



# FACULTAD DE INGENIERÍA

Carrera de Ingeniería de Sistemas Computacionales

“DESARROLLO DE UN SISTEMA INTELIGENTE  
BASADO EN VISIÓN COMPUTACIONAL PARA  
DETECTAR BACTERIAS ESCHERICHIA COLI EN  
VERDURAS FRESCAS”

Tesis para optar el título profesional de:

Ingeniero de Sistemas Computacionales

Autor:

Jorge Jonathan Jesús Rodríguez Alvarez

Asesor:

Ing. Juan Orlando Salazar Campos

Trujillo - Perú

2019

## **DEDICATORIA**

A Dios, por permitirme llegar a este momento tan importante de mi formación profesional.

A mi Madre, Sra. Sara Alvarez Santa Cruz, quien ha luchado por hacerme una persona de bien, que con paciencia y amor me ha podido impulsar para seguir adelante y siempre seguir por el camino correcto.

A mi Pareja y mi Hijo, por el ser el pilar importante y por demostrarme su apoyo incondicional para poder alcanzar esta meta.

## **AGRADECIMIENTO**

En primer lugar, deseo expresar mi agradecimiento a mi asesor Ing. Juan Orlando Salazar Campos, por la dedicación y apoyo que ha brindado a este trabajo, por el respeto a mis sugerencias e ideas y por la dirección y el rigor que ha facilitado a las mismas.

Asimismo, agradezco a mi Madre y mis hermanas por apoyarme en todo momento, y por haberme dado la fortaleza para seguir adelante en aquellos momentos de debilidad.

Gracias a mi pareja, por su apoyo mutuo brindado para lograr este objetivo.

Sin el apoyo de todos, este trabajo nunca se habría escrito, por eso, a todos los mencionados mil gracias.

## Tabla de contenidos

<b>DEDICATORIA .....</b>	<b>2</b>
<b>AGRADECIMIENTO.....</b>	<b>3</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>5</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>6</b>
<b>ABSTRACT .....</b>	<b>10</b>
<b>CAPÍTULO I. INTRODUCCIÓN.....</b>	<b>12</b>
<b>CAPÍTULO II. METODOLOGÍA.....</b>	<b>86</b>
<b>CAPÍTULO III. RESULTADOS.....</b>	<b>90</b>
<b>CAPÍTULO IV. DISCUSIÓN Y CONCLUSIONES .....</b>	<b>93</b>
<b>REFERENCIAS .....</b>	<b>95</b>
<b>ANEXOS.....</b>	<b>98</b>

## ÍNDICE DE TABLAS

<b>Tabla 1</b> Formato BMP .....	29
<b>Tabla 2</b> Formato TIFF .....	30
<b>Tabla 3</b> Formato GIF.....	31
<b>Tabla 4</b> Formato JPG o JPEG .....	31
<b>Tabla 5</b> Formato PNG .....	32
<b>Tabla 6</b> Tamaño de una imagen .....	35
<b>Tabla 7</b> Detección remota .....	41
<b>Tabla 8</b> Niveles de visión y procesos del PDI.....	50
<b>Tabla 9</b> E. coli enterotoxigénico (ETEC).....	78
<b>Tabla 10</b> E. coli Enteropatógena (EPEC).....	79
<b>Tabla 11</b> E. coli enterohemorrágico (EHEC).....	79
<b>Tabla 12</b> E. Coli enteroagregativa (EAEC).....	80
<b>Tabla 13</b> Tabla de Técnicas, Instrumento y procedimientos de recolección de datos. ....	87
<b>Tabla 14</b> Procedimiento de Análisis de Datos .....	89
<b>Tabla 15</b> Matriz de Confusión .....	90
<b>Tabla 16</b> Matriz de Confusión con datos .....	91

## ÍNDICE DE FIGURAS

<b>Figura 1:</b> Aumento de contraste: (a) imagen oscura debido a que su rango de grises es reducido, (b) ecualización del rango de grises.....	24
<b>Figura 2:</b> Reconocimiento de caracteres en base a su codificación radial .....	25
<b>Figura 3:</b> Tipos de Imágenes Digitales; (a) RGB; (b) Indexada; (c) Escala de Grises; (d) Binaria .....	34
<b>Figura 4:</b> Dispositivos de captura; (a) cámara digital; (b) escáner; (c) video cámara digital .....	35
<b>Figura 5:</b> Imagen con baja resolución .....	36
<b>Figura 6:</b> Imagen con Alta resolución .....	36
<b>Figura 7:</b> Tomografía por emisión de positrones (PET) .....	38
<b>Figura 8:</b> Angiograma de la aorta.....	39
<b>Figura 9:</b> Tomografía de la cabeza .....	39
<b>Figura 10:</b> Ejemplos de imágenes ultravioleta: a) Maíz sano. B) Maíz Enfermo (smut).....	40
<b>Figura 11:</b> Bandas del satélite LANDSAT.....	41
<b>Figura 12:</b> Control de calidad en circuito integrado, empaque de píldoras, nivel de líquidos, burbujas en plástico transparente, cereal, y fallas en una lente intraocular. ....	42
<b>Figura 13:</b> Imágenes MRI de una rodilla y una espina.....	43
<b>Figura 14:</b> Vecindad $N_4(p)$ .....	46
<b>Figura 15:</b> Vecindad $ND(p)$ .....	46
<b>Figura 16:</b> Conectividad de píxeles .....	47
<b>Figura 17:</b> Distancia euclidiana para una imagen de 5 por 5.....	48
<b>Figura 18:</b> Distancia tablero de ajedrez para una imagen de 5 por 5.....	48
<b>Figura 19:</b> Diferentes ruidos afectando a una imagen; (a) Original; (b) Gaussiano; (c) Multiplicativo; (d) Sal y Pimienta. ....	49
<b>Figura 20:</b> Microscopio Olympus CX21 .....	51
<b>Figura 21:</b> Imágenes con distintos niveles de brillo con sus histogramas. ....	52
<b>Figura 22:</b> Ecualizado del histograma de una imagen; (a) imagen original con su respectivo histograma; (b) ecualizado del histograma. ....	53
<b>Figura 23:</b> Realce de bordes en Imágenes; (a) Imágenes originales; (b) Imagen con bordes realzados. ....	54
<b>Figura 24:</b> Detección de contornos en Imágenes; (a) Imagen original; (b) Detección con filtro de Sobel; (c) Detección con filtro de Canny. ....	54
<b>Figura 25:</b> Apertura de una imagen binaria; (a) Imagen Original; (b) Imagen Aperturada.....	55
<b>Figura 26:</b> Dilatación de una imagen binaria; (a) Imagen Original; (b) Imagen Dilatada.....	55
<b>Figura 27:</b> Erosión de una imagen binaria; (a) Imagen Original; (b) Imagen Erosionada. ....	56
<b>Figura 28:</b> Cierre de una imagen binaria; (a) Imagen Original; (b) Imagen Cerrada. ....	56
<b>Figura 29:</b> Binarización.....	58
<b>Figura 30:</b> Segmentación por umbralización de la placa de un automóvil (a) imagen original; (b) Segmentación con umbral de 128. ....	59
<b>Figura 31:</b> Extracción del primer carácter de la placa. ....	62
<b>Figura 32:</b> Espacio paramétrico con dos parámetros y tres clases. Cada círculo engloba objetos del mismo tipo. ....	63
<b>Figura 33:</b> Ejemplo de optimización paramétrica. El modelo de la elipse (izquierda) se ajusta a las características -orillas- obtenidas en la imagen (derecha). ....	64
<b>Figura 34:</b> Red Neuronal .....	66
<b>Figura 35:</b> Esquema básico de una red neuronal convolucional .....	69
<b>Figura 36:</b> Operación Básica de Convolución .....	71
<b>Figura 37:</b> Capa Pooling o Capa de Reducción .....	71
<b>Figura 38:</b> Características del Escherichia Coli. ....	75
<b>Figura 39:</b> Esquema de la bacteria Escherichia Coli.....	76
<b>Figura 40:</b> Características Morfológicas y tintoriales. ....	76
<b>Figura 41:</b> Características Nutricionales.....	77
<b>Figura 42:</b> Grupos E. Coli.....	77
<b>Figura 43:</b> Colocación de Bacteria .....	81
<b>Figura 44:</b> Violeta en Bacteria E. Coli .....	81

<b>Figura 45:</b> Visualización de Bacteria.....	82
<b>Figura 46:</b> Esquema de Captura.....	99
<b>Figura 47:</b> Diagrama de Flujo del Pre Procesamiento para una sola Imagen .....	99
<b>Figura 48:</b> Diagrama de Flujo del Pre Procesamiento Para varias imágenes.....	100
<b>Figura 49:</b> Diseño de la red neuronal convolucional .....	101
<b>Figura 50:</b> Imagen de 150 x 150 .....	101
<b>Figura 51:</b> Imagen Reescalada a 1. / 255.....	102
<b>Figura 52:</b> Pre Procesamiento de Imágenes de Entrenamiento .....	103
<b>Figura 53:</b> Pre Procesamiento de Imágenes de Validación .....	103
<b>Figura 54:</b> Data de entrenamiento y validación .....	103
<b>Figura 55:</b> Capa Convolucional .....	104
<b>Figura 56:</b> Operación MaxPooling .....	104
<b>Figura 57:</b> Primera Capa Convolucional .....	105
<b>Figura 58:</b> Segunda Capa Convolucional .....	105
<b>Figura 59:</b> Ultimas Capas de una Red Neuronal Convolucional.....	106
<b>Figura 60:</b> Parámetros para Optimizar el Algoritmo.....	106
<b>Figura 61:</b> Entrenamiento Red Neuronal.....	107
<b>Figura 62:</b> Generación de Pesos y Modelos del entrenamiento .....	107
<b>Figura 63:</b> Parámetros de Predicción .....	108
<b>Figura 64:</b> Función de Predicción de las clases .....	108
<b>Figura 65:</b> Prueba de Detección de E. Coli Img. 01 .....	109
<b>Figura 66:</b> Prueba de Detección Negativa Img. 27.....	110
<b>Figura 67:</b> Prueba de Detección de E. Coli Img. 01 .....	110
<b>Figura 68:</b> Prueba de Detección de E. Coli Img. 02 .....	111
<b>Figura 69:</b> Prueba de Detección de E. Coli Img. 03 .....	111
<b>Figura 70:</b> Prueba de Detección de E. Coli Img. 04 .....	112
<b>Figura 71:</b> Prueba de Detección de E. Coli Img. 05 .....	112
<b>Figura 72:</b> Prueba de Detección de E. Coli Img. 06 .....	113
<b>Figura 73:</b> Prueba de Detección de E. Coli Img. 07 .....	113
<b>Figura 74:</b> Prueba de Detección de E. Coli Img. 08 .....	114
<b>Figura 75:</b> Prueba de Detección de E. Coli Img. 09 .....	114
<b>Figura 76:</b> Prueba de Detección de E. Coli Img. 10 .....	115
<b>Figura 77:</b> Prueba de Detección de E. Coli Img. 11 .....	115
<b>Figura 78:</b> Prueba de Detección de E. Coli Img. 12 .....	116
<b>Figura 79:</b> Prueba de Detección de E. Coli Img. 13 .....	116
<b>Figura 80:</b> Prueba de Detección de E. Coli Img. 14 .....	117
<b>Figura 81:</b> Prueba de Detección de E. Coli Img. 15 .....	117
<b>Figura 82:</b> Prueba de Detección de E. Coli Img. 16 .....	118
<b>Figura 83:</b> Prueba de Detección de E. Coli Img. 17 .....	118
<b>Figura 84:</b> Prueba de Detección de E. Coli Img. 18 .....	119
<b>Figura 85:</b> Prueba de Detección de E. Coli Img. 19 .....	119
<b>Figura 86:</b> Prueba de Detección de E. Coli Img. 20 .....	120
<b>Figura 87:</b> Prueba de Detección de E. Coli Img. 21 .....	120
<b>Figura 88:</b> Prueba de Detección de E. Coli Img. 22 .....	121
<b>Figura 89:</b> Prueba de Detección de E. Coli Img. 23 .....	121
<b>Figura 90:</b> Prueba de Detección de E. Coli Img. 24 .....	122
<b>Figura 91:</b> Prueba de Detección de E. Coli Img. 25 .....	122
<b>Figura 92:</b> Prueba de Detección de E. Coli Img. 26 .....	123
<b>Figura 93:</b> Prueba de Detección Negativa Img. 27.....	123
<b>Figura 94:</b> Prueba de Detección Negativa Img. 28.....	124
<b>Figura 95:</b> Prueba de Detección Negativa Img. 29.....	124
<b>Figura 96:</b> Prueba de Detección Negativa Img. 30.....	125
<b>Figura 97:</b> Prueba de Detección Negativa Img. 31.....	125

## RESUMEN

La resistencia a los antimicrobianos se está acentuando en muchos agentes infecciosos, pero se centra en la resistencia a los antibióticos en siete bacterias responsables de infecciones comunes graves, como la septicemia, la diarrea, la neumonía, las infecciones urinarias o la gonorrea. Los datos son muy preocupantes y demuestran la existencia de resistencia a los antibióticos, especialmente a los utilizados como último recurso, en todas las regiones del mundo. Entre los principales hallazgos destaca la bacteria llamada Escherichia Coli, ya que ésta ha dejado de ser una previsión para el futuro y está en todas las regiones del mundo, una enfermedad real que puede afectar a cualquier persona, por lo cual, la organización mundial de la salud expresó que para prevenir la infección hay que aplicar medidas de control en todas las etapas de la cadena alimentaria.

Esta problemática nos conllevó evaluar el entorno internacional y nacional verificando que en el entorno internacional existe tecnología como el llamado “Riboprinter” que permite realizar el reconocimiento de bacterias, a nivel nacional no existe tecnología similar que apoye la identificación de bacterias Escherichia Coli, esto nos brindó la posibilidad de explorar alternativas basadas en visión computacional.

La presente tesis planteó como objetivo principal la creación de un sistema inteligente basado en visión computacional para detectar bacterias Escherichia Coli en verduras frescas. Para lo cual, este desarrollo se basó en las etapas del ciclo de vida en cascada, donde se diseñó una solución siguiendo técnicas de aprendizaje supervisado con redes neuronales convolucionales y técnicas de pre procesamiento de imágenes, lo que permitió obtener un mejor resultado.

Después de haber desarrollado el sistema inteligente basado en visión computacional, los resultados obtenidos fueron una sensibilidad de 100%, una eficacia de 100% y una especificidad de 100%.

Analizando los resultados obtenidos se puede demostrar la hipótesis en la que se manifiesta que un sistema inteligente basado en de visión computacional detecta bacterias Escherichia Coli en Verduras Frescas.

**Palabras clave:** visión computacional, detectar bacterias Escherichia Coli.

## ABSTRACT

Antimicrobial resistance is accentuated in many infectious agents, but it focuses on antibiotic resistance in seven bacteria responsible for common serious infections, such as septicemia, diarrhea, pneumonia, urinary tract infections, or gonorrhea. The data are very worrying and demonstrate the existence of resistance to antibiotics, especially those used as a last resort, in all regions of the world. Among the main findings is the bacterium called Escherichia Coli, since it has ceased to be a forecast for the future and is in all regions of the world, a real disease that can affect anyone, therefore, the global organization of health expressed that to prevent infection, control measures must be applied at all stages of the food chain.

This problem led us to evaluate the international and national environment verifying that in the international environment there is technology such as the so-called "Riboprinter" that allows the recognition of bacteria, at the national level there is no similar technology that supports the identification of Escherichia Coli bacteria. It offered the possibility of exploring alternatives based on computational vision.

This thesis proposed as the main objective the creation of an intelligent system based on computational vision to detect Escherichia Coli bacterium in fresh vegetables. For which, this development was based on the stages of the cascade life cycle, where a solution was designed following supervised learning techniques with convolutional neural networks and image pre-processing techniques, which allowed obtaining a better result.

After having developed the intelligent system based on computational vision, the results obtained were a sensitivity of 100%, an efficiency of 100% and a specificity of 100%.

Analyzing the results obtained, we can demonstrate the hypothesis that an intelligent system based on computational vision detects Escherichia Coli bacteria in Fresh Vegetables.

**Keywords:** computational vision, detect bacteria Escherichia Coli.

## CAPÍTULO I. INTRODUCCIÓN

### 1.1. Realidad problemática

El informe, titulado Antimicrobial resistance: global report on surveillance, señala que la resistencia a los antimicrobianos se está acentuando en muchos agentes infecciosos (bacterias, parásitos, hongos, virus), pero se centra en la resistencia a los antibióticos en siete bacterias responsables de infecciones comunes graves, como la septicemia, la diarrea, la neumonía, las infecciones urinarias o la gonorrea. Los datos son muy preocupantes y demuestran la existencia de resistencia a los antibióticos, especialmente a los utilizados como último recurso, en todas las regiones del mundo. (Organización Mundial de la Salud, 2014). Entre los principales hallazgos del informe destaca la bacteria llamada Escherichia Coli. En un informe de la organización mundial de la salud, revela que la bacteria E. Coli ha dejado de ser una previsión para el futuro y es ya en todas las regiones del mundo una enfermedad real que puede afectar a cualquier persona de cualquier edad en cualquier país. (Organización Mundial de la Salud, 2014) Por lo tanto "la Organización Mundial de la Salud expresa que, para prevenir la infección, hay que aplicar medidas de control en todas las etapas de la cadena alimentaria, desde la producción agropecuaria en la granja hasta la elaboración, fabricación y preparación de los alimentos en las cocinas de establecimientos comerciales y hogares." (Marimón, 2014) A nivel mundial en tecnologías de reconocimiento de bacterias existen aparatos como el "Riboprinter" creada por DuPont empresa estadounidense dedicada a varias ramas industriales de la química. El cual permite identificar y clasificar bacterias en función de los genes del RNA ribosomal, a la vez automatiza el proceso de ribotipado y es capaz de analizar hasta 48 muestras

al día. (Du Pont, 2010). En la Universidad Purdue de Estados Unidos una científica usó un equipo basado en tecnología infrarroja para detectar la bacteria E. Coli en carne contaminada por aproximadamente una hora. En España un grupo de investigadores del consejo Superior de Investigaciones Científicas (CSIC) desarrolló una nueva técnica llamada "el detector Nanomecánico" que permite detectar e identificar con gran sensibilidad y selectividad bacterias y virus en función de su masa y su rigidez mecánica. En el año 2001 en el Perú, como parte del "Estudio transversal de los agentes etiológicos de diarrea aguda" en la Microrregión Sur del país, el Laboratorio Referencial de Tacna detectó en una muestra de heces la bacteria Escherichia Coli en un lactante de 11 meses de edad. Esta cepa fue confirmada por el Instituto Nacional de Salud, por lo cual se realizó un análisis del número de casos de diarrea aguda, en donde se encuentra de 197 a 235 casos. (Blanca Huapaya, 2001) Dicho todo esto en el año 2011 el Perú tuvo una alerta epidemiológica, así lo comunicó el Ministerio de Salud (Minsa) al declarar que todos los servicios sanitarios, a nivel nacional, estuvieron vigilantes y preparados para detectar la bacteria Escherichia Coli (E. Coli), un nuevo virus que cobró la vida de 29 personas traído por viajeros que llegaron de Europa. (MINSA, 2011) Al nivel tecnológico en el Perú tenemos la técnica de diagnóstico molecular que ayuda a la detección de microorganismos patógenos, la PCR simple y múltiple, que es el más aplicado ya que se ayuda a la obtención de resultados más rápidos y exactos, gracias a su elevada sensibilidad y eficiencia. (Palomino Camargo & González Muñoz, 2014). En el año 2012 en Trujillo, se implementó un Sistema de Telediagnóstico de Tuberculosis y Determinación de Multidrogorresistencia Basada en el Método MODS, donde se obtuvieron imágenes de 50 cultivos MODS positivos en pacientes con sospecha de tuberculosis multidrogorresistente. (Rodríguez, y otros,

2014). Según sugerencia de la Organización Mundial de Salud, la prevención en el ciclo alimenticio debe ser prioritaria en las ciudades para detener las infecciones causadas por la bacteria Escherichia Coli, por lo cual es de mucha importancia conocer si un alimento tiene una cantidad de hasta 100 NMP de E. Coli (número más probable), para que de esta manera se pueda prevenir casos aislados o brotes de diarrea, síndrome urémico hemolítico, colitis hemorrágica y cuadros de disentería, principalmente en niños; por esto es necesario conocer la bacteria y mantener la vigilancia epidemiológica. Localmente los análisis o estudios relacionados con el E. Coli no cuentan con tecnología de computacional que ayude a detectar dichas bacterias para poder mejorar el ciclo alimenticio de la sociedad.

Se han considerado los siguientes estudios, trabajos e investigaciones como antecedentes sobre visión computacional y bacterias Escherichia Coli:

- **Según (Roa, 2008)**, en su tesis de maestría "Automatización de los procesos de cuantificación y caracterización de Bacterias en imágenes digitales de Microscopía", presentada a la Universidad de los Andes, en Venezuela.

Los estudios que se llevaron a cabo en las áreas de investigación de la biología generalmente están relacionados con los microorganismos. Tales estudios, requieren de un elevado número de observaciones, lo que demanda una gran inversión de tiempo y esfuerzo por parte del investigador. Un ejemplo de ello es el estudio de bacterias, en donde los procesos de caracterización son generalmente manuales y esto es un proceso lento que produce resultados con errores significativos. Por lo cual este trabajo estudia la automatización del proceso de caracterización de bacterias mediante el uso de técnicas de procesamiento y reconocimiento de patrones, a partir de una muestra en

imagen. Donde para su identificación individual establece la cuantificación con el propósito de automatizar imágenes, en donde se establecieron las siguientes fases: homogeneización del fondo de la imagen, aplicación de filtros para el realce de contornos, segmentación de la imagen mediante la transformada Watershed, optimización de la imagen segmentada y aplicación del template matching.

Como resultado se determinó que el análisis de técnicas de procesamiento se realiza en función de las características que presentan las imágenes digitales de microscopía electrónica de muestras bacterianas. A la vez también se obtuvo que existen factores que afectan la cuantificación manual de bacterias realizadas en los laboratorios, generando para una misma muestra diferencias en los resultados. No obstante, estas diferencias deben ser científicamente aceptadas para que estos resultados tengan validez. Dentro de estos factores destacan: la dilución utilizada, el tiempo que tarda el conteo, la disposición de bacterias en grupos, y en general, la experiencia de la persona que realiza la cuantificación manual.

Por lo tanto, las imágenes procesadas directamente por técnicas de reconocimiento de patrones (Template Matching) no producen resultados confiables, debido a variaciones en los fondos de las imágenes, bacterias muy tenues, similitud en las tonalidades de algunas agrupaciones de bacterias y los fondos, y agrupaciones de bacterias con bordes poco diferenciados.

Igualmente existen factores que pueden afectar los resultados del proceso automático de cuantificación de bacterias, por ejemplo, pigmentación del fondo de la imagen, problemas de iluminación durante el proceso de

adquisición, fondos heterogéneos, bacterias muy tenues e incremento de las áreas ocupadas por las bacterias en la aplicación de filtro de realce.

- **Según** (Pablo, 2013), en su proyecto Fin de Master “Reconocimiento de Imágenes Utilizando Redes Neuronales Artificiales”, presentada a la Universidad Complutense de Madrid.

Nos muestra el proceso de extracción de patrones característicos de imágenes, mediante la ayuda de Redes Neuronales Artificiales. Dicho proceso fue almacenado en una base de datos, y un teléfono móvil consumía dicha información mediante un servicio web. Por lo cual, este proceso utilizó librerías OpenCV, tanto en el servidor como en el teléfono móvil. Para el estudio de la Red Neuronal, en este proyecto se tomaron 25 imágenes las cuales pasaron a formar su población, de ellas se escogieron 18 imágenes (80%) para crear el grupo de entrenamiento de la red neuronal y las otras 7 (20%) para el conjunto de test.

Teniendo en cuenta la dificultad que entraña la identificación de imágenes basadas en su contenido, los resultados ofrecidos en esta investigación han sido de un 80% de detección siendo aceptables ya que fue capaz de identificar las imágenes que se capturan en buenas condiciones. Además, se consiguió demostrar que la infraestructura creada era adecuada para el propósito de la investigación.

- Según (Rojas, Chávez y García, 2006), en su informe “Pruebas Microbiológicas para E. coli”, publicado por Bacteriología Diagnóstica, en Costa Rica.

Nos muestra la realidad de una cepa bacteriana específicamente, ya que, por ello es práctico conocer sobre qué tipo de medios de bacterias se diferencian de otras cepas, aunque sean metabólicamente muy similares. Es por eso que al recibir una muestra es importante definir el algoritmo de pruebas que debe manejarse para lograr identificar el microorganismo de interés, ya que la mayoría de los microorganismos pueden cultivarse sobre substratos nutritivos para el estudio de sus propiedades. Para ellos se realizaron las siguientes pruebas de identificación de enterobacterias para determinar el género y las especies bacterianas de interés: Agar TSI, prueba de otro nitrofenil- $\beta$ -galactopiranosido, prueba de movilidad, prueba de gelatinasa, prueba de rojo de metilo, prueba de Voges – Proskauer, prueba de fenilalanina desaminasa, producción de descarboxilasas y dehidrolasas de aminoácidos, prueba de Ureasa, producción de Indol, utilización de citrato, utilización de malonato, prueba de reducción de nitratos, prueba de fermentación de carbohidratos.

Dando diferentes resultados en cada prueba sobre detección de E. Coli, por ejemplo, para la prueba de reducción y nitratos, la detección de E. Coli es negativa. Pero en la siguiente prueba de Utilización de malonato se encontró un 95% de positividad para cepas de E. Coli.

- Según (Guadalupe Rodríguez, 2002), en el artículo científico titulado “Principales características y diagnóstico de los grupos patógenos de

Escherichia Coli”, publicado por Salud Pública, en México, nos da a conocer que la bacteria Escherichia Coli invade el intestino del hombre pocas horas después del nacimiento y se le considera un microorganismo de flora normal, pero hay cepas que pueden ser patógenas y causar daño produciendo diferentes cuadros clínicos, entre ellos diarrea. Esta bacteria se puede aislar e identificar con base a sus características bioquímicas o serológicas, pero también se pueden estudiar sus mecanismos de patogenicidad mediante ensayos de cultivos celulares o modelos animales, empleando técnicas de biología molecular que evidencian la presencia de genes involucrados en dichos mecanismos.

El autor a su vez da a conocer métodos para la identificación y caracterización de cepas de E. coli, algunos de ellos son de biología molecular, aislamiento de la bacteria y pruebas bioquímicas en tubo como TSI, LIA y MIO. Con esto el autor concluye que la intención de este trabajo es resaltar la importancia del estudio y diagnóstico de E. coli capaz de causar brotes de diarrea, colitis hemorrágica y cuadros de disentería, principalmente en niños.

Como resultado de este estudio realizado por Guadalupe Rodríguez, es que cuando un laboratorio reporta el aislamiento de E. Coli como patógeno, en un cuadro de diarrea, se debe tener presente su importancia como agente causal de cuadros graves de diarrea principalmente en niños menores de cinco años y no sólo considerarla como una bacteria de flora normal.

- **Según (Molina, 2011)**, en su tesis titulada "Papel de Plásmidos e integrones en la multiresistencia a antimicrobianos en cepas de Escherichia coli

uropatogenas y su posible asociación con adherencia e invasividad en células cultivadas", publicada por el instituto Politécnico Nacional, en México.

Nos informa que las cepas de *Escherichia coli* son capaces de causar enfermedades fuera del tracto gastrointestinal perteneciente a un grupo conocido como *E. coli* extra-intestinal (ExPEC). *E. coli* uropatógena (UPEC), es un miembro prominente de la familia ExPEC, es responsable del 90% de las infecciones no complicadas del tracto urinario. Por lo cual, esta tesis tuvo como objetivo principal estudiar las características de virulencia de cepas de *E. coli* aisladas de infecciones del tracto urinario (IsTU) de pacientes de la ciudad de México. Para esto, se investigaron en las diferentes cepas las siguientes propiedades fenotípicas; capacidad de adherencia e invasividad a células de vejiga humana in vitro, serotipificación y susceptibilidad a antibióticos, así como propiedades genotípicas: patrón de plásmidos, integrones y grupos filogenéticos. En donde, se estudiaron 119 cepas de *E. coli*; la susceptibilidad a los antibióticos se determinó por la técnica de difusión en disco. Para la tipificación serológica se utilizó el método de aglutinación en microplaca.

Los resultados de esta tesis mostraron que de las 119 cepas de *E. coli* estudiadas, se encontraron 36 serogrupos diferentes, de los cuales 39 (33.3%) concordaron con los serogrupos considerados como clásicos de uropatógena (O1, O2, O4, O6, O8, O16, O18, O25 y O75) (Ananias et al. 2008, Nowrouzian et al. 2006). También es que este es el primer trabajo realizado en México en donde se encuentran con alta frecuencia el serotipo O25:H4 en pacientes con infecciones del tracto urinario.

- **Según (Silvia Muñoz, Migue Vilca, Daphne Ramos y Juan Lucas, 2013),** realizaron un trabajo de “Frecuencias de enterobacterias en verduras frescas de consumo crudo expedidas en cuatro mercados de Lima - Perú”, publicado por Salud Publica y Salud Ambiental. Dan a conocer la realidad que se está viviendo en los mercados demostrando que los agentes patógenos, que incluyen bacterias E. Coli, toxinas de hongos, virus y helmintos, presentes en las verduras frescas, representan una fuente importante de enfermedades para el ser humano, otros factores a considerar son el incremento de consumidores inmunosuprimidos, así como los cambios en las prácticas agrícolas y de procesamiento de estas; la literatura señala como principal fuente de contaminación de las verduras al estiércol utilizado como fertilizante y el agua de riego sin tratamiento o contaminado. El autor con este proyecto tuvo como objetivo evaluar el grado de contaminación fecal en las verduras de mayor consumo que se expenden en mercados y la determinación de coliformes fecales, E. coli y Salmonella en la superficie de tres verduras de consumo crudo y de mayor demanda. En donde, se recolectaron 15 muestras de lechuga, 15 de col y 15 de espinaca, para ser procesadas por el método del Número más Probable para la detección y recuento de coliformes fecales y E. Coli Tipo I, así como por la prueba de Ausencia/Presencia para Salmonella.  
  
Dando como resultado el estudio demostró que el 18.9% del total de verduras presentaron niveles de coli fecales superiores a lo establecido como apto por

la International Commission on Microbiological Specifications for Foods (ICMSF), siendo la espinaca la más contaminada.

- **Según (Rodríguez y otros, 2014)**, en el artículo titulado "Implementación De Un Sistema De Telediagnóstico De Tuberculosis Y Determinación De Multidrogorresistencia Basada En El Método MODS En Trujillo, Perú", publicado por la Revista Peruana de medicina de Salud Pública.

Nos muestran que la tuberculosis es uno de los problemas de salud pública más serios en el Perú y en el mundo. Por lo cual este necesita un tratamiento adecuado que requiere un diagnóstico temprano, además del conocimiento de la susceptibilidad a las drogas. La falta de un diagnóstico temprano y la incapacidad de determinar la susceptibilidad a drogas, hacen que los tratamientos antituberculosos no sean oportunos y apropiados, lo que compromete seriamente los esfuerzos realizados por controlar esta enfermedad.

Por ello se llevó a cabo el siguiente algoritmo de reconocimiento de *Mycobacterium tuberculosis* a partir de imágenes digitales de cultivos MODS de muestras de esputo donde se obtiene un resultado de 95.4% positivo en la detección de la tuberculosis. El proceso general para el reentrenamiento de un algoritmo consiste en 3 pasos: Tratamiento de imagen, desenfoque gaussiano y binarización adaptativa; dichos procesos de tratamiento mencionados generan múltiples imágenes que terminan resaltando los objetos, sus bordes y los esqueletos.

Dando como resultados lo siguiente: la sensibilidad y la especificidad en objetos, para reconocer cordones de tuberculosis fueron de 92,04% y de 94,93% respectivamente y la sensibilidad y la especificidad en foto para determinar un campo positivo fueron de 95,4% y de 98,07%.

- **Según (Martha Vargas, Margarita Molina y Angela Lluque, 2012)**, en un informe titulado "Niveles de Resistencia a Quinolonas y otros Antimicrobianos en Cepas de Escherichia Coli Comensales en niños de la Zona Periurbana de Lima, Perú", publicado por Revista Peruana de Medicina Experimental y Salud Pública.

Nos da a conocer la realidad problemática de la resistencia a antimicrobianos, sobre todo la presencia de multiresistencia, es considerada uno de los problemas de salud de mayor relevancia a nivel mundial según la Organización Mundial de la Salud. Donde, el objetivo del presente trabajo es analizar los niveles de resistencia a antimicrobianos en E. Coli comensales aislados de niños peruanos hasta doce meses. Por lo cual, se utilizaron muestras de 560 niños sanos entre los 2 a 12 meses. Donde se aplicaron métodos bioquímicos convencionales y a la vez, se estableció sensibilidad a ampicilina, amoxicilina más ácido clavulánico, cefotaxima, ceftazidima, gentamicina, ácido nalidixico, ciprofloxacino, tetraciclina, cloranfenicol, cotrimoxazol y nitrofurantoina mediante la técnica de difusión en disco

Dando como resultados de sensibilidad obtenidos, se observaron elevados niveles de resistencia a ampicilina (62,6%), cotrimoxazol (48,6%) y tetraciclina (43,0%), mientras que la resistencia fue menos elevada a

cloranfenicol (15,8%) y gentamicina (10.4%). Los niveles al resto de antimicrobianos testados fue menor al 10%.

Conjuntamente, en este trabajo de investigación recogemos conceptos como:

### **Visión Computacional**

Visión es la ventana al mundo de muchos organismos. Su función principal es reconocer y localizar objetos en el ambiente mediante el procesamiento de las imágenes. La visión computacional es el estudio de estos procesos, para entenderlos y construir máquinas con capacidades similares. Existen varias definiciones de visión, entre estas podemos mencionar las siguientes. (Gómez & Sucar, 2008)

- Visión es saber que hay y dónde mediante la vista” (Aristóteles).
- “Visión es recuperar de la información de los sentidos (vista) propiedades válidas del mundo exterior”. (Gibson, 1979)
- “Visión es un proceso que produce a partir de las imágenes del mundo exterior una descripción que es útil para el observador y que no tiene información irrelevante” (Marr, 1982)

Las tres son esencialmente válidas, pero la que tal vez se acerca más a la idea actual sobre visión computacional es la definición de Marr. En esta definición hay tres aspectos importantes que hay que tener presentes: (i) visión es un proceso computacional, (ii) la descripción a obtener depende del observador y (iii) es necesario eliminar la información que no sea útil (reducción de información). (Gómez & Sucar, 2008)

Un área muy ligada a la de visión computacional es la de procesamiento de imágenes. Aunque ambos campos tienen mucho en común, el objetivo final es diferente. El

objetivo de procesamiento de imágenes es mejorar la calidad de las imágenes para su posterior utilización o interpretación, por ejemplo:

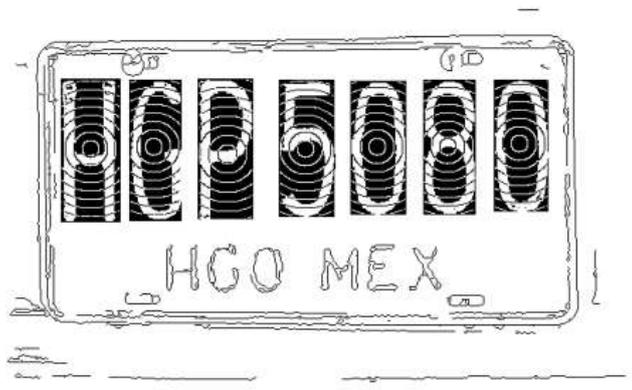
- Remover defectos,
- Remover problemas por movimiento o desenfoco,
- Mejorar ciertas propiedades como color, contraste, estructura, etc.
- Agregar “colores falsos” a imágenes monocromáticas.

