una herramienta capaz de segmentar en una imagen de vista palatina superior las regiones pertenecientes a los dientes y las regiones pertenecientes al paladar, con buenos niveles de confianza. La solución propuesta está fuertemente basada en algoritmos de aprendizaje de máquina y de optimización heurística. Estos algoritmos brindan la ventaja de poder detectar patrones automáticamente sin explícita manipulación humana, lo que permite generalizar características en los datos que un humano difícilmente puede captar.

El resultado final fue un sistema capaz de detectar las regiones de los dientes y paladar, con un porcentaje de 90.28 % y 88.66 % de los píxeles bien clasificados respectivamente, en un tiempo razonable de cómputo.

A diferencia de los trabajos previos, en los que se manejaron conjuntos de datos provenientes de radiografías, se logró implementar un mecanismo capaz de utilizar información proveniente de cámaras, con distintos niveles de calidad, lo cual permite garantizar que la solución responde efectivamente al problema planteado, a bajos costos.

Este trabajo funciona como plataforma para futuras etapas de detección de enfermedades,

en los cuales ya se ha empezado a trabajar. La utilización de mecanismos de segmentación automática, a la par de la utilización de datos provenientes de fotografías, permiten la reducción de costos de la solución al problema de diagnosticar este tipo de enfermedades tan comunes en países subdesarrollados. El uso de esta herramienta tiene un impacto positivo en la expansión de la red de atención médica en el país, permitiendo la introducción de éste a zonas de difícil acceso y reduciendo los gastos acarreados para los pacientes, generados por los mecanismos de control preventivo de estas enfermedades. En el caso particular venezolano, se cuenta únicamente con un centro de atención odontológica a pacientes con enfermedades infecto-contagiosas, ubicado en Caracas, en la Universidad Central de Venezuela.

Adicionalmente a las metas propuestas para el presente proyecto, se realizaron dos extensiones de notable importancia, incrementando así sus capacidades. La primera extensión logró romper la precondition de manejar únicamente imágenes de vista palatina superior, lo que permite llevar el sistema a un ambiente abierto, en el cual no se conozca la validez de los datos recibidos. Esta adaptación logró además mejorar la confianza del sistema en la segmentación entre dientes y paladar. En segundo lugar, se realizó un mecanismo de detección interdental, siendo este paso plataforma para la detección de elementos de interés diente a diente. Posibles extensiones en este sentido son la detección de la ausencia de algún diente, anomalías en estos, entre otras.

El presente trabajo será expuesto el mes de septiembre del presente año en el “*5th Workshop on Artificial Neural Networks in Pattern Recognition*” en la ciudad de Trento, Italia, para su posterior publicación en la serie “*Lecture Notes in Artificial Intelligence*” de la editorial Springer.

Para mejorar los resultados aquí planteados se proponen las siguientes recomendaciones:

- Realizar experimentos con otros tipos de máquinas y filtros, que permitan conducir

a la utilización de un mecanismo eficiente de integración de resultados entre diversas máquinas de aprendizaje.

- Considerar nuevas métricas en el uso de algoritmos de ajuste de curva que permitan una mejor exploración del espacio de búsqueda, así como la paralelización de estos algoritmos, permitiendo una mejora en los tiempos de respuesta. Otra alternativa sería mejorar la calidad de los datos previamente clasificados, de tal forma que la información tenga una menor cantidad de datos erróneos. En los experimentos realizados se mostró que estos algoritmos son susceptibles al ruido.
- Proponer un mecanismo de detección auxiliar en caso de que el paciente no posea ningún diente al cual ajustar la curva.
- Obtener un conjunto de datos diverso de imágenes a color que no contengan paladares. El conjunto utilizado tiene la ventaja de ser de libre uso a la par de contener imágenes de distintas fuentes. Si se consigue un conjunto de datos libres con calidad equivalente al utilizado sería posible extender el mecanismo de detección de paladares a imágenes a color.
- Incorporar un mayor conjunto de datos que permitan la utilización de máquinas de reconocimiento de patrones como los Clasificadores de Haar [WF06] que permiten ubicar un objeto en una imagen. Este podría ser un mecanismo de comparación de resultados con respecto al mecanismo planteado de detección de paladares, sin embargo, requieren un conjunto de datos positivos mayor al conjunto manejado para poder realizar un aprendizaje efectivo.

