UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

UNIDAD ACADÉMICA DE INGENIERÍA, INDUSTRIA Y CONSTRUCCIÓN

CARRERA DE INGENIERÍA AMBIENTAL

DISEÑO DE UN ALGORITMO COMPUTACIONAL DE IDENTIFICACIÓN DE MACROINVERTEBRADOS BASADOS EN PARÁMETROS DE TAXONOMÍA

TRABAJO DE TITULACIÓN O PROYECTO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO

AUTOR: PAOLA SOLEDAD CASTRO CALLE

DIRECTOR: ING. DIEGO AQUILES HERAS BENAVIDES

CUENCA- ECUADOR

2021

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

UNIDAD ACADÉMICA DE INGENIERÍA, INDUSTRIA Y CONSTRUCCIÓN

CARRERA DE INGENIERÍA AMBIENTAL

DISEÑO DE UN ALGORITMO COMPUTACIONAL DE IDENTIFICACIÓN DE MACROINVERTEBRADOS BASADOS EN PARÁMETROS DE TAXONOMÍA

TRABAJO DE TITULACIÓN O PROYECTO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO

AUTOR: PAOLA SOLEDAD CASTRO CALLE

DIRECTOR: ING. DIEGO AQUILES HERAS BENAVIDES

CUENCA – ECUADOR

2021

DECLARACIÓN

Yo, Paola Soledad Castro Calle, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento; y eximo expresamente a la Universidad Católica de Cuenca y a sus representantes legales de posibles reclamos o acciones legales.

La Universidad Católica de Cuenca puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y la normatividad institucional vigente.

Paola Soledad Castro Calle

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado bajo mi supervisión.

Ing. Diego Aquiles Heras Benavides

DIRECTOR

AGRADECIMIENTOS

Primero agradezco a Dios, por haberme guiado durante toda mi carrera y por ser mi fortaleza en los momentos de debilidad y por brindarme una vida llena de experiencias.

Le doy gracias a mis padres por apoyarme en todo momento, por los valores inculcados y por haberme dado la gran oportunidad de tener una excelente educación en el transcurso de mi vida.

Agradezco a los ingenieros del Centro de Innovación, Investigación y Transferencia de Tecnología (CITT) de la Universidad Católica de Cuenca y a todos los docentes de la carrera de Ingeniera Ambiental que han sabido y guiarme por el camino del conocimiento.

DEDICATORIA

Este trabajo va dedicado principalmente a Dios por haberme permitido llegar hasta este momento tan importante de mi formación profesional.

A mis queridos padres y hermanos por su cariño, apoyo, paciencia y confianza que depositaron en mí, ya que ellos fueron los pilares fundamentales que permitieron realizar de este sueño una realidad.

ÍNDICE DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

RESUMEN

El objetivo del presente trabajo fue diseñar un algoritmo computacional de identificación de macroinvertebrados que utilice parámetros morfométricos extraídos de las imágenes basado en la inteligencia artificial. Por la problemática que presenta la identificación y extracción de las características morfométricas mediante el método de inspección visual, se aplicó el método automatizado del Histograma de Gradientes Orientados (HOG) para la extracción de las características. Se probaron varios algoritmos de Machine Learning y técnicas de Deep Learning con dos sets de datos, el primero contenía 504 observaciones, siendo una cantidad pequeña ya que para la aplicación de estos algoritmos es necesario una base de datos extensa, entre más datos se tenga mejor es el entrenamiento y el rendimiento de los algoritmos, el segundo, para lograr los resultados esperados se incrementó artificialmente la base de datos mediante el método de Image Data Generator en Keras a 2242 observaciones. Como resultado de esta experimentación se determinó que el modelo de clasificación más adecuado para el caso es el clasificador Random Forest con la aplicación del método automatizado de extracción de características HOG por su Accuracy/precisión del 100%.

Palabras clave: algoritmos computacionales, características morfométricas, histograma de gradientes orientados, bioindicadores, macroinvertebrados, random forest.

ABSTRAC

This his work aimed at designing a computational algorithm for macroinvertebrate identification using morphometric parameters extracted from images based on artificial intelligence. Because of the problem of identifying and extracting morphometric features using the visual inspection method, the automated Histogram of Oriented Gradients (HOG) method was applied for feature extraction. Several Machine Learning algorithms and Deep Learning techniques were tested with two data sets, the first one contained 504 observations, being a small amount since for the application of these algorithms an extensive database is necessary, the more data the better the training and performance of the algorithms, the second one, to achieve the expected results the database was artificially increased employing the Image Data Generator method in Keras to 2242 observations. As a result of this experimentation, it was determined that the most suitable classification model for the case is the Random Forest classifier with the application of the automated HOG feature extraction method for its Accuracy/precision of 100%.

Keywords: computational algorithms, morphometric features, histogram of oriented gradients, bioindicators, macroinvertebrates, random forest.

CAPÍTULO I

1. INTRODUCCIÓN

Los seres humanos somos capaces de detectar y reconocer la mayoría de objetos presentados en imágenes con extrema facilidad, incluso si sufren variaciones de forma, tamaño, color, brillo, textura o están parcialmente solapados. En algunos casos se complica esta identificación, generalmente cuando se trabaja con seres vivos, con especies de tamaños relativamente pequeños y con alto nivel de detalle en sus características morfométricas, es por eso que para conseguir identificar especies sin ayuda de un experto en el campo se pretende diseñar un algoritmo computacional de reconocimiento automático, basado en los parámetros morfométricos de las especies y en las ciencias de la computación, mediante algoritmos para extraer características morfométricas de imágenes, la aplicación y evaluación de técnicas computacionales de Deep Learning y Machine Learning, como herramientas que faciliten el proceso de clasificación taxonómica a través de los parámetros morfométricos que presenten según el taxón (Pérez, 2012).

El propósito del aprendizaje supervisado es predecir el valor de una variable de salida, en base a múltiples variables de entrada, llamadas variables predictoras. La aplicación y el interés de Machine Learning, durante las últimas décadas han experimentado un gran crecimiento, convirtiéndolo en una ciencia aplicable en todos los campos de investigación industrial y académica. Da lugar a que conjuntos de algoritmos puedan identificar patrones presentes en los datos y con ello crear modelos. Es importante no olvidar que los sistemas de Machine Learning solamente pueden identificar lo visto anteriormente, no tienen la capacidad de memorizar patrones que no hayan estado presentes en los datos de entrenamiento. (Amat, 2018).

Por otra parte, como bioindicadores de la calidad del agua los macroinvertebrados son eficientes, sin embargo, la identificación de las familias es difícil de realizar debido a las pequeñas características que los diferencian unos de otros. Los macroinvertebrados son consumidores primarios y secundarios, frecuentemente son abundantes, sedentarios, su recolección es simple y barata y brindan información de largos periodos (Mosquera, 2015).

Los límites dimensionales de los macroinvertebrados dulceacuícolas no están claramente definidos, Springer (2010) define a los macroinvertebrados como organismos que se pueden observar a simple vista o que pueden ser retenidos en mallas de 125µm. Las comunidades de macroinvertebrados bentónicos son utilizadas comúnmente para la evaluación del deterioro de la calidad del agua, debido a su amplia respuesta a diferentes gradientes de estrés. El Ecuador geográficamente está compuesto por una importante red hidrográfica, con gran cantidad de ríos originarios de los altos relieves andinos, los cuales vierten sus aguas en dos cuencas: Amazonas y Pacífico, por esta razón la biodiversidad de macroinvertebrados es amplia (Liñero et al., 2016).

En los ecosistemas dulceacuícolas existe un alto valor biológico por poseer una biota rica y diversa, incluyendo una amplia variedad de peces, organismos invertebrados macroinvertebrados, plantas, algas, zooplancton y fitoplancton, por lo que para el desarrollo de la vida son considerados uno de los recursos naturales renovables más importantes. La calidad del agua no sólo puede estar determinada por sus características fisicoquímicas, por esta razón es que el uso de los macroinvertebrados acuáticos bentónicos como bioindicadores, tiene cada vez más aceptación en todo el mundo. Pese a su alta diversidad, abundancia y grado de adaptación, la identificación

de los taxas que componen la comunidad bentónica es escasa e incompleta (Encalada, et al., 2011).

Lograr una identificación eficaz de los macroinvertebrados a nivel familia, en gran parte depende de una correcta clasificación, en base a esto, el presente estudio estudio tiene como objetivo realizar el diseño de un algoritmo de preprocesamiento de imágenes para la extracción de las características morfométricas, mediante la implementación de algoritmos de aprendizaje supervisado y Deep learning para el entrenamiento y evaluación del rendimiento de máquinas de aprendizaje que sean capaces de clasificar correctamente las imágenes de macroinvertebrados, debido a las limitaciones que presenta la visión humana.

Actualmente están disponibles distintas guías de identificación de macroinvertebrados para ríos andinos, el algoritmo previamente determinado facilitará el proceso de identificación a nivel de familia, lo que facilitará la determinación de los índices preestablecidos como el ABI o el BMWPCol, ya que solamente se necesitarán ciertas características morfométricas proporcionadas por el usuario como: longitud, ancho, forma del cuerpo, segmentos corporales, número de patas, cubierta de pelos, antenas.

Objetivo general

Diseñar un algoritmo computacional de identificación de macroinvertebrados que utilice parámetros morfométricos de las imágenes basado en inteligencia artificial.

Objetivos específicos

- Diseñar métodos de extracción de características de las imágenes de macroinvertebrados mediante la definición de parámetros para la inspección visual o con técnicas automáticas de preprocesamiento de imágenes.
- Implementar algoritmos de máquinas de aprendizaje supervisado con machine learning y Deep learning para la clasificación automática de las imágenes de macroinvertebrados.
- Evaluar el mejor algoritmo para la clasificación de imágenes de macroinvertebrados mediante métricas de medición del error en la clasificación de imágenes.

CAPÍTULO II

2. REVISIÓN DE LITERATURA

2.1 Algoritmo computacional

2.1.1 Definición.

Es un método para resolver problemas por medio de una secuencia de pasos lógicos con el fin de realizar una tarea específica. En el lenguaje de programación se denomina como la metodología indispensable para la solución de problemas mediante el uso de programas (Joyanes, 2008).

La propuesta para la solución de problemas establecidos mediante algoritmos es la siguiente (Vasquez, 2012.):

1. Diseño del algoritmo: describe la secuencia a seguir que conduce a la solución del problema.
2. Fase de codificación: expresa como un programa en el lenguaje de programación adecuado al algoritmo.
3. Ejecución y validación del programa: realizado a través de un computador.

2.1.2 Partes.

Los algoritmos presentan tres partes: entrada, proceso y salida, descritas brevemente a continuación (Vasquez, 2012.):

- Entrada: llamado también inicio, punto de partida o cabecera, es la primera instrucción que da inicio al algoritmo y que origina su lectura.
- Proceso: es la elaboración precisa propuesta por el algoritmo, es decir, el cuerpo de sus claves para formular una instrucción.
- Salida: son las instrucciones puntuales resueltas por el algoritmo. También se llama cuerpo, pie o fin.

2.1.3 Características.

Los algoritmos computacionales deben cumplir con las siguientes características fundamentales (Vasquez, 2012.):

- Preciso: indica el orden a seguir en cada paso.
- Definido: el resultado a obtener debe ser el mismo, en caso de seguir un algoritmo varias veces.
- Finito: debe dar un resultado al final de sus pasos, en algún momento debe terminar.

2.1.4 Diseño.

Los algoritmos son independientes tanto del lenguaje de programación como del computador que lo ejecuta, una manera de resolver de forma eficaz los problemas complejos es dividir en subproblemas más fáciles de resolver que el original.

Generalmente, en el primer esbozo del algoritmo los pasos diseñados son incompletos. Tras esta primera descripción, se realiza un proceso denominado refinamiento del algoritmo, en el cual se amplían en una descripción más detallada y específica (Joyanes, 2008).

El objetivo del diseño es lograr que el algoritmo aprenda de forma automática las propiedades deseadas, estableciendo el tipo de modelo a utilizar, las variables a

incorporar y la formación del conjunto de entrenamiento mediante el procesamiento de la información (Izuareta, 2011).

2.1.5 Representación gráfica.

Para la representación gráfica se debe emplear un método que sea independiente del lenguaje de programación elegido. Joyanes Aguilar (2008) indica que el algoritmo debe ser representado de manera gráfica o numérica de tal manera que los pasos sucesivos no dependan de la sintaxis de los lenguajes de programación, sino que para la ejecución en un programa se emplee la descripción. Los métodos usuales para representar un algoritmo son:

- Diagrama de flujo
- Diagrama N-S (Nassi-Schneiderman)
- Lenguaje de especificación de algoritmos: pseudocódigo
- Lenguaje español, inglés
- Fórmulas

2.2 Inteligencia artificial

2.2.1 Definición.

Es una rama de las ciencias de la computación desarrolladora de procesos que imitan la inteligencia de los seres vivos. La aplicación principal es la creación de máquinas para automatizar labores que requieran de un comportamiento inteligente (Serrano, 2011).

Los siguientes autores definen la inteligencia artificial como:

Kurzweil, 1990: “El arte de crear maquinas con capacidad de realizar funciones que realizadas por personas requieren de inteligencia.”

Rich y Knight, 1991: “El estudio de cómo lograr que las computadoras realicen tareas que, por el momento, los humanos hacen mejor.”

Schalkoff, 1990: “Un campo de estudio que busca explicar y emular el comportamiento inteligente en términos de procesos computacionales”

2.2.2 Técnicas.

Las técnicas de la inteligencia artificial se apoyan en los conceptos de otras disciplinas, como en: la Ingeniería, Informática, Sociología, Psicología Cognoscitiva, Economía, etc. La aplicación de la inteligencia artificial está orientada a la satisfacción de diferentes necesidades en varias disciplinas (Sánchez, 1993).

a. Sistemas expertos.

Se deriva del término “sistema experto basado en conocimiento”. Es un software que simula el comportamiento de los humanos expertos para encontrar la solución a un problema tomando decisiones en un área específica, hasta un nivel de complejidad que maneja un experto (Turban, 1995).

b. Redes neuronales.

A través de modelos matemáticos computacionales imitan propiedades de un sistema neuronal, creados artificialmente por mecanismos con el fin de que sea un sistema de respuesta parecido al de un cerebro, por lo tanto, es un tipo de aprendizaje automático. Este modelo considera la representación de una neurona como una unidad binaria (Gutierrez, Wolfgang & Ospina, 2006).

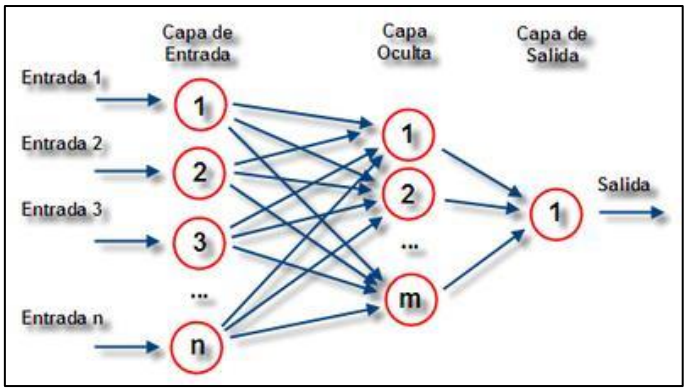

Figura 1: Funcionamiento de una red neuronal clásica
Fuente: Laguna, 2016

Se organizan en capas en donde cada una se conecta con las de la siguiente, siendo la operación principal una multiplicación entre los valores de una neurona y su conexión saliente. Los resultados son recibidos por las neuronas anteriores, son sumados en uno y aplicados a una función no lineal para crear un nuevo resultado. La función más común es la sigmoide, una de sus características es que la interpretación de su salida es una probabilidad debido a que su valor se encuentra en el rango [0, 1] (Laguna, 2016).

c. Aprendizaje automático (Machine Learning).

También llamado aprendizaje de máquina o aprendizaje automático, es una técnica que se encarga de hacer que las computadoras sean capaces de aprender y realicen acciones sin necesidad de programación explícita. El aprendizaje hace referencia a identificar patrones en una gran cantidad de datos y a través de ellos predecir comportamientos futuros de una situación utilizando teorías de probabilidad y algoritmia estadística (Torroledo & Beltrán, 2016)**.**

Normalmente este aprendizaje es de dos tipos: supervisado y no supervisado. En el supervisado los algoritmos se entrenan con una hipótesis trabajando con datos “etiquetados” intentando hallar la función que asigne una variable de salida adecuada dadas las variables de entrada. De este método se reconocen problemas de clasificación y problemas de regresión (Joyanes, 2008).

En el no supervisado, comúnmente se trabaja con entradas aleatorias al sistema, es decir, no están disponibles datos “etiquetados” para la fase de entrenamiento. La salida representa el grado de similitud entre la información presentada a la entrada y la información mostrada a la salida hasta ese momento (Rodríguez, 2018).

d. Aprendizaje profundo (Deep Learning).

Es un conjunto de algoritmos que representan el proceso que realizan las neuronas cerebrales para el reconocimiento de palabras, voces o imágenes. Se trata de un procesador de información que recibe entradas codificadas como números y después de pasar por una secuencia de operaciones matemáticas produce información de salida también codificada. Esta técnica además de ser usada para problemas de clasificación, también tiene una gran función en problemas de aprendizaje no supervisado (Rodríguez, 2018).

2.3 Preprocesamiento de imágenes

El preprocesamiento de imágenes consiste en el conjunto de técnicas que buscan mejorar la visualización de una imagen a una forma más adecuada para el

observador humano o análisis artificial. Este proceso incluye técnicas de eliminación de ruido, realce de detalles mediante la iluminación, binarización de imágenes y detección de bordes.

2.3.1 Imagen.

Una imagen se define como una función de dos dimensiones f(x,y) donde x e y son las coordenadas de un plano que contiene todos los puntos de la misma, y f(x,y) es la amplitud en el punto (x,y) a la cual se le llama intensidad o nivel de gris de la imagen en ese punto.

2.3.2 Imagen digital.

Es una imagen digital cuando las coordenadas x e y como valores de intensidad de la función f son discretos y finitos, está compuesta de un número finito de elementos tendiendo un valor y una localidad particular para cada uno. A estos elementos se les llama puntos elementales de la imagen o comúnmente conocidos como píxeles que funcionan por medio de números del 0 blanco al 255 negro, pasando por una gama de diferentes tonos grises (Córdoba, 2001).

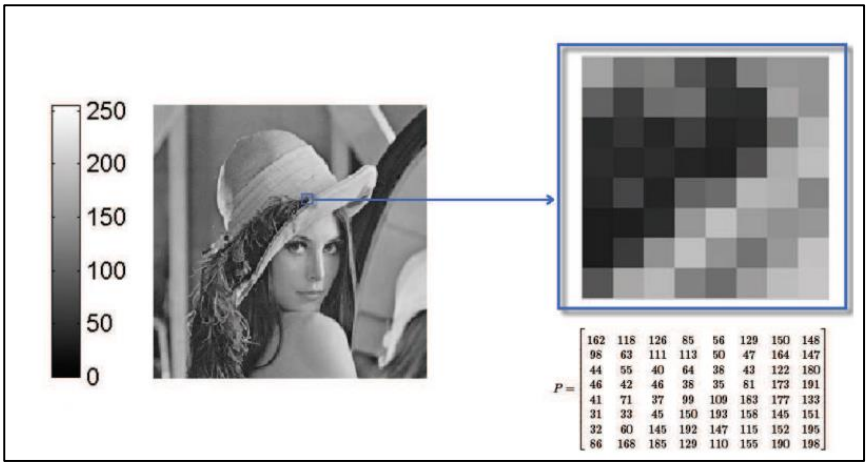

Figura 2: Imagen con 256 niveles de intensidad
Fuente: Serpa, 2014

El autor Webster define imagen como “una representación hecha por medios de dibujos, pinturas o fotografía”, además, relaciona la palabra digital con el cálculo por métodos numéricos o por unidades discretas. Por lo tanto, una imagen digital se define como la representación numérica de un objeto.