En la figura 2 se muestra un ejemplo de procesamiento de imágenes. La tarea a realizar es mejorar la imagen de entrada, la cual es oscura. La imagen de salida es esencialmente la misma, pero de mejor calidad o “más útil”. La figura 3 ilustra la diferencia entre procesamiento de imágenes y visión; nótese que la imagen muestra ciertas descripciones importantes, como los números, que previamente fueron detectados. La salida de este sistema de visión se complementa con un módulo de reconocimiento de patrones, es decir, “saber” qué letras y números contiene la placa. (Gómez & Sucar, 2008)



**Figura 1:** Aumento de contraste: (a) imagen oscura debido a que su rango de grises es reducido, (b) ecualización del rango de grises

**Fuente:** (Gómez & Sucar, 2008)



**Figura 2:** Reconocimiento de caracteres en base a su codificación radial

**Fuente:** (Gómez & Sucar, 2008)

Actualmente existen múltiples aplicaciones prácticas de la visión computacional, entre estas podemos mencionar las siguientes:

- Robótica móvil y vehículos autónomos. Se utilizan cámaras y otros tipos de sensores para localizar obstáculos, identificar objetos y personas, encontrar el camino, etc.
- Manufactura. Se aplica visión para la localización e identificación de piezas, para control de calidad, entre otras tareas.
- Interpretación de imágenes aéreas y de satélite. Se usa procesamiento de imágenes y visión para mejorar las imágenes obtenidas, para identificar diferentes tipos de cultivos, para ayudar en la predicción del clima, etc.
- Análisis e interpretación de imágenes médicas. La visión se aplica para ayudar en la interpretación de diferentes clases de imágenes médicas como rayos-X, tomografía, ultrasonido, resonancia magnética y endoscopía.

- Interpretación de escritura, dibujos, planos. Se utilizan técnicas de visión para el reconocimiento de textos, lo que se conoce como reconocimiento de caracteres. También se aplica a la interpretación automática de dibujos y mapas.
- Análisis de imágenes microscópicas. El procesamiento de imágenes y visión se utilizan para ayudar a interpretar imágenes microscópicas en química, física y biología.
- Análisis de imágenes para astronomía. Se usa la visión para procesar imágenes obtenidas por telescopios, ayudando a la localización e identificación de objetos en el espacio.
- Análisis de imágenes para compresión. Aunque la compresión de imágenes ha sido tradicionalmente una sub área del procesamiento de imágenes, recientemente se están desarrollando técnicas más sofisticadas de compresión que se basan en la interpretación de las imágenes. (Gómez & Sucar, 2008)

### **Imagen Digital**

Una imagen puede definirse como una función bidimensional  $f(x,y)$  donde  $x$  y  $y$  son coordenadas en el plano y la amplitud  $f$  es llamada intensidad o nivel de gris en ese punto.

Cuando  $(x, y)$  y  $f$  son todos finitos (cantidades discretas) llamamos a la función como imagen digital. Es decir, una imagen digital estará compuesta por un número finito de elementos llamados píxeles, cada uno de los cuales con un valor y una posición particular.

El valor es relativo a alguna propiedad del punto que representa, como por ejemplo su brillo o su matiz.

El término píxel (abreviación de Picture element o elemento de imagen), se trata de la unidad mínima de información de una imagen, la cual aparece como un punto en la pantalla o en una impresora. En realidad, cada píxel se compone de tres registros de color, mediante la combinación de cierta cantidad de rojo, verde y azul, el píxel adopta un color particular.

Las imágenes bidimensionales son el resultado de una proyección en perspectiva de escenas tridimensionales. Cuando se obtiene una imagen bidimensional del mundo tridimensional desaparece gran cantidad de información. (García Santillán, 2008)

#### **a) Clasificación de las Imágenes Digitales**

A grandes rasgos podríamos dividir las imágenes digitales en dos grupos:

- 1) Imágenes Vectoriales:** Los gráficos vectoriales conservan la nitidez de los bordes y no pierden detalles cuando se modifica el tamaño puesto que son independientes de la resolución. La información de cada uno de los puntos se recoge en forma de ecuación matemática que lo relaciona con el resto de los puntos que forman la imagen. Ofrece la gran ventaja de que la calidad de la imagen no varía al modificar el tamaño, ya que la información de cada punto no es absoluta sino relativa al resto de la imagen. Además, debido a su definición matemática, apenas ocupa espacio, ya que una fórmula que represente su forma es suficiente para representar todos los puntos que la componen. Es el tipo adecuado para el diseño de líneas, polígonos y figuras. No es soportado de forma directa por los navegadores de Internet como: Internet Explorer, Netscape Navigator, Firefox, Mozilla, etc.  
Algunos formatos de este tipo de imágenes son: DWG (autoCAD), SWF y FLA (Flash).

**2) Imágenes Raster o Mapa de Bits:** Las imágenes raster constan de un número fijo de píxeles y, por tanto, dependen de la resolución. Las imágenes raster pueden perder detalle y verse dentadas (pixeladas) cuando se amplían. Utilizan una cuadrícula rectangular de elementos de imagen (píxeles) para representar las imágenes. A cada píxel se le asigna una ubicación y un valor de color específico. La ventaja que presenta este formato es la posibilidad de recoger una amplia gama tonal, por lo que es el tipo adecuado para representar imágenes captadas de la realidad. En cambio, la variación de tamaño supondrá modificaciones en la calidad, ya que el número de celdas que forman la imagen permanece invariable, por lo que un aumento del tamaño hace que el único recurso posible sea ampliar el tamaño de cada una de las celdas. Podemos deducir que su tamaño es muy grande por la información de cada uno de los puntos que forman la imagen.

Dentro de este tipo se encuentran muchos formatos, algunos de los cuales son soportados directamente por los navegadores, siendo el tipo de imágenes con las que vamos a trabajar. Estas imágenes son creadas por los scanners y las cámaras digitales. (García Santillán, 2008)

Los formatos más populares son los siguientes:

- **BMP**

El formato bmp (Bit Map) es el formato de las imágenes de mapa de bits de Windows. Su uso fue muy extendido, pero los archivos son muy grandes dado la escasa compresión que alcanzan.

**Tabla 1** Formato BMP

Formato	Profundidad de Color	Modos Color	Canales Alfa	Compresión
BMP (.bmp)	1 bit: Blanco y negro.	• RGB		Si: Run Length
	4 – 8 bits: Escala de grises.	• Color Indexado	NO	Encoding (RLE) en 4
	8 bits: Color Indexado.	• Escala de Grises.		y 8 bits
	24 bits: color RGB	• Mapa de Bits.		

*Fuente: (Ordoñez Santiago, 2005)*

- **TIFF**

El formato TIFF (Tag Image File Format) se utiliza para imágenes de mapa de bits y es admitido prácticamente por todas las aplicaciones de autoedición y tratamiento de imágenes. Este formato fue desarrollado por Aldus Corporation. Lo reconocen casi todos los programas. Además, es compatible con PC y Mac. Su uso es de los más extendidos en la industria gráfica por la calidad de imagen y de impresión que presenta.

*Tabla 2 Formato TIFF*

Formato	Profundidad de Color	Modos Color	Canales Alfa	Compresión
		<ul style="list-style-type: none"> <li>• RGB</li> <li>• Color</li> </ul>		
TIFF (.tif)	32 bits	Indexado	SI	Si: Lempel Ziv Welc (LZW)
		<ul style="list-style-type: none"> <li>• Escala de Grisés.</li> <li>• Mapa de Bits.</li> </ul>		

*Fuente: (Ordoñez Santiago, 2005)*

- **GIF**

El formato GIF corresponde a las siglas de Graphics Interchange Format propiedad de eCompuServe. El formato GIF es preferible para las imágenes de tonos no continuos o cuando hay grandes áreas de un mismo color ya que utiliza una paleta de color indexado que puede tener un máximo de 256 colores. Una de sus mayores ventajas es que podemos elegir uno o varios colores de la paleta para que sean transparentes y podamos ver los elementos que se encuentren por debajo de estos. También es uno de los pocos formatos de imagen con el que podemos mostrar animaciones porque hace que distintos frames se ejecuten secuencialmente. Además, es un formato de compresión diseñado para disminuir el tiempo de transferencia de datos por las líneas telefónicas.

*Tabla 3 Formato GIF*

Formato	Profundidad de Color	Modos Color	Canales Alfa	Compresión
GIF (.gif)	8 bits (256 colores)	• RGB	NO	Si: Lempel
		• Color Indexado		Ziv Welc (LZW)
		• Mapa de Bits.		

*Fuente: (Ordoñez Santiago, 2005)*

• **JPG o JPEG**

Este formato toma su nombre de Joint Photographic Experts Group, asociación que lo desarrollo. Se utiliza usualmente para almacenar fotografías y otras imágenes de tono continuo. Gracias a que utiliza un sistema de compresión que de forma eficiente reduce el tamaño de los archivos. En contraste con GIF, JPEG guarda toda la información referente al color con millones de colores (RGB) sin obtener archivos excesivamente grandes. Además, los navegadores actuales reconocen y muestran con fidelidad este formato.

*Tabla 4 Formato JPG o JPEG*

Formato	Profundidad de Color	Modos Color	Canales Alfa	Compresión
JPEG (.jpg; .jpe)	24 bits	• RGB	NO	Si: con pérdidas
		• Escala de Grises		
		• CMYK		

*Fuente: (Ordoñez Santiago, 2005)*

- **PNG**

PNG son las siglas del grupo que lo desarrollo Portable Networks Graphics pensando en un formato ideal para su distribución en Internet. PNG posee ventajas respecto a los otros formatos más comunes en este medio: JPG y GIF. Ya que fue desarrollado especialmente para su distribución en red posee gran parte de las ventajas de un GIF y de un JPG. Por ejemplo, permite altos niveles de comprensión, además, permite utilizar la técnica de la indexación para crear colores transparentes, semitransparencias o transparencias degradadas. Finalmente, no está limitado a una paleta de 256 colores, sino que puede utilizar millones de colores. Su única limitación es que no podemos crear ficheros animados.

*Tabla 5 Formato PNG*

Formato	Profundidad de Color	Modos Color	Canales Alfa	Compresión
PNG (.png)	24 bits	<ul style="list-style-type: none"> <li>• RGB</li> <li>• Escala de Grises</li> <li>• Color Indexado</li> <li>• Mapa de Bits</li> </ul>	SI	Si: con pérdidas

*Fuente: (Ordoñez Santiago, 2005)*

## **b) Tipos de Imágenes Digitales**

En el procesamiento digital de imágenes (PDI) se maneja cuatro tipos de imágenes básicamente: imágenes RGB, imágenes indexadas, imágenes en escala de grises e imágenes binarias, las cuales se explicarán a continuación. (García Santillán, 2008)

### **1) Imágenes RGB (red-green-blue):**

- Utilizan tres canales para reproducir los colores en la pantalla.
- Utilizan 8 bits por canal (8 bits x 3), es decir, 24 bits de color para cada píxel.
- Reproducen hasta 16,7 millones de colores.
- Soporta algunos formatos como: JPG, BMP, PNG, etc.

### **2) Imágenes Indexadas**

- Reduce los colores de la imagen a un máximo de 256.
- Admiten los formatos GIF y PNG-8 y muchas aplicaciones multimedia.
- Reduce el tamaño de archivo porque elimina la información del color.

### **3) Imágenes en Escala de Grises**

- Utilizan distintos tonos de gris.
- En imágenes de 8 bits, puede tener hasta 256 tonos de gris.

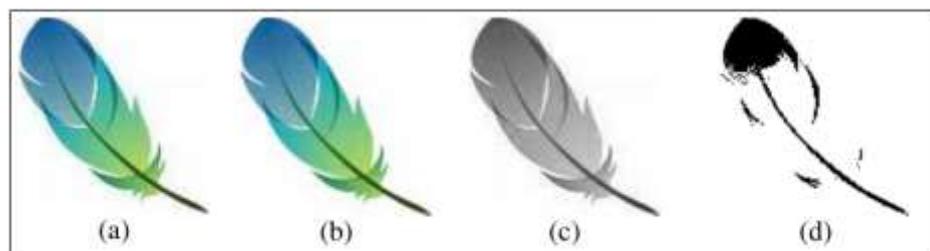
- Cada píxel tiene un valor de brillo comprendido entre 0 (negro) y 255 (blanco).

#### 4) Imágenes Binarias

- Tienen una profundidad de color de 1 bit.
- Utiliza uno de los dos valores de color (blanco o negro) para representar los píxeles de una imagen.

Existen además imágenes con una profundidad de píxel de 32 bits. Los 8 bits (1 byte) adicionales de profundidad sobre las imágenes de 24 bits, le permiten almacenar la transparencia de la imagen. Este byte adicional es generalmente llamado máscara o canal alfa, y almacena diferentes valores de transparencia.

En la Figura 3 se muestra algunos ejemplos correspondientes a los tipos de imágenes.



**Figura 3:** Tipos de Imágenes Digitales; (a) RGB; (b) Indexada; (c) Escala de Grises; (d) Binaria

**Fuente:** (Garcia Santillán, 2008)

## 5) Tamaño de una Imagen

**Tabla 6** Tamaño de una imagen

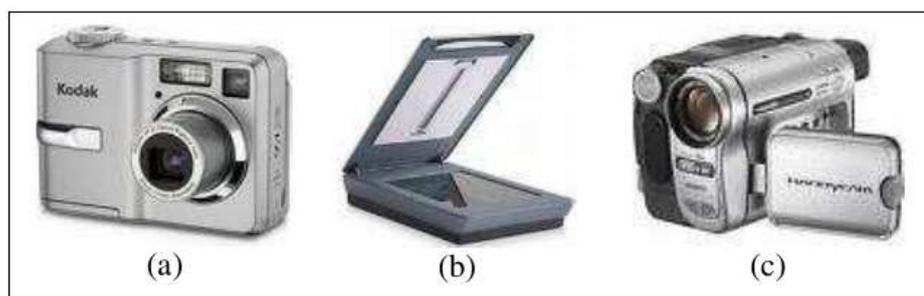
Resolución	Profundidad del Píxel	Tamaño del Archivo			
		Bits	Bytes	Kbytes	Mbytes
640 x 480	x 1 bit	= 307.200	= 38.400	= 37.5	= 0.036
640 x 480	x 8 bit	= 2'457.600	= 307.200	= 300	= 0.292
640 x 480	x 24 bit	= 7'372.800	= 921.600	= 900	= 0.878
640 x 480	x 32 bit	=9'830.400	= 1'228.800	= 1200	= 1.171

Fuente: (Garcia Santillán, 2008)

### c) Captura y Digitalización de Imágenes

#### 1) Captura de imágenes:

El proceso de captura se refiere a la adquisición de la imagen del mundo físico. La imagen puede ser capturada o generada de varias maneras: fotográficamente, con elementos químicos; o a través de dispositivos electrónicos como el computador, escáner, cámara digital o video-cámara digital.



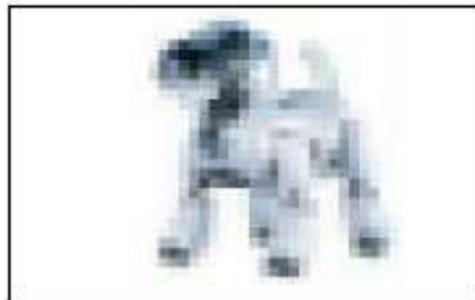
**Figura 4:** Dispositivos de captura; (a) cámara digital; (b) escáner; (c) video cámara digital

Fuente: (Garcia Santillán, 2008)

## 2) Digitalización:

En la digitalización normalmente se distinguen dos procesos: el Muestreo y la Cuantización

- **El Muestreo:** Es el proceso de obtener la imagen. Esto implica que la imagen se muestrea en una matriz con  $m$  filas y  $n$  columnas. El muestreo determina el tamaño del píxel y el valor del brillo. Cuando un dispositivo de captura muestrea la imagen fotográfica, divide la imagen en píxeles. El tamaño de los píxeles depende del número de foto celdas. En las figuras 5 y 6 se muestran ejemplos de muestreos con diferentes valores para  $m$  y  $n$ .



*Figura 5: Imagen con baja resolución*

*Fuente:* (Garcia Santillán, 2008)



*Figura 6: Imagen con Alta resolución*

*Fuente:* (Garcia Santillán, 2008)

- **La Cuantización:** Es el proceso de asignar valores a los elementos de la matriz. Cada valor representa al valor de la variable física en ese punto.

A efectos de representación visual se asume que el valor más pequeño del rango de valores corresponde a un nivel de gris negro y que el valor más grande al nivel de gris blanco. Dentro de este intervalo, cuantos más valores se puedan discriminar, mayor cantidad de matices se podrán representar. (Garcia Santillán, 2008)

### **Orígenes del Procesamiento Digital de Imágenes**

La historia del PDI se remonta a la década de los 60 y está directamente ligada con el desarrollo y evolución de las computadoras. Su progreso ha ido de la mano con el desarrollo de las tecnologías de hardware, ya que requiere un alto poder y recursos computacionales para almacenar y procesar las imágenes. De igual manera el desarrollo de los lenguajes de programación y los sistemas operativos han hecho posible el crecimiento continuo de aplicaciones relacionadas al procesamiento de imágenes, tales como: imágenes médicas, satelitales, astronómicas, geográficas, arqueológicas, biológicas, aplicaciones industriales, entre otras. (Garcia Santillán, 2008)

#### **a) Ejemplos de campos de aplicación PDI**

En este caso nuestro criterio de clasificación será por su fuente. La principal fuente de energía de las imágenes es el espectro electromagnético (otras fuentes son la acústica, la ultrasónica, y la electrónica). Las imágenes basadas en la radiación del espectro EM son los más familiares, especialmente las imágenes de rayos X y las bandas visuales del espectro.

Las ondas electromagnéticas pueden ser conceptualizadas como ondas sinusoidales de longitudes de ondas variadas, o como una corriente de partículas sin masas viajando en un patrón de ondas y moviéndose a la velocidad de la luz. Su unidad es el fotón.

Agrupando las bandas espectrales de acuerdo a su energía por fotón, obtenemos el espectro de la figura, desde los rayos gamma (con mayor energía) hasta las ondas de radio. (Mejía Vilet, 2004)

A continuación, se detalla algunos ejemplos de procesamientos de imágenes:

- **Rayos Gamma:**

Sus aplicaciones más importantes: la medicina nuclear y las observaciones astronómicas.

- Medicina nuclear

Se inyecta al paciente un isótopo radioactivo que emite rayos gamma cada vez más débiles, estos se detectan y se forma la imagen.

Otro caso es la Tomografía de Emisión de Positrones PET, similar a la tomografía de rayos X, pero en esta se inyecta al paciente un isótopo radioactivo que emite positrones.

Los positrones chocan con los electrones del paciente, se destruyen ambos, y se producen 2 rayos gamma que se detectan para formar la imagen. (Mejía Vilet, 2004)



*Figura 7:* Tomografía por emisión de positrones (PET)

*Fuente:* (Mejía Vilet, 2004)

- **Rayos X:**

Los rayos X no sólo se usan en medicina, sino también en gran manera en la industria y otras áreas, como la astronomía.

- Angiografía

Es una aplicación de la radiografía de mejora de contraste. Se utiliza para obtener angiogramas, imágenes de venas y arterias. Un cateter se inserta en una de estas y se guía hasta el área a estudiar. Una vez ahí, se inyecta por medio de éste un medio que mejora el contraste y permite al radiólogo ver irregularidades o bloqueos. (Mejía Vilet, 2004)



*Figura 8:* Angiograma de la aorta

*Fuente:* (Mejía Vilet, 2004)

- Tomografía axial computarizada (CAT)



*Figura 9:* Tomografía de la cabeza

*Fuente:* (Mejía Vilet, 2004)

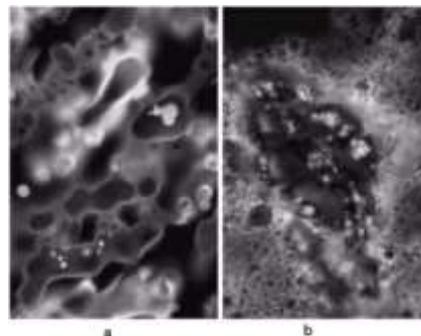
- **Ultravioleta:**

Las aplicaciones incluyen litografía, inspección industrial, microscopía, láser, imágenes biológicas y observaciones astronómicas.

- Microscopía de fluorescencia

La fluorescencia se observó en el siglo diecinueve, al dirigir luz ultravioleta a un mineral con flúor se observó que este brillaba. La luz ultravioleta no es visible, pero cuando un fotón ultravioleta choca con un electrón de un átomo de material fluorescente, este electrón se eleva a un nivel más alto de energía. Después éste regresa a un nivel bajo y emite luz en forma de un fotón de baja energía en la región de luz visible (rojo).

(Mejía Vilet, 2004)



*Figura 10: Ejemplos de imágenes ultravioleta: a) Maíz sano. B) Maíz Enfermo (smut)*

*Fuente:* (Mejía Vilet, 2004)

- **Visible e Infrarroja:**

Estas son, por mucho, las aplicaciones más numerosas. La banda infrarroja se utiliza usualmente en conjunto con la imagen visual. (Mejía Vilet, 2004)

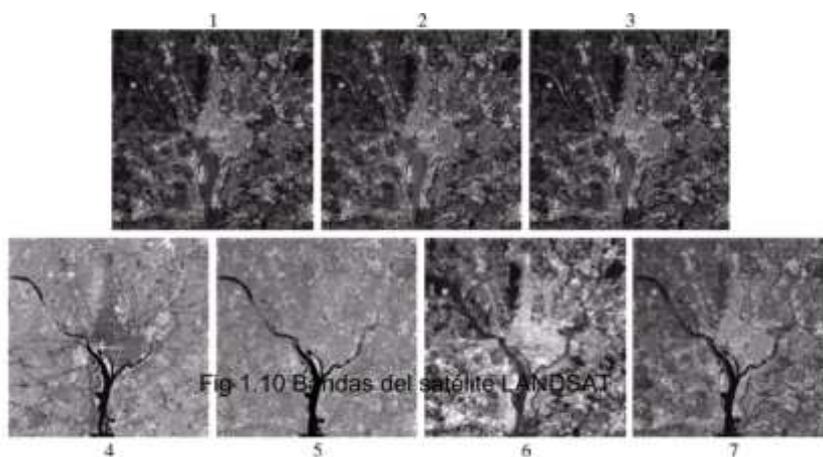
- Detección remota

Usualmente incluye varias bandas que detectan diferentes detalles.

*Tabla 7 Detección remota*

Banda	Nombre	Longitud de Onda	Usos y Características
1	Azul visible	0.45 – 0.52	Penetración del agua
2	Verde visible	0.52 – 0.60	Bueno para medir la salud de las plantas
3	Rojo visible	0.63 – 0.69	Discriminación de vegetación
4	Infrarrojo bajo	0.76 – 0.90	Mapeo de costas y biomasa
5	Infrarrojo medio	1.55 – 1.75	Contenido de humedad en el suelo y vegetación
6	Infrarrojo termal	10.4 – 12.5	Humedad del suelo y mapeo termal
7	Infrarrojo alto	2.08 – 2.35	Mapeo mineral

*Fuente:* (Mejía Vilet, 2004)



**Figura 11:** Bandas del satélite LANDSAT

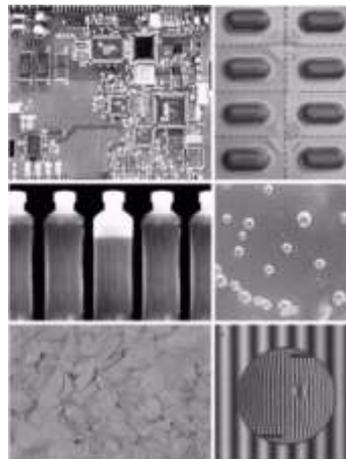
*Fuente:* (Mejía Vilet, 2004)

Estas imágenes pueden ser usadas para ver, en ciudades, movimientos o crecimiento de población, contaminación, u otros factores que dañan al ambiente. Otras aplicaciones incluyen observación y predicción del clima, o, como en el caso de la base de datos de fotografías Nighttime lights of the

world, un inventario de asentamientos humanos mundiales (infrarrojo).  
(Mejía Vilet, 2004)

- Control de calidad

Una aplicación muy utilizada de la luz visible e infrarroja es en el control de calidad, donde se utiliza para revisar productos de orden variado (niveles correctos en botellas, circuitos integrados con todos los componentes, errores de fabricación o material defectuoso). (Mejía Vilet, 2004)



**Figura 12:** Control de calidad en circuito integrado, empaque de píldoras, nivel de líquidos, burbujas en plástico transparente, cereal, y fallas en una lente intraocular.

**Fuente:** (Mejía Vilet, 2004)

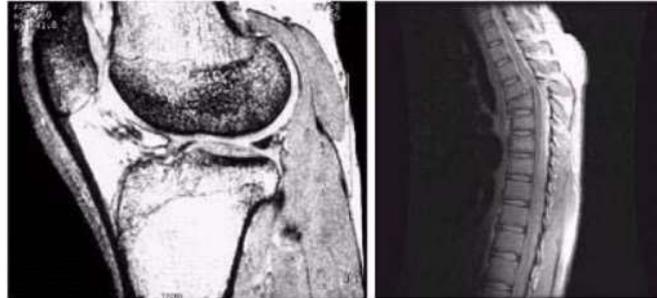
- **Ondas de Radio:**

Mayoritariamente se usa en medicina y astronomía.

- Medicina

Resonancia magnética (MRI). Se introduce al paciente en un imán de gran poder y se pasan ondas de radio por su cuerpo en pulsos cortos. Cada pulso causa un pulso resultante de ondas de radio desde los tejidos del paciente. La computadora determina el lugar y fuerza de estos pulsos, y

produce una imagen bidimensional de una sección del paciente. (Mejía Vilet, 2004)



*Figura 13:* Imágenes MRI de una rodilla y una espina

*Fuente:* (Mejía Vilet, 2004)

- **Aplicaciones de otras fuentes de energía:**

- Sonido

Se aplica en exploración geológica (petróleo y minerales), la industria, y medicina. En la primera se utiliza sonido en la parte baja del espectro (cientos de Hertz), y las otras áreas utilizan ultrasonido.

Por ejemplo, en geología se utiliza un gran camión para presionar un aplaca en el suelo, y se hace vibrar el camión a una frecuencia de cerca de 100 Hz. Las ondas que regresan se analizan.

Un método similar se utiliza para adquisición submarina, al disparar 2 pistolas de aire y recuperar las ondas de regreso para producir un mapa 3D del fondo del océano.

- Microscopía de electrones

Un microscopio de transmisión de electrones (TEM) trabaja como un proyector de acetatos. El proyector transmite un rayo de electrones a través del espécimen y se proyecta en una pantalla de fósforo. Un microscopio de escaneo de electrones escanea el rayo de electrones y

registra la interacción del rayo y la muestra en cada lugar (Mejía Vilet,  
2004)

## b) Componentes de un sistema de procesamiento de imágenes

- **Sensor**

Es un aparato físico sensible a la energía radiada por el objeto.

- **Hardware de procesamiento de imágenes especializado**

Usualmente consta de un digitalizador que convierte la salida del sensor a formato digital, y de hardware que realiza otras operaciones, que procesa operaciones en paralelo de imágenes completas. Su característica más distintiva es la velocidad (realiza operaciones que requieren salidas de datos muy rápidas).

- **Computadora**

En un sistema de propósito general puede ser desde una PC hasta una supercomputadora.

- **Software**

Módulos especializados que llevan a cabo tareas específicas.

- **Almacenamiento en masa**

Obligatorio en aplicaciones de procesamiento de imágenes. El almacenamiento se divide en tres categorías:

- ✓ A corto plazo, utilizado durante el procesamiento (ej. Memoria de la computadora, o memorias especiales “frame buffers”, éstas últimas se utilizan en zooms digitales)
- ✓ En línea, para carga relativamente rápida (discos magnéticos u ópticos)
- ✓ De archivo, de acceso poco frecuente (cintas magnéticas y discos ópticos)

- **Dispositivos de despliegue de imágenes**

Por ejemplo, monitores a color (de preferencia planos, para evitar distorsiones), o dispositivos estéreo tales como lentes para estéreo visión.

- **Dispositivos de copia dura (hardcopy)**

Impresoras láser o tinta, cámaras de película, dispositivos sensibles al calor, unidades digitales como cd-rom.

- **Red**

Casi una función por defecto en cualquier sistema computacional moderno, lo más importante es el ancho de banda.

## **Fundamentos del Procesamiento de Imágenes Digitales**

### **a) Relación entre Píxeles**

Un píxel  $p$  con coordenadas  $(x,y)$  tiene cuatro vecinos, dos horizontales y dos verticales, cuyas coordenadas son:  $(x+1,y)$ ,  $(x-1,y)$ ,  $(x,y-1)$ ,  $(x,y+1)$ . A este conjunto de píxeles se llama vecindad 4 de  $p$  y se denota por  $N_4(p)$ , ver la figura

14 nótese que para cada uno de estos píxeles hay una distancia de 1 (uno) desde  $p$  y que en los bordes de la imagen algunos de estos píxeles quedarán fuera de la imagen. (Garcia Santillán, 2008)

	$(x-1, y)$	
$(x, y-1)$	$(x, y)$	$(x, y+1)$
	$(x+1, y)$	

**Figura 14:** Vecindad  $N_4(p)$

*Fuente:* (Garcia Santillán, 2008)

Existen también 4 vecinos diagonales de  $p$  con coordenadas:  $(x+1, y+1)$ ,  $(x+1, y-1)$ ,  $(x-1, y-1)$ ,  $(x-1, y+1)$  y se les denota por  $N_D(p)$ , ver la figura 15 conjuntamente,  $N_4(p)$  y  $N_D(p)$  forman la vecindad 8 de  $p$  denotada por  $N_8(p)$ .

$(x-1, y-1)$		$(x-1, y+1)$
	$(x, y)$	
$(x+1, y-1)$		$(x+1, y+1)$

**Figura 15:** Vecindad  $N_D(p)$ .

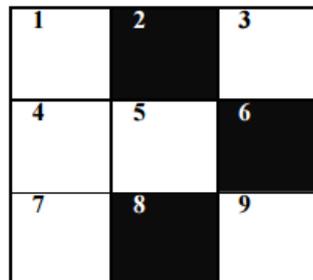
*Fuente:* (Garcia Santillán, 2008)

## b) Conectividad

La conectividad es un concepto importante utilizado para establecer los límites de objetos en regiones dentro de una imagen digital. Para determinar si dos píxeles están conectados se determina si son adyacentes en algún sentido, sea  $N_D(p)$  o  $N_4(p)$  por ejemplo, y si sus niveles de gris satisfacen algún criterio de similitud (si son iguales o parecidos). Por ejemplo, en una imagen binaria con valores de 1 y

0, dos píxeles pueden ser vecinos  $N_4(p)$ , pero se dice que están conectados sólo cuando tienen el mismo valor. (García Santillán, 2008)

En la figura 16 se observa la conectividad de píxeles en una imagen binaria. El píxel 6 está conectado con el 2 y 8. El píxel 3 está conectado con el 5.



**Figura 16:** Conectividad de píxeles

**Fuente:** (García Santillán, 2008)

### c) Distancia

La distancia o transformada de distancia proporciona una medición de la separación existente entre dos puntos dentro de una imagen. Dados dos píxeles  $p$  y  $q$  con coordenadas  $(x,y)$  y  $(s,t)$ , respectivamente, se puede definir una función de distancia  $D$  si se cumple:

$$D(p,q) \geq 0$$

$$D(p,q)=0, \text{ si } p=q$$

$$D(p,q)=D(q,p)$$

Las funciones de distancia comúnmente usadas son: distancia euclidiana y distancia tablero de ajedrez.

**Distancia euclidiana** entre  $p$  y  $q$ :  $D_E ( p,q) = \sqrt{(x - s)^2 + (y - t)^2}$

En la figura 17 se muestra la distancia euclidiana para una imagen de 5 por 5.

$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
2	1	0	1	2
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$

**Figura 17:** Distancia euclidiana para una imagen de 5 por 5.

*Fuente:* (Garcia Santillán, 2008)

**Distancia tablero de ajedrez** en donde se observa que los 4-vecinos están a una distancia unitaria del píxel central; si se desea que los 8-vecinos estén a la misma distancia se toma:  $D(p, q) = \text{Max}(x - s, y - t)$

En la figura 18 se muestra la distancia tablero de ajedrez para una imagen de 5 por 5.

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

**Figura 18:** Distancia tablero de ajedrez para una imagen de 5 por 5.

*Fuente:* (Garcia Santillán, 2008)

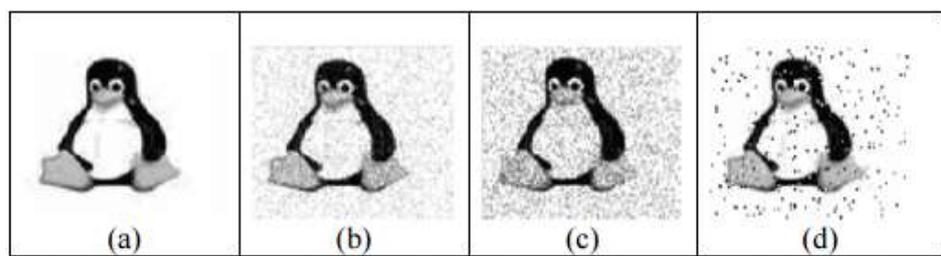
#### d) Ruido de Imágenes

Todas las imágenes tienen cierta cantidad de ruido, la cual se puede deber a la cámara, escáner o al medio de transmisión de la señal. Generalmente el ruido se manifiesta como píxeles aislados que toman un nivel de gris diferente al de sus vecinos. Los algoritmos de filtrado permiten eliminar o disminuir este ruido. .  
(Garcia Santillán, 2008)

El ruido puede clasificarse en los siguientes tipos:

- a) **Gaussiano:** Produce pequeñas variaciones en la imagen; generalmente se debe a diferentes ganancias en la cámara, ruido en los digitalizadores, perturbaciones en la transmisión. Se considera que el valor final del píxel sería el valor ideal más una cantidad correspondiente al error que puede describirse como una variable aleatoria gaussiana.
- b) **Impulsional (sal y pimienta):** el valor que toma el píxel no tiene relación con el valor ideal, sino con el valor del ruido que toma valores muy altos o bajos (puntos blancos y/o negros) causados por una saturación del sensor o por un valor mínimo captado, si se ha perdido la señal en ese punto.
- c) **Multiplicativo:** La imagen obtenida es el resultado de la multiplicación de dos señales.