Entre los trabajos futuros a realizar se encuentra, adicionalmente a las recomendaciones planteadas, la detección de anomalías en el paladar, como muestra de la efectividad del

sistema en la detección de enfermedades en general. Para, posteriormente, plantear máquinas que aprendan a detectar distintos tipos de enfermedades.

Bibliografía

- [AL97] E. Aarts and J.K. Lenstra. *Local search in combinatorial optimization*. Princeton University Press, 1997.
- [AT98] Cloete C. Sauer G. Arendorf T.M., Bredekamp B. Oral manifestation of hiv infection in 600 south african patients. *j oral pathol med.* pages 176–179, 1998.
- [BW01] L Batchelor and F. Waltz. *Intelligent Machine Vision*. Springer, 2001.
- [CA] W. Chimphee and A. Abdullah. Unsupervised anomaly detection with unlabeled data using clustering.
- [C.E88] Barr C.E. Dental management of hiv-associated oral mucosal lesions: current and experimental techniques. *Robertson P.B., Greenspan J.S. Perspectives on Oral Manifestation of AIDS: diagnosis and management of HIV-associated infections. Littleton, Mass: PSG Publishing Co, Inc*, pages 77–95, 1988.
- [Cyb89] G Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 1989.
- [CZ01] Liu X. Xu X. Cai, Y. and G. Zhou. Support vector machines for predicting protein structural class. *BMC Bioinformatics*, 2001.
- [Dja] Django. <https://www.djangoproject.com/>.

- [Dok] Ertoz L. Kumar V. Lazarevic A. Srivastava J. Tan P. Dokas, P. Data mining for network intrusion detection.
- [GPJ10] M Gendreau, Potvin, and J. *Handbook of Metaheuristics*. Springer, 2010.
- [GW07] R Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, 3rd edition, 2007.
- [LC10] M. Li and Y. Cheung. Automatic segmentation of color lip images based on morphological filter. *ICANN 2010, Part I, LNCS 6352*, pages 384–387, 2010.
- [LH09] Lai Y. H. Lin, P. L. and P. W. Huang. An effective classification and numbering system for dental bitewing radiographs using teeth region and contour information. *Pattern Recognition*, 2009.
- [LM98] G.B. Orr LeCun, L. Bottou and K.-R. Muller. Efficient backprop. in: Neural networks-tricks of the trade. *Springer Lecture Notes in Computer Sciences 1524*, pages 5–50, 1998.
- [Mit97] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
- [Nas06] Ammar H. Nassar, D. A neural network system for matching dental radiographs. *Pattern Recognition 40, Elsevier Ltd.*, pages 65–79, 2006.
- [OCV] Opencv: Open computer vision. <http://opencv.willowgarage.com/>.
- [Seo] Naotoshi Seo. Tutorial: Opencv haartraining (rapid object detection with a cascade of boosted classifiers based on haar-like features).
- [SS01] L. Shapiro and G Stockman. *Computer Vision*. Prentice Hall, 2001.
- [Vap98] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., 1998.

- [WF06] P. Wilson and J. Fernandez. Facial feature detection using haar classifiers. *JCSC* 21, 2006.
- [WHO] *World Health Organization: Progress report 2011: Global HIV/AIDS response Epidemic update and health sector progress towards universal access*. WHO Press.
- [XO07] Ding Xiao and J Ohya. Contrast enhancement of color images based on wavelet transform and human visual system. *International Conference Graphics and Visualization in Engeneering*, 2007.