2.3.3 Píxel.

El píxel en una imagen digital es la unidad más pequeña. Para formar una imagen completa se necesita un inmensurable número de ellos, cada uno es una unidad homogénea de color que en resulta una imagen relativamente compleja.

Las imágenes según el brillo que contiene cada píxel se clasificar de la siguiente manera:

- **Imágenes bitonales:** se componen de dos colores, blanco y negro con valores de 255 y 0 respectivamente.
- **Imágenes en escala de grises**: compuestas de 256 niveles de una gama de grises

- **Imágenes a color:** formadas de tres matrices monocromáticas con 256 niveles de representación, rojo, verde y azul, RGB por sus siglas en inglés (Mejía, 1996).

Figura 3: Imágenes bitonal, escala de grises y color
Fuente: Ferrero & Luján, 2008

En la Figura 3, de izquierda a derecha se distingue las imágenes digitales: la primera representa una imagen bitonal (1 bit), la segunda una a escala de grises (8 bits) y la tercera representa una imagen a color (24 bits) (Ferrero & Luján, 2008).

2.3.4 Formación de la imagen.

Para formar una imagen digital la trayectoria que sigue la cámara es la siguiente:

1. La luz es detectada por el objetivo de la cámara
2. La luz llega hasta el sensor de imagen formado por abundantes receptores fotosensibles llamados fotodiodos.
3. La luz genera una señal eléctrica pequeña a cada receptor, que posteriormente, se transformará en datos digitales (Alvarado, 2012).

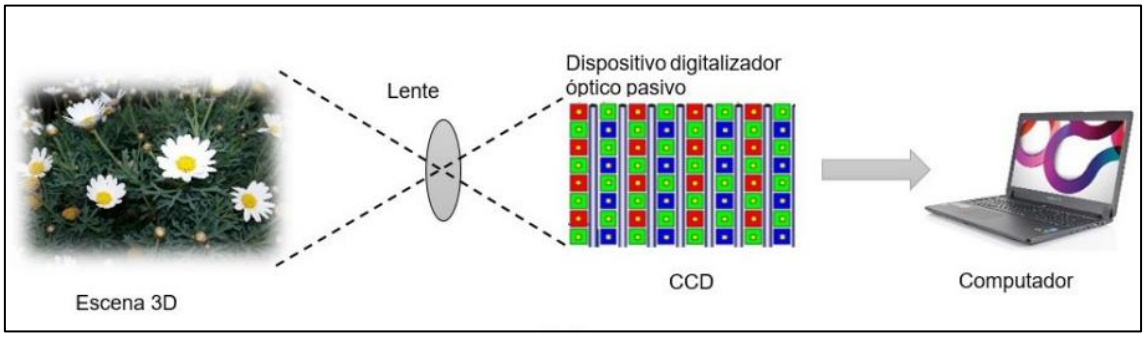

Figura 4: Formación de una imagen digital
Fuente: Heras, 2017

2.3.5 Redimensionamiento de la imagen.

Uno de los procesos de transformación importante es el redimensionamiento de imágenes para ello se utiliza el parámetro “ratio” (r), debido a que redimensiona la imagen sin distorsionar su aspecto. El valor de este se calcula con la siguiente ecuación:

$$r = \frac{\text{nuevo ancho}}{\text{ancho actual}} \quad (1)$$

Con este proceso las imágenes digitalizadas tendrán un ancho estándar, es importante recalcar que, al adquirir las imágenes estas deben tener el contraste equilibrado con la iluminación para no perder información relevante.

2.3.6 Variables del color.

Las propiedades innatas del color se describen a continuación.

- **Matiz:** es el valor cromático que recibe un color, depende de la longitud de onda dominante y permite la clasificación de los colores rojo, amarillo y violeta.
- **Luminosidad:** es la cantidad de luz presente en un color resultante de la mezcla de los colores con el blanco o con el negro.
- **Saturación**: hace referencia a la pureza de un color con relación al gris (Aguirre, 2015.)

2.3.7 Modelo RGB.

La descripción RGB (del inglés Red, Green, Blue) es la composición del color basado en la intensidad de los colores primarios. Es uno de los modelos más utilizados, basado en la llamada “síntesis aditiva”, para la creación y reproducción de los colores en monitores, en donde se suman las intensidades de la luz roja, verde y azul, también incluye el blanco y el negro (Flores, 2016).

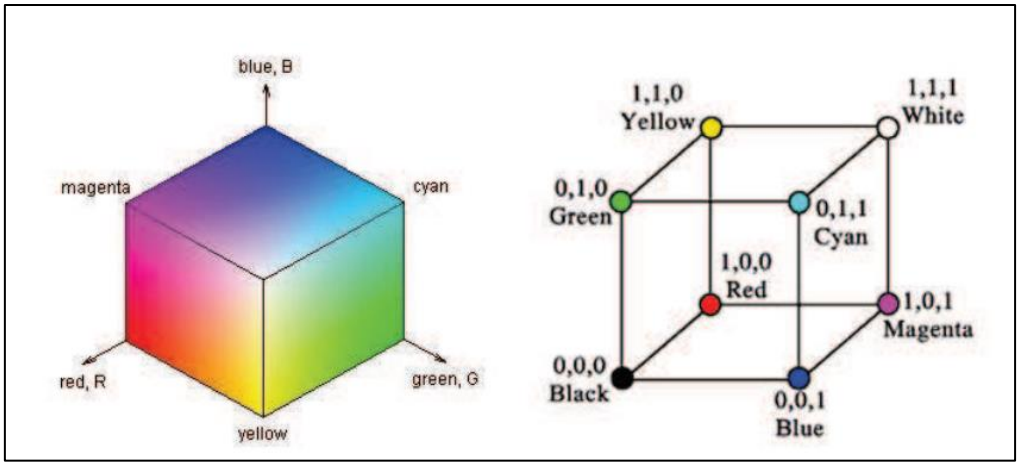

Figura 5: Representación gráfica del modelo RGB
Fuente: Aguirre, 2015

La Figura 5 corresponde a la representación gráfica del modelo RGB realizado mediante un cubo unitario con los ejes R, G y B (Aguirre, 2015.).

2.3.8 Imágenes a escala de grises.

En la escala de grises el valor de cada píxel es únicamente una muestra de la cantidad de luz, es decir la intensidad. Las imágenes a escala de grises se diferencian de las binarias porque estas tienen una gran cantidad de tonos de grises por medio, en comparación del blanco y negro. Las imágenes a escala de grises son el resultado del grado de la intensidad de la luz en cada uno de los píxeles según la combinación ponderada de frecuencias (Flores Ereña, 2016).

2.3.9 Conversión de RGB a escala de grises.

Utilizar los principios de colorimetría y fotometría es la estrategia más común que se utiliza para calcular los valores de la escala de grises y conseguir una luminancia igual que las imágenes a color. Para obtener un solo gris partiendo de un color basado en un modelo RGB típico se mezclan los siguientes porcentajes: Rojo 30%, Verde 59%, Azul 11%, según expertos estas cantidades son las más parecidas a como la visión humana percibe la intensidad de la luz dependiendo del color. Para realizar la conversión se aplica la ecuación de luminancia a cada pixel de la imagen RGB. Resultando una matriz de luminancia de un byte por pixel, creando una nueva paleta de grises (Cortes, Urueña & Mendoza, 2011).

$$Y = R * 0{,}3 + G * 0{,}59 + B * 0{,}11 \quad (2)$$

2.4. Lenguaje de programación *PYTHON*

Es un lenguaje de programación versátil, multiplataforma y multiparadigma, sencillo y fácil de aprender. Representa un alto nivel y permite procesar fácilmente todo tipo de datos ya sean numéricos o de texto. (Rodríguez, 2018).

En este proyecto, el lenguaje *Python* es aplicado para el preprocesamiento de las imágenes de los macroinvertebrados. Es una etapa muy importante para aumentar el acierto de los modelos, busca mejorar la calidad y facilitar la información aplicando técnicas de realce de contraste, supresión de ruido, extracción de borde, optimización de la distribución de la intensidad, etc. Para ello se usa *OpenCV*, que es una librería de código abierto altamente optimizada para aplicaciones de visión artificial.

2.5 Keras - *ImageDataGenerator*

La generación de datos de imágenes es una técnica que se utiliza para aumentar de forma artificial el tamaño de un conjunto de datos de entrenamiento modificando las imágenes del conjunto de datos original. Keras es una biblioteca de aprendizaje profundo que proporciona una serie de diferentes técnicas de aumento como estandarización, rotación, cambios, volteretas, cambio de brillo y muchas más, esto se logra mediante el uso de *ImageDataGenerator* (Chollet, 2018).

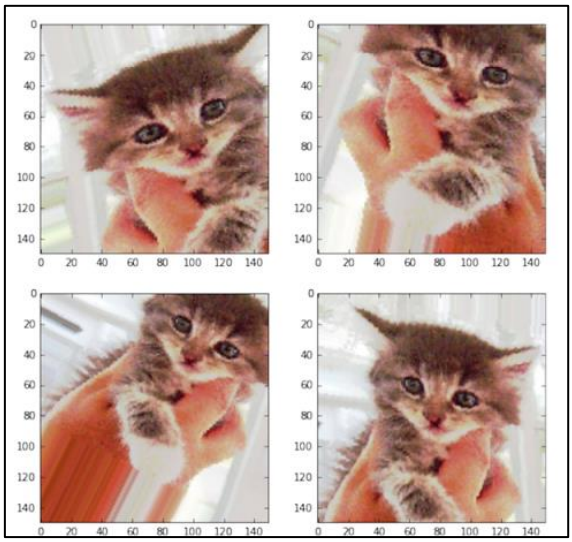

Figura 6: Generador de imágenes mediante aumento de datos aleatorios
Fuente: Chollet, 2018

2.6 Lenguaje de programación R

Según la información oficial ofrecida por el proyecto R3: “R es un lenguaje y entorno de libre disposición para la computación estadística y los gráficos que proporciona una amplia variedad de técnicas estadísticas y gráficas, modelado lineal y no lineal, pruebas estadísticas, análisis de series de tiempo, clasificación, agrupamiento, etc.”

Es uno de los lenguajes que domina en el campo de la estadística, data mining, *Deep learning y Machine Learning* siendo la herramienta de *Machine Learning* la más utilizada. (Paradis & Ahumada, 2003.)

2.7 Algoritmos con Machine Learning.

En Machine Learning, los algoritmos empleados se clasifican en tres conjuntos de aprendizaje (Simeone, 2018).

1. **Supervisado:** se aplica para problemas de regresión y/o clasificación. Utilizada cuando las etiquetas se asocian a ciertos datos y requieren de una predicción para ser aplicadas en otras instancias
2. **No supervisado:** aplicado en problemas de *clustering.* Favorece a las relaciones implícitas al disponer de información no clasificada.
3. **De refuerzo:** es una solución intermedia entre las anteriores. Retroalimenta las etapas de predicción, pero desconoce la etiqueta en particular.

2.7.1 Regresión logística.

La regresión logística es utilizada para clasificación, es un método lineal que en función de variables predictoras o independientes predice resultados de una variable categórica. Es muy útil en la modelación de probabilidades de ocurrencia de un evento en función de otros factores (Utrera, 2017).

Permite ver los valores categóricos de una clasificación como un cero y un uno, se utiliza cuando el interés es conocer si un evento ocurrirá o no. Para encontrar los valores con las variables independientes se utiliza la estimación *de Maximun likelihood* o la máxima verosimilitud (Amat, 2018).

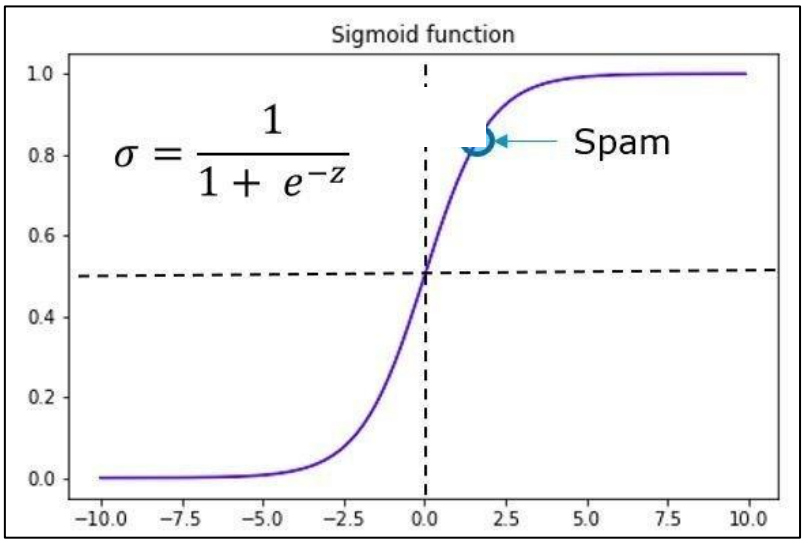

Figura 7: Modelo de regresión logística
Fuente: Recuero de los Santos, 2017

- Su valor máximo es 1 y su valor mínimo es 0.
- Se interpretan los resultados como probabilidades.
- En problemas de clasificación binaria, lo valores superiores a 0,5 pertenecen a la clase 1, mientras que los valores menores a 0,5 a la clase 0 (Amat, 2018).

2.7.2 Método de los K-Vecinos más cercanos.

Sirve para predecir un valor numérico y clasificar un valor categórico, es llamado así porque clasifica cada nuevo ejemplo calculando la distancia de éste con todos los del conjunto de *train*. La clase predicha para el nuevo ejemplo se da por la clase a la que pertenezcan los ejemplos más cercanos del conjunto de *train*, el valor de la k es el que determina en cuantos vecinos se tiene en cuenta para predecir la clase. Así, con un valor de k = 1, la clase predicha para cada nuevo ejemplo será la clase a la que pertenezca el ejemplo más cercano del conjunto de train (Moujahid, Inza, & Larrañaga, 2017).

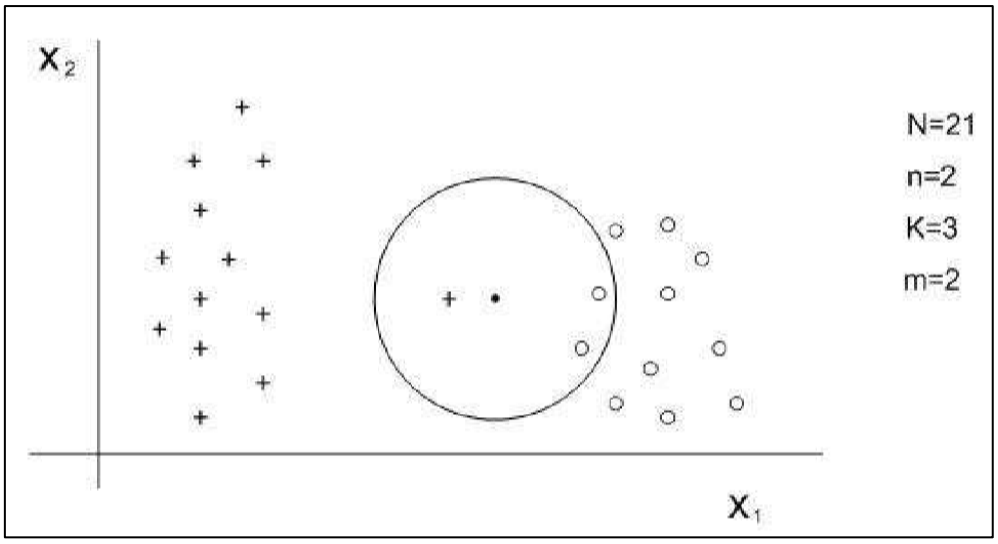

Figura 8: Modelo del método de los K-Vecinos más cercanos
Fuente: Moujahid, Inza, & Larrañaga, 2017

2.7.3 Máquina de soporte vectorial (SVM).

Las Máquinas de Soporte Vectorial o *Support Vector Machines* son un conjunto de algoritmos de aprendizaje supervisado relacionados con problemas de clasificación y regresión que construyen un hiperplano o un conjunto de hiperplanos en el espacio de dimensionalidad alto y con buena separación entre las clases. (Utrera, 2017).

Emplea un algoritmo de optimización para determinar la frontera óptima entre dos grupos, siendo un método de clasificación binario.Dado un conjunto de puntos de dos tipos en el lugar N dimensional, genera un hiperplano (N – 1) dimensional para separar esos puntos en dos grupos (Camacho, 2016).

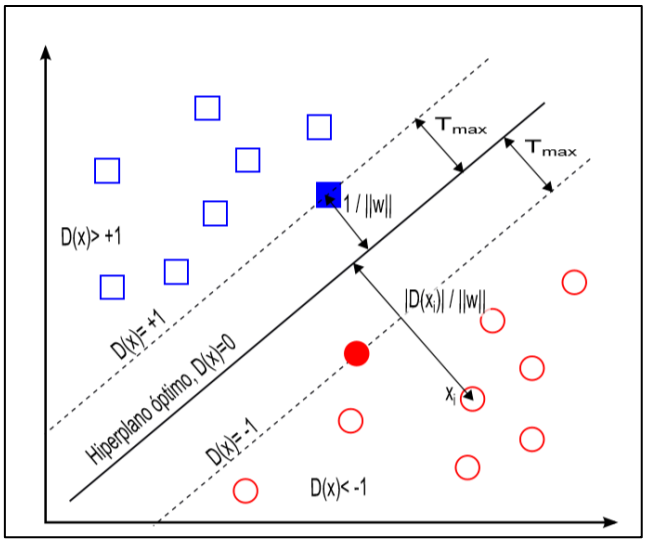

Figura 9: Modelo de una máquina de soporte vectorial
Fuente: Recuero de los Santos, 2017

2.7.4 Árboles de decisión.

La finalidad de los árboles de decisión es crear un modelo predictor del valor de una variable de salida en función de numerosas variables de entrada.

El árbol de decisión o de clasificación representa visualmente una estructura de decisiones compuesta de una serie de nodos internos y nodos externos conocidos como hojas del árbol, además de arcos que unen estos nodos. (Parra, 2016).

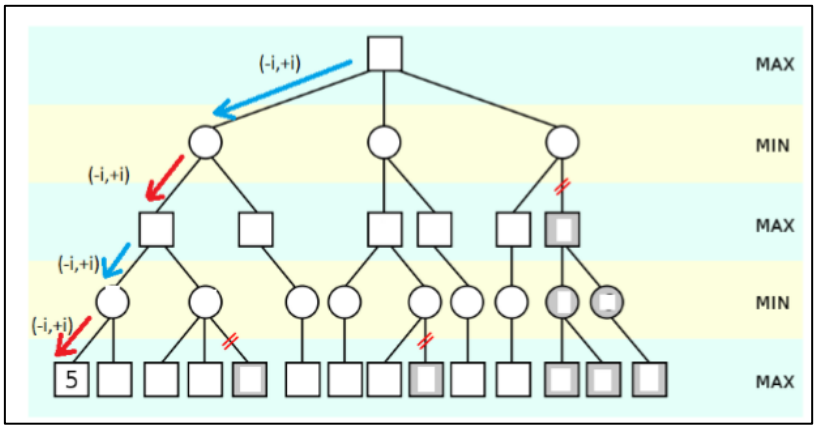

Figura 10: Modelo de árboles de decisión
Fuente: Parra,2016

2.7.5 Bosques Aleatorios o *Random Forest*

Es el conjunto de los árboles de decisión o de clasificación, trabaja con la recopilación de árboles correlacionados y los promedia dependiendo de los valores de un vector de la muestra aleatorio de forma independiente y con la misma cantidad de atribuciones de todos los árboles del bosque. (Hastie, et al., 2001).