En la figura 19 se muestran los diferentes ruidos afectando a una imagen.



**Figura 19:** Diferentes ruidos afectando a una imagen; (a) Original; (b) Gaussiano; (c) Multiplicativo; (d) Sal y Pimienta.

*Fuente:* (García Santillán, 2008)

### Procesamiento Digital de Imágenes

El procesamiento digital de imágenes (PDI) se refiere a procesar las imágenes del mundo real de manera digital por medio de un computador. Es un tema muy amplio, en el que se incluyen estudios de física, matemáticas, ingeniería eléctrica,

computación. Estudia los fundamentos conceptuales de la adquisición y despliegue de imágenes y con detalle los fundamentos teóricos y algorítmicos del procesamiento como tal. Además, como objetivo mejorar el aspecto de las imágenes y hacer más evidentes en ellas ciertos detalles que se desean hacer notar.

La visión artificial o visión computacional puede ser definida como los procesos de obtención, caracterización e interpretación de información de imágenes tomadas de un mundo tridimensional. Consideramos tres niveles de procesamiento: visión de bajo, medio y alto nivel, tal como se muestra en la tabla 8.

**Tabla 8** Niveles de visión y procesos del PDI

Procesos del PDI	Nivel de Visión
1. Captura / adquisición	Bajo
2. Preprocesamiento	
3. Segmentación	
4. Descripción	Medio
5. Reconocimiento	
6. Interpretación	Alto

**Fuente:** (García Santillán, 2008)

### 1) Captura y Adquisición

Es el proceso a través del cual se obtiene una imagen digital utilizando un dispositivo de captura, a continuación, mencionaremos los más relevantes:

#### a) Cámara Digital

- **Xiaomi Redmi Note 5:** Nos proporciona una captura con una calidad de 12 megapíxeles.

## b) Microscopio

- **Olympus CX 21:** El lente objetivo nos brinda 100x que viene hacer el máximo **por** el ocular que es de 10x.

Con lo proporcionado por este microscopio llegamos a 1000x de aumento para la visualización de la bacteria.



*Figura 20:* Microscopio Olympus CX21

*Fuente:* Elaboración Propia

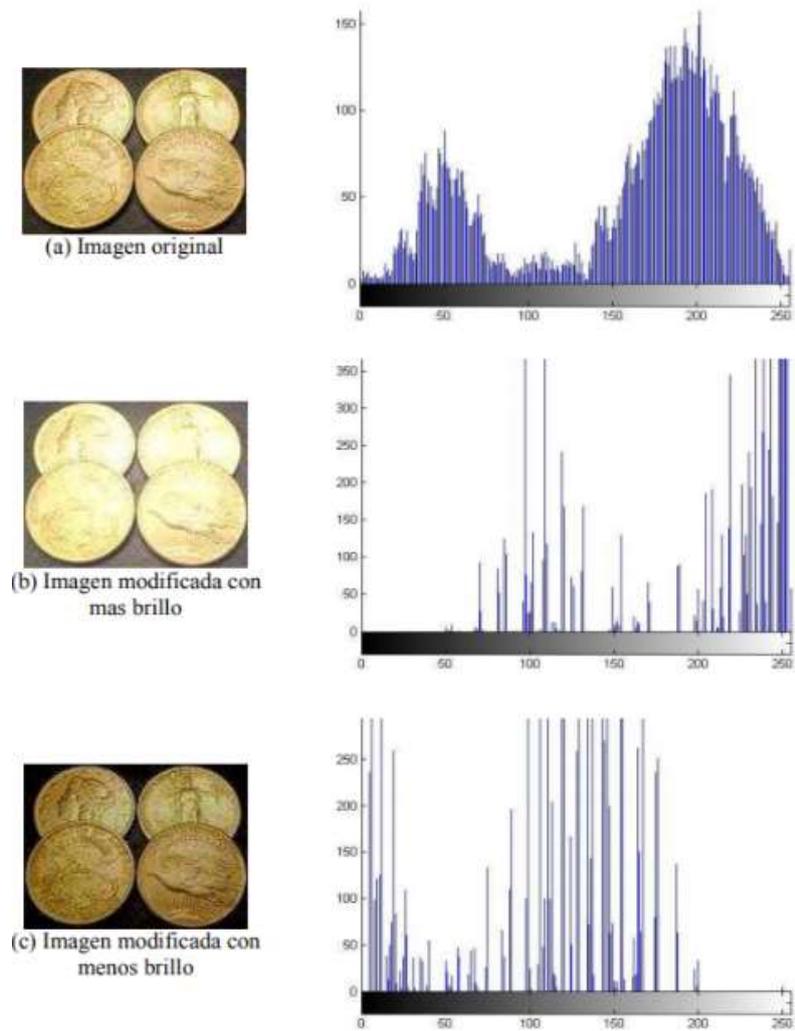
## 2) Pre-Procesamiento

Incluye técnicas tales como la reducción del ruido, realce del contraste, realce de ciertos detalles, o características de la imagen.

### a) Métodos de realce:

- **Aumento y Reducción de Contraste:**

Este método de acuerdo a un histograma ilustra en un gráfico cómo están distribuidos los píxeles de la imagen mostrando la cantidad de píxeles en cada nivel de intensidad del color. Dado lo mencionado se puede aumentar o reducir el contraste de una imagen.



*Figura 21:* Imágenes con distintos niveles de brillo con sus histogramas.

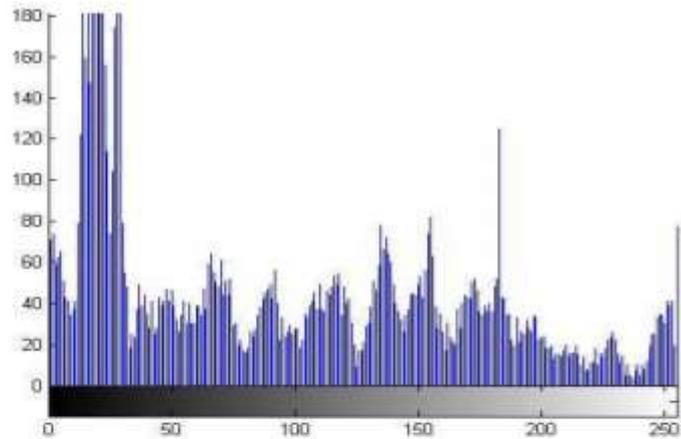
*Fuente:* (Garcia Santillán, 2008)

- **Ecuilizado del Histograma:**

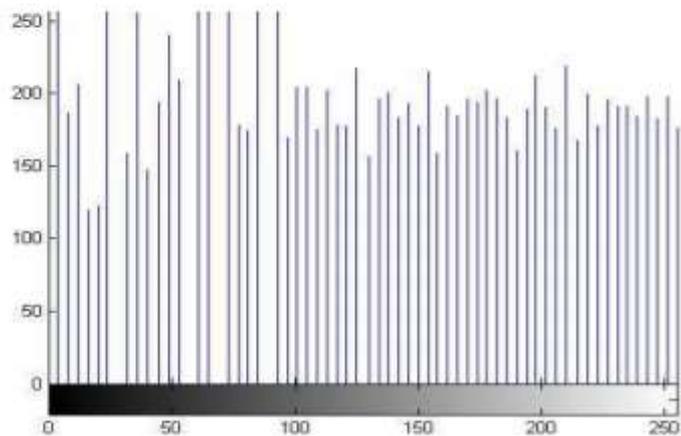
El proceso de ecualizado mejora el contraste de una imagen y tiene por objetivo obtener un nuevo histograma, a partir del histograma original, con una distribución uniforme de los diferentes niveles de intensidad. Además, mejora la calidad visual de las imágenes saturadas.



(a)



(b)

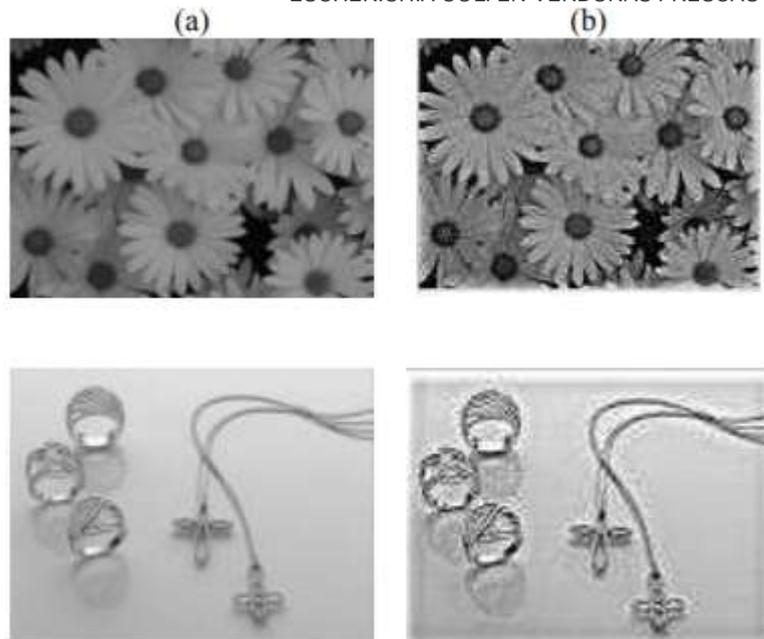


**Figura 22:** Ecuilizado del histograma de una imagen; (a) imagen original con su respectivo histograma; (b) ecualizado del histograma.

*Fuente:* (Garcia Santillán, 2008)

- **Realce de Bordes:**

Consiste en resaltar aquellos píxeles que tienen un valor de gris diferente al de sus vecinos. Si la imagen contiene ruido, su efecto se multiplicará, por lo que primero se debe eliminar el ruido.

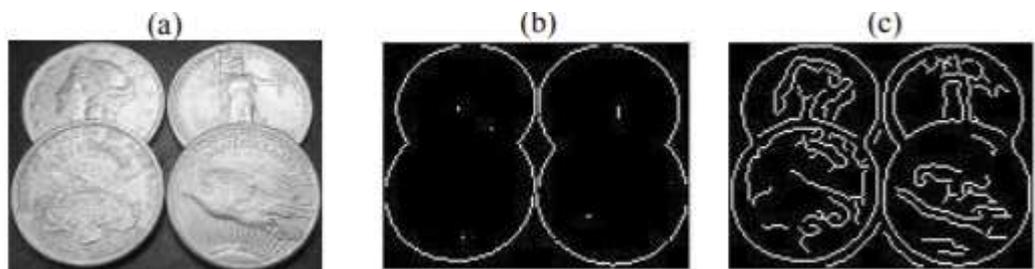


**Figura 23:** Realce de bordes en Imágenes; (a) Imágenes originales; (b) Imagen con bordes realzados.

*Fuente:* (García Santillán, 2008)

- **Detección de Contornos:**

Es un paso intermedio en el reconocimiento de patrones en imágenes digitales. En una imagen, los contornos corresponden a los límites de los objetos presentes en la imagen. Para hallar los contornos se buscan los lugares en la imagen en los que la intensidad del píxel cambia rápidamente.



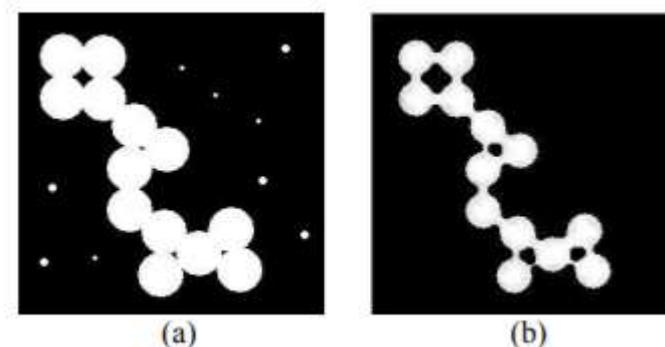
**Figura 24:** Detección de contornos en Imágenes; (a) Imagen original; (b) Detección con filtro de Sobel; (c) Detección con filtro de Canny.

*Fuente:* (García Santillán, 2008)

## b) Reducción del ruido

- **Apertura:**

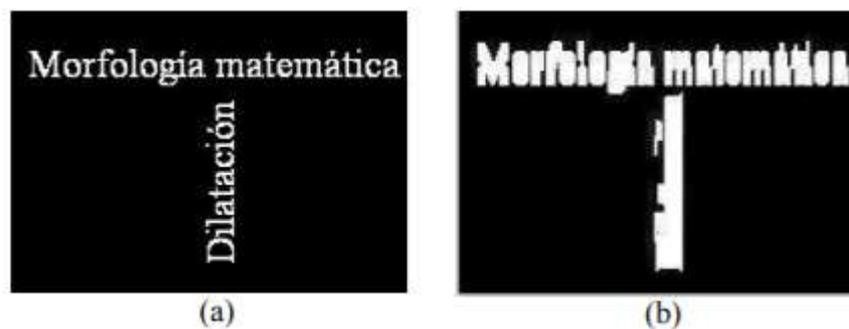
Erosión seguida de una dilatación. Elimina pequeños píxeles aislados que haya en la Imagen.



**Figura 25:** Apertura de una imagen binaria; (a) Imagen Original; (b) Imagen Aperturada.

*Fuente:* (Garcia Santillán, 2008)

**La dilatación** expande los píxeles de la imagen sobre la que se aplica.



**Figura 26:** Dilatación de una imagen binaria; (a) Imagen Original; (b) Imagen Dilatada.

*Fuente:* (Garcia Santillán, 2008)

**La erosión** adelgaza la imagen sobre la que se aplica siendo, en un sentido no estricto, opuesta a la dilatación.

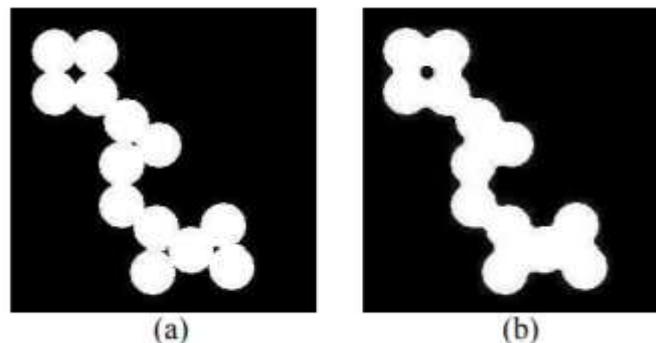


**Figura 27:** Erosión de una imagen binaria; (a) Imagen Original; (b) Imagen Erosionada.

*Fuente:* (García Santillán, 2008)

- **Cierre:**

Dilatación seguida de una erosión. Rellena los pequeños agujeros que existan en la imagen.



**Figura 28:** Cierre de una imagen binaria; (a) Imagen Original; (b) Imagen Cerrada.

*Fuente:* (García Santillán, 2008)

- **Filtro Paso Bajo Gaussiano:**

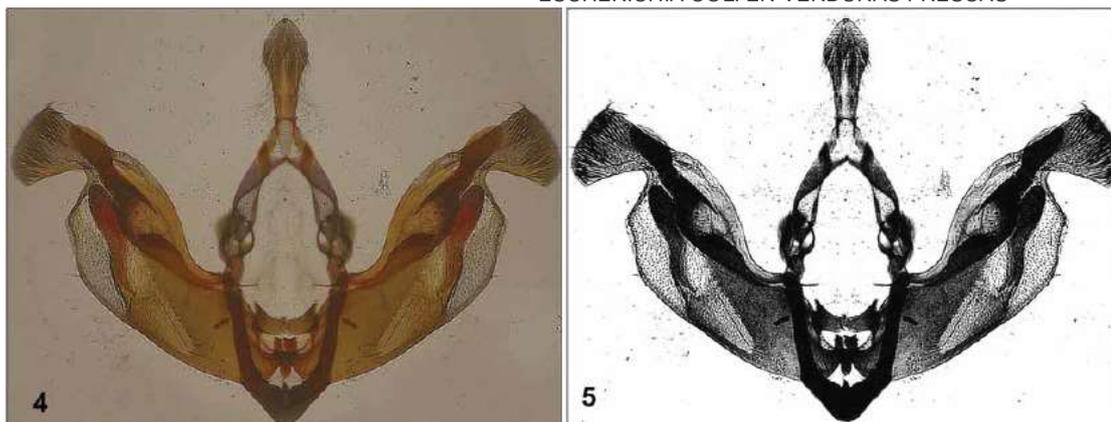
El filtrado lineal de imágenes mediante paso bajo gaussiano es una técnica común para atenuar frecuencias espaciales altas, pero tiene la desventaja de difuminar los contornos de los objetos en una imagen nítida. Esto puede destruir los detalles de la imagen necesarios para reconocer las pequeñas regiones de interés.

El peso se define por la función de Gauss de desviación estándar. Esto significa que la fuerza de la influencia depende de la distancia espacial entre los píxeles y no de sus valores.

La idea de suavizado de Gauss es utilizar esta distribución 2-D como una función dispersora de puntos mediante la convolución. Como resultado de ello, los bordes de los objetos en la imagen son borrosos porque los píxeles a través de las discontinuidades se promedian. En otras palabras, la acción de la convolución gaussiana es independiente del contenido de la imagen, y la influencia que un píxel tiene sobre otro depende sólo de la distancia entre estos en la imagen. (Rodríguez Oliva, 2014)

**c) Binarización:**

Es una técnica que consiste en la realización de un barrido en la matriz de la imagen digital, por medio de bucles o recursividad, con el fin de que el proceso produzca la reducción de la escala de grises a dos únicos valores. Negro (= 0) y blanco (= 255), o lo que es lo mismo, un sistema binario de ausencia y presencia de color 0-1. La comparación de cada píxel de la imagen viene determinada por el umbral de sensibilidad (valor  $T = \text{Threshold}$ ). Por ejemplo, los valores que sean mayores que el umbral toman un valor 255 (blanco) y los menores 0 (negro).



*Figura 29:* Binarización

*Fuente:* (Magro, 2013)

Fotografía de 16,8 millones de colores (24 bits, 4608 x 3456 píxeles) del andropigio de un lepidóptero, tomada a propósito con poco contraste y baja luminosidad, para observar su binarización de umbral medio. Abajo, la imagen resultante se ha escalado a 1000 x 750 píxeles, se muestra en B/N y es binaria, es decir, sólo tiene dos colores virtuales, blanco ( $255 = 1$ ) y negro ( $0 = 0$ ) (1 bit). (Magro, 2013)

### 3) Segmentación

Es el proceso que divide una imagen en objetos que sean de nuestro interés de estudio. Es una de las áreas más importantes y complejas de la visión artificial, la cual ha sido estudiada extensamente y continúa siendo tema de discusión.

#### a) Segmentación Basada en Umbral:

La umbralización es un proceso que permite convertir una imagen de niveles de gris o en color en una imagen binaria, de tal forma que los objetos de interés se etiqueten con un valor distinto de los píxeles del fondo.

La umbralización es una técnica de segmentación rápida, que tiene un coste computacional bajo y que incluso puede ser realizada en tiempo real durante la captura de la imagen usando un computador personal.

En la figura 30 se muestra la segmentación basada en umbralización de la placa de un automóvil, utilizando un umbral de 128 para el cálculo.



**Figura 30:** Segmentación por umbralización de la placa de un automóvil (a) imagen original; (b) Segmentación con umbral de 128.

*Fuente:* (Garcia Santillán, 2008)

#### b) Segmentación Basada en Regiones:

La segmentación por regiones es utilizada para separar los objetos de interés. En este caso, la imagen es particionada en diferentes regiones, quedándose cada una relacionada en ciertas características y conjuntos de píxeles conectados. Así, a partir de la segmentación de la imagen en regiones, pueden ser realizadas las medidas sobre cada región y las relaciones entre las regiones adyacentes pueden ser establecidas.

Sea  $R$  la región correspondiente a la imagen que se va a segmentar. Vamos a ver el proceso de segmentación como un proceso en el cual dividimos la región  $R$  en  $n$  subregiones  $R_1, R_2, \dots, R_n$ , tal que:

$$\bigcup_{i=1}^n R_i = R$$

La ecuación indica que la segmentación debe ser completa, es decir, que todo píxel debe estar en una región.

#### 4) Descripción

Es el proceso que obtiene características convenientes para diferenciar un tipo de objeto de otro, como: la forma, el tamaño, área, etc.

##### a) Descriptores de Frontera Según su Longitud y Diámetro:

La longitud de un contorno es uno de los descriptores más sencillos. Si se cuenta el número de pixels en el contorno se obtendrá una aproximación de su longitud. Si la frontera ha sido codificada usando cadenas de códigos, el número de componentes verticales más el de horizontales más el de diagonales multiplicados  $\sqrt{2}$  da la longitud exacta.

El diámetro de una frontera B se define como:

$$Diam(B) = \max_{i,j} [D(p_1, p_2)]$$

Donde D es una medida de distancia  $p_1$  y  $p_2$  son puntos de la frontera. Tanto el diámetro como la línea que une los dos puntos que definen el diámetro (llamada eje mayor de la frontera) son descriptores útiles. (Gonzales Marcos, y otros, 2006)

##### b) Descriptores de Región

Los descriptores de región se pueden obtener del análisis del propio contorno del objeto o por características internas del mismo.

- Medida de Área, para realizar la medida del área de un objeto, se utiliza el vector que define su contorno. El área se va a definir como el número de pixels que están dentro del contorno.

Uno de los métodos más rápidos, consiste en contar el número de pixels que hay en cada fila de puntos del objeto. (Gonzales Marcos, y otros, 2006)

Fórmula para calcular el área:

$$\text{Área} = \sum_{h=\min(filas)}^{\max(filas)} \max((columna(h)) - \min(columna(h)))$$

- Perímetro, existen dos formas de obtener el perímetro de un objeto.
  - ✓ Perímetro Convexo, se obtiene mediante el cálculo de diferentes diámetros de Feret con distintos ángulos. El resultado Obtenido es aproximado.
  - ✓ Perímetro Normal, se determina sumando los pixels del contorno y multiplicándolos por un factor. Los puntos horizontales y verticales tendrán un factor 1, mientras que los diagonales su factor es de  $\sqrt{1^2 + 1^2} = \sqrt{2}$  correspondiente a la longitud de la diagonal del pixel.
- Diámetros de Feret Horizontal y Vertical, corresponde al diámetro de la proyección horizontal y vertical del objeto.
- Ejes, corresponden a los ejes que atraviesan el centro de la gravedad del objeto. Generalmente se buscan ejes mayores y menores del objeto.
- Redondez, sirve para determinar la redondez de objetos redondos. Un objeto redondo perfecto, tendrá una redondez de 1.0, mientras que los objetos rectos y alargados se acercarán al cero. Se obtiene la relación entre el área calculada y el área de un círculo:

$$\text{Redondez} = \frac{4 \cdot \pi \cdot \text{Area\_Objeto}}{\text{Perímetro}^2}$$

$$\text{Redondez} = \frac{\text{Perímetro}}{2 \cdot \pi \cdot \text{Radio\_Mínimo}}$$

$$\text{Redondez} = \frac{\text{Radio\_Mínimo}^2}{\text{Área\_Objeto}}$$

Donde el Radio\_Mínimo es el equivalente a la distancia polar mínima.

(Gonzales Marcos, y otros, 2006)

## 5) Reconocimiento e Interpretación

Es el proceso que identifica los objetos, por ejemplo: una llave, un tornillo, moneda, coche, etc.

### a) Adaptación por Correlación

Se basa en la comparación de la imagen a clasificar con una o varias imágenes patrón que caracterizan a cada clase. Utilizan medidas de similitud basadas en correlaciones.

A continuación, se presenta un ejemplo para la extracción de los caracteres de la placa de un vehículo y su reconocimiento automático. La placa está compuesta de tres letras en mayúsculas y tres números, como se muestra en la figura 31.

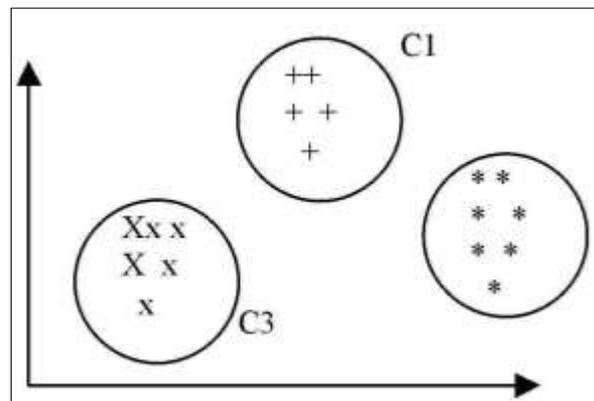


*Figura 31:* Extracción del primer carácter de la placa.

*Fuente:* (Garcia Santillán, 2008)

## b) Reconocimiento Estadístico de Patrones

Si representamos los objetos mediante una serie de parámetros globales, como vimos para 2-D y 3-D, podemos aplicar técnicas de reconocimiento estadístico de patrones. Estas consisten, básicamente, en buscar, dentro de un espacio paramétrico, la clase (modelo) más “cercana” a la descripción del objeto en la imagen. Si consideramos que se tienen, por ejemplo, dos parámetros y 3 tipos (clases) de objetos, el problema se puede visualizar como se representa en la figura 32. Cada punto en este representa un objeto. Los objetos similares se muestran con diferentes símbolos (x, +, \*), los cuales, normalmente, están agrupados en el espacio paramétrico. Dado un objeto desconocido, el problema es encontrar a que grupo (clase) pertenece dados sus parámetros. (Gómez & Sucar, 2008)



**Figura 32:** Espacio paramétrico con dos parámetros y tres clases. Cada círculo engloba objetos del mismo tipo.

*Fuente:* (Gómez & Sucar, 2008)

## c) Optimización Paramétrica

Las técnicas de optimización paramétrica se orientan a la correspondencia entre modelos paramétricos y representaciones de bajo nivel (por ejemplo, encontrar la correspondencia entre una serie de orillas y una curva). Los modelos se describen por un vector de parámetros  $\sim a = (a_1, a_2, \dots, a_N)$ .

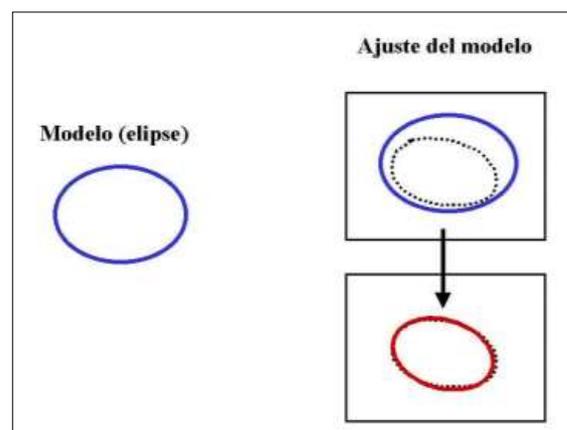
Se establece una función de mérito que mide que tan bien el modelo ( $\vec{a}$ ) describe a los atributos de la imagen. De forma que el reconocimiento se plantea como un problema de optimización, donde se busca maximizar la siguiente función:

$$M(\vec{a}, f(x, y))$$

Donde  $f(x, y)$  son los atributos obtenidos de la imagen. Si  $M$  es una función “bien comportada”, encontramos un máximo local.

Para encontrar este máximo se pueden usar diferentes tipos de técnicas:

- Técnicas analíticas, la función es simple y se puede encontrar el máximo analíticamente.
- Técnicas de gradiente (hill climbing) - se encuentra una solución aproximada que se va mejorando “moviéndose” en la dirección del gradiente.
- Perturbación de los coeficientes - si la derivada es difícil de obtener, se modifican ligeramente los coeficientes (partiendo de una solución inicial), en forma aleatoria o estructurada, y se mantienen si mejoran  $M$ .



**Figura 33:** Ejemplo de optimización paramétrica. El modelo de la elipse (izquierda) se ajusta a las características -orillas- obtenidas en la imagen (derecha).

En la figura 33 se ilustra el proceso en forma gráfica. En este caso el modelo es una elipse, la cual se “ajusta” al contorno obtenido de la imagen mediante la modificación de sus parámetros. (Gómez & Sucar, 2008)

### **Redes Neuronales**

Existen numerosas formas de definir redes neuronales; desde las definiciones cortas y genéricas hasta las que intentan explicar más detalladamente qué son las redes neuronales. Por ejemplo:

- a) Una nueva forma de computación, inspirada en modelos biológicos.
- b) Un modelo matemático compuesto por un gran número de elementos procesales organizados en niveles.
- c) Un sistema de computación compuesto por un gran número de elementos simples, elementos de procesos muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas.
- d) Redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico. (Matich, 2001)

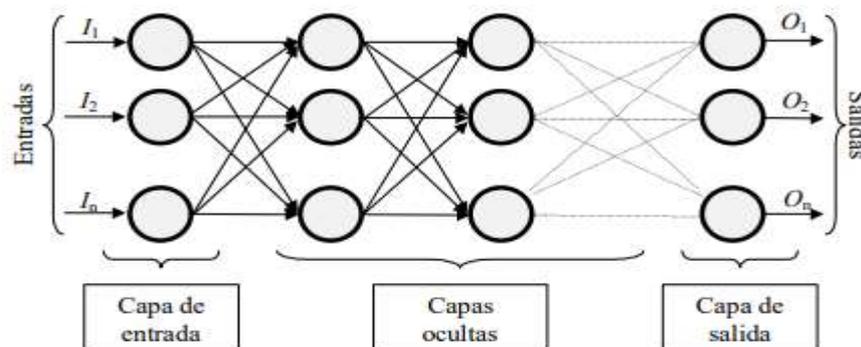
#### **1) Ventajas que ofrecen las redes neuronales**

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas

que representan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando en múltiples áreas. Entre las ventajas se incluyen:

- Aprendizaje Adaptativo. Capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.
  - Auto-organización. Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
  - Tolerancia a fallos. La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.
  - Operación en tiempo real. Los cálculos neuronales pueden ser realizados en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.
  - Fácil inserción dentro de la tecnología existente. Se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilitará la integración modular en los sistemas existentes.
- (Matich, 2001)

## 2) Elementos Básicos que componen una Red Neuronal



**Figura 34:** Red Neuronal

**Fuente:** (Matich, 2001)

La misma está constituida por neuronas interconectadas y arregladas en tres capas.

Los datos ingresan por medio de la “capa de entrada”, pasan a través de la “capa oculta” y salen por la “capa de salida”. Cabe mencionar que la capa oculta puede estar constituida por varias capas.

- a) **Capa de Entrada:** Es la capa que recibe directamente la información proveniente de las fuentes externas de la red.
- b) **Capa Ocultas:** Son internas a la red y no tienen contacto directo con el entorno exterior. El número de niveles ocultos puede estar entre cero y un número elevado. Las neuronas de las capas ocultas pueden estar interconectadas de distintas maneras, lo que determina, junto con su número, las distintas topologías de redes neuronales.
- c) **Capa de Salida:** Transfieren información de la red hacia el exterior. (Matich, 2001)

### 3) Mecanismos de Aprendizaje

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el mismo se reducen a la destrucción, modificación y creación de conexiones entre las neuronas.

Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, por lo tanto, se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables.

Siguiente mencionamos los dos métodos más importantes del aprendizaje:

- a) **Aprendizaje Supervisado:** Se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo

(supervisor, maestro) que determina la respuesta que debería generar la red a partir de una entrada determinada. El supervisor controla la salida de la red y en caso de que ésta no coincida con la deseada, se procederá a modificar los pesos de las conexiones, con el fin de conseguir que la salida obtenida se aproxime a la deseada.

- b) Aprendizaje No Supervisado:** Se caracteriza porque no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta.

Hay Muchos tipos de redes Neuronales, pero, para comprender este trabajo, interesa centrarse en las **Redes Neuronales Convolucionales**.

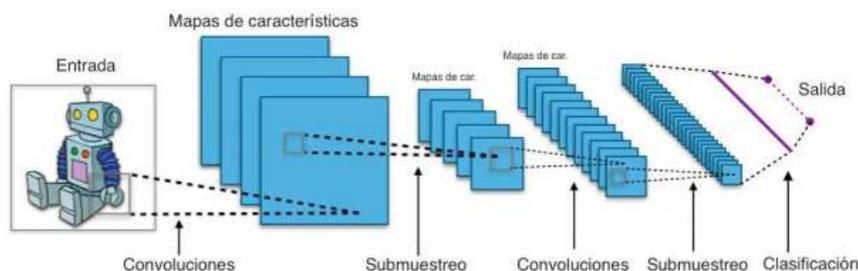
#### 4) **Redes Neuronales Convolucionales(CNN)**

Cuando se trabaja con imágenes, como en el caso de las redes neuronales convolucionales, se suele hacer con imágenes con resoluciones muy altas (por ejemplo, el standard actual es Full HD que consta de 1,980 x 1,080 píxeles). Esto conlleva un grave problema en las redes neuronales convencionales, que se han visto anteriormente. Éstas, al ser full-connected, tienen cada una de las neuronas que conforman la primera capa oculta conectada con todos y cada uno de los píxeles de la imagen de entrada a la vez. Si estas imágenes de entrada son de una resolución asumible, por ejemplo 28 x 28, el número de elementos que habrá que entrenar será lo suficientemente comedido como para que el sistema pueda funcionar correctamente, aunque todo el proceso será significativamente más lento. Sin embargo, si la resolución aumenta ligeramente (y no digamos ya a resoluciones actuales) los tiempos de entrenamiento y testeo se vuelven enormes.

Es un tipo de red multicapa que consta de diversas capas convolucionales y de pooling (submuestreo) alternadas, y al final tiene una serie de capas full-connected como una red perceptron multicapa. La entrada de una red capa convolucional suele ser, generalmente, una imagen  $m \times m \times r$ , donde  $m$  es tanto la altura como el ancho de la imagen y  $r$  es el número de canales.

