



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN

**Conteo de Colonias de Bacterias en
Cápsulas de Petri**

ALEJANDRO JOSÉ NARANJO ESPINOZA

Profesor Guía: MATTHEW BARDEEN

Memoria para optar al título de
Ingeniero Civil en Computación

Curicó – Chile
Noviembre, 2017

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Two circular stamps and signatures. The left stamp is from the 'DIRECCIÓN SISTEMA DE BIBLIOTECAS UNIVERSIDAD DE TALCA' with a signature over it. The right stamp is from the 'SISTEMA DE BIBLIOTECAS CAMPUS CURICO' with a signature over it.

Curicó, 2019

“Las ciencias aplicadas no existen; sólo las aplicaciones de la ciencia.” (Louis Pasteur).

RESUMEN

Las bacterias son esenciales para muchos procesos biológicos que ocurren día a día, tales como fermentación de algunos alimentos o ayudar en el crecimiento de plantas. Sin embargo, también son capaces de causar enfermedades mortales. Por estos motivos, las bacterias son sujeto de estudio en distintos campos de la ciencia. El cultivo de bacterias forma colonias bacterianas, que son las que se utilizamos para realizar los estudios, cuyo objetivo principal es analizar el comportamiento de las distintas cepas, independiente de su origen, para lograr repetir/mejorar los beneficios que otorgan, o prevenir o buscar curar las infecciones que pueden causarnos. Gran parte de los estudios pasan por un proceso de conteo de las colonias de bacterias que han proliferado. El problema que se aborda en este documento es realizar un buen conteo de la cantidad de colonias bacterianas presentes en un cultivo, ya que no todos los métodos de conteo son precisos y otras soluciones son muy costosas. Por ello, nos propusimos desarrollar una aplicación web que permita, a través de una fotografía de un cultivo bacteriano, identificar el mayor número de colonias presentes en un tiempo menor a 30 segundos.

Para este sistema se utiliza Procesamiento Digital de Imágenes, un área de la computación en la cual se busca realzar y extraer características de interés de las imágenes. Para alcanzar el objetivo propuesto fue necesaria una investigación que aborda dos campos diferentes, Procesamiento Digital de Imágenes y la Microbiología enfocada a bacterias. Para realizar la validación del sistema realizado, durante el desarrollo se obtuvieron alrededor de 50 muestras de colonias diferentes. Las pruebas consistieron en comparar los resultados obtenidos utilizando el sistema contra resultados manuales con algunas de las muestras más representativas. Estas pruebas nos permitieron identificar y validar las condiciones de precisión y tiempo preestablecidos.

Palabras claves: Procesamiento digital de imágenes, Transformada de Hough, Hough Transform, bacterias, colonias de bacterias.

SUMMARY

Bacteria are essential for many biological processes that happen day to day, such as fermenting some food or helping in the growth of plants process. However, they are also capable of causing deadly diseases. For these reasons, bacteria are subject of studies in different fields of science. The bacterial culture forms bacterial colonies, which are the ones used to carry out the studies, whose main objective is to analyze different strains behavior, regardless of their origin, in order to repeat/improve the benefits they provide, or prevent or search cure the infections that they can cause. Many of the studies go through a process of counting bacterial colonies that have proliferated. The problem approach in this document is to make a good count of the number of bacterial colonies on the petri dishes, because not all of counting methods are accurate and other solutions are very expensive. Therefore, we aim to develop a web application that allows, through a photograph of a bacterial culture, to identify the largest number of colonies present on a petri dish in less than 30 seconds.

This system uses Digital Image Processing (DIP), an computing science area that seeks to enhance and extract interesting features of images. To achieve the proposed objective, a research that addresses to two different fields was needed, Digital Image Processing and Microbiology focused on bacteria. To carry out with the validation of the system, during the development around 50 samples of different colonies were obtained. The tests consisted of comparing the results obtained using the system against manual results with some of the most representative samples. These tests allowed us to identify and validate the pre-established conditions of precision and execution time.

Keywords: Digital Image Processing, Hough Transform, bacteria, bacterial colonies, petri dishes.

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	IV
Índice de Figuras	VII
Índice de Tablas	X
Resumen	XI
1. Introducción	12
1.1. Descripción del Contexto	13
1.2. Objetivos	13
1.2.1. Objetivos Generales	13
1.2.2. Objetivos Específicos	14
1.3. Alcances	14
2. Marco Teórico	15
2.1. Procesamiento Digital de Imágenes	16
2.1.1. Adquisición de imagen	18
2.1.2. Mejora de imagen	18
2.1.3. Restauración de imagen	18
2.1.4. Procesamiento del color	18
2.1.5. Procesamiento de multiresolución	18
2.1.6. Compresión	18
2.1.7. Procesamiento morfológico	18
2.1.8. Segmentación	19
2.1.9. Representación y Descripción	19
2.1.10. Reconocimiento	19
2.2. Imagen Digital	19

2.2.1.	Profundidad de Color	21
2.2.2.	Escala de Grises	22
2.2.3.	Modelo RGB	22
2.3.	Histograma	24
2.4.	Selección por Umbral	24
2.4.1.	Umbralización de Imagen	25
2.4.2.	Umbralización mediante Método IsoData	26
2.5.	Reducción de ruido	26
2.5.1.	Filtro de Mediana	26
2.6.	Detección de Bordes	28
2.6.1.	Operador de Sobel	29
2.6.2.	Operador de Prewitt	30
2.6.3.	Operador de Roberts	31
2.6.4.	Operador de Laplace	31
2.7.	Transformada de Hough	32
2.8.	Bacterias	33
2.8.1.	Morfología de Bacterias	34
2.8.2.	Agrupación de Bacterias	35
2.8.3.	Colonias	36
2.8.4.	Importancia del Conteo de Colonias	36
2.8.5.	Métodos de Conteo más Utilizados	37
2.9.	Trabajo Existente	38
2.10.	Resumen Marco Teórico	40
3.	Diseño del Sistema	41
3.1.	Diseño del sistema	42
3.2.	Componentes	43
3.2.1.	OpenCV	43
3.3.	Metodología	44
4.	Desarrollo del Sistema	46
4.1.	Adquisición	47
4.2.	Procesamiento	48
4.2.1.	Umbralización	48

4.2.2. Reducción de Ruido	50
4.2.3. Detección y Realce de Bordes	53
4.3. Reconocimiento	54
4.4. Búsqueda de Parámetros	55
4.5. Resumen	56
5. Pruebas, Resultados y Discusión	57
5.1. Datos	58
5.1.1. Muestras Originales	59
5.2. Pruebas	64
5.2.1. Colonia Simple	65
5.2.2. Datos de Prueba	66
5.2.3. Muestra 1	67
5.2.4. Muestra 2	68
5.2.5. Muestra 3	69
5.2.6. Muestra 4	70
5.2.7. Muestra 5	71
5.2.8. No Procesable 1	72
5.2.9. No Procesable 2	73
5.2.10. No Procesable 3	74
5.2.11. No Procesable 4	75
5.3. Resultados	76
5.4. Discusión	77
6. Conclusión	78
6.1. Conclusión General	79
6.2. Resultados Obtenidos	80
6.3. Puntos Claves	80
6.4. Trabajo Futuro	81
Bibliografía	83

4.2. La humedad genera una distorsión de la imagen dificultando el reconocimiento de colonias. Por otra parte, el fondo de la imagen debe ser preferentemente de un alto contraste en relación a los colonias que se presentan.	48
4.3. Se presenta una muestra de una colonia de bacterias, su respectiva transformación a escala de grises y binarizado, sin optimizar.	49
4.4. Se presenta una binarización optimizada para la imagen anterior, mediante una técnica de mediana (izquierda) y mediante una técnica Gaussiana (derecha).	50
4.5. Armonización de la curva del histograma de una imagen.	50
4.6. Podemos apreciar las correcciones que realiza el filtro de mediana en la imagen binarizada.	51
4.7. Reducción de ruido de imagen binarizada con técnica de mediana.	51
4.8. Reducción de ruido de imagen binarizada con técnica Gaussiana. .	52
4.9. Presenta imágenes de la colonia una vez utilizado un filtro de promedio con kernel 3 y 5 (matriz de 3x3 y 5x5).	52
4.10. Detección de bordes mediante operador de Laplace y operador de Sobel.	53
4.11. Ejemplo ejecución algoritmo transformada de Hough.	54
5.1. Colonia simple que contiene 19 colonias.	59
5.2. Imagen de prueba con la cual se realizó la búsqueda de parámetros iniciales, contiene 21 colonias.	59
5.3. Muestra de colonia real, contiene 18 colonias.	60
5.4. Muestra de colonia real, contiene 26 colonias.	60
5.5. Muestra de colonia real, contiene 38 colonias.	61
5.6. Muestra de colonia real tomada desde un microscopio electrónico, contiene 82 colonias.	61
5.7. Muestra de colonia real, contiene 261 colonias.	62
5.8. Muestra de colonia real no procesable debido a la presencia de sombras y el flash de una cámara, contiene 17 colonias.	62
5.9. Imagen no procesable, no es real y presenta elementos distractores como figuras de cámaras fotográficas, líneas rectas y letras, contiene 181 colonias.	63

5.10. Muestra de colonia real no procesable debido a la presencia de sombras, contiene 293 colonias.	63
5.11. Muestra de colonias real no procesable debido a la presencia de excesiva humedad en la tapa de la cápsula de Petri, no es posible determinar el número de colonias debido al estado poco prolijo de la muestra.	64

ÍNDICE DE TABLAS

	página
2.1. Resumen niveles de procesos.	17
2.2. Gradiente de píxeles en una vecindad de 3x3.	29
2.3. En resultado de [16] vemos como la desviación estándar presentada es de un 10%, en general. Sin embargo, algunos resultados se escapan de esta desviación para presentar.	39
2.4. El resultado de [20] vemos como el error presentado al realizar el conteo de las colonias es, en promedio, de un 4%.	40
5.1. Tabla de muestras utilizadas para pruebas	58
5.2. Tabla de muestras utilizadas para pruebas	76

1. Introducción

1.1. Descripción del Contexto

En biología, más específicamente en microbiología y bacteriología, se utilizan métodos como el “cultivo” para la multiplicación de microorganismos como las bacterias, pues estos proveen un medio óptimo para favorecer su crecimiento. A medida que crecen y se reproducen, estas suelen agruparse en distintos puntos. A la aglomeración de bacterias le llamamos colonia bacteriana, o simplemente colonia. El cultivo se utiliza como método fundamental en el estudio de bacterias y otros microorganismos que causan enfermedades, puesto que permite observar su crecimiento en condiciones óptimas.

Para medir el crecimiento y establecer las características de una muestra, las colonias son contadas (pasan por un recuento cada cierto tiempo). Existen una gran variedad de métodos para llevar a cabo el conteo de las colonias [8]; sin embargo, la mayoría requiere de un tiempo y esfuerzo, que en muchas ocasiones, debido a la cantidad de colonias que pueden aparecer, es mayor al esperado. También se debe tener en cuenta que cada método llega a un valor estimado diferente, ya que no todos son exactos, el cual se basa principalmente en su complejidad intrínseca.

Buscando apoyar y mejorar la eficiencia del proceso que implica el cultivo y posterior conteo de colonias de bacterias, se ha propuesto una aplicación web que permita reducir el tiempo que toma realizar el conteo manualmente, de varios minutos a tan solo segundos.

1.2. Objetivos

1.2.1. Objetivos Generales

- Desarrollar una aplicación web que a partir de una fotografía o imagen realice el recuento del número de colonias que hay presentes en ella y muestre la información al usuario, dentro de un tiempo menor a 30 segundos y una precisión no menor al 85 % del total.

1.2.2. Objetivos Específicos

- Desarrollar un algoritmo capaz de procesar una imagen para reconocer los patrones de las colonias.
- Realizar una búsqueda de los parámetros adecuados del algoritmo para que identifique el mayor número de colonias en cualquier muestra.
- Realizar pruebas de precisión y tiempo de ejecución.
- Desarrollar una aplicación web que utilice este algoritmo de recuento de colonias y entregue los datos al usuario.

1.3. Alcances

- Será una aplicación web.
- No se clasifica el tipo de colonia o bacterias, sólo realiza el recuento de ellas.
- La interacción con el usuario se hace mediante la carga de una imagen al sistema y entrega de resultados.
- La aplicación no interactúa con cámaras, ya sean, cámaras-web en caso de equipos Desktop o Laptop, o cámara integrada de un Smartphone.
- En promedio, los métodos de recuento no son 100 % exactos, sin embargo la aplicación debe lograr conteos con un 85 % de exactitud, como mínimo.
- El algoritmo mejora parte de la imagen, pero cualquier mejora anterior por parte del usuario puede ayudar al proceso.
- El algoritmo no determina si una imagen posee errores que entorpezcan el trabajo, sólo intenta llegar a un resultado.

2. Marco Teórico

En este capítulo presentaremos las bases sobre las cuales se desarrolla esta investigación. Abordaremos el área del procesamiento digital de imágenes, un conjunto de algoritmos y herramientas como escalado de colores, filtros de imagen, operadores y segmentación de características. Además, introduciremos hacia el mundo de la microbiología de las bacterias, su presencia en nuestro diario vivir, su importancia y por qué es necesario contarlas o estudiarlas. Finalizando con una breve descripción del estado del arte existente en el marco de contadores automáticos de colonias de bacterias.

2.1. Procesamiento Digital de Imágenes

El procesamiento digital de imágenes se basa en un conjunto de técnicas que permiten trabajar una imagen digital para mejorarla, destacar puntos o características de interés y extraer información relevante. En cuanto a su definición, no hay un acuerdo entre los distintos autores, con respecto a donde se detiene el procesamiento de imágenes y comienzan otras áreas relacionadas, como la visión por computadora o visión artificial. En [13] se nos menciona que en ocasiones se hace una distinción definiendo el procesamiento de imágenes como una disciplina en la que tanto las entradas como salidas de un proceso son imágenes.

Por otra parte, también da a entender que no hay límites claros en la trayectoria desde procesamiento de imágenes por un extremo hasta la visión por computadora en el otro. No obstante, un paradigma útil, para lograr su entendimiento, es considerar tres tipos de procesos computarizados (nivel bajo, medio y alto).

En el nivel más bajo, están las operaciones básicas, como pre-procesamiento de imagen, reducción de ruido, mejoras de contraste y filtros de mediana, entre otros. Se caracteriza por el hecho de que sus entradas y salidas son imágenes.

El nivel medio implica tareas como la segmentación, particionar regiones u objetos de la imagen, descripción de objetos para reducirlos a una forma adecuada para su procesamiento y el reconocimiento de objetos individuales. Este nivel es caracterizado porque sus entradas son imágenes, pero sus salidas son atributos extraídos de ellas, como bordes, formas o identidad de objetos individuales.

En el nivel superior, se busca dar sentido al conjunto de objetos reconocidos, podemos resumirlo como la etapa donde realizamos análisis de las características extraídas a la imagen.

TABLA 2.1: Resumen niveles de procesos.

Nivel	Entrada	Salida
Bajo	Imagen	Imagen.
Medio	Imagen	Atributos de imagen: bordes, contornos, áreas.
Alto	Objetos Identificados	Análisis: resultados, información, estadísticas.

Los procesos fundamentales que forman parte de los niveles Bajo y Medio, abordados en [13] son ilustrados en la Figura 2.1.

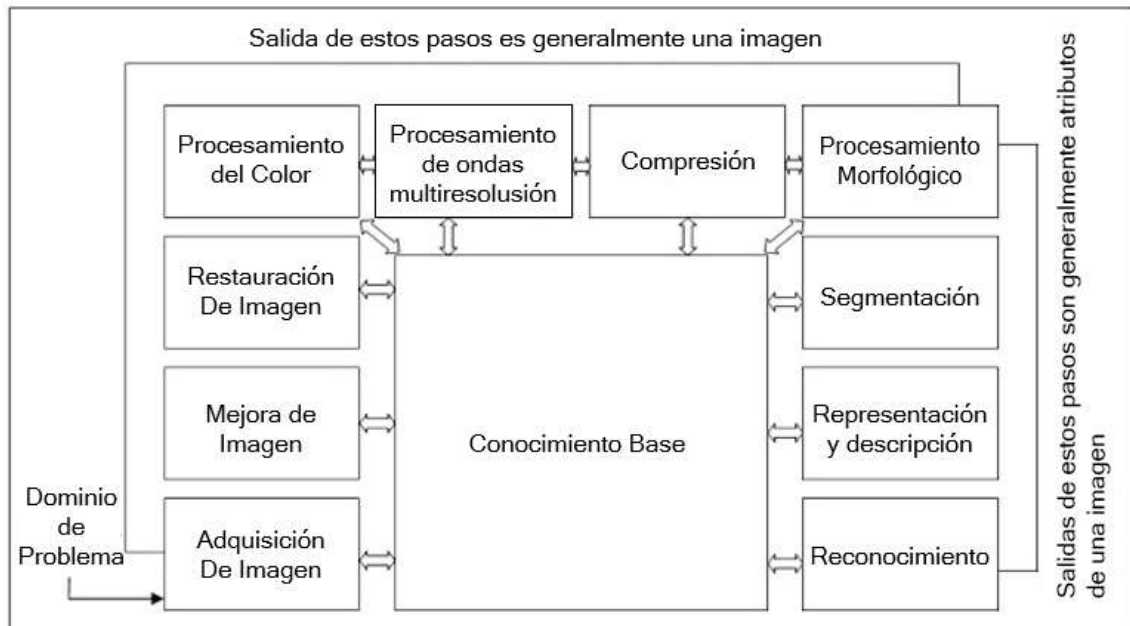


FIGURA 2.1: Procesos fundamentales del procesamiento digital de imágenes.