Apéndice A

Otros Filtros de Imágenes

Adicionalmente a los filtros mostrados en el capítulo 2, se implementaron otros filtros cuyos resultados fueron infructuosos en la segmentación de dientes. Estos filtros buscaban disminuir los efectos perjudiciales en la imagen provenientes de una iluminación no uniforme y la determinación de los dientes mediante la aplicación de un umbral predefinido. Si bien estas técnicas son comunes en el área de visión de computadoras, necesitan que se fijen un conjunto de parámetros que, en muchos casos, perjudican directamente la generalidad de la aplicación.

Adicionalmente, estos filtros trabajan sobre imágenes en escala de grises. Como se pudo observar en la sección 3, la información proveniente de la imagen en color es beneficiosa en la tarea de segmentación en este problema en particular.

Sin embargo, se muestran estos filtros y algunas imágenes asociadas a su uso con el fin de estudiar las consecuencias positivas y negativas en el uso de estos filtros.

A.1. Filtro Homomórfico

Los filtros homomórficos son utilizados para compensar el efecto de iluminación no uniforme en una imagen [LH09]. Una imagen puede ser considerada una función $I(x, y)$ determinada por la combinación multiplicativa de un componente de iluminación $i(x, y)$ y una de reflectancia $r(x, y)$ [LH09]. Es decir, I puede ser definida como:

$$I(x, y) = i(x, y) \cdot r(x, y)$$

El método definido a continuación es presentado en [LH09]. Para poder separar linealmente las componentes de iluminación y reflectancia, se aplica la función logarítmica sobre la imagen. Posteriormente, se aplica la transformada de Fourier al logaritmo de la imagen para obtener la suma de la componente de iluminación de baja frecuencia y de la componente de alta frecuencia de la reflectancia. Esto es:

$$\mathbf{I}(u, v) = \mathcal{F}(\ln(I(x, y))) = \mathcal{F}(\ln(i(x, y))) + \mathcal{F}(\ln(r(x, y)))$$

Posteriormente se aplica un filtro Gaussiano a \mathbf{I} que permita pasar valores bajos para remover las componentes detalladas, manteniendo su distribución de iluminación:

$$\mathbf{I}'(u, v) = \mathcal{G}(u, v) \cdot \mathbf{I}(u, v)$$

Finalmente, se devuelve la imagen al espacio original aplicando la inversa de la transformada de Fourier y el antilogaritmo, para luego dilatar la imagen.

La figura A.1 muestra la utilización de este filtro sobre una imagen. Es importante notar que si bien la iluminación es considerablemente más uniforme, los dientes terminan siendo menos diferenciables del contexto, especialmente en el caso de la imagen derecha.

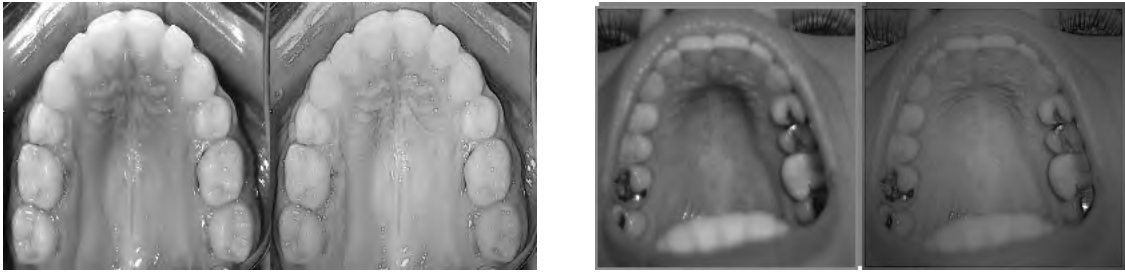


Figura A.1: Aplicación de filtro homomórfico sobre dos imágenes.