Las capas convolucionales tienen  $K$  filtros (o kernels) cuyas dimensiones son  $n \times n \times q$ , donde  $n$  y  $q$  son elegidas por el diseñador (generalmente  $q$  suele ser igual a  $r$ ). Cada filtro genera mediante convolución un mapa de rasgos o características de tamaño  $(m-n+1) \times (m-n+1) \times p$ , siendo  $p$  el número de filtros que se desean usar. Después cada mapa es sub-muestreado en la capa de pooling con la operación “mean pooling” o “max pooling” sobre regiones contiguas de tamaño  $p \times p$  donde  $p$  puede tomar valores desde 2 para imágenes pequeñas hasta, comúnmente, no más de 5 para imágenes grandes. Antes o después del submuestreo, se aplica una función de activación sigmoïdal más un sesgo para cada mapa de rasgos. (Suárez, Redes Neuronales Convolucionales en R, 2017)

Lo mencionado se puede ver en la siguiente figura:



**Figura 35:** Esquema básico de una red neuronal convolucional

**Fuente:** (Suárez, Redes Neuronales Convolucionales en R, 2017)

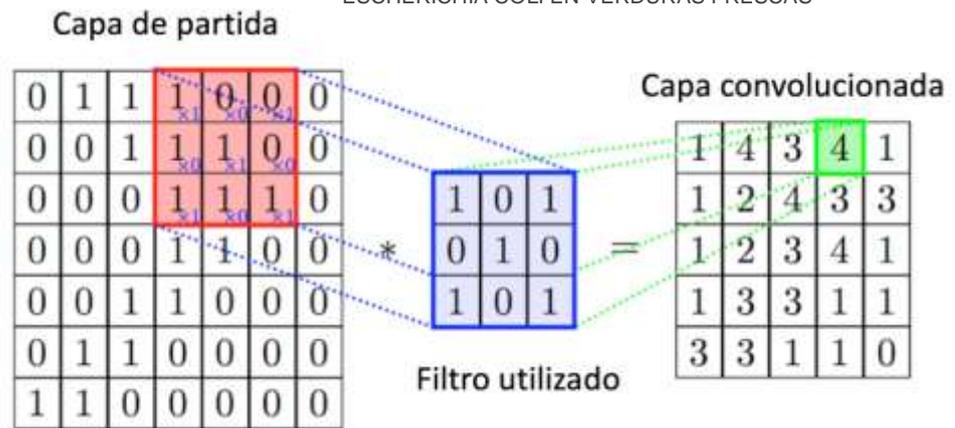
a) **Capa de Entrada:** La primera capa del esquema de una red convolucional se trata de la entrada a la red. Para este proyecto estas imágenes tendrán una entrada con dimensiones de 100 x 100 y estarán en escala grises.

b) **Capa Convolutiva:** Durante esta etapa de la red, se lleva a cabo una solución muy simple: se restringe el número de conexiones posibles entre las neuronas de la capa oculta y elementos de la imagen de entrada. De esta manera, cada neurona oculta solo estará conectada con un pequeño subconjunto de elementos de la imagen total.

Además, otra característica reseñable es que las imágenes naturales poseen la propiedad de ser “estacionarias”, esto quiere decir que las características o rasgos que existen en alguna parte determinada de la imagen pueden ser los mismos que otros que se encuentren en otra zona totalmente distinta

Por otro lado, la convolución es una operación de productos y sumas entre la imagen de entrada y un filtro que genera un mapa de características. La ventaja es que el mismo filtro (neurona) sirve para extraer el mismo rasgo en cualquier parte de la imagen, atendiendo al carácter estacionario de las imágenes que se ha comentado en el párrafo anterior.

En resumen, la capa convolutiva permite tanto reducir el número de elementos que conforman la red como detectar una serie de características que serán útiles a la hora de analizar la imagen. (Suárez, Redes Neuronales Convolucionales en R, 2017)

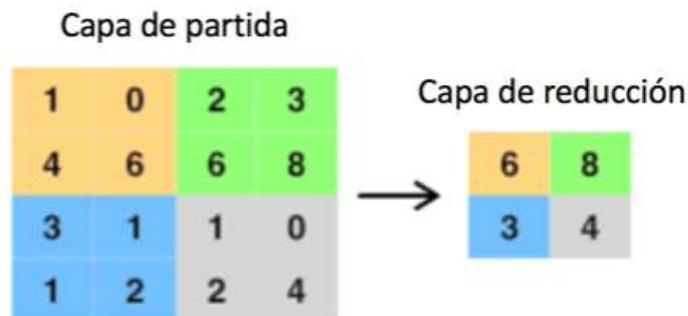


**Figura 36:** Operación Básica de Convolución

**Fuente:** (Suárez, Redes Neuronales Convolucionales en R, 2017)

c) **Capa de Pooling:** Después de haber obtenido las características en la capa de convolución, el siguiente paso en la lista es usarlas para la clasificación de las imágenes. En teoría, ya se podrá usar cualquier tipo de clasificador para llevar a cabo esta tarea.

El objetivo de esta capa es el disminuir aún más la carga computacional del sistema y, al mismo tiempo, ayudar con la caracterización de las imágenes obteniendo y localizando los rasgos predominantes de ella. (Suárez, Redes Neuronales Convolucionales en R, 2017)



**Figura 37:** Capa Pooling o Capa de Reducción

**Fuente:** (Calvo, 2017)

- d) Capa Full-Connected:** Esta se trata de la última capa del esquema de las redes neuronales convolucionales y se trata de un clasificador que determina a que clase pertenece la imagen de entrada, es decir, su trabajo en este proyecto es indicar que numero ‘cree’ la red que se le ha proporcionado a la entrada. (Suárez, Redes Neuronales Convolucionales en R, 2017)
- Está compuesta por un numero de neuronas que es igual al número de clases, en el caso de este proyecto son 2 neuronas.

### **Bacteria Escherichia Coli**

La Escherichia coli, también conocida como “E. coli”, es una bacteria que se encuentra comúnmente en el sistema digestivo de los seres humanos y animales de sangre caliente. Debido a su alta presencia en el intestino, la E. coli se utiliza como el indicador principal para detectar y medir la contaminación fecal en la evaluación de la inocuidad del agua y de los alimentos.

Consideradas comensales inofensivos, las cepas de E. coli constituyen alrededor del 1% de la población microbiana normal del intestino. Si bien la mayoría de las cepas dentro del intestino son agentes patógenos gastrointestinales beneficiosos para el ser humano, otros son perjudiciales.

Las E. coli patógenas se distinguen de otras E. coli por su capacidad de provocar graves enfermedades como resultado de su información genética para la producción de toxinas, capacidad de adhesión e invasión de células huéspedes, interferencia con el metabolismo celular y destrucción de tejidos. (Organización de las Naciones Unidas para la alimentación y agricultura, 2014)

**a) Condición de Supervivencia:**

Las cepas de E.coli verotoxigénica (ECVT) sobreviven durante meses en el estiércol contaminando las aguas superficiales (bebida y riego), las verduras y frutas y la superficie de las tierras de cultivo.

Estas bacterias se multiplican a temperaturas entre 6 y 50° C, con una temperatura óptima alrededor de 37° C. También, pueden crecer en presencia de un 6% de NaCl, ya que son más resistentes a estos compuestos que otras bacterias, como la Salmonella.

Para controlar el crecimiento hay que mantener los alimentos refrigerados y durante la congelación se inactiva. Son termorresistentes, pero se pueden eliminar con un tratamiento térmico a 65° C. (Elika, 2013)

**b) Vías de Transmisión**

Las bacterias E.coli verotoxigénicas pueden transmitirse al hombre a través de los alimentos por varias vías:

- **En origen:** en las explotaciones ganaderas por una inadecuada falta de higiene:
  - ✓ A través del contacto directo con animales o canales infectadas con E.coli.
  
  - ✓ Indirectamente a través de los alimentos de origen animal y del agua contaminados (contaminando a su vez a cultivos de vegetales). La presencia

de E.coli en los alimentos de origen animal es debida a contaminación de origen fecal.

- **En proceso por falta de higiene e inadecuada manipulación de los alimentos:**

- ✓ Contaminación cruzada en los mataderos y en las fases posteriores de transformación de los alimentos, y en la preparación y cocinado de los alimentos en el hogar.

- ✓ Personas: Los manipuladores de alimentos pueden ser portadoras de E.coli, de forma que al manipular los alimentos, sin tener en cuenta unas buenas prácticas de higiene, contaminan los alimentos.

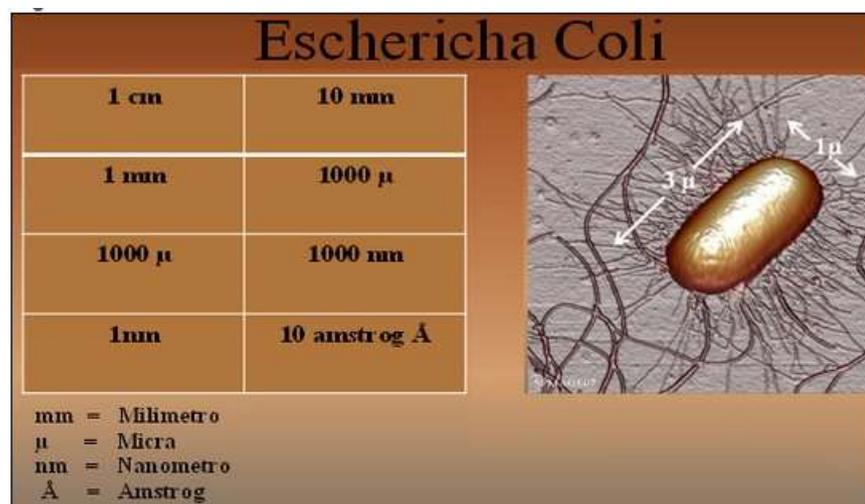
- ✓ Agua: El agua de riego puede estar contaminada con estiércol (que contiene E.coli procedente de las heces de los animales), transmitiéndose a las frutas y verduras frescas regadas con dicho agua.

También es muy importante la transmisión secundaria de persona a persona, sobre todo en el ámbito familiar, escolar y de centros de atención de personas mayores. (Elika, 2013)

### c) **Características E. Coli**

La EC, miden entre 1 hasta 3  $\mu$  (Figura 32), tiene capacidad móvil a través de flagelos (antígeno H) que son órganos filiformes que pueden medir varias micras, así mismo también tiene fimbrias (Antígeno F) que son órganos más pequeños, que a diferencia de los flagelos no poseen movilidad; pero por ser de naturaleza

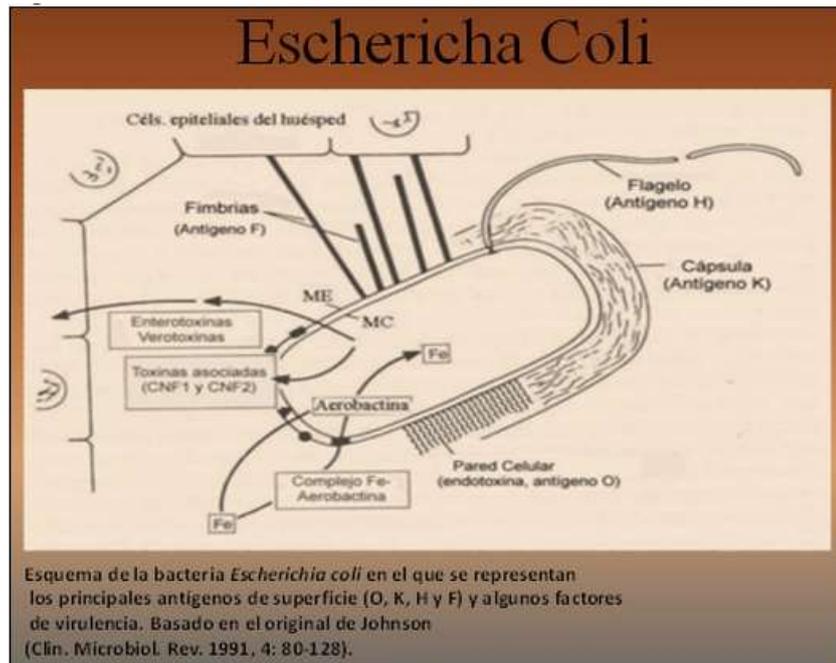
proteica (pilina), poseen propiedades antigénicas y hemoaglutinantes; También tiene importancia, pues median la ligadura de bacterias a las células a través de las adhesinas bacterianas; estas pueden medir entre 4 y 7 nm de diámetro y algunas micras de largo. En tanto que pili suele referir a los pelos ligeramente más largos que se utilizan en la conjugación bacteriana para intercambiar material genético desde la célula donadora hasta la receptora y a veces en el desplazamiento. (Carlos Vásquez MV, 2011)



**Figura 38:** Características del Escherichia Coli.

**Fuente:** (Carlos Vásquez MV, 2011)

Posee pared celular (antígeno O) que está conformada por Lipopolisacaridos (LPS), altamente antigénica y con capacidad de excretar endotoxinas, así mismo posee capsula (antígeno K) que le otorga algunas capacidades para evadir la fagocitosis y la acción de inmunidad primaria. (Carlos Vásquez MV, 2011)

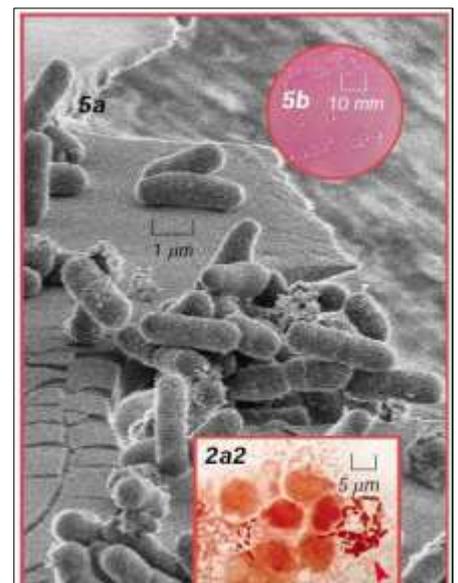


**Figura 39:** Esquema de la bacteria *Escherichia coli*.

**Fuente:** (Carlos Vásquez MV, 2011)

**d) Características Morfológicas y Tintoriales**

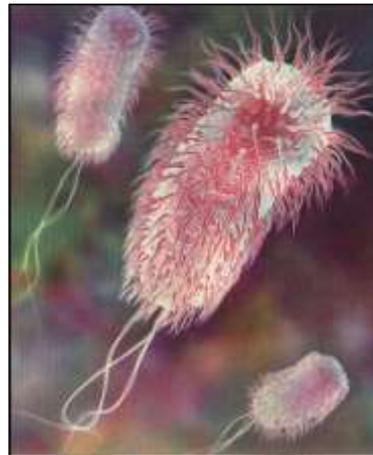
- Bacilo Gram negativo.
- No forma esporas
- Móviles (flagelos peritricos).
- Miden 0.5  $\mu$  de ancho por 3  $\mu$  de largo.
- Catalasa positivos.
- Oxidasa negativos.
- Reducen nitratos a nitritos.
- Producen vitamina B y K.



**Figura 40:** Características Morfológicas y tintoriales.

### e) Características Nutricionales

En agar MacConkey las colonias son rojas con halo turbio. (RAMIREZ, 2011)

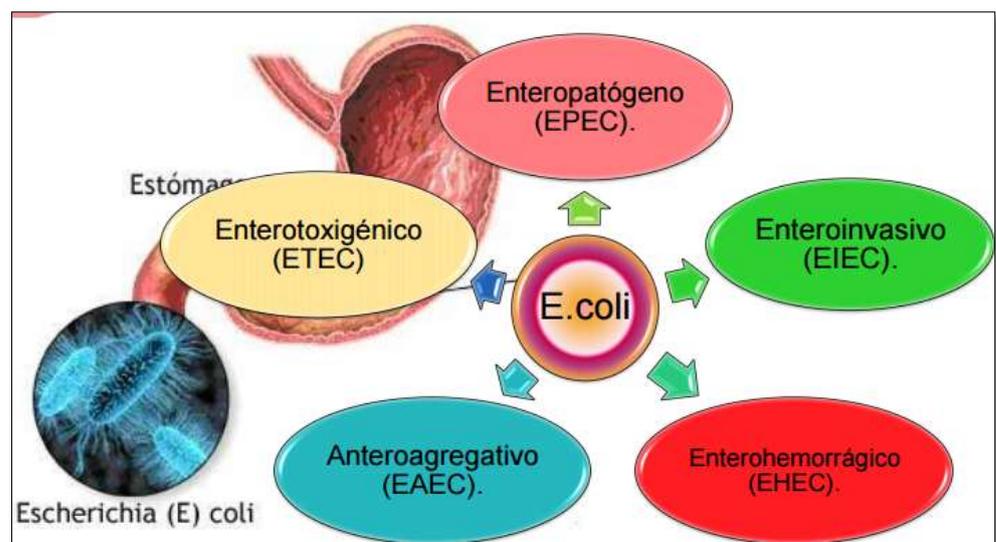


*Figura 41:* Características Nutricionales

*Fuente:* (RAMIREZ, 2011)

### f) Grupos E. Coli

Se han demostrado la existencia de clonas de E. Coli, en los cuales se reconocen 5 grupos.



*Figura 42:* Grupos E. Coli.

*Fuente:* (RAMIREZ, 2011)

- **E. coli enterotoxigénico (ETEC):** Entre 12-36 horas con ligera diarrea febril hasta síndrome grave parecido al cólera con heces acuosas sin sangre ni moco, dolores de estómago y vómito, deshidratación en niños. La enfermedad persiste 2-3 días.

*Tabla 9 E. coli enterotoxigénico (ETEC)*

Epidemiología	Cuadro clínico
<p>Las cepas ETEC son una causa frecuente de diarrea en lactantes de países en desarrollo, así como la causa más común de diarrea en individuos de países industrializados que viajan a zonas menos desarrolladas del mundo.</p>	<p>Presenta de 8 a 12 evacuaciones al día por un periodo de 4 a 5 días.</p> <p>Las cepas de ETEC son una causa importante de diarrea en niños menores de 5 años de edad y la causa más frecuencia de diarrea del viajero.</p>

---

*Fuente:* (RAMIREZ, 2011)

- **E. coli Enteropatógena (EPEC):** Malestar, vómitos, diarrea con deposiciones que contienen moco, pero no sangre o rara vez.

**Tabla 10** *E. coli* Enteropatógena (EPEC)

Epidemiología	Cuadro clínico
Se presenta como una enfermedad de niños menores de 2 años.	Afecta la mucosa intestinal. Perdida de disacaridasas.
En México, Brasil y África del sur entre el 30 y 40% de las diarreas son producidas por EPEC.	Produce diarrea secretora y se puede asociar con fiebre y si no se controla conduce a deshidratación y finalmente la muerte.

*Fuente:* (RAMIREZ, 2011)

- **E. coli enterohemorrágico (EHEC):** Productor de la verotoxina causante de **diarrea** y enfermedades que amenazan la vida como la colitis hemorrágica, el síndrome urémico hemolítico, la púrpura trombótica trombocitopénica.

**Tabla 11** *E. coli* enterohemorrágico (EHEC)

Epidemiología	Cuadro clínico
Se da en países de clima templado como USA, Canadá, Inglaterra, Argentina y Japón.	Colitis hemorrágica: Diarrea de inicio brusco con dolor abdominal.
Los serotipos involucrados más comúnmente son: O26, O111, O121, O145 y O157.	Las evacuaciones líquidas se acompañan de una descarga hemorrágica.  Síndrome urémico hemolítico.

*Fuente:* (RAMIREZ, 2011)

- **E. Coli Enteroagregativa(EAEC)**

*Tabla 12 E. Coli enteroagregativa (EAEC)*

Epidemiología	Cuadro clínico
<p>Se ha caracterizado por cuadros de diarrea persistente (dura más de 14 días) en niños, de diversas regiones geográficas, principalmente de países en desarrollo.</p>	<p>Diarrea secretora acuosa con moco y sangre, con fiebre en bajo grado.</p>

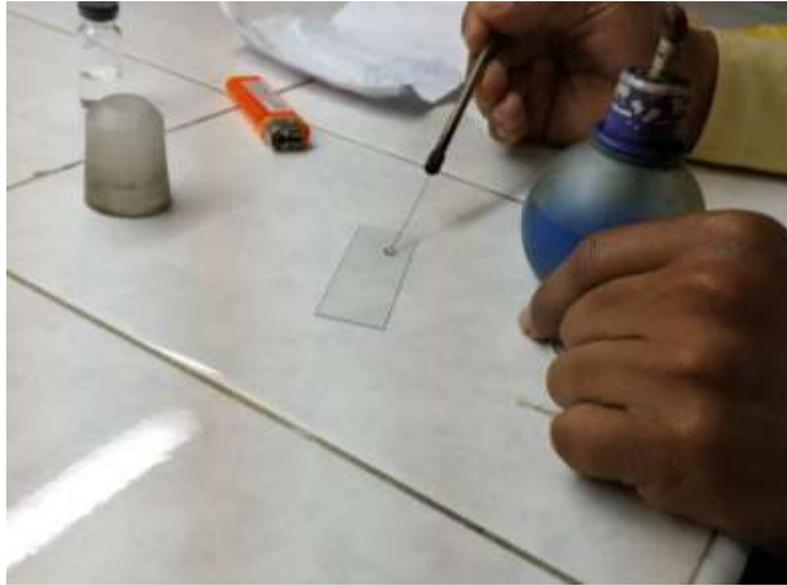
*Fuente:* (RAMIREZ, 2011)

### **Tinción Gram**

Es definida como una tinción diferencial, ya que utiliza dos colorantes y clasifica a las bacterias en dos grandes grupos: bacterias Gram negativas y bacterias Gram positivas. Fue desarrollada por el científico danés Hans Christian Gram en 1884; hoy en día, sigue siendo una de las tinciones más utilizadas universalmente debido a lo económico, sencillo y eficaz que resulta. (López Jácome, y otros, 2014)

Pasos:

- 1) Preparar la extensión del cultivo: En un portaobjetos limpio, libre de grasa, y con un asa bacteriológica colocar una pequeña cantidad de la colonia bacteriana



**Figura 43:** Colocación de Bacteria

**Fuente:** Elaboración Propia

- 2) Dejar secar al aire, luego fijar la preparación pasando la placa rápidamente por el mechero (2 ó 3 veces).
- 3) Agregar violeta cristal durante 20 a 30 segundos, el cual tiene afinidad con el peptidoglicano de la pared bacteriana (Colorante primario), enjuagar con agua seguidamente.



**Figura 44:** Violeta en Bacteria E. Coli

**Fuente:** Elaboración Propia

- 4) Agregar yodo gram durante 40 a 60 segundos que sirve como mordiente (compuesto químico intensificador del color) e impide la salida del cristal violeta por la formación de un complejo cristal violeta-yodo que satura los espacios de la pared bacteriana. Enjuagar con agua seguidamente.

- 5) Agregar alcohol acetona y enjuagar con agua inmediatamente (decoloración, disuelve lípidos de la pared de Gram negativas). Esta mezcla de alcohol-acetona, deshidrata la pared bacteriana y cierra los poros de la misma, también destruye la membrana externa de las bacterias Gram negativas debido a que ésta es soluble a la acción de solventes orgánicos, como la mezcla de alcoholacetona.
- 6) Agregar safranina durante 20 segundos, enjuagar con agua seguidamente. La safranina funciona como un colorante secundario o de contratinción y sirve para teñir las bacterias que no pudieron retener el complejo cristal violeta-yodo. (López Jácome, y otros, 2014)
- 7) Una vez logrado todos los pasos se procede a visualizar la bacteria en el microscopio.



*Figura 45:* Visualización de Bacteria

*Fuente:* Elaboración Propia

La presente investigación se justifica porque:

El presente proyecto pretende crear un Sistema Inteligente basado en algoritmos de visión computacional, realizando un análisis rápido de los resultados obtenidos para poder obtener un diagnóstico a la brevedad posible, permitiéndonos así detectar bacterias “E. Coli” ayudando a la sociedad para que se tomen las medidas pertinentes.

El presente proyecto brindará un aporte a estudios referentes a tecnologías, algoritmos, empleados en estudios de reconocimiento de bacterias, ya que se plasmarán las técnicas, algoritmos de procesamientos de imágenes, y resultados que se obtendrán para la detección de una bacteria E. Coli. Esto ayudará a que investigadores se motiven a trabajar con inteligencia artificial, ya sea en detección de bacterias y/o en otros proyectos que puedan aportar a nuestra sociedad.

Se encontró la siguiente limitación en el desarrollo del proyecto:

Una de las principales limitaciones es el acceso para poder obtener muestras de los laboratorios de dicha bacteria en verduras frescas.

### **Definición de Términos básicos**

E. Coli: Bacteria Escherichia Coli.

RNA: ácido ribonucleico

MINSA: Ministerio de Salud Del Perú

MODS: Susceptibilidad al fármaco de observación microscópica

NMP: Número más Probable.

PDI: Procesamiento Digital de Imagenes

RGB: red-green-blue

PCR: reacción en cadena de la polimerasa

TSI: Agar-hierro-triple azúcar

LIA: Lisina Hierro Agar

MIO: Movilidad, indol y ornitina

PET: Tomografía de Emisión de Positrones

CAT: Tomografía axial computarizada

MRI: Imagen por resonancia magnetica

TEM: microscopio de transmisión de electrones

ETEC: Escherichia Coli enterotoxigénico

EPEC: Escherichia coli Enteropatógena

EHEC: Escherichia coli enterohemorrágico

EAEC: Escherichia Coli enteroagregativa

Convolución: Es un operador matemático que transforma dos funciones  $f$  y  $g$  en una tercera función que en cierto sentido representa la magnitud en la que se superponen  $f$  y una versión trasladada e invertida de  $g$ .

IDE: Entorno de desarrollo integrado

#### **a. Formulación del problema**

Según las características descritas anteriormente podemos plantear el siguiente problema: ¿Cómo detectar bacterias Escherichia Coli en Verduras Frescas?

#### **b. Objetivos**

##### **i. Objetivo general**

Detectar bacterias Escherichia Coli en Verduras Frescas mediante un Sistema Inteligente basado en visión computacional.

**ii. Objetivos específicos**

- ✓ Conseguir un nivel de sensibilidad mayor o igual a 98.07%.
- ✓ Conseguir un nivel de especificidad mayor o igual a 95,4%.
- ✓ Detección de la bacteria Escherichia Coli en un 80% .

**c. Hipótesis**

Un Sistema Inteligente basado en visión computacional detecta bacterias Escherichia Coli en Verduras Frescas

## CAPÍTULO II. METODOLOGÍA

### 2.1. Tipo de investigación

El tipo de investigación es Experimental con un diseño Pre Experimental de un solo grupo con una sola medición, tomando en cuenta que se manipulará de manera discreta la variable independiente (Sistema Inteligente basado en visión computacional) sobre la variable dependiente (Detección de bacterias Escherichia Coli en verduras frescas).

$$G: X O_1$$

Donde:

G.: Grupo de Imágenes a Evaluar

$O_1$ : Detección de Bacterias.

X: Sistema Inteligente basado en visión computacional

### 2.2. Población y muestra (Materiales, instrumentos y métodos)

En la población se considera 315 imágenes de bacterias que son muestras de verduras en la ciudad de Trujillo.

Dichas imágenes fueron proporcionadas por el laboratorio de microbiología de la universidad nacional de Trujillo, estas imágenes tendrán una captura fija de resolución de 250 x 188 pixeles en formato JPG

Se conoce que la población es de 315 imágenes, por lo cual la población es finita.

Dónde:

$$n = \frac{z^2 \times p \times q \times N}{d^2(N - 1) + z^2 \times p \times q}$$

z: Nivel de confianza 95% = 1.96

p: prevalencia esperada del parámetro a evaluar, en caso de desconocerse ( $p = 0.5$ ), que hace mayor el tamaño muestral

q:  $1 - p$  (si  $p = 50\%$ ,  $q = 50\%$ )

d: precisión al 5% = 0.05

Reemplazando los valores:

$$n = \frac{1.96^2 \times 0.5 \times 0.5 \times 315}{0.05^2(315 - 1) + 1.96^2 \times 0.5 \times 0.5}$$

Por lo tanto, el tamaño de la muestra es de:

$n = 173,32 = 173$  imágenes.

### 2.3. Técnicas e instrumentos de recolección y análisis de datos

*Tabla 13 Tabla de Técnicas, Instrumento y procedimientos de recolección de datos.*

Variables	Dimensiones	Indicador	Técnica	Instrumento
Variable Independiente	Sensibilidad	Porcentaje de error del software.	Observación	Lista de Cotejo
	Especificidad	Porcentaje de falsos positivos	Observación	Lista de Cotejo
Variable Dependiente	Eficacia	Porcentaje de detección de bacterias	Observación	Lista de Cotejo

Fuente: Elaboración Propia

### 2.4. Procedimiento

Para la presente investigación se evaluaron las características de la experimentación, definiendo para ello que la técnica más adecuada para la recolección de los datos a analizar será la "Observación", para dicha técnica se diseñó como instrumento de recolección de datos una Lista de cotejo según los requerimientos de la técnica matriz de confusión que se utiliza para su posterior procesamiento.

Se seleccionaron verduras de mercados de la ciudad Trujillo y se llevaron al laboratorio de la Universidad Nacional de Trujillo para que el Microbiólogo Luis Llenque realice el respectivo análisis, donde el primer paso fue extraer muestras de las verduras para que pueda proceder con el cultivo de estas.

Luego de haber realizado el cultivo de estas, el Microbiólogo aisló colonias las cuales fueron procesadas por el método de Tinción Gram para poder identificar las bacterias. Obteniendo el análisis completo, el Microbiólogo procedió a obtener 315 imágenes.

Dichas imágenes fueron almacenadas en un USB y fueron proporcionadas a mi persona para su respectivo análisis.

Tendiendo las 315 imágenes que se convierte en nuestra población y procedimos a sacar nuestra muestra que son 173 imágenes, de las cuales para la presente tesis estamos utilizando una técnica de aprendizaje supervisada la cual requiere un conjunto de datos para entrenamiento y testeo, por lo cual se utilizaron el 80% para entrenamiento que incluye la validación y 20% para su testeo.

El 80% de las imágenes se utilizó para entrenar la red neuronal y enseñarle a determinar cuándo hay presencia de bacteria y cuando no. Por último, estamos realizando el testeo con el 20% de imágenes restantes equivalente a 35 imágenes por lo cual, se ingresó la imagen a la red neuronal y se esperó la salida, una vez obtenida procedemos a evaluar si la imagen procesada coincide con el resultado del Microbiólogo y se determina la situación de la imagen.

A continuación, se mostrará el cuadro de procedimiento de análisis de datos:

**Tabla 14** Procedimiento de Análisis de Datos

Dimensiones	Indicador	Método	Fórmula
Sensibilidad	Porcentaje de error del software.	Cualitativo	$S = \frac{VP}{VP + FN} \times 100$ <p>Donde: S: Sensibilidad VP: Verdaderos Positivos. FN: Falsos Negativos.</p>
Especificidad	Porcentaje de falsos positivos	Cualitativo	$ES = \frac{VN}{VN + FP} \times 100$ <p>Donde: ES: Especificad VN: Verdaderos Negativos. FP: Falsos Positivos.</p>
Eficacia	Porcentaje de detección de bacterias	Cuantitativo	$EF = \frac{A}{T} \times 100$ <p>Donde EF: Eficacia A: Cantidad de bacterias detectadas correctamente T: Cantidad total de imágenes a analizar</p>

*Fuente:* Elaboración Propia

## CAPÍTULO III. RESULTADOS

A continuación, se presentan los resultados obtenidos al hacer uso del sistema inteligente basado en visión computacional construido en base a los algoritmos determinados en el capítulo de desarrollo. Dichos algoritmos fueron implementados en Visual Studio Code utilizando el IDE Python Versión 3.6.7 y su ejecución se realizó en una computadora con sistema operativo Windows 10, procesador Intel Core i7-7500 CPU @ 2.70 GHz - 2.90 GHz, 12 GB de memoria RAM y una tarjeta de video Radeon 530 de 2 GB.

### 3.1. Análisis de Indicadores:

Para verificar que el sistema inteligente esté funcionando correctamente en la detección de bacterias se realizaron experimentos con 173 imágenes, de las cuales 60 imágenes se tomaron para el entrenamiento del software, 78 para su validación y 35 para validar el correcto funcionamiento. Estas imágenes fueron capturadas bajo la supervisión de un microbiólogo.

Los resultados de detección se clasificaron según los criterios de la matriz de confusión:

*Tabla 15 Matriz de Confusión*

Cantidades	Verdadero	Falso
Positivos	VP	FP
Negativos	VN	FN

*Fuente: Elaboración Propia*

#### **Verdadero Positivo (VP):**

El sistema inteligente reconoce que la imagen contiene Escherichia Coli.

#### **Verdadero Negativo (VN):**

El sistema inteligente reconoce que la imagen no contiene Escherichia Coli.

**Falso Positivo (FP):**

El sistema inteligente reconoce erróneamente la bacteria Escheria Coli, calificando como correcto o reconocida, aquellas imágenes que no contienen bacterias.

**Falso Negativo (FN):**

El sistema inteligente no reconoce la bacteria Escheria Coli, cuando la bacteria está presente en la imagen.

Los resultados de los procesos realizados a las 35 imágenes, obteniendo 27 verdaderos positivos, 8 verdaderos negativos, 0 falsos positivos y 0 falsos negativos.

*Tabla 16* Matriz de Confusión con datos

Cantidades	Verdadero	Falso
Positivos	27	0
Negativos	8	0

*Fuente: Elaboración Propia*

**3.2. Indicador 1: Sensibilidad**

Capacidad del Software para detectar correctamente si la imagen contiene Escherichia Coli.

$$VP = 27$$

$$FN = 0$$

$$S = \frac{VP}{VP + FN} \times 100 = 100 \%$$

El nivel de Sensibilidad es de 100 %.

### 3.3. Indicador 2: Especificidad

Capacidad del Software para reconocer que la imagen no contiene Escherichia Coli.

$$VN = 8$$

$$FP = 0$$

$$ES = \frac{VN}{VN + FP} \times 100 = 100 \%$$

El nivel de Especificidad es de 100 %.

### 3.4. Indicador 3: Eficacia

Cantidad de imágenes detectadas correctamente usando el software.

El software clasificó correctamente 26 imágenes de bacterias Escherichia Coli.

E= eficacia

A= Número de aciertos (VP + VN)

T= Total de imágenes clasificadas para Escherichia Coli.

$$EF = \frac{A}{T} * 100$$

El nivel de Eficacia es de 100 %

Ver **Anexo 02** para encontrar el origen del total de los valores de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos.

## CAPÍTULO IV. DISCUSIÓN Y CONCLUSIONES

### 4.1 Discusión

La presente investigación hemos podido observar según el indicador 1 que hemos obtenido un nivel de sensibilidad de 100%, el indicador 2 con un nivel de especificidad del 100% y en el indicador 3 un nivel de eficacia del 100%.

Los resultados obtenidos anteriormente fueron obtenidos para una muestra en un entorno controlado de la bacteria, la misma que tuvo que pasar por una fase de preparación, hasta la obtención de las muestras para luego poder ser procesadas e identificar si existe Escherichia Coli en la muestra o no.