Como se puede apreciar, no existe un orden predefinido para aplicar los procesos, todo depende del dominio del problema y las propiedades que la imagen posea. La unión de ciertos procesos que van desde la Mejora de la imagen hasta el procesamiento morfológico se denomina normalmente como pre-procesamiento, y corresponden al nivel Bajo de la Tabla 2.1.

2.1.1. Adquisición de imagen

En este proceso obtenemos la imagen digital, mediante algún dispositivo de captura. Incluido el procesamiento como escalamiento de la imagen.

2.1.2. Mejora de imagen

En este proceso se modifica la imagen para adecuarla y mejorarla para el trabajo a realizar, buscando resaltar características de interés y así lograr una mejora sustancial, con algunos métodos como suavizado y filtro de mediana. La percepción de la mejora generalmente está criterio del observador.

2.1.3. Restauración de imagen

Este proceso también se busca mejorar la imagen. Mediante técnicas de restauración basadas en modelos matemáticos o probabilísticos de degradación de la imagen, como la transformada de Fourier.

2.1.4. Procesamiento del color

Es un proceso específico para las imágenes digitales a colores, abarca técnicas relacionadas al modelo e intensidad del mismo, por ejemplo escalado de grises.

2.1.5. Procesamiento de multiresolución

Es la base para representar imágenes en varios grados de resolución. Es utilizado mayormente en la compresión.

2.1.6. Compresión

Proceso mediante el cual se busca reducir el almacenamiento requerido para guardar una imagen o el ancho de banda necesario para transmitirla, como delta encoding.

2.1.7. Procesamiento morfológico

Utilizado para extraer componentes de la imagen que son útiles para la representación y descripción de formas. Es una técnica de procesamiento no lineal de

la señal, caracterizada por realzar la geometría y forma de los objetos, mediante un fundamento basado en la teoría de conjuntos.

2.1.8. Segmentación

Proceso en el cual se divide la imagen en objetos, distinguiéndolos del fondo, haciéndolos más visibles, con técnicas como operador de Sobel o Laplace.

2.1.9. Representación y Descripción

Proceso en el cual se extraen las características necesarias para representar un objeto y poder diferenciarlo de otros, como su forma, tamaño o área.

2.1.10. Reconocimiento

Proceso mediante el cual se identifican los objetos segmentados de acuerdo a sus descriptores, asignándoles una clase o etiqueta como por ejemplo "circunferencia". Más adelante en el documento trataremos este punto con mayor profundidad.

2.2. Imagen Digital

En el proceso de adquisición, esperamos obtener como salida una imagen. Sin embargo, podemos ver una imagen como la representación finita de una matriz de píxeles, los que a su vez son la unidad mínima en que puede dividirse una imagen digital, y solamente pueden representar a un color. Entonces, la cantidad de píxeles de una imagen depende del tamaño de esta; a mayor tamaño más píxeles tiene. Cabe destacar que un píxel no tiene un tamaño propio, ya que depende de la configuración y el tamaño del dispositivo que lo muestra.

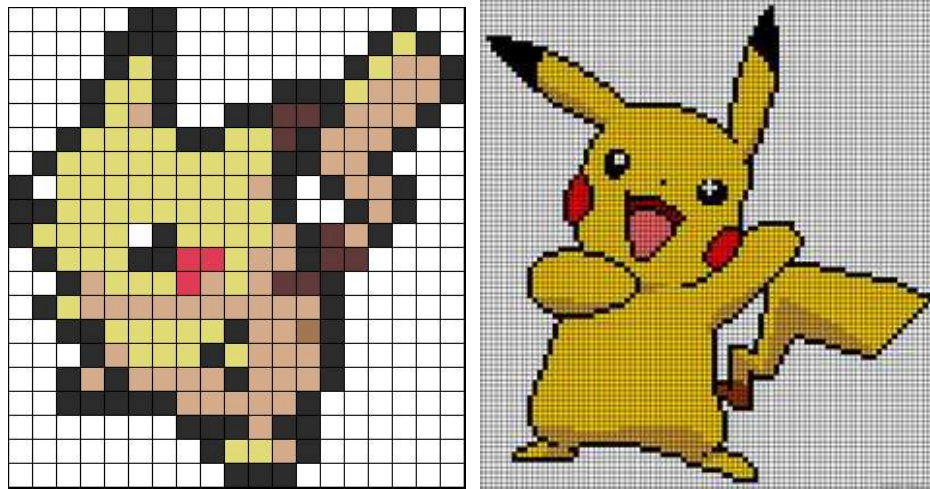


FIGURA 2.2: Diferente cantidad de píxeles para representar una misma figura.

En la Figura 2.2, podemos observar cómo la cantidad de píxeles actúa para representar una imagen del mismo personaje. En estos casos, un mayor número de píxeles representa un detalle mucho mayor, a un nivel casi imposible de alcanzar por la imagen de la izquierda. También requiere una representación matricial más extensa.

De ese modo, podemos asociar imagen a una materia más conocida y representarla como una función $p = f(x, y)$ donde p es un píxel que pertenece a la imagen y, a su vez, x, y son coordenadas espaciales de la misma. Luego, podemos afirmar que cuando x e y y todo valor de p evaluado en $f(x, y)$ tal que es un valor finito y discreto, se dice que es una imagen digital.

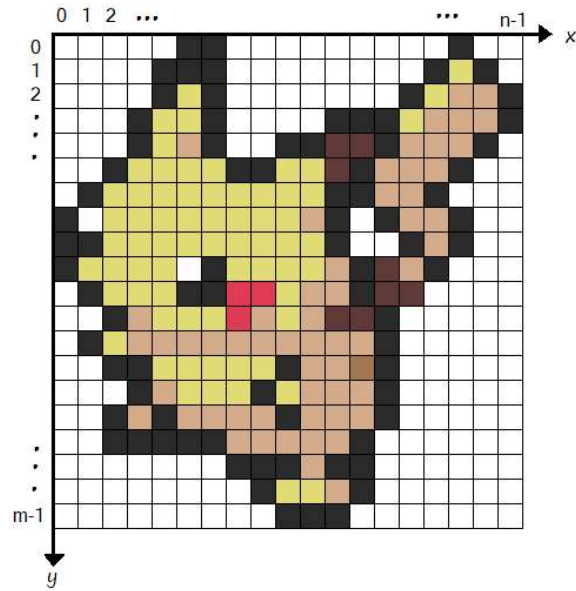


FIGURA 2.3: Representación matricial de Figura 2.2.

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, n-1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, n-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(m-1, 0) & f(m-1, 1) & \cdots & f(m-1, n-1) \end{bmatrix} \quad (2.1)$$

La Ecuación 2.1 representa la definición matricial de una imagen como una la función.

2.2.1. Profundidad de Color

Es un término con el que se conoce al valor asociado al número o cantidad de bits utilizados para representar el color de un solo píxel, es decir, la cantidad de opciones a los posibles colores que puede tomar un píxel. También se conoce como bits por píxel (bpp).

Por ejemplo, si la profundidad de color de una imagen es 1 bit, que es lo mismo a decir $2^1 = 2$ opciones (o colores) y estas serán blanco o negro. Del mismo modo, 2 bits darán acceso a 4 colores, 3 bits a 8 colores, 4 bits a 16 y así sucesivamente.



FIGURA 2.4: Una profundidad de color diferente para una misma imagen genera estos resultados, de izquierda a derecha tenemos la imagen representada con 1 bit, 4 bits y 32 bits.

Cabe destacar que por debajo de una profundidad de color de 24 bits (conocido como color verdadero), las imágenes que pueden llegar a representarse son bastante limitadas desde el punto de vista del detalle que puede presentarse.

2.2.2. Escala de Grises

En una escala de grises, una imagen digital contiene una profundidad de color de 8 bits, 256 opciones para los colores de sus píxeles. Donde el 0, la mínima intensidad de color, es negro y 255 corresponde a blanco. Los valores que podemos encontrar en el rango de [0-255] son diferentes variaciones del color gris [3].

Para realizar el cálculo del valor de un píxel en la escala de grises, se utiliza el modelo RGB.

2.2.3. Modelo RGB

Este modelo está basado en los colores rojo, verde y azul, sus siglas provienen del inglés Red, Green y Blue. Es el modelo más utilizado por las imágenes a color y el sistema más frecuente en aparatos como televisores, monitores de computadoras, cámaras fotográficas, etc. En este modelo, la imagen tiene una profundidad de 24 bits, utilizando 3 canales de 8 bits para cada color.

Para llevar a cabo esta investigación es necesario convertir la imagen RGB a escala de grises. El método de conversión más simple consiste en un Método Promedio [10] que consiste en sumar la intensidad de cada píxel y dividirlo entre

3.

$$GrayScale = (R + G + B)/3 \quad (2.2)$$

No obstante, este método puede entregar un resultado distinto al esperado pues las imágenes obtenidas son, en su mayoría, más oscuras de lo esperado. Esto surge debido a que los tres colores tienen longitudes de onda y su propia contribución a la imagen. Para lograr un mejor resultado es preciso utilizar el promedio de acuerdo a cada distribución. Utilizar el Método Ponderado.

En el Método Ponderado, o Método de Luminosidad, dado que el color rojo tiene una mayor longitud de onda de los tres colores, y el verde es el que tiene no sólo menos longitud de onda que color rojo, sino que también proporciona un efecto de suavizado a los ojos. Por lo tanto, debemos disminuir la contribución de rojo, aumentar la contribución del color verde y dejar el azul como el resto. Finalmente, la nueva ecuación que vincula la escala de grises con el modelo RGB es:

$$GrayScale = 0,3R + 0,59G + 0,11B \quad (2.3)$$

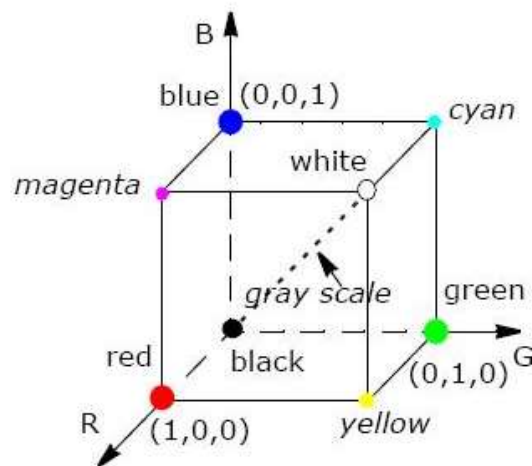


FIGURA 2.5: Modelo RGB que contiene escala de grises. Fuente [24].

2.3. Histograma

Un histograma es una representación gráfica de la distribución de los distintos tonos de una imagen digital. Puede ayudarnos a controlar la exposición de las fotografías, así como también corregir los colores. En términos matemáticos, un histograma es la representación gráfica de la distribución estadística de los niveles de intensidad que posee una imagen digital, que está dentro de un rango $[0, F]$ definido como una función discreta tal que $H(F_k) = N_k$ donde g_k es el k -ésimo nivel de intensidad en el intervalo y N_k es la cantidad de píxeles en la imagen que tiene un nivel de intensidad g_k .

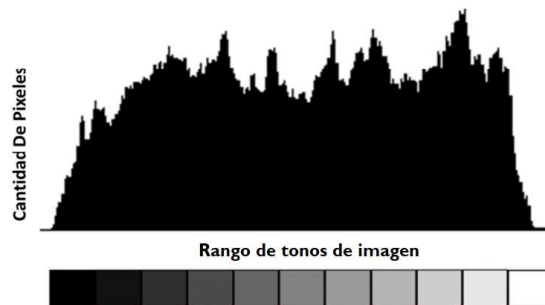


FIGURA 2.6: Histograma de imagen en escala de grises.

El histograma es la base de numerosas técnicas del dominio espacial de una imagen, como la ecualización, donde los píxeles se redistribuyen en los niveles de intensidad que sean necesarios para mejorar el contraste de una imagen. Además, nos permite identificar a simple vista, por medio de la distribución que presentan sus píxeles en el gráfico, la exposición de una imagen. Observar la gráfica del histograma de una imagen nos permite averiguar rápidamente si ésta presenta tonos apagados, zonas oscuras, sombras pálidas, etc. No necesariamente indica que sea una imagen sin utilidad, todo depende del uso que se le vaya a dar. En síntesis, es una herramienta que nos permite detectar a tiempo características que deseamos evitar y corregirlas

2.4. Selección por Umbral

La segmentación es una técnica en la cual se tiene como objetivo dividir la imagen en objetos de interés del resto de la imagen, de acuerdo al histograma.

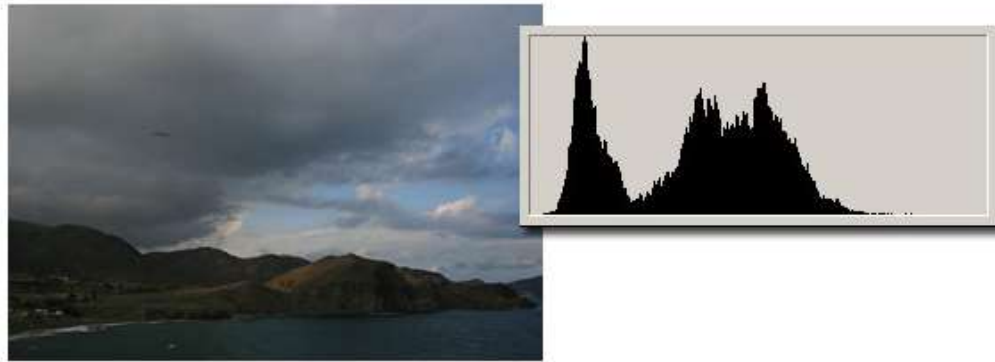


FIGURA 2.7: Imagen con tonos apagados, característica que puede ser reconocida al analizar el histograma.

Realizar la conversión de la imagen original a escala de grises ayuda a mejorar los resultados que podemos obtener. Si bien, la umbralización es una técnica de segmentación (ver Figura 2.4), su salida suele ser una imagen binarizada, es decir, blanco y negro, con una profundidad de 1 bit.

2.4.1. Umbralización de Imagen

Si suponemos que el histograma $f(x, y)$ en una imagen de escala de grises (8 bit), tiene niveles de intensidad de color que son oscuros (cerca de 0) y el fondo claro (más cerca de 255). Una forma de separar los objetos del fondo de la imagen es seleccionar un umbral T dentro del rango $[0, 255]$, para luego procesar la imagen y determinar si cada píxel es parte del fondo, mayor que T , o es un objeto, en caso contrario. Por tanto, la función para binarizar una imagen $b(x, y)$, está dada por:

$$b(x, y) = \begin{cases} 0 & \text{si } f(x, y) \leq T \\ 1 & \text{si } f(x, y) > T \end{cases} \quad (2.4)$$

El nivel de dificultad de separar los objetos y el fondo utilizando este método, así como su correctitud dependen netamente de su imagen y su histograma. Según [18], los factores que aumentan la dificultad son el bajo contraste, la iluminación del ambiente, el ruido y los tamaños desproporcionados de los objetos. En el mismo [18], los algoritmos de umbralización se clasifican de acuerdo a la información que se pueda extraer del histograma. En este proyecto de investigación se utiliza el método de Isodata.

2.4.2. Umbralización mediante Método IsoData

Es una técnica iterativa utilizada para la obtención del umbral correcto [18]. Para comenzar, tomamos un valor inicial para el umbral u y calculamos un promedio con los menores valores de u ($p_0(u)$) y luego los mayores valores para u ($p_f(u)$) [19]. Para luego, calcular el promedio de la iteración dado por:

$$PI = \frac{p_0(u) + p_f(u)}{2}; p_0(u) = \sum_u^{g=0} gh(g); p_f(u) = \sum_G^{g=u+1} gh(g) \quad (2.5)$$

Donde $h(g)$ es la cantidad de píxeles que tienen un nivel de intensidad g y G es el máximo nivel de intensidad del histograma h . En las siguientes iteraciones se incrementa el umbral, $u = u + 1$, y se calcula $PI(u)$. Las iteraciones se detienen cuando el valor de $|PI(u - 1) - PI(u)|$ es menor a una constante predefinida, resultando como umbral óptimo el valor de $PI(u)$ redondeado a entero. Existen variaciones de este algoritmo que se diferencian en la fórmula de $p_0(u)$ y $p_f(u)$, en el umbral inicial y en la condición para finalizar las iteraciones.

2.5. Reducción de ruido

El ruido de una imagen digital son aquellos píxeles que tienen un valor que no corresponden al real. Son un atributo negativo que desfavorece el aspecto visual y entorpece el proceso de segmentación. Su origen se produce en la adquisición y ocurre por motivos ambientales, niveles de luz, oscuridad, sombras, partículas en el aire, entre otros. Un factor fundamental es la calidad de los sensores de luz de los dispositivos con los que realizamos la adquisición.