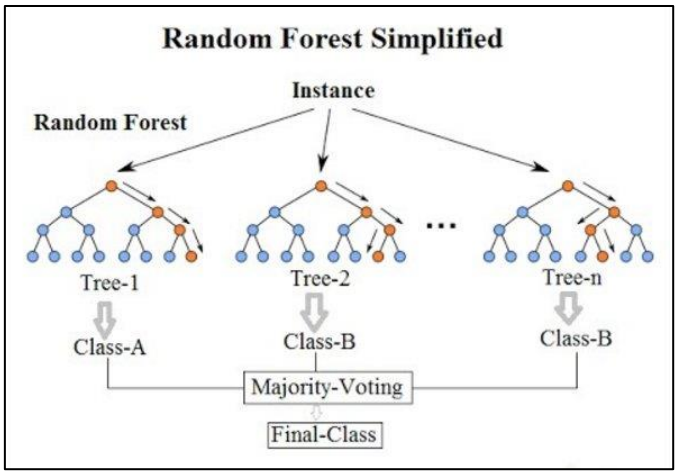

Figura 11: Modelo de bosques aleatorios
Fuente: Medina & Ñique, 2017

2.7.6 Naive Bayes.

Este algoritmo es aplicado específicamente para clasificación, reciben esta denominación porque están basados en el teorema de Bayes. Realizan clasificaciones de cada valor como independiente de cualquier otro, permitiendo predecir con mucha efectividad una clase o categoría dentro de un conjunto dado modelos probabilísticos (Larrañaga, Inza, & Moujahid, 2018).

Al ser empleado calcula las probabilidades de condición de que una observación pertenezca a cada una de las clases. Debido a que el algoritmo supone que las variables son independientes y no están influenciadas por las demás se le atribuye el término Naive que es ingenuo en inglés (Baez, 2013).

2.8 Método de *Deep Learning* en h2o

El *Deep Learning* en H2O es desarrollado en Java. Permite construir una red neuronal que toma un conjunto de entradas, las comprime y codifica, y luego intenta reconstruir la entrada con la mayor precisión posible. H2o implementa casi todos los algoritmos comunes de aprendizaje automático, como regresión lineal, regresión logística, *Naive Bayes, Random Forest* (Candel & Parmar, 2015).

Los cálculos de los parámetros del modelo global se pueden ejecutar en un solo nodo o en un clúster de múltiples nodos. Para un *clúster* de múltiples nodos, se entrena una copia de los parámetros del modelo global en los datos locales de un nodo, a través de computación paralela distribuida y multiproceso. El modelo se promedia en toda la red y cada nodo de cálculo contribuye periódicamente al modelo global (Candel & Parmar, 2015).

2.9 Método para la extracción de características HOG

Los Histogramas de Gradientes Orientados (HOG) es un método aplicado en la extracción de características, dividiendo la imagen en una serie de bloques que son distribuidos y solapados a lo ancho y largo de la misma. La Figura 12 representa de manera gráfica los pasos que sigue el método de extracción de características de los Histogramas de Gradientes Orientados (Jiménez, 2015).

Para realizar los cambios en el contraste e iluminación de las celdas adyacentes, los gradientes deben ser normalizados localmente agrupándose en bloques (Cordero, 2015).

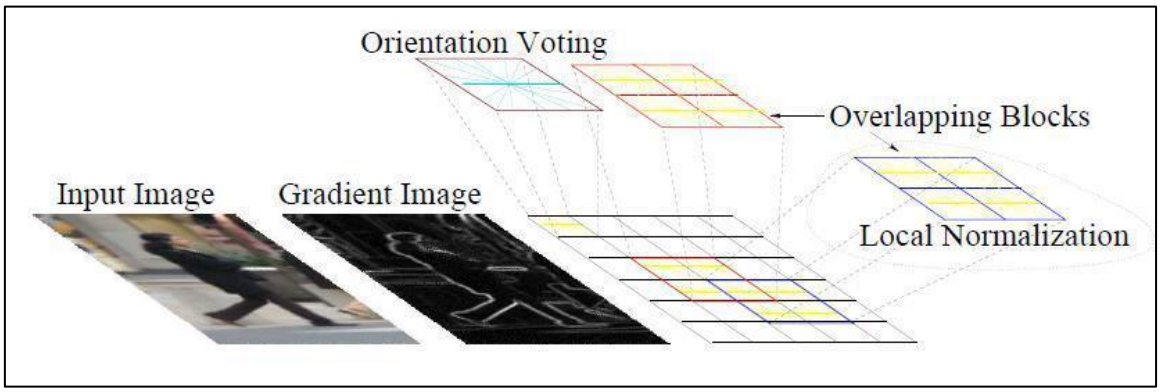

Figura 12: Desarrollo del método HOG
Fuente: Jiménez, 2015

2.10 Métricas de evaluación de las máquinas de aprendizaje

Si bien la preparación de los datos y el entrenamiento de un modelo son pasos claves en el proceso de aprendizaje automático, es igualmente importante medir el rendimiento del modelo entrenado, usando diferentes métricas generalmente utilizadas en recuperación de información, las cuales son adaptadas en función de los casos clasificados correctamente e incorrectamente (Mena, 2008).

2.10.1 Accuracy

Representa la porción de documentos que son clasificados correctamente sobre el total de casos, es el más utilizado en los problemas de clasificación (Elgueta, 2017).

$$Accuracy = \frac{número\ de\ predicción\ correcta}{número\ total\ de\ predicciones\ realizadas} \quad (3)$$

La exactitud también puede ser calculada en términos negativos y positivos en la clasificación binaria de la siguiente manera (Elgueta, 2017):

$$Accuracy = \frac{V_P + V_n}{V_p + V_n + F_p + F_n} \quad (4)$$

Donde:

V_p = verdaderos positivos
V_n = verdaderos negativos
F_p = falsos positivos
F_n = falsos negativos

2.10.2 Curva ROC

El área bajo la curva ROC es una métrica de rendimiento para problemas de clasificación binaria, representa mediante un gráfico la tasa de falsos positivos y la tasa

de verdaderos positivos para un conjunto dado de predicciones de probabilidad que se utilizan para asignar las probabilidades a las etiquetas de clase (Mena, 2008).

Un área de 1,00 representa un modelo que realizó todas las predicciones a la perfección, mientras que en 0,50 no hay diferencias en la distribución de los valores de la prueba entre los 2 grupos. Por ejemplo, el valor del área de 0,80 significa que un individuo aleatoriamente seleccionado del primer grupo tiene un valor de la prueba mayor que uno seleccionado aleatoriamente del segundo grupo en el 80% de las veces (Domínguez & González, 2002).

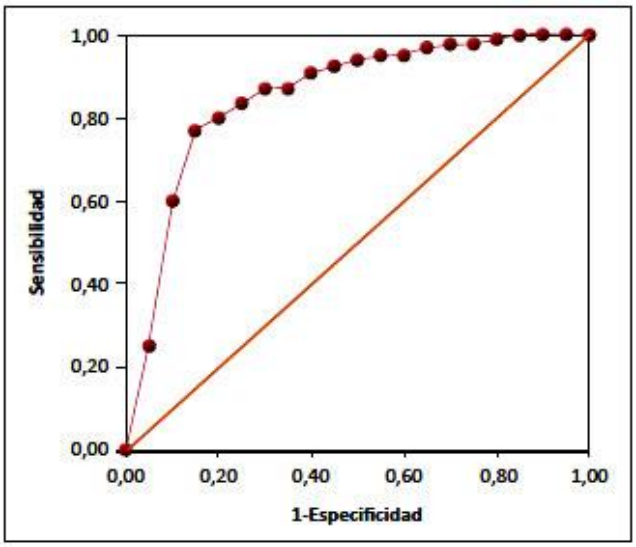

Figura 13: Gráfico de curva ROC de un test hipotético
Fuente: Domínguez & González, 2002

2.10.3 Matriz de confusión

Mediante la matriz de confusión se visualiza el desempeño de un algoritmo de aprendizaje supervisado, en dónde cada columna representa el número de predicciones de cada clase, mientras que cada fila es la representación de las instancias en la clase real, permitiendo que el acierto y error que representa el modelo al momento de pasar los datos por el proceso de aprendizaje sea el correcto (Mena, 2008).

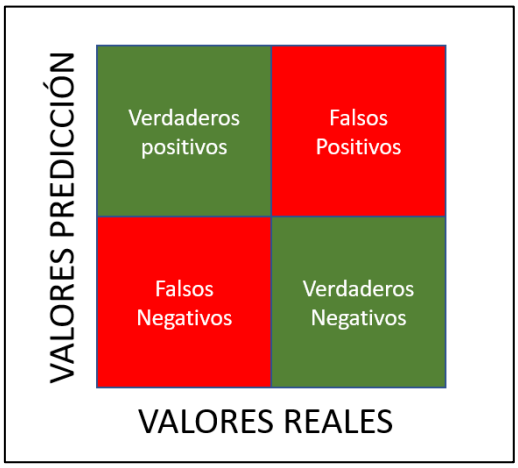

Figura 14: Matriz de confusión binaria
Fuente: Mena, 2008

2.10.4 Kappa

Cohen Kappa propuso esta métrica con la finalidad de introducir el efecto de aleatoriedad obteniendo la diferencia normalizada entre la tasa de Accuracy/presición observado y la tasa de Accuracy/presición que se esperaría por casualidad (Borja, Monleon & Rodellar, 2020).

$$k = \frac{P_o - P_e}{1 - P_e} \quad (5)$$

Donde:

P_o = proporción de accuracy observado
P_e = proporción de accuracy esperado en la hipótesis de independencia entre los observadores, es decir, accuracy al azar

2.11 Macroinvertebrados acuáticos

2.11.1 Definición.

Roldán (1992) describe a los macroinvertebrados como “organismos que no tienen espina dorsal y que son visibles sin usar un microscopio”.

Son aquellos organismos que al menos algún periodo de su ciclo de vida, exclusivamente viven en un ambiente acuático, además pueden ser observados a simple vista ya que tiene un tamaño superior a 0,5 mm de longitud. Entre el 70 y 90% de estos organismos son acuáticos en sus estados inmaduros de huevos y larvas, mientras que los adultos suelen ser terrestres, en la mayoría son insectos y son usados con éxito como bioindicadores de la calidad del agua. (Mosquera, 2015).

a. Macroinvertebrados bentónicos

Son organismos que habitan en el sustrato de lagos, cursos de agua, aguas marinas y estuarios. Pueden construir tubos o redes fijas para vivir dentro o sobre ellos o simplemente viven libres sobre las rocas, residuos orgánicos y otros sustratos durante toda o fragmento de su vida, los macroinvertebrados bentónicos han adoptado gran atención por su importancia como eslabones tróficos intermediarios entre los productores primarios y consumidores dentro de los cuerpos de aguas continentales (Suárez , 2012).

2.11.2 Hábitat.

Los macroinvertebrados pueden vivir y desarrollarse en troncos caídos descompuestos, hojas flotantes y sus restos, en arena o lodo del fondo de los ríos, sobre o debajo de piedras, en sistemas lóticos y lénticos. Su reproducción es en grandes cantidades por lo que se pueden encontrar miles en un metro cuadrado (Roldán, 1996).

2.11.3 Taxonomía.

Los científicos para poder ordenar la clasificación de tanta diversidad de organismos, han dividido y organizado a cada grupo en diferentes “categorías

taxonómicas” destacando que este orden no es al azar, sino que está basado en una serie de relaciones de parentesco evolutivo (Palma, 2013).

- Reino
- Phylum
- Clase
- Orden
- Familia
- Género
- Especie

En honor al biólogo Carlos Linneo (1707 – 1778) la taxonomía es estudiada bajo el sistema taxonómico de Linneo o taxonomía linneana, siendo fundamental para mantener el preciso orden y la individualidad de cada especie en este planeta, clasificando a los seres vivos en distintos niveles jerárquicos, comenzando originalmente por el de reino.

2.11.4 Morfometría.

La morfometría es el estudio cuantitativo de la variación de las formas biológicas, es decir, organismos de diferentes tamaños tendrán diferentes formas, aunque sean de la misma especie, debido al desarrollo y crecimiento natural de los organismos vivos. Se utiliza para la cuantificación de un carácter de significancia evolutiva para la detección de los cambios en la función evolutiva y desarrollo de los organismos. El principal objetivo de la morfometría es probar estadísticamente las hipótesis sobre los factores que afectan la forma (López, 2015).

Cada familia de macroinvertebrados posee características diferentes, para su identificación adecuada se deben realizar medidas de las diferentes estructuras corporales, como la maxila, mandíbula y las estructuras de locomoción, a pesar de sus semejanzas morfométricas algunas son pertenecientes a diferentes especies (Garrido, et al., 2015).

2.11.5 Técnicas de recolección

Existen diferentes métodos para recolectar macroinvertebrados acuáticos los cuales varían conforme al sustrato de piedras, arena, fango y/o vegetación (Ramírez, 2010).

a. Aguas corrientes poco profundas.

La red de mano es el instrumento más eficiente y sencillo para la obtención de una abundante y diversa fauna béntica. El proceso consiste en que una persona sostenga una red por sus dos mangos fijándola al sustrato en dirección contraria al de la corriente, esta puede ser plástica o metálica de aproximadamente un metro cuadrado, con una malla de 0,5 a 1 mm., al mismo tiempo otra persona, remueve con sus pies el fondo aguas arriba. Las larvas son arrastradas por la corriente y atrapadas en la red (Ramírez, 2010) .

Figura 15: Forma de operar red de mano
Fuente: Miguel & Ojeda,1996

b. Aguas lentas o corrientes con vegetación marginal.

La red triangular llamada “D-net” es una de las más usadas para realizar el proceso de “barrido” a lo largo de las orillas con vegetación. Con este método los insectos nadadores o los que viven adheridos a las hojas y tallos de la vegetación sumergida son atrapados con este método (Roldán, 1996).

Figura 16: Forma de usar la red triangular
Fuente: Miguel & Ojeda,1996

c. Aguas corrientes o aguas lentas profundas.

Es pertinente el uso de dragas para este tipo de hábitat, facilita la toma de muestras de sedimentos a distintas profundidades. Para la toma de muestras en fondo blando consiste de dos especies de palas, las cuales se cierran en el fondo a través de una plomada con una cuerda. El sedimento tomado se deposita luego en un cernidor donde se lava, quedando así atrapados los organismos recolectados.

En la figura 17 se observa el uso de la draga Peterson para muestrear fondos duros pedregosos (Roldán, 1996).

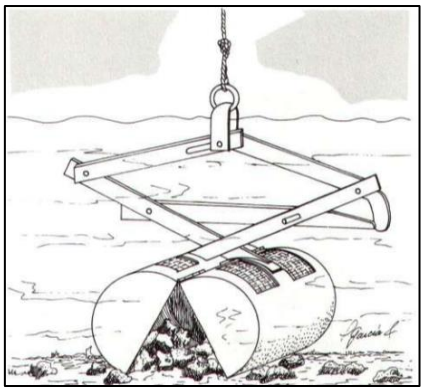

Figura 17: Draga Peterson
Fuente: Miguel & Ojeda,1996

2.11.6 Preservación de las muestras.

La preservación de las muestras se realiza en alcohol etílico al 70%. La cantidad del preservante que se utiliza debe ser la suficiente para que cubra toda la muestra recolectada, además los frascos deben ser rotulados y etiquetados con la localidad, fecha, nombre de la cuenca, tipo de sustrato y nombre del recolector (Samanez, et al., 2014).

2.12 Importancia de los macroinvertebrados

2.12.1 Importancia ecológica.

Los macroinvertebrados poseen un rol importante en todos los procesos ecológicos de los ecosistemas acuáticos. Son un enlace fundamental para mover energía a los diversos niveles tróficos de las cadenas alimenticias acuáticas, creando pequeños fragmentos de materia orgánica, al mismo tiempo las partículas finas del agua son convertidas en partículas fecales densas, lo que provocan su hundimiento para ser proveedoras de alimento para otros organismos acuáticos. Este proceso garantiza la presencia de los nutrientes en cada partícula, y así estos no sean expulsados del ecosistema a causa de las corrientes (Ramírez & Gutiérrez, 2014)

Son consumidores de una gran cantidad de algas y otros microorganismos ligados al plancton en lagos, además son controladores de la productividad primaria en los ecosistemas acuáticos. Son fragmentadores en sistemas compuestos de materiales alóctonos como la hojarasca (Ramírez & Gutiérrez, 2014).

2.12.2 Importancia biológica.

Los macroinvertebrados son indicadores biológicos de las condiciones de calidad de determinados recursos hídricos superficiales por excelencia, cuando hay evidencia de niveles de contaminación química u orgánica estos son utilizados (Suárez, 2015).

Son capaces de otorgar información de perturbaciones, alteración física del cauce y de la ribera. El grupo más utilizado son los macroinvertebrados bentónicos acuáticos debido a las siguientes razones (Roldán, 1996):

- Elevada diversidad.
- Nivel de muestreo fácil.
- Diferentes taxones presentan requerimientos ecológicos diferentes.

- Poseen un ciclo de vida largo que permite integrar los efectos de la contaminación a largo plazo.

2.12.3 Importancia económica.

Los macroinvertebrados son muy poco probables que puedan convertirse en plagas de cultivos, ya que existen muy pocas especies acuáticas que alcanzan poblaciones tan altas debido a que los cultivos que se realizan en el agua también son muy escasos, un ejemplo es el arroz inundado, y aun así la mayoría de los insectos terrestres son los causantes de los ataques en las partes aéreas de las plantas.

En los casos de los cultivos de peces y camarones, los macroinvertebrados pueden ocasionar problemas, por depredación o por competencia, principalmente en los estanques con larvas o estadios jóvenes generando molestias y en muchas ocasiones pérdidas económicas para los seres humanos (Hanson, et al., 2010).

2.13 Índices bióticos de la calidad del agua

Un índice de calidad del agua es un único número que expresa la calidad del recurso hídrico a través de determinadas medidas de los parámetros físicos, químicos y biológicos, siendo su uso cada vez más habitual para identificar los cambios que se generan de manera natural o antrópica. Reducen a una expresión de entendimiento e interpretación fácil una gran cantidad de parámetros, entre técnicos, ingenieros ambientales y todo el público en general (Madsen, Lévêque, Omiste, & Miyagi, 2018).

Las condiciones ambientales del sistema son reflejadas por las comunidades u organismos del ecosistema fluvial del que forman parte. El uso como bioindicadores generalmente demuestra mayor facilidad y menor costo de observación frente a la valoración o análisis directo de los parámetros mediante otros métodos. La principal ventaja que presentan los indicadores biológicos es la capacidad de integrar las variaciones temporales de las condiciones ambientales del medio (Madsen et al., 2018).

2.13.1 Índice ABI.

El ABI es un índice biótico aplicable en la evaluación de la calidad de cuerpos hídricos y en la integridad ecológica de los ecosistemas acuáticos andinos. Su determinación es mediante la asignación a cada familia muestreada los valores numéricos entre 1 y 10. Depende del nivel de tolerancia a la contaminación, perteneciendo el número 1 a las familias más tolerantes y el número 10 a las familias más sensibles. El puntaje ABI total es el resultado de la suma de los puntajes de cada familia encontradas en el sitio determinado. (Encalada, et al., 2011)

Tabla 1: Índice ABI

ABI	Calidad de agua	Color
> 96	Muy bueno	
59 – 96	Bueno	
35 – 58	Regular	
< 35	Malo	

Fuente: Encalada, et al., 2011

2.13.2 Índice biológico BMWP/Col.

