



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE INGENIERÍA ELECTRÓNICA

DESARROLLO DE UN SISTEMA PARA DETERMINAR LA CALIDAD DE
AGUA EN LA CUENCA DEL RÍO PAUTE, BASADO EN LA IDENTIFICACIÓN
DE MACROINVERTEBRADOS ACUÁTICOS COMO INDICADORES DE
CALIDAD DE AGUA, UTILIZANDO TÉCNICAS DE VISIÓN POR
COMPUTADOR

Trabajo de titulación previo a la obtención del
título de Ingeniera Electrónica

AUTORA: MARÍA ISABEL CAÑAR UYAGUARI
TUTOR: ING. MARCELO ESTEBAN FLORES VÁZQUEZ, MST.

Cuenca - Ecuador

2023

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, María Isabel Cañar Uyaguari con documento de identificación N° 0106805393, manifiesto que:

Soy la autora y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 17 de junio del 2023

Atentamente,

A handwritten signature in blue ink, consisting of a large circle with a vertical line through it and some scribbles below, followed by a horizontal line extending to the right.

María Isabel Cañar Uyaguari

0106805393

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO
DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, María Isabel Cañar Uyaguari con documento de identificación N° 0106805393, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autora del Proyecto técnico: “Desarrollo de un sistema para determinar la calidad de agua en la cuenca del río Paute, basado en la identificación de macroinvertebrados acuáticos como indicadores de calidad de agua, utilizando técnicas de visión por computador”, el cual ha sido desarrollado para optar por el título de: Ingeniera Electrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 17 de junio del 2023

Atentamente,

A handwritten signature in blue ink, consisting of a large, stylized 'M' and 'I' followed by 'Cañar Uyaguari'.

María Isabel Cañar Uyaguari

0106805393

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Marcelo Esteban Flores Vázquez con documento de identificación N° 0102408978, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO DE UN SISTEMA PARA DETERMINAR LA CALIDAD DE AGUA EN LA CUENCA DEL RÍO PAUTE, BASADO EN LA IDENTIFICACIÓN DE MACROINVERTEBRADOS ACUÁTICOS COMO INDICADORES DE CALIDAD DE AGUA, UTILIZANDO TÉCNICAS DE VISIÓN POR COMPUTADOR, realizado por María Isabel Cañar Uyaguari con documento de identificación N° 0106805393, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 17 de junio del 2023

Atentamente,



Ing. Marcelo Esteban Flores Vázquez, Mst.

0102408978

AGRADECIMIENTOS

Agradezco eternamente a mi amada familia, de manera especial a Mercedes, Zoilita, María Teresa, Pablo y Juan Carlos, cuyo apoyo incondicional y aliento constante han sido la piedra angular de mi trayectoria académica. Su respaldo afectuoso ha sido un faro en los momentos de incertidumbre y un impulso en los logros alcanzados. Gracias familia por brindarme su apoyo.

De igual forma agradezco a mi asesor de tesis, Mgt. Marcelo Flores, por su guía, a lo largo de todo el proceso. Su experiencia y conocimientos fueron fundamentales para dirigir mi investigación y brindarme orientación valiosa en cada etapa de esta investigación.

María Isabel Cañar.

DEDICATORIAS

A Dios, fuente de vida, amor y sabiduría infinita. Dedico este logro a ti, reconociendo que, sin tu bendición y gracia, nada de esto hubiese sido posible.

A mi amada madre, mi fuente inagotable de inspiración y fortaleza. Desde el primer día de mi vida, usted ha sido mi guía y mi apoyo incondicional. Su amor inquebrantable y su ejemplo de perseverancia me han enseñado que no hay límites para lo que puedo lograr. Espero que este logro sea una muestra de mi agradecimiento eterno. Gracias, mamá, por ser mi luz en los momentos oscuros, mi apoyo en los momentos difíciles y mi mayor admiradora en cada éxito. Esta tesis lleva su amor y sus enseñanzas en cada página, y mi vida siempre llevará el legado de su amor y sabiduría.

A mi querida abuelita, un pilar de sabiduría y amor incondicional. Su guía y sus enseñanzas han dejado una huella imborrable en mi vida y en mi formación académica. Siempre ha estado ahí para escucharme, aconsejarme y celebrar mis logros. Mi gratitud hacia usted es infinita, le estoy profundamente agradecida por su presencia constante y su invaluable contribución a mi vida.

Con profunda gratitud, quiero expresar mi agradecimiento a mi tía por sus palabras de aliento, sus valiosos consejos y su apoyo constante tanto a mis hermanos como a mí. Sin su respaldo, este logro no hubiese sido posible. Mi reconocimiento especial por su incondicional apoyo y por enseñarme que el éxito se alcanza con dedicación y perseverancia.

A mis hermanos, mis compañeros de vida. Su presencia ha sido un regalo invaluable. Desde nuestra infancia, hemos compartido risas, aventuras y desafíos. A través de su amor incondicional y apoyo constante, me han motivado a alcanzar mis metas y superar mis límites. Estoy eternamente agradecida con ustedes; los quiero mucho.

Deseo expresar también mi más profundo agradecimiento a mi toda mi familia y a aquellos que no he mencionado de manera individual, pero que han estado presentes en cada etapa de este camino. Su amor, comprensión y apoyo incondicional han sido mi mayor fortaleza a lo largo de esta investigación y en todos los aspectos de mi vida. Los llevo siempre en mi corazón y dedico este logro a cada uno de ustedes.

Con todo mi amor y gratitud,

María Isabel Cañar.

INDICE DE CONTENIDOS

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN	I
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA	II
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.....	III
AGRADECIMIENTOS	IV
DEDICATORIAS.....	V
INDICE DE CONTENIDOS	VII
INDICE DE FIGURAS	IX
INDICE DE TABLAS	XIII
RESUMEN.....	XIV
INTRODUCCIÓN	XV
ANTECEDENTES DEL PROBLEMA DE ESTUDIO	XVI
JUSTIFICACIÓN (IMPORTANCIA Y ALCANCES).....	XVIII
OBJETIVOS.....	XIX
OBJETIVO GENERAL	XIX
OBJETIVOS ESPECÍFICOS	XIX
CAPÍTULO 1: ESTADO DEL ARTE	1
1.1 Fundamentos teóricos.....	1
1.2 Macroinvertebrados acuáticos que habitan en la cuenca del rio paute.....	4
1.3 Inteligencia artificial	7
1.4 Machine learning.....	11
1.5 Deep learning.....	23
1.6 Técnicas de visión por computador.....	24
1.7 Servidor web.....	25
1.8 Funcionalidades de un servidor web	25
1.9 Protocolo http.....	26
1.10 Backend	27
1.11 Frontend.....	27

1.12 Python.....	27
1.13 Flutter	28
CAPÍTULO 2: MARCO METODOLÓGICO	29
2.1 Librerías utilizadas en la CNN.....	29
2.2 Diseño de la arquitectura	38
2.3 Desarrollo del código para el entrenamiento de la CNN.....	39
2.4 Implementación del código para la identificación de macroinvertebrados utilizando técnicas de visión por computador, mediante opencv	41
2.5 Configuración de flutter para el desarrollo de la app móvil	42
2.6 Configuración de django para el desarrollo de la aplicación web.....	45
CAPÍTULO 3: IMPLEMENTACIÓN Y ANÁLISIS DE RESULTADOS	48
3.1 Visualización del desarrollo del código para la identificación de macroinvertebrados implementando técnicas de visión por computador, mediante opencv	48
3.2 Creación y configuración de VPC en la nube.....	49
3.3 Despliegue de la aplicación en la VPC	49
3.4 Visualización de la interfaz de la plataforma en línea.....	50
3.5 Visualización de la interfaz de la aplicación móvil.	56
CAPÍTULO 4:	57
CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS.	57
4.1 Conclusiones.....	57
4.2 Análisis realizado en la CNN.....	57
4.3 Analisis de la matriz de confusión de la CNN.....	69
4.4 Recomendaciones.....	72
4.5 Investigaciones futuras	72
4.6 Referencias bibliograficas	75
4.7 Anexos.....	77

INDICE DE FIGURAS

Figura 1. Grupo Odonata (Libélulas) en ninfas y adulto, perteneciente a los macroinvertebrados. [13]	3
Figura 2. Macroinvertebrados pertenecientes a la clase Oligochaeta. [15].....	4
Figura 3. Macroinvertebrados pertenecientes a la clase Insecta [15]	5
Figura 4. Macroinvertebrados pertenecientes a la clase Gastropoda. [15]	5
Figura 5. Macroinvertebrados pertenecientes a la clase Crustácea. [15].....	5
Figura 6. Macroinvertebrados pertenecientes a la clase Turbellaria. [15].....	6
Figura 7. Árboles de decisión [17]	13
Figura 8. Clasificación binaria mediante el método k-NN con un valor de $k=5$ [18]	13
Figura 9. Máquina de vectores de soporte: hiperplanos de decisión lineal en un espacio de características. [19]	14
Figura 10. Estructura de una red neuronal artificial feedforward. [19]	15
Figura 11. Red Neuronal Convolutiva.[19]	16
Figura 12. Explicación detallada de las capas convolucionales.[19].....	17
Figura 13. Operación de convolución.[19].....	18
Figura 14. Explicación de Max Pooling.[19]	19
Figura 15. Fully connected layer.[19].....	20
Figura 16. Función ReLU. [19]	21
Figura 17. Regresión Lineal.[19].....	22
Figura 18. Ejemplificación del funcionamiento de un servidor web. [22].....	25
Figura 19. Verificación de la instalación de OpenCV 4.7.0.72 en python (Windows). [Autor].....	29
Figura 20. Función cv2.GaussianBlur. [Autor].....	29
Figura 21. Función cv2.cvtColor. [Autor]	30
Figura 22. Función cv2.threshold. [Autor].....	31
Figura 23. Función cv2.Canny. [Autor].....	32
Figura 24. Función cv2.dilate. [Autor]	32
Figura 25. Función cv2.findContours. [Autor].....	33
Figura 26. Función cv2.contourArea. [Autor].....	33
Figura 27. Función cv2.arcLength [Autor]	34
Figura 28. Función cv2.approxPolyDP [Autor]	34

Figura 29. Función cv2.boundingRect [Autor]	35
Figura 30. Función cv2.rectangle. [Autor].....	35
Figura 31. Función cv2.putText. [Autor]	36
Figura 32. Verificación de la instalación de Numpy 1.21.5 en python (Windows). [Autor].....	37
Figura 33. Verificación de la instalación de Keras 2.9.0 en python (Windows). [Autor].....	38
Figura 34. Diseño de la arquitectura del proyecto. [Autor]	39
Figura 35. Imágenes seleccionadas para el entrenamiento. [Autor].....	39
Figura 36. Imágenes seleccionadas para la validación. [Autor].....	40
Figura 37. Visualización de los resultados del entrenamiento de la CNN. [Autor] ...	40
Figura 38. Verificación de la instalación de SDK. [Autor].....	42
Figura 39. Ejemplificación del archivo main.dart. [Autor].....	43
Figura 40. Ejemplificación de StatelessWidget. [Autor]	44
Figura 41. Ejemplificación de StatefulWidget. [Autor].....	45
Figura 42. Ejemplificación del método build. [Autor]	45
Figura 43. Ejemplificación de la clase CustomFileSystemStorage [Autor]	46
Figura 44. Función index. [Autor].....	46
Figura 45. Ejemplificación del backend y frontend que conforman la aplicación web. [Autor].....	47
Figura 46. Imagen almacenada en el dataset. [Autor]	48
Figura 47. Imagen detectada. [Autor].....	49
Figura 48. Despliegue de la aplicación web en el servidor. [Autor]	50
Figura 49. Verificación de ejecución del servidor. [Autor]	50
Figura 50. Interfaz de la aplicación web. [Autor].....	51
Figura 51. Interfaz de la aplicación web. [Autor].....	51
Figura 52. Interfaz de la aplicación web. [Autor].....	52
Figura 53. Interfaz de la aplicación web para el análisis de macroinvertebrados, con énfasis en aquellos correspondientes a la clase Crustácea. [Autor].....	52
Figura 54. Interfaz de la aplicación web para el análisis de macroinvertebrados, con énfasis en aquellos correspondientes a la clase Gastropoda. [Autor].....	53
Figura 55. Interfaz de la aplicación web para el análisis de macroinvertebrados, con énfasis en aquellos correspondientes a la clase Insecta de agua limpia. [Autor]	54

Figura 56. Interfaz de la aplicación web para el análisis de macroinvertebrados, con énfasis en aquellos correspondientes a la clase Insecta de agua muy contaminada. [Autor].....	54
Figura 57. Interfaz de la aplicación web para el análisis de macroinvertebrados, con énfasis en aquellos correspondientes a la clase Insecta de agua moderadamente contaminada. [Autor]	54
Figura 58. Interfaz de la aplicación web para el análisis de macroinvertebrados, con énfasis en aquellos correspondientes a la clase Oligochaeta. [Autor]	55
Figura 59. Interfaz de la aplicación web para el análisis de macroinvertebrados, con énfasis en aquellos correspondientes a la clase Turbellaria. [Autor].....	55
Figura 60. Imagen perteneciente al dataset. [Autor].....	56
Figura 61. Interfaz de la aplicación móvil. [Autor]	56
Figura 62. Gráfico correspondiente a la precisión en el entrenamiento y la validación (izquierda). Gráfico correspondiente a la pérdida en el entrenamiento y la validación (derecha). [Autor]	59
Figura 63. Gráfico correspondiente a la precisión en el entrenamiento y la validación (izquierda). Gráfico correspondiente a la pérdida en el entrenamiento y la validación (derecha). Implementando técnicas de mejoramiento en la red como es data Augmentation y Dropout. [Autor]	60
Figura 64. Gráfico correspondiente a la precisión en el entrenamiento y la validación (izquierda). Gráfico correspondiente a la pérdida en el entrenamiento y la validación (derecha). [Autor]	62
Figura 65. Gráfico correspondiente a la precisión en el entrenamiento y la validación (izquierda). Gráfico correspondiente a la pérdida en el entrenamiento y la validación (derecha). Implementando técnicas de mejoramiento en la red como es data Augmentation y Dropout. [Autor]	64
Figura 66. Gráfico correspondiente a la precisión en el entrenamiento y la validación (izquierda). Gráfico correspondiente a la pérdida en el entrenamiento y la validación (derecha). [Autor]	66
Figura 67. Gráfico correspondiente a la precisión en el entrenamiento y la validación (izquierda). Gráfico correspondiente a la pérdida en el entrenamiento y la validación (derecha). Implementando técnicas de mejoramiento en la red como es data Augmentation y Dropout. [Autor]	68

Figura 68. Gráfico de la matriz de confusión correspondiente al conjunto de datos utilizado en el entrenamiento. [Autor].....	69
Figura 69. Resumen del texto que incluye la precisión, recuperación y puntuación F1 para cada clase de macroinvertebrados analizados en el conjunto de datos de entrenamiento [Autor].....	69
Figura 70. Gráfico de la matriz de confusión correspondiente al conjunto de datos utilizado en el testeo. [Autor].....	71
Figura 71. Resumen del texto que incluye la precisión, recuperación y puntuación F1 para cada clase de macroinvertebrados analizados en el conjunto de datos de testeo. [Autor].....	71
Figura 72. Vista frontal del prototipo del diseño de un robot móvil utilizado para evaluar las condiciones actuales del agua. [Autor].....	73
Figura 73. Vista lateral del prototipo del diseño de un robot móvil utilizado para evaluar las condiciones actuales del agua. [Autor].....	73
Figura 74. Vista superior del prototipo del diseño de un robot móvil utilizado para evaluar las condiciones actuales del agua. [Autor].....	74

INDICE DE TABLAS

Tabla 3.1 Parámetros ingresados para el análisis de la CNN.	57
Tabla 3.2 Implementación de parámetros en el modelo secuencial para el análisis de la CNN.	58
Tabla 3.3 Parámetros ingresados implementando técnicas de mejoramiento en la red como es Data Augmentation y Dropout.	59
Tabla 3.4 Configuración-Data Augmentation	59
Tabla 3.5 Implementación de parámetros adicionando (dropout), en el modelo secuencial para el mejoramiento de la CNN.	59
Tabla 3.6 Parámetros ingresados para el análisis de la CNN.	61
Tabla 3.7 Implementación de parámetros en el modelo secuencial para el análisis de la CNN.	61
Tabla 3.8 Parámetros ingresados implementando técnicas de mejoramiento en la red como es Data Augmentation y Dropout.	62
Tabla 3.9 Configuración-Data Augmentation	63
Tabla 3.10 Implementación de parámetros adicionando (dropout), en el modelo secuencial para el mejoramiento de la CNN.	63
Tabla 3.11 Parámetros ingresados para el análisis de la CNN.	65
Tabla 3.12 Implementación de parámetros en el modelo secuencial para el análisis de la CNN.	65
Tabla 3.13 Parámetros ingresados implementando técnicas de mejoramiento en la red como es Data Augmentation y Dropout.	67
Tabla 3.14 Configuración-Data Augmentation	67
Tabla 3.15 Implementación de parámetros adicionando (dropout), en el modelo secuencial para el mejoramiento de la CNN.	67

RESUMEN

Este proyecto tiene como finalidad desarrollar un sistema para detectar la calidad del agua haciendo uso de los macroinvertebrados acuáticos, que habitan a lo largo del río Sinincay, algunos tramos pertenecientes a este río se consideran poco contaminados, medianamente contaminados y muy contaminados, siguiendo dos métodos de análisis (ICA-NSF) y (BMWP/Col), se determinó el nivel de contaminación del agua y se clasificó en siete grupos a las diferentes especies de macroinvertebrados que habitan en esos sectores. Se consideró como un tema de tesis ya que es beneficioso determinar la calidad de agua de un determinado río, evitando de esta manera que las personas ingieran, y posteriormente presenten enfermedades producidas a causa de parásitos, además se pretende mitigar desde cierto punto la contaminación y sus posteriores consecuencias.

Se consideró el uso de los macroinvertebrados acuáticos como organismos de análisis, ya que absorben en sus cuerpos los cambios que ha ocurrido en el lugar donde habitan. Con las imágenes obtenidas en los diferentes tramos del río, se procede a ingresar a una red neuronal, se optó por el desarrollo de una red neuronal convolucional, ya que es la óptima para este caso de análisis, para reducir el sobreajuste se procede a realizar métodos de mejoramiento en la red como es, data Augmentation y dropout, reduciendo considerablemente el sobreajuste.

Posteriormente se analiza los resultados obtenidos en la precisión y la pérdida tanto en los datos correspondientes al entrenamiento y la validación, si son óptimos se exporta el modelo en formato h5, luego se procede a hacer uso del modelo h5 junto con las técnicas de visión por computador “opencv” para el debido análisis y clasificación de las imágenes, en la implementación se hace uso del IDE VS Code.

Una vez realizado el análisis mencionado anteriormente, Se visualizará en la aplicación web las imágenes correspondientes a los macroinvertebrados acuáticos con una descripción breve de sus características más representativas, mediante la aplicación móvil también se puede identificar las especies de macroinvertebrados acuáticos a las que pertenece la imagen captada por la cámara.

INTRODUCCIÓN

El acceso a agua potable y la preservación de ecosistemas acuáticos saludables son desafíos cruciales en la actualidad. El río Sinincay, ubicado en Cuenca, es un importante recurso hídrico utilizado para el abastecimiento de agua potable, entre otros usos. Sin embargo, su calidad se ha visto amenazada debido a diversas actividades humanas.

La calidad del agua se evalúa tradicionalmente mediante análisis químicos y físicos, los cuales son costosos, requieren tiempo y no siempre proporcionan una imagen completa de la salud del ecosistema acuático. Por otro lado, los macroinvertebrados acuáticos, son organismos ampliamente reconocidos como indicadores biológicos de la calidad del agua. Su presencia, abundancia y diversidad pueden revelar información valiosa sobre la salud y la contaminación del ecosistema acuático.

En este contexto, se propone el desarrollo de un sistema innovador basado en técnicas de visión por computador para determinar la calidad del agua en la cuenca del río Paute. El objetivo principal es utilizar algoritmos de visión por computador para identificar y clasificar macroinvertebrados acuáticos presentes en este río. Una vez conseguidas las imágenes se procesarán utilizando técnicas de reconocimiento de patrones y aprendizaje automático.

La ventaja de este enfoque radica en su capacidad para automatizar el proceso de evaluación de la calidad del agua, permitiendo obtener resultados más rápidos, precisos y consistentes en comparación con los métodos tradicionales.

Este proyecto de investigación combina conocimientos multidisciplinarios en las áreas de ecología, procesamiento de imágenes y aprendizaje profundo. Se espera que los resultados obtenidos contribuyan al entendimiento y conservación de los ecosistemas acuáticos en la cuenca del río Paute, así como a la toma de decisiones informada en relación con el uso sostenible de los recursos hídricos.

ANTECEDENTES DEL PROBLEMA DE ESTUDIO

Actualmente en el Ecuador la importancia que se le otorga a la contaminación de fuentes hídricas como ríos, lagos y lagunas es mínima. De acuerdo con los resultados de la investigación realizada en el artículo, “Principios fundamentales en torno a la calidad del agua, el uso de bioindicadores acuáticos y la restauración ecológica fluvial en el Ecuador”, se puede observar que en el país existe desentendimiento tanto de las personas como de las autoridades en cuanto a nivel de recursos humanos como recursos económicos destinados a realizar dichos estudios.[1] [2]

En el transcurso de los años las autoridades a cargo se han encaminado en generar proyectos cuyo enfoque es abastecer de líquido vital a ciudades, cantones, parroquias o lugares rurales, más no se han centrado en analizar la calidad y los niveles de contaminación que la misma puede presentar.

De acuerdo con el estudio realizado por Solanes, en el libro la gobernabilidad efectiva del agua en las américas, uno de los limitantes es la falta de recursos económicos y poco conocimiento, lo que conlleva a que sea mínimo el interés por mitigar la contaminación de las fuentes de agua. [3]

Realizando un análisis más aproximado de la realidad en la que vivimos, se puede examinar la situación actual de Quito, una ciudad con más de 2 millones de habitantes, la misma que desde hace más de 20 años presenta contaminación en los ríos. Este problema se debe a la falta de capacidad para tratar las aguas contaminadas que producen a cada segundo los ciudadanos y las diferentes empresas. Actualmente los representantes de la entidad encargada para el manejo de las aguas contaminadas indican que dan tratamiento solamente al 3.5% de estas aguas, ya que existen únicamente siete plantas pequeñas, según estudios realizados se estima que Quito necesita 22 plantas de tratamiento para lograr tratar todas las aguas contaminadas de la ciudad. Una pequeña cantidad de agua contaminada de Quito confluye en el alcantarillado, el resto de agua desemboca en los ríos Guayllabamba, Machángara, Monjas, San Pedro, Santa Clara y Pita. [4] [5]

A nivel nacional Cuenca, Loja cuentan con sistemas de tratamiento de aguas residuales, esto ha permitido que los ríos de estas ciudades presenten pocos parásitos, las personas que beben esta agua presentan una cantidad muy reducida de

enfermedades intestinales, también va en beneficio de las especies acuáticas ya que habitan en gran cantidad en estos riachuelos.