La reducción de ruido consiste en restaurar una imagen para procesarla, y sus técnicas se clasifican como filtros de suavizado.

2.5.1. Filtro de Mediana

El filtrado de mediana es una técnica de filtrado digital no lineal utilizada para eliminar el ruido de una imagen o señal. Corresponde a un paso de procesado de

una imagen digital, la cual ayuda a mejorar los resultados de la segmentación o pasos posteriores como detección de bordes. El filtro de mediana es uno de los más utilizados debido a que bajo ciertas condiciones conserva los bordes eliminando el ruido, siendo sumamente efectivo en los casos de Ruido sal y pimienta (salt and pepper noise).



FIGURA 2.8: Filtro de mediana aplicado en una imagen que presenta ruido del tipo Sal y Pimienta.

Para mostrar mejor la idea del funcionamiento del filtro de mediana, explicaremos el funcionamiento mediante un pseudo-código y su ejecución. Para ello, definimos $X = [2, 80, 6, 3]$.

Result: Entrada filtrada por mediana.

```

N = length(X);
med = [ ];
while i < N do
    if i = 0 then
        | y = [X[i], X[i], X[i + 1]];
    else
        | if i = N-1 then
            | y = [X[i - 1], X[i], X[i]];
        else
            | y = [X[i - 1], X[i], X[i + 1]];
        end
    end
    sort(y);
    med[i] = y[len(y)/2];
end

```

Algorithm 1: Filtro de Mediana.

El resultado para X con este algoritmo de Filtro de Mediana es $[2, 6, 6, 3]$. No obstante, este resultado está aplicado en sólo una dimensión. Si nos mudamos al mundo 2D, donde están contenidas las imágenes, podemos aplicar el algoritmo anterior sin modificaciones fila por fila. Sin embargo, existen mejoras para el algoritmo basadas en la utilización de matrices menos complejas, linealizándolas para recurrir a las mismas reglas del principio anterior.

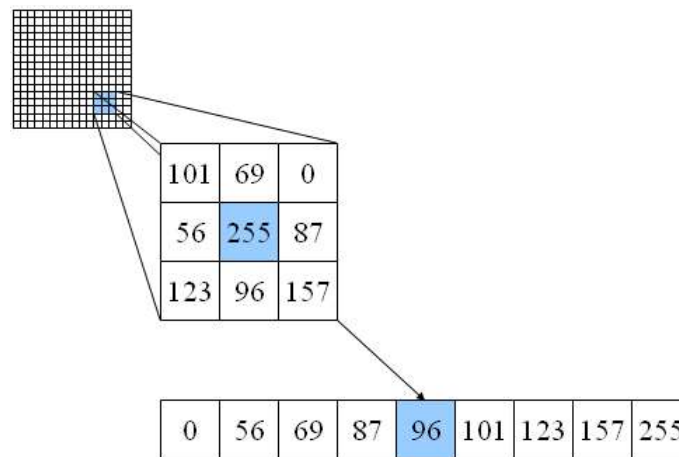


FIGURA 2.9: Filtro de mediana en una imagen con matriz de 3x3.

2.6. Detección de Bordes

La detección o realce de bordes es una técnica de segmentación fundamental en el procesamiento digital de imágenes, específicamente en detección y extracción de características. Tiene como objetivo identificar puntos en que una imagen cambia drásticamente, o dicho de manera formal, presenta discontinuidades.

Para realizar su cometido, se utilizan métodos basados en la identificación de variaciones fuertes de la intensidad que corresponden a fronteras de los objetos visualizados. Estos métodos se basan en el gradiente y determinan los bordes en base a las derivadas espaciales de la imagen que se calculan mediante operadores de convolución.

Para una imagen $Z = E(x, y)$, el gradiente está dado por:

$$\text{grad}(E) = \left(\frac{\delta E}{\delta x}, \frac{\delta E}{\delta y} \right)^T \quad (2.6)$$

Sus vectores normales son:

$$n_0 = \left(\frac{\delta E}{\delta x}, \frac{\delta E}{\delta y}, -1 \right)^T \quad n_1 = \left(-\frac{\delta E}{\delta x}, -\frac{\delta E}{\delta y}, +1 \right)^T \quad (2.7)$$

Usaremos el vector gradiente definido como:

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix} \quad (2.8)$$

Y el módulo o magnitud de este vector es:

$$\nabla f = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2} = \sqrt{\frac{\delta f^2}{\delta x} + \frac{\delta f^2}{\delta y}} \quad (2.9)$$

Para simplificar los cálculos, los valores de las derivadas parciales de primer orden (g_x y g_y) se aproximan utilizando filtros espaciales, operados sobre una vecindad de píxeles de 3x3 en la imagen a procesar. De forma que un valor aproximado para el gradiente δf o G , depende del filtro que se aplique para calcular g_x y g_y , donde:

$$G = \sqrt{g_x^2 + g_y^2} \quad (2.10)$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

TABLA 2.2: Gradiente de píxeles en una vecindad de 3x3.

2.6.1. Operador de Sobel

Es un operador que calcula una aproximación del gradiente de la función de intensidad de la imagen. En cada punto, el resultado del operador es el vector

gradiente correspondiente a la norma del vector. Se basa en la convolución de la imagen con un filtro pequeño, separable y de valores enteros en las direcciones horizontal y vertical y, por lo tanto, es relativamente económico en términos de cálculos. Sin embargo, la aproximación de gradiente que produce es relativamente tosca, en particular para las variaciones de alta frecuencia en la imagen.

(a) Horizontal	(b) Vertical																		
<table border="1" style="border-collapse: collapse; width: 60px; height: 60px;"> <tr><td style="padding: 2px 10px;">-1</td><td style="padding: 2px 10px;">-2</td><td style="padding: 2px 10px;">-1</td></tr> <tr><td style="padding: 2px 10px;">0</td><td style="padding: 2px 10px;">0</td><td style="padding: 2px 10px;">0</td></tr> <tr><td style="padding: 2px 10px;">1</td><td style="padding: 2px 10px;">2</td><td style="padding: 2px 10px;">1</td></tr> </table>	-1	-2	-1	0	0	0	1	2	1	<table border="1" style="border-collapse: collapse; width: 60px; height: 60px;"> <tr><td style="padding: 2px 10px;">-1</td><td style="padding: 2px 10px;">0</td><td style="padding: 2px 10px;">1</td></tr> <tr><td style="padding: 2px 10px;">-2</td><td style="padding: 2px 10px;">0</td><td style="padding: 2px 10px;">2</td></tr> <tr><td style="padding: 2px 10px;">-1</td><td style="padding: 2px 10px;">0</td><td style="padding: 2px 10px;">1</td></tr> </table>	-1	0	1	-2	0	2	-1	0	1
-1	-2	-1																	
0	0	0																	
1	2	1																	
-1	0	1																	
-2	0	2																	
-1	0	1																	

FIGURA 2.10: Filtro de Sobel.

Así, obtendremos los valores de:

$$g_x = z_7 + 2z_8 + z_9 - (z_1 + 2z_2 + z_3) \quad g_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7). \quad (2.11)$$

2.6.2. Operador de Prewitt

Al igual que el anterior realiza su función mediante la aproximación del gradiente de la función de intensidad de la imagen, aplicando filtros horizontales y verticales sobre una vecindad de píxeles.

(a) Horizontal	(b) Vertical																		
<table border="1" style="border-collapse: collapse; width: 60px; height: 60px;"> <tr><td style="padding: 2px 10px;">-1</td><td style="padding: 2px 10px;">-1</td><td style="padding: 2px 10px;">-1</td></tr> <tr><td style="padding: 2px 10px;">0</td><td style="padding: 2px 10px;">0</td><td style="padding: 2px 10px;">0</td></tr> <tr><td style="padding: 2px 10px;">1</td><td style="padding: 2px 10px;">1</td><td style="padding: 2px 10px;">1</td></tr> </table>	-1	-1	-1	0	0	0	1	1	1	<table border="1" style="border-collapse: collapse; width: 60px; height: 60px;"> <tr><td style="padding: 2px 10px;">-1</td><td style="padding: 2px 10px;">0</td><td style="padding: 2px 10px;">1</td></tr> <tr><td style="padding: 2px 10px;">-1</td><td style="padding: 2px 10px;">0</td><td style="padding: 2px 10px;">1</td></tr> <tr><td style="padding: 2px 10px;">-1</td><td style="padding: 2px 10px;">0</td><td style="padding: 2px 10px;">1</td></tr> </table>	-1	0	1	-1	0	1	-1	0	1
-1	-1	-1																	
0	0	0																	
1	1	1																	
-1	0	1																	
-1	0	1																	
-1	0	1																	

FIGURA 2.11: Filtro de Prewitt.

Podemos observar como cambian los valores de los filtros, y por tanto:

$$g_x = z_7 + z_8 + z_9 - (z_1 + z_2 + z_3) \quad g_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7). \quad (2.12)$$

2.6.3. Operador de Roberts

El operador o detector de bordes de Roberts [1963] es uno de los primeros detectores de bordes y se logra calculando la suma de los cuadrados de las diferencias entre los píxeles diagonalmente adyacentes, mediante la aplicación de filtros.

Gradiente fila	Gradiente columna																		
<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>-1</td><td>0</td></tr> </table>	0	0	0	0	0	1	0	-1	0	<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <tr><td>-1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	-1	0	0	0	1	0	0	0	0
0	0	0																	
0	0	1																	
0	-1	0																	
-1	0	0																	
0	1	0																	
0	0	0																	

FIGURA 2.12: Filtro de Roberts.

Al igual que en la anterior, dada la definición de gradiente y como mostramos la vecindad, el gradiente para el operador de Roberts está dado por:

$$g_x = z_5 - z_1 \quad g_y = z_6 - z_8. \quad (2.13)$$

2.6.4. Operador de Laplace

El operador de Laplace o laplaciano es la derivada de segundo orden de una función, y está definida por:

$$\nabla^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2} \quad (2.14)$$

Realiza la detección de bordes mediante su propiedad de paso por 0, *zero crossing*. A pesar de que los resultados del laplaciano suelen bordes falsos, esto ocurre en regiones donde la varianza local es pequeña. Sin embargo, en imágenes donde la varianza local es suficientemente grande, como imágenes con profundidad de 2 bits, los resultados son, en general mejores.

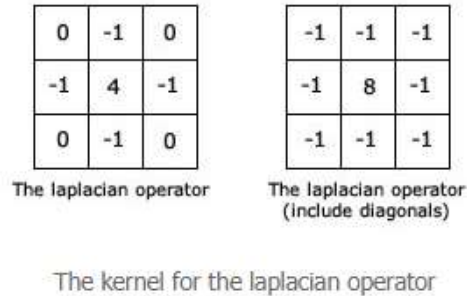


FIGURA 2.13: Filtro de Laplace.

2.7. Transformada de Hough

La Transformada de Hough es un método de extracción de características propuesto por Paul Hough en 1962 para la detección de bordes en imágenes, parametrizando y representándolas por medio de ecuaciones conocidas. Fue diseñado para la extracción de líneas rectas, aunque actualmente ha sido extendido para identificar otras formas como círculos o elipses.

Trabaja mediante la transformación del espacio cartesiano xy de la imagen a un espacio de parámetros y un proceso de votación para identificar los puntos que describe una línea recta u otra figura.

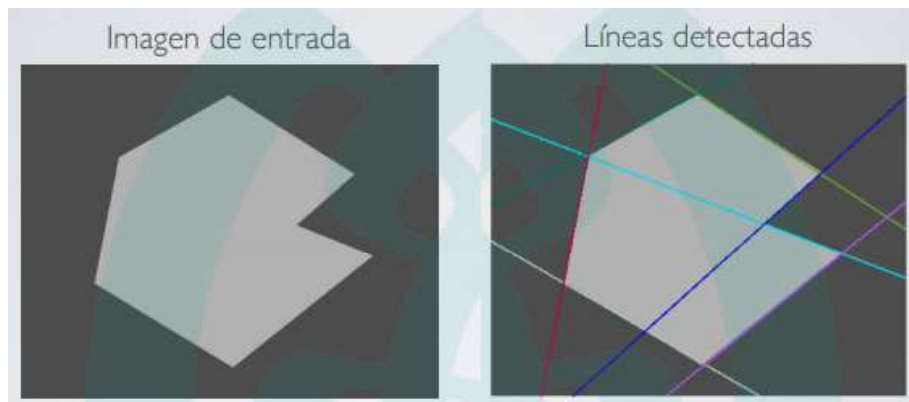


FIGURA 2.14: Aplicación transformada de Hough con líneas rectas.

Para ejemplificar el proceso que se lleva a cabo en el uso de Transformada de Hough, consideremos la siguiente situación. Digamos que fotografiamos un pos-

te y calculamos los píxeles del borde (utilizando algún detector de bordes como Canny, Sobel u otro). Ahora para una representación geométrica del borde del poste deseamos conocer su pendiente, sus puntos de intersección, etc. Pero ahora el borde es sólo una secuencia de píxeles. Podemos recorrer todos los píxeles y, de alguna forma, calcular la pendiente e interceptar. Pero esa es una tarea difícil. Las imágenes nunca son perfectas y, por lo tanto, buscamos algún mecanismo que otorgue más peso a los píxeles que ya están en una línea. Eso es lo que realiza el algoritmo de Transformada de Hough. Otorga a cada punto de la imagen un valor mediante una votación. Debido a las propiedades matemáticas de la transformación, esta votación nos permite descubrir líneas prominentes en la imagen. Para identificar las colonias utilizamos el mismo principio, sin embargo, llevado a un enfoque diferente pues presentan formas circulares, lo que hace que sus bordes no se encuentren totalmente claros cuando miramos los píxeles.

2.8. Bacterias

Las bacterias, también llamadas microbios, son microorganismos procariontes unicelulares, todas con formas y tamaños diferentes, por lo general en el orden de los micrones y que sólo pueden ser vistas por medio de un microscopio. Son esenciales para la realización de muchos procesos que ocurren diariamente [14], como fermentación de algunos alimentos, desarrollo de nuevos medicamentos, producción de oxígeno en el planeta [11], curar o estimular plantas; y en nuestras vidas, mantener el equilibrio del organismo, curarnos o prevenir cualquier afección corporal.

En ocasiones, pueden ser infectadas por virus, los cuales las utilizan como un medio para reproducirse. Del mismo modo, algunas bacterias son responsables de millones de muertes de personas a nivel mundial. Entre algunas de las enfermedades infecciosas bacterianas, causantes de grandes epidemias se encuentran: difteria, cólera, tuberculosis, sífilis, tétanos, tos ferina y fiebre tifoidea, entre otras [14].

Por esta razón, es esencialmente importante realizar investigaciones con bacterias, para evitar pandemias mortales, encontrar curas a enfermedades o como

un medio de desarrollo de nuevos medicamentos.

2.8.1. Morfología de Bacterias

Como ya se dijo anteriormente, las bacterias muestran un gran rango de diferentes formas y tamaños, lo que se conoce como morfología, la cual cambia dependiendo principalmente entre las diferentes especies.

En general, las bacterias tienen un tamaño de 0.5 a 5 micrómetros, aunque algunas pueden llegar a medir hasta medio milímetro de largo y pueden ser visualizadas sin la necesidad de un microscopio. Del mismo modo, hay registros de pequeñas bacterias que pueden llegar a medir la solamente hasta 0.3 micrómetros.

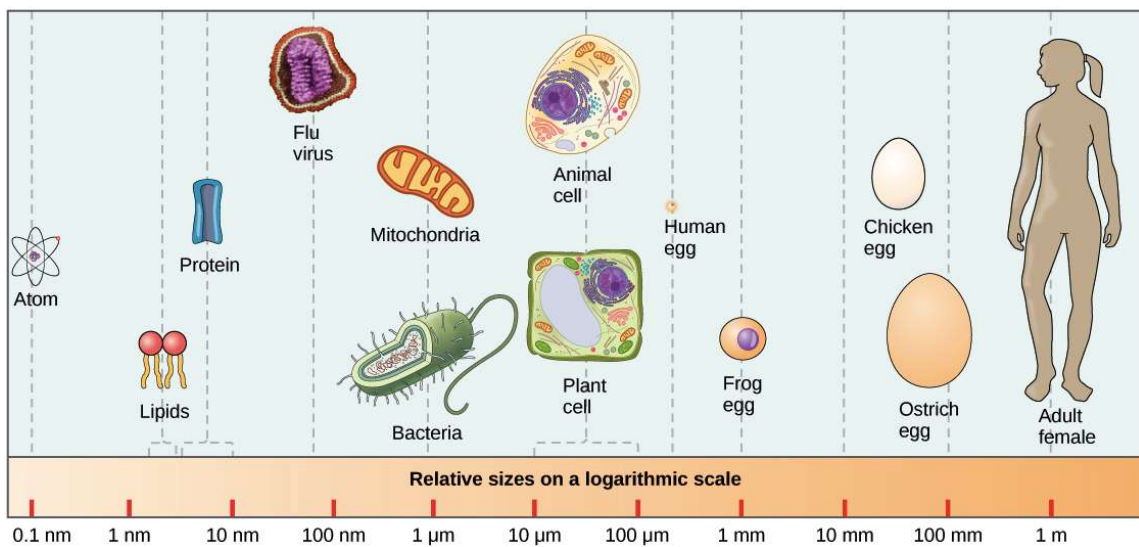


FIGURA 2.15: Tamaño bacterias.