Es un método simple y rápido para determinar la calidad del agua usando macroinvertebrados como bioindicadores. Trabaja con datos cualitativos, es decir, presencia o ausencia, y solamente es necesario llegar a nivel de familia. El valor numérico va de 1 a 10 de acuerdo con la tolerancia de los diferentes grupos a la

contaminación orgánica. Reciben el valor de 10 las familias más sensibles, mientras que las más tolerantes el valor de 1. Posteriormente parar calcular el índice se suman los valores asignados por familia, independientemente de la cantidad o género de individuos encontrados (Quintero & Ramirez, 2013).

Tabla 2: Índice biológico BMWP/Col

Clase	Calidad	BMWP/Col	Significado	Color
I	Buena	>150, 101 – 120	Aguas muy limpias a limpias	
II	Aceptable	61 – 100	Aguas ligeramente contaminadas	
III	Dudosa	36 – 60	Aguas moderadamente contaminadas	
IV	Crítica	16 – 35	Aguas muy contaminadas	
V	Muy crítica	<15	Aguas fuertemente contaminadas	

Fuente: Roldán, 2003

Roldán (2003) propone como primera aproximación de evaluación de los ecosistemas acuáticos de montaña, la aplicación del índice BMWP para Colombia bajo el nombre de BMWP/Col, ya que en esta región es donde más trabajan con los macroinvertebrados acuáticos. En ciertos casos existen géneros dentro de una misma familia con valores de identificación distintos, representando tanto aguas limpias como aguas con algún grado de contaminación, por esto este índice únicamente se basa en nivel de familia.

CAPÍTULO III

3. MATERIALES Y MÉTODOS

3.1 Materiales y equipos

- Gasa
- Alcohol
- Cajas Petri
- Computador
- Estereoscopio
- Memoria Flash
- Materiales de oficina
- Pinzas hemostáticas
- Documentos Bibliográficos
- Equipo de Protección Personal (EPP)
- Macroinvertebrados bentónicos acuáticos

3.2 Metodología

La clasificación e identificación de los taxones, obtención de fotografías, extracción y ponderación de las características morfométricas de las especies de macroinvertebrados, diseño y evaluación de los algoritmos se realizaron en el Centro de Innovación, Investigación y Transferencia de Tecnología (CITT) de la Universidad Católica de Cuenca. A continuación, se describe el método de inspección visual de extracción de características morfométricas y el método de clasificación automatizado.

3.2.1. Método de inspección visual de extracción de características morfométricas

- Mediante el uso de referencia bibliográfica, catalográfica y ayuda de la docente bióloga para la identificación de las taxas se clasificaron e identificaron los macroinvertebrados.
- Adquisición de imágenes mediante un estereoscopio para la extracción de características visuales de cada taxa clasificada previamente.
- Selección, extracción y ponderación de características morfométricas de los individuos de las familias clasificadas en una hoja de datos. Anexo 1.
- Transformación de las características extraídas a variables “dummy”. Anexo 2.
- Aplicación de algoritmos de Machine Learning en las tareas de clasificación aplicados a la hoja de datos de características morfométricas definidas
- Evaluación del modelo de Machine Learning aplicado
- Predicción en la clasificación de datos nuevos

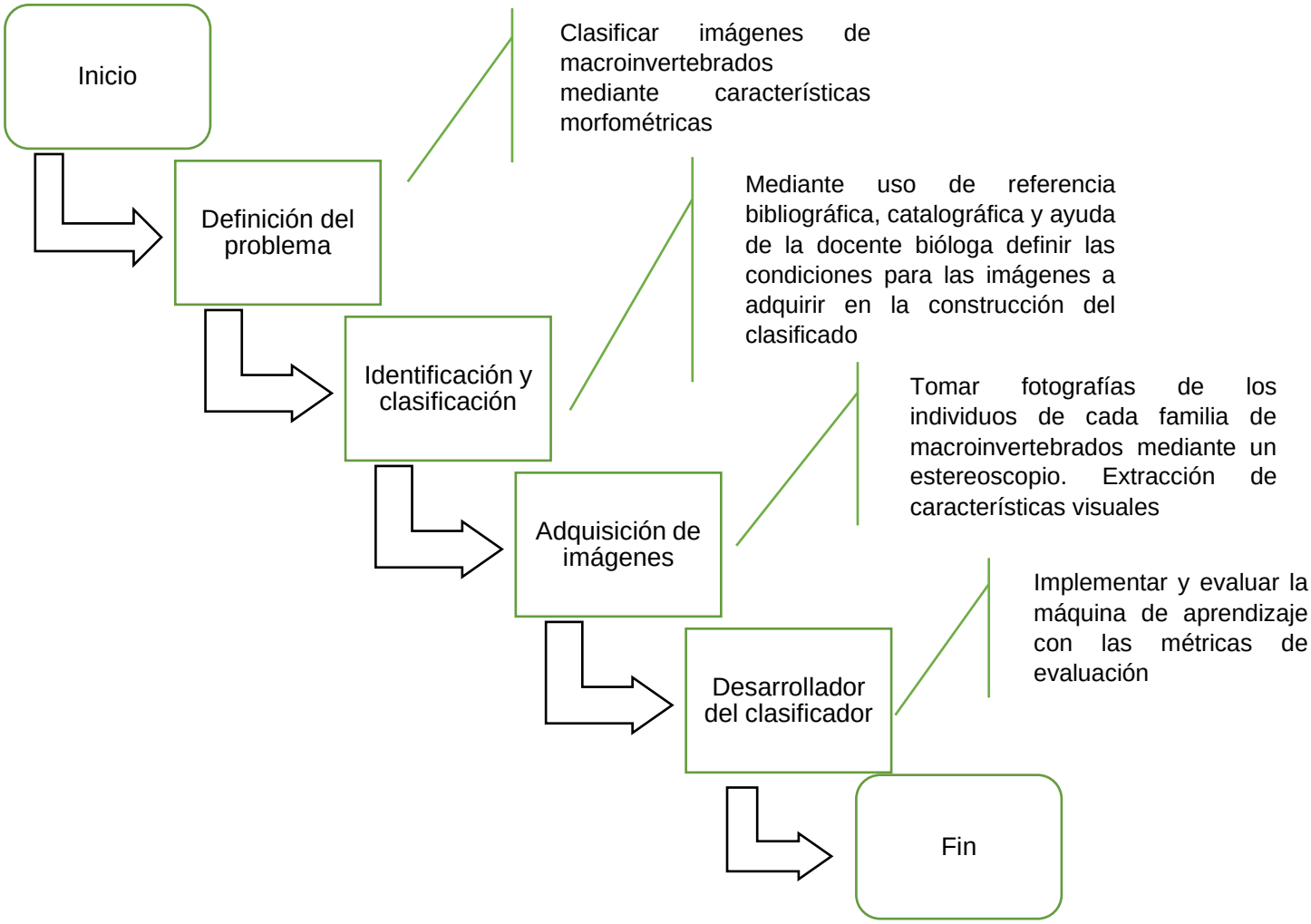

Figura 18: Metodología propuesta para el desarrollo manual del trabajo

3.2.1.1 Identificación de los taxones de los macroinvertebrados

Mediante el protocolo simplificado y guía de evaluación de la calidad ecológica de ríos andinos (2011) y la guía rápida para la identificación de macroinvertebrados de los ríos altoandinos del cantón Cuenca (2018), los macroinvertebrados procedentes de la quebrada del río Tabacay se identificaron a nivel familia y clase.

Figura 19: Macroinvertebrados procedentes de la quebrada del río Tabacay

Se tomó una muestra de 30 individuos correspondientes a cada familia y clase. Para su conservación se colocaron en pequeños frascos de vidrio de aproximadamente 30 ml de capacidad, se usó alcohol al 80%. Cada frasco fue etiquetado con su respectivo nombre.

Figura 20: Preservación de las muestras en el laboratorio

3.2.1.2 Adquisición de imágenes

Con el estereoscopio previamente calibrado con iluminación media y aumento de 40x se obtuvieron las fotografías de dimensión 3072x2048 píxeles en formato JPG de cada individuo en tres vistas, dorsal, ventral y lateral.

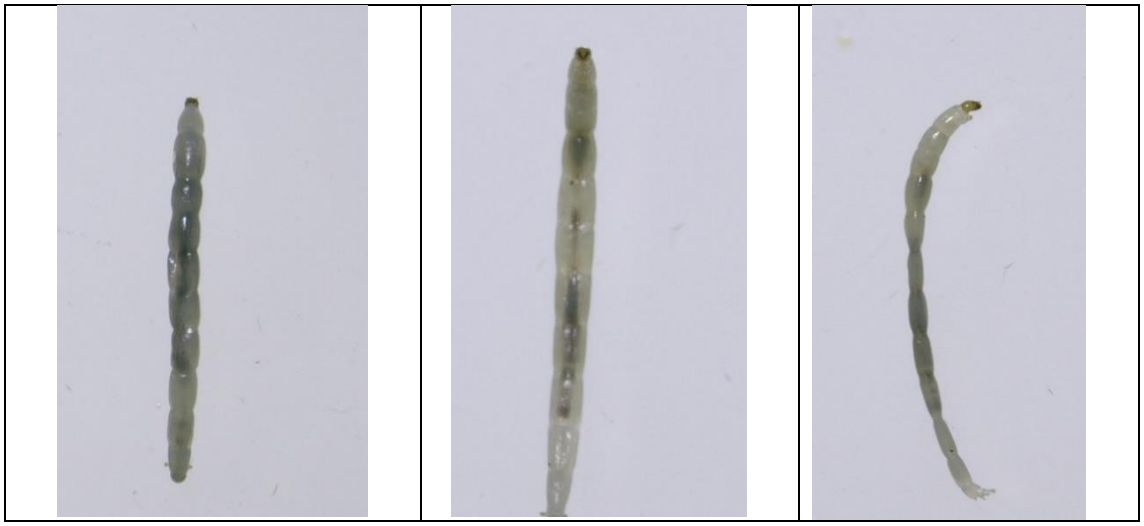

Figura 21: Chironomidae vista dorsal, ventral y lateral

Figura 22: Elmidae vista dorsal, ventral y lateral

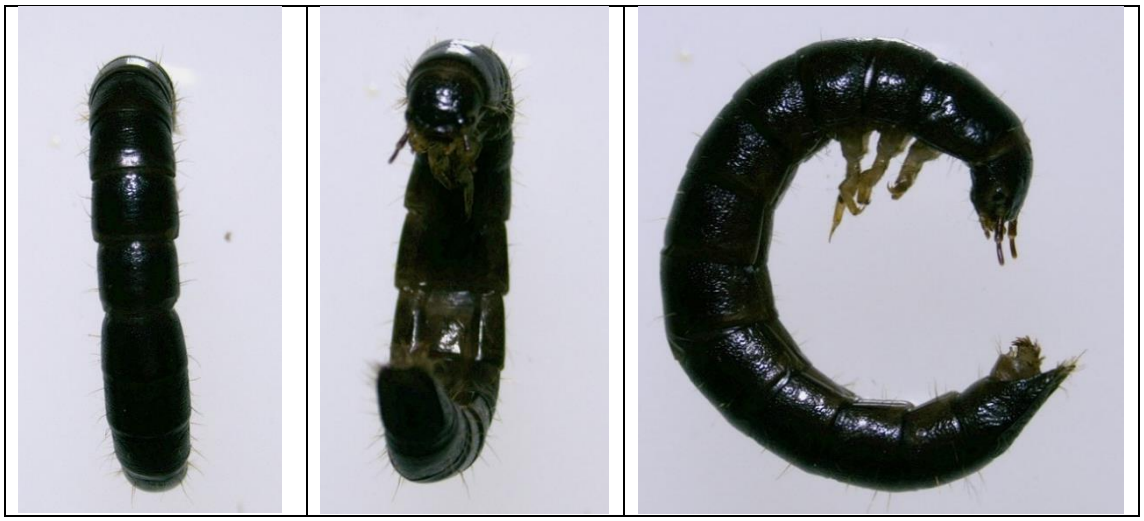

Figura 23: Ptilodactylidae vista dorsal, ventral y lateral

Figura 24: Perdilae vista dorsal, ventral y lateral

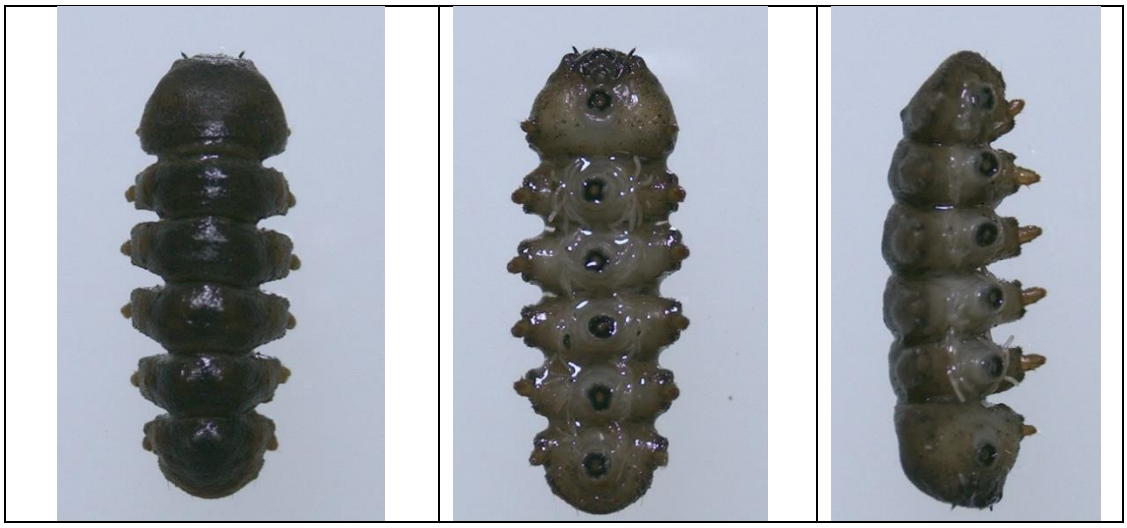

Figura 25: Blephariceridae vista dorsal, ventral y lateral

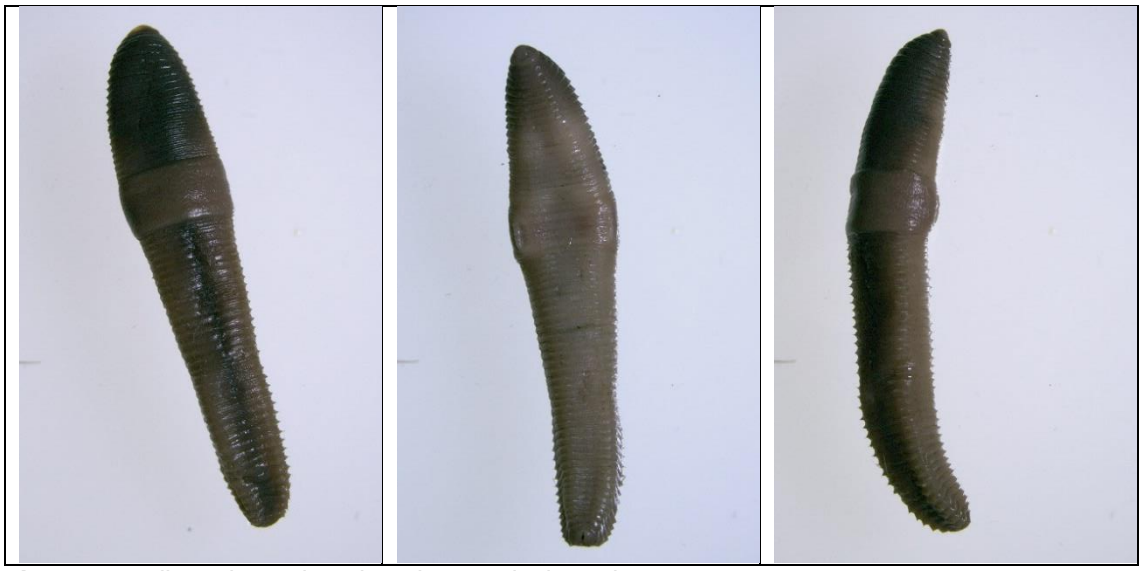

Figura 26: Oligocaheta vista dorsal, ventral y lateral

3.2.1.3 Extracción de diferentes combinaciones comunes de las características morfométricas

Se extrajeron diferentes características morfométricas basados en “Id-Tax. Catálogo y claves de identificación de organismos utilizados en las redes de control del estado ecológico en aguas continentales” (2012) y en “Id-Tax. Catálogo y claves de identificación de organismos invertebrados utilizados como elementos de calidad en las redes de control del estado ecológico” (2012). Las características extraídas son las siguientes:

Tabla 3: Características morfométricas extraídas de los macroinvertebrados

			Características morfométricas de macroinvertebrados						
Clase	**Orden**	**Familia**	**Longitud (mm)**	**Ancho (mm)**	**Forma del cuerpo**	**Segmentos corporales**	**nº patas**	**Cubierto de pelos**	**Antenas**
Insecta	Díptera	Chir	2 - 10	0,4 - 0,8	Alargado y tubular	12	4	Si	No
Oligochaeta	Lumbriculida		1-30	1 - 3	Cilíndrico y segmentado	3	0	No	No
Insecta	Coleóptera	Elmi	1 - 10	0,4 - 1	Ovalado	2	6	No	Si
Insecta	Coleóptera	Ptil	3 - 15	0,7 - 3	Convexo, alargado y ovalado	4 - 11	6	Si	Si
Insecta	Plecóptera	Perl	20 - 50	0,7 - 2	Alargado	3	6	Si	Si
Insecta	Díptera	Blep	7 - 8	0, 6 - 3	Alargado	6	12	No	No

La caracterización morfométrica de las familias y orden se realizó mediante experimentación con el estereoscopio previamente calibrado, con iluminación media y aumento de 40x.

Tabla 4: Codificación de los nombres e indicadores de calidad

Abreviatura	Familia	Calidad (índice ABI)
Chir	Chironomidae	Mala
Olig	Oligocaheta	Mala
Elmi	Elmidae	Media
Ptil	Ptilodactylidae	Media
Perl	Perlidae	Buena
Blep	Blepharoceridae	Buena

3.2.1.4 Transformación de las características a variables "dummy"

Las características extraídas anteriormente se transformaron a variables ficticias, binarias o "dummy" para posteriormente ser empleadas en los modelos de regresión y medir el efecto del factor cualitativo. Las variables ficticias toman la categoría si en unos y el resto la categoría no.

Tabla 5: Transformación de las características a variables “dummy”

Clase	Orden	Labels	AlargTub	CilinSeg	Ovalado	ConvexAlargOval	Alargado	SegCorp	Npatas	CubPelos	Antenas
Insecta	Díptera	Chir	Si	No	No	No	No	12	4	Si	No
Oligocaheta	Lumbriculida	Olig	No	Si	No	No	No	3	0	No	No
Insecta	Coleóptera	Elmi	No	Si	Si	No	No	2	6	No	Si
Insecta	Coleóptera	Ptil	No	Si	No	Si	No	4a11	6	Si	Si
Insecta	Plecóptera	Perl	No	Si	No	No	Si	3	6	Si	Si
Insecta	Díptera	Blep	No	Si	No	No	Si	6	12	No	No

3.2.1.5 Diseño del algoritmo computacional

Mediante el software científico RStudio se diseñó el algoritmo implementando máquinas de aprendizaje supervisado de los datos de imágenes de macroinvertebrados extraídos visualmente y encasillados en las características morfométricas establecidas.