De acuerdo con el estudio citado en el artículo “principios fundamentales en torno a la calidad del agua”, se ha demostrado que una de las formas más exitosas y económicas para determinar el nivel de contaminación y el tipo de contaminante presente en algún afluente es trabajando con macroinvertebrados acuáticos, ya que son considerados excelentes bioindicadores biológicos. Estos organismos presentan cambios ya sea en su morfología, distribución, tamaño de las agrupaciones que conviven, conducta cuando el ecosistema en el que habitan es alterado o contaminado.[1]

Cada grupo de macroinvertebrados acuáticos muestra su hábitat de preferencia, algunos optan por ocupar lugares fangosos, arcillosos, arenosos, zonas donde la corriente es fuerte o débil, por ende, todos estos indicadores como es la cantidad, preferencia, conducta, permiten identificar las condiciones en las que se encuentra el sector analizado del afluente. [6]

JUSTIFICACIÓN (IMPORTANCIA Y ALCANCES)

Previo al desarrollo de la tesis es necesario tener en cuenta que se abordará acerca de los tipos de macroinvertebrados acuáticos, por eso es útil conocer su clasificación para su posterior recolección y análisis respectivo.

Este proyecto está considerado como un tema de tesis, ya que es beneficioso determinar la calidad que posee el agua de un determinado río, evitando de esta manera que las personas ingieran, y posteriormente presenten enfermedades producidas a causa de parásitos, además se pretende mitigar desde cierto punto la contaminación y sus posteriores consecuencias, para ello se empleará herramientas digitales que permitan recoger información para posteriormente emitir alertas sobre el nivel de contaminación de un arroyo.

Según análisis realizados sobre la contaminación del río Guayas y sus afluentes se determinó que la principal causa de contaminación se debe a la desembocadura de agua no tratada de las empresas industriales y aguas residuales de diferentes hogares, las mismas que contienen residuos de fertilizantes y plaguicidas. [7]

Actualmente no existen datos estadísticos consolidados, que informen acerca del agua contaminada que llega a los diferentes afluentes y a su vez desembocan en el mar, según los datos extraídos de los informes regionales relacionados a la contaminación de áreas costeras en América Latina se determinó que en Chile el 82% del total de las aguas residuales llegan al mar a través de veinte y siete cuencas hidrográficas, estos afluentes captan desechos domésticos e industriales, de la misma manera se determinó que en México existen 210 cuencas hidrográficas, en 20 de ellas, se genera el 89% de los contaminantes totales de la región. [8]

En base a estudios preliminares y documentos citados en los antecedentes el análisis de la contaminación de los ríos se realiza haciendo uso de los macroinvertebrados acuáticos, debido a que son organismos que absorben en sus cuerpos los cambios que ha ocurrido en el lugar donde habitan.

OBJETIVOS

OBJETIVO GENERAL

Desarrollar un sistema para la identificación de macroinvertebrados acuáticos como indicadores de calidad de agua en los ríos, utilizando técnicas de visión por computador.

OBJETIVOS ESPECÍFICOS

Realizar un estudio del arte de la identificación de macroinvertebrados acuáticos utilizando las técnicas de visión por computador a través de la investigación de publicaciones científicas y bibliografía especializada a fin de seleccionar la técnica más adecuada.

Desarrollar el software e integrar el algoritmo que permitirá identificar los macroinvertebrados acuáticos y la calidad del agua en los ríos, mediante el uso de OpenCV.

Realizar un análisis sobre los límites de operación mediante las pruebas de contrastación con métodos y sistemas de identificación tradicionales en la determinación de la calidad de agua en los ríos a fin de determinar la fiabilidad del sistema desarrollado.

CAPÍTULO 1: ESTADO DEL ARTE

1.1 Fundamentos teóricos.

Bioindicadores

Los bioindicadores son organismos vivos, que se utilizan para determinar las condiciones actuales del agua analizada, pueden ser plantas, animales o microorganismos.

Cada uno de ellos tienen características específicas que les permiten reaccionar ante cambios en el medio ambiente, como es la contaminación, la alteración en el suelo o el cambio climático.

Se puede deducir, que los bioindicadores son organismos vivos o características de un ecosistema que se utilizan para medir la calidad ambiental. Su presencia o ausencia, así como su condición, son factores que indican el estado de un ecosistema y la influencia de las acciones humanas en el mismo. El bioindicador es una herramienta valiosa para la evaluación y el monitoreo de la salud ambiental ya sea a nivel local o global. [9]

Principios de la bioindicación

Los efectos que produce un determinado contaminante en un ecosistema acuático es evidente, existen diferentes parámetros que son analizados para determinar el grado de contaminación del medio ambiente entre ellos se tiene:

Especificidad: Cada organismo responde a un conjunto específico de factores ambientales, por lo tanto, la selección de especies bioindicadores debe basarse en su capacidad para reflejar la presencia o ausencia de un factor ambiental particular.

Sensibilidad: Los organismos bioindicadores deben ser sensibles a cambios en el factor ambiental que se está monitoreando.

Rapidez: proporcionar una respuesta rápida a los cambios en el ambiente. Algunos organismos responden a los cambios en el ambiente en cuestión de horas, mientras que otros tardan días o incluso semanas.

Fiabilidad: se refiere a la capacidad de un bioindicador para proporcionar resultados precisos y consistentes.

Representatividad: se refiere a la capacidad de un bioindicador para ser distintivo de un área amplia o de un ecosistema completo.

Organismos utilizados como bioindicadores

Los principales organismos que son empleados como indicadores biológicos son los siguientes:

Bacterias:

Este tipo de bioindicadores bacterianos son de mucha utilidad al momento de monitorear la calidad del agua, aire o suelo, así como para detectar contaminantes tóxicos y evaluar la salud de los ecosistemas.

Las bacterias son utilizadas como bioindicadores por su alto grado de sensibilidad a ciertas condiciones ambientales, algunas bacterias son indicadoras de la presencia de contaminantes químicos ya sea en el suelo o en el agua.

La presencia de la bacteria coliformes fecales, es un indicativo de que el agua está contaminada con materia fecal, esta puede ser humana o animal, lo que resultaría perjudicial en la salud de los seres humanos. [10]

Protozoos:

Son organismos unicelulares, que habitan en una variedad de ambientes acuáticos y terrestres; resultan ser excelentes indicadores de la presencia de contaminantes y las condiciones actuales del agua analizada ya que son calidad del agua, ya que son organismos que presentan cambios en su cuerpo al estar expuestos en el agua.

Además, su presencia y cantidad indican la actividad microbiana y la calidad del suelo, lo cual es muy importante para la producción agrícola y la conservación del suelo. [11]

Fitoplancton:

Son utilizados principalmente como bioindicadores, ya que son sensibles a los cambios en el ecosistema acuático. La abundancia del fitoplancton en el agua indica la presencia de contaminantes o nutrientes. [12]

Macroinvertebrados acuáticos:

Son considerados como organismos acuáticos, notoriamente visibles, no poseen columna vertebral, ejemplo las larvas de insectos, moluscos y crustáceos.

Los macroinvertebrados acuáticos son sensibles a cambios en el ambiente, como la contaminación, el cambio de temperatura o la alteración del hábitat, y pueden ser utilizados como indicadores del estado actual del ecosistema acuático.

Son excelentes indicadores de la actividad microbiana y la calidad del suelo, lo que es importante para la producción agrícola y la conservación del territorio. Además, la presencia de ciertas especies de macroinvertebrados en el agua puede indicar que existencia de contaminantes provenientes del aire.



Figura 1. Grupo Odonata (Libélulas) en ninfas y adulto, perteneciente a los macroinvertebrados. [13]

Teniendo en cuenta la incidencia de los contaminantes en estas especies, existen otros fenómenos como el clima, la geografía, que afectan notoriamente la densidad poblacional de estas especies, estos parámetros deben ser detectados a tiempo para que no influya en la interpretación de los datos. [14]

Peces:

Son utilizados como bioindicadores debido a su gran sensibilidad a los cambios en el ecosistema acuático. Los peces pueden acumular y metabolizar contaminantes, esto incide en su salud y comportamiento, proporcionando detalles sobre la presencia y la cantidad de contaminantes existentes en el agua.

Los peces pueden utilizarse para evaluar una amplia variedad de factores ambientales, por ejemplo, condiciones actuales del agua, temperatura, oxígeno presente en el agua, salinidad, la presencia de contaminantes.

1.2 Macroinvertebrados acuáticos que habitan en la cuenca del río paute.

A través de investigaciones llevadas a cabo en el río Yanuncay, que forma parte de la cuenca del río Paute, se determinó que existen las siguientes clases de macroinvertebrados acuáticos, el estudio fue realizado en los meses de junio, julio y septiembre. [15]

Oligochaeta:

Es una clase de gusanos anélidos que incluye a las lombrices de tierra. Se caracterizan por tener un cuerpo segmentado y órganos internos básicos, entre ellos un tubo digestivo y un sistema circulatorio. Las lombrices de tierra son los miembros más conocidos de la clase Oligochaeta, desempeñan una función crucial en la ecología del suelo, al descomponer los componentes orgánicos, incrementando la fertilidad del suelo.



Figura 2. Macroinvertebrados pertenecientes a la clase Oligochaeta. [15]

Insecta:

Los insectos constituyen un grupo diverso y abundante de animales que poseen tres segmentos del cuerpo (cabeza, tórax y abdomen), seis patas, antenas y alas. Pueden ser herbívoros, carnívoros u omnívoros, algunas especies son importantes polinizadores de plantas.



Figura 3. Macroinvertebrados pertenecientes a la clase Insecta [15]

Gastropoda:

Es una clase de moluscos que incluye caracoles, babosas. Estos animales se caracterizan por tener un solo caparazón en espiral o en algunos casos, no tener ningún caparazón. Se pueden encontrar en varios entornos, incluidos entornos marinos, de agua no salada y terrestres. Algunas especies de gasterópodos también son importantes por su capacidad para limpiar la materia vegetal y animal muerta, lo que ayuda a mantener los ecosistemas saludables.



Figura 4. Macroinvertebrados pertenecientes a la clase Gastropoda. [15]

Crustácea:

Los crustáceos juegan papeles ecológicos importantes, ya que contribuyen en el equilibrio de los ecosistemas marinos y de agua dulce.



Figura 5. Macroinvertebrados pertenecientes a la clase Crustácea. [15]

Turbellaria:

Son animales acuáticos, viven en ecosistemas marinos, de agua dulce y terrestres.

Los turbelarios se caracterizan por su cuerpo aplanado y blando. Los turbelarios son importantes depredadores y carroñeros en los ecosistemas acuáticos.



Figura 6. Macroinvertebrados pertenecientes a la clase Turbellaria. [15]

Puntos de monitoreo pertenecientes a la cuenca del río Paute seleccionados para analizar la calidad del agua.

El estudio se efectuó en el río Sinincay y se consideraron ocho puntos esenciales para monitorear la calidad del agua, el primer punto analizado se ubica en el sector denominado “Cochas”.

Otros puntos analizados son: sector “El Chorro”, Centro Parroquial, Daniel Durán, Chamana, Las Orquídeas, Camino a Patamarca, Parque Industrial.

En el sector “Cochas” el agua se considera ligeramente contaminada, este sector se encuentra en la parte inicial del río y existe poca intervención humana, los sectores de “El Chorro”, “Daniel Durán”, “Chamana”, “Las Orquídeas”, son lugares que se catalogan con agua moderadamente contaminada, mientras tanto que “Camino a Patamarca” y el “Parque Industrial” presentan una calidad de agua muy contaminada, análisis que tiene su fundamento ya que estos dos puntos finales analizados reciben el agua proveniente de los otros puntos, en estos lugares existe la presencia de desechos humanos, los mismos que desembocan en el río Sinincay, sumado a esto, la arena, el lodo, incrementa la turbidez del agua.

El período en el cual se realizó el monitoreo comprende los meses de junio, julio, septiembre del año 2021, correspondiendo cada mes a época lluviosa, seca y transición. [15]

1.3 Inteligencia artificial

La IA es una de las ciencias modernas dentro de la informática, imita la inteligencia humana, mediante la implementación de algoritmos. Es considerado como un conjunto de tecnologías y métodos que permiten a las máquinas que tengan su propio aprendizaje a la par de inferir, planificar, percibir, comunicarse y tomar decisiones sin intervención humana directa. La IA se fundamenta en la idea de que las máquinas o robots son capaces de configurarse para que realicen tareas, que habitualmente demandarían la intervención de una o más personas, estas tareas incluyen la identificación de patrones, la toma de decisiones en tiempo muy rápido, el análisis y gestión de datos así como la automatización de procesos en múltiples sectores industriales.

Tipos de inteligencia artificial

Según investigaciones se llega a determinar que existen dos clases en la actualidad de inteligencia artificial:

Inteligencia Artificial Estrecha

Se le conoce también como inteligencia artificial débil, cuyo objetivo es atender a tareas no complejas y en tiempo real, respondiendo con información aprendida de un conjunto de datos concretos, ejemplos muy específicos de IA estrecha son Siri, AlphaGo y los vehículos autónomos, estos son ejemplos claros de que realizan tareas específicas y responden a preguntas poco abstractas.

Inteligencia Artificial General

Actualmente, no se han creado máquinas con inteligencia general artificial completa para aprender, percibir, comprender y funcionar como humanos. Estas máquinas son muy importantes porque pueden realizar actividades exigentes o muy peligrosas para los humanos. Además, estas máquinas serían sensibles, conscientes y estarían impulsadas emocionalmente.

Hoy en día, las máquinas con inteligencia artificial pueden diseñar estrategias, pensar de manera abstracta, pero aún no existen máquinas que tengan la capacidad de razonamiento, resolución de problemas, planificación, aprendizaje y tener su propia imaginación y creatividad.

Superinteligencia Artificial

La inteligencia que es mucho más avanzada que el cerebro humano en todos los sentidos, desde la creatividad científica hasta las habilidades sociales, se denomina superinteligencia artificial. El desarrollo de ASI se considera el mayor logro en IA porque puede replicar la inteligencia compleja y multidisciplinaria de los humanos. También sobresale en todos los ámbitos con su memoria masiva, procesamiento de datos más rápido y capacidades de toma de decisiones.

Máquinas reactivas

Denominada como la clasificación más elemental de la IA. Es decir, no llegan a formar recuerdos ni tienen la capacidad de basarse en historias del pasado para realizar tareas o tomar decisiones de una manera óptima. Tienen la capacidad de responder solamente a las tareas simples para las que están programados.

Su particularidad es que siempre se comporta según lo programado no acepta modificaciones.

Memoria limitada

Estos sistemas son capaces de utilizar datos históricos para tomar decisiones futuras, actualmente las aplicaciones existentes desarrolladas con IA se basan en el concepto de memoria limitada. El deep learning es un componente esencial en el campo de la inteligencia artificial.

Teoría de la mente

Se refiere al hecho de que las máquinas tienen la capacidad de expresar estados mentales humanos como creencias, intenciones y deseos. Dos ejemplos destacados que cumplen con ciertos parámetros de la teoría de la mente son Kismet y Shopia.

Kismet es un robot humanoide que puede reconocer rasgos faciales humanos y reproducir esas emociones. Shopia, por otro lado, es un robot humanoide avanzado que se asemeja físicamente a los humanos, y su reconocimiento de imagen avanzado le permite interactuar con los humanos respondiendo adecuadamente a las expresiones faciales.

Ventajas de la inteligencia artificial

Automatización: La Inteligencia artificial es de gran ayuda ya que en diferentes campos industriales se emplea para automatizar actividades repetitivas ahorrando mano de obra y agilizando la producción para cumplir con los plazos establecidos.

Eficiencia: Es de gran utilidad la IA ya que aumenta la eficiencia en distintos campos, destacándose en campos como la medicina, la ingeniería y la manufactura industrial, su capacidad de operar con grandes cantidades de datos, permite la ejecución de tareas complejas en periodos de tiempo cortos.

Precisión: La IA tiene la capacidad de realizar tareas cuya precisión supera la capacidad humana, esto resulta de mucha utilidad en sectores industriales y proyectos de ingeniería que requieren exactitud en sus procesos.

Personalización: La IA se adapta a los gustos y necesidades individuales, permitiéndonos personalizar productos y servicios para cada cliente.

Seguridad mejorada: La IA puede ayudar a mejorar la seguridad y prevenir amenazas y ataques maliciosos en muchas áreas, desde la seguridad del hogar hasta la ciberseguridad.

Investigación avanzada: La IA puede acelerar el proceso de investigación y descubrimiento en diversos campos, actualmente ya se emplea en campos como la medicina, la biociencia o biotecnología.

Desventajas de la inteligencia artificial

Aunque tiene muchas ventajas, también hay que destacar que existen algunos inconvenientes. Para utilizar la IA de una manera muy sensata y en pro de la sociedad, es importante considerar estas deficiencias y abordarlas de manera adecuada.

Dependencia: si los humanos se vuelven demasiado dependientes de la IA para realizar sus tareas, pueden perder sus habilidades y conocimientos básicos.

Privacidad y seguridad: la capacidad de la IA para almacenar y analizar volúmenes grandes de datos genera preocupación en la sociedad, ya que son vulnerables a ataques de hackers.

Campos en donde se aplica la inteligencia artificial

La inteligencia artificial tiene aplicabilidad en diferentes campos, por ejemplo:

Detección y clasificación de objetos: Este es el proceso que implica el uso de algoritmos y modelos de aprendizaje en la identificación y clasificación de objetos. Este proceso se puede realizar utilizando diferentes enfoques de aprendizaje automático, por ejemplo la detección de caracteres en las imágenes analizadas, la división de imágenes y el reconocimiento de patrones. Esta función es aplicable en varias industrias como la automotriz, médica, biotecnológica y de automatización de procesos.

Visualización de preferencias de contenido en redes sociales: El uso de esta herramienta en la actualidad se considera en auge, facilitando a los usuarios de las diferentes redes sociales buscar información de su preferencia. Para lograr aquello se emplea técnicas sofisticadas de aprendizaje automático y análisis de datos, en la primera fase se recopila información como reacciones a publicaciones, comentarios emitidos por el usuario, una vez extraída la información se procede a usar algoritmos de aprendizaje automáticos para extraer información relevante, de esta manera se sabe las preferencias de los usuarios, posteriormente se procede a visualizar los datos empleando técnicas avanzadas por ejemplo se emplea mapas de color, diagramas de red facilitando a los usuarios su comprensión.

Seguridad cibernética: es empleada para la detección de anomalías en las entidades bancarias, como las transferencias de cantidades económicas considerables. Además, pueden ser entrenados los sistemas de inteligencia artificial para detectar ataques cibernéticos y también pueden alertar sobre posibles ataques futuros.

Área médica: La lógica difusa es una técnica de datos tolerante a la ambigüedad que captura y explota el concepto de fuzzy de una manera computacional muy eficiente, lo que la hace ideal para aplicaciones médicas. Las técnicas de lógica difusa se han explorado en diversas aplicaciones médicas. Por ejemplo, el análisis de regresión logística múltiple se usa para diagnosticar el cáncer de pulmón y también se usa para diagnosticar la leucemia aguda y el cáncer de mama.

Aplicación a data sets contables: La integración de sistemas inteligentes con bases de datos contables ayuda a explorar grandes cantidades de datos con o sin la participación gerencial directa. Por lo tanto, el sistema analiza los datos y permite al usuario

comprender o interpretar las transacciones para determinar qué eventos contables está capturando el sistema.

Inteligencia artificial aplicada a los juegos de computadora: los juegos de computadora constituyen una de las aplicaciones más populares de la tecnología informática. Existen varias maneras en que la IA contribuye a los juegos de ordenador. Algunos de los más notorios incluyen el movimiento de unidades, la percepción simulada, el análisis situacional, el razonamiento espacial, el aprendizaje, la coordinación de grupos, la asignación de recursos, la orientación, la agrupación y la selección de objetivos. Las técnicas de inteligencia artificial utilizadas son búsqueda de rutas, redes bayesianas, lógica difusa y algoritmos genéticos.

Aplicación en la Robótica: consiste en un campo dentro de la inteligencia artificial, que emplea dispositivos mecánicos, generalmente controlados por computadoras, para realizar tareas que requieren una precisión extrema o involucran trabajo que es peligroso para los humanos. Actualmente, la robótica se considera el área más interesante de la tecnología de IA y, de hecho, el área de mayor beneficio para la humanidad. El uso de robots industriales para realizar tareas repetitivas con precisión ha mejorado la productividad de las líneas de montaje en las plantas de fabricación.

1.4 Machine learning

Dentro de la inteligencia artificial existe una rama denominada machine learning o aprendizaje automático, la misma que tiene como principal función crear algoritmos que puedan auto aprender a partir de datos iniciales proporcionados.

El aprendizaje automático se centra en el concepto que las máquinas se basen en modelos e información histórica de ejemplos anteriores para mejorar sus habilidades de resolución de problemas cuando se les proporcionan datos. Por esta razón, se cree que los modelos de aprendizaje automático inician de un dataset de datos observados y derivan reglas de clasificación o patrones de comportamiento que se aplican a datos distintos a los utilizados para el análisis.

Componentes del aprendizaje automático:

Considerando esa definición se puede englobar los distintos componentes del aprendizaje automático:

Fuentes de información considerada materia prima del machine learning los conjuntos de datos por ejemplo las imágenes, grabaciones de audio, estos datos se emplean en el entrenamiento y posteriormente para testear los modelos de aprendizaje automático.

Algoritmos de aprendizaje: son programas informáticos empleados en el entrenamiento, ajuste y calibración de modelos de aprendizaje automático

Existen tres tipos de aprendizaje automático:

Aprendizaje supervisado: en esta clase de aprendizaje automático, los algoritmos son receptores de un conjunto de datos ya organizados, lo que permite indicar como debería ser categorizada la nueva información.

1. Algoritmo de clasificación: este tipo de algoritmo determina a qué grupo pertenece el elemento analizado, ya que encuentra patrones en los datos proporcionados y los clasifica en grupos. Posteriormente compara los nuevos datos y los ubica en el conjunto perteneciente. Existe dos etapas en los algoritmos de clasificación:

Entrenamiento:

El algoritmo de clasificación recibe un conjunto de datos, en donde cada conjunto de datos de datos presenta características únicas. El algoritmo de clasificación analiza los datos de entrenamiento e intenta encontrar patrones en los atributos que ayuden a predecir etiquetas para nuevas instancias de datos. Existen varios algoritmos de clasificación:

Árboles de decisión:

Consiste en un método de aprendizaje automático supervisado, que se encuentra constituido por un nodo llamado raíz. Desde el nodo raíz, se crean nodos secundarios o nodos internos que representan preguntas adicionales sobre atributos de datos. Cada nodo secundario se divide en dos o más ramas, dependiendo de las respuestas posibles a la pregunta planteada, hasta culminar en el nodo hoja que es el nodo que tiene la decisión final.

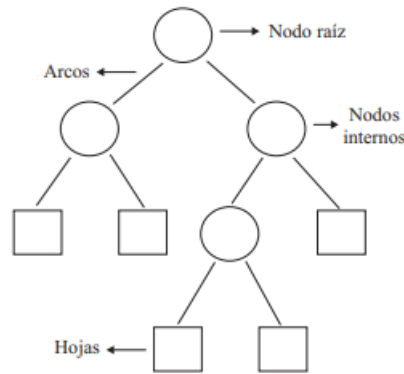


Figura 7. Árboles de decisión [17]

Regresión Logística:

Consiste en un método que se basa en un dígito binario dependiente este puede ser 0/1 lo que significaría (verdadero/falso), además consta de una o más variables independientes, es muy utilizado en el análisis de datos.

k-vecinos más cercanos:

El algoritmo de k-vecinos(neighbors) más cercanos es una técnica de aprendizaje automático, este método consiste en la determinación de pertenencia a un determinado grupo un conjunto de datos desconocidos.