En la morfología de las bacterias nos encontraremos con que gran parte de las especies son esféricas (cocos) o en forma de barra (bacilos, la elongación suele estar relacionada con su capacidad de nadar). Otras tantas formas de bacterias, son más bien curvas, en forma de espiral o enrolladas, e incluso algunas pueden llegar a tomar la forma cuboide, tetraédrica.

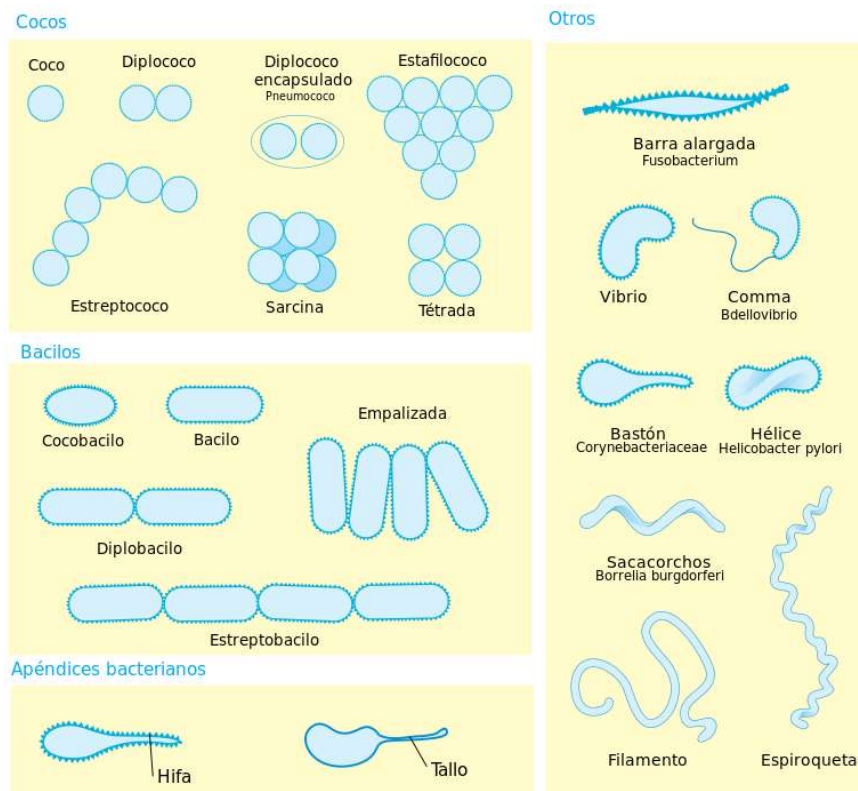


FIGURA 2.16: Morfología Bacteria.

Todos los diferentes tipos morfológicos que poseen las bacterias son determinados por su pared celular y su citoesqueleto. Sus formas son tan importantes que llegan a determinar la capacidad de sobrevivir y conseguir nutrientes.

2.8.2. Agrupación de Bacterias

Si bien existen especies de bacterias que viven y se desenvuelven como una sola célula simple, se establece que otras se agrupan para llevar a cabo su ciclo de vida; pueden ser pares, cadenas o forma de racimos, tal como se muestra en la Figura 2.17.

Podemos clasificar a las agrupaciones de bacterias en 2 grandes grupos principales: las agrupaciones microscópicas y agrupaciones macroscópicas.

Las agrupaciones microscópicas, dependen de una bacteria aislada y la rigidez de su pared celular, pues está estrechamente relacionado a cómo se lleva a cabo la fisión binaria (método de reproducción de célula procariota) de la célula y que

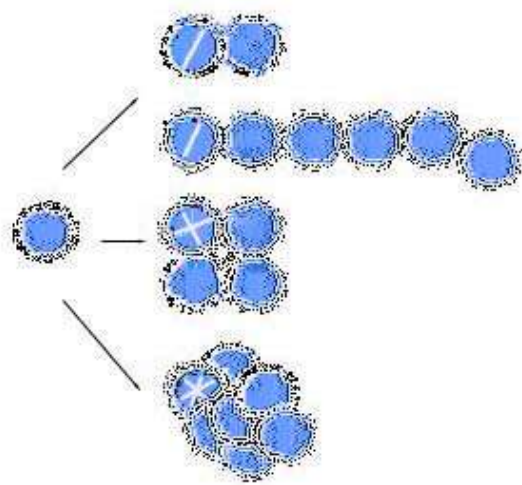


FIGURA 2.17: División bacterias tipo coco.

las células resultantes se queden unidas o no. La agrupación es una característica que se da en miembros del mismo grupo taxonómico.

Existen los grupos de formas cocoides y bacilares. En las formas cocoides nos encontramos con los diplococos, tétradas, estafilococos y sarcinas, mientras que en las bacilares están los diplobacilos, estroptobacilos y formas irregulares que muchas veces se asemejan a letras chinas [2]. Ver Figura 2.16.

A las agrupaciones macroscópicas las llamamos colonias.

2.8.3. Colonias

Estas agrupaciones se caracterizan por ser visualizadas a simple vista, corresponden al crecimiento en un punto de una única línea celular hasta formar una colonia (todas las células que componen la colonia provienen de una única célula). El medio donde crecen se caracteriza por ser un medio óptimo para su crecimiento y sólido, que impide el movimiento de los individuos de la colonia y favorece su agrupación, para formar colonias distintas y de características para cada tipo [9].

2.8.4. Importancia del Conteo de Colonias

El cultivo de colonias es una actividad en la cual se realizan estudios para identificar los agentes patógenos responsables del proceso infeccioso y conocer

las implicaciones patogénicas/patológicas, la evolución clínica y aplicar una terapia antimicrobiana eficaz para combatirla [6]. En caso de que sea lo contrario, de ser necesario, como acelerar el proceso de reproducción de una célula altamente beneficiosa para un proceso, como por ejemplo, la producción de oxígeno [11].

Dado que el conteo es utilizado para señalar la cantidad de colonias, se puede realizar por diversas técnicas que se basan en el tipo de medida: conteo celular, directamente en un microscopio, contador electrónico de partículas o conteo de colonias (denominadas UFC, unidad formadora de colonia); masa celular, pesando el contenido; y actividad celular, relacionando el grado de actividad bioquímica y al tamaño de la población bacteriana. Para llevar a cabo este trabajo se utiliza el conteo de colonias de bacterias.

2.8.5. Métodos de Conteo más Utilizados

Conteo en cápsula de Petri

Es el método más usado para contar colonias de bacterias, consiste en contar directamente cada una de las colonias presentes en la cápsula. Es muy importante considerar un número limitado de colonias, en caso contrario, estas podrían sobrepoblarse y dificultar el proceso. El rango sugerido, según la FDA (Administración de Alimentos y Medicamentos o Food and Drug Administration, por sus siglas en inglés), es de 25 a 250 colonias [15].

Este conteo se facilita mediante el uso de un contador de colonias, el cual es un método deseable pues entrega el total de células viables (vivas). Su principal desventaja es el tiempo requerido para, en primer lugar, realizar el cultivo, con un mínimo de 24 horas; segundo, por el tiempo que toma contar cada una de las colonias, en especial cuando su número se acerca al borde; y por último, no es ciento por ciento fiable pues generalmente las colonias crecen unidas en cadenas o grupos, lo cual hay que evitar de ser posible.

Para efectos de este proyecto, este será el método que abordaremos para agilizar el segundo punto anteriormente descrito, el tiempo que toma realizar el conteo del total de colonias presentes en una cápsula de Petri.

Número más probable

Es una técnica de estimación estadística que aproxima un valor total de células bacterianas a partir de la cantidad de unidades formadoras de colonias. Está basado en el hecho de que a mayor número de bacterias en una muestra, mayor será la dificultad para realizar el conteo. Es más útil cuando se tiene constancia de que las células a contar no continuarán creciendo en el medio en que se encuentran. El método del Número más Probable utiliza una media geométrica para estimar el total de bacterias presentes en una muestra. Además, existen tablas predefinidas que ahorran a los investigadores estar realizando el proceso del cálculo. También es utilizado para determinar el número de colonias [4].

Se ha determinado que este método tiene una precisión probabilística aproximada del 95% de que la población bacteriana se encuentre dentro del rango correcto [12].

Determinación directa en microscopio

En este método, se cuentan el número de células directo en un microscopio, las cuales son teñidas y luego contadas. Dado que no es práctico recorrer todo el área de un cápsula, se realiza un conteo de un número de células en unos cuantos campos microscópicos seleccionados al azar, por ejemplo, se realiza el conteo en un campo de 1 cm^2 y luego se divide por 100, para convertirlo a 0.01 mL, pudiendo estimar el número de bacterias por mililitro. Por esta razón, a pesar de ser utilizado, es un método bastante criticado debido a la falta de uniformidad que podría llegar a presentar.

2.9. Trabajo Existente

Durante la concepción de este proyecto, uno de los puntos importantes fue realizar una investigación acerca de la existencia de un contador de colonias automático en el mercado. Si bien, la idea no es nueva, existen unas pocas empresas que comercializan contadores de colonias automáticos. Algunos puntos muy importantes a convenir a la hora de obtener uno de estos productos o de encontrar motivos para realizar un propio contador automático de bacterias son el valor co-

mercial y las especificaciones técnicas.

Los productos distribuidos por las distintas empresas presentan como mayor desventaja frente a otras soluciones el alto precio que estos alcanzan [21]. Además no hay mucha documentación de la precisión o funcionamiento en general del dispositivo que se comercializa. Para obtener esa información es necesario adquirir un producto.

Por otra parte, también se han realizado investigaciones y estudios acerca de como realizar el proceso utilizando tecnologías de más fácil acceso. Desde hace unos años, investigadores, institutos y universidades, se encuentran realizando estudios de cómo realizar el proceso de forma menos costosa y con buenos resultados.

De los resultados obtenidos por distintos investigadores se destacan los siguientes:

TABLA 2.3: En resultado de [16] vemos como la desviación estándar presentada es de un 10 %, en general. Sin embargo, algunos resultados se escapan de esta desviación para presentar.

Image	Actual Count	Manual Count
1	25	25
2	10	10
3	97	91
4	29	29
5	52	45
6	54	37
7	118	127
8	31	30
9	197	201
10	128	122

TABLA 2.4: El resultado de [20] vemos como el error presentado al realizar el conteo de las colonias es, en promedio, de un 4 %.

# colonias	muestras	erradas[%]	falsas[%]
0 - 17	21	1.47	0
18 - 20	19	2.99	0
21 - 23	25	3.31	0
24 - 26	29	4.00	0
27 - 29	27	3.28	0
30 - 40	35	4.17	0
41 - 49	30	4.00	0
50 - 60	29	3.87	2.00
>60	30	5.36	3.22

2.10. Resumen Marco Teórico

A lo largo de este capítulo vimos los conceptos y procesos que aborda el Procesamiento Digital de Imágenes, indispensables para esta investigación, tales como el escalado de grises, filtros de mediana, detección de bordes y segmentación.

Del mismo modo, nos adentramos en el mundo de las diminutas bacterias. Caracterizándolas y numerando algunas de sus cualidades que las hacen especiales y a la vez peligrosas. Definimos la importancia de realizar el conteo y algunas propiedades que poseen los métodos a la hora de definir restricciones.

3. Diseño del Sistema

El objetivo de este capítulo es dar a conocer el modelo y funcionamiento del sistema. Informar acerca de los componentes utilizados durante la implementación y modelado, describiendo las tecnologías que nos facilitaron el realizar el conteo y la metodología con la cual se trabajó para llevar a cabo el desarrollo.

3.1. Diseño del sistema

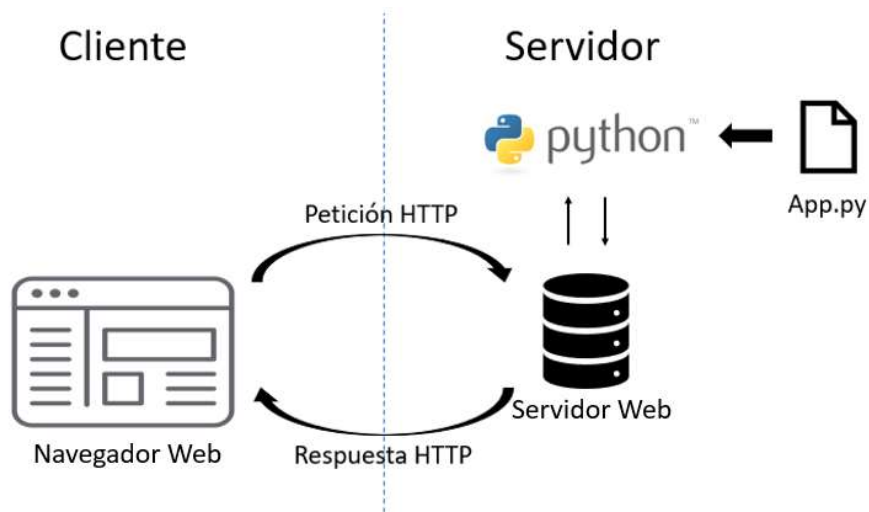


FIGURA 3.1: Diseño de arquitectura del sistema.

En la Figura 3.1 se esquematiza la arquitectura Cliente-Servidor diseñada para el sistema. Está compuesta lógicamente por el servidor que provee el servicio y el cliente que lo solicita.

En este caso en particular, el cliente accede al servicio mediante una petición HTTP, la cual le permite cargar una imagen digital, para ser enviada al Servidor para ser procesada y analizada por la aplicación que se ejecuta sobre en Python 2.7. La aplicación se encarga de realizar las tareas de:

- Mejoramiento de imagen.
- Procesamiento de imagen.
- Segmentación de características.
- Interpretación de datos.

Es necesario destacar que el servidor web no entrega información acerca de que proceso se encuentra realizando el algoritmo.

El algoritmo implementado utiliza la arquitectura tipo pipeline [17] para realizar los procesos, pues cada uno de ellos necesita la salida del anterior para realizar una acción concreta. Además, no se encarga de determinar si la imagen a trabajar es adecuada para llevar a cabo el proceso o no, solamente muestra la información que pueda interpretar al usuario.

El usuario es quién está encargado de llevar a cabo el proceso de adquisición de imagen de la colonia que desee analizar. Por lo tanto, cualquier mejora previa que este pueda realizar como por ejemplo, un fondo de alto contraste o tomar una fotografía utilizando instrumentos de precisión, como una cámara térmica, que realza las propiedades que son buscadas, aporta significativamente al proceso.

3.2. Componentes

3.2.1. OpenCV

OpenCV (Open Source Computer Vision) es una librería de código abierto desarrollada por la compañía Intel. Inicialmente fue escrita en C, lo que la llevo a ser portable a casi cualquier lenguaje y plataforma (Windows, Linux, MacOS, Android, iOS, Maemo, CUDA, entre otros).

Sus últimas versiones contienen más de 2500 algoritmos optimizados que abarcan detección de movimiento, visión por computadora, reconocimiento de objetos, calibración de cámaras, visión etérea y robótica [22]. En este proyecto en específico se utilizó la versión 2.4 de OpenCV para Python 2.7 en Windows, liberada en 2016.

En la Figura 3.2 se muestra que OpenCV presenta una estructura de capas. Donde, en la capa superior se encuentra el sistema operativo, luego están los enlaces del lenguaje. Debajo se encuentra el código contribuido con “opencv_contrib” que contiene la funcionalidad en alto nivel. Finalmente están el núcleo y las opti-

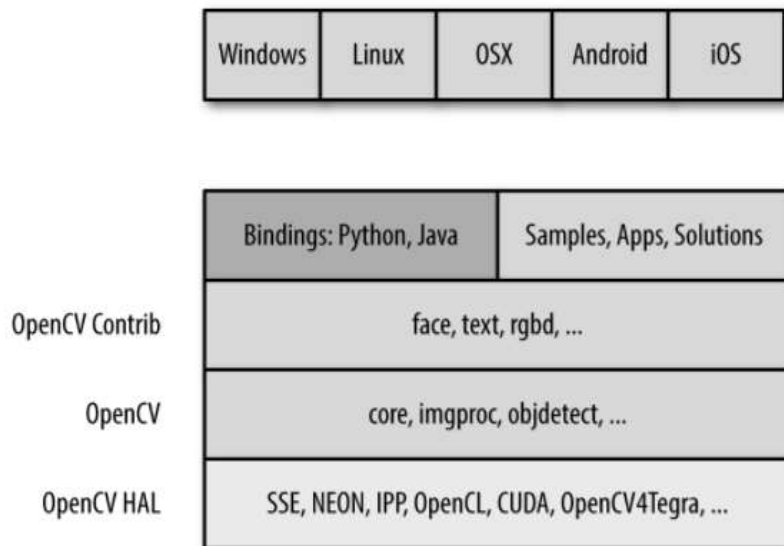


FIGURA 3.2: Diagrama de bloques de OpenCV con sistemas operativos compatibles.

mizaciones de hardware en la capa Hardware Acceleration Layer (HAL) [7].