En la Figura 27 se importó la librería necesaria para la lectura de los datos en Excel.

```
1  # Importación de librería para leer datos de excel
2  library(readxl)
3  # Se importan los datos
4  data <- data.frame(read_excel("_MEDIDAS.xlsx", sheet = "CARAC3"))
5  dat<- data[,4:17]
6  # Se filtran los datos desde familia
7  str(dat)
```

```
'data.frame':       504 obs. of  14 variables:
 $ Labels        : chr  "Chironomidae" "Chironomidae" "Chironomidae"
"Chironomidae" ...
 $ AlargTub      : chr  "Si" "Si" "Si" "Si" ...
 $ CilinSeg      : chr  "No" "No" "No" "No" ...
 $ Ovalado       : chr  "No" "No" "No" "No" ...
 $ ConvexAlargOval: chr  "No" "No" "No" "No" ...
 $ Alargado      : chr  "No" "No" "No" "No" ...
 $ SegCorp       : chr  "12" "12" "12" "12" ...
 $ Npatas        : num  4 4 4 4 4 4 4 4 4 4 ...
 $ CubPelos      : chr  "Si" "Si" "Si" "Si" ...
 $ Antenas       : chr  "No" "No" "No" "No" ...
 $ Alas          : chr  "No" "No" "No" "No" ...
 $ Longitud      : num  7.29 6.31 6.42 7.48 7.21 ...
 $ Ancho         : num  0.677 0.666 0.677 0.555 0.521 ...
 $ L.A           : num  0.0928 0.1054 0.1053 0.0742 0.0723 ...
```

Figura 27: Importación de librerías

Se convirtió a factores las variables de la 1 a la 14, solamente las características de medición son del tipo numéricos.

```
8  # Conversión a factores
9  for(i in 1:11){dat[,i]<-factor(dat[,i])}
10 str(dat)
```

```
'data.frame': 504 obs. of  13 variables:
 $ Labels      : Factor w/ 6 levels "Blepharoceridae",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ AlargTub  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
 $ CilinSeg    : Factor w/ 2 levels "No","Si": 1 1 1 1 1 1 1 1 1 1 ...
 $ Ovalado   : Factor w/ 2 levels "No","Si": 1 1 1 1 1 1 1 1 1 1 ...
 $ ConvexAlargOval: Factor w/ 2 levels "No","Si": 1 1 1 1 1 1 1 1 1 1 ...
 $ Alargado  : Factor w/ 2 levels "No","Si": 1 1 1 1 1 1 1 1 1 1 ...
 $ SegCorp   : Factor w/ 5 levels "12","2","3","4a11",..: 1 1 1 1 1 1 1 1 1 1...
 $ Npatas      : Factor w/ 4 levels "0","4","6","12": 2 2 2 2 2 2 2 2 2 2 ...
 $ CubPelos  : Factor w/ 2 levels "No","Si": 2 2 2 2 2 2 2 2 2 2 ...
 $ Antenas     : Factor w/ 2 levels "No","Si": 1 1 1 1 1 1 1 1 1 1 ...
 $ Longitud    : Factor w/ 406 levels "1.0786","1.1593",..: 256 223 227 270 253
260 248 239 248 267 ...
 $ Ancho   : num  0.677 0.666 0.677 0.555 0.521 ...
 $ L.A     : num  0.0928 0.1054 0.1053 0.0742 0.0723 ...
```

Figura 28: Conversión de las variables a factores

La salida o variable predictora es el tipo de familia en la variable 1, los 13 restantes son características regresores.

En la Figura 29 se realizó un resumen descriptivo de los datos. Se elimina la categoría alas, ya que en esta base de datos no existen familias con alas.

```
11 # Resumen
12 summary(dat)
```

```
                Labels    AlargTub  CilinSeg  Ovalado   ConvexAlargOval Alargado  SegCorp
Blepharoceridae :90   No:414    No: 90    No:414    No:414          No:324    12  : 90
Chironomidae    :90   Si: 90    Si:414    Si: 90    Si: 90          Si:180    2   : 90
Elmidae         :90                                   3   :144
Oligocaheta     :54                                   4a11: 90
Perlidae        :90                                   6   : 90
Ptilodactylidae :90

 Npatas   CubPelos  Antenas     Longitud        Ancho             L.A
 0 : 54   No:234    No:234      1.8638 :  4   Min.   :0.4105   Min.   :0.06811
 4 : 90   Si:270    Si:270      4.9923 :  4   1st Qu.:0.7433   1st Qu.:0.14271
 6 :270                         7.1445 :  4   Median :1.2536   Median :0.24021
 12: 90                         1.6364 :  3   Mean   :1.2450   Mean   :0.25285
                                2.0635 :  3   3rd Qu.:1.6410   3rd Qu.:0.35159
                                2.2077 :  3   Max.   :3.1968   Max.   :1.28574
                               (Other):483
```

Figura 29: Conversión de las variables a factores

Se determinó los atributos o variables independientes altamente correlacionadas.

```
13 #Cargar la librería Caret
14  library(caret)
15 #Determinar atributos
16 (cormatrix<-cor(dat[,11:13]))
17 higlyCorrelated<-findCorrelation(cormatrix,cutoff = 0.5)
18 higlyCorrelatedNames<-findCorrelation(cormatrix,cutoff = 0.4,names = TRUE)
19 print(higlyCorrelated)
20 print(higlyCorrelatedNames)
```

```
            Longitud    Ancho       A/L
Longitud    1.0000000   0.5161430   -0.6479027
Ancho       0.5161430   1.0000000   0.1971063
A.L         -0.6479027  0.1971063   1.0000000
[1] 1
[1] "Longitud"
```

Figura 30: Variables independientes altamente correlacionadas

Para los modelos predictivos se utilizó la librería Caret y se dividió el conjunto total de datos en dos partes:

1. Conjunto de datos de entrenamiento "data_train" con porcentaje de 75% (Figura 31, línea 30).
2. Conjunto de datos de prueba "data_test" con el porcentaje complementario, 25% (Figura 31, línea 32).

```
21 # Se carga la librería
22 library(caret)
23 # Se establece una semilla
24 set.seed(2021)
25 # Se establece el porcentaje para los datos de entrenamiento
26 train_perc = 0.75
27 # Se crea el índice para realizar las particiones
28 train_index = createDataPartition(dat$Labels, p = train_perc, list = FALSE)
29 # Con el índice se extraen los datos de entrenamiento
30 data_train = dat[train_index,]
31 # Se extraen los datos de prueba "test" con los índices complementarios
32 data_test = dat[-train_index,]
```

```
data_train    381 obs. of 10 variables
data_test     123 obs. of 17 variables
```

Figura 31: Conjunto de datos de entrenamiento y prueba

Como modelo predictivo se realizó el Random Forest, ya que es el más usado en *Machine Learning* y el que brinda mejores resultados en la mayoría de los casos, luego se creó la matriz de confusión para la evaluación del mismo.

```
library(caret)
library(randomForest)
set.seed(2021)
validationIndex<-createDataPartition(dat$Labels,p=0.75,list = FALSE)
validation<-dat[-validationIndex,]
training<-dat[validationIndex,]
fit.RF<-randomForest(Labels~.,training, mtyr=2,ntree=200)
saveRDS(fit.RF,"fit.RF")

pred<-predict(superModel,validation[,2:13])
confusionMatrix(pred,validation$Labels)
```

Figura 32: Modelo Random Forest

Confusion Matrix and Statistics

Prediction \ Reference	Blepharoceridae	Chironomidae	Elmidae	Oligocaheta	Perlidae	Ptilodactylidae
Blepharoceridae	22	0	0	0	0	0
Chironomidae	0	22	0	0	0	0
Elmidae	0	0	22	0	0	0
Oligocaheta	0	0	0	13	0	0
Perlidae	0	0	0	0	22	0
Ptilodactylidae	0	0	0	0	0	22

Overall Statistics

Accuracy	1
95% CI	(0.9707, 1)
No Information Rate	0.3629
P-Value [Acc > NIR]	< 2.2e-16
Kappa	1
Mcnemar's Test P-Value	NA

Statistics by Class:

Class:	Blepharoceridae	Chironomidae	Elmidae	Oligocaheta	Perlidae	Ptilodactylidae
Sensitivity	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Specificity	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Pos Pred Value	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Neg Pred Value	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Prevalence	0.1774	0.1774	0.1774	0.1048	0.1774	0.1774
Detection Rate	0.1774	0.1774	0.1774	0.1048	0.1774	0.1774
Detection Prevalence	0.1774	0.1774	0.1774	0.1048	0.1774	0.1774
Balanced Accuracy	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Figura 33: Matriz de confusión del modelo Random Forest

Finalmente, en la Figura 34, se cargó el modelo guardado y se predijeron valores manualmente.

```
#Se carga el modelo
superModel<-readRDS("fit.RF")
(ingreso<-validation[1,2:13])
# Ingresar manualmente las características del macroinvertebrado
ingreso$AlargTub = factor("No",levels = c("Si","No"))
ingreso$CilinSeg = factor("Si",levels = c("Si","No"))
ingreso$Ovalado = factor("No",levels = c("Si","No"))
ingreso$ConvexAlargOval= factor("Si",levels = c("Si","No"))
ingreso$Alargado = factor("No",levels = c("Si","No"))
ingreso$SegCorp = factor("4a11",levels = c("12", "2", "3", "4a11", "6"))
ingreso$Npatas = factor("4",levels = c("0","4","6","12"))
ingreso$CubPelos = factor("Si",levels = c("Si","No"))
ingreso$Antenas = factor("No",levels = c("Si","No"))
ingreso$Longitud = 12
ingreso$Ancho = 1
ingreso$L.A=ingreso$Ancho/ingreso$Longitud
ingreso<-data.frame(ingreso)
(pred1<-pred<-predict(superModel,ingreso))
```

```
Ptilodactylidae
Levels: Blepharoceridae Chironomidae Elmidae Oligocaheta Perlidae
Ptilodactylidae
```

Figura 34: Predicción manual de valores

3.2.2 Método automatizado de extracción de características morfométricas

En la actualidad existen varias metodologías para realizar proyectos de minería de datos o modelado, normalmente estas indican que hacer en cada etapa, pero no cómo, es por esto que cada organización adopta una metodología propia. Basado en la metodología que aplica Heras (2017) en “Clasificador de imágenes de frutas basado en inteligencia artificial” se propone la siguiente metodología:

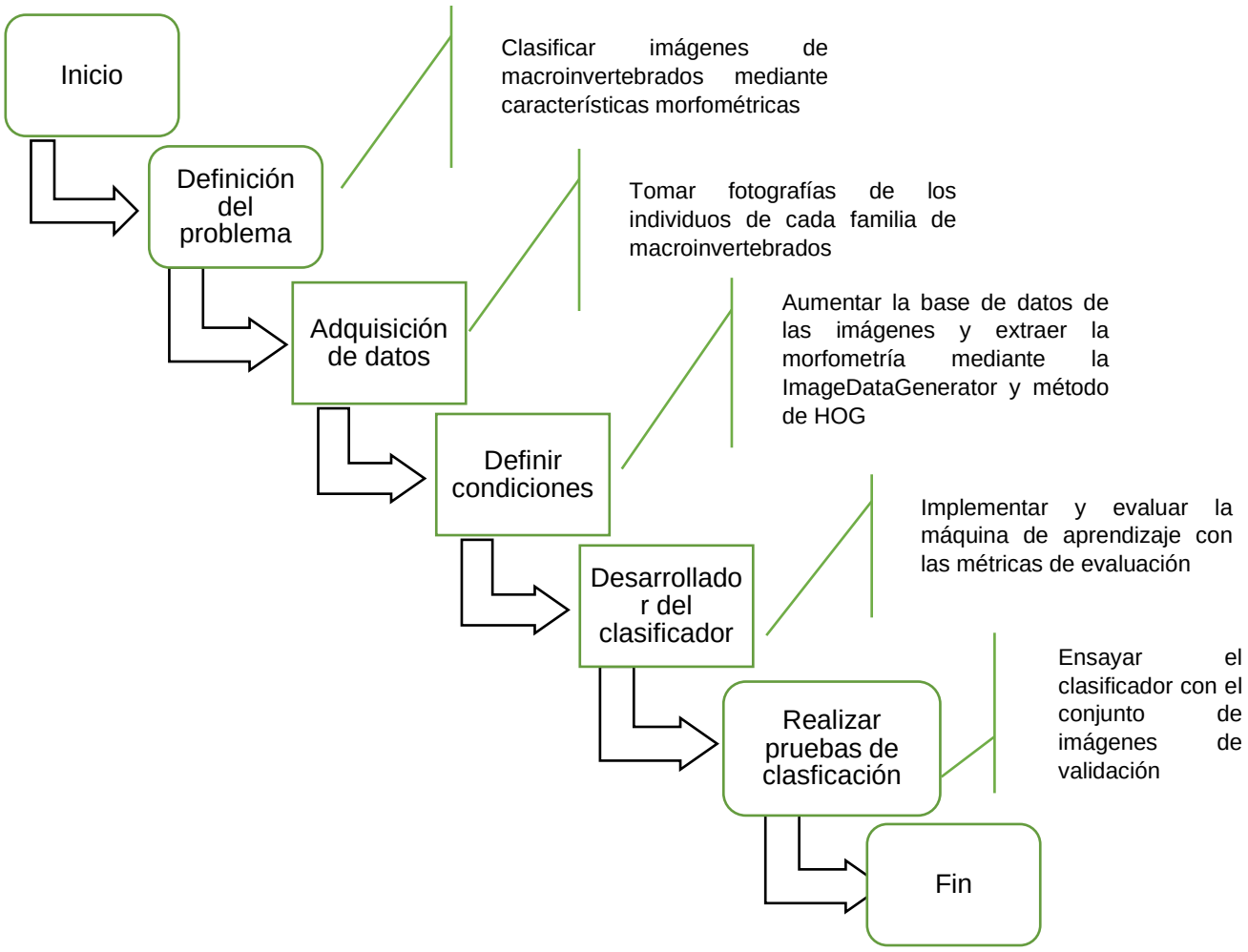

Figura 35: Metodología propuesta para el desarrollo del trabajo

3.2.2.1 Generación de nuevas imágenes a partir de las existentes

Es importante destacar que el rendimiento de las máquinas de aprendizaje muchas veces mejora con la cantidad de imágenes disponibles. Por tanto, para este proyecto se generaron de forma artificial más imágenes a partir de las disponibles mediante transformaciones geométricas simples y aleatorias procedentes de las imágenes del conjunto de datos existente.

El incremento de imágenes se realizó en el lenguaje de programación **Pyhton** con la biblioteca de aprendizaje profundo de Keras a través de la clase *ImageDataGenerator.*

Figura 36: Método más común de incremento de imágenes con Keras

3.2.2.2 Pre procesamiento de imágenes

Para solucionar la problemática de tener datos numéricos muy grandes se redujo el tamaño de las imágenes y este tamaño se estandarizó para todas las imágenes para evitar la llamada “maldición de la dimensión en *Machine Learning*”, previo al diseño de los algoritmos mediante *Python*. Por ejemplo, en la familia Chironomidae cada imagen tiene un tamaño de 6.291.456 píxeles (3072x2048), en total 566.231.040 píxeles de las 90 imágenes.

En la Figura 37 se importaron las librerías necesarias para el pre procesamiento de las imágenes de la línea 2 a la línea 9, la línea 11 imprime la versión de *OpenCV.*

```
1  # Importación de librerías
2  import math
3  import cv2
4  import numpy as np
5  import pandas as pd
6  import matplotlib.pyplot as plt
7  from scipy import ndimage
8  import os
9  from glob import glob
10 from tqdm import tqdm as tqdm
11 from time import sleep
12 from random import uniform
13 #Imprime la versión de OpenCV
14 print(cv2.__version__)
```

Figura 37: Importación de librerías para el preprocesamiento de las imágenes

Se creó un bucle “*for*” en tqdm para recortar las imágenes en formato de escala de grises. En la línea 15 de la Figura 38 se integró una barra gráfica de progreso de nombre “procesando imágenes” con la librería tqdm. En la línea 21 se determinó el valor numérico para la binarización mediante el algoritmo de OTSU, a partir del límite determinado, en la línea 23 se generó las máscaras de las imágenes resaltando las regiones de interés de cada una.

Para suavizar las imágenes se aplicó el filtro Gaussiano con kernel de 3x3, línea 31. En las líneas 34 a la 38, se detectaron los objetos conectados con cada imagen y se crearon máscaras. También se detectó el contorno de la imagen y se extrajeron las coordenadas del rectángulo que encierra a cada una (x,y,w,h). En la línea 51 se recortaron las imágenes con las coordenadas determinadas anteriormente.

```
for i in tqdm(range(len(archivos)),desc="Procesando imagenes"):
# lee la imagen de la dirección i, en escala de grises
img = cv2.imread(archivos[i],0)

# determina el valor numérico para la binarización mediante el algoritmo
de OTSU
tr,_=cv2.threshold(img,0,255,cv2.THRESH_OTSU)
# Crea la máscara a partir del límite de valor determinado por Otsu (tr)
mascara = np.uint8((img<tr)*255)

# Segmentación de la imagen
segmenta = np.uint8(np.zeros(img.shape)) # crea un fondo vacío
mascara1 = mascara/255 # se normalizan los valores de la imagen
segmenta[:,:] = mascara1*img[:,:]+255*(mascara1 == 0)

# Filtro Gaussiano con kernel (3x3)
segmenta = cv2.GaussianBlur(segmenta,(3,3),0)

# Detecta los objetos cercanos conectados en la imagen
output = cv2.connectedComponentsWithStats(mascara,4,cv2.CV_32S)
labels = output[1]
stats = output[2]
mascara2=(np.argmax(stats[1:,4])+1 == labels).astype(int)
mascaraN=np.uint8(mascara2*255)

# Se detectan contornos en la imagen
contours,_=cv2.findContours(mascaraN,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
cnt = contours[0]
# Se extraen coordenadas del rectángulo qu encierra la imagen
x,y,w,h =cv2.boundingRect(cnt)
# Recorta la imagen con el rectángulo detectado que contiene la imagen
crop = segmenta[y:y+h,x:x+w]

# Rotación de imágenes en sentido vertical
rows, cols = crop.shape[:2]
      if cols > rows:
          crop = cv2.rotate(crop, cv2.ROTATE_90_COUNTERCLOCKWISE)


# Escalado de la imagen (downscaling)
scale_percent = 30 # percent of original size
width  = int((crop.shape[1]) * scale_percent / 100)
height = int((crop.shape[0]) * scale_percent / 100)
dim = (width, height)
crop = cv2.resize(crop, dim, interpolation = cv2.INTER_AREA)
```

Figura 38: Bucle para recortar las imágenes, rotación y reducción de la escala

En la Figura 38 las nuevas imágenes ya recortadas se rotaron en sentido vertical y se redujo la escala utilizando el concepto del promediado local que representa el 30% del tamaño original.