De las investigaciones sobre nuestros antecedentes podemos especificar que la sensibilidad y especificidad que ambas son de 100% es una sensibilidad y especificidad que ha mejorado el antecedente de “Implementación de un Sistema de Telediagnóstico De Tuberculosis Y Determinación De Multidrogorresistencia Basada En El Método MODS En Trujillo, Perú”, en un 4.6% y 1.93%, posiblemente no sea muy significativo debido a que ellos obtuvieron 50 muestras y en la presente investigación se trabajó con más muestras. Es rescatable ya que trabajamos con redes neuronales convolucionales que son mucho más rápida que un modelo de algoritmo de reconocimiento de patrones.

Podemos especificar que la detección de la bacteria es de un 100%, es una detección que ha mejorado el antecedente de “Reconocimiento de Imágenes Utilizando Redes Neuronales Artificiales”, en un 20%, posiblemente no sea muy significativo debido que ellos tuvieron 25 muestras de las cuales obtuvieron 18 para entrenamiento y 7

para test, y la presente investigación se trabajó con 173 muestras de las cuales se obtuvieron 60 para entrenamiento, 78 para validación y 35 para test.

## 4.2 Conclusiones

Culminado el presente trabajo de investigación, podemos concluir que:

- Se logró desarrollar el sistema inteligente basado en visión computacional el cual permite identificar bacterias Escherichia Coli en Verduras Frescas.
- El sistema inteligente, fue capaz de detectar bacterias Escherichia Coli con una sensibilidad de 100%, superando el 98.07% planteado.
- Se logró obtener la especificidad, es decir el sistema es capaz de detectar verdaderos negativos en un 100%, superando el 95.4% planteado.
- Se logró determinar la eficacia del sistema, la cual tuvo un valor de 100%, superando el 80% planteado.

## REFERENCIAS

- Blanca Huapaya, J. H. (2001). PRIMER AISLAMIENTO DE Escherichia coli O157:H7 ENTEROHEMORRÁGICA EN EL PERÚ. *Revista Peruana de Medicina Experimental y Salud Pública*, 1-2.
- Calvo, D. (20 de Julio de 2017). *diegocalvo*. Obtenido de <http://www.diegocalvo.es/red-neuronal-convolucional-cnn/>
- Carlos Vásquez MV, M. (07 de 07 de 2011). *Engormix*. Obtenido de Engormix: <http://www.engormix.com/avicultura/articulos/escherichia-coli-patogeno-aviar-t28854.htm>
- Du Pont. (12 de MAYO de 2010). Du Pont Qualicon RiboPrinter System. *Du Pont*. Obtenido de AGENCIA IBEROAMERICANA PARA LA DIFUSIÓN DE LA CIENCIA Y LA TECNOLOGÍA.
- Elika. (2013). Escherichia Coli. *Elika - Fundación Vasca para la Seguridad Agroalimentaria*, 1-5.
- Garcia Santillán, I. D. (2008). *Visión Artificial y Procesamiento Digital de Imágenes usando Matlab*. Ibarra - Ecuador.
- Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Boston: Houghton Mifflin.
- Gómez , G., & Sucar, L. (2008). *Visión Computacional*. Puebla, México.
- Gonzales Marcos, A., Martínez de Pisón Ascacibar, F. J., Pernía Espinoza, A. V., Alba , E. F., Castejón, L. M., Ordieres, M. J., & Vergara, G. E. (2006). *TÉCNICAS Y*

ALORITMOS BÁSICOS DE VISIÓN ARTIFICIAL. España: UNIVERSIDAD DE LA  
RIOJA SERVICIO DE PUBLICACIONES.

López Jácome, L. E., Hernández Durán, M., Colín Castro, C. A., Ortega Peña, S., Cerón  
González, G., & Franco Cendejas, R. (2014). Las tinciones básicas en el laboratorio  
de microbiología. *Investigación en Discapacidad*, 10-18.

Magro, R. (2013). BINARIZACIÓN DE IMÁGENES DIGITALES Y SU ALGORITMIA  
COMO HERRAMIENTA APLICADA A LA ILUSTRACIÓN ENTOMOLÓGIC.  
*Boletín de la Sociedad Entomológica Aragonesa (S.E.A.)*, n° 53, 443–464.

Marimón, J. M. (01 de Enero de 2014). *Efe Salud*. Obtenido de Efe Salud:  
<http://www.efesalud.com/noticias/e-coli-la-bacteria-peligrosa/>

Marr, D. (1982). *Vision*. San Francisco: Freeman.

Matich, D. J. (2001). *Redes Neuronales: Conceptos Básicos y Aplicaciones*. Argentina.

Mejía Vilet, J. R. (2004). *Procesamiento Digital de Imágenes*. Mexico.

MINSA. (10 de JUNIO de 2011). BACTERIA E. COLI PODRÍA LLEGAR AL PERÚ. *LA  
REPUBLICA*.

Ordoñez Santiago, C. A. (2005). FORMATOS DE IMAGEN DIGITAL. *Revista Digital  
Universitaria Volumen 5*, 7-9.

Organización de las Naciones Unidas para la alimentación y agricultura. (2014). Prevención  
de la E. Coli en los Alimentos. *Organización de las Naciones Unidas para la  
Alimentacion y Agricultura*, 1 - 12.

*Organización Mundial de la Salud*. (30 de Abril de 2014). Obtenido de Organización  
Mundial de la Salud: <http://www.who.int/mediacentre/news/releases/2014/amr-report/es/>

Pablo, G. G. (2013). *Reconocimiento de Imágenes Utilizando Redes Neuronales Artificiales*.

Madrid: Universidad Complutense de Madrid.

Palomino Camargo, C., & González Muñoz, Y. (2014). TÉCNICAS MOLECULARES PARA LA DETECCIÓN E. *Rev Peru Med Exp Salud Publica*, 535-546.

RAMIREZ, A. R. (2011). *ESHERICHIA COLI*. Mexico: E.E. MICROBIOLOGIA GENERAL.

Rodriguez Oliva, K. J. (2014). *Suavizado mediante técnicas de filtrado morfológico en imágenes de microscopía celular*. Santa Clara: Universidad Central “Marta Abreu” de Las Villas.

Rodriguez, L., Alva, A., Coronel, J., Caviedes, L., Mendoza, A., Gilman, R., . . . Zimic, M. (2014). Implementación de un Sistema de Telediagnóstico de Tuberculosis y Determinación de Multidrogorresistencia Basada en el Método MODS en Trujillo, Perú. *Rev. Perú Exp. Salud Pública*, 445 - 453.

Suárez, J. D. (2017). *Redes Neuronales Convolucionales en R*. Sevilla.

## ANEXOS

### ANEXO N.º 01

#### DESARROLLO

En el presente capítulo, se realiza la descripción del desarrollo del sistema inteligente basado en visión computacional para detectar bacterias Escherichia Coli en Verduras Frescas.

Para la presente investigación nos vamos a basar en el ciclo de vida en cascada.

#### 1.1. Análisis:

Para la presente investigación se tienen el siguiente requerimiento:

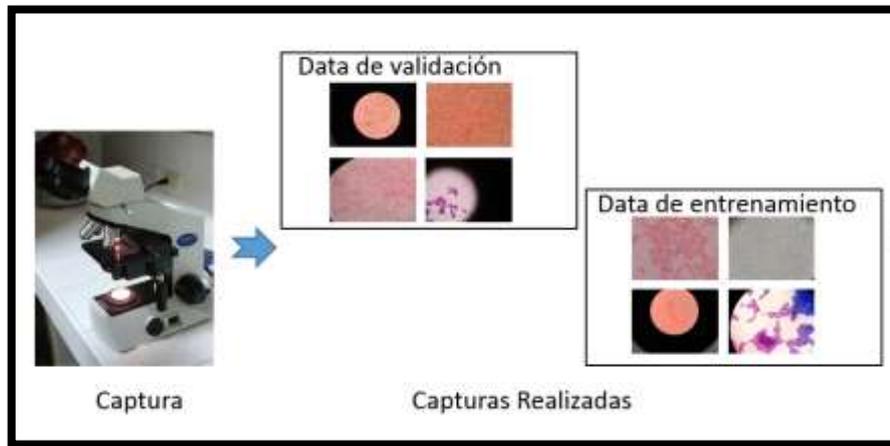
<b>Requerimiento Funcional</b>	Reconocer la bacteria Escheria Coli en muestras preparadas
--------------------------------	--

#### 1.2. Diseño

##### 1. Adquisición:

- Se toman las capturas de imágenes haciendo uso de un equipo móvil xiaomi Redmi note 5, que nos brinda una calidad de 12 megapíxeles.
- El enfoque a la bacteria se realiza mediante el microscopio Olympus CX 21, cuyo lente es de 100x que viene hacer el máximo acercamiento, por el ocular que fue de 10x.
- Por lo cual se obtuvo una visualización de la bacteria con 1000x de aumento.
- De dichos equipos mencionados se obtuvieron 173 imágenes, de las cuales procedemos a dividir en dos grupos:
  - Entrenamiento: 60 imágenes se destinaron para entrenamiento.

- Validación: 80 imágenes se destinaron para validación

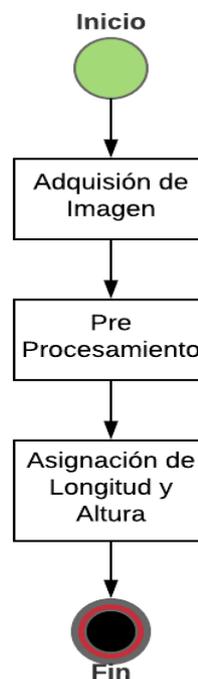


**Figura 46:** Esquema de Captura

*Fuente:* Elaboración Propia

## 2. Pre Procesamiento:

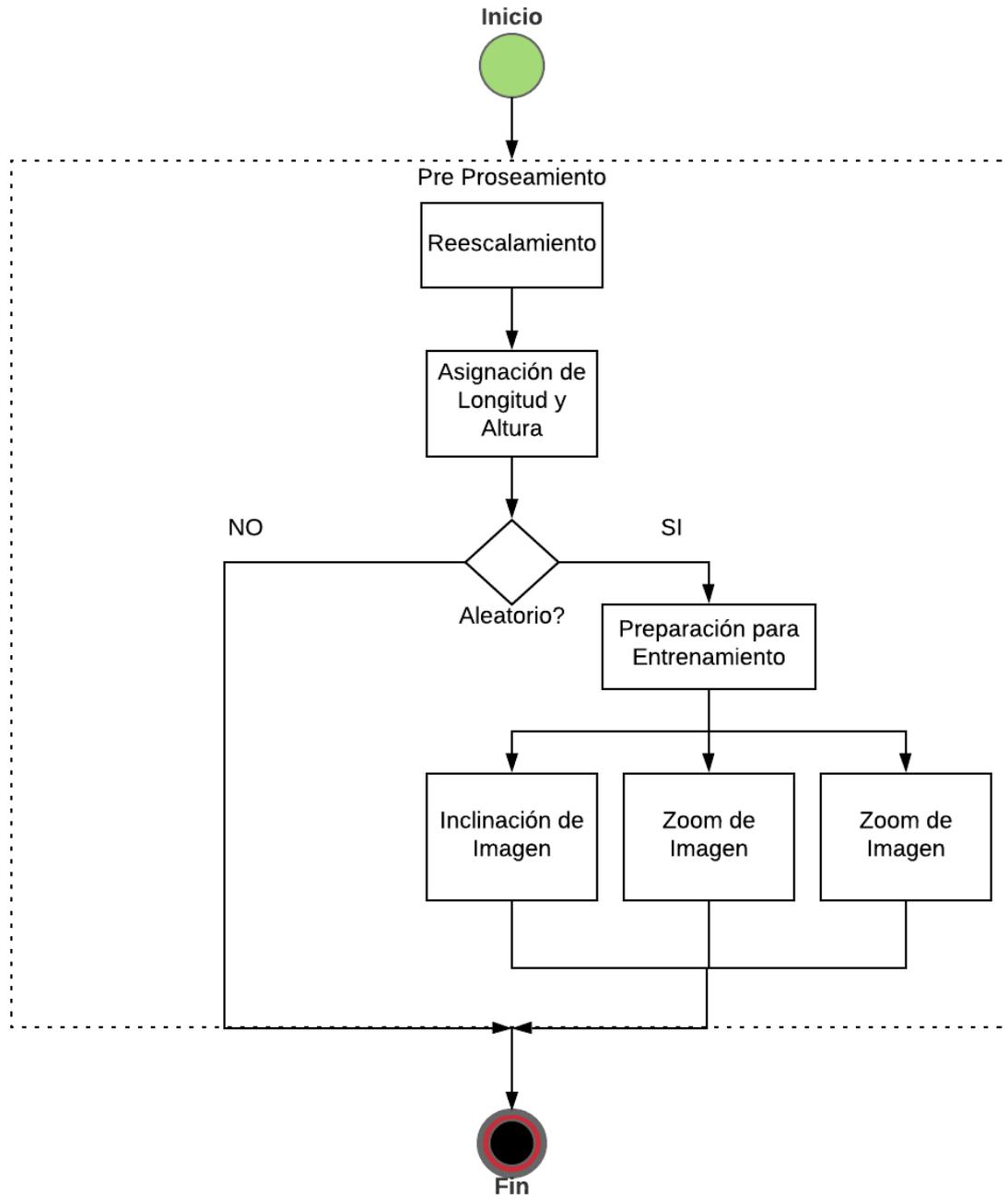
### 2.1. Procesamiento para imágenes



**Figura 47:** Diagrama de Flujo del Pre Procesamiento para una sola Imagen

*Fuente:* Propia

## 2.2. Procesamiento para imágenes de entrenamiento



**Figura 48:** Diagrama de Flujo del Pre Procesamiento Para varias imágenes

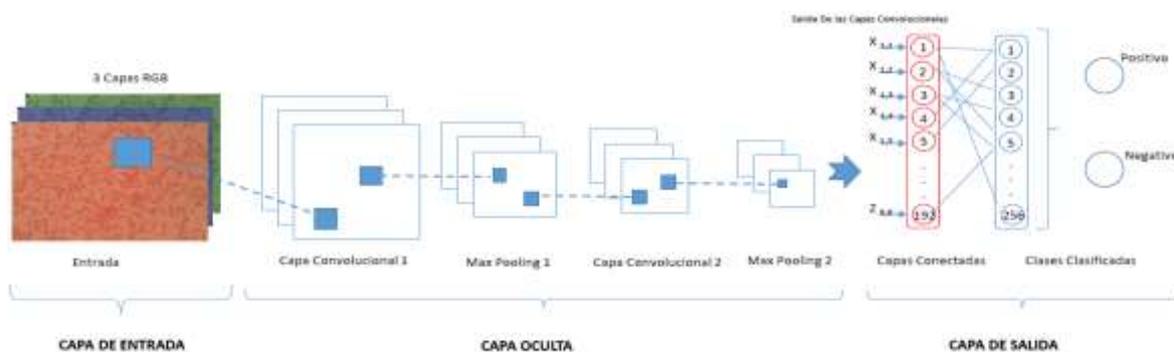
**Fuente:** Propia

### 3. Reconocimiento:

Capa de entrada: La capa de entrada está constituida por imágenes de 150 x 150 RGB.

Capa oculta: Las imágenes pasan por un proceso de convolución y maxpooling, dando como salida 192 neuronas, en imágenes de 8 x 8 RGB.

Capa de salida: En esta capa cada pixel se consideró como una neurona, con estas neuronas se podrá clasificar si es positivo o negativo.



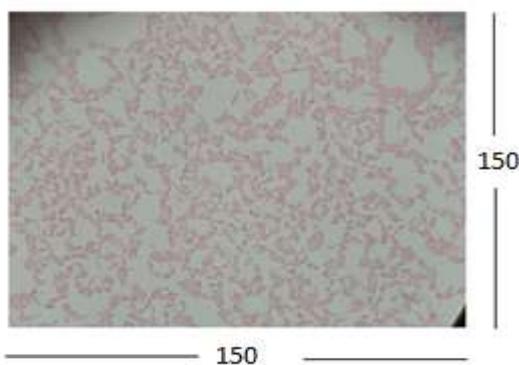
*Figura 49:* Diseño de la red neuronal convolucional

*Fuente:* Propia

### 1.3. Implementación

#### PreProcesamiento

Antes de pasar las imágenes a la red neuronal, primero se realiza un cambio de la altura y la longitud de la imagen de 150 x 150 píxeles.



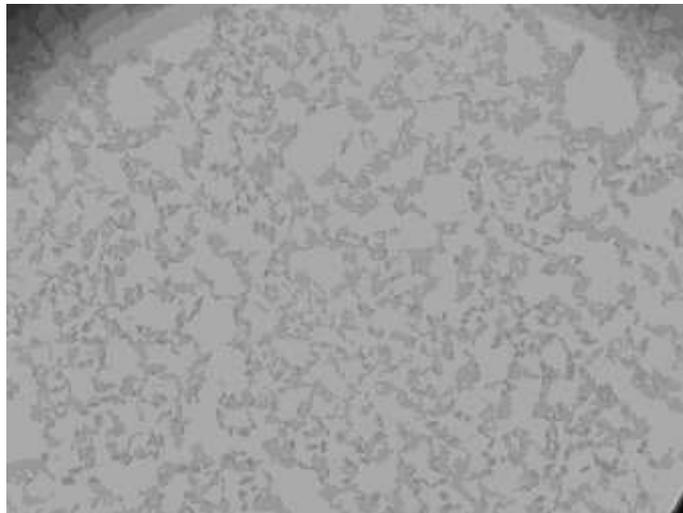
*Figura 50:* Imagen de 150 x 150

*Fuente:* Elaboración Propia

Para hacer la transformación de nuestras imágenes se crea un generador, consiste en lo siguiente:

Pre Procesamos nuestras imágenes de entrenamiento:

- Reescalamiento de imágenes a  $1. / 255$ , esto quiere decir que cada uno de las pixeles tiene un rango de 0 a 255 y al hacer este reescalamiento en vez de 1 a 255 estén de 0 a 1.



*Figura 51:* Imagen Reescalada a  $1. / 255$

*Fuente:* Elaboración Propia

- Luego, se toman imágenes aleatorias y se inclinan.
- Se toman imágenes aleatorias y se realiza un escalamiento.
- Por último tomamos imágenes aleatorias y las invertimos, para que nuestra red neuronal aprenda a distinguir direccionalidad.

```
entrenamiento_datagen = ImageDataGenerator(  
    rescale=1./255,      ## Reescalamos la imagen de 0 a 1  
    shear_range=0.3,    ## Genera imagenes inclinadas  
    zoom_range=0.3,     ## Toma imagenes y le hace zoom  
    horizontal_flip=True ## Toma imagenes y las invierte  
)
```

*Figura 52:* Pre Procesamiento de Imágenes de Entrenamiento

*Fuente:* Elaboración Propia

Pre Procesamiento de imágenes de validación:

- Para las imágenes de validación se realiza reescalamiento de 0 a 1.

```
validacion_datagen = ImageDataGenerator(rescale=1./ 255) ## Reescalamos la imagen de 0 a 1
```

*Figura 53:* Pre Procesamiento de Imágenes de Validación

*Fuente:* Elaboración Propia

A continuación, se explicara el envío de imágenes a entrenar y validar:

- Primero, enviamos la ruta donde se encuentran las imágenes.
- Asignamos altura y longitud de 150 x 150 a procesar.
- Brindamos el número de imágenes a procesar en cada uno de los pasos.
- Por último, asignamos una clase categorical.

```
entrenamiento_generador = entrenamiento_datagen.flow_from_directory( ## imagenes pre procesadas  
    data_entrenamiento,      ## ruta donde se encuentran las imagenes de entrenamiento  
    target_size=(altura, longitud), ## procesara toda las imagenes según la altura y longitud  
    batch_size=batch_size,    ## 32 imagenes a procesar  
    class_mode='categorical') ## categoria de imagenes  
  
validacion_generador = validacion_datagen.flow_from_directory( ## imagenes pre procesadas  
    data_validacion,        ## ruta donde se encuentran las imagenes de validación  
    target_size=(altura, longitud), ## procesara toda las imagenes según la altura y longitud  
    batch_size=batch_size,    ## 32 imagenes a procesar  
    class_mode='categorical') ## categoria de imagenes
```

*Figura 54:* Data de entrenamiento y validación

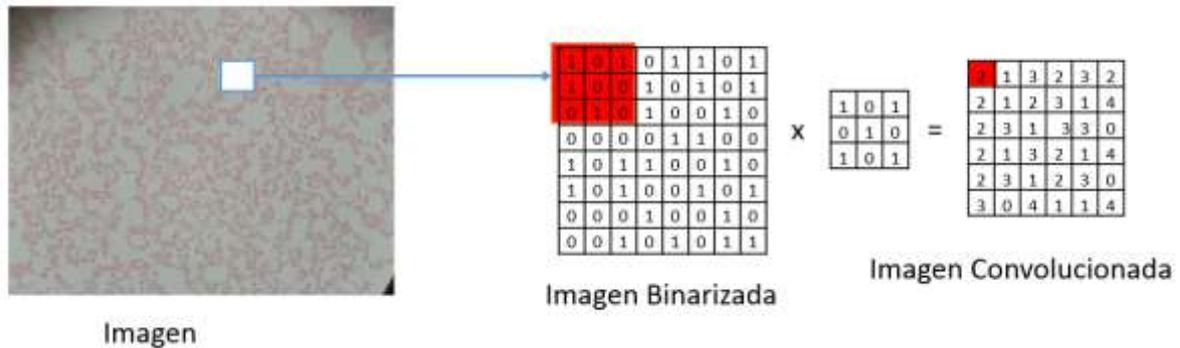
*Fuente:* Elaboración Propia

## Red Neuronal

Obteniendo la data de entrenamiento y validación Pre Procesadas, se procede a crear las siguientes capas:

- Primera Capa Convolutiva con Activación Relu y MaxPooling

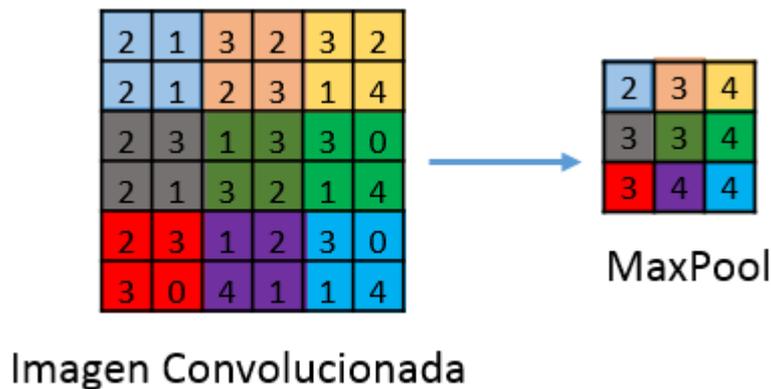
a) Nuestra capa convolutiva tendrá 32 filtros de 3 x 3.



**Figura 55:** Capa Convolutiva

**Fuente:** Elaboración Propia

b) Siguiendo, obteniendo el resultado de nuestra primera capa convolutiva, se realiza un pooling de 2 x 2



**Figura 56:** Operación MaxPooling

**Fuente:** Elaboración Propia

c) Seguidamente, se le asigna una activación relu, lo cual nos permite el paso de todos los valores positivos sin cambiarlos, pero se le asigna 0 a todos los valores negativos.

d) También asignamos padding “same”, donde esta función intenta rellenar uniformemente hacia la izquierda y hacia la derecha, pero si la cantidad de

columnas que se agregarán es impar, se agregará una columna adicional a la derecha. La misma lógica se aplica verticalmente

Lo descrito en los puntos anteriores, se define en la siguiente línea de código:

```
## Primera Capa Convolutional con activación Relu y Max-Pooling  
cnn.add(Convolution2D(filtrosConv1, tamaño_filtrol, padding ="same", input_shape=(altura, longitud, 3), activation='relu'))  
cnn.add(MaxPooling2D(pool_size=tamaño_pool))
```

**Figura 57:** Primera Capa Convolutional

**Fuente:** Elaboración Propia

- Segunda Capa Convolutional con Activación Relu y MaxPooling tendrá la misma configuración que la primera capa a diferencia, de que nuestra capa de convolución tendrá 64 filtros 2 x 2.

```
## Segunda Capa Convolutional con activación Relu y Max-Pooling  
cnn.add(Convolution2D(filtrosConv2, tamaño_filtro2, padding ="same", activation='relu'))  
cnn.add(MaxPooling2D(pool_size=tamaño_pool))
```

**Figura 58:** Segunda Capa Convolutional

**Fuente:** Elaboración Propia

- Se aplanar la salida de las capas convolucionales en una sola.
- Después de aplanar la información, se crea una capa con 256 neuronas y activación Relu.
- Se le apaga el 50% de neuronas aleatoriamente durante el enteramiento en cada paso, para evitar sobre ajustar los pesos, ya que si todo el tiempo dejamos nuestras neuronas prendidas al 100 % el sistema va aprender un camino en específico para clasificar, entonces, si le decimos que de las 256 neuronas apague el 50 % va aprender caminos alternos para clasificar nuestra data.
- Se crea otra capa Dense con el número de neuronas de acuerdo a nuestra clase, para este proyecto se va trabajar con 2 clases: Bacterias E. Coli Positivas y Bacterias E.

Coli Negativas. Esto va tener una activación Softmax, la cual nos ayuda a predecir la probabilidad de que sean positivas y negativas.

```
## Aplanamos la salida de las capas convolucionales
cnn.add(Flatten())

## Primera Capa completamente conectada con activación Relu
cnn.add(Dense(256, activation='relu')) ## se crea una capa con 256 neuronas
cnn.add(Dropout(0.5)) ## se va a apagar el 50% de las neuronas en cada paso

## Ultima capa con activación Softmax para la clasificación de las imagenes
cnn.add(Dense(clases, activation='softmax')) ## nos ayuda a predecir de acuerdo a cada clase
```

*Figura 59:* Ultimas Capas de una Red Neuronal Convolutcional

*Fuente:* Elaboración Propia

- Teniendo todas nuestras capas, se realiza lo siguiente para optimizar el algoritmo:
  - a) Durante el entrenamiento se verá su función de perdida.
  - b) La tasa de aprendizaje es de 0.0005, lo cual estará mejorando el porcentaje de aprendizaje.
  - c) Y por último el porcentaje de aprendizaje que nos estará indicando si se está clasificando bien nuestras imágenes.

```
## Parametros para optimizar nuestro algoritmo
cnn.compile(loss='categorical_crossentropy', ## funcion de perdida
            optimizer=optimizers.Adam(lr=lr), ## tasa de aprendizaje de 0.0005
            metrics=['accuracy']) ## porcentaje de aprendizaje
```

*Figura 60:* Parámetros para Optimizar el Algoritmo

*Fuente:* Elaboración Propia

- Se procede a entrenar la red neuronal con imágenes de entrenamiento pre procesadas, las cuales se va correr en 1000 pasos en 20 épocas. A la vez serán procesadas las imágenes de validación pre procesadas y para la validación al finalizar de cada época, se recorrerá 200 pasos de validación.

```

Entrenar Red Neuronal
"""
cnn.fit_generator(
    entrenamiento_generator,          ## imagenes de entrenamiento pre procesadas
    steps_per_epoch=pasos,          ## 1000 pasos
    epochs=epocas,                  ## 20 epocas
    validation_data=validacion_generator, ## imagenes de validación pre procesadas
    validation_steps=validation_steps) ## Pasos de validación que va recorrer en cada época

```

**Figura 61:** Entrenamiento Red Neuronal

*Fuente:* Elaboración Propia

- Ejecutamos el siguiente código para saber el valor que se le esta asignando a cada clase: `print(entrenamiento_generator.class_indices)`. Teniendo como resultado lo siguiente:
  - a) Escherichia Coli Positivo: se le asignó 0
  - b) Escherichia Coli Negativo: Se le asignó 1
- Siguiendo, se procede a guardar los pesos y modelos obtenidos por el entrenamiento, ver la siguiente imagen:

```

target_dir = './modelo/'          ## Guarda en la carpeta Modelo
if not os.path.exists(target_dir): ## Si no existe la carpeta, esta la crea
    os.mkdir(target_dir)
cnn.save('./modelo/modelo.h5')    ## Guarda los modelos del entrenamiento
cnn.save_weights('./modelo/pesos.h5') ## Guarda los pesos del entrenamiento

```

**Figura 62:** Generación de Pesos y Modelos del entrenamiento

*Fuente:* Elaboración Propia

Obteniendo nuestros pesos y modelos se procede a realizar las predicciones de reconocimiento:

- Para las predicciones se va a trabajar con librerías de keras.
- Asignamos a cada imagen una longitud y una altura de 150 x 150.
- Se ingresa la ruta del modelo y pesos generados en el entrenamiento.

```
longitud, altura = 150, 150
modelo = './modelo/modelo.h5'      ## ruta del archivo de modelos
pesos_modelo = './modelo/pesos.h5' ## ruta del archivo de pesos
cnn = load_model(modelo)           ## cargamos el modelo
cnn.load_weights(pesos_modelo)     ## cargamos los pesos del modelo
```

**Figura 63:** Parámetros de Predicción

**Fuente:** Elaboración Propia

- Teniendo los parámetros de predicción, se procede a realizar una función, donde recibe una imagen, la convierte a un arreglo, se le añade una dimensión extra en el eje 0, esto es para que la información se pueda procesar sin ningún problema. Después se ingresa los parámetros con las rutas de modelos y pesos, para realizar una predicción sobre la imagen dada y que nos devuelva un arreglo, la cual nos va devolver un arreglo de 2 dimensiones, por lo tanto, nos importa que solo nos retorne la primera dimensión. Obteniendo la primera dimensión, traeremos el valor más alto para poder identificar a que clase pertenece dicha imagen.

```
## realizamos una funcion
def predict(file):      ## recibe una image
    x = load_img(file, target_size=(longitud, altura)) ## cargamos la imagen
    x = img_to_array(x)      ## convertimos en un arreglo nuestra imagen
    x = np.expand_dims(x, axis=0) ## dimension extra en el eje 0
    array = cnn.predict(x)   ## realizamos una predicion de la imagen donde nos trae 2 dimensiones
    result = array[0]       ## retornamos la primera dimension
    answer = np.argmax(result) ## nos trae el indice del valor mas alto de resultado
    if answer == 0:
        print("pred: Se Detecto E. Coli")
    elif answer == 1:
        print("pred: No contiene E. Coli")

    return answer ## retornamos la prediccion
```

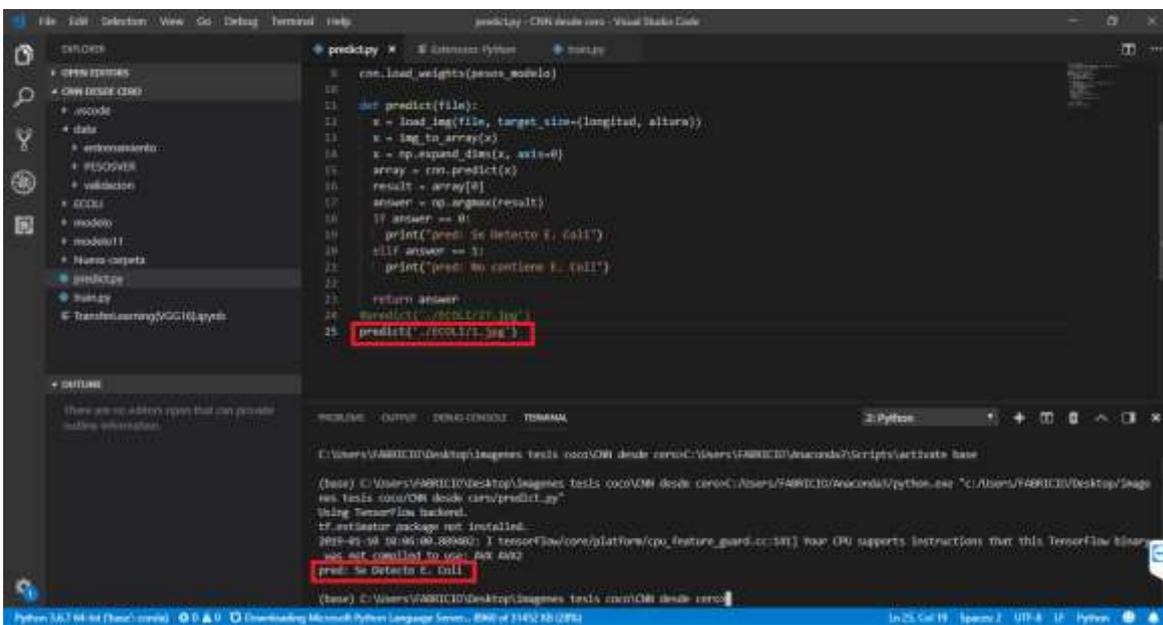
**Figura 64:** Función de Predicción de las clases

**Fuente:** Elaboración Propia

## 1.4. Pruebas

Para las pruebas del sistema inteligente se usó una computadora con sistema operativo Windows 10, procesador Intel Core i7-7500 CPU @ 2.70 GHz - 2.90 GHz, 12 GB de memoria RAM y una tarjeta de video Radeon 530 de 2 GB.