3.3. Metodología

El desarrollo en espiral es un modelo de desarrollo orientado a riesgos. Las gráficas de actividades de este modelo conforman una espiral, que le da su nombre. Cada iteración representa un conjunto de actividades a realizar. Éstas no son fijadas en base a prioridades, como en otras metodologías, sino en función a un análisis de riesgo [5].

La idea de la espiral surgió a partir de un enfoque distinto a la metodología tradicional cascada, pues tomaba en cuenta el riesgo a la hora de desarrollar software. Así, según el autor [5] es que se define una serie de tareas básicas a realizarse en cada ciclo:

- **Objetivos:** necesidad a cubrir.
- **Análisis de Riesgo:** determinar causas de posibles amenazas y/o eventos no deseables con consecuencias negativas en la iteración.

- **Desarrollar y Verificar:** programar y probar el software.
- **Planificación:** determinar objetivos, alternativas y restricciones para la próxima iteración, de ser necesario.

Si para el final de una iteración, el resultado no es el adecuado, o se necesita realizar mejoras o implementar nuevas funcionalidades, se planifica una nueva iteración para comenzar nuevamente un ciclo de la espiral. La forma de caracola y sus dimensiones indican el avance y aumento del coste del proyecto.



FIGURA 3.3: Tareas para un ciclo en el desarrollo en cascada. Fuente: [23]

La mayor ventaja que presenta este modelo es un análisis de riesgos como parte de la metodología, de forma explícita y clara, no como un proceso alternativo. Esto reduce el riesgo del proyecto e incorpora objetivos de calidad en cada iteración.

Mientras que la desventaja es que genera un modelo difícil de modificar. Se estima mucho tiempo para el desarrollo y requiere experiencia en la fase de identificación de riesgos.

4. Desarrollo del Sistema

A continuación presentaremos el desarrollo realizado necesario para la implementación del diseño del sistema presentado en el capítulo anterior. Se abarcan y utilizan conceptos introducidos en el marco teórico, desde el punto de vista práctico.

4.1. Adquisición

Cuando trabajamos con Procesamiento Digital de Imágenes, el primer paso es realizar el proceso de adquisición, mediante la captura de una fotografía de la colonia de bacterias que deseamos analizar. En este proceso, es muy importante la calidad de la imagen en la fotografía, en términos de condiciones ambientales que presenta la muestra y un fondo de alto contraste. Durante el procesamiento de la imagen se realizan actividades que buscan realzar la nitidez de las características buscadas, así como también reducir ciertas imperfecciones.

Por las condiciones que puedan existir en un laboratorio o el lugar donde se está trabajando con la muestra, pueden surgir ciertos problemas a la hora de realizar la adquisición. Algunos se presentan a continuación:

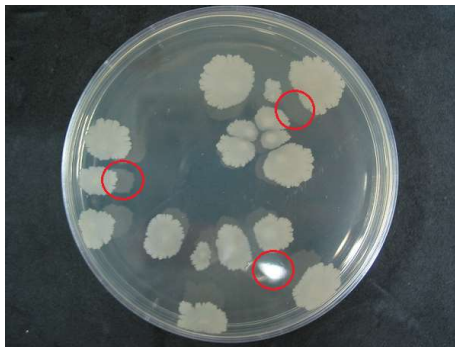


FIGURA 4.1: Cuando empleamos el flash de una cámara sobre la cápsula de Petri podemos obtener reflejos no deseados y sombras de las colonias.

Al momento de realizar el escalado de grises y posterior binarización de la imagen, se ven unas sombras que no son más que el reflejo de las colonias que contenía la cápsula de Petri. La luz del flash, que también se refleja genera una figura que el algoritmo intenta identificar alterando el resultado final.

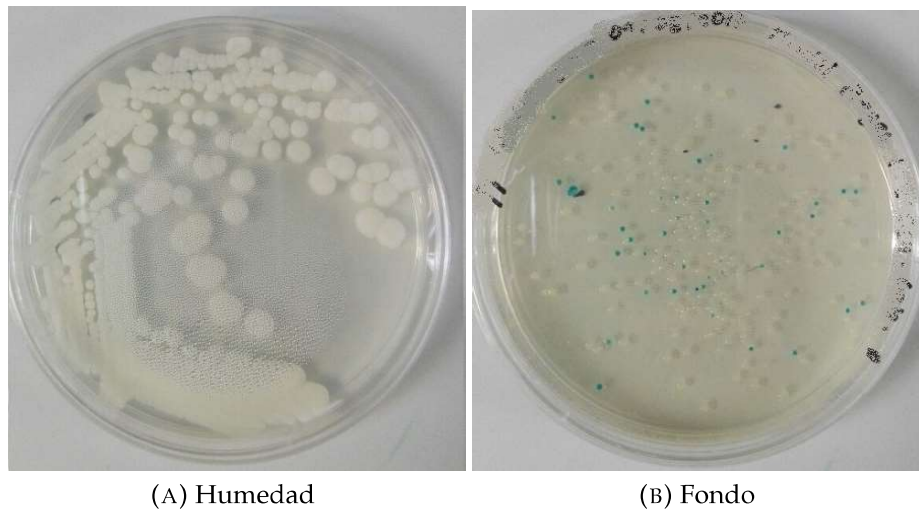


FIGURA 4.2: La humedad genera una distorsión de la imagen dificultando el reconocimiento de colonias. Por otra parte, el fondo de la imagen debe ser preferentemente de un alto contraste en relación a los colonias que se presentan.

La presencia de humedad y la poca claridad presentan el mayor problema a la hora de realizar el reconocimiento de las colonias, ya que entorpecen una correcta segmentación de las características por parte de una computadora.

4.2. Procesamiento

En esta etapa, se muestran las técnicas utilizadas previamente al conteo de las colonias. Para fines de este proyecto, se utilizaron 3 técnicas fundamentales: **Umbralización**, **Reducción de Ruidos** y **Detección de Bordes**, aplicada cada una en el orden descrito.

Es necesario dejar en claro que existen más técnicas de procesamiento de imagen, sin embargo, para esta investigación se utilizaron solamente éstas.

4.2.1. Umbralización

En esta primera fase del procesamiento, antes de umbralizar tomaremos la imagen digitalizada para convertirla a escala de grises utilizando la librería OpenCV, reduciendo la profundidad de los colores a 8 bits. Este proceso previo simplifica

el trabajo de umbralizar la imagen y reduce el número de errores que puedan producirse debido a una profundidad de colores. La binarización es como conocemos formalmente al proceso de umbralización, pues buscamos reducir la profundidad de colores del objetivo a sólo 2 bits de profundidad, descartando características irrelevantes o poco trascendentales.

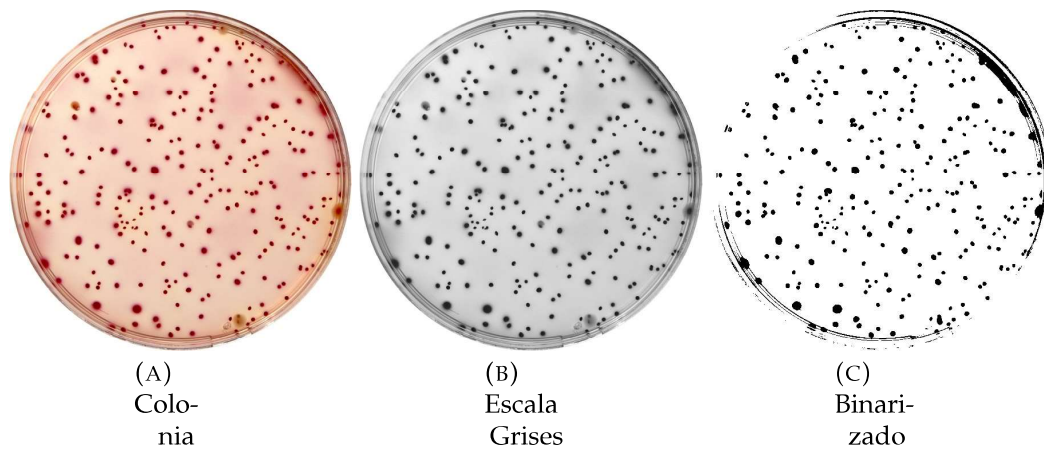


FIGURA 4.3: Se presenta una muestra de una colonia de bacterias, su respectiva transformación a escala de grises y binarizado, sin optimizar.

Es necesario volver a destacar que este paso es esencial para el proceso, pues reduce la cantidad de información presente en la imagen y mejora los resultados de la reducción de ruido y posterior detección de bordes.

Por otra parte, OpenCV también entrega la opción de utilizar un umbral automático optimizado, el cual entrega mejores resultados, como se muestra en la Figura 4.4.

En la Figura 4.4 podemos apreciar la mejora con respecto a la Figura 4.3(C), en la cual se conservan de mejor manera las propiedades de la imagen original. Sin embargo, entre ellas, vemos como la técnica Gaussiana entrega los mejores resultados.

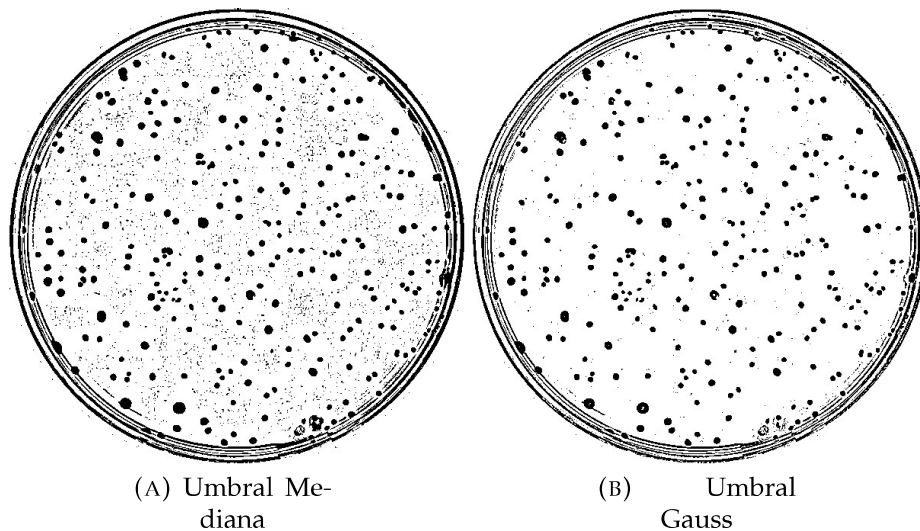


FIGURA 4.4: Se presenta una binarización optimizada para la imagen anterior, mediante una técnica de mediana (izquierda) y mediante una técnica Gaussiana (derecha).

4.2.2. Reducción de Ruido

El siguiente paso para mejorar la imagen es reducir la cantidad de píxeles que interfieren con las características buscadas, deformándolas en ocasiones.

Para realizar la reducción del ruido utilizaremos la técnica de filtro de mediana, cuyo objetivo es armonizar el trazado de la imagen en cuestión. Si lo vemos desde la perspectiva del histograma de la imagen, aplicar el filtro resultaría de la siguiente forma:

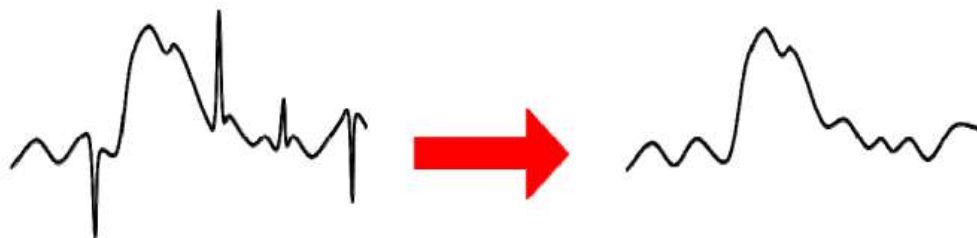


FIGURA 4.5: Armonización de la curva del histograma de una imagen.

El resultado de aplicar esta técnica en la imagen es un “suavizado” de los contornos de las figuras de la misma. Sin embargo, no todos los sectores pueden ser

corregidos completamente, pero podemos deshacernos de gran parte del ruido. Este se aprecia en las Figuras 4.6, 4.7 y 4.8

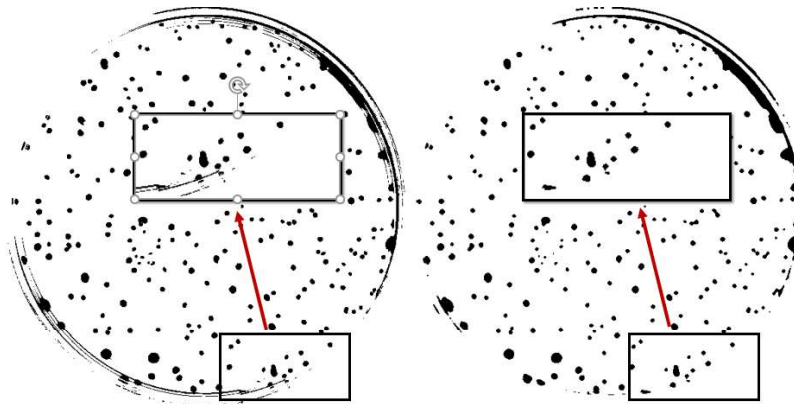


FIGURA 4.6: Podemos apreciar las correcciones que realiza el filtro de mediana en la imagen binarizada.

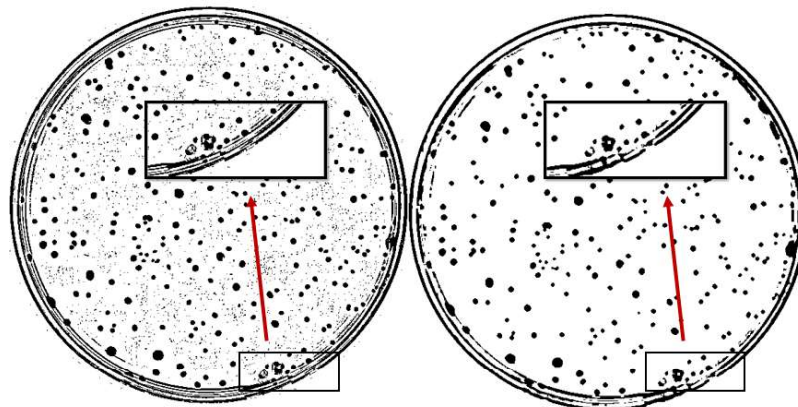


FIGURA 4.7: Reducción de ruido de imagen binarizada con técnica de mediana.

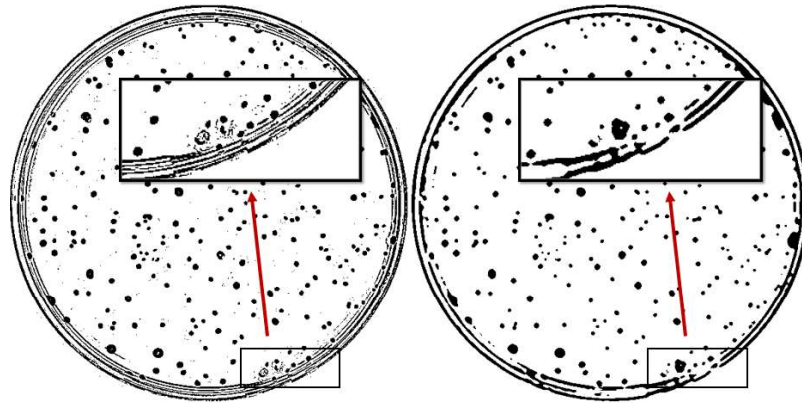


FIGURA 4.8: Reducción de ruido de imagen binarizada con técnica Gaussiana.

En las Figuras 4.6, 4.7 y 4.8, se muestra el resultado de aplicar una corrección mediante un filtro de mediana con un núcleo de 5, es decir, toma como muestra una matriz de 5x5 píxeles (para más información consulte [1]), el cual, en simples palabras, indica las dimensiones del segmento de la matriz con la cual se trabaja.

Antes de continuar, es necesario destacar que existen más técnicas de filtrado, como por ejemplo, el filtro de promedio. Este filtro no fue utilizado para este trabajo debido a que, si bien, realza las características, en este caso las colonias, también genera un aumento en los defectos que la muestra pueda poseer. Como veremos a continuación en la Figura 4.9.

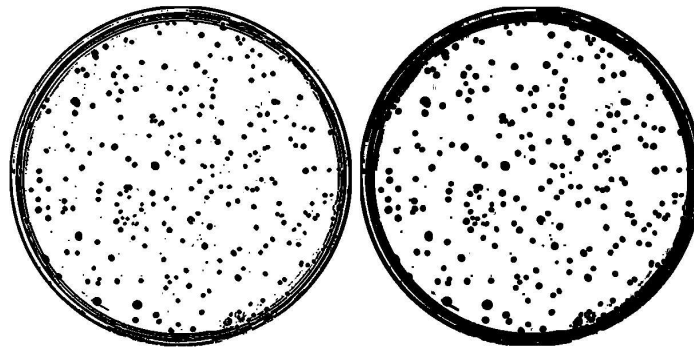


FIGURA 4.9: Presenta imágenes de la colonia una vez utilizado un filtro de promedio con kernel 3 y 5 (matriz de 3x3 y 5x5).

4.2.3. Detección y Realce de Bordes

En este punto del desarrollo, la entrada para este proceso ha sido tratada para presentar las características buscadas en forma realzada. La imagen ha sido binarizada y se ha reducido el ruido que puede poseer.