A.2. Filtro de Umbral

Otro filtro comúnmente utilizado es el filtro por umbral. Este filtro tiene el efecto de asignar un valor neutro a todos aquellos píxeles cuyo valor que exceda cierto umbral. El mismo enfoque se puede utilizar para el caso inverso, es decir, asignando un valor neutro a los píxeles que no alcancen cierto umbral.

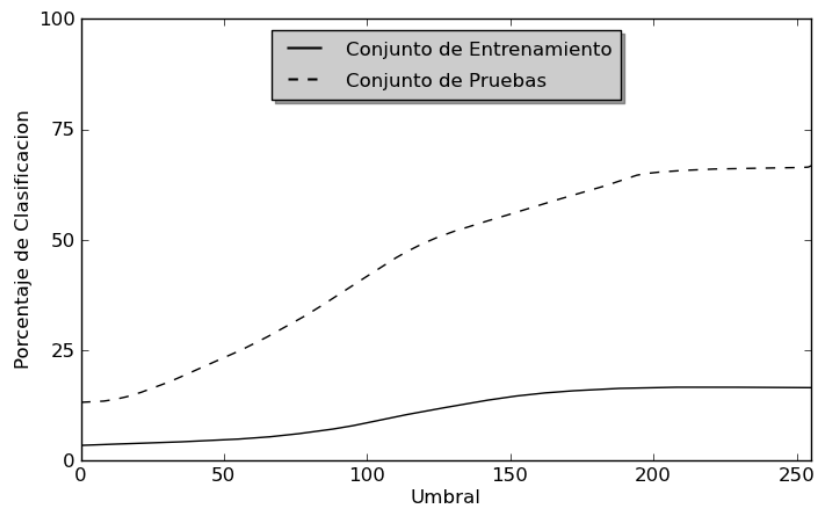


Figura A.2: Porcentaje de Clasificación utilizando distintos umbrales

Este filtro suele utilizarse debido a que es fácil de implantar. Además, en problemas en los que se conoce especialmente el tipo de imágenes a manejar, se puede realizar una calibración exacta de este umbral con el fin de obtener precisamente la información deseada. Por ejemplo,

seguir una pelota roja sobre un tablero azul.

En el caso de este proyecto se utilizó bajo la hipótesis de que los dientes pueden ser separados linealmente del resto de la imagen utilizando un umbral.

Para buscar probar esto se realizó un experimento variando el valor de umbral cuyo resultado se puede apreciar en la figura A.2, la cual muestra el mejor porcentaje de clasificación obtenido en la tarea de segmentación de dientes. Como se puede apreciar, el porcentaje de clasificación no supera el 16.549 % en el conjunto de entrenamiento y el 66.77 % en el de pruebas. Esto da muestras de que no es posible obtener un buen umbral que logre generalizar la frontera entre los píxeles pertenecientes a los dientes del resto.