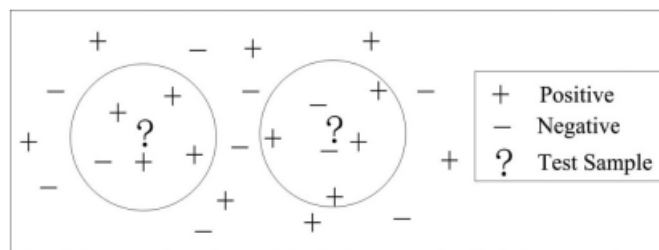


Figura 8. Clasificación binaria mediante el método k-NN con un valor de k=5 [18]

El algoritmo k-NN es simple y fácil de implementar, y puede ser utilizado para clasificar datos en diferentes campos, por ejemplo, en la medicina, la biología, imagenología, minería de datos, entre otros.

Teoría de Bayes:

La teoría bayesiana es una herramienta matemática fundamental en el aprendizaje supervisado, que se implementa en la predicción de datos futuros, es decir un algoritmo

recibe un data set y se somete a un proceso de entrenamiento. Este tipo de aprendizaje supervisado se basa en una función que toma los datos y proporciona como resultado las etiquetas correspondientes, este método implica que la información se actualice constantemente permitiendo ajustar nuestras afirmaciones sobre un dato específico.

Máquinas de vectores de soporte (SVM):

Este algoritmo de aprendizaje automático tiene la capacidad para clasificar los datos en diferentes grupos. Este tipo de algoritmo tiene la capacidad de clasificar características en conjuntos grandes, es muy útil en aplicaciones como la clasificación de imágenes, detección de spam. .

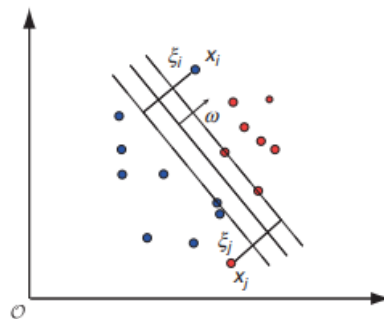


Figura 9. Máquina de vectores de soporte: hiperplanos de decisión lineal en un espacio de características. [19]

Redes Neuronales:

Las neuronas poseen un cuerpo circular con múltiples ramificaciones. El axón dirige la señal de salida proveniente de las dendritas de otra célula. Las dendritas son las que conducen y hacen que la información llegue hacia otra neurona.

Las neuronas se consideran la base de nuestra capacidad de aprender, por lo que este comportamiento es lo que se pretende imitar en términos computacionales para procesar y aprender a partir de los datos. Simulando el comportamiento de las neuronas se asigna un peso numérico a cada conexión neuronal. Las redes neuronales identifican dos estados 0 cuando no existe la presencia de un impulso y 1 cuando existe la presencia de un impulso. La característica más relevante de las redes neuronales es que tienen la capacidad de operar con conjuntos de datos de muchas características, detentando patrones complejos.

Las redes neuronales son muy flexibles y pueden ser utilizadas en una amplia gama de aplicaciones de aprendizaje supervisado, como la clasificación de imágenes, la detección de fraudes y la predicción en los valores de los costos de las acciones. Una de las ventajas de las redes neuronales es que pueden operar conjuntos de datos con muchas características y detectar patrones complejos en los datos.

Tipos de redes neuronales:

Redes Neuronales Feedforward:

En esta red neuronal los datos ingresan a través de las neuronas situadas en la capa de ingreso, la información avanza hacia la siguiente capa denominada oculta, donde la neurona receptora recibe toda la información que llega hacia a ella, cada enlace existente significa un peso y cuando un estímulo supera la magnitud establecida, la información se procesa.

Las redes neuronales recurrentes se emplean en casos de análisis que implican la traducción de idiomas, reconocimiento de caracteres en textos escritos a mano.

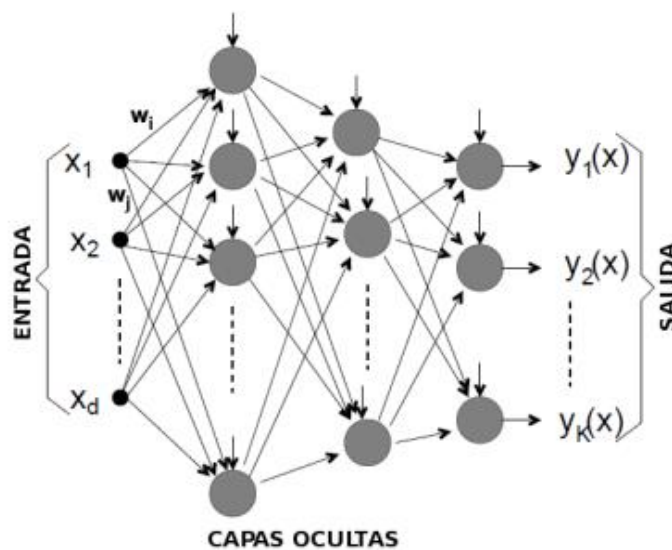


Figura 10. Estructura de una red neuronal artificial feedforward. [19]

Redes Neuronales Recurrente (RNN):

Se caracterizan por tener la capacidad de utilizar información previa en análisis actuales. En este tipo de red neuronal, las neuronas se establecen en varias capas, las conexiones entre neuronas en la mayoría de los casos van en una sola dirección, pero también existe conexiones bidireccinales.

Existe un tipo de red neuronal recurrente denominada LSTM, aquí las neuronas están organizadas en capas recurrentes, poseen un mecanismo de puerta con lo cual controlan el flujo de información que atraviesa la capa.

Las puertas en una red LSTM son tres:

Puerta de entrada (Input Gate): esta puerta permite controlar la cantidad de información que se va a otorgar en la memoria a largo plazo.

Puerta de olvido (Forget Gate): esta puerta permite controlar la cantidad de información que se elimina de la memoria a largo plazo.

Puerta de salida (Output Gate): esta puerta permite contralar la cantidad de información que se va a remitir a la capa siguiente.

Redes Neuronales Convolucionales (CNN):

Este tipo de red utiliza varias capas convolucionales, además se utiliza capas de agrupamiento, funciones de activación, todo esto en conjunto con la finalidad de obtener características relevantes de las imágenes analizadas. Son utilizadas para la detección de características en imágenes médicas, además son muy útiles en aplicaciones relacionadas con la visión por computador, como por ejemplo en el reconocimiento y la clasificación de imágenes.

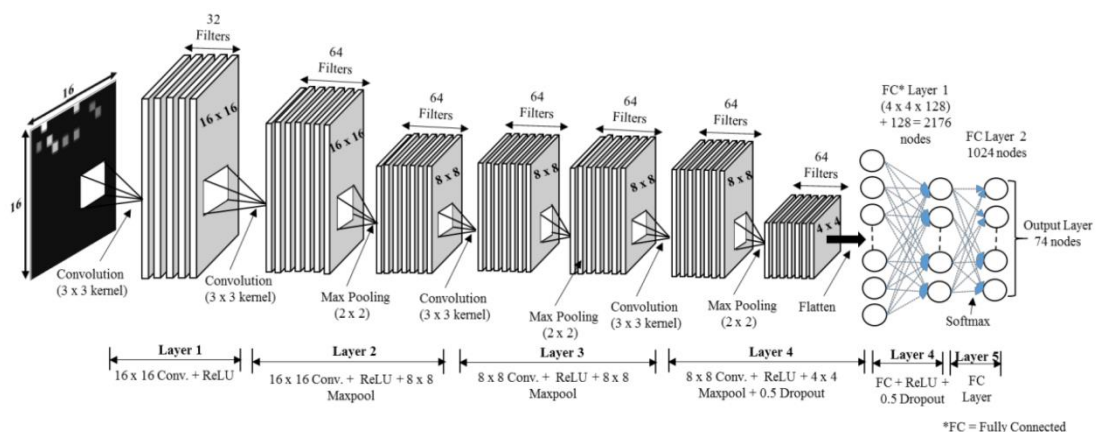


Figura 11. Red Neuronal Convolutiva.[19]

En la red neuronal convolutiva de la figura 2.3 se observa una red compuesta por seis capas, en la primera capa se observa imágenes de entrada en escala de grises de 16x16x1 con una unidad lineal rectificadora (ReLU), como existe un pequeño grupo de datos al ingreso, la primera capa no utiliza agrupación máxima (MaxPooling). En la

segunda capa se tiene una convolución de 16x16 con activación ReLu y una capa de agrupación máxima de 8x8 teniendo 18496 parámetros, produciendo una salida para la tercera capa de convolución de 8x8 con activación ReLu y MaxPooling de 8x8, la cuál produce una salida que alimenta a la cuarta capa que es una capa de convolución de 8x8 con activación ReLu y una capa de agrupación máxima de 8x8, con dropout de 0.5. Esta salida alimenta directamente a una capa fully connected con 2176 nodos, lo que conduce a la siguiente capa oculta FC con 1024 nodos. Finalmente, la salida se calcula utilizando una capa softmax con 74 nodos. La tasa de aprendizaje del modelo CNN es 0,001 y el número total de parámetros es 233.418.

Convolución: se refiere a la aplicación de un filtro o kernel a una imagen colocada para el análisis respectivo con la finalidad de obtener características relevantes.

La imagen de ingreso puede una imagen en escala de grises o una imagen a color, en el desarrollo de la tesis se tiene la segunda opción, en la siguiente imagen se presenta una imagen a color con las siguientes características (224x224x3) lo que significa el alto, ancho y los canales existentes en la imagen, aquí las capas convolucionales usan filtros o kernels para procesar los datos de ingreso. Este filtro se mueve a lo largo de la imagen de ingreso y en cada ubicación del filtro, se realiza una operación de convolución que produce un número. Posteriormente este valor se introduce en una función de activación y el resultado de la función de activación se conoce como mapa de activación que en la imagen será (224x224x1).

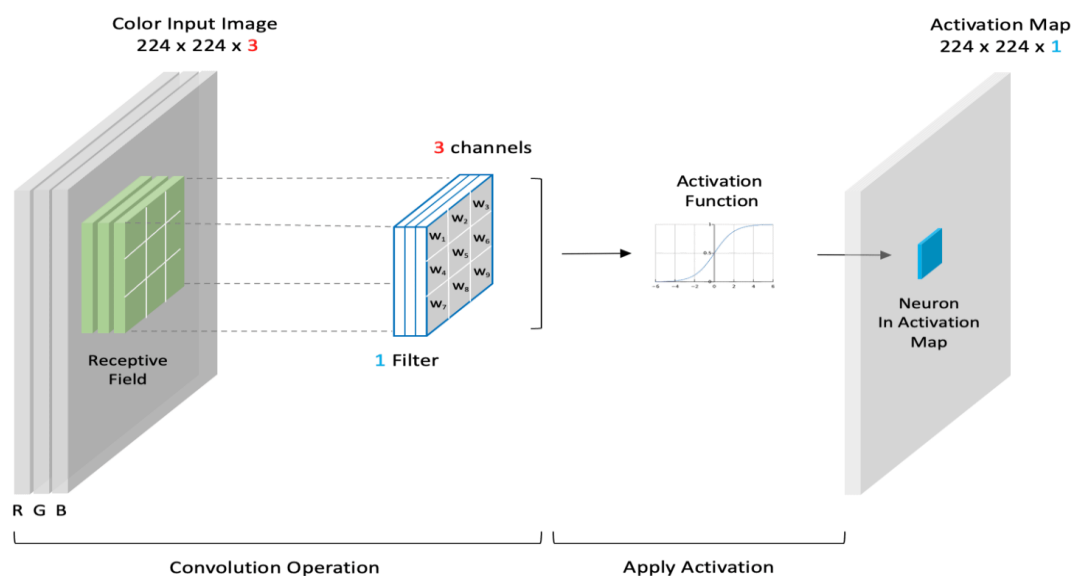


Figura 12. Explicación detallada de las capas convolucionales.[19]

La operación de convolución se realiza de la siguiente manera: teniendo una capa convolucional se utiliza un pequeño filtro para procesar los datos que ingresan. En el siguiente ejemplo se tiene un segmento de entrada de tamaño (6x6), el cual se convoluciona con un kernel (3x3), en este ejemplo se utiliza un kernel Sobel, que es empleado para detectar bordes verticales.

En una CNN, la convolución se realiza deslizando el filtro sobre la imagen de entrada y calculando el producto punto a punto entre los datos obtenidos por el filtro y los datos correspondientes de la región de la imagen cubierta por el filtro, posteriormente estos productos se suman y se coloca el resultado en una nueva matriz conocida como mapa de características.

El tamaño del filtro suele ser más pequeño que la imagen de entrada, lo que permite extraer características locales. A medida que se realiza la convolución en la imagen de entrada, se obtiene un conjunto de mapas de características, cada uno de los cuales representa la respuesta del filtro a una característica específica en la imagen.

En el gráfico a continuación, se presenta un ejemplo en donde se observa cada ubicación del filtro, la operación produce un solo número colocado en la ubicación correspondiente.

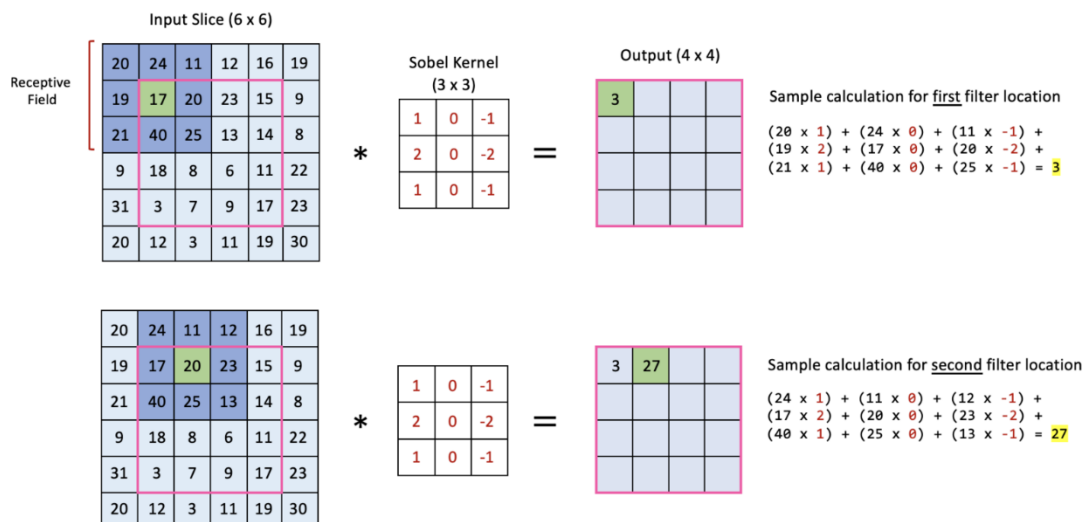


Figura 13. Operación de convolución.[19]

Capas de agrupación máxima: Maxpooling es una capa muy empleada en las redes neuronales convolucionales tiene el objetivo de reducir el tamaño espacial de las

representaciones y disminuir el número de parámetros en la red, preservando las características más relevantes de la imagen de entrada. [16]

La operación de maxpooling tiene varios parámetros importantes:

Tamaño del filtro o ventana: se define el tamaño de la ventana que se desliza sobre la entrada. Es común utilizar ventanas de tamaño 2x2 o 3x3.

Stride (desplazamiento): especifica el número de píxeles que se mueve la ventana en cada paso. Un stride de valor 2 reducirá a la mitad el tamaño de la salida en comparación con la entrada.

Padding: determina si se debe agregar o no ceros alrededor del borde de la entrada antes de aplicar el maxpooling.

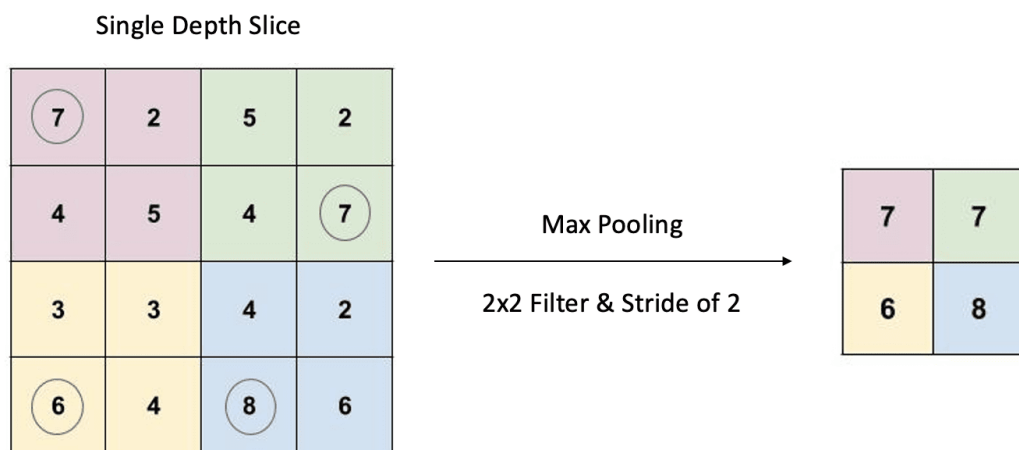


Figura 14. Explicación de Max Pooling.[19]

La agrupación es una forma de reducción de tamaño que utiliza un filtro deslizante 2D. El filtro pasa por el corte de entrada de acuerdo con un parámetro configurable llamado zancada. Se conoce como zancada al número de píxeles que el filtro se mueve a través del segmento de entrada de una posición hacia la siguiente. En la figura se aprecia que se tiene un segmento de entrada 4x4, un filtro de 2x2 y un paso de dos, por ende, la salida correspondiente es una de 2x2 que es la representación reducida del segmento de entrada.

Capa densa : Las capas completamente conectadas se caracterizan ya que trabajan con los datos transformados en un vector plano, este tipo de capa es de gran utilidad ya que al tratar a los datos como un vector unidimensional tiene la capacidad de aprender

características más abstractas, en este tipo de capas se puede utilizar técnicas para reducir el sobreajuste en el entrenamiento de la red, entre ellas tenemos dropout y regularización.

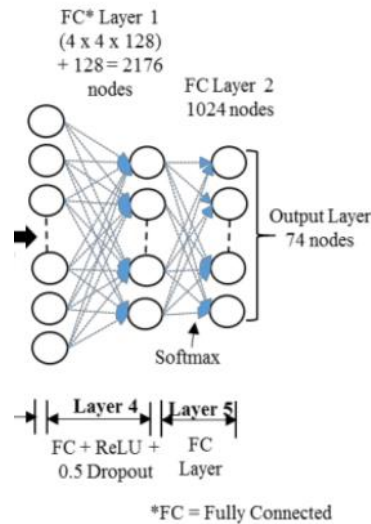


Figura 15. Fully connected layer.[19]

Funciones de activación: Son componentes esenciales dentro de una red neuronal convolucional ya que forzan a la red para que aprenda patrones más complejos al introducir parámetros no lineales

Función de activación Relu: esta función es de mucha utilidad en las CNN ya que la red aprende a detectar patrones de gran complejidad , es utilizada dentro de las capas ocultas su finalidad es validar la información que se asemeja a uno y rechazar la información cercana a cero, en la actualidad es la función de activación que más se emplea en la creación de modelos de redes neuronales secuenciales.

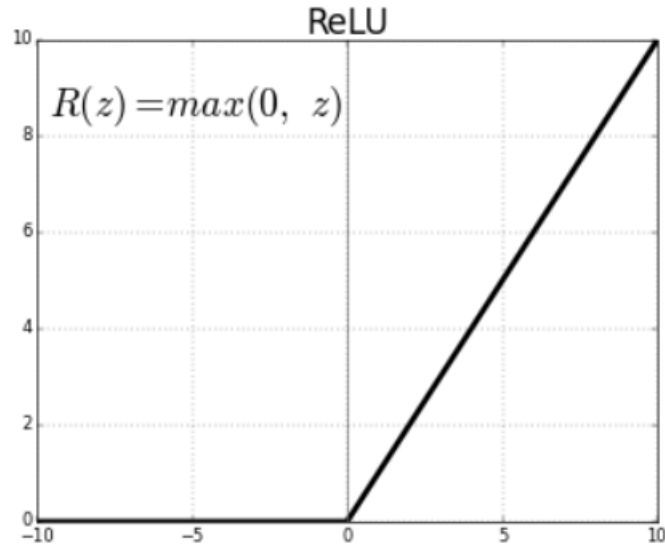


Figura 16. Función ReLU. [19]

Predicción:

El algoritmo de predicción consiste en utilizar el modelo de aprendizaje automático entrenado para predecir a que conjunto pertenece determinada agrupación de datos, se realiza el proceso de predicción luego de haber entrenado un data set específico y de visualizar el resultado de las clases a las que pertenecen determinado conjunto de datos, en algunos casos se tiene varias clases en un solo conjunto entrenado, esto depende del algoritmo creado para el entrenamiento. Cabe mencionar que el resultado de la predicción es variable dependiendo de los datos ingresados y del algoritmo seleccionado para el entrenamiento, además se debe tomar en cuenta el modelo empleado, las capas de entrenamiento utilizadas y los parámetros introducidos para evitar el sobreajuste de los datos de entrenamiento.

2. Algoritmo de regresión:

Hace referencia a un modelo matemático, el cual funciona de la siguiente manera: se tiene datos de ingreso, puede ser imágenes u otro parámetro y se utilizan en la predicción de valores continuos. Son muy útiles en diferentes campos por ejemplo para la predicción de precios de domicilios, predecir ventas comerciales. Existen los siguientes algoritmos de regresión:

Regresión Lineal:

En este algoritmo de regresión el modelo tiene la capacidad de determinar el resultado obtenido al trabajar con las variables dependientes en función de las variables independientes. Este modelo es muy útil para determinar la relación existente entre las variables.

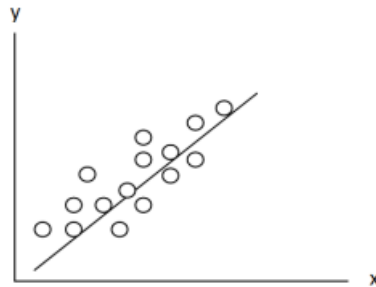


Figura 17. Regresión Lineal.[19]

Árboles de decisión:

Son definidos como modelos de predicción, su finalidad es captar la mayor información durante la etapa de aprendizaje. Este modelo está estructurado en forma de árbol, ramificado por nodos, hojas y ramas.

Aprendizaje no supervisado: se caracteriza ya que el algoritmo recibe un data set sin etiquetar, debe encontrar patrones y estructuras de forma autónoma para posteriormente clasificar la información. El objetivo es descubrir información relevante, identificar conjuntos de datos similares y agrupar los datos, se caracteriza ya que tiene la capacidad para detectar patrones significativos por sí mismo. Estas técnicas empleadas son:

Clustering:

Consiste en la agrupación de datos similares entre sí, en un solo conjunto. Para agrupar datos similares en clusters es necesario que compartan características equivalentes. Los grupos obtenidos estarán compuestos por individuos análogos teniendo en cuenta que la distancia entre ellos es muy pequeña.