De acuerdo a lo visto en la Sección 2.7, existen operadores que nos permiten aplicar filtros en una imagen, basados en la función gradiente, resaltando sus bordes. Se introdujeron 4 tipos de operadores, sin embargo, se utiliza el operador Laplaciano, pues se comprobó empíricamente que entrega mejores resultados.

OpenCV trae multiples implementaciones para algoritmos de detección de bordes, lo que nos ahorra tiempo de desarrollo. Comprobamos que el operador de Laplace era mejor que los otros, pues entregaba trazos más finos y detallados.

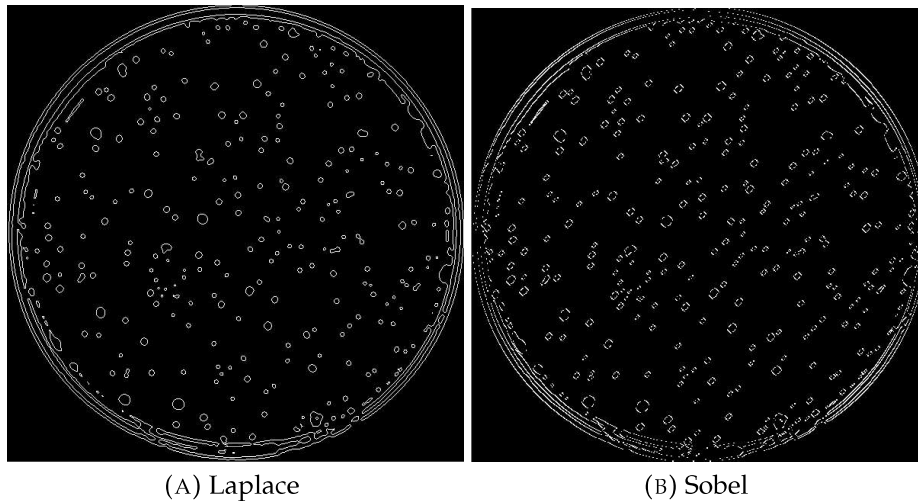


FIGURA 4.10: Detección de bordes mediante operador de Laplace y operador de Sobel.

4.3. Reconocimiento

La colonia con la que comenzamos a trabajar, Figura 4.3(A), a pasado por diversos procesos: escalado de grises, umbralización, filtrado de mediana y finalmente realce de bordes. Estos la han transformado, resaltando sus características principales, las colonias de bacterias.

Anteriormente, en el Sección 2.7, presentamos la Transformada de Hough, algoritmo que es utilizado para realizar el reconocimiento de patrones de las colonias bacterianas presentes en la imagen, mediante la búsqueda de figuras con forma circular. Adicionalmente, el algoritmo guarda un arreglo con los puntos identificados, lo que nos permite más tarde ubicar y resaltar las colonias reconocidas al momento de realizar pruebas y contrastar resultados.

La biblioteca que se utilizó durante todo el desarrollo, OpenCV, posee una implementación básica para utilizar la transformada de Hough.

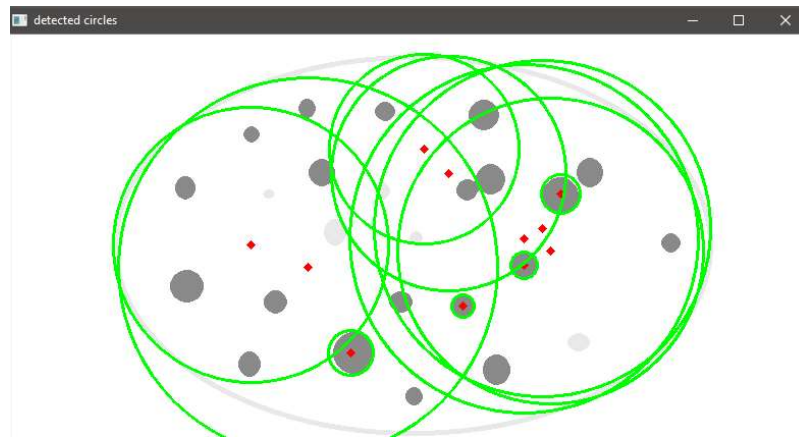


FIGURA 4.11: Ejemplo ejecución algoritmo transformada de Hough.

Como podemos apreciar en la Figura 4.11, hay muchas colonias que se escapan del algoritmo. Esto ocurre debido a que se debe realizar una búsqueda de los valores adecuados para los parámetros del algoritmo. Estos depende de la funcionalidad que se le vaya a dar.

4.4. Búsqueda de Parámetros

Para un correcto funcionamiento del algoritmo anteriormente nombrado, es necesario realizar una búsqueda para determinar los parámetros adecuados para el problema que deseamos resolver.

Para realizar mejoras a la ejecución del algoritmo, primero debemos tener en cuenta los parámetros necesarios para mejorar.

- **IMG** es la imagen sobre la cual se realiza la búsqueda de características.
- **Method** define el método de detección. Actualmente sólo hay uno disponible en OpenCV.
- **DP** es la relación inversa de resolución.
- **MinDist** es la distancia mínima entre centros detectados.
- **Param1** umbral superior para el detector interno Canny Edge.
- **Param2** umbral para la detección del centro
- **MinRadius** radio mínimo a detectar. Cero como predeterminado.
- **MaxRadius** Radio máximo a detectar. Cero como predeterminado.

Los parámetros que debemos buscar son DP, Param1 y Param2. Utilizar umbrales pequeños detecta más círculos, incluyendo falsos positivos, mientras que uno mayor es más selectivo con los mismos.

Para realizar la búsqueda de los parámetros se utilizó una búsqueda de matriz (GridSearch), con los parámetros ya descritos. Fueron necesarias varias búsquedas pues no todas arrojaban buenos resultados. En ocasiones, los parámetros resultaban estar inclinados a reconocer sólo un tipo de colonias, las de la muestra con que se trabajó. Fue muy importante realizar múltiples procesos de búsqueda pues sin ellos, las pruebas hubiesen resultado infructuosas.

4.5. Resumen

En este capítulo observamos las fases del Procesamiento Digital de Imágenes sobre las cuales nos basamos para implementar la solución propuesta.

En primer lugar la Adquisición, donde nos encargamos que conseguir la imagen de la colonia que deseamos trabajar. También se introdujeron características que buscamos evitar a la hora de realizar la adquisición. Como por ejemplo el exceso de luz que produce sombras o reflejos; el exceso de humedad que genera desenfoque y algo similar a una nube; y un fondo con poco contraste con respecto a las colonias a analizar.

El Procesamiento está orientado a explicar los procesos de escalado a grises, umbralización, reducción de ruidos y detección y realce de bordes, mediante los cuales reducimos las características negativas y realzamos las propiedades de las colonias. Se presentaron distintos niveles y métodos de umbralización, siendo el umbral con una técnica Gaussiana el mejor. En el realce de bordes también vimos más de un método, el operador de Laplace resultó ser el mejor.

Para el Reconocimiento vimos que la Transformada de Hough necesita pasar por un proceso de búsqueda de valores para los parámetros del algoritmo, para obtener el mejor resultado posible.

5. Pruebas, Resultados y Discusión

A continuación presentaremos los datos con los cuales se trabajó para realizar las pruebas del algoritmo implementado, el contenido de estos datos, la explicación de por qué fueron escogidos para realizar las pruebas, el proceso de pruebas y el resultado obtenido del conteo del procesamiento de cada muestra.

5.1. Datos

A continuación, en la Tabla 5.1 se presentan el número de colonias de bacterias. Se presentan los datos de pruebas, algunas muestras reales y unas muestras no procesables por el algoritmo (ver Sección 4.1).

TABLA 5.1: Tabla de muestras utilizadas para pruebas

Etiqueta	# Colonias (en cápsula)
Colonia simple	19
Datos de Prueba	21
Muestra 1	18
Muestra 2	26
Muestra 3	38
Muestra 4	82
Muestra 5	261
No procesable 1	17
No procesable 2	181
No procesable 3	293
No procesable 4	No Determinable

5.1.1. Muestras Originales

Colonia simple



FIGURA 5.1: Colonia simple que contiene 19 colonias.

Datos de prueba

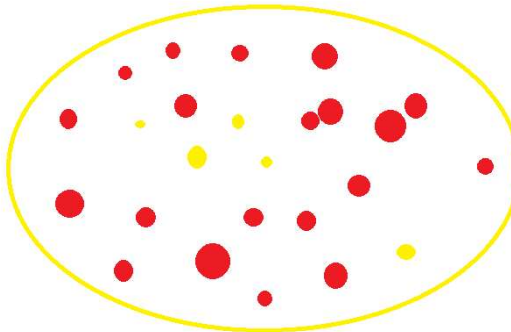


FIGURA 5.2: Imagen de prueba con la cual se realizó la búsqueda de parámetros iniciales, contiene 21 colonias.

Muestra 1

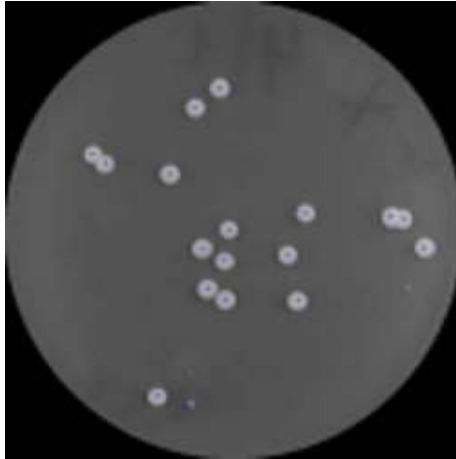


FIGURA 5.3: Muestra de colonia real, contiene 18 colonias.

Muestra 2



FIGURA 5.4: Muestra de colonia real, contiene 26 colonias.

Muestra 3



FIGURA 5.5: Muestra de colonia real, contiene 38 colonias.

Muestra 4



FIGURA 5.6: Muestra de colonia real tomada desde un microscopio electrónico, contiene 82 colonias.

Muestra 5



FIGURA 5.7: Muestra de colonia real, contiene 261 colonias.

Imagen No Procesable 1



FIGURA 5.8: Muestra de colonia real no procesable debido a la presencia de sombras y el flash de una cámara, contiene 17 colonias.

Imagen No Procesable 2



FIGURA 5.9: Imagen no procesable, no es real y presenta elementos distractores como figuras de cámaras fotográficas, líneas rectas y letras, contiene 181 colonias.

Imagen No Procesable 3

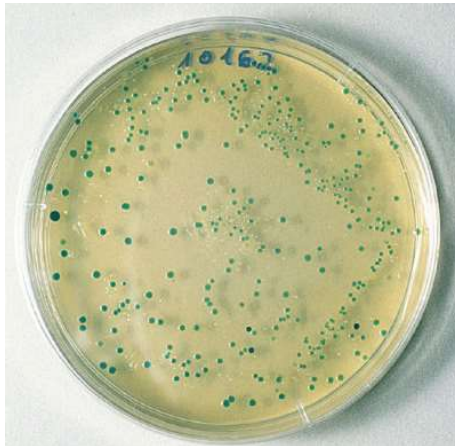


FIGURA 5.10: Muestra de colonia real no procesable debido a la presencia de sombras, contiene 293 colonias.

Imagen No Procesable 4



FIGURA 5.11: Muestra de colonias real no procesable debido a la presencia de excesiva humedad en la tapa de la cápsula de Petri, no es posible determinar el número de colonias debido al estado poco prolijo de la muestra.

5.2. Pruebas

Las pruebas que fueron realizadas, tabuladas y contrastadas con los datos reales. Durante la búsqueda de parámetros revisada en la Sección 4.4, se demostró que obtener exactamente el mismo número de colonias reales era bastante complicado. Por tanto, fue necesario enfocar la búsqueda hacia el número más cercano que se encontrase dentro de los límites establecidos.

El proceso de pruebas fue documentado completamente, el algoritmo fue modificado para que arrojase el estado de cada uno de los procesos por los que pasa la imagen. Así se obtuvo lo siguiente.

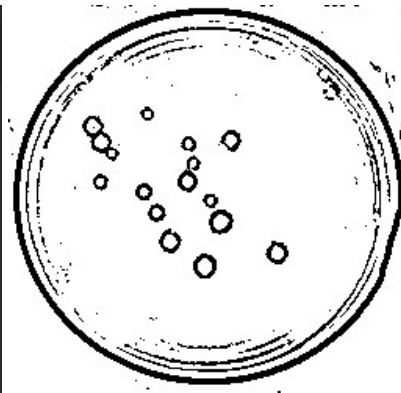
5.2.1. Colonia Simple

Esta cápsula es interesante pues presenta pequeñas colonias que desaparecerán durante el proceso de filtrado de mediana.

Escala de grises y Binarizado

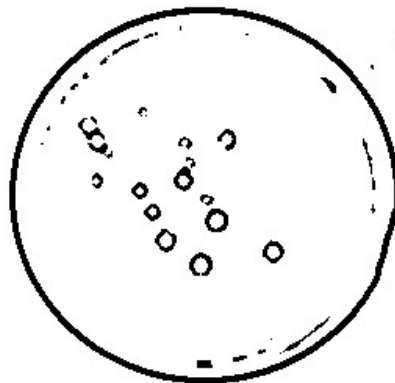


(A) Escala-
do de Gri-
ses



(B) Binari-
zado

Aplicado Filtro de mediana y reconocimiento de Bordes de Laplace



(A) Filtro de
Mediana

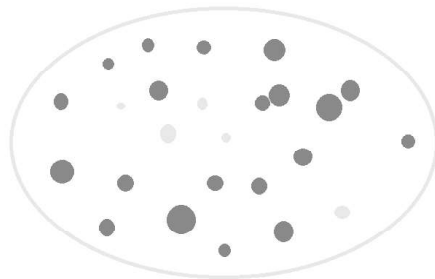


(B) Realce
de bordes
Algoritmo
Laplace

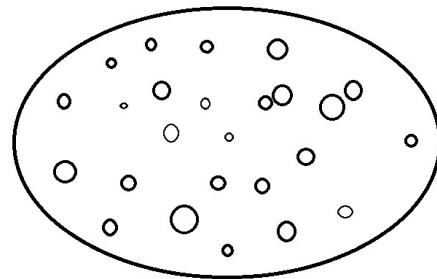
5.2.2. Datos de Prueba

Esta muestra artificial fue utilizada para realizar la búsqueda de parámetros es interesante conocer que ocurre con el proceso completo.

Escala de grises y Binarizado

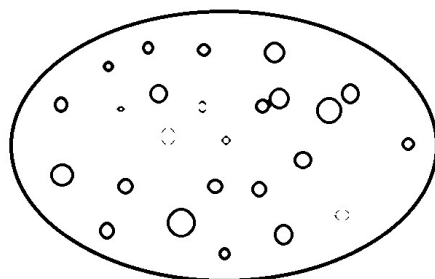


(A) Escalado de Grises

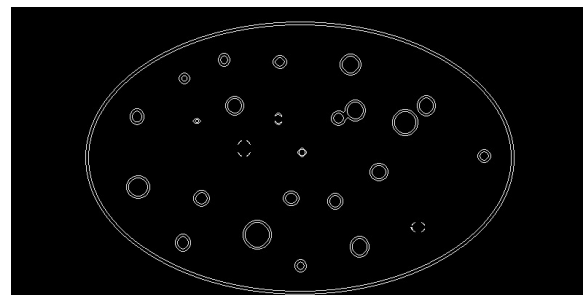


(B) Binarizado

Aplicado Filtro de mediana y reconocimiento de Bordes de Laplace



(A) Filtro de Mediana

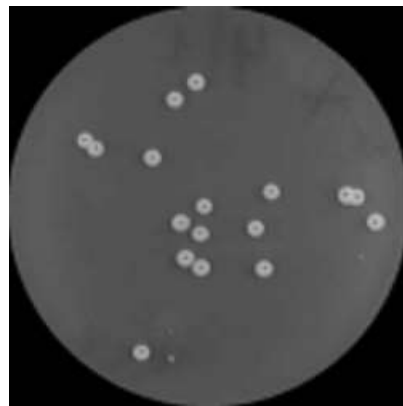


(B) Realce de bordes Algoritmo Laplace

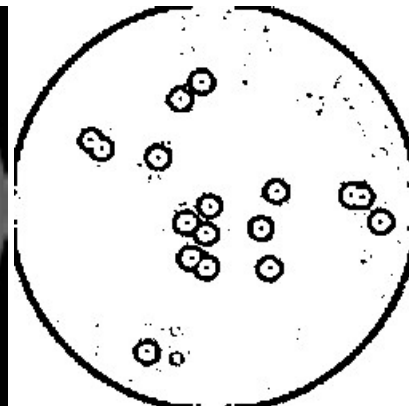
5.2.3. Muestra 1

Esta muestra es interesante pues presenta grupos de colonias juntos, que en el filtrado de mediana se convierten en 1 colonia.