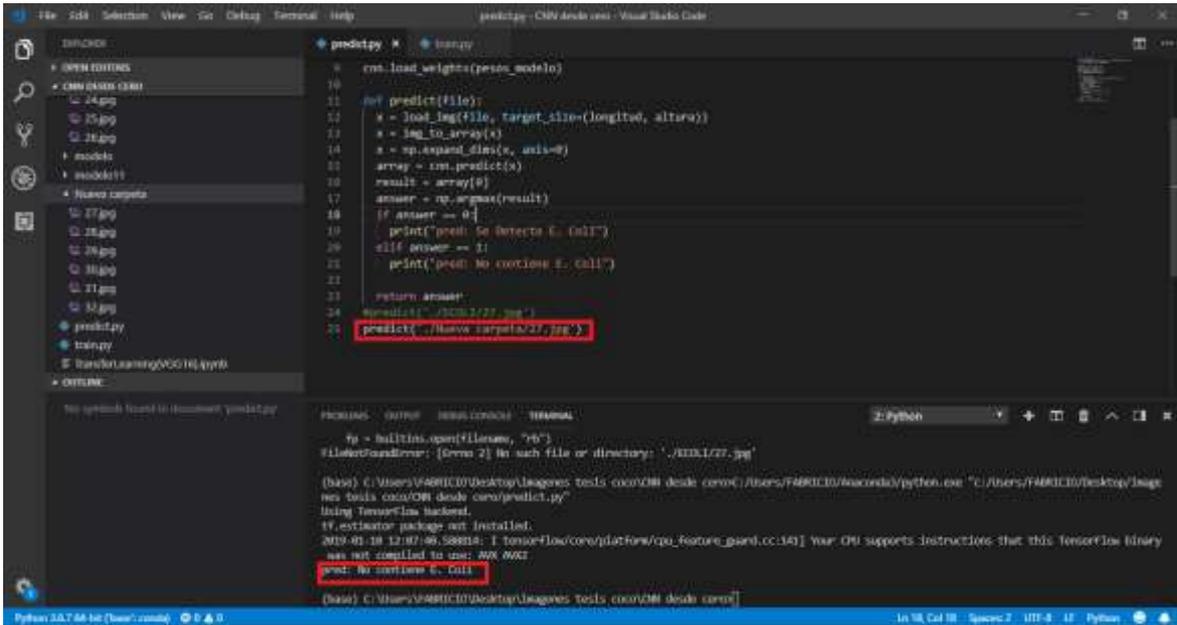
Se realizaron las pruebas necesarias para validar si el sistema está reconociendo correctamente:



```
File Edit Selection View Go Debug Terminal Help predict.py - CNN desde cero - Visual Studio Code
EXPLORER
+ OPEN SYSTEMS
+ OWAS DOWSE CIRO
+ .procode
+ data
+ entrenamiento:
+ PESOOVER
+ validacion
+ CCEI
+ modelo
+ model11
+ Nueva carpeta
+ predict.py
+ train.py
+ TransferLearningVGG16.pytest
+ DATASET
There are no editors open that can provide
additional information.
predict.py
9  cnn.load_weights(sesses_modelo)
10
11  def predict(file):
12      x = load_img(file, target_size=(imgitud, altura))
13      x = img_to_array(x)
14      x = np.expand_dims(x, axis=0)
15      array = cnn.predict(x)
16      result = array[0]
17      answer = np.argmax(result)
18      if answer == 0:
19          print("pred: Se Detecta E. coli")
20      elif answer == 1:
21          print("pred: No contiene E. coli")
22
23      return answer
24  predict("../COLI/27.jpg")
25  predict("../COLI/1.jpg")
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Python
C:\Users\FABRICEI\Desktop\imagenes tests coco\081 desde cero\C:\Users\FABRICEI\AppData\Local\Scripts\activate base
(base) C:\Users\FABRICEI\Desktop\imagenes tests coco\081 desde cero\C:\Users\FABRICEI\AppData\Local\Scripts\python.exe "c:\Users\FABRICEI\Desktop\image
tes tests coco\081 desde cero\predict.py"
Using TensorFlow backend.
tf.keras package not installed.
2019-05-08 08:05:00.899400: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary
was not compiled to use: AVX AVX2
pred: Se Detecta E. coli
(base) C:\Users\FABRICEI\Desktop\imagenes tests coco\081 desde cero
```

Figura 65: Prueba de Detección de E. Coli Img. 01

Fuente: Elaboración Propia



```

10  crn.load_weights(pesos_modelo)
11
12  def predict(file):
13      x = load_img(file, target_size=(longitud, altura))
14      x = img_to_array(x)
15      x = np.expand_dims(x, axis=0)
16      array = crn.predict(x)
17      result = array[0]
18      answer = np.argmax(result)
19      if answer == 0:
20          print("pred: Se Detecta E. Coli")
21      elif answer == 1:
22          print("pred: No contiene E. Coli")
23
24      return answer
25
26  predict('./Imagen carpeta/27.jpg')

```

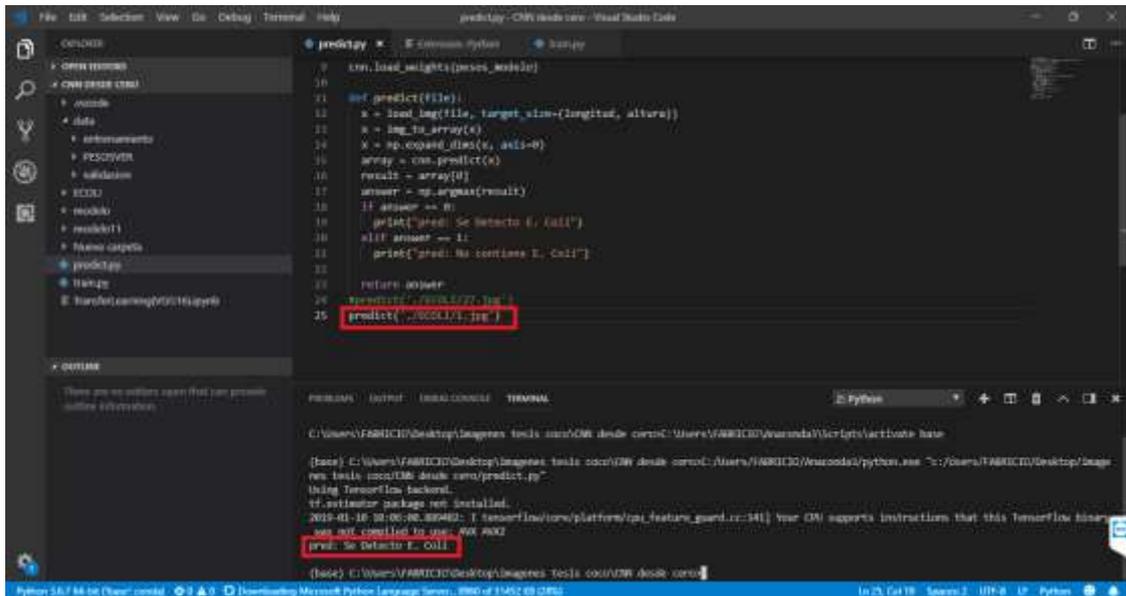
```

(base) C:\Users\FABRICIO\Desktop\imagenes tests coco\ON desde coco\C:\Users\FABRICIO\AppData\Local\Microsoft\WindowsApps\python.exe "C:\Users\FABRICIO\Desktop\imagenes tests coco\ON desde coco\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-18 12:07:46.588114: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
pred: No contiene E. Coli
(base) C:\Users\FABRICIO\Desktop\imagenes tests coco\ON desde coco>

```

Figura 66: Prueba de Detección Negativa Img. 27

Fuente: Elaboración Propia



```

7  crn.load_weights(pesos_modelo)
8
9
10
11  def predict(file):
12      x = load_img(file, target_size=(longitud, altura))
13      x = img_to_array(x)
14      x = np.expand_dims(x, axis=0)
15      array = crn.predict(x)
16      result = array[0]
17      answer = np.argmax(result)
18      if answer == 0:
19          print("pred: Se Detecta E. Coli")
20      elif answer == 1:
21          print("pred: No contiene E. Coli")
22
23      return answer
24
25  predict('./Imagen carpeta/01.jpg')

```

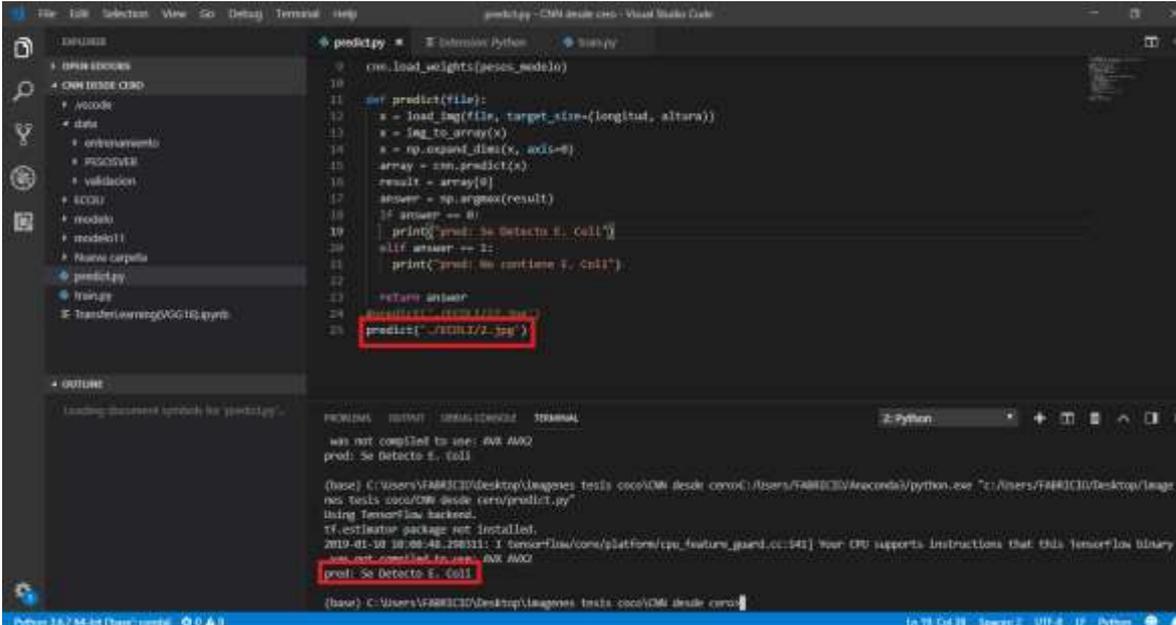
```

(base) C:\Users\FABRICIO\Desktop\imagenes tests coco\ON desde coco\C:\Users\FABRICIO\AppData\Local\Microsoft\WindowsApps\python.exe "C:\Users\FABRICIO\Desktop\imagenes tests coco\ON desde coco\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-18 12:08:02.898424: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli
(base) C:\Users\FABRICIO\Desktop\imagenes tests coco\ON desde coco>

```

Figura 67: Prueba de Detección de E. Coli Img. 01

Fuente: Elaboración Propia



```

File Edit Selection View Go Debug Terminal Help
predict.py - CHN desde cero - Visual Studio Code

EXPLORER
+ OPEN EXISTING
+ CHN desde cero
+ .vscode
+ data
+ entrenamiento
+ PESQUERA
+ validacion
+ ECOLI
+ modelo
+ modelo11
+ Nueva carpeta
+ predict.py
+ train.py
Transfer Learning VGG16.ipynb

OUTLINE
Loading document symbols for predict.py...

predict.py
10 cnn.load_weights(pesos_modelo)
11
12 def predict(file):
13     x = load_img(file, target_size=(longitud, altura))
14     x = img_to_array(x)
15     x = np.expand_dims(x, axis=0)
16     array = cnn.predict(x)
17     result = array[0]
18     answer = np.argmax(result)
19     if answer == 0:
20         print("pred: Se Detecta E. Coli")
21     elif answer == 1:
22         print("pred: no contiene E. Coli")
23
24     return answer
25
26 predict("../COLI/2.jpg")

TERMINAL
was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli

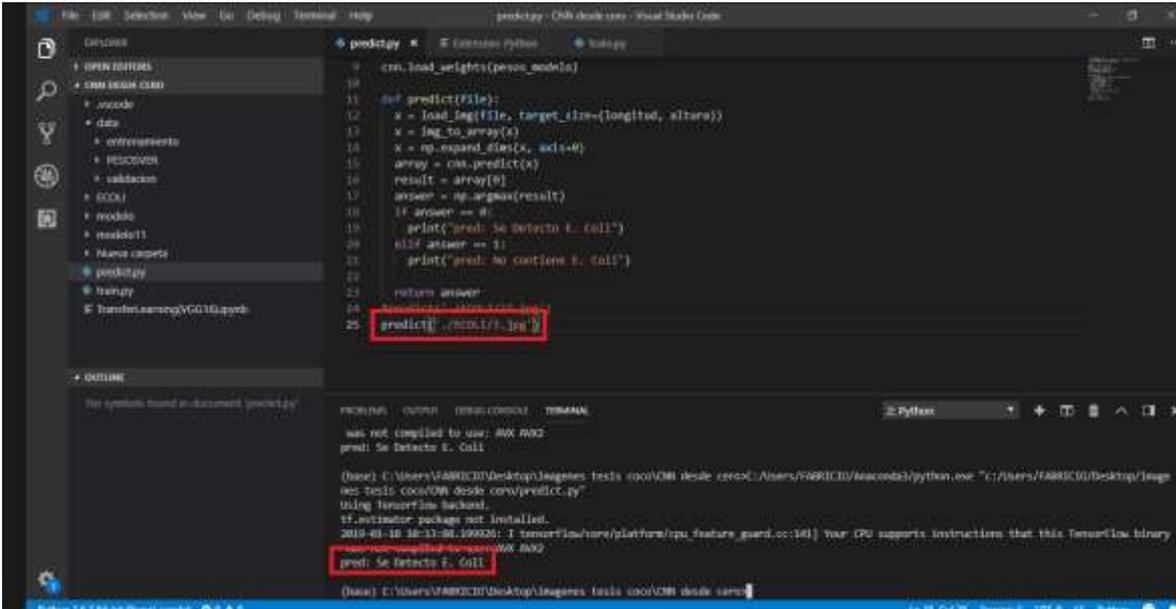
(base) C:\Users\FABRICIO\Desktop\imagenes tests coco\CHN desde cero C:\Users\FABRICIO\AppData\Local\Python\envs\test_coco\CHN desde cero\python.exe "C:\Users\FABRICIO\Desktop\imagenes tests coco\CHN desde cero\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-10 10:08:40.298111: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary
was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli

(base) C:\Users\FABRICIO\Desktop\imagenes tests coco\CHN desde cero
Python 3.6.7 64-bit [base\conda]

```

Figura 68: Prueba de Detección de E. Coli Img. 02

Fuente: Elaboración Propia



```

File Edit Selection View Go Debug Terminal Help
predict.py - CHN desde cero - Visual Studio Code

EXPLORER
+ OPEN EXISTING
+ CHN desde cero
+ .vscode
+ data
+ entrenamiento
+ PESQUERA
+ validacion
+ ECOLI
+ modelo
+ modelo11
+ Nueva carpeta
+ predict.py
+ train.py
Transfer Learning VGG16.ipynb

OUTLINE
No symbols found in document predict.py

predict.py
10 cnn.load_weights(pesos_modelo)
11
12 def predict(file):
13     x = load_img(file, target_size=(longitud, altura))
14     x = img_to_array(x)
15     x = np.expand_dims(x, axis=0)
16     array = cnn.predict(x)
17     result = array[0]
18     answer = np.argmax(result)
19     if answer == 0:
20         print("pred: Se Detecta E. Coli")
21     elif answer == 1:
22         print("pred: no contiene E. Coli")
23
24     return answer
25
26 predict("../COLI/3.jpg")

TERMINAL
was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli

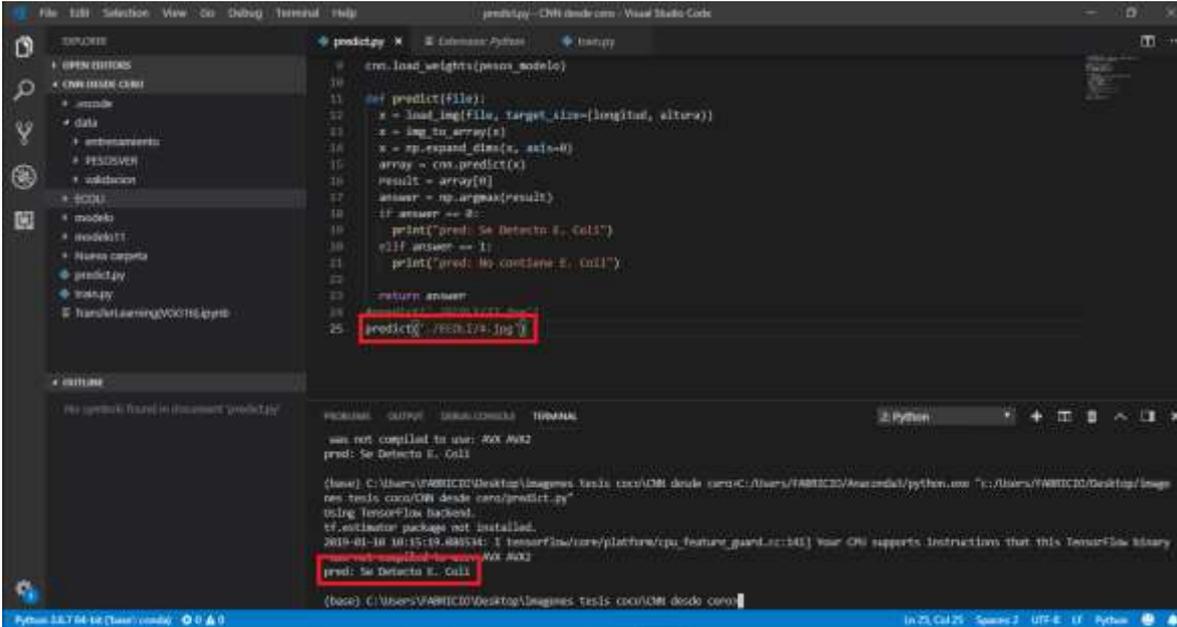
(base) C:\Users\FABRICIO\Desktop\imagenes tests coco\CHN desde cero C:\Users\FABRICIO\AppData\Local\Python\envs\test_coco\CHN desde cero\python.exe "C:\Users\FABRICIO\Desktop\imagenes tests coco\CHN desde cero\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-10 10:11:00.109026: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary
was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli

(base) C:\Users\FABRICIO\Desktop\imagenes tests coco\CHN desde cero
Python 3.6.7 64-bit [base\conda]

```

Figura 69: Prueba de Detección de E. Coli Img. 03

Fuente: Elaboración Propia



```

10
11 def predict(file):
12     x = load_img(file, target_size=(longitud, altura))
13     x = img_to_array(x)
14     x = np.expand_dims(x, axis=0)
15     array = cnn.predict(x)
16     result = array[0]
17     answer = np.argmax(result)
18     if answer == 0:
19         print("pred: Se Detecta E. Coli")
20     elif answer == 1:
21         print("pred: No contiene E. Coli")
22
23     return answer
24
25 predict('./Ecoli4.jpg')

```

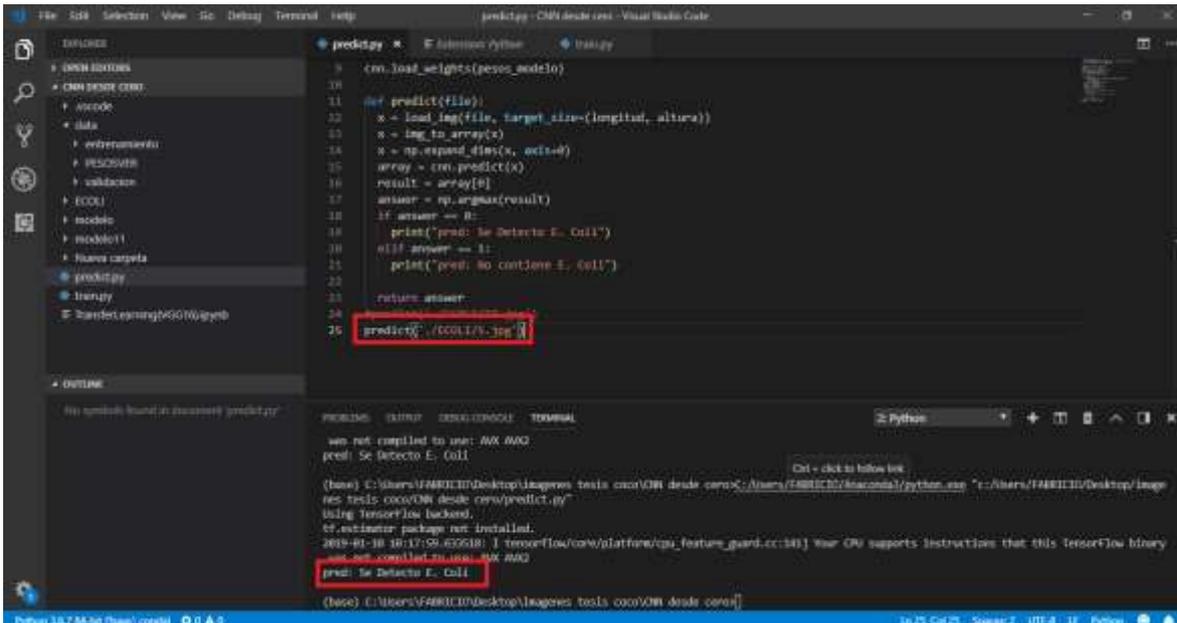
```

(base) C:\Users\FABRICIO\Desktop\imagenes tesis coc\OHN desde casa>python.exe "c:\Users\FABRICIO\Desktop\imagenes tesis coc\OHN desde casa\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-10 10:15:19.885344: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary
was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli

```

Figura 70: Prueba de Detección de E. Coli Img. 04

Fuente: Elaboración Propia



```

10
11 def predict(file):
12     x = load_img(file, target_size=(longitud, altura))
13     x = img_to_array(x)
14     x = np.expand_dims(x, axis=0)
15     array = cnn.predict(x)
16     result = array[0]
17     answer = np.argmax(result)
18     if answer == 0:
19         print("pred: Se Detecta E. Coli")
20     elif answer == 1:
21         print("pred: No contiene E. Coli")
22
23     return answer
24
25 predict('./Ecoli5.jpg')

```

```

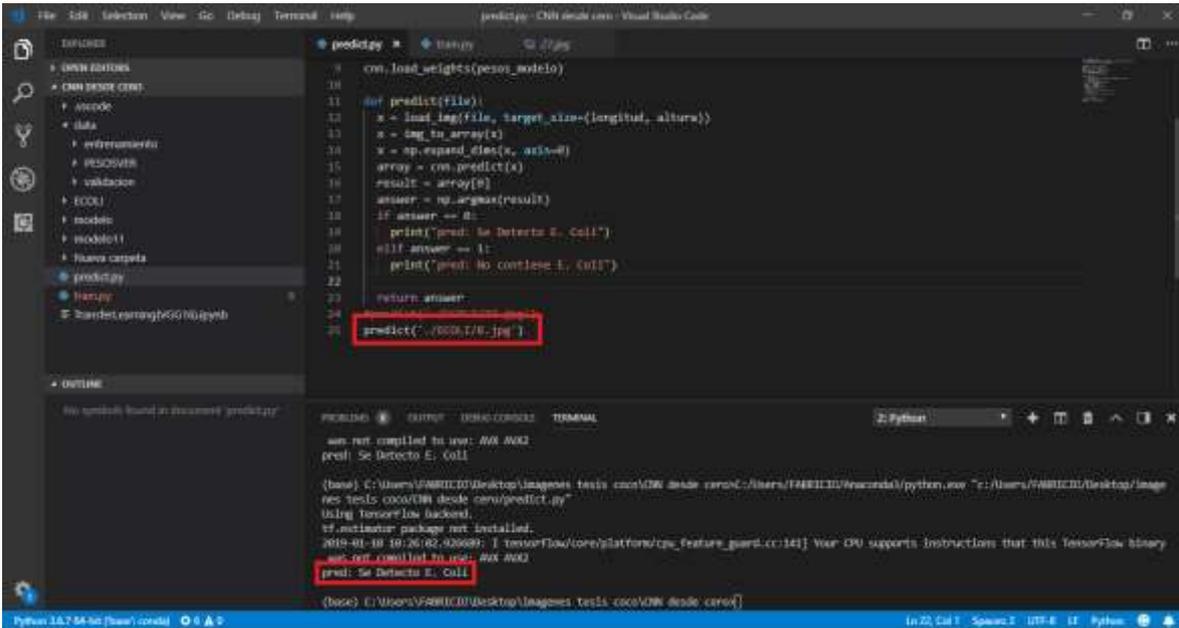
(base) C:\Users\FABRICIO\Desktop\imagenes tesis coc\OHN desde casa>python.exe "c:\Users\FABRICIO\Desktop\imagenes tesis coc\OHN desde casa\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-10 10:15:19.885344: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary
was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli

```

Figura 71: Prueba de Detección de E. Coli Img. 05

Fuente: Elaboración Propia





```

10  cm.load_weights(pesos_modelo)
11  def predict(file):
12      x = load_img(file, target_size=(longitud, altura))
13      x = img_to_array(x)
14      x = np.expand_dims(x, axis=0)
15      array = cm.predict(x)
16      result = array[0]
17      answer = np.argmax(result)
18      if answer == 0:
19          print("pred: Se Detecta E. Coli")
20      elif answer == 1:
21          print("pred: No contiene E. Coli")
22
23      return answer
24
25  predict('./0001/0.jpg')

```

```

was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli

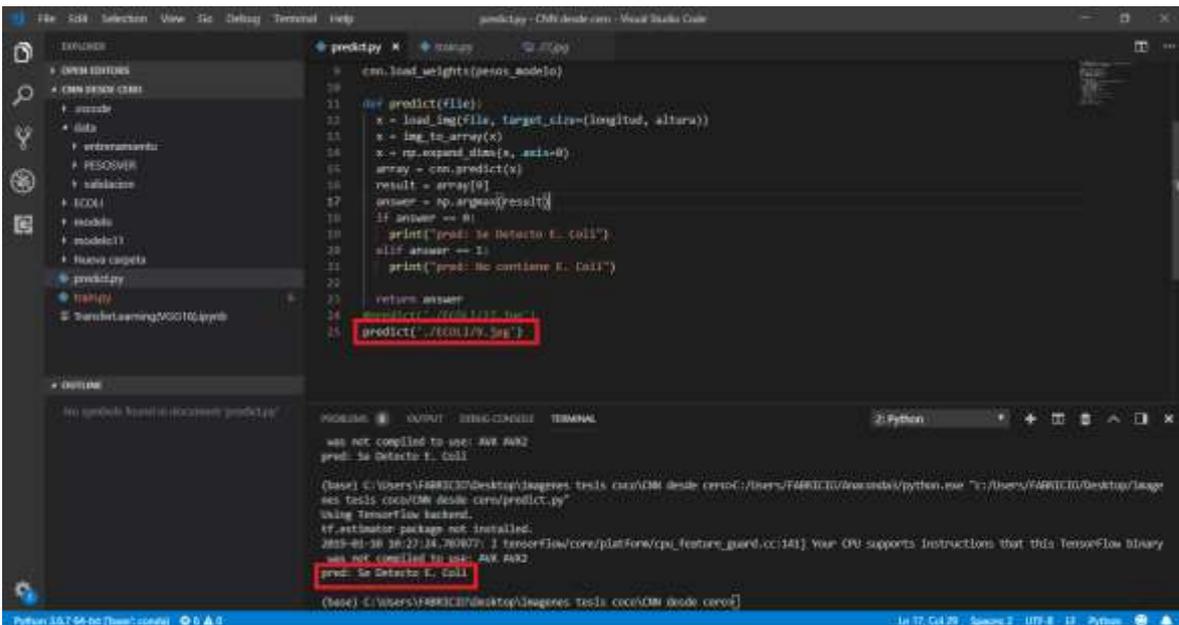
(base) C:\Users\FABRICIO\Desktop\imagenes tests cccc\DN desde cccc\C:\Users\FABRICIO\AppData\Local\python.exe "C:\Users\FABRICIO\Desktop\imagenes tests cccc\DN desde cccc\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-10 10:26:02.026899: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow library was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli

(base) C:\Users\FABRICIO\Desktop\imagenes tests cccc\DN desde cccc\

```

Figura 74: Prueba de Detección de E. Coli Img. 08

Fuente: Elaboración Propia



```

10  cm.load_weights(pesos_modelo)
11  def predict(file):
12      x = load_img(file, target_size=(longitud, altura))
13      x = img_to_array(x)
14      x = np.expand_dims(x, axis=0)
15      array = cm.predict(x)
16      result = array[0]
17      answer = np.argmax(result)
18      if answer == 0:
19          print("pred: Se Detecta E. Coli")
20      elif answer == 1:
21          print("pred: No contiene E. Coli")
22
23      return answer
24
25  predict('./001/9.jpg')

```

```

was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli

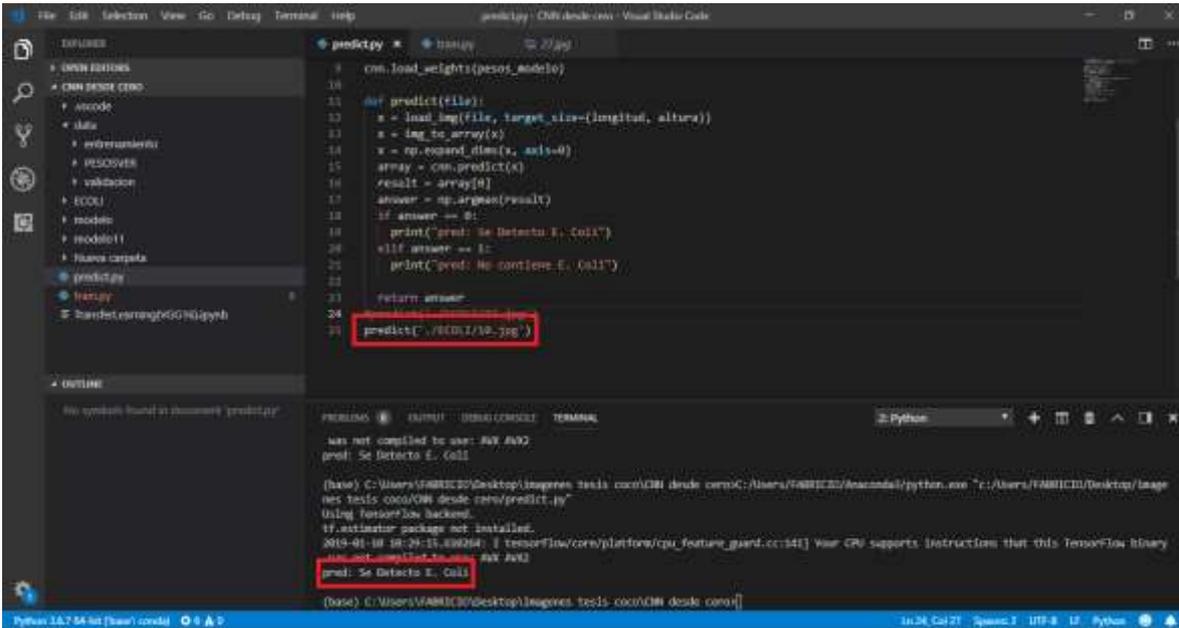
(base) C:\Users\FABRICIO\Desktop\imagenes tests cccc\DN desde cccc\C:\Users\FABRICIO\AppData\Local\python.exe "C:\Users\FABRICIO\Desktop\imagenes tests cccc\DN desde cccc\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-10 10:27:14.200072: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow library was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli

(base) C:\Users\FABRICIO\Desktop\imagenes tests cccc\DN desde cccc\

```

Figura 75: Prueba de Detección de E. Coli Img. 09

Fuente: Elaboración Propia



```

10  cm.load_weights(pesos_modelo)
11
12  def predict(file):
13      x = load_img(file, target_size=(longitud, altura))
14      x = img_to_array(x)
15      x = np.expand_dims(x, axis=0)
16      array = cm.predict(x)
17      result = array[0]
18      answer = np.argmax(result)
19      if answer == 0:
20          print("pred: Se Detecta E. Coli")
21      elif answer == 1:
22          print("pred: No contiene E. Coli")
23
24      return answer
25
26  predict('./E01/10.jpg')

```

```

was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli

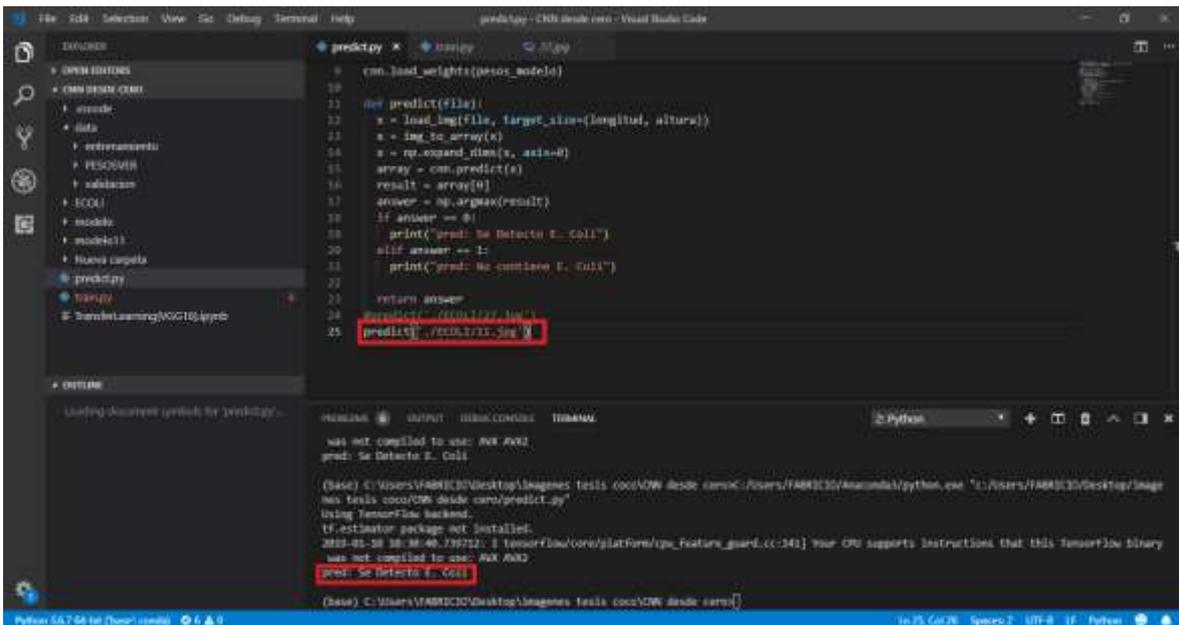
(base) C:\Users\FABRICIO\Desktop\imagenes tesis\coco\OW desde coco>C:\Users\FABRICIO\AppData\Local\python.exe "c:\Users\FABRICIO\Desktop\imagenes tesis\coco\OW desde coco\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-18 18:29:13.836264: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary
was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli

(base) C:\Users\FABRICIO\Desktop\imagenes tesis\coco\OW desde coco>

```

Figura 76: Prueba de Detección de E. Coli Img. 10

Fuente: Elaboración Propia



```

10  cm.load_weights(pesos_modelo)
11
12  def predict(file):
13      x = load_img(file, target_size=(longitud, altura))
14      x = img_to_array(x)
15      x = np.expand_dims(x, axis=0)
16      array = cm.predict(x)
17      result = array[0]
18      answer = np.argmax(result)
19      if answer == 0:
20          print("pred: Se Detecta E. Coli")
21      elif answer == 1:
22          print("pred: No contiene E. Coli")
23
24      return answer
25
26  predict('./E01/11.jpg')