Modelos de aprendizaje automático:

Se llama así a la representación que se le da a un conjunto de datos, los modelos creados utilizan algún determinado algoritmo de aprendizaje, será elegido el más

conveniente según el caso de estudio, estos modelos deben ser ajustados para que cumplan con las predicciones establecidas.

Optimización:

En esta etapa se procede a realizar ajustes en los parámetros del modelo con la finalidad de reducir la función de pérdida.

Validación:

En esta etapa se realiza la evaluación del modelo de aprendizaje automático, se tiene los datos pertenecientes al entrenamiento y a la validación, los mismos que son seccionados y los datos de entrenamiento son utilizados para el entrenamiento, mientras que los datos seleccionados para el testeo son utilizados para la evaluación de la precisión del modelo creado.

1.5 Deep learning

Es una división del aprendizaje automático se enfoca principalmente en el análisis y reconocimiento de patrones. Además, se caracterizan, ya que tienen la capacidad para auto aprender en base a los datos iniciales dados, son muy útiles en casos de análisis, en donde se requiere alta precisión de los resultados, por ejemplo, se emplean con frecuencia en análisis de características en las imágenes, procesamiento de voz, entre otras aplicaciones. Estos modelos de aprendizaje son capaces de procesar conjuntos significativos de datos, consumiendo a su vez recursos computacionales en grandes proporciones, como lo es la unidad de procesamiento de gráficos.

El Deep Learning tiene aplicabilidad en diferentes ámbitos, por ejemplo:

Visión por computadora: se utiliza ampliamente en la visión por computador en casos de análisis que requieran la identificación de parámetros en las imágenes.

Procesamiento del lenguaje natural: El Deep Learning se utiliza en el procesamiento del lenguaje natural para tareas como el reconocimiento de voz, la traducción automática, la clasificación de texto y la generación de texto.

Robótica: El Deep Learning se utiliza en robótica para la percepción del medio, la planificación de movimientos y el control de diferentes articulaciones en los robots.

1.6 Técnicas de visión por computador

Esta rama de la inteligencia artificial se especializa en desarrollar algoritmos que tengan la capacidad de identificar, interpretar y analizar parámetros en las imágenes o videos similar a la manera en la que procede una persona. Existen varias etapas del proceso de visión por computadora para el análisis de parámetros en las imágenes, inicialmente se procede a capturar imágenes, la calidad de las imágenes es de gran importancia, ya que de esto dependerá la eficacia en los resultados, luego se procede a eliminar el ruido.

Posteriormente se extrae las características más importantes de las imágenes, las características como color y la forma, para después clasificar y etiquetar las imágenes. Consecuentemente se procede a reconocer diferentes parámetros en las imágenes como rostros, identificar objetos específicos, finalmente se toman decisiones en función de la información extraída, en algunos casos los resultados se visualizan mediante gráficos.

Campos de aplicación de la visión por computador.

Se aplica en varios sectores industriales, algunos incluyen la robótica, la seguridad y vigilancia, la medicina, la industria manufacturera, la publicidad, entre otros.

Reconocimiento facial:

Identificación de personas en imágenes o videos. Esto implica la detección y extracción de características del rostro, para luego comparar con el data set con el objetivo de tomar decisiones basadas en la comparación de particularidades.

Detección de objetos:

Identificación y seguimiento de objetos en movimiento. La detección de objetos con visión por computador es una tarea importante en muchas aplicaciones, como la vigilancia, el control de calidad y la conducción autónoma.

Realidad aumentada:

Consiste en una tecnología que implica la combinación de la información real con objetos virtuales, para lograr esta combinación se emplean diferentes métodos de procesamiento de imágenes, redes neurales, entre otros algoritmos.

1.7 Servidor web

Consiste en un software, el mismo que se ejecuta en un servidor, para hacer uso de un servidor web se procede de la siguiente manera: el usuario o cliente digita una dirección web en el navegador, este a su vez envía una solicitud al servidor web correspondiente. El servidor web llega a procesar la solicitud, busca los archivos necesarios para construir la página y los envía de retorno al navegador en forma de código etiquetado como HTML, CSS, JavaScript, u otros.

Además de alojar y servir páginas web, un servidor web puede realizar otras tareas, como alojar y enviar correos electrónicos, almacenar y distribuir archivos, y alojar aplicaciones web complejas. En este proyecto de tesis en el servidor web se manejarán archivos.

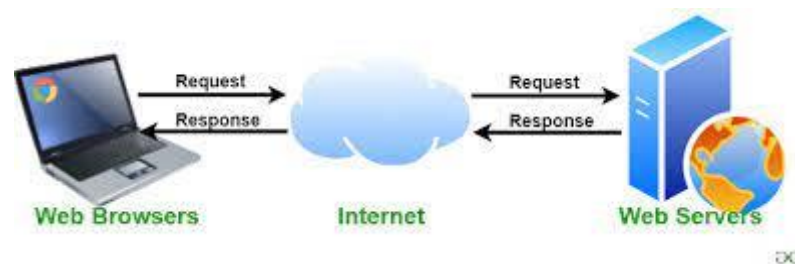


Figura 18. Ejemplificación del funcionamiento de un servidor web. [22]

1.8 Funcionalidades de un servidor web

1. Escuchar peticiones: el servidor web está constantemente escuchando las peticiones que llegan desde los navegadores web a través del protocolo HTTP. Cuando una solicitud es recibida, el servidor web la procesa y envía la respuesta correspondiente al navegador.
2. Manejo de conexiones: El servidor web se encarga de gestionar todas las conexiones que se establecen con los clientes.
3. Administración de archivos: El servidor web almacena los archivos que componen las páginas web (HTML, CSS, imágenes, etc), y los sirve a los navegadores web cuando estos los solicitan. Además, se encarga de gestionar la caché de los archivos para mejorar la velocidad de acceso a los mismos.

1.9 Protocolo http

Es un protocolo de la capa de comunicación, denominado Transferencia de Hipertexto (HTTP) es utilizado para transferir datos en la World Wide Web (www).

El protocolo HTTP consta de varios elementos clave, incluyendo:

1. Métodos de solicitud: HTTP define varios métodos de solicitud, como GET, PUT, POST, DELETE, que se utilizan para solicitar diferentes acciones al servidor web.

GET: se utiliza para pedir un recurso determinado al servidor web. Cuando se envía una solicitud GET, el navegador web solicita una copia del recurso especificado y el servidor web responde enviando la copia del recurso.

POST: se utiliza para el envío de datos desde el navegador web al servidor web, como información de un formulario.

PUT: se utiliza para actualizar un recurso específico en el servidor web. Cuando se envía una solicitud PUT, el navegador web envía los datos que deben ser actualizados al servidor web. Si el recurso especificado no existe, se crea un nuevo recurso en el servidor web. El servidor web responde con una confirmación de que el recurso ha sido actualizado correctamente.

DELETE: se utiliza para llegar a eliminar un recurso específico del servidor web. Cuando se envía una solicitud DELETE, el navegador web solicita al servidor web que elimine el recurso especificado. El servidor web responde con una confirmación de que el recurso ha sido eliminado correctamente.

2. Códigos de estado: HTTP define códigos que indican la respuesta de alguna solicitud el más conocido es 404 Not Found (el recurso solicitado no existe o no se encontró). [17]

3. Encabezados: HTTP utiliza encabezados para brindar información adicional a las respuestas entregadas.

3. Cookies: HTTP utiliza cookies para almacenar información, permitiendo reconocer al cliente en visitas posteriores a dichas páginas web.

1.10 Backend

El "backend" se refiere al software de una aplicación. La sección del backend puede dividirse en tres componentes principales: el servidor, la base de datos y la lógica de negocio.

El servidor es el componente que recibe las solicitudes de cualquier tipo y procesa la información que se encuentra en la base de datos. La base de datos es el componente donde se almacenan los datos que se emplean en la aplicación. La lógica de negocio constituye un conjunto de normas que son empleadas al momento de procesar los datos almacenados en la base de datos.

1.11 Frontend

El frontend de una aplicación se refiere a lo visual de la aplicación que se ejecuta para que el usuario pueda interactuar. El frontend está compuesto principalmente de tres tecnologías: HTML, CSS y JavaScript.

HTML se clasifica como un lenguaje de etiquetado muy usado para estructurar el contenido de una página web. Posee una estructura la cual permite definir los elementos básicos de la página, tales como encabezados, párrafos, imágenes, enlaces.

CSS se utiliza para personalizar la página web, aplicando varios elementos visuales como lo es color, tamaño de la imagen, letra, entre otras propiedades visuales de la página.

JavaScript es un lenguaje de programación orientado a la web, proporciona funcionalidades a la página web, es decir se puede realizar animaciones a elementos que conforman la página web, entre otras funcionalidades.

1.12 Python

Python es un lenguaje de programación que se utiliza ampliamente en la industria, la investigación y la educación debido a su facilidad de uso, legibilidad y sintaxis clara. En Python, los programas se escriben en texto plano utilizando un editor de texto y luego se ejecutan mediante un intérprete de Python.

Python se destaca por su capacidad para administrar estructuras de datos complejas y realizar tareas de análisis de datos, aprendizaje automático y procesamiento de

imágenes. También es requerido en la creación de aplicaciones web, juegos y aplicaciones de escritorio. Una de las características más útiles de Python es su amplia biblioteca de módulos predefinidos que proporcionan funcionalidades adicionales a los programas.

1.13 Flutter

Flutter es un framework creado y potenciado por Google, usado en el desarrollo de aplicaciones móviles que permite crear aplicaciones multiplataforma de alta calidad desde una única base de código. Con su enfoque en widgets, rendimiento nativo, hot reload y una comunidad activa, Flutter es una gran opción para desarrolladores que buscan crear aplicaciones móviles rápidas, atractivas y altamente personalizables.

CAPÍTULO 2: MARCO METODOLÓGICO

2.1 Librerías utilizadas en la CNN

Instalación de librerías de Python

Las bibliotecas de Python requeridas en la red neuronal son las siguientes:

OPENCV

Se describe como una biblioteca relacionada con la visión por computadora, utilizada especialmente para el análisis de varios parámetros en las imágenes. Esta biblioteca es empleada en diferentes campos incluyendo la robótica, la automatización industrial, la vigilancia, la realidad aumentada, la automatización de procesos, entre otros. OpenCV proporciona una extensa variedad de algoritmos y funciones que facilitan la identificación de características, el reconocimiento de objetos, el fraccionamiento de imágenes, entre otras funciones.

```
C:\Windows\System32>pip show opencv-contrib-python
WARNING: Ignoring invalid distribution -pencv-python-headless (c:\users\maris\anaconda3\lib\site-packages)
Name: opencv-contrib-python
Version: 4.7.0.72
Summary: Wrapper package for OpenCV python bindings.
Home-page: https://github.com/opencv/opencv-python
Author:
Author-email:
License: Apache 2.0
Location: c:\users\maris\anaconda3\lib\site-packages
Requires: numpy, numpy, numpy
Required-by:
C:\Windows\System32>
```

Figura 19. Verificación de la instalación de OpenCV 4.7.0.72 en python (Windows). [Autor]

Dentro de Opencv se encuentran funciones que se han integrado en la creación de la aplicación web, tales como:

- OpenCV (GaussianBlur)

```
imgBlur = cv2.GaussianBlur(imagen,(7,7),1)
```

Figura 20. Función cv2.GaussianBlur. [Autor]

La función cv2.GaussianBlur() forma parte de la biblioteca OpenCV en Python, es empleada para la aplicación de un filtro Gaussiano a una determinada imagen. Este filtro suaviza la imagen, disminuyendo la presencia de ruido.

La sintaxis básica de la función `cv2.GaussianBlur()` es la siguiente:

“cv2.GaussianBlur(src, ksize, sigmaX[, dst[, sigmaY[, borderType]])”

`src`: El parámetro `src` se encarga de especificar la imagen de origen en la que se aplicará el filtro Gaussiano. Es necesario que se trate de una imagen en formato Numpy.

`ksize`: Es el tamaño del kernel o matriz de convolución que define la intensidad del suavizado. Debe ser un número impar positivo mayor que 1. Un valor más alto produce un suavizado más fuerte.

`sigmaX`: Es la desviación estándar en la dirección X. Controla cuánto se expande el filtro Gaussiano. Si se establece en 0, se calcula automáticamente a partir de `ksize`.

- OpenCV (`cvtColor`)

```
gray = cv2.cvtColor(imgBlur, cv2.COLOR_BGR2GRAY)
```

Figura 21. Función `cv2.cvtColor`. [Autor]

La función `cv2.cvtColor()` forma parte de la biblioteca OpenCV en Python empleada para transformar el espacio de color de una imagen, realizando conversiones entre formatos distintos. Esta función posibilita la conversión de una imagen a color en escala de grises o cambiar entre esquemas de color como RGB, BGR, HSV.

La sintaxis básica de la función `cv2.cvtColor()` es la siguiente:

“cv2.cvtColor(src, code[, dst[, dstCn]])”

`src`: Es el parámetro obligatorio que especifica la imagen de origen que se va a convertir.

`code`: Hace referencia al código de conversión que especifica el tipo de transformación de color que se llevará a cabo, como, por ejemplo:

`cv2.COLOR_BGR2GRAY` se emplea para convertir a escala de grises.

- OpenCV (`threshold`)

```
ret, thresh = cv2.threshold(gray, 50, 255, cv2.THRESH_BINARY)
```

Figura 22. Función `cv2.threshold`. [Autor]

La función `cv2.threshold()` está inmiscuida en la biblioteca OpenCV y es empleada para la realización de la operación de umbralización en una imagen. Este método consiste en la división de los píxeles de una determinada imagen en dos grupos, por lo general se divide en blanco y negro. Esta función devuelve dos parámetros, el primero llamado `retval`, representa el valor de umbral empleado y el segundo parámetro denominado `threshold` es la imagen resultante posterior a la aplicación de la umbralización.

“`retval, threshold = cv2.threshold(src, thresh, maxval, type [, dst])`”

La sintaxis básica de la función `cv2.threshold()` es la siguiente:

`src`: Es el parámetro obligatorio que especifica la imagen de origen en la que se aplicará la umbralización.

`thresh`: Este parámetro hace referencia al número de umbral que se colocará para catalogar a los píxeles que conforman la imagen. Los píxeles con un valor mayor al umbral se clasificarán en una categoría, mientras que los píxeles con un valor menor o igual al umbral se clasificarán en otra categoría.

`maxval`: Este parámetro consiste en la colocación de un número, el mismo que será asignado a los píxeles que excedan el umbral. Dependiendo del tipo de umbralización que se utilice, este valor puede ser diferente.

`type`: Este parámetro indica el tipo de umbralización que se empleará en la imagen, por ejemplo:

`cv2.THRESH_BINARY`: Se conoce así al umbral binario. Los píxeles superiores al umbral se asignan a `maxval`, mientras que los píxeles inferiores o iguales al umbral se determinan en 0.

- OpenCV (Canny)

```
img_canny = cv2.Canny(gray,33,22)
```

Figura 23. Función cv2.Canny. [Autor]

La función cv2.Canny() se utiliza para identificar bordes en una imagen mediante el uso del algoritmo de identificación de márgenes denominado Canny. Este algoritmo es muy aceptado debido a la exactitud que presenta para reducir el ruido presente en la imagen antes de detectar los bordes.

La sintaxis básica de la función cv2.Canny() es la siguiente:

image: Es el parámetro obligatorio que especifica la imagen de origen en la que se realizará la detección de bordes.

threshold1: Consiste en el primer parámetro que se emplea para determinar que píxeles se consideran como márgenes débiles.

threshold2: Consiste en el segundo parámetro que se emplea para determinar que píxeles se consideran como márgenes fuertes.

- OpenCV (dilate)

```
imgDil = cv2.dilate(img_canny,kernel,iterations=1)
```

Figura 24. Función cv2.dilate. [Autor]

La función cv2.dilate() se emplea para efectuar una operación de dilatación en la imagen analizada. Esta acción consiste en ampliar las áreas que poseen píxeles blancos en una imagen y se utiliza comúnmente para completar espacios vacíos, unir bordes y acrecentar las dimensiones de los objetos en una imagen.

La sintaxis básica de la función cv2.dilate() es la siguiente:

```
“dst = cv2.dilate(src, kernel [, dst[, anchor[, iterations[, borderType[, borderValue]]]])”
```

src: Es el parámetro obligatorio que especifica la imagen de origen en la que se aplicará la dilatación.

kernel: Es el kernel o elemento estructurante que define la forma y el tamaño de la dilatación.

- OpenCV (findContours)

```
contornos,hierarchy = cv2.findContours(gray,cv2.RETR_CCOMP,cv2.CHAIN_APPROX_NONE)
```

Figura 25. Función `cv2.findContours`. [Autor]

La función `cv2.findContours()` es parte de la biblioteca OpenCV, es empleada para para identificar y extraer los márgenes en una imagen. Esta función devuelve dos parámetros, el primero denominado `contours` consiste en una lista de los puntos que conforman los contornos hallados y el segundo parámetro denominado `hierarchy` es una matriz que almacena información sobre la estructura jerárquica de los contornos.

La sintaxis básica de la función `cv2.findContours()` es la siguiente:

```
“contours, hierarchy = cv2.findContours(image, mode, method [, contours [, hierarchy[, offset]]])”
```

`image`: Es el parámetro obligatorio que especifica la imagen binaria en la que se buscarán los contornos.

`mode`: Este parámetro hace referencia a la forma de devolución de los contornos identificados en una imagen. Por ejemplo:

`cv2.RETR_CCOMP`: Se devuelven todos los contornos en una lista jerárquica de dos niveles. En el primer nivel se encuentran los contornos externos, mientras que en el segundo nivel se hallan los contornos internos.

- OpenCV (contourArea)

```
area = cv2.contourArea(cnt)
```

Figura 26. Función `cv2.contourArea`. [Autor]

La función `cv2.contourArea()` forma parte de la biblioteca OpenCV y se emplea para calcular la extensión de la región encerrada por el contorno.

La sintaxis básica de la función `cv2.contourArea()` es la siguiente:

“area = cv2.contourArea(contour)”

contour: Es el parámetro obligatorio que especifica el contorno del cual se desea calcular el área. El contorno debe ser una lista de puntos que representan el contorno.

La función cv2.contourArea() devuelve el valor del área del contorno como un número de punto flotante.

- OpenCV (arcLength)

```
peri = cv2.arcLength(cnt, True)
```

Figura 27. Función cv2.arcLength [Autor]

La función cv2.arcLength() está inmiscuida en la biblioteca OpenCV, se emplea para calcular la longitud de una curva. Esta función devuelve el valor de la longitud de la curva como un dato de punto flotante.

La sintaxis básica de la función cv2.arcLength() es la siguiente:

“length = cv2.arcLength(curve, closed)”

curve: Es el parámetro obligatorio que especifica el contorno o la curva de la cual se desea calcular la longitud.

closed: Es un dato booleano que cumple la función de mostrar si la curva es cerrada o abierta. Si se establece en True, se considera que la curva es cerrada y se calcula la longitud del contorno completo. Si se establece en False, se considera que la curva es abierta y se calcula la longitud de la curva.

- OpenCV (approxPolyDP)

```
approx = cv2.approxPolyDP(cnt, 0.02*peri, True)
```

Figura 28. Función cv2.approxPolyDP [Autor]

La función cv2.approxPolyDP() está inmiscuida en la biblioteca OpenCV, empleada para aproximar un contorno poligonal con menos puntos manteniendo su forma general.

La sintaxis básica de la función cv2.approxPolyDP() es la siguiente:

`“approx = cv2.approxPolyDP(curve, epsilon, closed)”`

curve: Es el parámetro obligatorio que especifica el contorno o la curva de la cual se desea realizar la aproximación. El contorno o la curva deben ser una lista de puntos que representan la forma.

epsilon: Es un parámetro que especifica la precisión de aproximación deseada. Un valor más pequeño de *epsilon* produce una aproximación más precisa, mientras que un valor más grande produce una aproximación más generalizada.

closed: Es un dato booleano que cumple la función de mostrar si la curva es cerrada o abierta. Si se establece en True, se considera que la curva es cerrada y se realizará una aproximación cerrada. Si se establece en False, se considera que la curva es abierta y se realizará una aproximación abierta.

- OpenCV (boundingRect)

```
x,y,w,h = cv2.boundingRect(approx)
```

Figura 29. Función *cv2.boundingRect* [Autor]

La función *cv2.boundingRect()* se utiliza para deducir el rectángulo delimitador más pequeño que logra sujetar totalmente un contorno.

La sintaxis básica de la función *cv2.boundingRect()* es la siguiente:

`“x, y, width, height = cv2.boundingRect(points)”`

points: Es el parámetro obligatorio que especifica el contorno o la lista de puntos del cual se desea calcular el rectángulo delimitador.

La función *cv2.boundingRect()* devuelve cuatro valores: *x*, *y*, *width* y *height*. Estos valores representan la coordenada (*x*, *y*) del vértice superior izquierdo del rectángulo delimitador, así como la anchura y la altura del rectángulo.

- OpenCV (rectangle)

```
cv2.rectangle(imagen, (x,y), (x+w,y+h), (0,255,0),3)
```

Figura 30. Función *cv2.rectangle*. [Autor]

La función `cv2.rectangle()` está inmiscuida dentro de la biblioteca OpenCV, es empleada para dibujar un rectángulo en una imagen o en una matriz NumPy.

La sintaxis básica de la función `cv2.rectangle()` es la siguiente:

“cv2.rectangle(image, pt1, pt2, color, thickness)”

`image`: Es el parámetro obligatorio que especifica la imagen o la matriz NumPy en la cual se desea dibujar el rectángulo.

`pt1`: Es el parámetro que especifica las coordenadas del vértice superior izquierdo del rectángulo.

`pt2`: Es el parámetro que indica las coordenadas del vértice inferior derecho del rectángulo.

`color`: Es el parámetro que especifica el color del rectángulo.

`thickness`: Es el parámetro que especifica el grosor del contorno del rectángulo. Si se establece en un valor negativo o igual a -1, se dibujará un rectángulo relleno.

La función `cv2.rectangle()` modifica la imagen o la matriz NumPy directamente y dibuja el rectángulo especificado.

- OpenCV (`putText`)

```
cv2.putText(imagen, "Insecto" + str(len(approx)), (x+w+20, y+20), cv2.FONT_HERSHEY_COMPLEX, .7, (255, 0, 0), 2, cv2.LINE_AA)
```

Figura 31. Función `cv2.putText`. [Autor]

La función `cv2.putText()` está inmiscuida en la biblioteca OpenCV y se emplea para incorporar texto en una imagen o en una matriz NumPy.

La sintaxis básica de la función `cv2.putText()` es la siguiente:

“cv2.putText(image, text, org, fontFace, fontScale, color, thickness)”

`image`: Es el parámetro que especifica la imagen o la matriz NumPy en la cual se desea insertar el texto.

`text`: Es el parámetro que detalla el texto que se desea insertar.

org: Es el parámetro que especifica las coordenadas del punto de inicio del texto.

fontFace: Es el parámetro que especifica el tipo de fuente del texto.

fontScale: Es el parámetro que especifica la escala del tamaño de la fuente.

color: Es el parámetro que especifica el color del texto. Puede ser una tupla (B, G, R) que represente los valores de intensidad de color azul, verde y rojo, respectivamente.

thickness: Es el parámetro que especifica el grosor del texto. Puede ser un valor entero que indica el grosor del contorno del texto. Si se establece en un valor negativo, el texto se dibujará con un contorno relleno.