Escala de grises y Binarizado

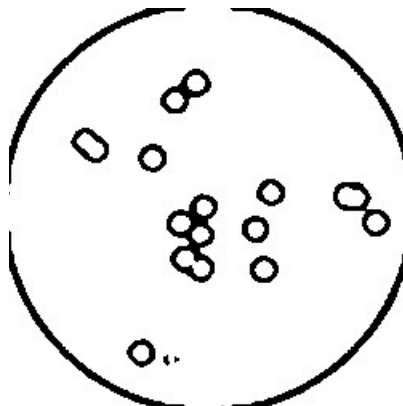


(A) Escala-
do de Gri-
ses

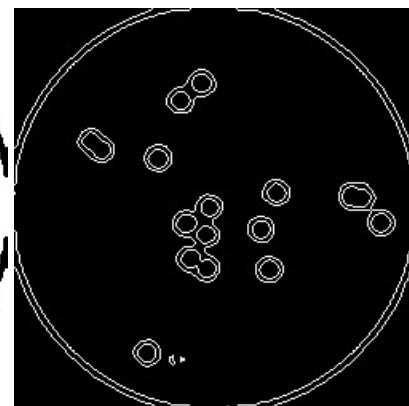


(B) Binari-
zado

Aplicado Filtro de mediana y reconocimiento de Bordes de Laplace



(A) Filtro de
Mediana



(B) Realce
de bordes
Algoritmo
Laplace

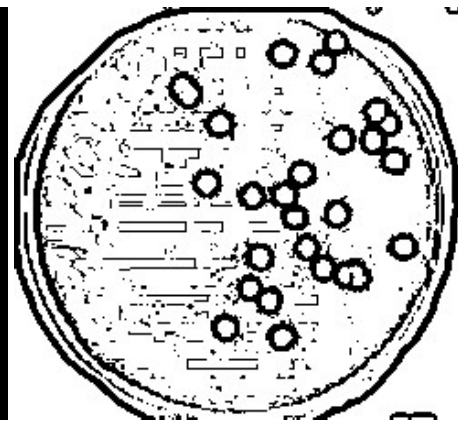
5.2.4. Muestra 2

En esta muestra, además de presentar colonias que están muy juntas, también se aprecia el ruido que se genera por marcar en la cápsula de Petri.

Escala de grises y Binarizado

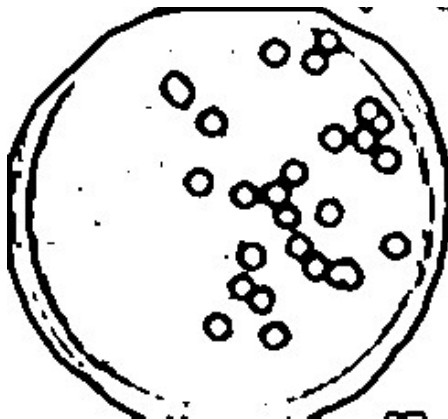


(A) Escalado de Grises

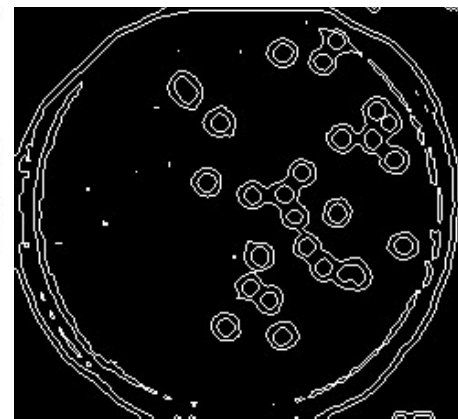


(B) Binarizado

Aplicado Filtro de mediana y reconocimiento de Bordes de Laplace



(A) Filtro de Mediana

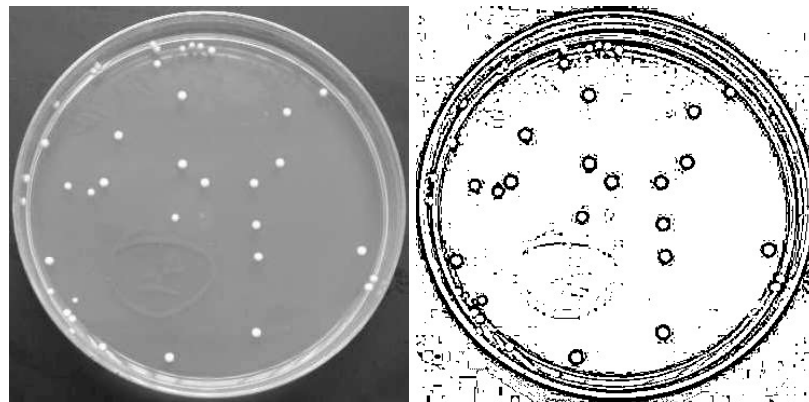


(B) Realce de bordes Algoritmo Laplace

5.2.5. Muestra 3

Esta muestra presenta marcas que fueron realizadas sobre la original, es interesante descubrir que estas marcas desaparecen durante el filtrado de mediana.

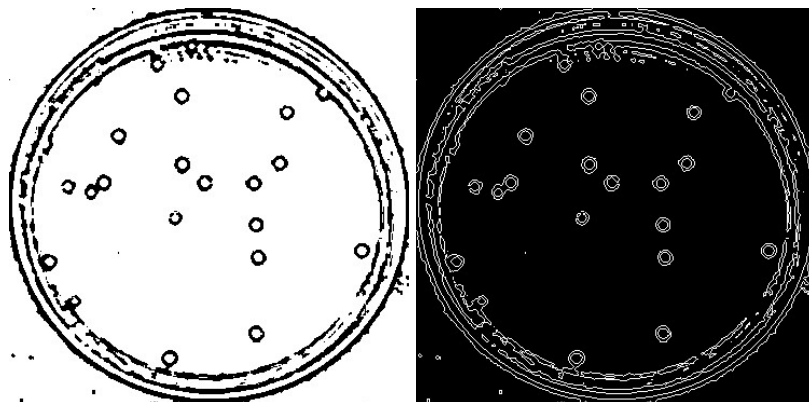
Escala de grises y Binarizado



(A) Escala-
do de Gri-
ses

(B) Binari-
zado

Aplicado Filtro de mediana y reconocimiento de Bordes de Laplace



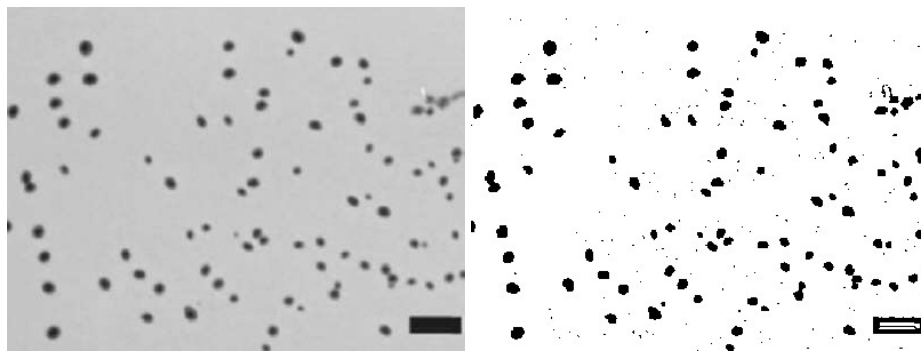
(A) Filtro de
Mediana

(B) Realce
de bordes
Algoritmo
Laplace

5.2.6. Muestra 4

Esta muestra fue tomada directamente desde un microscopio electrónico. Que no halle una cápsula de petri no es impedimento para realizar el conteo.

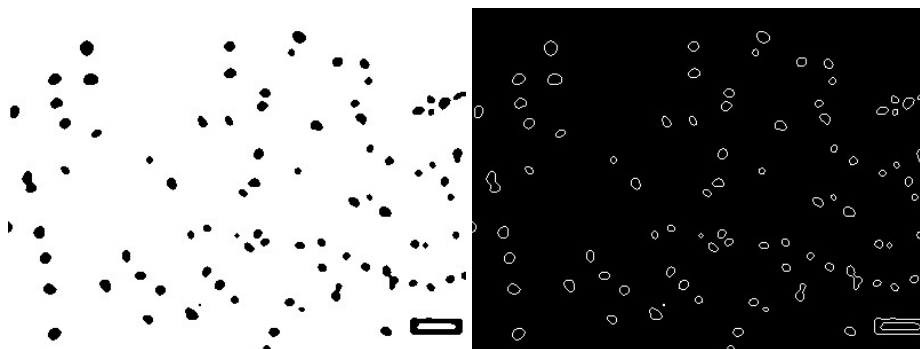
Escala de grises y Binarizado



(A) Escalado de Grises

(B) Binarizado

Aplicado Filtro de mediana y reconocimiento de Bordes de Laplace



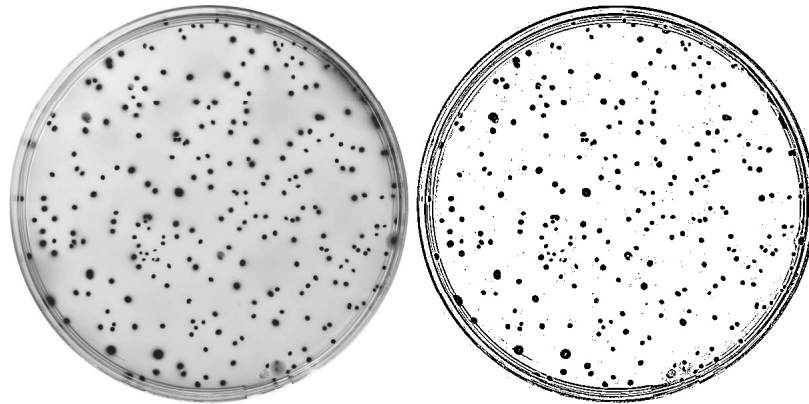
(A) Filtro de mediana

(B) Realce de bordes Algoritmo Laplace

5.2.7. Muestra 5

Muestra real que presenta el mayor número de colonias en esta prueba. Presenta algunas sombras en los bordes.

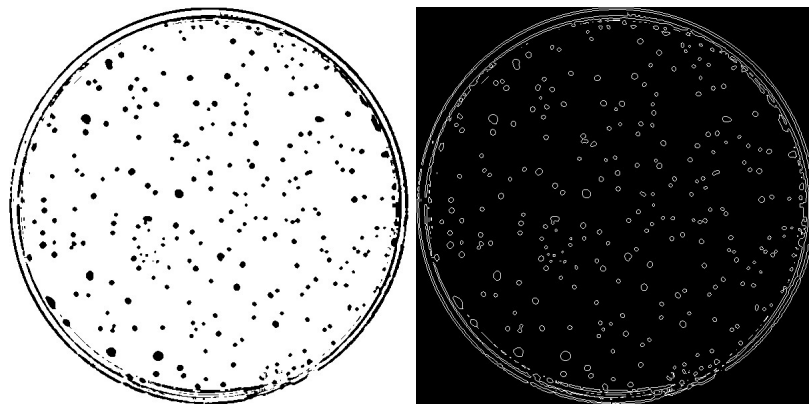
Escala de grises y Binarizado



(A) Escala-
do de Gri-
ses

(B) Binari-
zado

Aplicado Filtro de mediana y reconocimiento de Bordes de Laplace



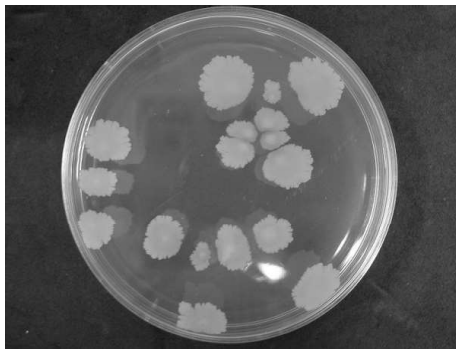
(A) Filtro de
Mediana

(B) Realce
de bordes
Algoritmo
Laplace

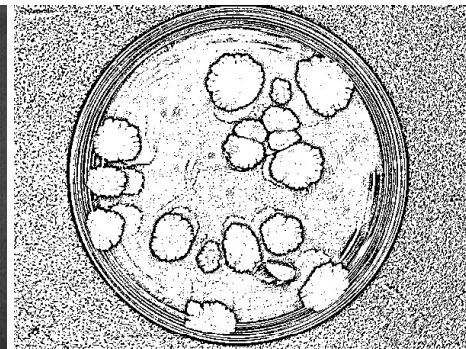
5.2.8. No Procesable 1

Esta muestra presenta gran parte de los problemas presentados como impedimentos en la Sección 4.1; reflejos, sombras y un fondo poco sobrio.

Escala de grises y Binarizado

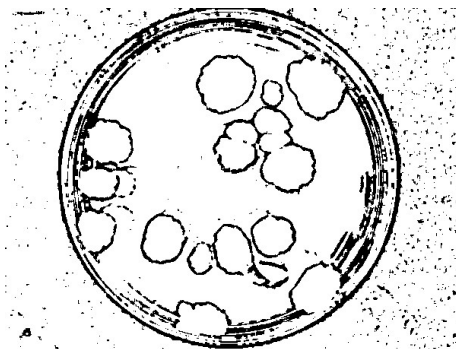


(A) Escalado de Grises

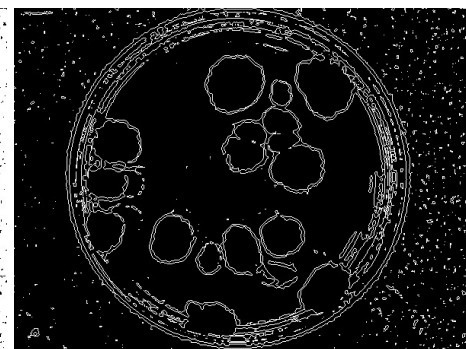


(B) Binarizado

Aplicado Filtro de mediana y reconocimiento de Bordes de Laplace



(A) Filtro de Mediana



(B) Realce de bordes Algoritmo Laplace

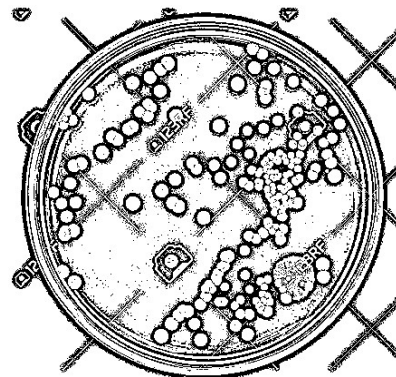
5.2.9. No Procesable 2

Más que una muestra, esta imagen generada presenta numerosas colonias juntas y un fondo con varios elementos distractores.

Escala de grises y Binarizado

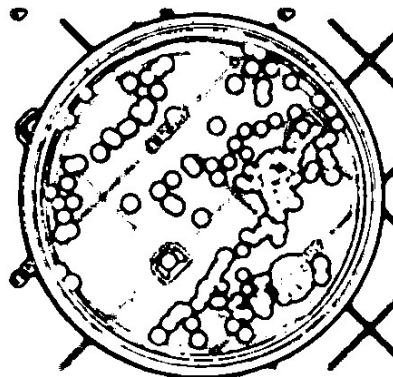


(A) Escala-
do de Gri-
ses

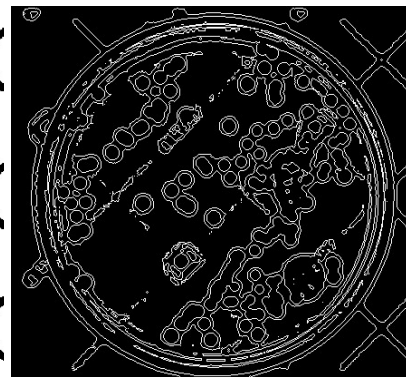


(B) Binari-
zado

Aplicado Filtro de mediana y reconocimiento de Bordes de Laplace



(A) Filtro de
Mediana

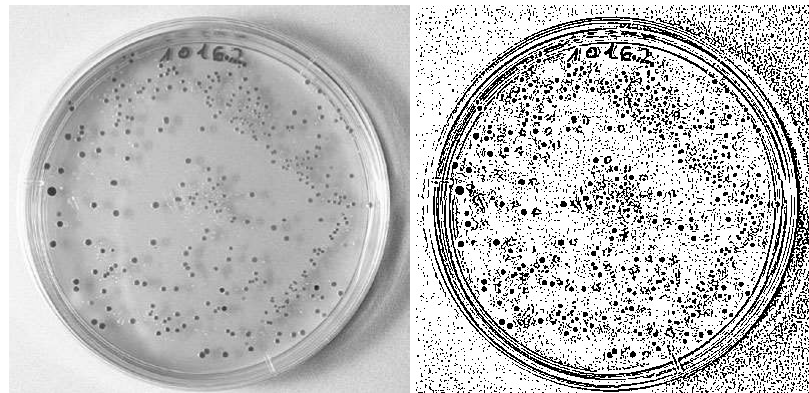


(B) Realce
de bordes
Algoritmo
Laplace

5.2.10. No Procesable 3

En esta muestra en particular, la presencia de sombras y reflejos entorpece en mayor medida la opción de un buen reconocimiento/conteo de colonias.