Se determinaron las dimensiones máximas y mínimas del alto y ancho de las imágenes. Para ello se cargó el directorio de las imágenes de origen y se creó un bucle “for” en tqdm con la descripción “Procesando alto y ancho max y min” y se estandarizaron todas las imágenes a las dimensiones de la más pequeña.

```
# Extracción de direcciones de las imágenes
archivos=ls4(d,"*.jpg")
num=len(archivos)
list_altos = np.zeros(num)
list_anchos = np.zeros(num)

for i in tqdm(range(num), desc= 'Procesando alto y ancho max y min'):
    list_altos[i] = cv2.imread(archivos[i]).shape[0]
    list_anchos[i] = cv2.imread(archivos[i]).shape[1]

maximo_alto = max(list_altos)
maximo_ancho = max(list_anchos)
min_alto = np.int(min(list_altos))
min_ancho = np.int(min(list_anchos))

print("Alto mínimo:",min_alto,"pixeles")
print("Ancho mínimo:",min_ancho,"pixeles")
print("\n")

# Estandarización de las dimensiones al tamaño más pequeño
for i in tqdm(range(num), desc= 'Redimensionando imágenes'):
    imagen = cv2.imread(archivos[i],0)
    redim = resizeAndPad(imagen, (min_alto,min_ancho))
    dirsave2 = './imf2'
    os.chdir(dirsave2)
    nombre = archivos[i].split('imr2\\')[1]
    cv2.imwrite(nombre,redim)
    os.chdir(dir) # Restaura el directorio original de trabajo
print("Imágenes redimensionadas a:", min_alto,"x",min_ancho)
```

Figura 39: Determinación las dimensiones máximas y mínimas del alto y ancho y estandarización de las imágenes a las dimensiones del tamaño más pequeño

Por último, las imágenes estandarizadas se transformaron a una matriz de datos numéricos, el resultado es un archivo CSV (valores separados por comas). Figura 40.

```
# Directorio de las imágenes origen
dirf="C:\\Users\Zona Informatica\Desktop\projmacro\imf2/"
# Extracción de direcciones de las imágenes
archivos=ls4(dirf,"*.jpg")
num=len(archivos)
nombres1=ls4(dirf,"*.jpg")
nombres2=ls4(dirf,"*.jpg")
imag = cv2.imread(archivos[1],0)
hn, wn = imag.shape[:2]
m = num
n = hn*wn
dat = np.zeros((m,n))

for i in tqdm(range(num), desc='Compilando archivo de imágenes'):
    imag = cv2.imread(archivos[i],0).flatten()
    for j in range(n):
        dat[i,j] = imag[j]
nombres1[i] = archivos[i].split('imf2\\')[1]
nombres1[i] = nombres1[i].split('.jpg')[0]
nombres2[i] = nombres1[i].split('_')[0]

print("Imagenes redimensionadas a:", min_alto,"x",min_ancho)

nombres1 = pd.DataFrame(nombres1)
nombres2 = pd.DataFrame(nombres2)
df = pd.DataFrame(dat)
nombres1.to_csv('nombres1_macro.csv', sep=',')
nombres2.to_csv('nombres2_macro.csv', sep=',')
df.to_csv('dat_macro.csv', sep=',')

# Lectura del CSV
dfr = pd.read_csv('dat_macro.csv')
```

Figura 40: Transformación de las imágenes en una matriz de datos numéricos

3.2.2.3 Método de HOG

La extracción de características, se realizó en el software RStudio. Se cargó la librería OpenImageR y directamente se calculó el método con el comando HOG_apply con la matriz de datos numéricos obtenida en el preprocesamiento.

```
d.train1 <- data_train[,-1]
d.train <- as.matrix(d.train1)

library(OpenImageR)

train.hog <- HOG_apply(d.train,  cells = 6, orientations = 9,
                  rows = 84, columns = 39, threads = 6)
dim(d.train1)
dim(train.hog)

d.test1 <- data_test[,-1]
d.test <- as.matrix(d.test1)

test.hog <- HOG_apply(d.test, cells = 6, orientations = 9,
                  rows = 84, columns = 39, threads = 6)

dim(d.test1)
dim(test.hog)
test.hog<-as.data.frame(test.hog)
test.hog$labels<-data_test[,1]
```

```
time to complete: 0.1764591 secs
[1]    381 3276
[1]    381 324
time to complete: 0.05196691 secs
[1]   123 3276
[1]   123 324
```

Figura 41: Método de HOG

3.2.2.4 Diseño de los algoritmos computacionales con 504 observaciones

En RStudio se cargó la matriz de datos numéricos con la función read_csv de la librería readr y se creó el dataframe llamado “dat” el cual tiene 504 observaciones de 3277 variables.

```
library(readr)
dm <- read_csv("dat_macro.csv")
dm<-dm[,-1]
nm <- read_csv("nombres2_macro.csv")
nm<-nm[,-1]
names(nm)<-"labels"
dat<-cbind(nm,dm)
dat$labels<-as.factor(dat$labels)

row = 84
col = 39
l=row*col
```

Figura 42: Matriz de datos numéricos cargada en RStudio

En la Figura 43 se verificó que el set de datos esté balanceado en sus categorías y se dibujó una imagen de muestra.

```
Blep Chir Elmi Olig Perl Ptil
  90   90   90   54   90   90
```

Figura 43: Balance del set de datos

```
sample_4<-matrix(as.numeric(dat[4,-1]),nrow = row, ncol =
col, byrow = TRUE)
image(sample_4,col = grey.colors(255))
```

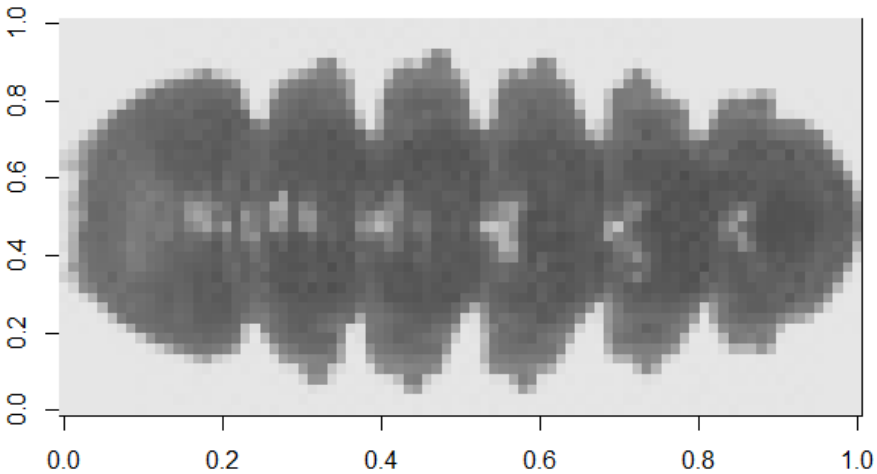

Figura 44: Imagen de muestra

Los modelos predictivos se realizaron con el software RStudio. Utilizando la librería Caret se dividió el conjunto total de datos en dos partes:

1. Conjunto de datos de entrenamiento “data_train” con porcentaje de 75%.
2. Conjunto de datos de prueba “data_test” con el porcentaje complementario, 25%.

```
# Se carga la librería
library(caret)
# Se establece una semilla
set.seed(2020)
# Se establece el porcentaje para los datos de entrenamiento
train_perc = 0.75
# Se crea el índice para realizar las particiones
train_index = createDataPartition(dat$Labels, p = train_perc, list =
FALSE)
# Se extraen los datos de entrenamiento
data_train = dat[train_index,]
# Se extraen los datos de prueba "test" con los índices complementarios
data_test = dat[-train_index,]
```

```
data_train    381 obs. of 3277variables
data_test     123 obs. of 3277 variables
```

Figura 45: División del conjunto de datos en entrenamiento y prueba

- ***Modelo de regresión logística***

Para crear el modelo de regresión logística se importó la librería nnet y se aplicó la función multinom() al set de datos de entrenamiento “data_train”. Para evaluar la calidad del modelo se creó la matriz de confusión con el conjunto de datos de prueba “data_test”.

```
# Se carga la librería nnet para usar el modelo de regresión logística
library(nnet)
mlr<-multinom(labels ~ ., data = data_train,
              MaxNWts=20000, decay=5e-2,maxit=10)
```

```
prediction_lr
           Blep Chir Elmi Olig Perl Ptil
  Blep       20    0    0    2    0    0
  Chir        0   22    0    0    0    0
  Elmi        0    0   19    0    2    1
  Olig        0    5    0    5    1    2
  Perl        0    0    3    1   15    3
  Ptil        6    2    0    4    1    9
[1] 0.7317073
```

Figura 46: Modelo de regresión logística

- ***Modelo de clasificación Deep learning con h2o***

Se cargó la librería h2o y con el comando h2o.deeplearning se creó la red neuronal profunda y se creó la matriz de confusión del modelo de Deep learning con el conjunto de datos de prueba “data_test”.

```
# Inicializar h2O
set.seed(1980)
local.h2o <- h2o.init(ip = 'localhost', port = 54321,
                      startH2O = T, nthreads = 1)
# Subir los datasets
data_train.h2o <- as.h2o(data_train)
data_test.h2o <- as.h2o(data_test)
# Entrenar el modelo:
mod.macro.H2O <- h2o.deeplearning(x = 2:3277, y = 1,
                                  data_train.h2o, activation = 'Tanh',
                                  hidden = rep(160,5),
                                  epochs = 20)
pred.H2O <- h2o.predict(mod.macro.H2O, newdata = data_test.h2o[,-1])
pred.H2O.df <- as.data.frame(pred.H2O)
confusionMatrix(pred.H2O.df$predict, data_test[,1])

#h2o.confusionMatrix(pred.H2O, data_test[,1])

# Accuracy/Precisión:0.8049

h2o.shutdown(prompt = F)

# Se guarda el modelo
h2o.saveModel(object = mod.macro.H2O, path = getwd(), force = TRUE)
# Para cargar el modelo
modelo <- h2o.loadModel(path = "./nombre del modelo")
```

Figura 47: Inicialización de h2o

```
                 Confusion Matrix and Statistics

          Reference
Prediction Blep Chir Elmi Olig Perl Ptil
      Blep   21    0    0    0    0    6
      Chir    0   21    0    1    0    0
      Elmi    0    0   22    0    7    0
      Olig    0    0    0    8    1    3
      Perl    0    0    0    0   14    0
      Ptil    1    1    0    4    0   13

Overall Statistics

               Accuracy : 0.8049
                 95% CI : (0.7237, 0.8708)
    No Information Rate : 0.1789
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7644

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Blep Class: Chir Class: Elmi Class: Olig
Sensitivity               0.9545      0.9545      1.0000     0.61538
Specificity               0.9406      0.9901      0.9307     0.96364
Pos Pred Value            0.7778      0.9545      0.7586     0.66667
Neg Pred Value            0.9896      0.9901      1.0000     0.95495
Prevalence                0.1789      0.1789      0.1789     0.10569
Detection Rate            0.1707      0.1707      0.1789     0.06504
Detection Prevalence      0.2195      0.1789      0.2358     0.09756
Balanced Accuracy         0.9476      0.9723      0.9653     0.78951
                     Class: Perl Class: Ptil
Sensitivity               0.6364      0.5909
Specificity               1.0000      0.9406
Pos Pred Value            1.0000      0.6842
Neg Pred Value            0.9266      0.9135
Prevalence                0.1789      0.1789
Detection Rate            0.1138      0.1057
Detection Prevalence      0.1138      0.1545
Balanced Accuracy         0.8182      0.7658
```

Figura 48: Matriz de confusión de modelo de Deep learning con h2o

- ***Modelo de clasificación usando método de HOG y KNN***

Anteriormente obtenido el histograma de gradientes orientados como predictor se creó el modelo KNN aplicando los comandos de la librería KernelKnn.

```
library(KernelKnn)

etiquetas<-as.numeric(data_train[,1])

modelo.HOG <- KernelKnnCV(train.hog,
                          etiquetas, k = 20,
                          folds = 4,method = 'braycurtis',
                          weights_function = 'biweight_tricube_MULT',
                          regression = F,
                          threads = 6,
                          Levels = sort(unique(etiquetas)))

precision <- function(etiquetas, predicciones) {
 CM <- table(etiquetas, max.col(predicciones, ties.method =
'random'))
 precision <- sum(diag(CM))/sum(CM)}
precision.Modelo <- unlist(lapply(1:length(modelo.HOG$preds),
                                  function(x)
precision(etiquetas[modelo.HOG$folds[[x]]],
modelo.HOG$preds[[x]])))

mean(precision.Modelo)

# Accuracy / Precisión: 0.866

saveRDS(modelo.HOG, "mod.KNN.rds")
```

```
[1] 0.8854167 0.8842105 0.8526316 0.8421053
[1] 0.866091
```

Figura 49: Modelo KNN

- ***Modelo de clasificación usando el método de HOG y regresión logística***

Se aplicó el mismo comando que se utilizó en el modelo anterior de regresión logística, Figura 46, pero se trabajó con los dataframes de train y test obtenidos con el método de HOG. Además, para la evaluación se creó la matriz de confusión con el conjunto de datos de prueba de HOG "test.HOG".

```
# Se carga la librería nnet para usar el modelo de regresión logística
library(nnet)
train.hog<-as.data.frame(train.hog)
train.hog$labels<-data_train$labels
mlr.hog<-multinom(labels ~ ., data = train.hog,
MaxNWts=20000,decay=5e-2,maxit=10)

saveRDS(mlr.hog, "mod.RLog.rds")

prediction_lr.hog<-predict(mlr.hog,test.hog,type="class")
(cmlr<-table(test.hog$labels,prediction_lr.hog))
(accuracy_lr = mean(prediction_lr.hog == test.hog$labels))

# Accuracy / Precisión: 0.7560
```

```
# weights:  1956 (1625 variable)
initial  value 682.660358
iter  10 value 263.955704
final  value 263.955704
stopped after 10 iterations
```

Figura 50: Modelo de clasificación usando el método de HOG y regresión logística

```
prediction_lr.hog<-predict(mlr.hog,test.hog,type="class")
(cmlr<-table(test.hog$labels,prediction_lr.hog))
(accuracy_lr = mean(prediction_lr.hog == test.hog$labels))

# Accuracy / Precisión: 0.7560
```

```
prediction_lr.hog
       Blep Chir Elmi Olig Perl Ptil
  Blep   21    0    0    1    0    0
  Chir    0   21    0    1    0    0
  Elmi    0    0   21    0    0    1
  Olig    5    0    0    7    0    1
  Perl    0    0   10    2    8    2
  Ptil    6    0    1    0    0   15
[1] 0.7560976
```

Figura 51: Matriz de confusión del modelo de regresión logística usando el método de HOG

- ***Modelo de clasificación usando el método de HOG y Naive Bayes***

Con los dataframes de train y test obtenidos con el método de HOG se creó el modelo Naive Bayes con la librería caret.

```
# Se carga las librerías necesarias
library(caret)
library(klaR)

# Naive Bayes con Caret

modelo.NaiveBayes.Caret <- train(train.hog[,-325],
                                 train.hog$labels,
                                 method = 'nb',
                                 trControl = trainControl
                                (method = 'cv', number = 10))

prediccion.NaiveBayes.Caret <- predict(modelo.NaiveBayes.Caret,
                                       newdata = test.hog,
                                       type = "raw")

confusionMatrix(prediccion.NaiveBayes.Caret, test.hog[,325])

# Accuracy/Precisión: 0.8211

saveRDS(modelo.NaiveBayes.Caret, "mod.NB.rds")
```

Figura 52: Modelo de clasificación usando el método de HOG y Naive Bayes

Para la evaluación se creó la matriz de confusión con el conjunto de datos de prueba de HOG “test.HOG”.

```
                  Confusion Matrix and Statistics
          Reference
Prediction Blep Chir Elmi Olig Perl Ptil
      Blep   19    0    0    1    0    3
      Chir    0   22    0    1    0    0
      Elmi    0    0   20    0    7    0
      Olig    0    0    0   10    0    0
      Perl    2    0    2    1   14    3
      Ptil    1    0    0    1    1   16

Overall Statistics

               Accuracy : 0.8211
                 95% CI : (0.7418, 0.8844)
    No Information Rate : 0.1789
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7837
```

Figura 53: Matriz de confusión del modelo de Naive Bayes usando el método de HOG

- ***Modelo de clasificación usando el método de HOG y Random Forest***

Se creó el modelo random forest usando el método de HOG y se creó la matriz de confusión del mismo para evaluar el modelo.

```
1  library(caret)
2  set.seed(1980)
3
4  # Random Forest con el paquete Caret
5
6  pc <- proc.time()
7  modelo.RandomForest.Caret <- train(train.hog,
8                                     train.hog$labels,
9                                     method = 'rf')
10 proc.time() - pc
11
12 prediccion.RandomForest.Caret <-predict(modelo.RandomForest.Caret,
13                                          newdata = test.hog,
14                                          type = 'raw')
15
16 confusionMatrix(prediccion.RandomForest.Caret, test.hog[,325])
17
18 # Accuracy/Precisión: 0.9705
19
20 saveRDS(modelo.RandomForest.Caret, "mod.RF.rds")
```

```
   user   system  elapsed
 287.81    1.39   294.25
```

Figura 54: Modelo de clasificación usando el método de HOG y Random Forest

```
prediccion.RandomForest.Caret <-predict(modelo.RandomForest.Caret,
                                        newdata = test.hog,
                                        type = 'raw')

confusionMatrix(prediccion.RandomForest.Caret, test.hog[,325])

# Accuracy/Precisión: 0.9705

saveRDS(modelo.RandomForest.Caret, "mod.RF.rds")
```

```
                  Confusion Matrix and Statistics
          Reference
Prediction Blep Chir Elmi Olig Perl Ptil
      Blep   22    0    0    0    0    0
      Chir    0   22    0    0    0    0
      Elmi    0    0   22    0    0    0
      Olig    0    0    0   13    0    0
      Perl    0    0    0    0   22    0
      Ptil    0    0    0    0    0   22

Overall Statistics
               Accuracy : 1
                 95% CI : (0.9705, 1)
    No Information Rate : 0.1789
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

Statistics by Class:
                     Class: Blep Class: Chir Class: Elmi Class: Olig
Sensitivity               1.0000      1.0000      1.0000      1.0000
Specificity               1.0000      1.0000      1.0000      1.0000
Pos Pred Value            1.0000      1.0000      1.0000      1.0000
Neg Pred Value            1.0000      1.0000      1.0000      1.0000
Prevalence                0.1789      0.1789      0.1789      0.1057
Detection Rate            0.1789      0.1789      0.1789      0.1057
Detection Prevalence      0.1789      0.1789      0.1789      0.1057
Balanced Accuracy         1.0000      1.0000      1.0000      1.0000
                     Class: Perl Class: Ptil
Sensitivity               1.0000      1.0000
Specificity               1.0000      1.0000
Pos Pred Value            1.0000      1.0000
Neg Pred Value            1.0000      1.0000
Prevalence                0.1789      0.1789
Detection Rate            0.1789      0.1789
Detection Prevalence      0.1789      0.1789
Balanced Accuracy         1.0000      1.0000
```

Figura 55: Matriz de confusión del modelo de clasificación usando el método de HOG y Random Forest