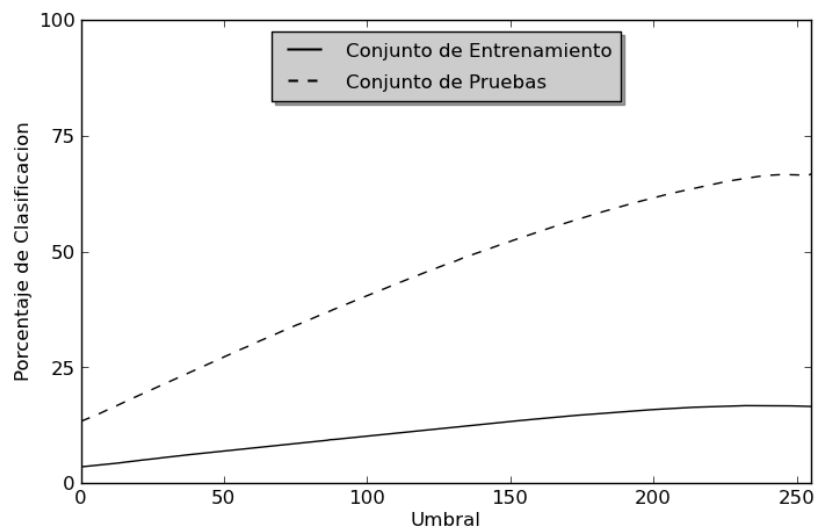


Figura A.3: Porcentaje de Clasificación utilizando distintos umbrales

Por lo tanto, se intentó preprocesar la imagen con la ayuda de diversos filtros, entre ellos los filtros de equalización de histograma y normalización de distribución con el fin de estandarizar las imágenes de entrada. Los resultados obtenidos se pueden apreciar en la figura A.3 y son similares a los mostrados en el experimento anterior, este porcentaje es inferior al obtenido por las redes neurales.

El mejor umbral encontrado en ambos casos es 255, el cual no permite el paso de ningún píxel, esto se corresponde con asumir que la imagen no tiene píxeles de dientes, la cual es la clase con más instancias en el conjunto de datos (66.77 % en el conjunto de pruebas).

Apéndice B

Descripción General de la Implementación

La implementación de este proyecto fue desarrollada completamente en *C++*, con el fin de minimizar los costos de evaluación de cada imagen. Los procesos aquí expuestos requieren fundamentalmente de procesador, por lo que es necesario contar con una herramienta que aproveche al máximo este recurso.

Para el manejo de imágenes, se utilizó la librería OpenCV (Open Source Computer Vision) [OCV], la cual es una librería de visión de computadoras optimizada para funcionar en tiempo real.

OpenCV opera bajo la licencia BSD, por lo que permite un uso libre académico y co-



Figura B.1: Logotipo de OpenCV

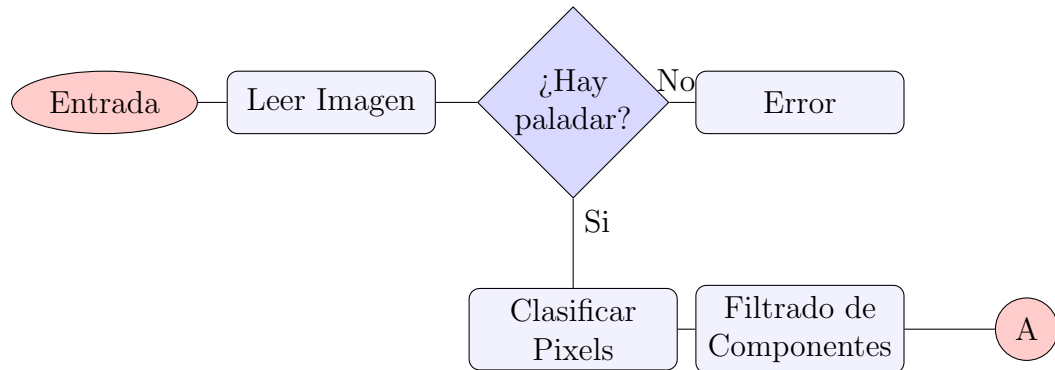


Figura B.2: Diagrama de Flujo de la Tarea de Clasificación de Píxeles como Diente/No Diente