```

```

was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli

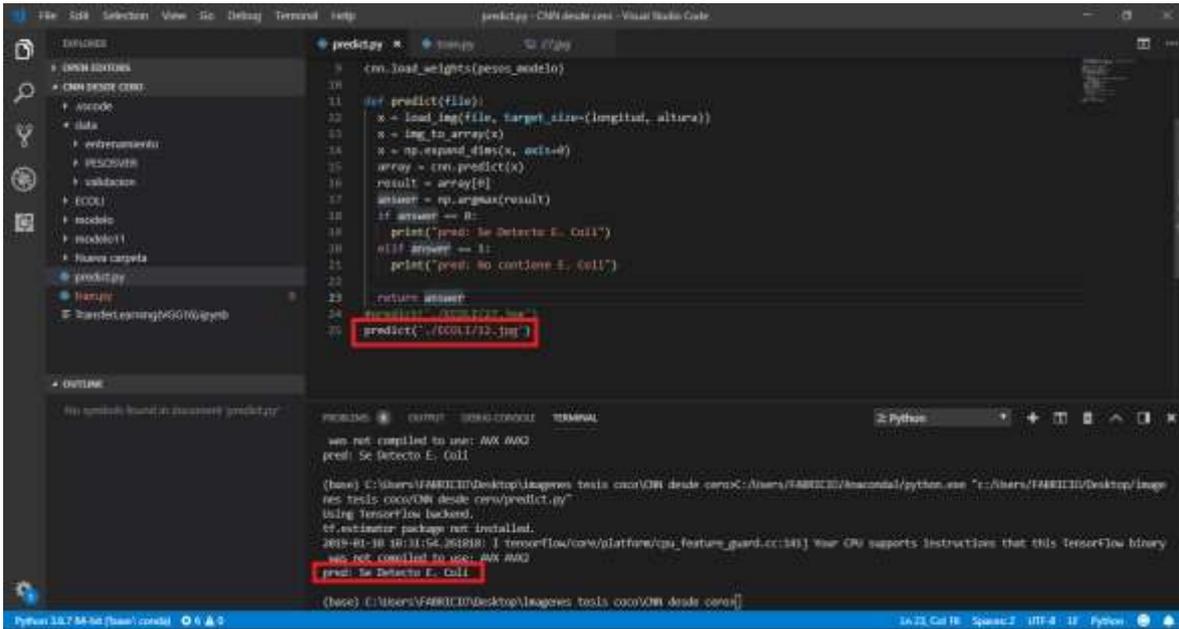
(base) C:\Users\FABRICIO\Desktop\imagenes tesis\coco\OW desde coco>C:\Users\FABRICIO\AppData\Local\python.exe "c:\Users\FABRICIO\Desktop\imagenes tesis\coco\OW desde coco\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-18 18:40:133721: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary
was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli

(base) C:\Users\FABRICIO\Desktop\imagenes tesis\coco\OW desde coco>

```

Figura 77: Prueba de Detección de E. Coli Img. 11

Fuente: Elaboración Propia



```

10:  crn.load_weights(pescoc_modelo)
11:
12: def predict(file):
13:     x = load_img(file, target_size=(longitud, altura))
14:     x = img_to_array(x)
15:     x = np.expand_dims(x, axis=0)
16:     array = crn.predict(x)
17:     result = array[0]
18:     answer = np.argmax(result)
19:     if answer == 0:
20:         print("pred: Se Detecta E. Coli")
21:     elif answer == 1:
22:         print("pred: No contiene E. Coli")
23:
24:     return answer
25:
26: predict('C:/Users/FABRICIO/Desktop/imagenes/tesis_coco/001 desde coco/predict.py')

```

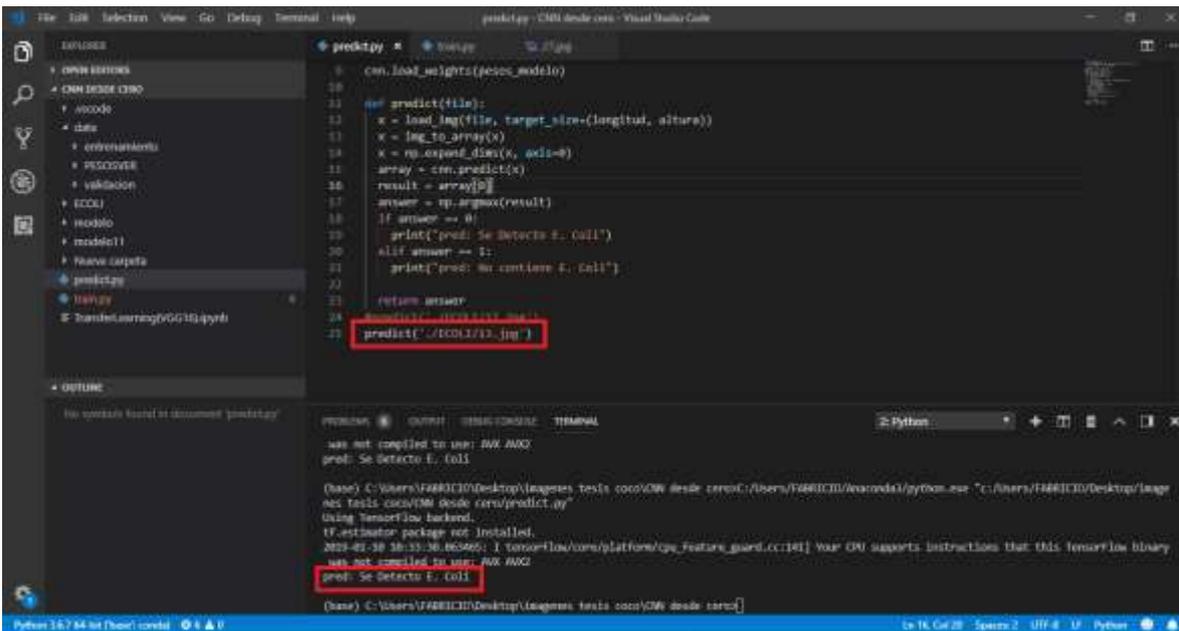
```

Python 3.8.7 Shell [base]: conda
[base] C:\Users\FABRICIO\Desktop\imagenes\tesis_coco\001 desde coco> python.exe "c:/Users/FABRICIO/Desktop/imagenes/tesis_coco/001 desde coco/predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2020-01-10 10:11:04.261810: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli
[base] C:\Users\FABRICIO\Desktop\imagenes\tesis_coco\001 desde coco>

```

Figura 78: Prueba de Detección de E. Coli Img. 12

Fuente: Elaboración Propia



```

10:  crn.load_weights(pescoc_modelo)
11:
12: def predict(file):
13:     x = load_img(file, target_size=(longitud, altura))
14:     x = img_to_array(x)
15:     x = np.expand_dims(x, axis=0)
16:     array = crn.predict(x)
17:     result = array[0]
18:     answer = np.argmax(result)
19:     if answer == 0:
20:         print("pred: Se Detecta E. Coli")
21:     elif answer == 1:
22:         print("pred: No contiene E. Coli")
23:
24:     return answer
25:
26: predict('C:/Users/FABRICIO/Desktop/imagenes/tesis_coco/001 desde coco/predict.py')

```

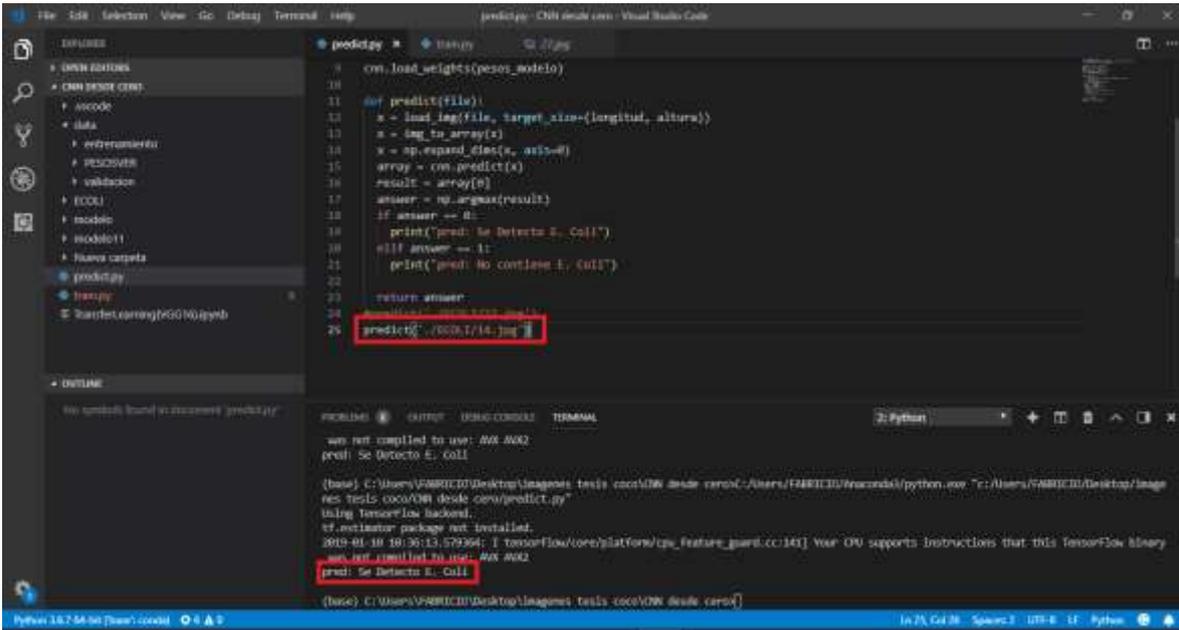
```

Python 3.8.7 Shell [base]: conda
[base] C:\Users\FABRICIO\Desktop\imagenes\tesis_coco\001 desde coco> python.exe "c:/Users/FABRICIO/Desktop/imagenes/tesis_coco/001 desde coco/predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2020-01-10 10:11:04.261810: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli
[base] C:\Users\FABRICIO\Desktop\imagenes\tesis_coco\001 desde coco>

```

Figura 79: Prueba de Detección de E. Coli Img. 13

Fuente: Elaboración Propia



```

10  cm.load_weights(pesos_modelo)
11  def predict(file):
12      x = load_img(file, target_size=(longitud, altura))
13      x = img_to_array(x)
14      x = np.expand_dims(x, axis=0)
15      array = cm.predict(x)
16      result = array[0]
17      answer = np.argmax(result)
18      if answer == 0:
19          print("pred: Se Detecta E. Coli")
20      elif answer == 1:
21          print("pred: No contiene E. Coli")
22
23      return answer
24
25  predict('./0001/14.jpg')

```

Terminal Output:

```

was not compiled to use: MAX 8002
pred: Se Detecta E. Coli

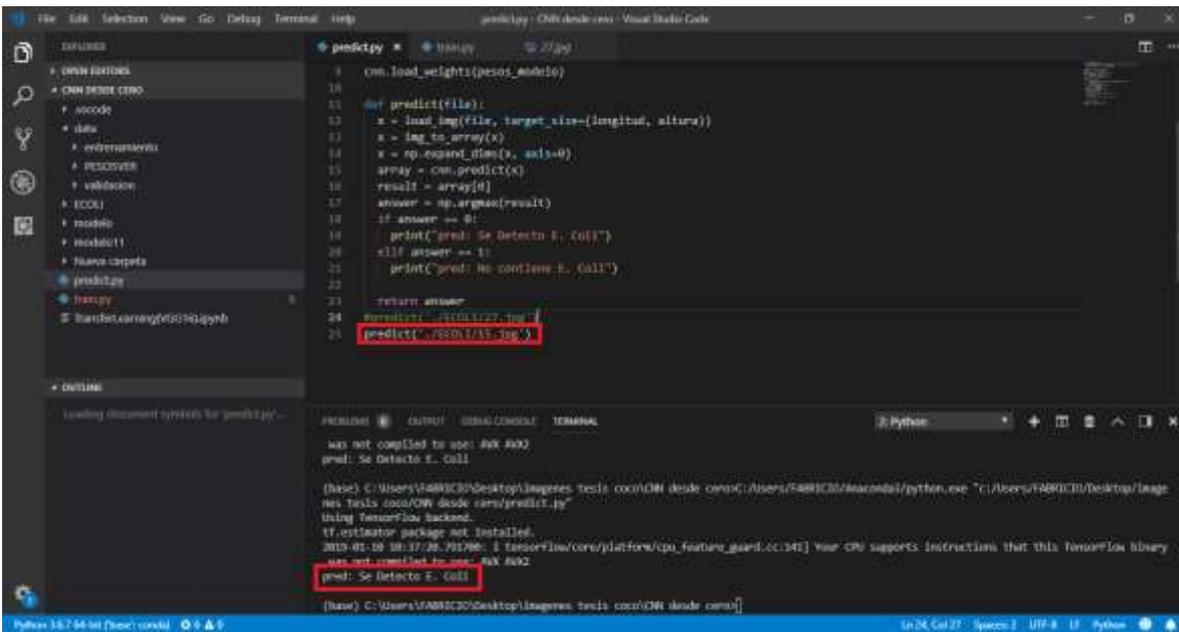
(base) C:\Users\FABRICIO\Desktop\imagenes tests\coco\OW desde casa\C:\Users\FABRICIO\AppData\Local\python.exe "c:\Users\FABRICIO\Desktop\imagenes tests\coco\OW desde casa\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-10 19:36:13.579364: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow library
was not compiled to use: MAX 8002
pred: Se Detecta E. Coli

(base) C:\Users\FABRICIO\Desktop\imagenes tests\coco\OW desde casa>

```

Figura 80: Prueba de Detección de E. Coli Img. 14

Fuente: Elaboración Propia



```

10  cm.load_weights(pesos_modelo)
11  def predict(file):
12      x = load_img(file, target_size=(longitud, altura))
13      x = img_to_array(x)
14      x = np.expand_dims(x, axis=0)
15      array = cm.predict(x)
16      result = array[0]
17      answer = np.argmax(result)
18      if answer == 0:
19          print("pred: Se Detecta E. Coli")
20      elif answer == 1:
21          print("pred: No contiene E. Coli")
22
23      return answer
24
25  predict('./0001/15.jpg')

```

Terminal Output:

```

was not compiled to use: MAX 8002
pred: Se Detecta E. Coli

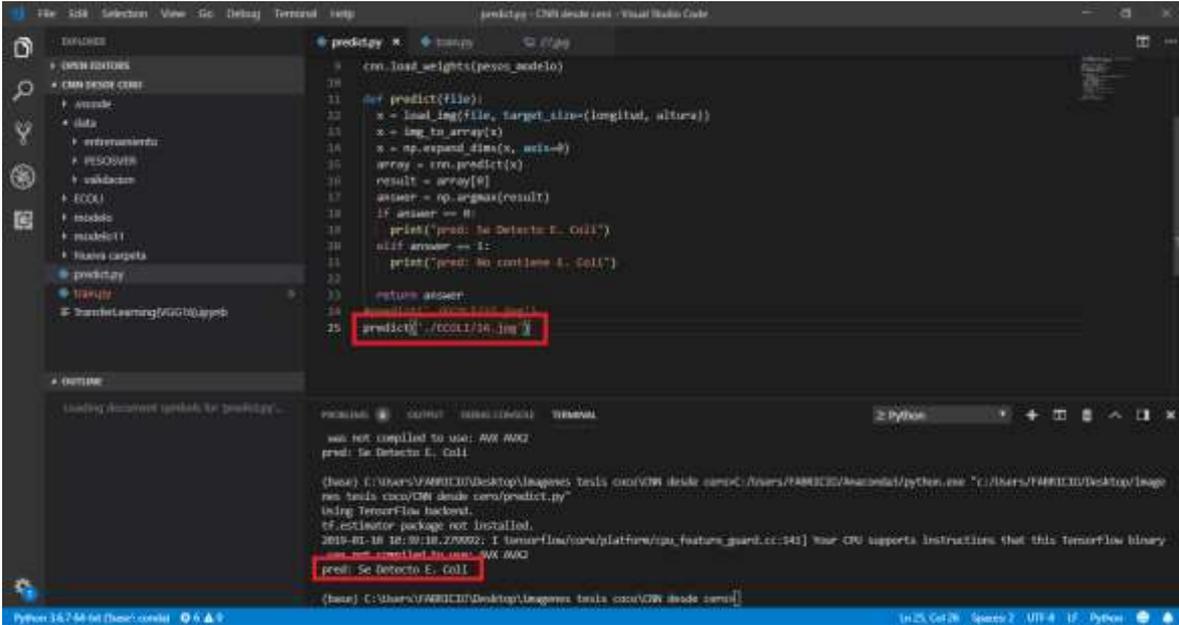
(base) C:\Users\FABRICIO\Desktop\imagenes tests\coco\OW desde casa\C:\Users\FABRICIO\AppData\Local\python.exe "c:\Users\FABRICIO\Desktop\imagenes tests\coco\OW desde casa\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-10 19:37:28.791796: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow library
was not compiled to use: MAX 8002
pred: Se Detecta E. Coli

(base) C:\Users\FABRICIO\Desktop\imagenes tests\coco\OW desde casa>

```

Figura 81: Prueba de Detección de E. Coli Img. 15

Fuente: Elaboración Propia



```

10 crn.load_weights(pesos_modelo)
11
12 def predict(file):
13     x = load_img(file, target_size=(longitud, altura))
14     x = img_to_array(x)
15     x = np.expand_dims(x, axis=0)
16     array = crn.predict(x)
17     result = array[0]
18     answer = np.argmax(result)
19     if answer == 0:
20         print("pred: Se Detecto E. Coli")
21     elif answer == 1:
22         print("pred: No contiene E. Coli")
23
24     return answer
25 predict("./COOL16.jpg")

```

```

[base] C:\Users\FABRICIO\Desktop\imagenes tesis coco\OV desde cero[
Python 3.6.7 64-bit (base) console
Python
no ref compiled to use: AVX AVX2
pred: Se Detecto E. Coli

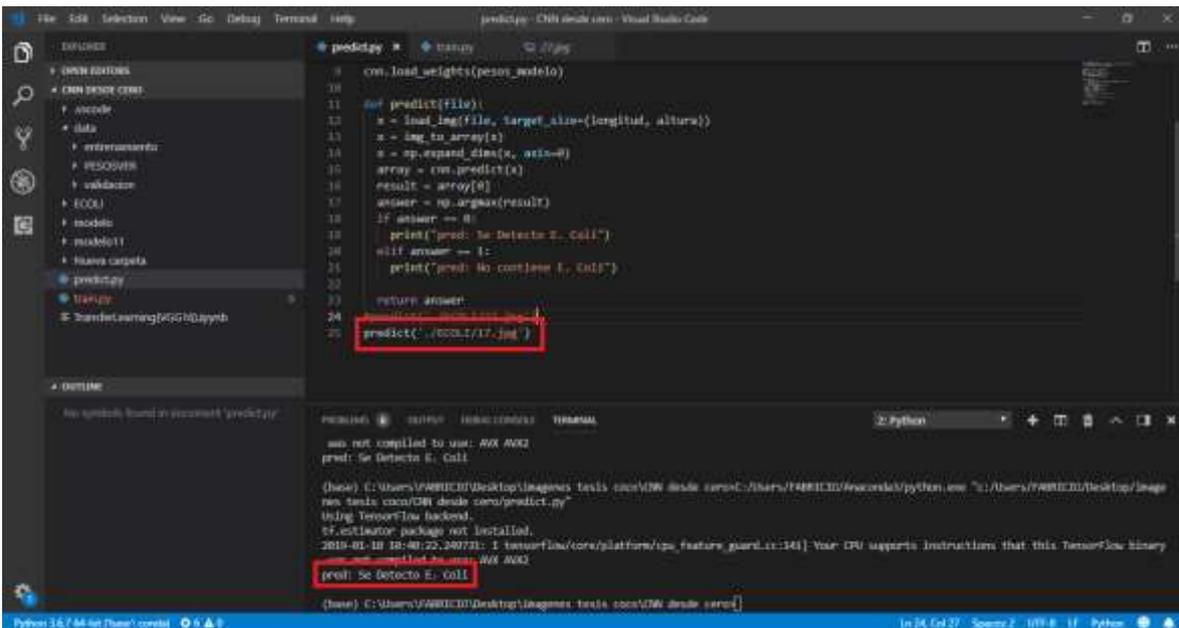
(base) C:\Users\FABRICIO\Desktop\imagenes tesis coco\OV desde cero> python.exe "c:/Users/FABRICIO/Desktop/imagenes tesis coco/OV desde cero/predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-18 12:40:18.279002: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary
was not compiled to use: AVX AVX2
pred: Se Detecto E. Coli

(base) C:\Users\FABRICIO\Desktop\imagenes tesis coco\OV desde cero[

```

Figura 82: Prueba de Detección de E. Coli Img. 16

Fuente: Elaboración Propia



```

10 crn.load_weights(pesos_modelo)
11
12 def predict(file):
13     x = load_img(file, target_size=(longitud, altura))
14     x = img_to_array(x)
15     x = np.expand_dims(x, axis=0)
16     array = crn.predict(x)
17     result = array[0]
18     answer = np.argmax(result)
19     if answer == 0:
20         print("pred: Se Detecto E. Coli")
21     elif answer == 1:
22         print("pred: No contiene E. Coli")
23
24     return answer
25 predict("./COOL17.jpg")

```

```

[base] C:\Users\FABRICIO\Desktop\imagenes tesis coco\OV desde cero[
Python 3.6.7 64-bit (base) console
Python
no ref compiled to use: AVX AVX2
pred: Se Detecto E. Coli

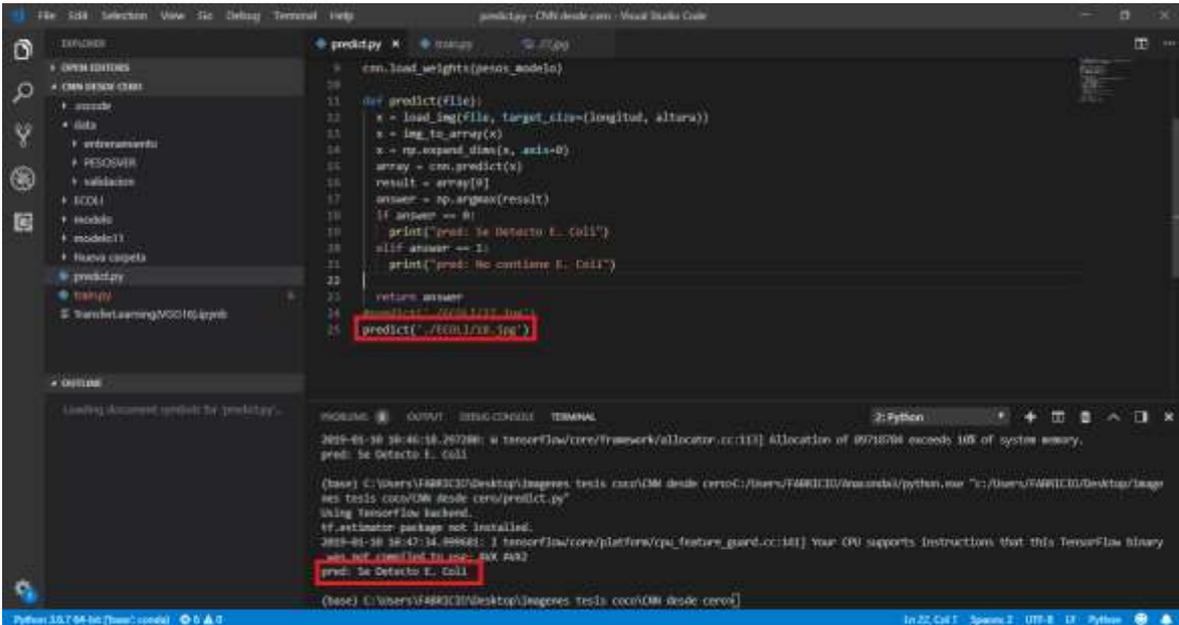
(base) C:\Users\FABRICIO\Desktop\imagenes tesis coco\OV desde cero> python.exe "c:/Users/FABRICIO/Desktop/imagenes tesis coco/OV desde cero/predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-18 12:40:22.240721: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary
was not compiled to use: AVX AVX2
pred: Se Detecto E. Coli

(base) C:\Users\FABRICIO\Desktop\imagenes tesis coco\OV desde cero[

```

Figura 83: Prueba de Detección de E. Coli Img. 17

Fuente: Elaboración Propia



```

10  cm.load_weights(pesos_modelo)
11
12  def predict(file):
13      x = load_img(file, target_size=(longitud, altura))
14      x = img_to_array(x)
15      x = np.expand_dims(x, axis=0)
16      array = cm.predict(x)
17      result = array[0]
18      answer = np.argmax(result)
19      if answer == 0:
20          print("pred: Se Detecto E. Coli")
21      elif answer == 1:
22          print("pred: No contiene E. Coli")
23
24      return answer
25
26  predict('./ECOLI18.jpg')

```

```

2023-01-18 18:46:14.297788: W tensorflow/core/framework/allocator.cc:113] Allocation of 8971696 exceeds 10% of system memory.
pred: Se Detecto E. Coli

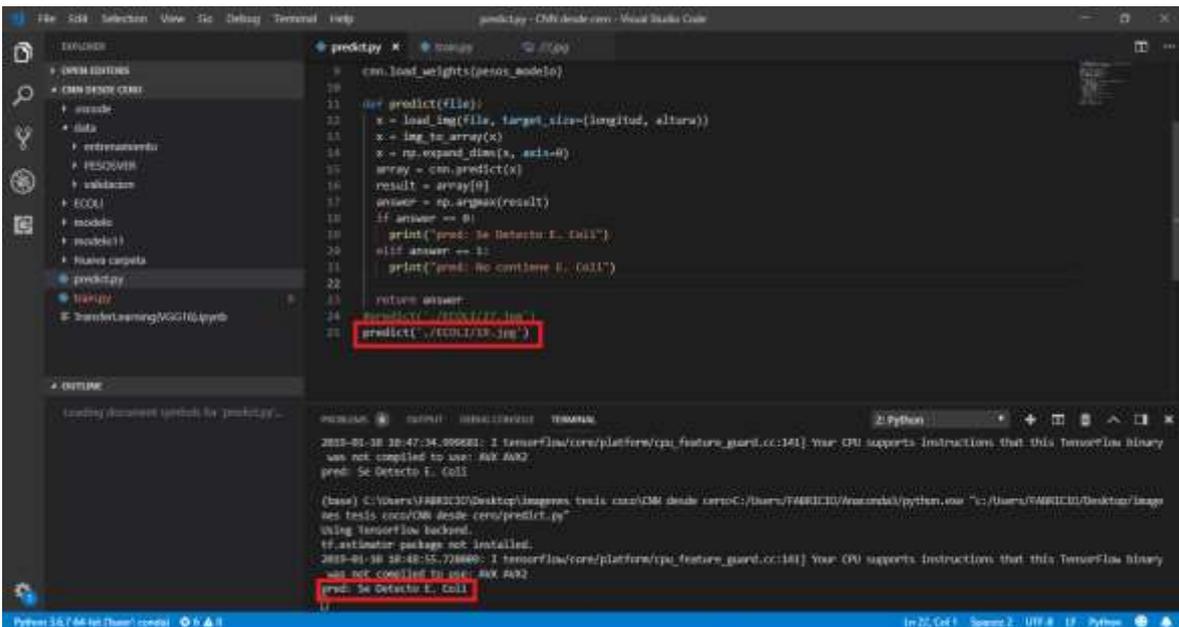
(base) C:\Users\FABRICIO\Desktop\imgenes tests c:\Users\FABRICIO\AppData\Local\Python\envs\tests\Scripts\python.exe "c:/Users/FABRICIO/Desktop/imgenes tests/coco/0M desde cero/predict.py"
Using TensorFlow backed.
tf.estimator package not installed.
2023-01-18 18:47:34.896681: I tensorflow/core/platforms/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
pred: 34 Detecto E. Coli

(base) C:\Users\FABRICIO\Desktop\imgenes tests\coco\0M desde cero\

```

Figura 84: Prueba de Detección de E. Coli Img. 18

Fuente: Elaboración Propia



```

10  cm.load_weights(pesos_modelo)
11
12  def predict(file):
13      x = load_img(file, target_size=(longitud, altura))
14      x = img_to_array(x)
15      x = np.expand_dims(x, axis=0)
16      array = cm.predict(x)
17      result = array[0]
18      answer = np.argmax(result)
19      if answer == 0:
20          print("pred: Se Detecto E. Coli")
21      elif answer == 1:
22          print("pred: No contiene E. Coli")
23
24      return answer
25
26  predict('./ECOLI19.jpg')

```

```

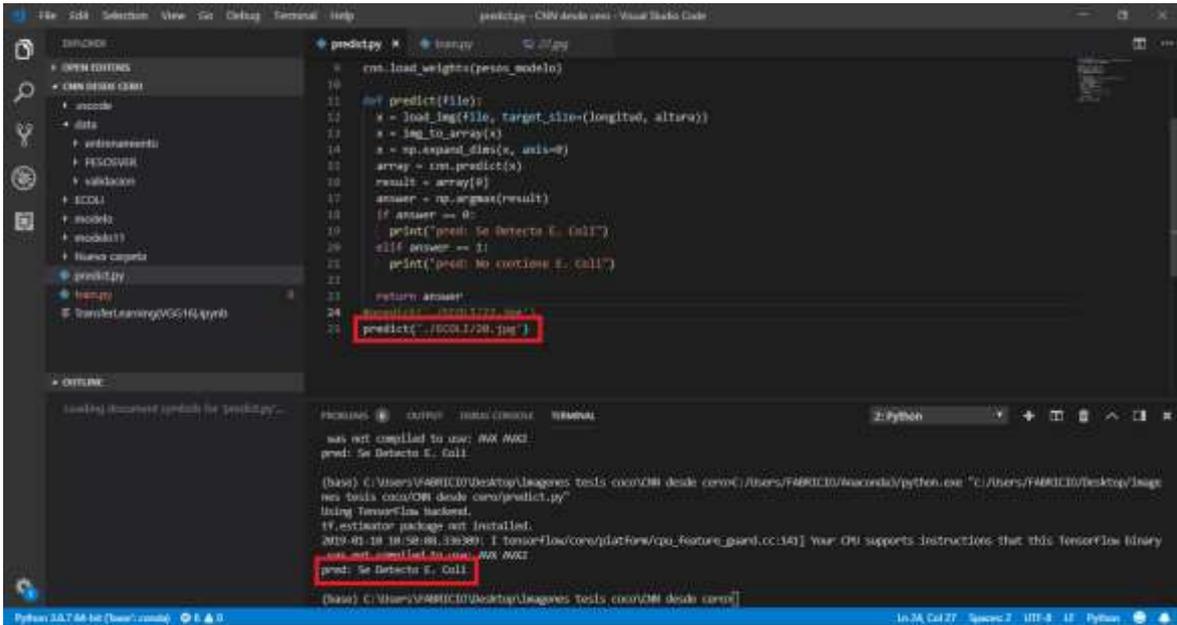
2023-01-18 18:47:34.896681: I tensorflow/core/platforms/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
pred: Se Detecto E. Coli

(base) C:\Users\FABRICIO\Desktop\imgenes tests\coco\0M desde cero\C:/Users/FABRICIO/AppData/Local/Python/envs/tests/Scripts/python.exe "c:/Users/FABRICIO/Desktop/imgenes tests/coco/0M desde cero/predict.py"
Using TensorFlow backed.
tf.estimator package not installed.
2023-01-18 18:48:55.728680: I tensorflow/core/platforms/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
pred: 34 Detecto E. Coli

```

Figura 85: Prueba de Detección de E. Coli Img. 19

Fuente: Elaboración Propia



```

File Edit Selection View Go Debug Terminal Help
predict.py - C:\Users\caro\... - Visual Studio Code

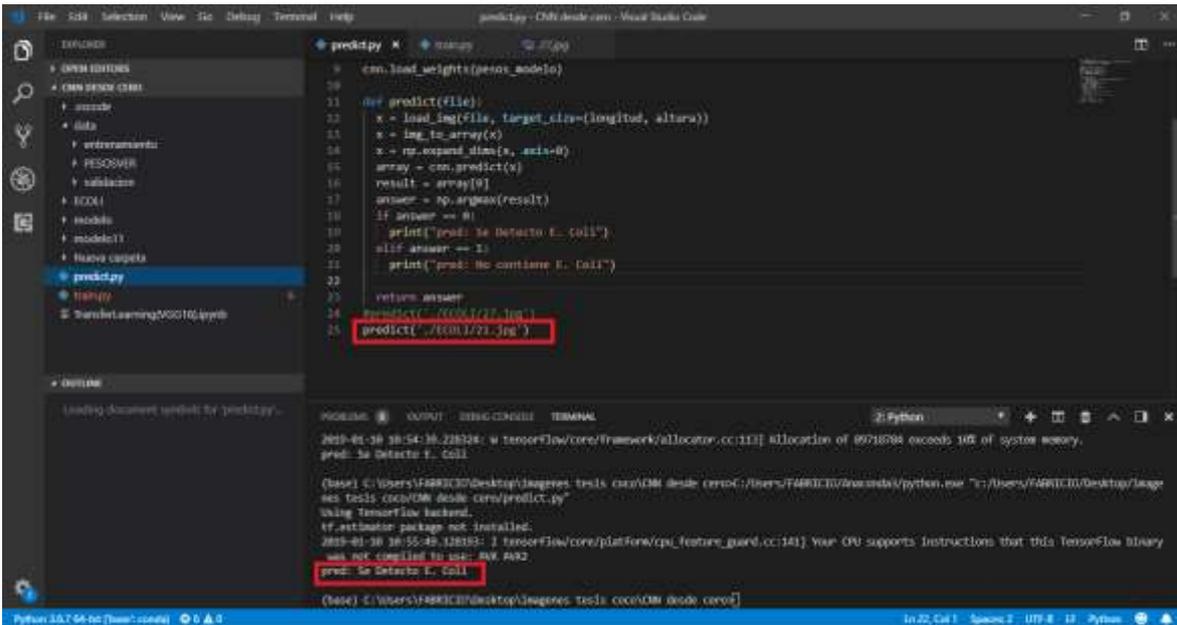
EXPLORER
+ OPEN EDITORS
+ C:\Users\caro\...
+ predict.py
+ imagen

TERMINAL
Python 3.8.10 [base:conda]
C:\Users\caro\...
python predict.py
[base] C:\Users\FABRICIO\Desktop\imagenes tests coco\08 desde coco\C:\Users\FABRICIO\AppData\Local\Python\Python38-64\python.exe "C:\Users\FABRICIO\Desktop\imagenes tests coco\08 desde coco\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-18 18:58:01.330389: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli
[base] C:\Users\FABRICIO\Desktop\imagenes tests coco\08 desde coco\