- ***Modelo de clasificación usando el método de HOG y SVM***

Para la creación de este modelo además de la librería caret se utilizó la librería e1071. Se creó el modelo con la aplicación del comando svm y se realizó su matriz de confusión.

```
library(caret)
library(e1071)

set.seed(1980)

pc <- proc.time()
modelo.SVM <- svm(labels ~ ., data = train.hog)
proc.time() - pc

prediccion.SVM <- predict(modelo.SVM, newdata = test.hog)

confusionMatrix(prediccion.SVM, test.hog$labels)

# Accuracy / Precisión: 0.9593

saveRDS(modelo.SVM, "mod.SVM.rds")
```

```
user  system elapsed
0.47    0.02    0.52

                  Confusion Matrix and Statistics
          Reference
Prediction Blep Chir Elmi Olig Perl Ptil
      Blep   22    0    0    0    0    0
      Chir    0   22    0    0    0    0
      Elmi    0    0   21    0    1    0
      Olig    0    0    0   12    0    1
      Perl    0    0    1    1   20    0
      Ptil    0    0    0    0    1   21

Overall Statistics
               Accuracy : 0.9593
                 95% CI : (0.9077, 0.9867)
    No Information Rate : 0.1789
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.951

 Mcnemar's Test P-Value : NA

Statistics by Class:
                     Class: Blep Class: Chir Class: Elmi Class: Olig
Class: Perl Class: Ptil
Sensitivity               1.0000      1.0000      0.9545     0.92308
0.9091      0.9545
Specificity               1.0000      1.0000      0.9901     0.99091
0.9802      0.9901
Pos Pred Value            1.0000      1.0000      0.9545     0.92308
0.9091      0.9545
Neg Pred Value            1.0000      1.0000      0.9901     0.99091
0.9802      0.9901
Prevalence                0.1789      0.1789      0.1789     0.10569
0.1789      0.1789
Detection Rate            0.1789      0.1789      0.1707     0.09756
0.1626      0.1707
Detection Prevalence      0.1789      0.1789      0.1789     0.10569
0.1789      0.1789
Balanced Accuracy         1.0000      1.0000      0.9723     0.95699
0.9446      0.9723
```

Figura 56: Modelo de clasificación y matriz de confusión usando el método de HOG y svm

- ***Modelo Deep learning con h20 y método de HOG***

Se creó un modelo de Deep learning con h2o utilizando los dataframes obtenidos en el método de HOG. Y se realizó la matriz de confusión para la evaluación de este modelo.

```
set.seed(1980)
local.h2o <- h2o.init(ip = 'localhost',
                      port = 54321,
                      startH2O = T,
                      nthreads = 1)
# Se suben los datasets
train.hog.h2o <- as.h2o(train.hog)
test.hog.h2o <- as.h2o(test.hog)
# Se entrena el modelo:
mod.macro.hog.H2O <- h2o.deeplearning(x = 1:324, y = 325,
                                      train.hog.h2o, activation = 'Tanh',
                                      hidden = rep(160,5), epochs = 20)
pred.hog.H2O <- h2o.predict(mod.macro.hog.H2O,
                            newdata = test.hog.h2o[,-325])
pred.hog.H2O.df <- as.data.frame(pred.hog.H2O)
confusionMatrix(pred.hog.H2O.df$predict, test.hog[,325])
#h2o.confusionMatrix(pred.hog.H2O, test.hog.h2o[,1])

# Accuracy/Precisión:0.8699
h2o.shutdown(prompt = F)
```

```
                    Confusion Matrix and Statistics
          Reference
Prediction Blep Chir Elmi Olig Perl Ptil
      Blep   21    0    0    0    0    1
      Chir    1   22    0    0    0    0
      Elmi    0    0   22    0    3    0
      Olig    0    0    0   12    0    2
      Perl    0    0    0    0   19    0
      Ptil    0    0    0    1    0   19

Overall Statistics
               Accuracy : 0.935
                 95% CI : (0.8759, 0.9715)
    No Information Rate : 0.1789
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9216

 Mcnemar's Test P-Value : NA

Statistics by Class:
                     Class: Blep Class: Chir Class: Elmi Class: Olig Class:
Perl Class: Ptil
Sensitivity              0.9545      1.0000      1.0000     0.92308
0.8636      0.8636
Specificity              0.9901      0.9901      0.9703     0.98182
1.0000      0.9901
Pos Pred Value           0.9545      0.9565      0.8800     0.85714
1.0000      0.9500
Neg Pred Value           0.9901      1.0000      1.0000     0.99083
0.9712      0.9709
Prevalence               0.1789      0.1789      0.1789     0.10569
0.1789      0.1789
Detection Rate           0.1707      0.1789      0.1789     0.09756
0.1545      0.1545
Detection Prevalence     0.1789      0.1870      0.2033     0.11382
0.1545      0.1626
Balanced Accuracy        0.9723      0.9950      0.9851     0.95245
0.9318      0.9269
0.9446      0.9723
```

Figura 57: Modelo Deep learning con h20 y método de HOG y matriz de confusión

3.2.2.5 Diseño de los algoritmos computacionales con 2244 observaciones

Con la base de datos aumentada generada en el software Python con la librería Keras, descrito en la Figura 36, y la nueva matriz de datos numéricos con 2442 observaciones y 872 variables se ejecutaron los algoritmos realizados con la primera base de datos de 504 observaciones.

En Rstudio se cargó la nueva matriz de datos numéricos. Figura 46.

```
library(readr)
dm <- read_csv("dat_macro.csv")
dm<-dm[,-1]
nm <- read_csv("nombres2_macro.csv")
nm<-nm[,-1]
names(nm)<-"labels"
dat<-cbind(nm,dm)
dat$labels<-as.factor(dat$labels)

row = 67
col = 13
l=row*col
```

Figura 58: Nueva matriz de datos numéricos cargada en RStudio

Se verificó que el set de datos esté balanceado en sus categorías y se dibujó una imagen de muestra.

```
Blep Chir Elmi Olig Perl Ptil
 407 407  407  407  407  407
```

Figura 59: Balance del set de datos

Se dividió el nuevo conjunto de datos en dos:

1. Conjunto de datos de entrenamiento “data_train_2” con porcentaje de 75%.
2. Conjunto de datos de prueba “data_test_2” con el porcentaje de 25%.

```
# Se carga la librería
library(caret)
# Se establece una semilla
set.seed(2020)
# Se establece el porcentaje para los datos de entrenamiento
train_perc = 0.75
# Se crea el índice para realizar las particiones
train_index = createDataPartition(dat$Labels, p = train_perc,
                                  list = FALSE)
# Se extraen los datos de entrenamiento
data_train = dat[train_index,]
# Se extraen los datos de prueba "test" con los índices complementarios
data_test = dat[-train_index,]
```

```
data_train    1836 obs. of  872variables
data_test      606 obs. of  872 variables
```

Figura 60: División del nuevo conjunto de datos en entrenamiento y prueba

En la creación de los modelos se utilizaron los códigos descritos anteriormente en cada uno, y se generaron las matrices de confusión correspondientes a cada modelo para la posterior evaluación.

En el modelo de regresión logística se aplicó el código de la Figura 46 y se obtuvo la matriz de confusión.

```
prediction_lr
       Blep Chir Elmi Olig Perl Ptil
  Blep   50    0    7   36    1    7
  Chir    1   99    0    1    0    0
  Elmi    7    0   81    9    3    1
  Olig   23    8   11   53    0    6
  Perl   11   12   36   19   22    1
  Ptil   20    4   11   37    3   26
[1] 0.5462046
```

Figura 61: Matriz de confusión del modelo de regresión logística con datos aumentados

En el modelo de clasificación Deep learning con h2o se aplicó el código de la Figura 47 y se obtuvo la matriz de confusión.

```
                          Confusion Matrix and Statistics
          Reference
Prediction Blep Chir Elmi Olig Perl Ptil
      Blep   91    0    3   36    2   37
      Chir    0   99    0    8    2    1
      Elmi    4    0   96   10   60    7
      Olig    3    0    1   43    2   29
      Perl    0    1    1    3   35    3
      Ptil    3    1    0    1    0   24
Overall Statistics
               Accuracy : 0.6403
                 95% CI : (0.6006, 0.6785)
    No Information Rate : 0.1667
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5683

 Mcnemar's Test P-Value : NA

Statistics by Class:
                     Class: Blep Class: Chir Class: Elmi Class: Olig Class: Perl Class: Ptil
Sensitivity               0.9010      0.9802      0.9505     0.42574     0.34653     0.23762
Specificity               0.8455      0.9782      0.8396     0.93069     0.98416     0.99010
Pos Pred Value            0.5385      0.9000      0.5424     0.55128     0.81395     0.82759
Neg Pred Value            0.9771      0.9960      0.9883     0.89015     0.88277     0.86655
Prevalence                0.1667      0.1667      0.1667     0.16667     0.16667     0.16667
Detection Rate            0.1502      0.1634      0.1584     0.07096     0.05776     0.03960
Detection Prevalence      0.2789      0.1815      0.2921     0.12871     0.07096     0.04785
Balanced Accuracy         0.8733      0.9792      0.8950     0.67822     0.66535     0.61386
```

Figura 62: Matriz de confusión del modelo de clasificación Deep learning con h2o con datos aumentados

En el modelo de clasificación usando método de HOG y KNN se aplicó el código de la Figura 49 y se obtuvo su accurcy.

```
time to complete : 18.6965 secs

[1] 0.7457265 0.7083333 0.7061404 0.7368421
[1] 0.7242606
```

Figura 63: Modelo de clasificación usando método de HOG y knn con datos aumentados

En el modelo de regresión logística usando el método de HOG se aplicó el código de la Figura 50 y se obtuvo la matriz de confusión

```
prediction_RL_HOG.hog
       Blep Chir Elmi Olig Perl Ptil
  Blep   72    0    2    1    5    21
  Chir    3   96    0    0    2     0
  Elmi   16    0   47    0   28    10
  Olig   46   17    3   12    5    18
  Perl   27    8    5    4   50     7
  Ptil   39    6    0   10    4    42
[1] 0.5264026
```

Figura 64: Matriz de confusión del modelo de regresión logística usando el método de HOG con datos aumentados

En el modelo de Naive Bayes usando el método de HOG se aplicó el código de la Figura 52y se obtuvo la matriz de confusión.

```
                  Confusion Matrix and Statistics
          Reference
Prediction Blep Chir Elmi Olig Perl Ptil
      Blep   37    3    7   17   10    22
      Chir   10   55    2    8    8     7
      Elmi   44   30   86   63   66    58
      Olig    7    3    1    6    5     5
      Perl    2    5    5    5   12     2
      Ptil    1    5    0    2    0     7

Overall Statistics

               Accuracy : 0.335
                 95% CI : (0.2975, 0.3741)
    No Information Rate : 0.1667
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.202
```

Figura 65: Matriz de confusión del modelo de Naive Bayes usando el método de HOG con datos aumentados

En el modelo de clasificación usando método de HOG y Random Forest se aplicó el código de la Figura 54 y se obtuvo la matriz de confusión.

```
                          Confusion Matrix and Statistics
          Reference
Prediction Blep Chir Elmi Olig Perl Ptil
      Blep   101    0    0    0    0    0
      Chir     0  101    0    0    0    0
      Elmi     0    0  101    0    0    0
      Olig     0    0    0  101    0    0
      Perl     0    0    0    0  101    0
      Ptil     0    0    0    0    0  101

Overall Statistics
               Accuracy : 1
                 95% CI : (0.9939, 1)
    No Information Rate : 0.1667
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

Statistics by Class:
                     Class: Blep Class: Chir Class: Elmi Class: Olig Class: Perl Class: Ptil
Sensitivity               1.0000      1.0000      1.0000      1.0000      1.0000      1.0000
Specificity               1.0000      1.0000      1.0000      1.0000      1.0000      1.0000
Pos Pred Value            1.0000      1.0000      1.0000      1.0000      1.0000      1.0000
Neg Pred Value            1.0000      1.0000      1.0000      1.0000      1.0000      1.0000
Prevalence                0.1667      0.1667      0.1667      0.1667      0.1667      0.1667
Detection Rate            0.1667      0.1667      0.1667      0.1667      0.1667      0.1667
Detection Prevalence      0.1667      0.1667      0.1667      0.1667      0.1667      0.1667
Balanced Accuracy         1.0000      1.0000      1.0000      1.0000      1.0000      1.0000
```

Figura 66: Matriz de confusión usando el método de HOG y Random Forest con datos aumentados

En el modelo de clasificación usando método de HOG y SVM se aplicó el código de la Figura 57 y se obtuvo la matriz de confusión.

```
     user  system elapsed
   6.97    0.11    7.25
                          Confusion Matrix and Statistics
          Reference
Prediction Blep Chir Elmi Olig Perl Ptil
      Blep   79    0    5    8    3   22
      Chir    1   99    0    9    2    2
      Elmi    6    0   84    7   16    4
      Olig    2    0    0   59    7   14
      Perl    0    2    7    5   73    2
      Ptil   13    0    5   13    0   57
Overall Statistics
               Accuracy : 0.7442
                 95% CI : (0.7075, 0.7785)
    No Information Rate : 0.1667
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6931

 Mcnemar's Test P-Value : NA
Statistics by Class:
                     Class: Blep Class: Chir Class: Elmi Class: Olig Class: Perl Class: Ptil
Sensitivity               0.7822      0.9802      0.8317     0.58416      0.7228     0.56436
Specificity               0.9248      0.9723      0.9347     0.95446      0.9683     0.93861
Pos Pred Value            0.6752      0.8761      0.7179     0.71951      0.8202     0.64773
Neg Pred Value            0.9550      0.9959      0.9652     0.91985      0.9458     0.91506
Prevalence                0.1667      0.1667      0.1667     0.16667      0.1667     0.16667
Detection Rate            0.1304      0.1634      0.1386     0.09736      0.1205     0.09406
Detection Prevalence      0.1931      0.1865      0.1931     0.13531      0.1469     0.14521
Balanced Accuracy         0.8535      0.9762      0.8832     0.76931      0.8455     0.75149
```

Figura 67: Matriz de confusión usando el método de HOG y SVM con datos aumentados

Para el último modelo de Deep learning con h20 y método de HOG se aplicó el código de la Figura 57 y se obtuvo la matriz de confusión.

```
                          Confusion Matrix and Statistics
          Reference
Prediction Blep Chir Elmi Olig Perl Ptil
      Blep   95    1   18   63   13   65
      Chir    2   90    0    8    5    0
      Elmi    1    1   79    3   46    8
      Olig    0    1    1   22    2    5
      Perl    0    6    2    1   35    1
      Ptil    0    2    1    4    0   22

Overall Statistics
               Accuracy : 0.566
                 95% CI : (0.5255, 0.6059)
    No Information Rate : 0.1667
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4792

 Mcnemar's Test P-Value : NA

Statistics by Class:
                     Class: Blep Class: Chir Class: Elmi Class: Olig Class: Perl
Class: Ptil
Sensitivity               0.9901      0.9307     0.56436    0.009901     0.20792    0.049505
Specificity               0.5406      0.9347     0.92079    0.994059     0.97822    0.982178
Pos Pred Value            0.3012      0.7402     0.58763    0.250000     0.65625    0.357143
Neg Pred Value            0.9964      0.9854     0.91356    0.833887     0.86063    0.837838
Prevalence                0.1667      0.1667     0.16667    0.166667     0.16667    0.166667
Detection Rate            0.1650      0.1551     0.09406    0.001650     0.03465    0.008251
Detection Prevalence      0.5479      0.2096     0.16007    0.006601     0.05281    0.023102
Balanced Accuracy         0.7653      0.9327     0.74257    0.501980     0.59307    0.515842
```

Figura 68: Matriz de confusión del modelo Deep learning con h20 y método de HOG con datos aumentados

Finalmente se evaluó el rendimiento de cada uno los modelos mediante las métricas de evaluación Accuracy y matriz de confusión, luego se comparó los modelos ejecutados con los datos originales y con la base de datos aumentada y se eligió el óptimo para el caso.

CAPITULO IV

4. RESULTADOS Y DISCUSIÓN

4.1 Resultados de la clasificación aplicando el método de inspección visual

A causa de que no existe variedad de guías para identificar y caracterizar macroinvertebrados, se identificaron y separaron cinco familias: Chironomidae, Elmidae, Ptilodactylidae, Perdilae y Blephariceridae y una clase la Oligocaheta y se extrajeron 8 características morfométricas comunes entre las familias: longitud, ancho, forma del cuerpo, segmentos corporales, número de patas, cubierta de pelos, antenas y alas. Tabla 4.

Familia Chironomidae	Familia Elmidae	Familia Ptilodactylidae
Familia Perlidae	Familia Blephariceridae	Clase Oligochaeta

Figura 69: Identificación de macroinvertebrados

Al momento de extraer las características morfométricas, en la categoría Ancho (mm) se realizó un rango tomando en cuenta el menor y el mayor valor medido de los 30 individuos pertenecientes a cada familia, debido a que no se encontró esta información en referencia bibliográfica.

Previo a al diseño del algoritmo se transformó la categoría “Forma de cuerpo” a variables “dummy”, además, se estableció la relación ancho/largo. Tabla 6.
Con esta base de datos se ejecutó el modelo predictivo *Random Forest*, dividiendo el 75% de las muestras para entrenamiento (381 imágenes) y el 25% restante para la evaluación (123 imágenes), este modelo presenta un Accuracy/precisión de 1, es decir, el 100%, de las 123 imágenes de evaluación clasificó correctamente todas.

Tabla 6: Extracción de las características y transformación a variables “dummy”

Clase	Labels	AlargTub	CilinSeg	Ovalado	ConvexAlargOval	Alargado	SegCorp	Npatas	CubPelos	Antenas	Alas	Longitud	Ancho	A/L
Insecta	Chir	Si	No	No	No	No	12	4	Si	No	No	2-10	0,4-0,8	-
Oligocaheta	Olig	No	Si	No	No	No	3	0	No	No	No	1-30	1-3	-
Insecta	Elmi	No	Si	Si	No	No	2	6	No	Si	No	1-10	0,4-1	-
Insecta	Ptil	No	Si	No	Si	No	4a11	6	Si	Si	No	3-15	0,7-3	-
Insecta	Perl	No	Si	No	No	Si	3	6	Si	Si	No	20-50	0,7-2	-
Insecta	Blep	No	Si	No	No	Si	6	12	No	No	No	7-8	0,6-3	-

Finalmente se evaluó el modelo de manera manual ingresando cada una de las categorías, se ingresaron datos correspondientes a la familia Ptilodactylidae y se ejecutó el modelo. Se comprobó que lo clasifica correctamente ya que el resultado que brinda pertenece a la familia Ptilodactylidae.

Ptilodactylidae
Levels: Blepharoceridae Chironomidae Elmidae Oligocaheta Perlidae Ptilodactylidae

Figura 70: Resultado identificación de la familia Ptilodactylidae

Se ingresaron distintos datos correspondientes a la familia Elmidae y se ejecutó el modelo, de igual manera que el anterior, el resultado es correcto, clasifica al individuo como familia Elmidae.

```
# Ingresar manualmente las características del macroinvertebrado
ingreso$AlargTub = factor("No",levels = c("Si","No"))
ingreso$CilinSeg = factor("Si",levels = c("Si","No"))
ingreso$Ovalado = factor("No",levels = c("Si","No"))
ingreso$ConvexAlargOval= factor("Si",levels = c("Si","No"))
ingreso$Alargado = factor("No",levels = c("Si","No"))
ingreso$SegCorp = factor("4a11",levels = c("12", "2", "3", "4a11", "6"))
ingreso$Npatas = factor("4",levels = c("0","4","6","12"))
ingreso$CubPelos = factor("Si",levels = c("Si","No"))
ingreso$Antenas = factor("No",levels = c("Si","No"))
ingreso$Longitud = 12
ingreso$Ancho = 5
ingreso$L.A=ingreso$Ancho/ingreso$Longitud
ingreso<-data.frame(ingreso)
(pred1<-pred<-predict(superModel,ingreso))
```

Elmidae
Levels: Blepharoceridae Chironomidae Elmidae Oligocaheta Perlidae Ptilodactylidae

Figura 71: Resultado identificación de la familia Chironomidae

4.2 Resultados de la clasificación aplicando el método automatizado

Para solucionar el problema que se tenía del tamaño de las imágenes, estas se estandarizaron con el algoritmo de *Python*, de 1536x1024= 1.572.864 píxeles que presentaban, se aplanaron a vectores de 84x39 = 3276 píxeles toda la base de datos original de 504 imágenes.