NUMPY

Hace referencia a una biblioteca que proporciona una extensa gama de funciones matemáticas, las mismas que son utilizadas en los arreglos. Esta biblioteca facilita la gestión eficiente de grandes cantidades de datos mediante el uso de arrays que es la estructura de datos propia de esta biblioteca.

```
C:\Windows\System32>pip show numpy
WARNING: Ignoring invalid distribution -pencv-python-headless (c:\users\maris\anaconda3\lib\site-packages)
Name: numpy
Version: 1.21.5
Summary: NumPy is the fundamental package for array computing with Python.
Home-page: https://www.numpy.org
Author: Travis E. Oliphant et al.
Author-email:
License: BSD
Location: c:\users\maris\anaconda3\lib\site-packages
Requires:
Required-by: astropy, bkcharts, bokeh, Bottleneck, daal4py, datashader, datashape, gensim, h5py, holoviews, hvplot, imagecodecs, imageio, Keras-Preprocessing, matplotlib, mkl-fft, mkl-random, numba, numexpr, opencv-contrib-python, opencv-contrib-python-headless, opencv-python, opt-einsum, pandas, patsy, pyerfa, PyWavelets, scikit-image, scikit-learn, scipy, seaborn, statsmodels, tables, tensorboard, tensorflow, tifffile, xarray
```

Figura 32. Verificación de la instalación de Numpy 1.21.5 en python (Windows). [Autor]

KERAS

Esta biblioteca es empleada para proyectos en donde se usa el aprendizaje profundo, haciendo uso de una interfaz de programación de aplicaciones que permite entrenar modelos de aprendizaje profundo, instituyendo varias capas de red neuronal y enlazándolas entre sí.

```
C:\Windows\System32>pip show keras
WARNING: Ignoring invalid distribution -pencv-python-headless (c:\users\maris\anaconda3\lib\site-packages)
Name: keras
Version: 2.9.0
Summary: Deep learning for humans.
Home-page: https://keras.io/
Author: Keras team
Author-email: keras-users@googlegroups.com
License: Apache 2.0
Location: c:\users\maris\anaconda3\lib\site-packages
Requires:
Required-by: tensorflow
C:\Windows\System32>
```

Figura 33. Verificación de la instalación de Keras 2.9.0 en python (Windows). [Autor]

2.2 Diseño de la arquitectura

La realización de este proyecto consta de varias etapas que se detallarán a continuación:

Adquisición de datos: Las imágenes de macroinvertebrados acuáticos, recolectadas para realizar el análisis mediante una red neuronal, fue tomado en diferentes tramos del río Sinincay, el cual pertenece a la cuenca del río Paute.

Back End: Las imágenes recolectadas se utilizan para entrenar la CNN, una vez entrenada la red mediante tensorFlow y Python, se analiza los resultados tanto de precisión como en pérdida en los conjuntos de datos correspondientes al entrenamiento y la validación, si son óptimos se exporta el modelo en formato h5, posteriormente se procede a hacer uso del modelo h5 junto con las técnicas de visión por computador “opencv” para proceder el análisis y clasificación de las imágenes. También en la implementación de este proceso, se utiliza el IDE VS Code.

Front End: Se visualizará en la aplicación web las imágenes correspondientes a los macroinvertebrados acuáticos con una descripción breve de sus características más representativas, mediante la aplicación móvil también se puede identificar las especies de macroinvertebrados acuáticos a las que pertenece la imagen captada por la cámara. Además, utilizando el software inventor se realizará un prototipo de un robot donde se implementará la CNN en un pequeño ordenador y juntamente con una cámara, tendrá la capacidad de explorar, captar, analizar e identificar al macroinvertebrado.

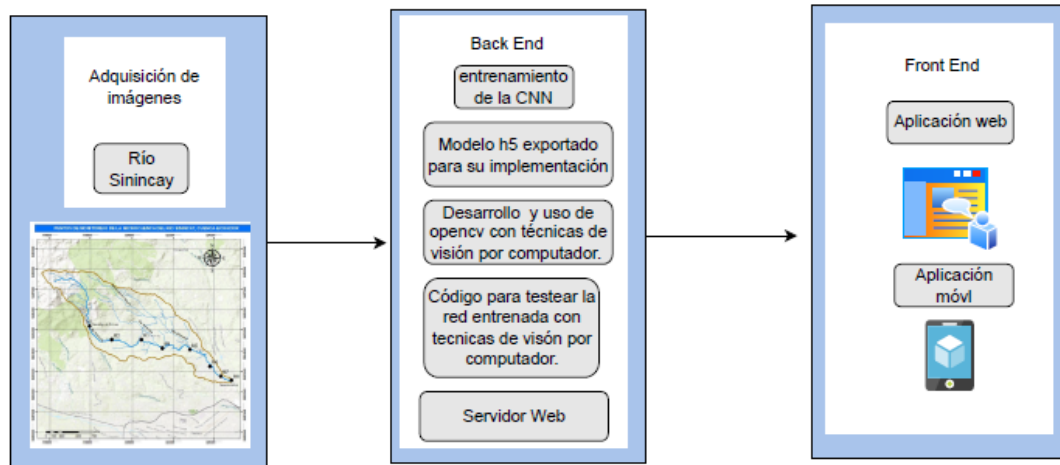


Figura 34. Diseño de la arquitectura del proyecto. [Autor]

2.3 Desarrollo del código para el entrenamiento de la CNN

1. Importación de librerías (matplotlib, numpy, os, PIL, tensorflow, pathlib, cv2), necesarias en la implementación de la CNN.
2. Cargar el dataset, el cual está conformado por 5002 imágenes, dimensionadas a una altura de 220 y un ancho de 220, las imágenes son reconocidas en formato png.
3. Seleccionar el 80% de las imágenes del dataset para el entrenamiento, esto corresponde a 4002 imágenes.

Mediantela función `tf.keras.utils.image_dataset_from_directory` se crea un conjunto de datos mediante la inclusión de archivos de imagen en un directorio, segmentándose el conjunto de datos total en 7 agrupaciones de imágenes.

```
Found 5002 files belonging to 7 classes.
Using 4002 files for training.
```

Figura 35. Imágenes seleccionadas para el entrenamiento. [Autor]

4. Seleccionar el 20% de las imágenes del dataset para la validación, esto corresponde a 1000 imágenes.

Haciendo uso de la función `tf.keras.utils.image_dataset_from_directory` se crea un conjunto de datos mediante la inclusión de archivos de imagen en un directorio,

segmentándose el conjunto de datos total en 7 agrupaciones de imágenes, de las cuales solo 1000 imágenes se ocupan para la validación.

```
Found 5002 files belonging to 7 classes.  
Using 1000 files for validation.
```

Figura 36. Imágenes seleccionadas para la validación. [Autor]

5. Normalizar y configurar los datos para el entrenamiento.
6. Elaborar el modelo, para la detección de los macroinvertebrados se utilizó un modelo secuencial, conformado por tres bloques de convolución Conv2D, en cada uno de estos bloques está integrada una capa de agrupación máxima denominada MaxPooling2D. Posterior a esto se tiene una capa totalmente conectada conocida como Dense compuesta por 128 unidades y activada mediante la función ReLu.
7. Compilar el modelo, utilizando el optimizador Adam.
8. Entrenar el modelo, ajustando el número de épocas.
9. Visualizar los resultados del entrenamiento de la red, en el gráfico de la izquierda se observa los gráficos de precisión resultantes , tanto de los datos de entrenamiento como de la validación, mientras que en el gráfico de la derecha se observa los gráficos de pérdida, tanto de los datos de entrenamiento como de la validación.

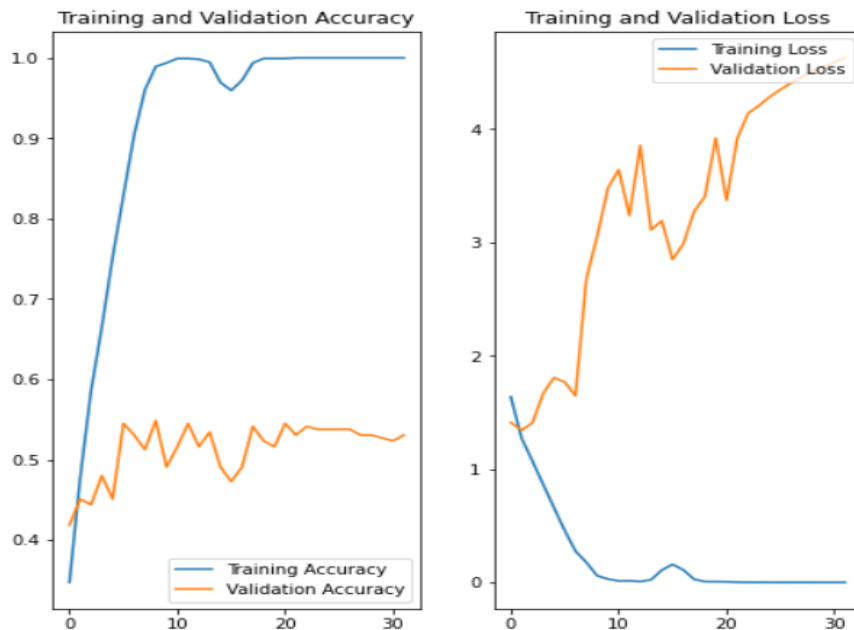


Figura 37. Visualización de los resultados del entrenamiento de la CNN. [Autor]

10. Se procedió a generar métodos de mejoramiento en la CNN, como lo es el aumento de datos logrando instaurar nuevos datos de entrenamiento generados a partir de los datos existentes. Para generar nuevos datos de entrenamiento se utiliza las capas de preprocesamiento denominadas RandomFlip la misma que permite voltear la imagen ya sea vertical o horizontalmente, RandomRotation esta función permite rotar la imagen en un determinado ángulo y RandomZoom que permite acercar o alejar las imágenes.

11. Introducir la regularización de abandonos en la red (dropout), esto ayuda a reducir el sobreajuste en la red, dropout toma el valor de 0.1 lo que significa que se eliminará aleatoriamente el 10% de las unidades de salida.

12. Compilar y entrenar nuevamente el modelo con los ajustes realizados de data augmentation y dropout.

13. Visualizar los nuevos resultados del entrenamiento con los ajustes realizados con lo que se espera que en la CNN ya no exista sobreajuste.

2.4 Implementación del código para la identificación de macroinvertebrados utilizando técnicas de visión por computador, mediante opencv

1. Importación de bibliotecas necesarias:

Se importan las bibliotecas cv2 (OpenCV) para el procesamiento de imágenes, numpy para el manejo de matrices, PIL para trabajar con imágenes, keras para cargar y utilizar un modelo previamente entrenado, os para operaciones relacionadas con el sistema operativo y tensorflow para el cálculo de predicciones.

2. Carga del modelo:

Se carga un modelo previamente entrenado mediante tensorflow, el cual está ubicado en la carpeta "./models". El modelo se guarda en la variable model.

3. Se crea un objeto de captura de video utilizando la cámara predeterminada del dispositivo.

4. El bucle principal se ejecuta de forma constante, para finalizar se debe presionar la tecla 'q'.

5. Se captura el fotograma del video en la variable frame.
6. Se realiza una copia del fotograma original para dibujar contornos.
7. Aplicación de desenfoque gaussiano al fotograma, el fotograma se suaviza utilizando un filtro de desenfoque gaussiano con un kernel de tamaño (7, 7) y una desviación estándar de 1.
8. El fotograma suavizado se transforma a escala de grises.
9. Para lograr obtener una imagen binaria a la imagen en tono de grises se realiza un proceso de umbralización.
10. Haciendo uso del algoritmo Canny ,se revelan los bordes en la imagen en tonos de grises.
11. Teniendo la imagen a blanco y negro se ejecuta una operación de dilatación utilizando un kernel de tamaño (5,5).

2.5 Configuración de flutter para el desarrollo de la app móvil

Instalación de SDK en Flutter.

Flutter SDK hace referencia a un conjunto de implementos, y bibliotecas que permiten la creación de aplicaciones móviles, para proceder con el desarrollo de la aplicación se emplea el lenguaje de programación Dart. Una vez instalado, se puede comprobar la correcta instalación y la versión instalada ejecutando en una terminal de comandos el comando flutter doctor.

```
C:\Users\maris>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.3.10, on Microsoft Windows [Version 10.0.22621.1555], locale en-US)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.2)
```

Figura 38. Verificación de la instalación de SDK. [Autor]

Instalación de Visual Studio Code:

Consiste en un entorno de desarrollo integrado que tiene inmersas varias funciones que permiten desarrollar aplicaciones en varios lenguajes de programación, en este caso se instaló la versión: 1.78.2.

Desarrollo de la aplicación:

Para iniciar con el desarrollo de un nuevo proyecto de Flutter usando el IDE de VS Code, ingrese al menú “command Palette” seguido del comando “flutter: new Project” y seleccione la opción “application”.

Considerando que existen varios directorios con funciones nativas de flutter y que cumplen funciones específicas que permiten el óptimo desarrollo de la aplicación, se tiene un directorio denominado “lib” en donde se encuentra el código principal de la aplicación. Aquí es donde se escriben los archivos Dart que definen la lógica a usar en el desarrollo de la aplicación y la interfaz gráfica.

Directorio:

Dentro del directorio “lib” se pueden encontrar diferentes archivos y carpetas, dependiendo de cómo se organice el proyecto. En el desarrollo de esta aplicación se tiene:

Archivo:

Archivo main.dart: Contiene la función “main()” que se activa al iniciar la aplicación. Aquí se crea la clase “MyApp” y se establece la estructura esencial de la interfaz de usuario de la aplicación.

```
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  cameras = await availableCameras();
  runApp(new MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Análisis de Calidad del Agua',
      theme: ThemeData(
        primarySwatch: Colors.orange,
      ), // ThemeData
      debugShowCheckedModeBanner: false,
      home: const Home(),
    ); // MaterialApp
  }
}
```

Figura 39. Ejemplificación del archivo main.dart. [Autor]

Archivo home.dart: Contiene configuraciones para que detecte la cámara y las imágenes sean analizadas mediante tensorflow.

Widget:

Se considera como widget a un componente básico que es empleado en la construcción de las interfaces de usuario en Flutter. Dentro de Flutter existe varios tipos de widgets que cumplen cada uno con una función específica, son aplicables por ejemplo para el texto, imágenes, botones. En la imagen del archivo main.dart se puede evidenciar la existencia de:

StatelessWidget:

StatelessWidget que se caracteriza por ser inmutable, lo que significa que una vez que se crea y se muestra en la interfaz de usuario, no puede cambiar su apariencia ni su contenido. Dado que un StatelessWidget no tiene estado interno convertible, no puede almacenar ni modificar datos durante su ciclo de vida, por lo que son eficientes en términos de rendimiento.

```
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Análisis de Calidad del Agua',
      theme: ThemeData(
        primarySwatch: Colors.orange,
      ), // ThemeData
      debugShowCheckedModeBanner: false,
      home: const Home(),
    ); // MaterialApp
  }
}
```

Figura 40. Ejemplificación de StatelessWidget. [Autor]

StatefulWidget:

Los StatefulWidget pueden modificar y mantener un estado cambiante, lo que les permite actualizar su apariencia o contenido en respuesta a eventos o cambios en la aplicación.


```
class Home extends StatefulWidget {
  const Home({Key? key}) : super(key: key);

  @override
  State<Home> createState() => _HomeState();
}
```

Figura 41. Ejemplificación de StatefulWidget. [Autor]

Método:

El método build() es un método clave que se encuentra en las clases de widgets, ya sean StatelessWidget o StatefulWidget. El método build() es responsable de construir y retornar la representación visual del widget en la interfaz de usuario.

```
Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Análisis de Calidad del Agua',
    theme: ThemeData(
      primarySwatch: Colors.orange,
    ), // ThemeData
    debugShowCheckedModeBanner: false,
    home: const Home(),
  ); // MaterialApp
}
```

Figura 42. Ejemplificación del método build. [Autor]

Creación de la aplicación:

1. Se firma la apk, en este caso como tenemos el sistema operativo “windows”, se hace uso del siguiente comando en la terminal:

```
“keytool -genkey -v -keystore %userprofile%\upload-keystore.jks -storetype JKS -keyalg RSA -keysize 2048 -validity 10000 -alias upload”.
```

2. Ingresar al directorio del proyecto mediante la terminal de comandos.

3. Ejecutar “flutter build apk –release”

2.6 Configuración de django para el desarrollo de la aplicación web

Archivos:

Para crear una aplicación web utilizando el framework Django, se debe instalar las librerías necesarias, además se debe crear archivos como por ejemplo “views.py” que es un módulo que contiene las funciones o clases de vista que definen la lógica de manejo de las solicitudes entrantes y la generación de respuestas para una aplicación web.

Clase:

CustomFileSystemStorage se refiere a la creación de una clase personalizada que extiende la funcionalidad del sistema de almacenamiento de archivos predeterminado de Django, llamado FileSystemStorage, es de mucha utilidad ya que se puede personalizar el comportamiento de los archivos según los requerimientos específicos para los que se va a desarrollar la aplicación.

```
class CustomFileSystemStorage(FileSystemStorage):
    def get_available_name(self, name, max_length=None):
        self.delete(name)
        return name
```

Figura 43. Ejemplificación de la clase CustomFileSystemStorage [Autor]

También se cuenta con la siguiente función:

Función index perteneciente al backend:

La función index es la encargada de obtener los datos desde la interfaz gráfica, en este caso una imagen que posterior será procesada mediante la aplicación de técnicas de visión por computador juntamente con la red neuronal analizan y en base a ese análisis devuelven un resultado que ayude a la toma de decisiones al usuario.

```
def index(request):
    mensaje = ''
    prediccion = ''
    fss = CustomFileSystemStorage()
    lista = []
    obj = dict()
```

Figura 44. Función index. [Autor]

Index.html perteneciente al frontEnd:

“index.html” funciona juntamente con el archivo “views.py”, está compuesto por una estructura HTML básica. La etiqueta “title” define el título de la página que se mostrará en la pestaña del navegador. Dentro del “body”, se encuentra un encabezado “h1” que muestra un título denominado "Identificación de macroinvertebrados como bioindicadores de la calidad del agua".

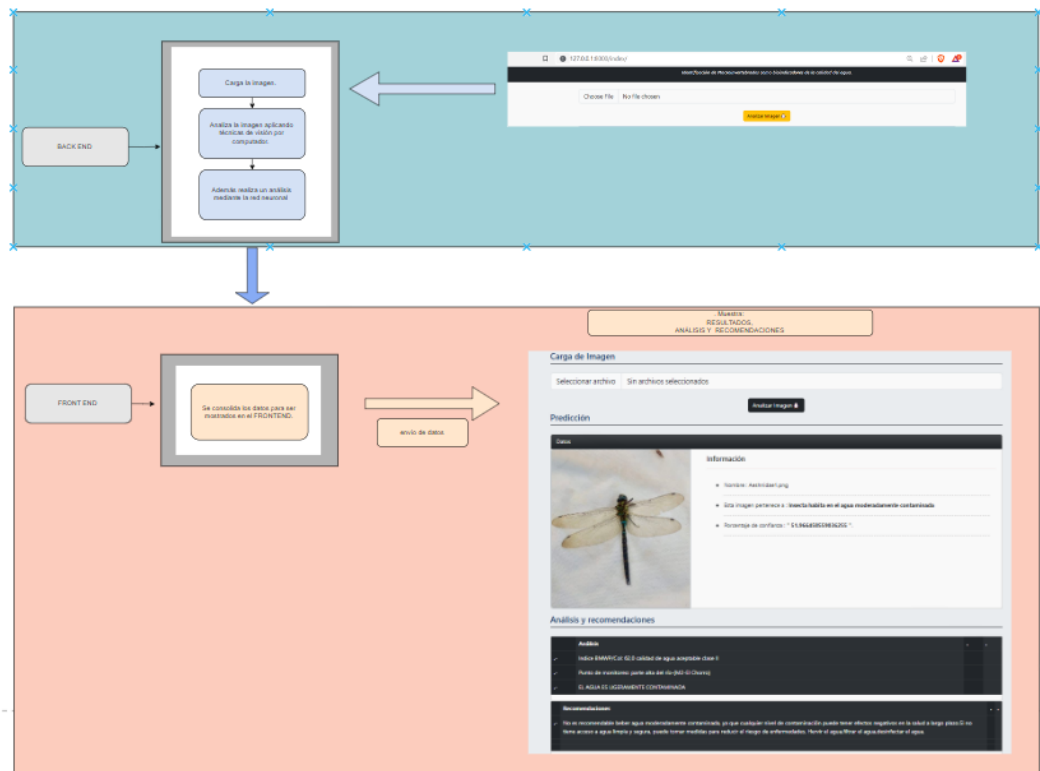


Figura 45. Ejemplificación del backend y frontend que conforman la aplicación web. [Autor]

CAPÍTULO 3: IMPLEMENTACIÓN Y ANÁLISIS DE RESULTADOS

3.1 Visualización del desarrollo del código para la identificación de macroinvertebrados implementando técnicas de visión por computador, mediante opencv

IMAGEN ALMACENADA EN LA BASE DE DATOS:



Figura 46. Imagen almacenada en el dataset. [Autor]

IMAGEN DETECTADA APLICANDO OPENCV:



3.2 Creación y configuración de VPC en la nube

Para el despliegue de la aplicación web en DigitalOcean se hizo uso de una VPC con Ubuntu Server 20.04, 8 GB de memoria RAM, donde se llevó a cabo lo siguiente:

1. En el panel de control, es necesario la creación de una máquina virtual, indispensable para la ejecución de la aplicación.
2. Creación de una VPC, donde se elige el sistema operativo que vamos a usar.
3. En la sección "Choose an image", seleccionar "Ubuntu" y elegir la versión "20.04".
4. En la sección "Select additional options", marcar la casilla "Private networking" para habilitar la comunicación entre las máquinas virtuales dentro de la misma VPC.
5. Una vez que el Droplet esté creado, se obtendrá una dirección IP pública para acceder a él.
6. Abrir una terminal en la máquina local y utilizar el comando SSH para conectarse a tu Droplet utilizando la dirección IP y la clave SSH privada correspondiente.