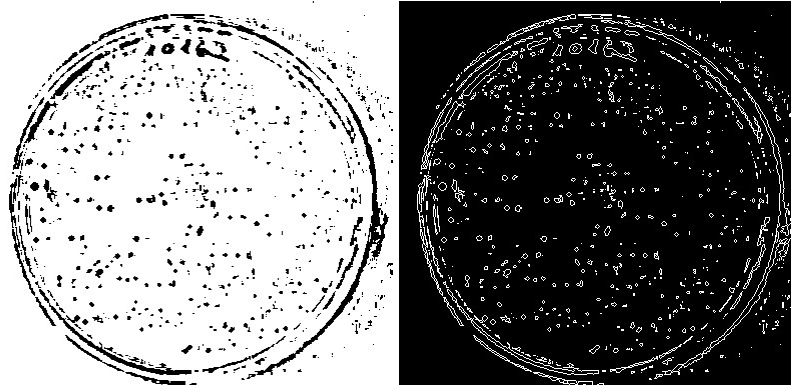
Escala de grises y Binarizado



(A) Escala-
do de Gri-
ses

(B) Binari-
zado

Aplicado Filtro de mediana y reconocimiento de Bordes de Laplace



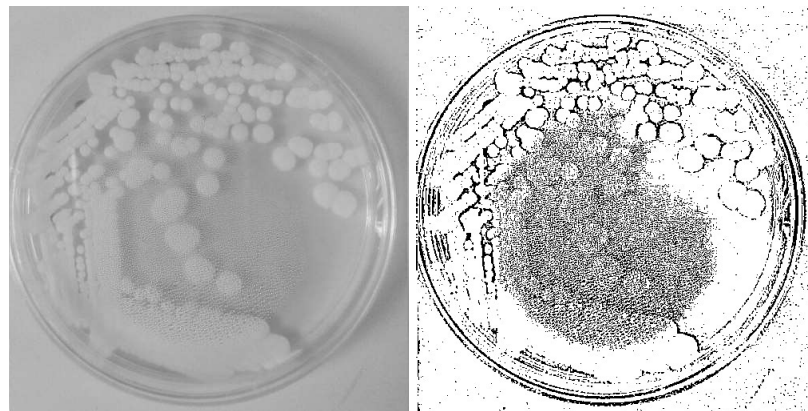
(A) Filtro de
Mediana

(B) Realce
de bordes
Algoritmo
Laplace

5.2.11. No Procesable 4

Esta muestra presenta el que se ha visto, experimentalmente, como el mayor inconveniente cuando se trata de realizar el conteo, la humedad.

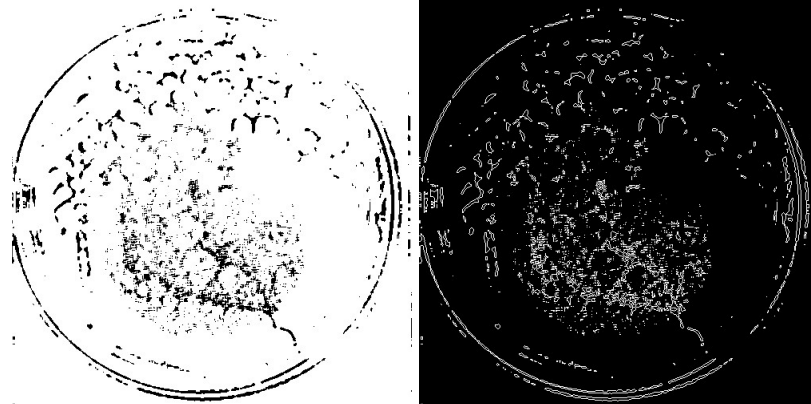
Escala de grises y Binarizado



(A) Escala-
do de Gri-
ses

(B) Binar-
izado

Aplicado Filtro de mediana y reconocimiento de Bordes de Laplace



(A) Filtro de
Mediana

(B) Realce
de bordes
Algoritmo
Laplace

5.3. Resultados

De la sección anterior, podemos observar claramente las ventajas y desventajas de realizar el procesamiento de la imagen antes de realizar el reconocimiento y posterior conteo de colonias.

Por una parte apreciamos como realizar un filtrado de mediana reduce gran cantidad de las imperfecciones, marcas o inscripciones que fueron realizadas sobre las muestras. El proceso de binarizado ayuda a mejorar la nitidez que algunas muestras carecen.

Por otro lado, el procesado produce que algunas colonias se hagan imperceptibles.

El fondo que presentan tiene un papel fundamental para determinar la capacidad de una imagen para ser procesada.

Las sombras generadas por la iluminación no natural o flash de cámara pueden generar un ruido difícil de corregir en la imagen. De igual forma, los filtros generan que algunas colonias más pequeñas desaparezcan al presentar un bajo contraste.

Finalmente la humedad que está presente en algunas colonias es una de las peores características que se pueden presentar. Se produce una especie de nube que hace imposible el reconocimiento.

TABLA 5.2: Tabla de muestras utilizadas para pruebas

Etiqueta	# Colonias	# Algoritmo	Tiempo(ms)	Desviación
Colonia simple	19	21	89	+10.5 %
Datos de Prueba	21	88	353	+319 %
Muestra 1	18	21	65	+16.6 %
Muestra 2	26	30	128	+15.3 %
Muestra 3	38	39	353	+2.6 %
Muestra 4	82	77	137	-7.1 %
Muestra 5	261	223	1600	-14.5 %
No procesable 1	17	114	2836	+570 %
No procesable 2	181	161	1809	-11 %
No procesable 3	293	103	578	-64.8 %
No procesable 4	S/I	160	2927	S/I

5.4. Discusión

En la Tabla 5.2 podemos observar que, en general, para muestras que pueden ser procesadas, el promedio del error porcentual al reconocer y realizar el conteo del número de colonias es del 11.1 %, y la mediana de 12.5 %. En estos casos fue excluido el dato atípico obtenido (Datos de Prueba).

Además, podemos observar que los datos y pruebas de las muestras no procesables reflejan claros problemas en términos de precisión y generalmente tiempo. Donde su desviación estándar se aleja del 15 % que se buscaba en primera instancia, y un tiempo un 50 % mayor, en promedio. En base a las pruebas realizadas, podemos decir que la desviación de la muestra “No procesable 2” es sólo una coincidencia o dato atípico, al igual que presentó el conjunto de datos procesables.

En cuanto a los resultados esperados obtuvimos que:

De las muestras que podían ser analizadas obtuvimos solamente una que se escapaba, anormalmente, de las predicciones iniciales. Cabe destacar que fue la muestra con la cual se realizó la búsqueda de los parámetros necesarios, que es más curioso aún. Lo que nos remonta a analizar el proceso de búsqueda de parámetros.

En aquel punto, para realizar la búsqueda, no se realizó el proceso completo de preprocesado, sólo se realizó un escalado de grises y filtro de mediana, en la forma en que está implementado por defecto. Además, se realizó de esa forma para evitar segmentar el reconocimiento a un tipo de colonias.

Las colonias etiquetadas como “No procesables” respondieron al comportamiento esperado. Se demostró que los problemas introducidos en la Sección 4.1, como la presencia de humedad, sombras, flash de cámaras y un fondo con poco contraste o no adecuado, son una realidad al momento de procesar la imágenes y realizar el reconocimiento y conteo, entorpeciendo el resultado final.

6. Conclusión

En este Capítulo se da por finalizada la investigación realizada. Se presentarán tres puntos fundamentales del proceso como conclusiones. En primer lugar se da respuesta a los objetivos planteados, tanto general como específicos que fueron propuestos al comienzo del documento, seguido de los resultados obtenidos del procesamiento de las muestras. En tercer lugar se exhibe los factores condicionantes del éxito del trabajo y finalmente, el trabajo futuro.

6.1. Conclusión General

Si bien el objetivo principal de la investigación, “reconocer y contabilizar colonias de bacterias” fue logrado, el objetivo general no fue cumplido a cabalidad, debido a que por alcances en los límites de tiempo establecidos no se logró desarrollar la aplicación web que utilizaría el algoritmo de conteo de colonias de bacterias.

El logro obtenido, el reconocimiento de los patrones que presentan las colonias, fue un éxito. Resultó que el error porcentual obtenido en el reconocimiento estaba dentro del rango esperado de $\pm 15\%$ del total de colonias. Algunas muestras presentaron porcentaje de error mayor al esperado, un menor al 2% (que es intrascendente). A pesar de esto, el promedio para imágenes de muestras reales procesables fue de un 11%, con un límite de tiempo menor a los 3 segundos. Esta última medición nos lleva a determinar que podemos cumplir con la limitante del tiempo.

Dentro del marco de las pruebas, se identificó una muestra procesable que generó la mayor sorpresas, al presentar error del 320%. Creemos que esto se debe a que durante el proceso de búsqueda de parámetros, debido a que no se trabajó con una imagen procesada adecuadamente, es decir, no pasó por los procesos descritos en el Capítulo 4, solamente se vio expuesta a un filtro de mediana y escalado de grises. Sin embargo, dado que en las pruebas finales se obtuvieron buenos resultados para el resto de muestras, se estipula que sólo se trata de un dato atípico.

6.2. Resultados Obtenidos

De las muestras obtenidas logramos establecer que el uso de Transformada de Hough para realizar la investigación fue acertado, ya que nos permitió identificar los patrones asociados a las colonias de bacterias, lo cuales son mayoritariamente circulares.

Mediante las pruebas pudimos corroborar y listar los factores que entorpecen el buen funcionamiento del algoritmo. Definimos que factores como **reflejos** producto del uso de flash de una cámara o luz del ambiente; **sombras** producto de la luz natural, artificial o ángulo con el cual se tome la fotografía; **fondos** con poco contraste en relación a las colonias (Figura 4.2[B]) o inapropiados (Figura 5.9); y **humedad** (Figura 5.11) presente en algunas cápsulas de petri, usada en ocasiones para aumentar el índice de crecimiento de determinados cultivos. Estas características son las que buscamos evitar a la hora de realizar el conteo.

En la Tabla 5.2, al fijarnos en los datos que podían ser procesados, apreciamos que los resultados no fueron 100 % exactos. Esto, sin embargo, era algo totalmente esperado. El campo de reconocimiento de patrones es un territorio que necesita de búsquedas de parámetros o entrenamientos exhaustivos (en caso de utilizar redes neuronales) para lograr obtener un resultado preciso al 100 %. Para solventarlo, realizamos la elección del mejor candidato o número más cercano al del objetivo. Por esta razón fue necesario establecer un porcentaje de error para el algoritmo.

Adicionalmente, podemos añadir que el objetivo de cumplir con un tiempo menor a los 30 segundos pasó a ser un factor irrelevante. El tiempo de ejecución del algoritmo no tomó más de 3 segundos (en el peor de los casos) incluidas muestras etiquetadas como no procesables”.

6.3. Puntos Claves

Los puntos claves para el desarrollo de esta investigación estuvieron enfocados principalmente en la elección de la tecnología a utilizar. OpenCV nos proporciona una amplia gama de implementaciones para los distintos algoritmos y

procedimientos que debemos realizar durante el Procesamiento Digital de Imágenes, simplificando el proceso de desarrollo y permitiendo centrarnos trabajar en la búsqueda de valores para los parámetros que optimizaran el conteo.

Del mismo modo, identificar rápidamente los factores que interfieren negativamente en el proceso de segmentación y posterior reconocimiento fue un logro. El no haberlo realizado a tiempo habría modificado los resultados obtenidos, debido a que el objetivo principal no se hubiese alcanzado.

6.4. Trabajo Futuro

Realizar la implementación de la Aplicación Web es fundamental, pues está es la que nos lleva a alcanzar el objetivo de permitir a cualquier persona realizar el recuento de la cantidad de colonias en una muestra. El que no exista una aplicación web que nos facilite el trabajo de realizar el conteo de colonias en un corto período de tiempo, no genera una alternativa frente a las costosas máquinas del mercado.

En términos generales, realizar una mejora a la implementación de la Transformada de Hough puede ser una solución rápida para mejorar el rendimiento y capacidad de reconocimiento del algoritmo, así como utilizar una implementación de la Transformada de Hough para el reconocimiento de elipses, lo que puede dirigir la investigación hacia un nivel algo más profesional y con mejores índices de reconocimiento.

No obstante, este último punto tiene otra directriz, actualizar la versión utilizada de OpenCV, durante la investigación se utilizó la versión 2.4.13. La última entrega liberada es la versión 3.3 y presenta una mayor cantidad de propiedades, como la nombrada anteriormente. A pesar de esto, existen las posibilidades para poder encontrar nuevas formas de realizar un algoritmo óptimo, utilizando una red neuronal, por ejemplo, que realice la selección de parámetros más adecuados dependiendo de la situación en base a trabajos previos.

Finalmente, realizar una aplicación web que permita la interacción con el usuario. Permitirle escoger las mejores configuraciones para el procesamiento de la

imagen antes de realizar el reconocimiento/conteo de las colonias presentes en ella. Mostrarle los resultados paso a paso, aumentar el nivel del control que éste tiene sobre la aplicación.

Bibliografía

- [1] A. Abder-Rahman. Filtrado de imagen. <https://code.tutsplus.com/es/tutorials/image-filtering-in-python--cms-29202>. Accedido: 2017-11-11.
- [2] M. Aliaga. Morfología y estructura de las bacterias. <https://medicinaupv.files.wordpress.com/2011/04/2-3-clase-morfologc3ada-y-estructura-de-laa-bacterias.pdf>.
- [3] F. Bellas Aláez. Curso de procesado digital de imagen. https://www.uv.es/gpoei/eng/Pfc_web/indice.htm.
- [4] R. Blodgett. Most probable number from serial dilutions. <https://www.fda.gov/food/foodscienceresearch/laboratorymethods/ucm109656.htm>.
- [5] B. Boehm. A spiral model of software development and enhancement. *IEEE Computer*, 21(5):61–72, 1988. Accedido: 2017-11-02.
- [6] G. Bou, A. Fernández-Olmos, C. García, J. Sáez-Nieto, and S. Valdezate. Enfermedades infecciosas y microbiología clínica. *Elsevier*, 29(8):601–608, 2011.
- [7] G. Brandski and A. Kaehler. Learning opencv 3. <https://www.safaribooksonline.com/library/view/learning-opencv-3/9781491937983/ch01.html>. Accedido: 2017-10-31.
- [8] Facultad de Química UNAM. Medición del crecimiento microbiano. http://depa.fquim.unam.mx/amyd/archivero/U4b_MedicionCrecimiento_19837.pdf.
- [9] Universidad Nacional del Nordeste. Trabajo practico nº 6 : Identificación. <http://www.biologia.edu.ar/microgeneral/tp6.pdf>.

- [10] Learn DIP. Grayscale to rgb conversion. https://www.tutorialspoint.com/dip/grayscale_to_rgb_conversion.htm. Accedido: 2018-03-26.
- [11] K. Ettwig, M. Butler, D. Le Paslier, E. Pelletier, S. Mangenot, M. Kuypers, F. Schreiber, B. Dutilh, J. Zedelius, D. de Beer, Jolein Gloerich, Hans J. C. T. Wessels, Theo van Alen, Francisca Luesken, Ming L. Wu, K. van de Pas-Schoonen, H. Op den Camp, E. Janssen-Megens, K. Francoijs, H. Stunnenberg, J. Weissenbach, M. Jetten, and M. Strous. Nitrite-driven anaerobic methane oxidation by oxygenic bacteria. *Nature*, 464:543–548, 2010.
- [12] US Food and Drug Administration (FDA). Statistics applied to microbiological analysis. <https://www.fda.gov/scienceresearch/fieldscience/laboratorymanual/ucm174294.htm>.
- [13] R. Gonzalez and R. Woods. *Digital Image Processing*. Pearson Education, Inc, Upper Saddle River, New Jersey, 2008.
- [14] J. Molina López and T. Uribarren Berrueta. Generalidades de bacterias. <http://www.facmed.unam.mx/deptos/microbiologia/bacteriologia/generalidades.html>.
- [15] L. Maturin and J. Peeler. Bacteriological analytical manual, chapter 3, aerobic plate count. <https://www.fda.gov/food/foodscienceresearch/laboratorymethods/ucm063346.htm>.
- [16] S. Nagpal. An automated approach for bacterial colony counter. *Int.J.Computer Technology Applications*, 3(2):724–728.
- [17] J. Pearce. Pipeline architecture. <http://www.cs.sjsu.edu/~pearce/modules/patterns/distArch/pipeline.htm>. Accedido: 2017-11-26.
- [18] S. Sahan, A. Degirmenci, and I. Çankaya. A study on clustering based image thresholding techniques with matlab gui. *IJESC*, 6(8):2152–2160, 2016. Accedido: 2017-10-31.
- [19] D. Salcedo. Identificación de árboles mediante la fotografía de una de sus hojas obtenida con un smartphone android. <http://dspace.usalca.cl/handle/1950/10472>. Accedido: 2017-10-31.

- [20] J. Schier and B. Kovác. Automated counting of yeast colonies using the fast radial transform algorithm. <https://www.utia.cas.cz/files/soutez11/schier.pdf>. Accedido: 2017-11-16.
- [21] SciQuip. Automatic colony counters. <https://www.sciquip.co.uk/automatic-colony-counters.html>. Accedido: 2017-11-16.
- [22] OpenCV team. About. <https://opencv.org/about.html>. Accedido: 2017-10-31.
- [23] Wikipedia. Desarrollo en espiral. https://es.wikipedia.org/wiki/Desarrollo_en_espiral. Accedido: 2017-10-31.
- [24] Intel Developers Zone. Color models. <https://software.intel.com/en-us/ipp-dev-reference-color-models>. Accedido: 2017-10-11.