```

Figura 86: Prueba de Detección de E. Coli Img. 20

Fuente: Elaboración Propia



```

File Edit Selection View Go Debug Terminal Help
predict.py - C:\Users\caro\... - Visual Studio Code

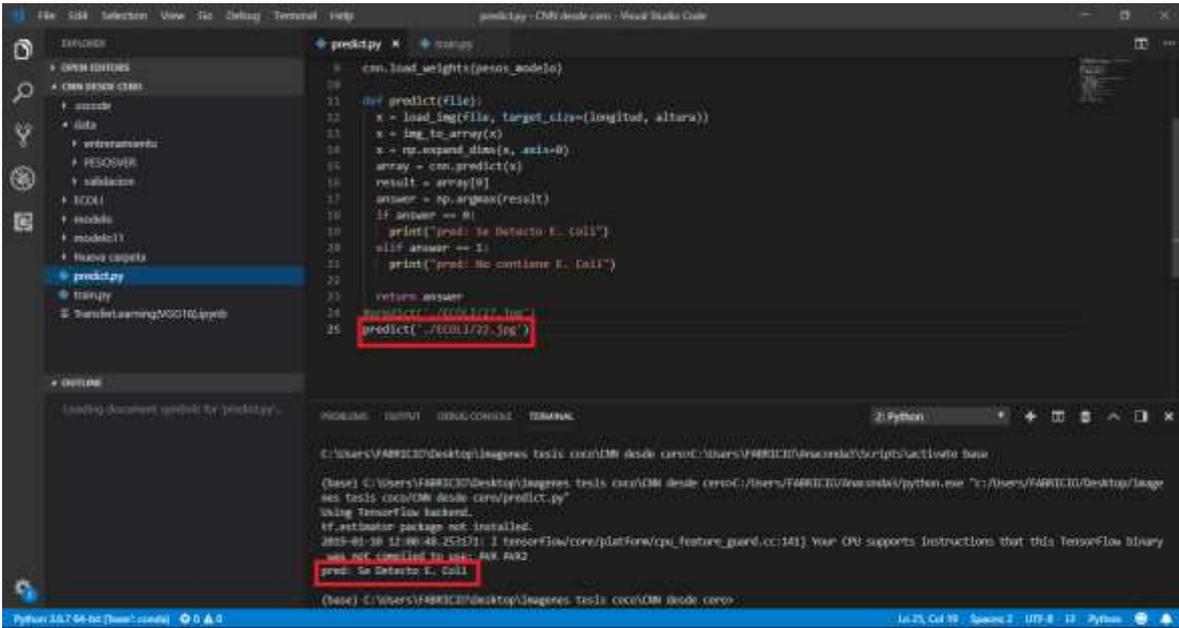
EXPLORER
+ OPEN EDITORS
+ C:\Users\caro\...
+ predict.py
+ imagen

TERMINAL
Python 3.8.10 [base:conda]
C:\Users\caro\...
python predict.py
[base] C:\Users\FABRICIO\Desktop\imagenes tests coco\08 desde coco\C:\Users\FABRICIO\AppData\Local\Python\Python38-64\python.exe "C:\Users\FABRICIO\Desktop\imagenes tests coco\08 desde coco\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-18 18:55:49.128185: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli
[base] C:\Users\FABRICIO\Desktop\imagenes tests coco\08 desde coco\

```

Figura 87: Prueba de Detección de E. Coli Img. 21

Fuente: Elaboración Propia



```

10  cnn.load_weights(pesos_modelo)
11
12  def predict(file):
13      x = load_img(file, target_size=(imgitud, altura))
14      x = img_to_array(x)
15      x = np.expand_dims(x, axis=0)
16      array = cnn.predict(x)
17      result = array[0]
18      answer = np.argmax(result)
19      if answer == 0:
20          print("pred: Se Detecta E. coli")
21      elif answer == 1:
22          print("pred: No contiene E. coli")
23
24      return answer
25
26  predict('ECOLI/22.jpg')
27

```

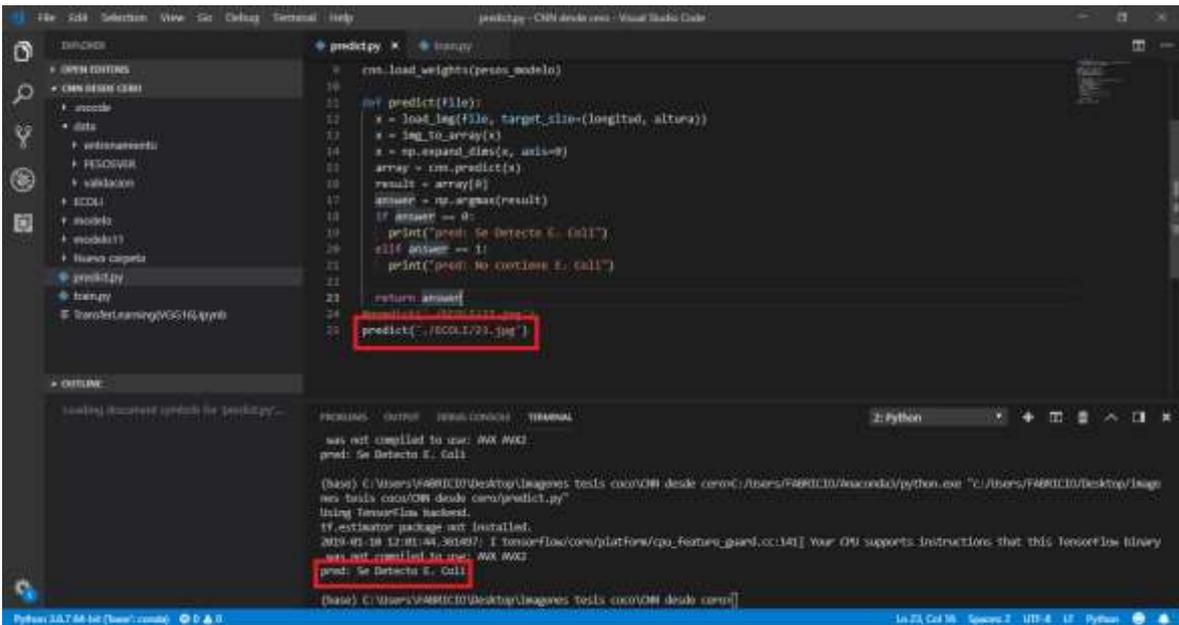
```

(base) C:\Users\FABRICIO\Desktop\imagenes tests cocc\ON desde cocc\Users\FABRICIO\anaconda\python.exe "C:/Users/FABRICIO/Desktop/imagenes tests cocc\ON desde cocc/predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-05-18 12:00:48.253741: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli
(base) C:\Users\FABRICIO\Desktop\imagenes tests cocc\ON desde cocc

```

**Figura 88:** Prueba de Detección de E. Coli Img. 22

*Fuente:* Elaboración Propia



```

10  cnn.load_weights(pesos_modelo)
11
12  def predict(file):
13      x = load_img(file, target_size=(imgitud, altura))
14      x = img_to_array(x)
15      x = np.expand_dims(x, axis=0)
16      array = cnn.predict(x)
17      result = array[0]
18      answer = np.argmax(result)
19      if answer == 0:
20          print("pred: Se Detecta E. coli")
21      elif answer == 1:
22          print("pred: No contiene E. coli")
23
24      return answer
25
26  predict('ECOLI/23.jpg')
27

```

```

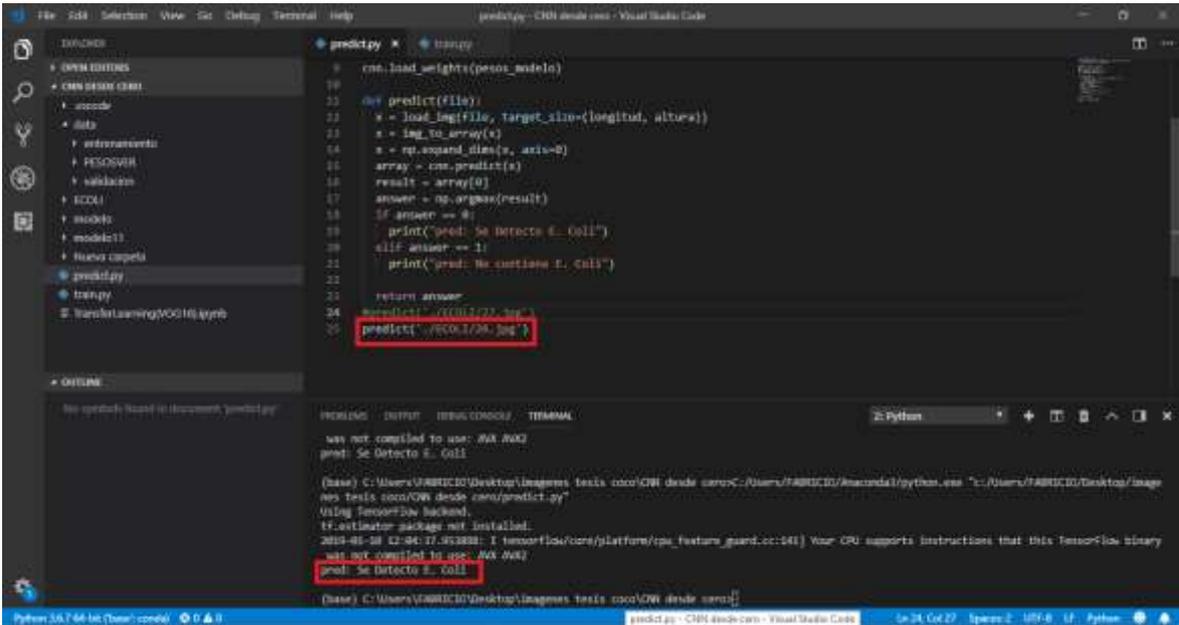
(base) C:\Users\FABRICIO\Desktop\imagenes tests cocc\ON desde cocc\Users\FABRICIO\anaconda\python.exe "C:/Users/FABRICIO/Desktop/imagenes tests cocc\ON desde cocc/predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2019-05-18 12:01:44.302897: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
pred: Se Detecta E. Coli
(base) C:\Users\FABRICIO\Desktop\imagenes tests cocc\ON desde cocc

```

**Figura 89:** Prueba de Detección de E. Coli Img. 23

*Fuente:* Elaboración Propia





```

10  from tensorflow.keras.models import load_model
11  def predict(file):
12      x = load_img(file, target_size=(longitud, altura))
13      x = img_to_array(x)
14      x = np.expand_dims(x, axis=0)
15      array = cnn.predict(x)
16      result = array[0]
17      answer = np.argmax(result)
18      if answer == 0:
19          print("pred: Se Detecto E. Coli")
20      elif answer == 1:
21          print("pred: No contiene E. Coli")
22
23      return answer
24  predict("./COLO/26.jpg")

```

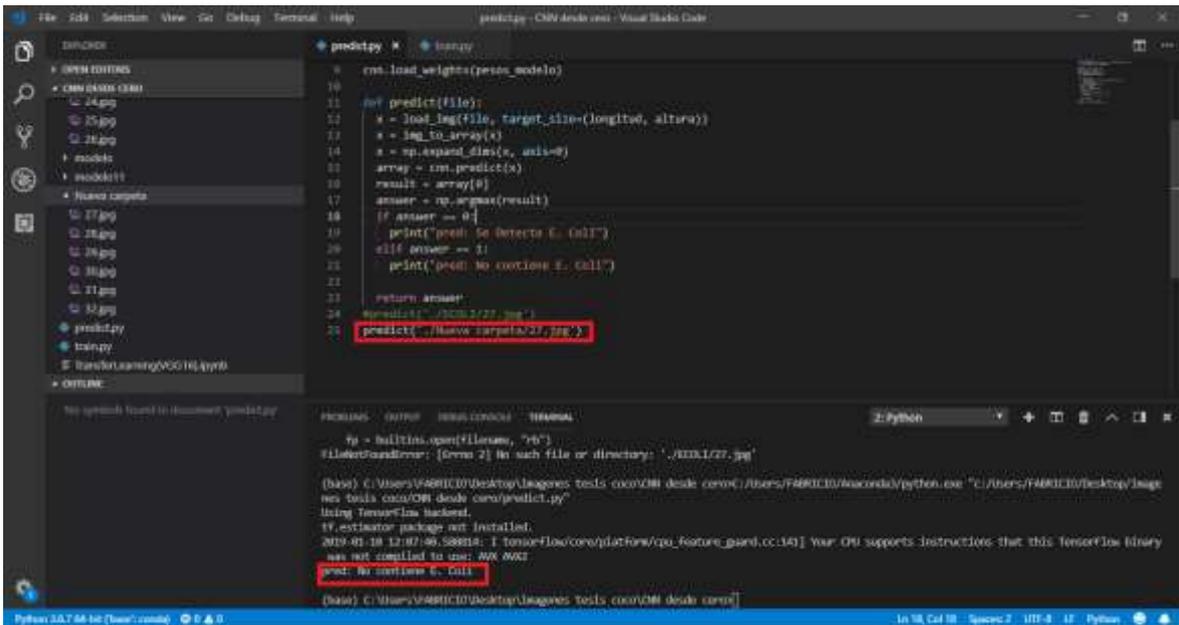
```

Python 3.8.7 64-bit (base: conda)
(base) C:\Users\FABRICIO\Desktop\imagenes tests coco\OW desde coco> python.exe "C:\Users\FABRICIO\Desktop\image tests coco\OW desde coco\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2020-05-18 12:04:27.5328814: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
pred: Se Detecto E. Coli

```

Figura 92: Prueba de Detección de E. Coli Img. 26

Fuente: Elaboración Propia



```

10  from tensorflow.keras.models import load_model
11  def predict(file):
12      x = load_img(file, target_size=(longitud, altura))
13      x = img_to_array(x)
14      x = np.expand_dims(x, axis=0)
15      array = cnn.predict(x)
16      result = array[0]
17      answer = np.argmax(result)
18      if answer == 0:
19          print("pred: Se Detecto E. Coli")
20      elif answer == 1:
21          print("pred: No contiene E. Coli")
22
23      return answer
24  predict("./Nueva carpeta/27.jpg")

```

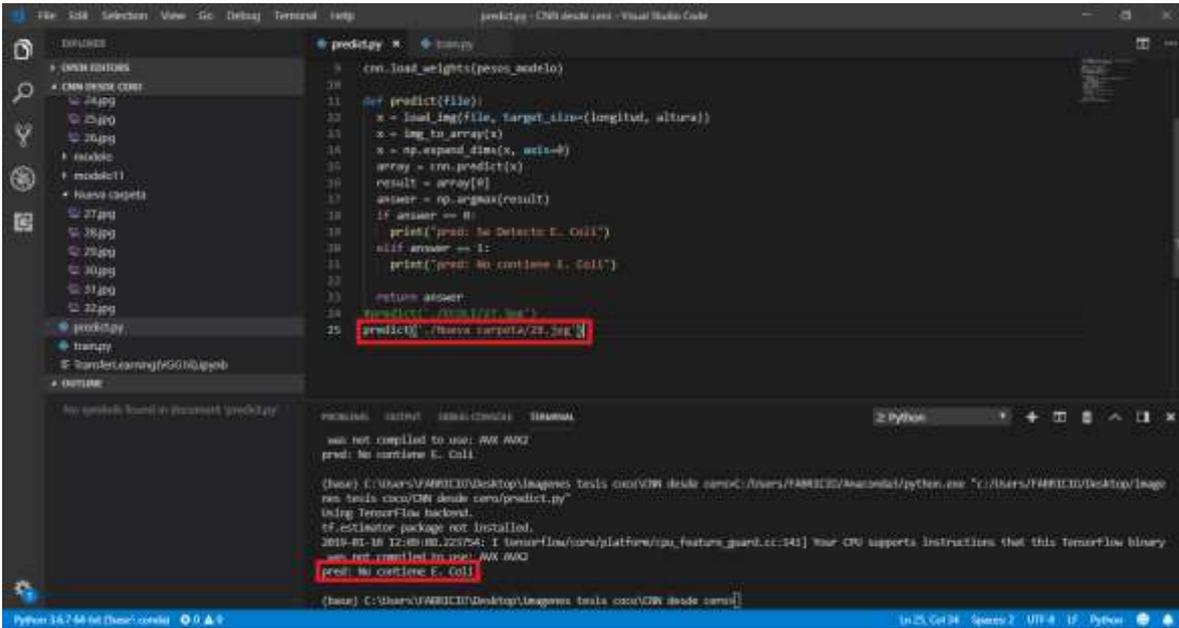
```

Python 3.8.7 64-bit (base: conda)
(base) C:\Users\FABRICIO\Desktop\imagenes tests coco\OW desde coco> python.exe "C:\Users\FABRICIO\Desktop\image tests coco\OW desde coco\predict.py"
Using TensorFlow backend.
tf.estimator package not installed.
2020-05-18 12:07:46.588214: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
pred: No contiene E. Coli

```

Figura 93: Prueba de Detección Negativa Img. 27

Fuente: Elaboración Propia



```

File Edit Selection View Go Debug Terminal Help
predict.py - C:\Users\carol\Desktop - Visual Studio Code

EXPLORER
+ OPEN EDITORS
+ C:\Users\carol\Desktop
  24.jpg
  25.jpg
  26.jpg
+ models
+ model1
+ Nueva carpeta
  27.jpg
  28.jpg
  29.jpg
  30.jpg
  31.jpg
  32.jpg
+ predict.py
+ train.py
+ transferLearningVGG16.py
+ outline

No symbols found in document 'predict.py'

predict.py
10 crn.load_weights(pesos_modelo)
11 def predict(file):
12     x = load_img(file, target_size=(longitud, altura))
13     x = img_to_array(x)
14     x = np.expand_dims(x, axis=0)
15     array = crn.predict(x)
16     result = array[0]
17     answer = np.argmax(result)
18     if answer == 0:
19         print("pred: se Detecta E. Coli")
20     elif answer == 1:
21         print("pred: No contiene E. Coli")
22
23     return answer
24 predict("./210177.jpg")
25 predict("./Nueva carpeta/27.jpg")

PROBLEMAS OUTPUT TERMINAL
was not compiled to use: MAX_ARGV
pred: se detecta E. Coli

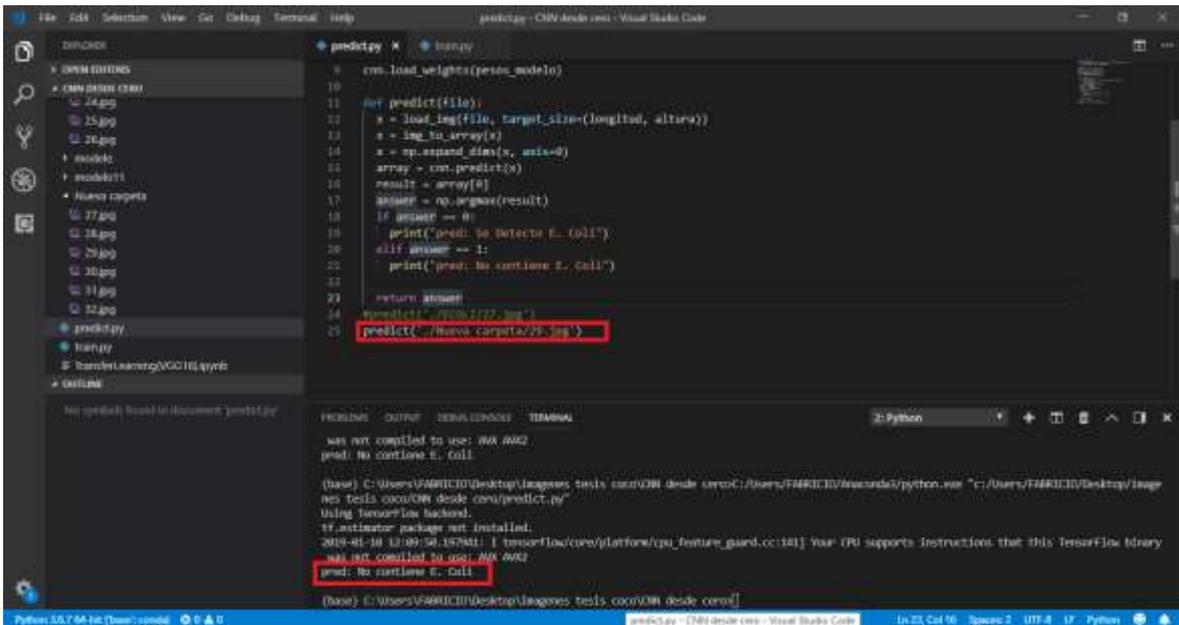
(base) C:\Users\FABRICIO\Desktop\imagenes tesis cocov\VM desde cervo> cd C:\Users\FABRICIO\Desktop\imagenes tesis cocov\VM desde cervo/predict.py
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-18 12:09:58.323754: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary
was not compiled to use: AVX AVX2
pred: No contiene E. Coli

(base) C:\Users\FABRICIO\Desktop\imagenes tesis cocov\VM desde cervo>
Python 3.6.7 64-bit (base): console
In: 25, Col: 34, Space: 2, UTF-8, Python

```

Figura 94: Prueba de Detección Negativa Img. 28

Fuente: Elaboración Propia



```

File Edit Selection View Go Debug Terminal Help
predict.py - C:\Users\carol\Desktop - Visual Studio Code

EXPLORER
+ OPEN EDITORS
+ C:\Users\carol\Desktop
  24.jpg
  25.jpg
  26.jpg
+ models
+ model1
+ Nueva carpeta
  27.jpg
  28.jpg
  29.jpg
  30.jpg
  31.jpg
  32.jpg
+ predict.py
+ train.py
+ transferLearningVGG16.py
+ outline

No symbols found in document 'predict.py'

predict.py
10 crn.load_weights(pesos_modelo)
11 def predict(file):
12     x = load_img(file, target_size=(longitud, altura))
13     x = img_to_array(x)
14     x = np.expand_dims(x, axis=0)
15     array = crn.predict(x)
16     result = array[0]
17     answer = np.argmax(result)
18     if answer == 0:
19         print("pred: se detecta E. coli")
20     elif answer == 1:
21         print("pred: No contiene E. Coli")
22
23     return answer
24 predict("./210177.jpg")
25 predict("./Nueva carpeta/29.jpg")

PROBLEMAS OUTPUT TERMINAL
was not compiled to use: MAX_ARGV
pred: No contiene E. Coli

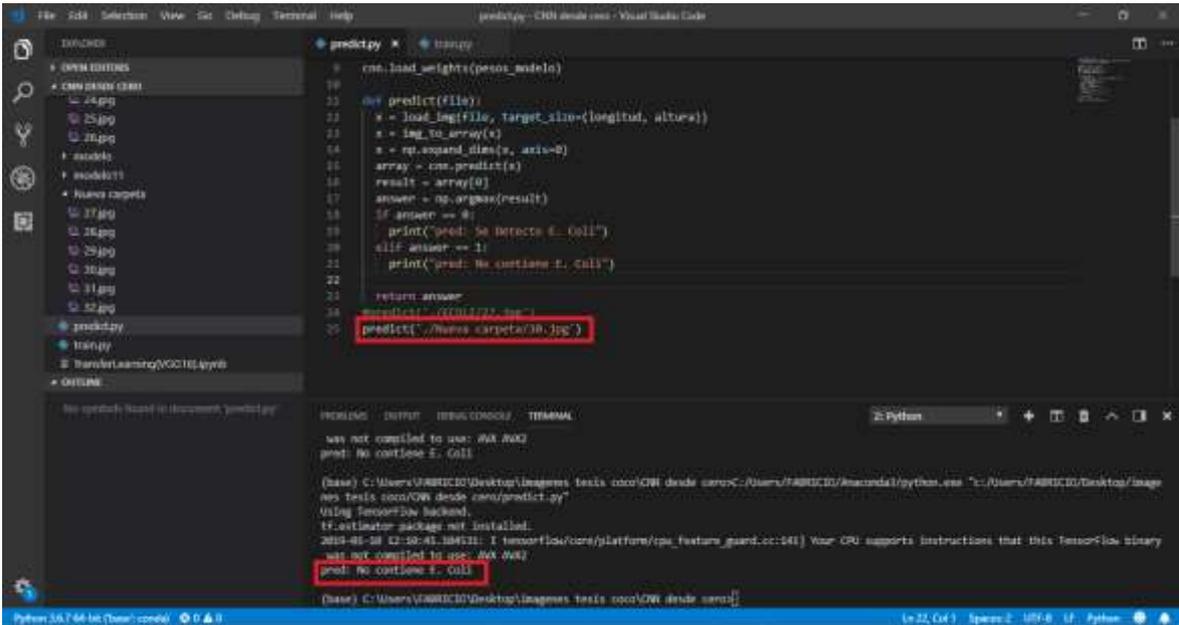
(base) C:\Users\FABRICIO\Desktop\imagenes tesis cocov\VM desde cervo> cd C:\Users\FABRICIO\Desktop\imagenes tesis cocov\VM desde cervo/predict.py
Using TensorFlow backend.
tf.estimator package not installed.
2019-01-18 12:09:58.157941: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary
was not compiled to use: AVX AVX2
pred: No contiene E. Coli

(base) C:\Users\FABRICIO\Desktop\imagenes tesis cocov\VM desde cervo>
Python 3.6.7 64-bit (base): console
In: 25, Col: 36, Space: 2, UTF-8, Python

```

Figura 95: Prueba de Detección Negativa Img. 29

Fuente: Elaboración Propia



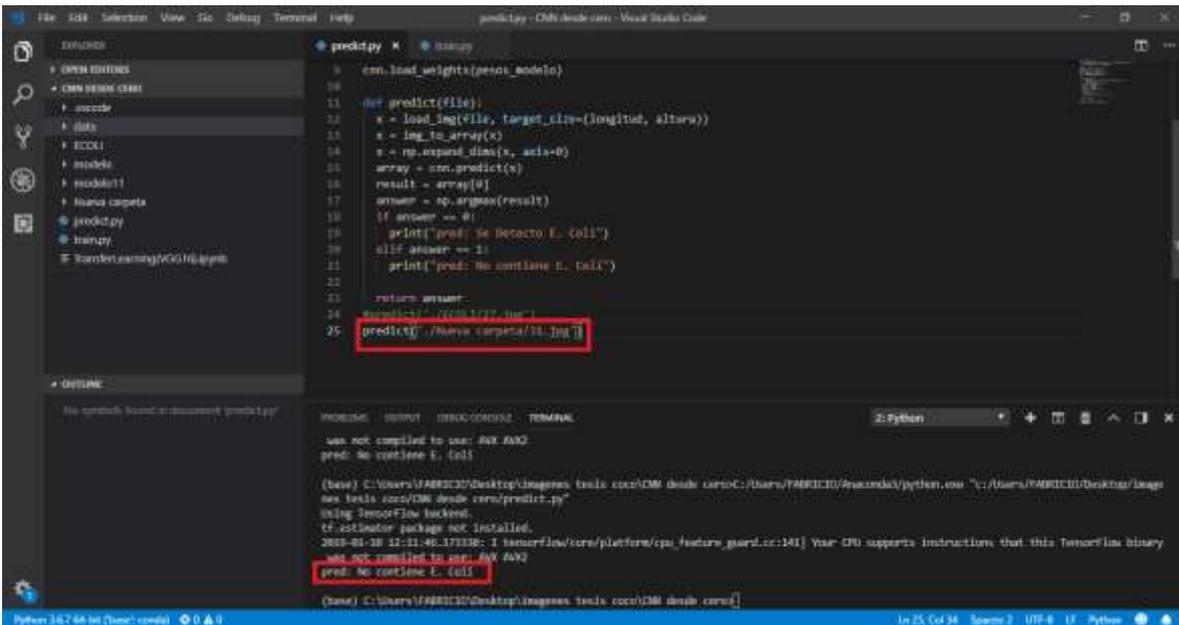
```

10  cm.load_weights(peso_modelo)
11
12  def predict(file):
13      x = load_img(file, target_size=(longitud, altura))
14      x = img_to_array(x)
15      x = np.expand_dims(x, axis=0)
16      array = cm.predict(x)
17      result = array[0]
18      answer = np.argmax(result)
19      if answer == 0:
20          print("pred: Se Detecto E. Coli")
21      elif answer == 1:
22          print("pred: No contiene E. Coli")
23
24      return answer
25
26  predict("../imagenes/tesis_coco/OW desde coco/predict.py")
27  predict("../imagenes/tesis_coco/OW desde coco/predict.py")

```

Figura 96: Prueba de Detección Negativa Img. 30

Fuente: Elaboración Propia



```

10  cm.load_weights(peso_modelo)
11
12  def predict(file):
13      x = load_img(file, target_size=(longitud, altura))
14      x = img_to_array(x)
15      x = np.expand_dims(x, axis=0)
16      array = cm.predict(x)
17      result = array[0]
18      answer = np.argmax(result)
19      if answer == 0:
20          print("pred: Se Detecto E. Coli")
21      elif answer == 1:
22          print("pred: No contiene E. Coli")
23
24      return answer
25
26  predict("../imagenes/tesis_coco/OW desde coco/predict.py")
27  predict("../imagenes/tesis_coco/OW desde coco/predict.py")

```

Figura 97: Prueba de Detección Negativa Img. 31

Fuente: Elaboración Propia

## ANEXO N.º 02

### LISTA DE COTEJO

Leyenda:

VP: Verdadero Positivo

VN: Verdadero Negativo

FP: Falso Positivo

FN: Falso Negativo

Imágenes	VP	VN	FP	FN
1.jpg	1	0	0	0
2.jpg	1	0	0	0
3.jpg	1	0	0	0
4.jpg	1	0	0	0
5.jpg	1	0	0	0
6.jpg	1	0	0	0
7.jpg	1	0	0	0
8.jpg	1	0	0	0
9.jpg	1	0	0	0
10.jpg	1	0	0	0
11.jpg	1	0	0	0
12.jpg	1	0	0	0
13.jpg	1	0	0	0
14.jpg	1	0	0	0

15.jpg	1	0	0	0
16.jpg	1	0	0	0
17.jpg	1	0	0	0
18.jpg	1	0	0	0
19.jpg	1	0	0	0
20.jpg	1	0	0	0
21.jpg	1	0	0	0
22.jpg	1	0	0	0
23.jpg	1	0	0	0
24.jpg	1	0	0	0
25.jpg	1	0	0	0
26.jpg	1	0	0	0
27.jpg	1	0	0	0
28.jpg	0	1	0	0
29.jpg	0	1	0	0
30.jpg	0	1	0	0
31.jpg	0	1	0	0
32.jpg	0	1	0	0
33.jpg	0	1	0	0
34.jpg	0	1	0	0
35.jpg	0	1	0	0

**ANEXO N.º 03**  
**MATRIZ DE CONSISTENCIA**

<b>DESARROLLO DE UN SISTEMA INTELIGENTE BASADO EN VISIÓN COMPUTACIONAL PARA DETECTAR BACTERIAS ESCHERICHIA COLI EN VERDURAS FRESCAS</b>				
<b>PROBLEMA</b>	<b>HIPÓTESIS</b>	<b>OBJETIVO GENERAL</b>	<b>VARIABLE INDEPENDIENTE</b>	<b>METODOLOGÍA</b>
<p>¿Cómo detectar bacterias Escherichia Coli en Verduras Frescas?</p>	<p>Un Sistema Inteligente basado en visión computacional detecta bacterias Escherichia Coli en Verduras Frescas</p>	<p>Detectar bacterias Escherichia Coli en Verduras Frescas mediante un Sistema Inteligente basado en visión computacional.</p>	<p><i>Sistema Inteligente basado en visión computacional</i></p>	<p><b>Diseño</b></p> <p><i>El tipo de investigación es <b>Experimental</b> con un diseño Pre Experimental de un solo grupo con una sola medición</i></p> <p style="text-align: center;"><math>G: X O_1</math></p> <p><i>Donde:</i></p> <p><math>G =</math> Grupo de Imágenes a Evaluar</p> <p><math>X =</math> Sistema Inteligente basado en visión computacional</p> <p><math>O_1:</math> Detección de bacterias Escherichia Coli en Verduras Frescas</p>
				<p><b>Población</b></p>
				<p><i>315 imágenes de bacterias que son muestras de verduras</i></p>

		<b>OBJETIVOS ESPECIFICOS</b>	<b>VARIABLE DEPENDIENTE</b>	<b>Muestra</b>
		<p><i>Conseguir un nivel de sensibilidad mayor o igual a 98.07%.</i></p> <p><i>Conseguir un nivel de especificidad mayor o igual a 95,4%.</i></p> <p><i>Detección de la bacteria Escherichia Coli en un 80%</i></p>	<p><i>Detección de bacterias Escherichia Coli en Verduras Frescas</i></p>	<p>Sabiendo que nuestra población es finita aplicamos lo siguiente:</p> <p>z: Nivel de confianza 95% = 1.96</p> <p>p: prevalencia esperada del parámetro a evaluar, en caso de desconocerse (<math>p = 0.5</math>), que hace mayor el tamaño muestral</p> <p>q: <math>1 - p</math> (si <math>p = 50\%</math>, <math>q = 50\%</math>)</p> <p>d: precisión al 5% = 0.05</p> <p>Dádonos como resultado <b>173 imágenes</b> como muestra</p>

**ANEXO N.º 04**

**MATRIZ DE OPERACIONALIZACION**

<b>VARIABLE INDEPENDIENTE</b>	<b>DEFINICIÓN CONCEPTUAL</b>	<b>DEFINICIÓN OPERACIONAL</b>	<b>DIMENSIONES</b>	<b>INDICADORES</b>
<i>Sistema Inteligente basado en visión computacional</i>	<i>Software que posee la capacidad de percibir su entorno a través de imágenes digitales, procesa dichos datos y emite un resultado a su entorno. . (Gómez &amp; Sucar, 2008)</i>	<i>El sistema inteligente basado en visión computacional es medido mediante el ingreso de imágenes digitales procesadas, emitiendo un resultado positivo o negativo. (Rodríguez y otros, 2014)</i>	<i>Sensibilidad</i>	<i>Porcentaje de error del software.</i>
			<i>Especificidad</i>	<i>Porcentaje de falsos positivos</i>

VARIABLE DEPENDIENTE	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	DIMENSIONES	INDICADORES
<p><i>Detección de bacterias Escherichia Coli en Verduras Frescas</i></p>	<p><i>Proceso mediante el cual se analizan muestras de verduras para poder determinar la existencia de bacterias Escherichia Coli. (Garcia Santillán, 2008)</i></p>	<p><i>La detección de bacterias escherichia coli en verduras frescas se analiza mediante el sistema inteligente basado en visión computacional, emitiendo un resultado positivo o negativo. (Pablo, 2013),</i></p>	<p><i>Eficacia</i></p>	<p><i>Porcentaje de detección de bacterias</i></p>

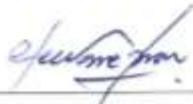
## ANEXO N.º 05

### CONSTANCIA DEL ANALISIS DE MUESTRAS

Se realizó un muestreo conformado por 60 muestras de verduras, que procedieron a ser cultivadas para lograr la identificación de bacterias Escherichia Coli.

Se lograron identificar colonias de bacterias las cuales pasaron por un proceso de Tinción Gram y se identificaron las siguientes bacterias: Escherichia Coli, Salmonella y cocobacilos.

Se analizó una colonia de bacterias de la cual se obtuvieron 315 imágenes en total.



Ms. C. Luis Llenque Díaz

CODIGO UNT 4704

DNI: 16598579.

Fac. CC. Biológicas

Dpto. Académico de Microbiología.