3.3 Despliegue de la aplicación en la VPC

1. Se instala Python y luego los paquetes tales como tensorflow, keras, opencv, django para ejecutar la aplicación en un entorno Linux
2. Configuración del servidor web, es posible emplear un servidor web como Apache o Nginx para ejecutar la aplicación de django.
3. En el archivo settings.py en django se debe agregar la ip de la vpc para poder ejecutar con el siguiente comando la aplicación:

- *python manage.py runserver 192.168.1.104:8000*

```
root@ubuntuServerTesis:~/TesisDjango/TESIS# python3 manage.py runserver 159.223.191.157:8000
--STATIC_ROOT--> /root/TesisDjango/TESIS/static
--STATICFILES_DIRS--> [PosixPath('/root/TesisDjango/TESIS/assets')]
--STATIC_ROOT--> /root/TesisDjango/TESIS/static
--STATICFILES_DIRS--> [PosixPath('/root/TesisDjango/TESIS/assets')]
Watching for file changes with StatReloader
Performing system checks...

2023-07-18 00:48:22.410696: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2023-07-18 00:48:22.465590: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will not be used.
2023-07-18 00:48:22.466174: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU
al operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-07-18 00:48:23.427803: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
System check identified no issues (0 silenced).
July 18, 2023 - 00:48:24
Django version 2.2.12, using settings 'TESIS.settings'
Starting development server at http://159.223.191.157:8000/
Quit the server with CONTROL-C.
[18/Jul/2023 00:48:39] "GET /index/ HTTP/1.1" 200 7929
[18/Jul/2023 00:48:39] "GET /static/css/bootstrap.min.css HTTP/1.1" 304 0
[18/Jul/2023 00:48:39] "GET /static/images/logo.png HTTP/1.1" 304 0
[18/Jul/2023 00:48:40] "GET /static/images/1im.png HTTP/1.1" 304 0
[18/Jul/2023 00:48:40] "GET /static/js/index.js HTTP/1.1" 304 0
[18/Jul/2023 00:48:40] "GET /static/images/3im.png HTTP/1.1" 304 0
[18/Jul/2023 00:48:40] "GET /static/images/load.png HTTP/1.1" 304 0
[18/Jul/2023 00:48:40] "GET /static/images/2im.png HTTP/1.1" 304 0
Not Found: /favicon.ico
[18/Jul/2023 00:48:40] "GET /favicon.ico HTTP/1.1" 404 2336
```

Figura 48. Despliegue de la aplicación web en el servidor. [Autor]

3.4 Visualización de la interfaz de la plataforma en línea.

El FrontEnd comprende la interfaz del servidor web, que puede ser visualizada mediante la dirección (<http://127.0.0.1:8000/index>), localmente. En el caso de que se ejecute en un servidor en producción se debería acceder a la ip pública asignada desde el navegador.

Para la creación de la interfaz de usuario de la aplicación web se utilizó el framework Django, el cual está basado en Python. Posteriormente para ingresar a la interfaz en la terminal de comandos se coloca:

“python manage.py runserver”

```
Django version 3.2.18, using settings 'TESIS.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Figura 49. Verificación de ejecución del servidor. [Autor]

Al desplegarse el servidor se observa la siguiente interfaz, en donde se elige la imagen del macroinvertebrado al dar click en el recuadro de choose file, carga la imagen y realiza un análisis indicando a qué clase de macroinvertebrado pertenece, el nivel de contaminación del agua, los sectores donde habitan, además se indica algunas recomendaciones que se debe tener en cuenta al momento de consumir el agua.



Figura 50. Interfaz de la aplicación web. [Autor]



Figura 51. Interfaz de la aplicación web. [Autor]

Identificación de macroinvertebrados acuáticos.


Los macroinvertebrados acuáticos son organismos pequeños sin columna vertebral que viven en ambientes acuáticos, como ríos, lagos y arroyos. Estos organismos incluyen insectos acuáticos, crustáceos, gusanos y moluscos.



- clase Crustácea
- clase Gastrópoda
- clase Insecta

Figura 52. Interfaz de la aplicación web. [Autor]

Datos



Información

- Nombre : CrustaceaHyalellidae8.png
- Esta imagen pertenece a : **Crustacea**
- Porcentaje de confianza : " 98.82423281669617 ".

Análisis y recomendaciones

Análisis


- Índice BMWP/Cob(42,3)-(48,7)-(37,7)-calidad de agua dudosa clase III
- Punto de monitoreo: parte media y baja del río-(M4-Daniel Durán, M5-Chamana, M6-Las Orquídeas)
- EL AGUA ES MODERADAMENTE CONTAMINADA

Recomendaciones

- No es recomendable beber agua moderadamente contaminada, ya que cualquier nivel de contaminación puede tener efectos negativos en la salud a largo plazo. Si no tiene acceso a agua limpia y segura, puede tomar medidas para reducir el riesgo de enfermedades. Hervir el agua, filtrar el agua, desinfectar el agua.

Figura 53. Interfaz de la aplicación web para el análisis de macroinvertebrados, con énfasis en aquellos correspondientes a la clase Crustácea. [Autor]

Datos



Información

- Nombre : GastropodaPhysidae15.png
- Esta imagen pertenece a : **Gastropoda**
- Porcentaje de confianza : " 44.6880578994751 "

Análisis y recomendaciones

Análisis


- Índice BMWP/Col: 31,0 calidad de agua crítica clase IV
- Punto de monitoreo: parte baja del río- (M7-Camino a Patamarca, M8-Parque Industrial)
- EL AGUA ES MODERADAMENTE CONTAMINADA

Recomendaciones

- No es recomendable beber agua moderadamente contaminada, ya que cualquier nivel de contaminación puede tener efectos negativos en la salud a largo plazo. Si no tiene acceso a agua limpia y segura, puede tomar medidas para reducir el riesgo de enfermedades. Hervir el agua, filtrar el agua, desinfectar el agua.

Figura 54. Interfaz de la aplicación web para el análisis de macroinvertebrados, con énfasis en aquellos correspondientes a la clase Gastropoda. [Autor]

Datos



Información

- Nombre : Dytiscidae14.png
- Esta imagen pertenece a : **Insecta habita en el agua pura**
- Porcentaje de confianza : " 85.68809032440186 "

Análisis y recomendaciones

Análisis


- Índice BMWP/Col: 72,3 calidad de agua aceptable clase II
- Punto de monitoreo: parte alta del río- (M1-Cochas)
- EL AGUA ES MODERADAMENTE CONTAMINADA

Recomendaciones

- Beber agua de vertientes naturales no contaminadas puede ser una opción saludable para hidratarse, pero aún así existen algunos riesgos que es importante tener en cuenta: Contaminación bacteriana: Aunque el agua de la vertiente puede parecer limpia, todavía existe la posibilidad de que esté contaminada con bacterias y otros microorganismos patógenos, especialmente si hay animales cerca que puedan contaminar el agua. La exposición a estas bacterias puede causar enfermedades gastrointestinales como diarrea, vómitos y fiebre.

Figura 55. Interfaz de la aplicación web para el análisis de macroinvertebrados, con énfasis en aquellos correspondientes a la clase Insecta de agua limpia. [Autor]

Datos



Información

- Nombre : chironomidae7.png
- Esta imagen pertenece a : **Insecta habita en el agua muy contaminada**
- Porcentaje de confianza : " 79.32060956954956 "

Análisis y recomendaciones

Análisis


- Índice BMWP/Col: 31,0 calidad de agua crítica clase IV
- Punto de monitoreo: parte media y baja del río -(M3-Centro Parroquial, M7-Camino a Patamarca, M8-Parque Industrial)
- EL AGUA ES MUY CONTAMINADA

Recomendaciones

- Beber agua muy contaminada puede tener graves riesgos para la salud, que pueden variar dependiendo de los tipos de contaminantes presentes en el agua. Algunos de los riesgos más comunes incluyen: enfermedades gastrointestinales, problemas de salud crónicos, daño al sistema inmunológico, problemas en el desarrollo infantil

Figura 56. Interfaz de la aplicación web para el análisis de macroinvertebrados, con énfasis en aquellos correspondientes a la clase Insecta de agua muy contaminada. [Autor]

Datos



Información

- Nombre : Aeshnidae43.png
- Esta imagen pertenece a : **Insecta habita en el agua moderadamente contaminada**
- Porcentaje de confianza : " 81.87962770462036 "

Análisis y recomendaciones

Análisis

- Índice BMWP/Col: 62,0 calidad de agua aceptable clase II
- Punto de monitoreo: parte alta del río -(M2-El Chorro)
- EL AGUA ES LIGERAMENTE CONTAMINADA


Recomendaciones

- No es recomendable beber agua moderadamente contaminada, ya que cualquier nivel de contaminación puede tener efectos negativos en la salud a largo plazo. Si no tiene acceso a agua limpia y segura, puede tomar medidas para reducir el riesgo de enfermedades. Hervir el agua, filtrar el agua, desinfectar el agua.

Figura 57. Interfaz de la aplicación web para el análisis de macroinvertebrados, con énfasis en aquellos correspondientes a la clase Insecta de agua moderadamente contaminada. [Autor]

Predicción

Datos



Información

- Nombre : oligochaeta14.png
- Esta imagen pertenece a : **Oligochaeta**
- Porcentaje de confianza : " 56.42474889755249 ".

Análisis y recomendaciones

Análisis


- Índice BMWP/Col: 31,0 calidad de agua crítica clase VI
- Punto de monitoreo: parte baja del río-(M7-Camino a Patamarca, M8-Parque Industrial)
- EL AGUA ES MUY CONTAMINADA

Recomendaciones

- Beber agua muy contaminada puede tener graves riesgos para la salud, que pueden variar dependiendo de los tipos de contaminantes presentes en el agua. Algunos de los riesgos más comunes incluyen: enfermedades gastrointestinales, problemas de salud crónicos, daño al sistema inmunológico, problemas en el desarrollo infantil

Figura 58. Interfaz de la aplicación web para el análisis de macroinvertebrados, con énfasis en aquellos correspondientes a la clase Oligochaeta. [Autor]

Datos



Información

- Nombre : planariidae32.png
- Esta imagen pertenece a : **Turbellaria**
- Porcentaje de confianza : " 81.6626250743866 ".

Análisis y recomendaciones

Análisis

- Índice BMWP/Col: 42,3-48,7-37,7- calidad de agua dudosa clase III
- Punto de monitoreo: parte media y baja del río-(M4-Daniel Durán, M5-Chamana, M6-Las Orquídeas)
- EL AGUA ES MODERADAMENTE CONTAMINADA

Recomendaciones

- No es recomendable beber agua moderadamente contaminada, ya que cualquier nivel de contaminación puede tener efectos negativos en la salud a largo plazo. Si no tiene acceso a agua limpia y segura, puede tomar medidas para reducir el riesgo de enfermedades. Hervir el agua, filtrar el agua, desinfectar el agua.

Figura 59. Interfaz de la aplicación web para el análisis de macroinvertebrados, con énfasis en aquellos correspondientes a la clase Turbellaria. [Autor]

3.5 Visualización de la interfaz de la aplicación móvil.

IMAGEN ALMACENADA EN EL DATASET:



Figura 60. Imagen perteneciente al dataset. [Autor]

IMAGEN DETECTADA EN LA APLICACIÓN MÓVIL:



Figura 61. Interfaz de la aplicación móvil. [Autor]

CAPÍTULO 4:

CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS.

4.1 Conclusiones

Para la ejecución de esta tesis, se realizó un análisis detallado del estado del arte en la identificación de macroinvertebrados acuáticos utilizando técnicas de visión por computador, mediante esta investigación previa se determinó que la técnica más apropiada para el desarrollo del sistema es la creación de una red neuronal secuencial, en la cual se empleó técnicas de mejoramiento de la red como lo es data Augmentation y Dropout, lo cual ayuda en gran medida a disminuir el sobreajuste de la red y a su vez mejora el rendimiento. Este tipo de red neuronal es muy eficaz en el análisis de características en una imagen, debido a su excelente capacidad de detección de las características más relevantes, logrando obtener una respuesta muy precisa en casos de estudio como lo es en la identificación, clasificación, fragmentación de objetos.

Los organismos analizados, son excelentes indicadores de la calidad del agua, mediante el análisis automatizado empleando técnicas de visión por computador, como OpenCV, permite agilizar el proceso de evaluación de las condiciones actuales del agua en el río examinado, logrando obtener información instantánea, eficiente y a su vez empleando métodos que no son nocivos para el ecosistema.

Además, la utilización de técnicas de visión por computador abre la puerta a la integración de sistemas de monitoreo continuo, lo que proporcionaría una supervisión en tiempo real y una detección rápida de la existencia de contaminantes deteriorando la calidad del agua en la cuenca del río Paute.

4.2 Análisis realizado en la CNN

Configuración de CNN propuesta:

Tabla 3.1 Parámetros ingresados para el análisis de la CNN.

Parámetros ingresados	
Parámetros	Valor
batch_size	50
Épocas	20

Tabla 3.2 Implementación de parámetros en el modelo secuencial para el análisis de la CNN.

Símbolo	Descripción	Configuraciones
layers.Rescaling	Re-escalamiento de la imagen de ingreso	220x220
layers.Conv2D	Convolución 2D	16,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Conv2D	Convolución 2D	32,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Conv2D	Convolución 2D	64,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Flatten	Capa de aplanamiento	
layers.Dense	Capa completamente conectada	128, activation='relu'
layers.Dense	Capa completamente conectada	num_classes

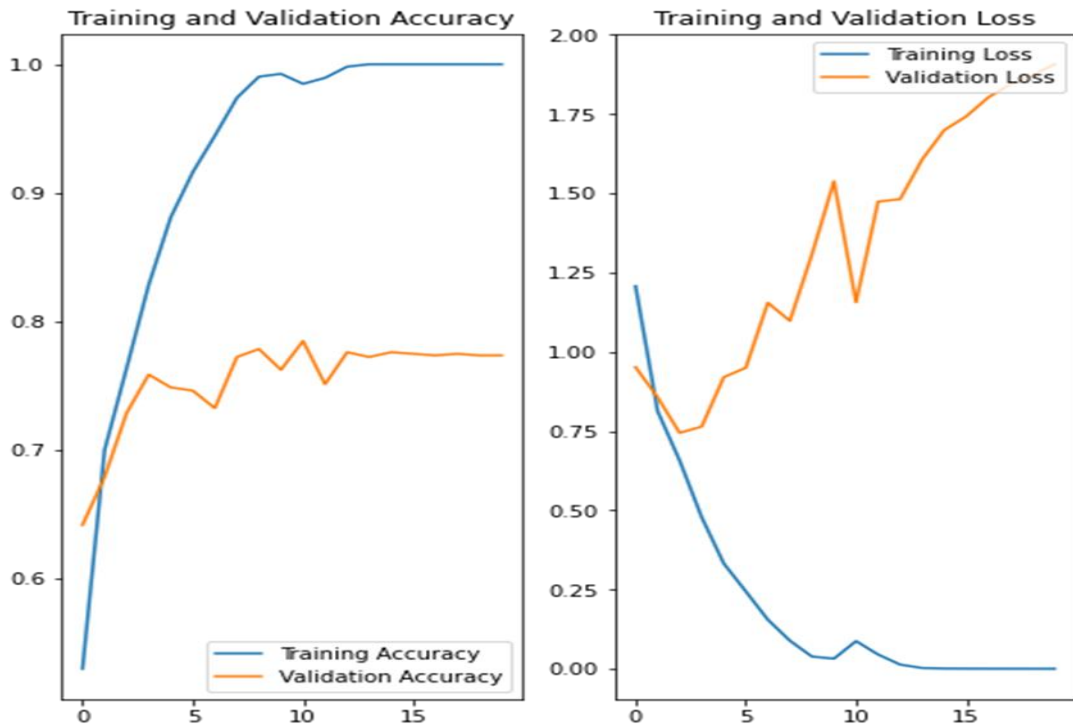


Figura 62. Gráfico correspondiente a la precisión en el entrenamiento y la validación (izquierda).

Gráfico correspondiente a la pérdida en el entrenamiento y la validación (derecha). [Autor]

El gráfico correspondiente a la precisión en el conjunto de entrenamiento y la validación indican que existe sobreajuste o lo que se conoce como “overfitting” ya que están desviadas por amplios márgenes, alcanzando aproximadamente el 77% de precisión en el conjunto de validación, mientras que la precisión del conjunto de entrenamiento aumenta linealmente con el tiempo. En el gráfico de pérdida (derecha), correspondiente a los conjuntos de datos de entrenamiento y validación se puede evidenciar que existe una diferencia notable ya que en los datos de validation loss a medida que se incrementa las épocas existe mayor pérdida de información.

Configuración de parámetros en la CNN empleando métodos de mejoramiento de la red (data Augmentation y Dropout.)

Tabla 3.3 Parámetros ingresados implementando técnicas de mejoramiento en la red como es Data Augmentation y Dropout.

Parámetros ingresados	
Parámetros	Valor
batch size	50
epocas	20

Tabla 3.4 Configuración-Data Augmentation

Parámetros ingresados	
Parámetros	Valor
layers.RandomFlip	“horizontal and vertical”
layers.RandomRotation	0.1
Layers.RandomZoom	0.1

Tabla 3.5 Implementación de parámetros adicionando (dropout), en el modelo secuencial para el mejoramiento de la CNN.

Simbolo	Descripción	Configuraciones
layers.Rescaling	Re-escalamiento de la imagen de ingreso	220x220
layers.Conv2D	Convolución 2D	16,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Conv2D	Convolución 2D	32,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Conv2D	Convolución 2D	64,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Dropout	Capa de abandono	0.2
layers.Flatten	Capa de aplanamiento	
layers.Dense	Capa completamente conectada	128, activation='relu'
layers.Dense	Capa completamente conectada	num_classes

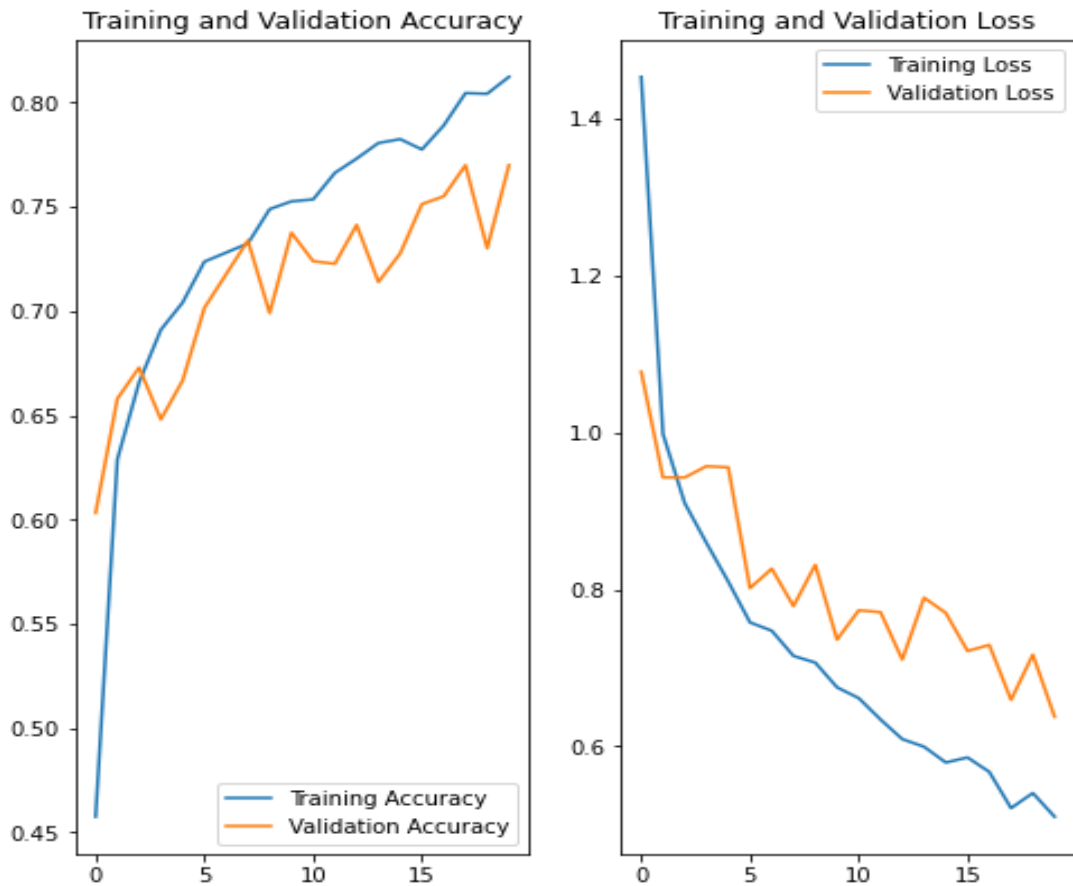


Figura 63. Gráfico correspondiente a la precisión en el entrenamiento y la validación (izquierda). Gráfico correspondiente a la pérdida en el entrenamiento y la validación (derecha). Implementando técnicas de mejoramiento en la red como es data Augmentation y Dropout. [Autor]

Se implementó técnicas de mejoramiento en la red como es data Augmentation y Dropout.

La técnica de mejoramiento denominada data Augmentation crea nuevos datos a partir de los datos ya existentes, esto ayuda al modelo ya que es expuesto a más aspectos y su capacidad de aprendizaje mejora notablemente. Otra técnica para reducir el sobreajuste es la implementación de dropout esto consiste en la desactivación de una capa, colocándolo en cero, una cantidad de unidades de salida en el transcurso de la etapa de entrenamiento, con estas técnicas implementadas se observa que los gráficos correspondientes a la precisión en el entrenamiento y la validación, indican que existe el 76% de precisión en los datos pertenecientes al conjunto de validación, a la vez que la precisión en los datos pertenecientes al conjunto de entrenamiento aumenta linealmente con el tiempo. En el gráfico de pérdida tanto para los datos de entrenamiento como de validación se puede evidenciar que existe una mejora notable ya que la pérdida de datos disminuye notablemente alcanzando el 0.6377.

Configuración de CNN propuesta:

Tabla 3.6 Parámetros ingresados para el análisis de la CNN.

Parámetros ingresados	
Parámetros	Valor
batch size	32
epocas	20

Tabla 3.7 Implementación de parámetros en el modelo secuencial para el análisis de la CNN.

Simbolo	Descripción	Configuraciones
layers.Rescaling	Re-escalamiento de la imagen de ingreso	220x220
layers.Conv2D	Convolución 2D	16,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Conv2D	Convolución 2D	32,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Conv2D	Convolución 2D	64,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Flatten	Capa de aplanamiento	
layers.Dense	Capa completamente conectada	128, activation='relu'
layers.Dense	Capa completamente conectada	num_classes

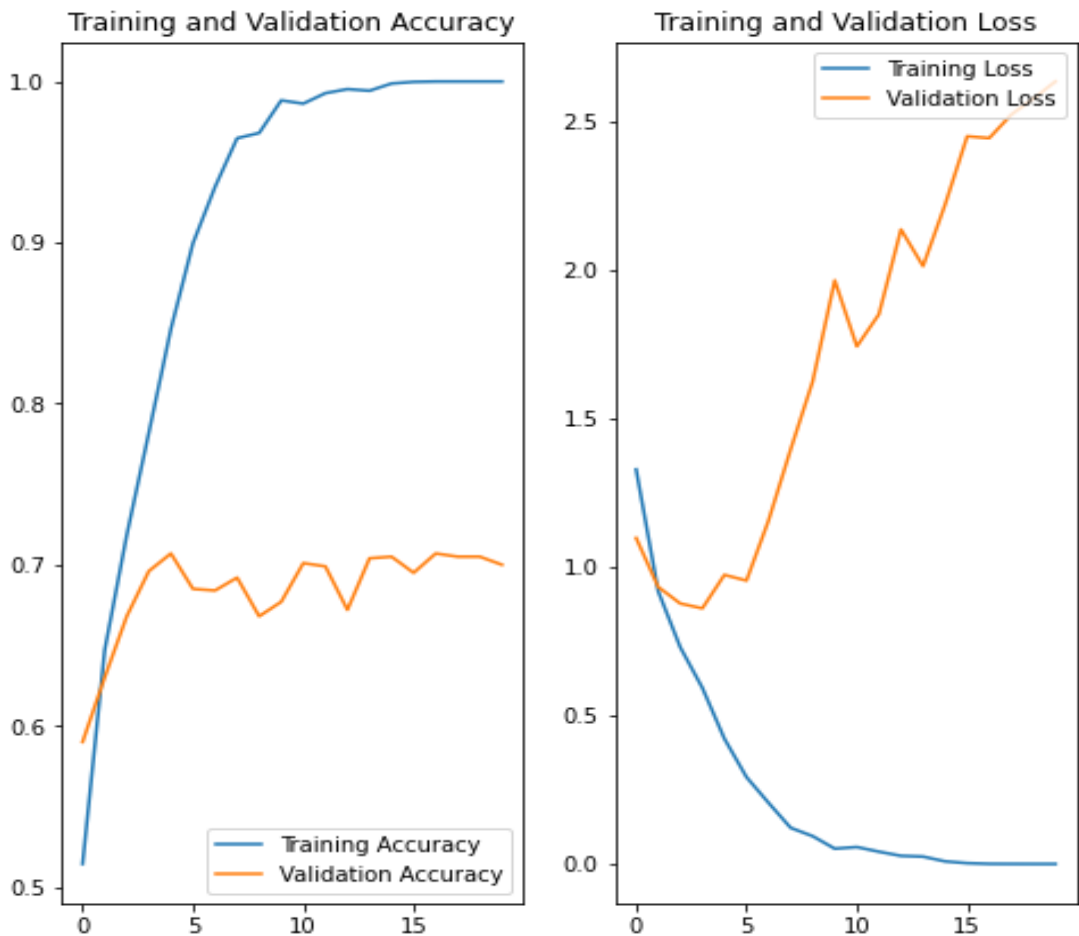


Figura 64. Gráfico correspondiente a la precisión en el entrenamiento y la validación (izquierda). Gráfico correspondiente a la pérdida en el entrenamiento y la validación (derecha). [Autor]

El gráfico correspondiente a la precisión en el conjunto de entrenamiento y la validación indican que existe sobreajuste o lo que se conoce como “overfitting” ya que están desviadas por amplios márgenes, alcanzando aproximadamente el 70% de precisión en el conjunto de datos pertenecientes a la validación, mientras que la precisión del conjunto de entrenamiento se incrementa proporcionalmente con el tiempo. En el gráfico de pérdida (derecha), correspondiente a los conjuntos de datos de entrenamiento y validación se puede evidenciar que existe una diferencia notable ya que en los datos de validation loss a medida que se incrementa las épocas existe mayor pérdida de información.

Configuración de parámetros en la CNN empleando métodos de mejoramiento de la red (data Augmentation y Dropout.)

Tabla 3.8 Parámetros ingresados implementando técnicas de mejoramiento en la red como es Data Augmentation y Dropout.

Parámetros ingresados	
Parámetros	Valor
batch size	50
Épocas	20

Tabla 3.9 Configuración-Data Augmentation

Parámetros ingresados	
Parámetros	Valor
layers.RandomFlip	“horizontal and vertical”
layers.RandomRotation	0.1
Layers.RandomZoom	0.1

Tabla 3.10 Implementación de parámetros adicionando (dropout), en el modelo secuencial para el mejoramiento de la CNN.

Símbolo	Descripción	Configuraciones
layers.Rescaling	Re-escalamiento de la imagen de ingreso	220x220
layers.Conv2D	Convolución 2D	16,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Conv2D	Convolución 2D	32,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Conv2D	Convolución 2D	64,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Dropout	Capa de abandono	0.2
layers.Flatten	Capa de aplanamiento	