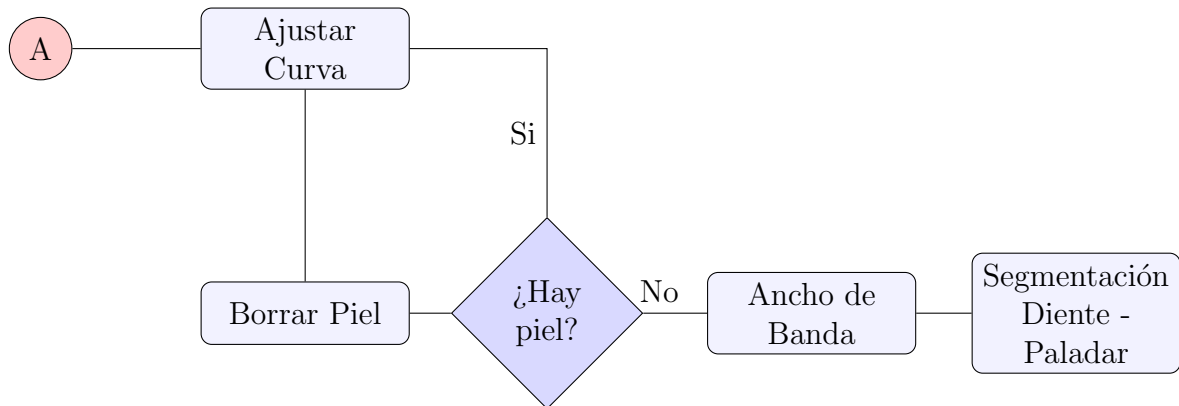


Figura B.3: Diagrama de Flujo de la Tarea de Segmentación del Paladar

mercial. Esta librería cuenta con implementaciones en C, C++ y Python en múltiples plataformas. Además es ampliamente utilizada, por lo que existe una gran documentación de su uso.

Tal como se mostró en el capítulo 3, el proceso general de la herramienta se puede dividir en dos etapas:

- Clasificación de Píxeles como Dientes.
- Segmentación del Paladar.

En la figuras B.2 y B.3 se puede apreciar el esquema general de estas dos etapas.

Apéndice C

Sistema de Clasificación de Imágenes

Como medio de interacción del sistema con los usuarios finales (odontólogos expertos y pacientes a diagnosticar) se implementó un sistema web, el cual fue desarrollado utilizando el lenguaje de programación *Python 2.7* y el *framework* de desarrollo web *Django* [Dja] en su versión 1.4. Como manejador de base de datos se utilizó *PostgreSQL*, versión 8.4.9.



Figura C.1: Herramientas Utilizadas. Python (izquierda), Django (centro), PostgreSQL (derecha).

El sistema, en principio, posee las funcionalidades necesarias para añadir nuevas imágenes al banco de entrenamiento y la clasificación de éstas como casos de pacientes sanos, con enfermedades o sin diagnóstico preestablecido por un usuario experto. Cada una de estas funcionalidades se pueden apreciar en la figura C.2. El diagnóstico de cada caso es individual por usuario, de tal forma que sea posible la determinación de diversos diagnósticos por parte de un grupo de odontólogos.



Detección de Enfermedades en Imágenes de Vista Palatina Superior


Usuario: [Salir](#)

Diagnostica un caso

(Quedan 100 imágenes por evaluar)




Diagnostico:


Descripción:

* En caso de ser un diagnóstico positivo, especifique la(s) enfermedad(es) que posee el paciente.

[Volver](#)

Universidad Simón Bolívar.
 Grupo de Inteligencia Artificial.
 Kelwin Fernández



Detección de Enfermedades en Imágenes de Vista Palatina Superior


Usuario: [Salir](#)

Subir una nueva imagen

Image:

Especificaciones:

Sarcoma de Kaposi

[Volver](#)

Universidad Simón Bolívar.
 Grupo de Inteligencia Artificial.
 Kelwin Fernández

Figura C.2: Diagnóstico de Imágenes (Arriba).
Carga de Nuevas Imágenes (Abajo).

Este sistema será utilizado como plataforma para el diagnóstico automático de enfermedades una vez que dicho objetivo sea cumplido.

El uso de tecnologías web en la presentación de esta herramienta permite maximizar su alcance, permitiendo que un mayor número de personas puedan disfrutar de sus beneficios. Esto acarrea como resultado la disminución del costo de determinación de un diagnóstico inicial a un paciente, por lo que se plantea como una alternativa viable a los métodos tradicionales, los cuales, tal como se mencionó anteriormente, son en muchos países escasos.