layers.Dense	Capa completamente conectada	128, activation='relu'
layers.Dense	Capa completamente conectada	num_classes

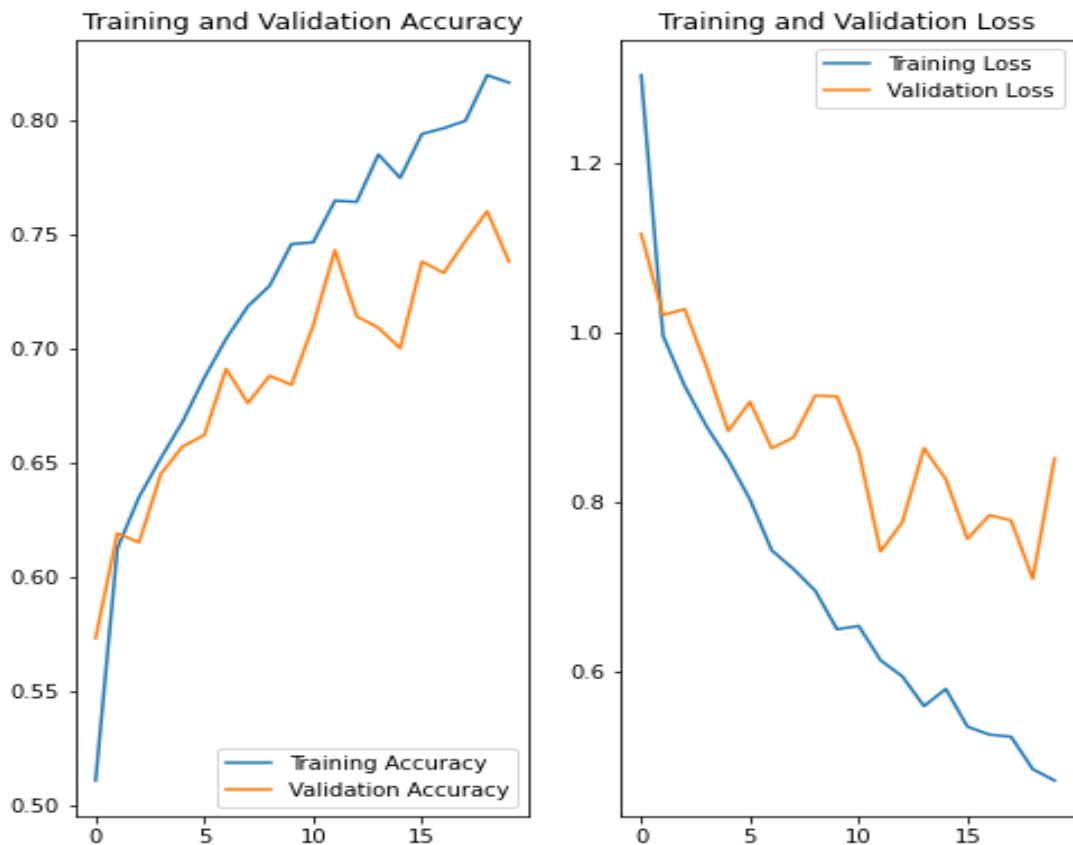


Figura 65. Gráfico correspondiente a la precisión en el entrenamiento y la validación (izquierda). Gráfico correspondiente a la pérdida en el entrenamiento y la validación (derecha). Implementando técnicas de mejoramiento en la red como es data Augmentation y Dropout. [Autor]

Se implementó técnicas de mejoramiento en la red como es data Augmentation y Dropout.

La técnica de mejoramiento denominada data Augmentation crea nuevos datos a partir de los datos ya existentes, esto ayuda al modelo ya que es expuesto a más aspectos y su capacidad de aprendizaje mejora notablemente. Otra técnica para reducir el sobreajuste es la implementación de dropout esto consiste en la desactivación de una capa, colocándolo en cero, una cantidad de unidades de salida en el transcurso de la etapa de entrenamiento, con estas técnicas implementadas se observa que los gráficos correspondientes a la precisión en el entrenamiento y la validación, indican que existe el 73% de precisión en los datos pertenecientes al conjunto de validación, a la vez que la precisión en los datos pertenecientes al conjunto de entrenamiento aumenta linealmente con el tiempo. En el gráfico de pérdida tanto para los datos de entrenamiento como de validación se puede evidenciar que existe una mejora notable ya que la pérdida de datos disminuye notablemente alcanzando el 0.8515.

Configuración de CNN propuesta:

Tabla 3.11 Parámetros ingresados para el análisis de la CNN.

Parámetros ingresados	
Parámetros	Valor
Batch_size	72
Épocas	30

Tabla 3.12 Implementación de parámetros en el modelo secuencial para el análisis de la CNN.

Simbolo	Descripción	Configuraciones
layers.Rescaling	Re-escalamiento de la imagen de ingreso	220x220
layers.Conv2D	Convolución 2D	16,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Conv2D	Convolución 2D	32,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Conv2D	Convolución 2D	64,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Flatten	Capa de aplanamiento	
layers.Dense	Capa completamente conectada	128, activation='relu'
layers.Dense	Capa completamente conectada	num_classes

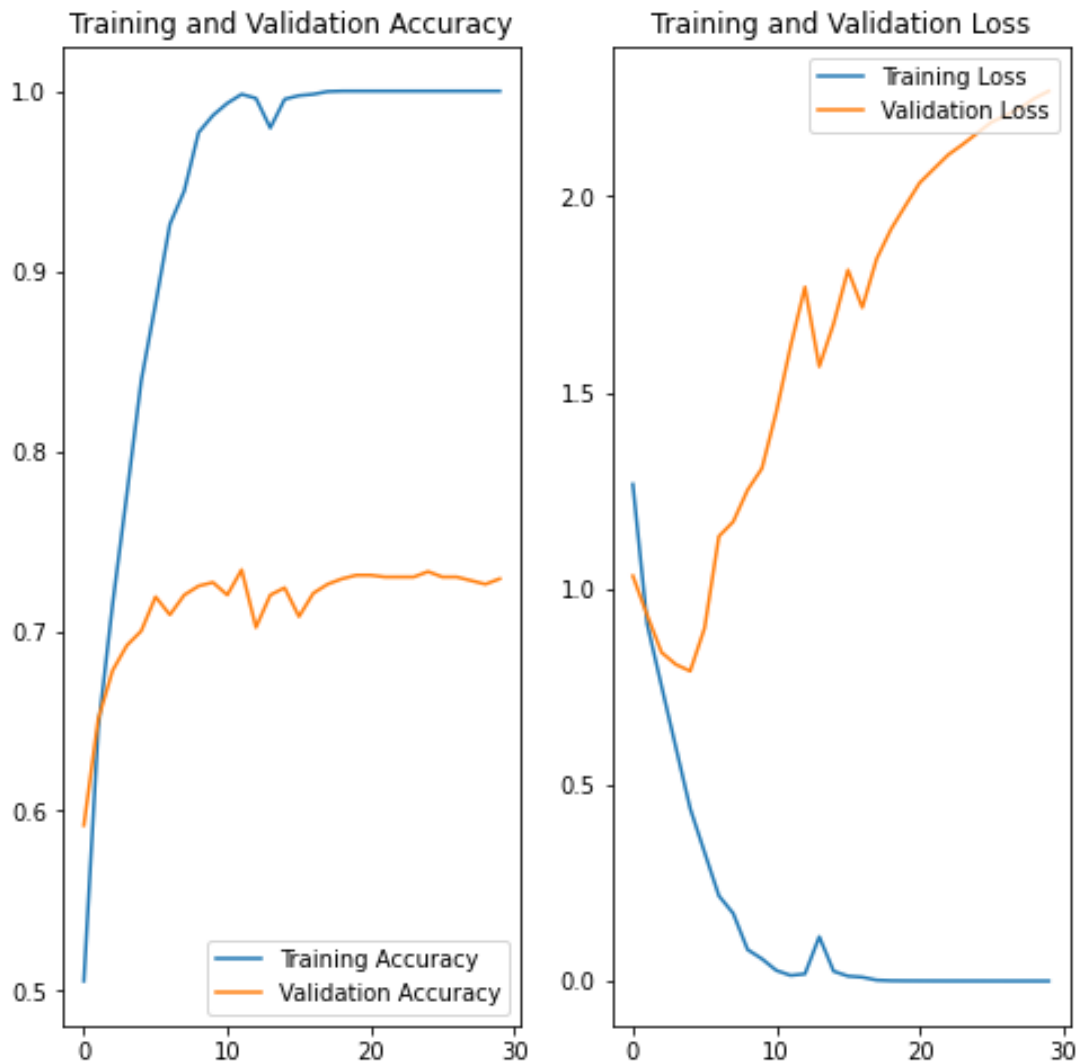


Figura 66. Gráfico correspondiente a la precisión en el entrenamiento y la validación (izquierda).

Gráfico correspondiente a la pérdida en el entrenamiento y la validación (derecha). [Autor]

El gráfico correspondiente a la precisión en el conjunto de entrenamiento y la validación indican que existe sobreajuste o lo que se conoce como “overfitting” ya que están desviadas por amplios márgenes, alcanzando aproximadamente el 74% de precisión en el conjunto de datos pertenecientes a la validación, mientras que la precisión del conjunto de entrenamiento se incrementa proporcionalmente con el tiempo. En el gráfico de pérdida (derecha), correspondiente a los conjuntos de datos de entrenamiento y validación se puede evidenciar que existe una diferencia notable ya que en los datos de validation loss a medida que se incrementa las épocas existe mayor pérdida de información.

Configuración de parámetros en la CNN empleando métodos de mejoramiento de la red (data Augmentation y Dropout.)

Tabla 3.13 Parámetros ingresados implementando técnicas de mejoramiento en la red como es Data Augmentation y Dropout.

Parámetros ingresados	
Parámetros	Valor
Batch_size	72
Épocas	8

Tabla 3.14 Configuración-Data Augmentation

Parámetros ingresados	
Parámetros	Valor
layers.RandomFlip	“horizontal and vertical”
layers.RandomRotation	0.1
Layers.RandomZoom	0.1

Tabla 3.15 Implementación de parámetros adicionando (dropout), en el modelo secuencial para el mejoramiento de la CNN.

Símbolo	Descripción	Configuraciones
layers.Rescaling	Re-escalamiento de la imagen de ingreso	220x220
layers.Conv2D	Convolución 2D	16,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Conv2D	Convolución 2D	32,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Conv2D	Convolución 2D	64,3, padding='same', activation='relu'
layers.MaxPooling2D	MaxPooling 2D	
layers.Dropout	Capa de abandono	0.2
layers.Flatten	Capa de aplanamiento	
layers.Dense	Capa completamente conectada	128, activation='relu'
layers.Dense	Capa completamente conectada	num_classes



Figura 67. Gráfico correspondiente a la precisión en el entrenamiento y la validación (izquierda). Gráfico correspondiente a la pérdida en el entrenamiento y la validación (derecha). Implementando técnicas de mejoramiento en la red como es data Augmentation y Dropout. [Autor]

Se implementó técnicas de mejoramiento en la red como es data Augmentation y Dropout.

La técnica de mejoramiento denominada data Augmentation crea nuevos datos a partir de los datos ya existentes, esto ayuda al modelo ya que es expuesto a más aspectos y su capacidad de aprendizaje mejora notablemente. Otra técnica para reducir el sobreajuste es la implementación de dropout esto consiste en la desactivación de una capa, colocándolo en cero, una cantidad de unidades de salida en el transcurso de la etapa de entrenamiento, con estas técnicas implementadas se observa que los gráficos correspondientes a la precisión en el entrenamiento y la validación, indican que existe el 65% de precisión en los datos pertenecientes al conjunto de validación, a la vez que

la precisión en los datos pertenecientes al conjunto de entrenamiento aumenta linealmente con el tiempo. En el gráfico de pérdida tanto para los datos de entrenamiento como de validación se puede evidenciar que existe una mejora notable ya que la pérdida de datos disminuye notablemente alcanzando el 0.72.

4.3 Analisis de la matriz de confusión de la CNN

conjunto de datos utilizado en el entrenamiento:



Figura 68. Gráfico de la matriz de confusión correspondiente al conjunto de datos utilizado en el entrenamiento. [Autor]

	precision	recall	f1-score	support
0	0.9464	0.9552	0.9508	536
1	0.9091	0.7582	0.8268	765
2	0.7043	0.6111	0.6544	756
3	0.7650	0.9220	0.8362	1462
4	0.5758	0.5048	0.5380	414
5	0.7500	0.4000	0.5217	15
6	0.7647	0.4815	0.5909	54
accuracy			0.7854	4002
macro avg	0.7736	0.6618	0.7027	4002
weighted avg	0.7857	0.7854	0.7801	4002

Figura 69. Resumen del texto que incluye la precisión, recuperación y puntuación F1 para cada clase de macroinvertebrados analizados en el conjunto de datos de entrenamiento [Autor]

En la matriz de confusión del conjunto de datos correspondiente al entrenamiento, se llevó a cabo el análisis de la diagonal principal, que representa las evaluaciones correctas. En este contexto, se identificaron los verdaderos positivos, que en este caso son 1348 imágenes pertenecientes a la clase de insectos que habitan en agua muy contaminada. Los verdaderos negativos, por otro lado, corresponden a las demás evaluaciones que no pertenecen a esta clase, pero están en la matriz diagonal.

Además, se identificaron los falsos positivos en el sentido vertical, que son las imágenes clasificadas incorrectamente como pertenecientes a una clase específica cuando no lo son. También se identificaron los falsos negativos en el sentido horizontal, que son las imágenes clasificadas incorrectamente como si no pertenecieran a la clase cuando realmente sí pertenecen.

También, se observan los diferentes porcentajes obtenidos en los parámetros analizados, tales como precisión, recall, F1-score y support. Si se busca un clasificador que minimice los falsos positivos o los mantenga en niveles mínimos, se debería elegir un clasificador que ofrezca una mayor precisión. Por el contrario, si se necesita un sistema que reduzca al mínimo los falsos negativos, asignando la mayoría de las imágenes a la especie correspondiente, se optaría por un clasificador de recall. De manera similar, en esta gráfica se aprecia el porcentaje de precisión global obtenido en todo el sistema.

conjunto de datos utilizado en el testeo:

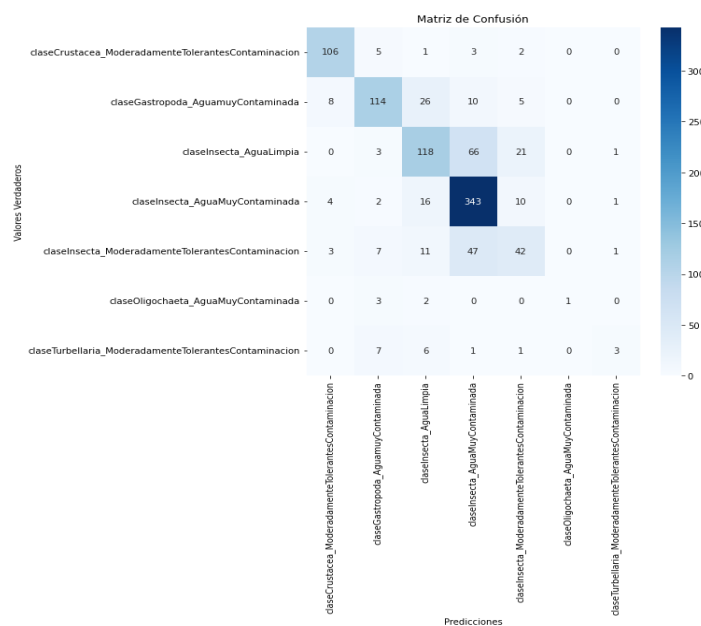


Figura 70. Gráfico de la matriz de confusión correspondiente al conjunto de datos utilizado en el testeo. [Autor]

	precision	recall	f1-score	support
0	0.8760	0.9060	0.8908	117
1	0.8085	0.6994	0.7500	163
2	0.6556	0.5646	0.6067	209
3	0.7298	0.9122	0.8109	376
4	0.5185	0.3784	0.4375	111
5	1.0000	0.1667	0.2857	6
6	0.5000	0.1667	0.2500	18
accuracy			0.7270	1000
macro avg	0.7269	0.5420	0.5759	1000
weighted avg	0.7182	0.7270	0.7129	1000

Figura 71. Resumen del texto que incluye la precisión, recuperación y puntuación F1 para cada clase de macroinvertebrados analizados en el conjunto de datos de testeo. [Autor]

En la matriz de confusión del conjunto de datos correspondiente al testeo, se llevó a cabo el análisis de la diagonal principal, que representa las evaluaciones correctas. En este contexto, se identificaron los verdaderos positivos, que en este caso son 343 imágenes pertenecientes a la clase de insectos que habitan en agua muy contaminada. Los verdaderos negativos, por otro lado, corresponden a las demás evaluaciones que no pertenecen a esta clase, pero están en la matriz diagonal.

Además, se identificaron los falsos positivos en el sentido vertical, que son las imágenes clasificadas incorrectamente como pertenecientes a una clase específica cuando no lo son. También se identificaron los falsos negativos en el sentido horizontal, que son las imágenes clasificadas incorrectamente como si no pertenecieran a la clase cuando realmente sí pertenecen.

También, se observan los diferentes porcentajes obtenidos en los parámetros analizados, tales como precisión, recall, F1-score y support. Si se busca un clasificador que minimice los falsos positivos o los mantenga en niveles mínimos, se debería elegir un clasificador que ofrezca una mayor precisión. Por el contrario, si se necesita un sistema que reduzca al mínimo los falsos negativos, asignando la mayoría de las imágenes a la especie correspondiente, se optaría por un clasificador de recall. De manera similar, en esta gráfica se aprecia el porcentaje de precisión global obtenido en todo el sistema.

4.4 Recomendaciones

Una vez entrenada la CNN se logró determinar que es necesario tener una base de datos muy extensa, además se debe ajustar adecuadamente los hiperparámetros para alcanzar mayor precisión al momento de entrenar los datos pertenecientes al conjunto de entrenamiento y a la validación.

4.5 Investigaciones futuras

Se plantea la fabricación de un robot móvil equipado con sensores y una Raspberry Pi para determinar las condiciones actuales del agua. Este robot se desplazará por diferentes puntos de la cuenca del río analizado, recopilando datos que serán capturados por una cámara. Estos datos serán procesados por una red neuronal utilizando técnicas de visión por computadora mediante la Raspberry Pi.

Además, se adjunta un prototipo del diseño de un robot móvil realizado en Autodesk Inventor, el cual servirá como referencia para la construcción física del robot. El diseño incluirá los componentes necesarios, como los sensores, la cámara y la Raspberry Pi, estará adaptado para funcionar en entornos acuáticos.

Este trabajo permitirá recopilar datos precisos y en tiempo real sobre las condiciones actuales del agua a lo largo del río, brindando información valiosa para la valoración y vigilancia de la salud del ecosistema acuático. Además, la integración de técnicas de visión por computadora y una red neuronal mejorará la capacidad de detección y análisis de indicadores de calidad del agua.

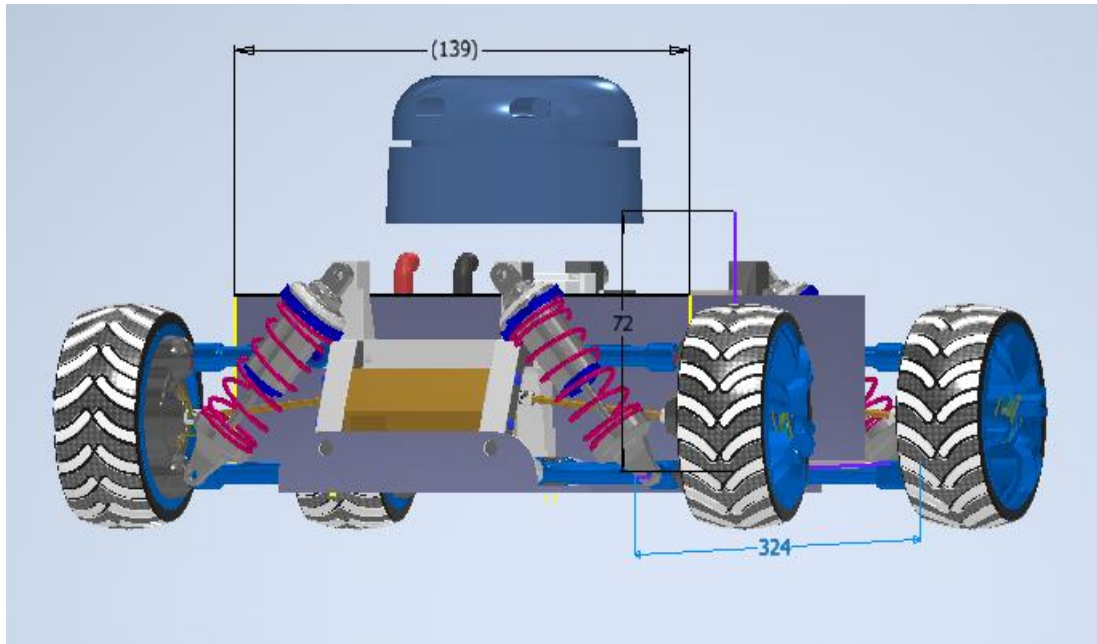


Figura 72. Vista frontal del prototipo del diseño de un robot móvil utilizado para evaluar las condiciones actuales del agua. [Autor]

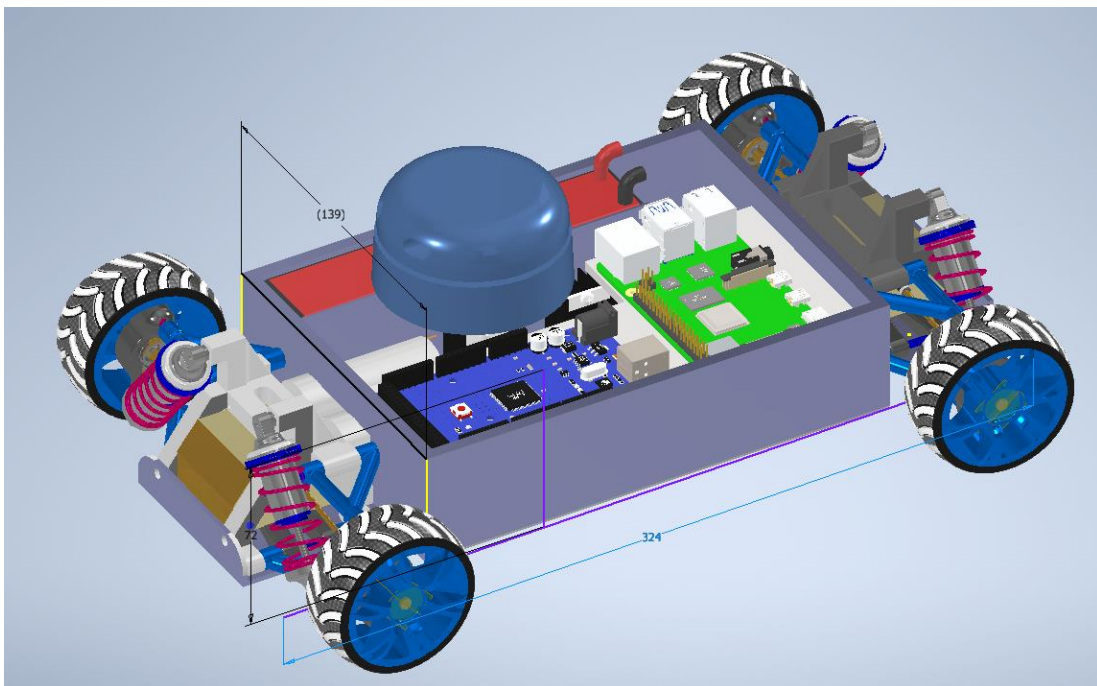


Figura 73. Vista lateral del prototipo del diseño de un robot móvil utilizado para evaluar las condiciones actuales del agua. [Autor]

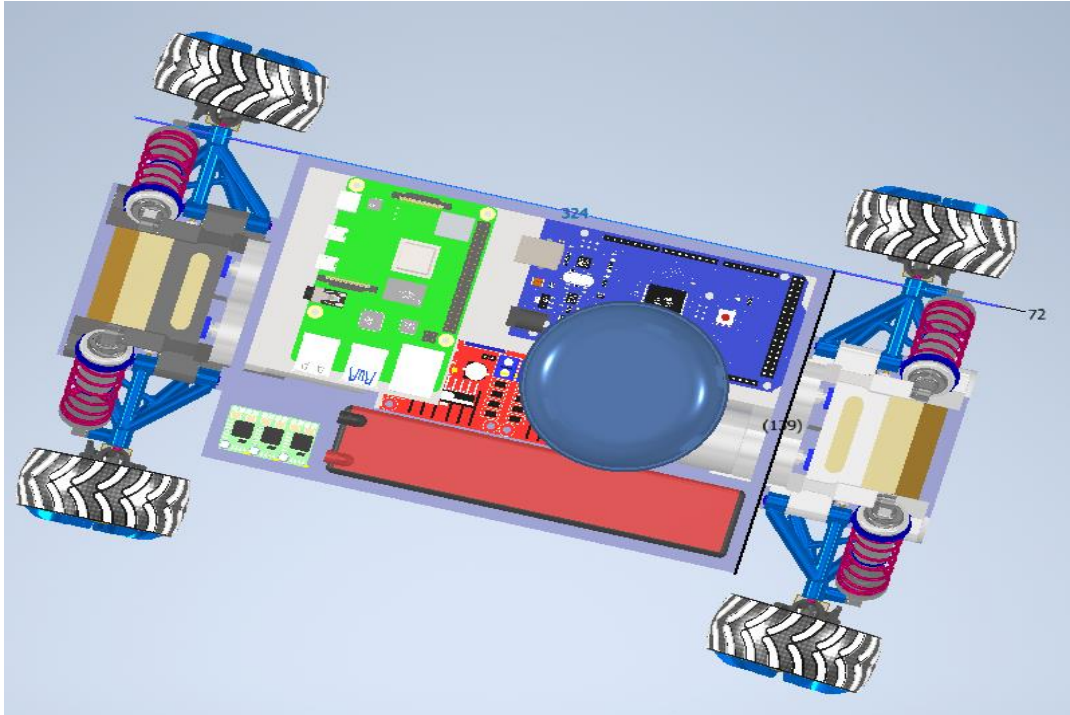


Figura 74. Vista superior del prototipo del diseño de un robot móvil utilizado para evaluar las condiciones actuales del agua. [Autor]

4.6 Referencias bibliográficas

- Terneus-Jácome, E., & Yánez, P. (2018). Principios fundamentales en torno a la calidad del agua, el uso de bioindicadores acuáticos y la restauración ecológica fluvial en Ecuador. LA GRANJA. Revista de Ciencias de la Vida, 27(1), 36-50.
- [2] Cando Cisneros, J. A., & Jibaja Robles, S. D. (2018). Diseño del sistema de alcantarillado combinado y planta de tratamiento de aguas residuales para la zona baja del barrio El Rosario de la parroquia de Píntag, cantón Quito, provincia de Pichincha (Bachelor's thesis, Quito: UCE).
- [3] Fernández, R. L. (2012). Los macroinvertebrados acuáticos como indicadores del estado ecológico de los ríos. Páginas de información ambiental, (39), 24-29.
- [4] Del Ecuador, A. C. (2008). Constitución de la República del Ecuador. Quito: Tribunal Constitucional del Ecuador. Registro oficial Nro, 449, 79-93.
- [5] Ladrera Fernández, R., & Rieradevall Sant, M. (2013). Macroinvertebrados acuáticos como indicadores biológicos: una herramienta didáctica. Ikastorratza. e-Revista de didáctica.
- [6] (Murillo-Montoya et al., 2018)
- [7] *([PDF] Determinación de Cadmio En Eichhornia Crassipes y Otras Plantas Acuáticas En Los Ríos Guayas, Daule y Babahoyo: Propuesta de Monitoreo Ambiental | Semantic Scholar, n.d.)*
- [8] *(La Contaminación de Los Ríos, n.d.)*
- [9] (Hawksworth et al., 2005)
- [10] *(Bacteria Coliformes Fecales Tesis - Buscar Con Google, n.d.)*
- [11] (Olmo Rísquez & Olmo Rísquez, 1998)
- [12] (Ciencias Manejo Sustentable De Recursos Bioacuáticos Y Medio Ambiente, n.d.)
- [13] *(Los Macroinvertebrados y Su Importancia En Los Cuerpos de Agua, n.d.)*
- [14] (F. De et al., n.d.)
- [15] *(UNIVERSIDAD POLITÉCNICA SALESIANA SEDE CUENCA CARRERA DE INGENIERÍA AMBIENTAL "DETERMINACIÓN DEL ÍNDICE DE CALIDAD DE AGUA MEDIANTE EL, n.d.)*
- [16] *(INTELIGENCIA ARTIFICIAL 101 COSAS QUE DEBES SABER HOY SOBRE NUESTRO FUTURO INTELIGENCIA ARTIFICIAL, 2018)*
- [17] (Arcega et al., n.d.)

[18] (Zhang et al., 2017)

[19] (Salcedo-Sanz et al., 2014)

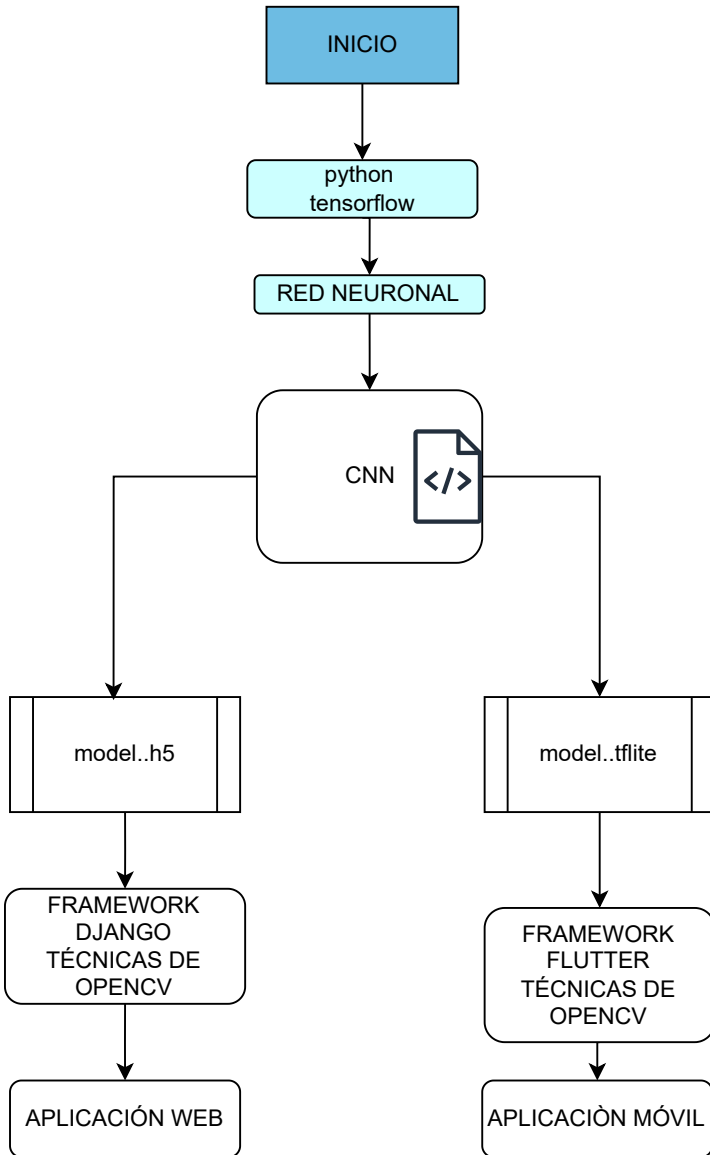
[20] (*TAZ-TFG-2021-3094*, n.d.)

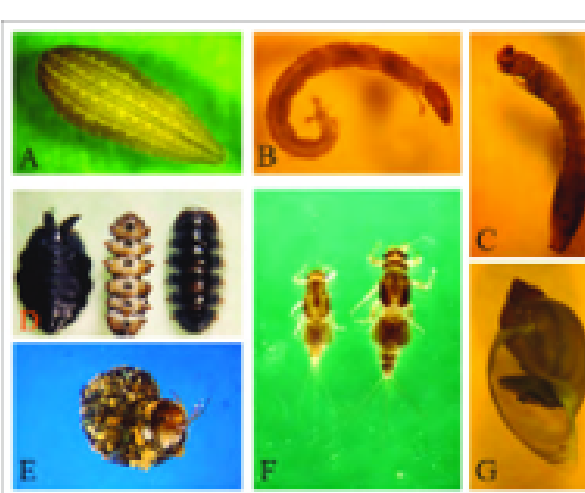
[21] (Shinde & Shah, 2018)

[22] (*UNIVERSIDAD TÉCNICA DE AMBATO*, n.d.)

4.7 Anexos

ARQUITECTURA DE LA APLICACIÓN:





Adquisición de datos

Cargar el dataset: 5002 imágenes (220x220)

80% entrenamiento (4002)

20% validación (1000)

MODELO SECUENCIAL

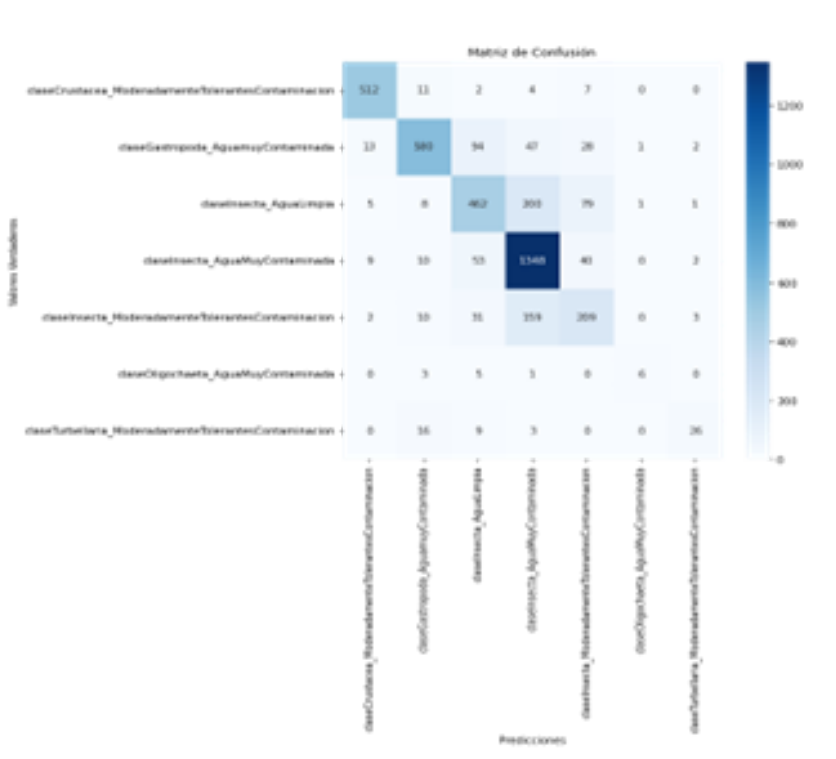
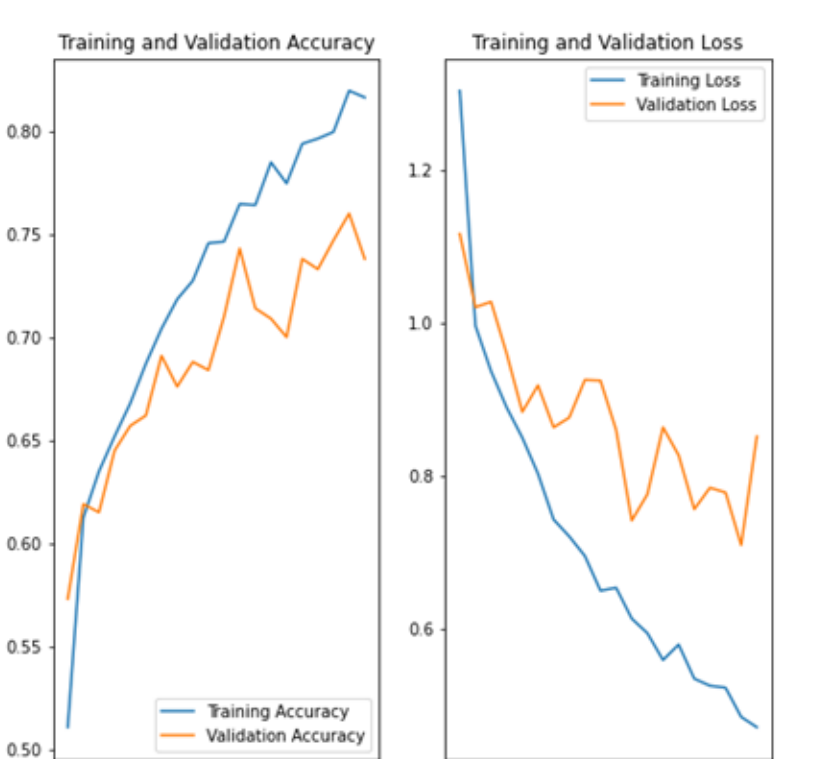
```
num_classes = len(class_names)
model = Sequential([
    layers.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])
```

APLICACIÓN DE TÉCNICAS DE MEJORAMIENTO EN LA RED (DATA AUMENTATION-DROPOUT)


VISUALIZACION DE RESULTADOS

MODEL.H5

APLICACIÓN WEB



Datos



Información

- Nombre : planariidae32.png
- Esta imagen pertenece a : **Turbellaria**
- Porcentaje de confianza : " **81.6626250743866** "

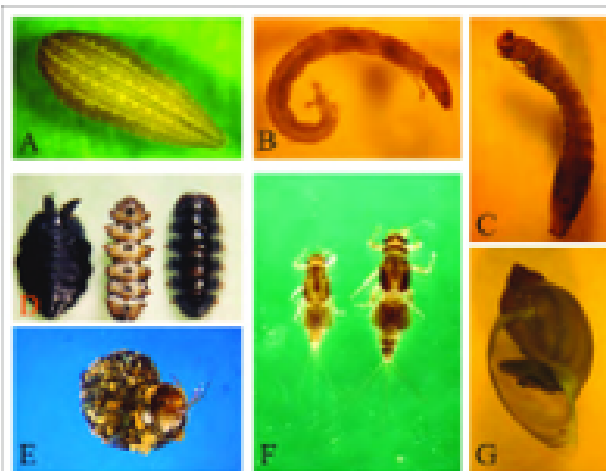
Análisis y recomendaciones

Análisis

- Índice BMWP/Cob: 42,3-48,7-37,7- calidad de agua dudosa clase III
- Punto de monitoreo: parte media y baja del río (M4-Daniel Durán, M5-Chamana, M6-Las Orquídeas)
- EL AGUA ES MODERADAMENTE CONTAMINADA

Recomendaciones

- No es recomendable beber agua moderadamente contaminada, ya que cualquier nivel de contaminación puede tener efectos negativos en la salud a largo plazo. Si no tiene acceso a agua limpia y segura, puede tomar medidas para reducir el riesgo de enfermedades. Hervir el agua, filtrar el agua, desinfectar el agua.



Adquisición de datos

Cargar el dataset:
5002 imágenes
(220x220)

80%
entrenamiento
(4002)

20%
validación
(1000)

MODELO
SECUENCIAL

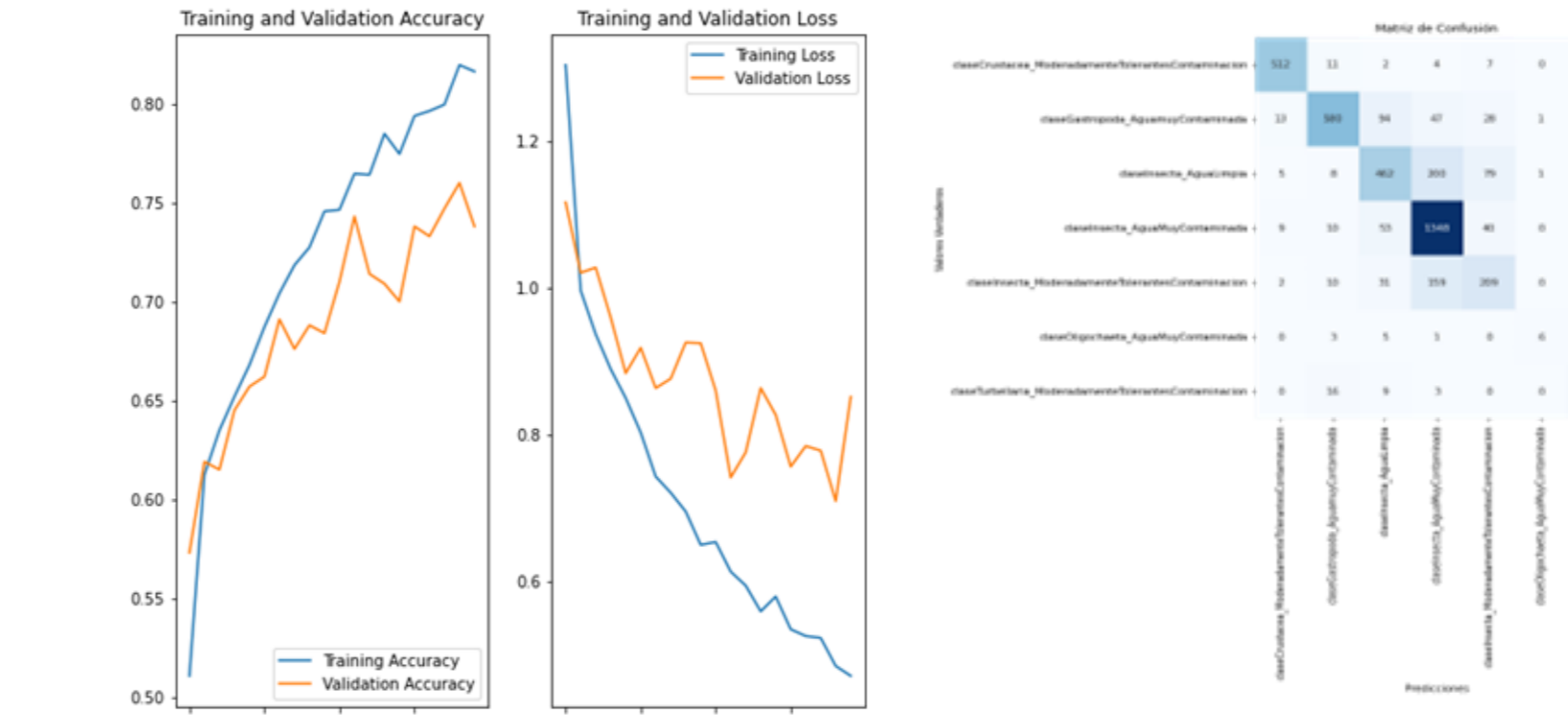
```
num_classes = len(class_names)
model = Sequential([
    layers.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])
```

APLICACIÓN DE
TÉCNICAS DE
MEJORAMIENTO
EN LA RED(DATA
AUMENTATION-
DROPOUT)

VISUALIZACION
DE RESULTADOS

MODEL.TFLITE

APLICACIÓN
MÓVIL



Calidad del Agua

Oligochaeta
Accuracy: 95.86%

- BMWP/Col: 31,0 calidad de agua crítica clase VI
- Parte baja del río-(M7-Camino a Patamarca,M8-Parque Industrial)

EL AGUA ES MUY CONTAMINADA

Cámara

Galería