La extracción de características es una etapa importante en el procesamiento de imágenes, las características extraídas manualmente en el laboratorio no implementaron una base de datos muy completa acera de la morfometría detallada de cada macroinvertebrado, por esta razón se debe la aplicación del método de HOG, ya que este obtiene más información de morfometría dividiendo en una cuadricula y tomando gradientes de cada uno de los contornos de cada macroinvertebrado. En el modelo de regresión logística al ser ejecutado con123 observaciones de 3276 variables se tiene un Accuracy/precisión de 73,1%, sin embargo, este porcentaje aumenta a 75,6 al ejecutarse con 123 observaciones de 324 variables, resultado de la aplicación del método de HOG para extraer características. Es por esto que se trabajó los modelos restantes: Knn, Naive Bayes, *Random Forest*, SVM y clasificación con h2o, con el conjunto de datos de entrenamiento del método de HOG.

Utilizando el 75% de las muestras para entrenamiento y el 25% restante para la evaluación al implementar los algoritmos descritos en las Figuras 46, 47, 49, 50, 52, 54, 56 y 67, resultó que los modelos *Random Forest*, SVM y el de *Deep learning* con h2o tienen un Accuracy/precisión bastante alta de 100%, 95% y 93% respectivamente. Mientras que el de los otros modelos son inferiores al 90%, por lo que los convierte en una opción no óptima para el caso. Figura 57.

El modelo que mayor precisión presentó es el *Random Forest* con 97%, es decir, de las 123 observaciones del set de datos de prueba clasificó mal solamente 4.

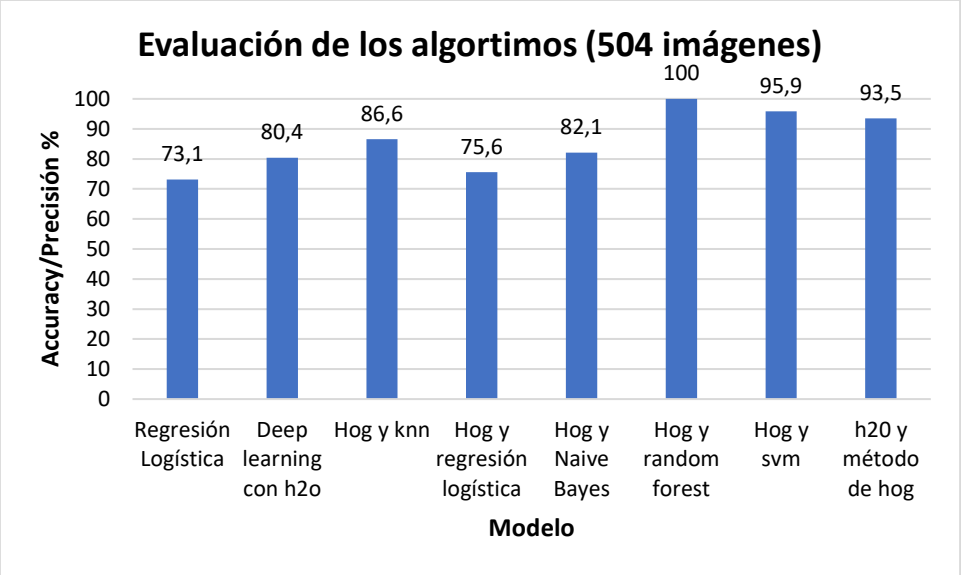

Figura 72: Evaluación de los algoritmos con 504 observaciones

Por lo expresado anteriormente en la metodología, se ejecutaron los mismos modelos, pero esta vez con la base de datos aumentada resultado del método, *ImageDataGenerator*, se logró aumentar en un 79% la base de datos original, de 504 imágenes se generaron 2442. Las nuevas imágenes generadas se estandarizaron con el algoritmo de *Python*, y se aplanaron a vectores de 67x13 = 871 píxeles.

Se dividió el conjunto de datos en 75% de las muestras para entrenamiento (1836 imágenes) y el 25% restante para la evaluación (606 imágenes).

Los resultados de la implementación de los algoritmos están resumidos en la Figura 72. Se observa que las precisiones de los modelos de: Regresión Logística*, Deep Learning* con h2o, h2o con el método de HOG y Naive Bayes bajaron considerablemente con relación a los anteriores, tienen un porcentaje menor al 70%, por lo que son descartados para el caso.

Sin embargo, el modelo de *Random Forest* mantiene su *Accuracy*/precisión en 100%, es decir que las 606 observaciones de la base de datos de prueba, clasificó bien todas.

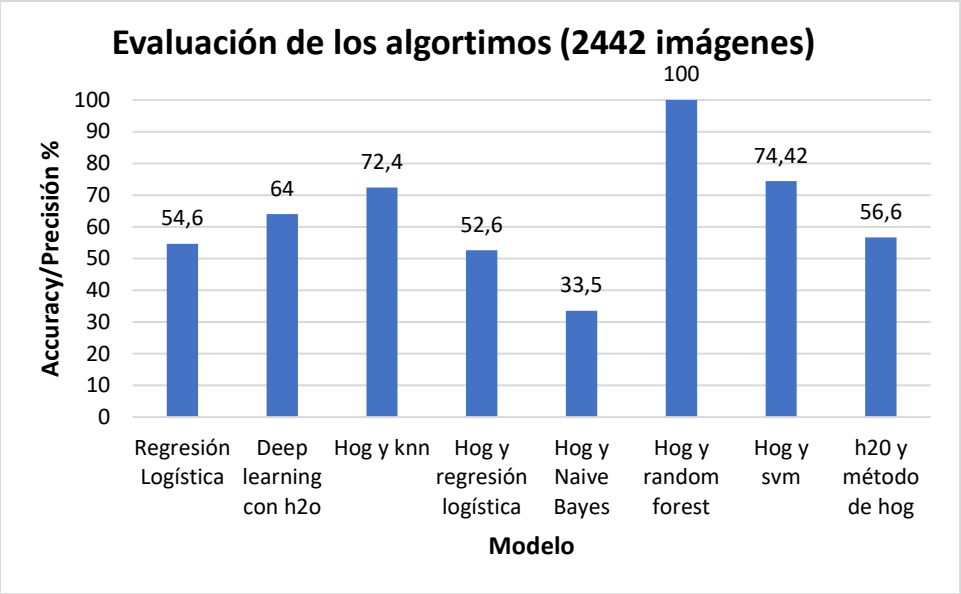

Figura 73: Evaluación de los algoritmos con 2442 observaciones

Si bien ya se tenía el máximo *Accuracy* con el primer modelo de *Random Forest* usando el método de HOG con 504 observaciones, este se mantuvo al ser ejecutado con la base de datos aumentada, es importante acotar que en el caso de *Random Forest* no existió un sobreajuste ya que se introduce de manera aleatoria la formación de los árboles.

Tabla 7: Comparación de la precisión del modelo Random Forest con HOG

Modelo Random Forest usando HOG	**504 observaciones**	**2442 observaciones**
Precisión %	100	100

```
     Confusion Matrix and Statistics
               Reference
Prediction Blep Chir Elmi Olig Perl Ptil
      Blep   22    0    0    0    0    0
      Chir    0   22    0    0    0    0
      Elmi    0    0   22    0    0    0
      Olig    0    0    0   13    0    0
      Perl    0    0    0    0   22    0
      Ptil    0    0    0    0    0   22

Overall Statistics
                Accuracy : 1
                  95% CI : (0.9705, 1)
```

```
     Confusion Matrix and Statistics
               Reference
Prediction Blep Chir Elmi Olig Perl Ptil
      Blep  101    0    0    0    0    0
      Chir    0  101    0    0    0    0
      Elmi    0    0  101    0    0    0
      Olig    0    0    0  101    0    0
      Perl    0    0    0    0  101    0
      Ptil    0    0    0    0    0  101

Overall Statistics
                Accuracy : 1
                  95% CI : (0.9939, 1)
```

Figura 74: Comparación de la matriz de confusión del modelo Random Forest con HOG

Los macroinvertebrados bentónicos acuáticos son el grupo más utilizado dentro de los bioindicadores de calidad de agua de los ríos andinos debido que son capaces de otorgar información de perturbaciones, alteración física del cauce y de la ribera, además poseen alta diversidad, el nivel de muestreo se realiza de manera fácil, y los diferentes taxones presentan requerimientos ecológicos diferentes. A pesar, de los beneficios ambientales que representan, en la actualidad no existen guías completas que faciliten la identificación de los macroinvertebrados ya sea por familia, género o especie.

La aplicación de los modelos computacionales para la identificación de macroinvertebrados disminuye el tiempo empleado en esta actividad, lo que se traduce a un ahorro de recursos. Para que exista un modelo óptimo se debe generar un set con grandes volúmenes de datos. Además, en base a los algoritmos ya generados y evaluados, se integren otras familias de macroinvertebrados, para que el sistema siga creciendo, entrenando y mejorando su funcionamiento.

Por todo lo expresado se decidió que la mejor opción para la identificación de los macroinvertebrados es la implementación del modelo de *Random Forest* usando el método de HOG para la extracción de las características, ya que al automatizar este proceso de identificación por familias facilitará la determinación de los índices de calidad y brindaran resultados más precisos referentes a la calidad de agua de los ríos andinos.

CAPÍTULO V

5. CONCLUSIONES

Según el estado del arte realizado sobre el tema, en los métodos establecidos para la caracterización de macroinvertebrados no se dispone de metodologías actuales establecidas que apoyen el enfoque educativo sobre el tema de detección automática o sistemas expertos que fomenten el auto aprendizaje, pues en la mayoría de los casos se requiere de la ayuda de un experto en el tema, ya que la información que brindan es muy puntual. Es por esto que el presente estudio implementa dos metodologías que pueden incrementar la base de datos para fortalecer y crear un sistema experto.

El clasificador de macroinvertebrados se fundamenta principalmente en las máquinas de aprendizaje y en la extracción de las características de la imagen, siendo el método de inspección visual no tan preciso ya que está expuesta a errores de estimación de parámetros visuales por parte de quien lo realiza debido a que existen macroinvertebrados que se diferencian unos de otros por pequeñas características, además para realizar este proceso se debe invertir muchas horas de trabajo. Mediante la aplicación de sistemas automatizados la identificación optimiza precisión y tiempo, para el caso, el Histograma de Gradientes Orientados (HOG) es el método que se utilizó para describir y extraer las características de forma automática para su posterior aplicación en los algoritmos de *Machine Learning* y *Deep Learning.*

El Accuracy/precisión del modelo de clasificación *Random Forest* generado con los datos de la extracción manual, no presentó ningún inconveniente, ya que su valor es del 100%, esto se debe a que las variables son categóricas, extraídas manualmente, no existen variables exteriores que puedan modificar el resultado, además no existe ningún tipo un sobreajuste en el modelo. Sin embargo, este método manual es eficaz para que el investigador interactúe directamente con los macroinvertebrados, conozca sus taxones, los identifique, extraiga sus características y sobre todo aprenda y domine el tema. Puesto a que puede apoyarse en el clasificador automático para verificar su acierto o en caso de haber fallado conocerlo, ya que el mayor inconveniente que se puede presentar en este sistema de identificación es que, el observador cometa un error al momento de ingresar la estimación de los parámetros.

El modelo de clasificación Random Forest usando el método del HOG es el que mejor se adapta para lograr el objetivo del proyecto, mantuvo su Accuracy/precisión máximo del 100% al ampliar la base de datos de 504 a 2442, es decir que, de las 123 y 606 imágenes de prueba, en los casos clasificó correctamente todas. La efectividad va a depender de las variables de entrada, cantidad y calidad de datos que tenga el set de datos de entrenamiento y validación.

CAPÍTULO VI

6. RECOMENDACIONES

Si el investigador pretende mejorar la técnica de identificación y extracción visual de características morfométricas mediante el uso del estereoscopio es recomendable que los resultados obtenidos los compare con los resultados que proporcione el clasificador automatizado y así verifique a el acierto o el error en caso de haber fallado.

Para el clasificador automatizado se recomienda que las imágenes que se vayan a tomar o adquirir presenten todas las mismas condiciones de iluminación, color de fondo, calidad, para que en momento del pre procesamiento, estas no tengan factores externos que afecten el resultado.

La presente tesis es un aporte muy importante para futuras investigaciones en donde se aplique la metodología del metaanálisis. Además, continuará alimentando un posible proyecto a futuro sobre la implementación de un sistema online con interfaces gráficas que permitirá a estudiantes y profesionales consultar e identificar imágenes de macroinvertebrados y a la vez con posibilidad de alimentar la base de datos sobre calidad de agua de sectores georreferenciados.

En trabajos futuros para la construcción de varios clasificadores los algoritmos implementados pueden ser generalizados, ampliando la base de datos con otros macroinvertebrados, clasificándolos ya sea por familias, especies o géneros, y buscando otros sistemas de extracción de descriptores para que los macroinvertebrados se puedan diferenciar por otras características. De esta manera la determinación de la calidad de los cuerpos de agua, cauce y de la ribera, de donde proceden estos bioindicadores será precisa y confiable, ya que para determinar el índice correcto de calidad ya sea el ABI o el BMWP/Col, es fundamental realizar una buena identificación de estos bioindicadores. Además, para la clasificación manual, con la ayuda de un experto informático se puede realizar un interfaz en dónde se puedan ingresar las características extraídas de una manera más cómoda.

REFERENCIAS BIBLIOGRÁFICAS

Aguirre, N. (2015.). *Procesamiento de imágenes*. 42–72.

Alonso Ramírez. (2010, December). *Capítulo 2: Métodos de recolección*. 41–50.

Alvarado, J. P. (2012). Procesamiento y análisis de imagen digitales. *Escuela de Electrónica, Instituto Tecnológico de Costa Rica*.

Amat, J. (2018). Machine Learning con R y caret. Retrieved from https://rstudio-pubs-static.s3.amazonaws.com/293405_4029f1f23f834b7195189d5504a436b2.html

Bernardo, J., & Gomez, V. (n.d.). *Análisis y diseño de algoritmos*.

Camacho Sosa, J. M. (2016). Máquinas de Soporte Vectorial - EcuRed. https://doi.org/10.1016/j.biocon.2014.10.019

Candel, A., & Parmar, V. (2015). *Deep Learning with H2O*. 1–21.

Chollet, F. (2018). Deep Learning with Python. In *2018 21st International Conference on Information Fusion, FUSION 2018*. https://doi.org/10.23919/ICIF.2018.8455530

Córdoba C. (2001) Photoshop 7. Apéndice. Editorial Alfa-omega; 9: 443-470.

Cordero, R. A. R. (2015). *Sistema de detección y rastreo de personas en tiempo real para el robot Golem-II+*. Retrieved from http://golem.iimas.unam.mx/pubs/romero15sistema.pdf

Cortes Osorio, J., Urueña, W., & Mendoza Vargas, J. (2011). Técnicas alternativas para la conversión de imágenes a color a escala de grises en el tratamiento digital de imágenes. *Scientia Et Technica*, *XVII*(47), 207–212. https://doi.org/10.22517/23447214.533

Elgueta Morales, J. A.--jorelgue@gmail. co., & Jorelgue@gmail.com, J. A.--. (2017). *Comparación de rendimiento de técnicas de aprendizaje automático para análisis de afecto sobre textos en español*. Retrieved from http://repobib.ubiobio.cl/jspui/handle/123456789/1772

Flores Ereña, J. G. (2016). *Sistesis digital de color utilizando tonos de gris*. *4*(1), 64–75.

Gutierrez, Juan; Riss, Wolfgang; Ospina, R. (2006). Bioindication of water quality at the sabanna of Bogota-Colombia , using neuro-adaptative fuzzy logic LA UTILIZACIÓN DE LA LÓGICA DIFUSA NEURO-. *Limnología*, *28*(1), 45–56.

Heras Benavides, D., 2017. Clasificador de imágenes de frutas basado en inteligencia artificial. *Revista Killkana Técnica,* mayo-agosto, 1(2), pp. 21-30.

Jiménez Ochoa, M. G. (2015). Desarrollo de un sistema de visión artificial para la detección de aglomerción de personas en un semáforo. *Dspace.Unl.Edu.Ec*. Retrieved from http://dspace.unl.edu.ec/jspui/bitstream/123456789/11225/1/Jiménez Ochoa%2C Magaly Gabriela.pdf

Laguna, L. (2016). *reconocimiento de imágenes en Raspberry Pi 2*.

Lemus Serrano, C. E. (2011). *Diseño De Un Modelo Basado En Técnicas De Inteligencia Artificial Para El Desarrollo De Un Sistema Inteligente Orientado Al Aprendizaje*. Retrieved from http://www.redicces.org.sv/jspui/bitstream/10972/419/1/Diseño de

un modelo basado en técnicas de inteligencia artificial para el desarrollo de un sistem.pdf

Liñero, I., Balarezzo, V., Eraso, H., Pacheco, F., Ramos, C., Muzo, R., & Calva, C. (2016). Calidad del agua de un río andino ecuatoriano a través del uso de macroinvertebrados. *Cuadernos de Investigación UNED*, *8*(1), 69–75.

Madsen, L. B., Lévêque, C., Omiste, J. J., & Miyagi, H. (2018). CHAPTER 11: Time-dependent Restricted-active-space Self-consistent-field Theory for Electron Dynamics on the Attosecond Timescale. *RSC Theoretical and Computational Chemistry Series*, *2018-Janua*(13), 386–423. https://doi.org/10.1039/9781788012669-00386

Mejía, J. (1996). Procesamiento digital de imágenes. *Perfiles Educativos*, (72).

Mena, L. J. (2008). *Aprendizaje Automático a partir de Conjuntos de Datos No Balanceados y su Aplicación en el Diagnóstico y Pronóstico Médico*.

Mora, S. L. (2008). *Imagen digital Índice de contenidos Intro Plataformas y hardware*. 1–48.

Moujahid, A., Inza, I., & Larrañaga, P. (2017). *Clasificadores K-NN*. Retrieved from http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t9knn.pdf

Palma, A. (2013). *Guía para la identificación*.

Paradis, E., & Ahumada, J. A. (n.d.). R para Principiantes.

Quintero, O. C., & Ramirez, N. A. (2013). Aplicación de los índices de calidad de agua NSF, DINIUS y BMWP en la quebrada La Ayurá, Antioquia, Colombia. *Gestión y Ambiente*, *16*(1), 97–107.

Ramírez, A., & Gutiérrez-Fonseca, P. E. (2014). Functional feeding groups of aquatic insect families in Latin America: A critical analysis and review of existing literature. *Revista de Biologia Tropical*, *62*(April), 155–167. https://doi.org/10.15517/rbt.v62i0.15785

Rodríguez, O. (2018). *Desarrollo de una aplicación de reconocimiento en imágenes utilizando Deep Learning con OpenCV*.

Roldán, G. (1996). *Guía para el estudio de los macroinvertebrados acuáticos del Departamento de Antioquia. Universidad de Antioquia, Fondo FEN, Medellín.*

Simeone, O. (2018). A Very Brief Introduction to Machine Learning with Applications to Communication Systems. In *IEEE Transactions on Cognitive Communications and Networking* (Vol. 4). https://doi.org/10.1109/TCCN.2018.2881442

Sánchez Tomás, A. (1993): "Sistemas expertos en Auditoría", Técnica Contable, nº 536-537, pp. 529-544.

Suárez Thelma, S. (2015). Macroinvertebrados acuáticos como indicadores biológicos de la calidad del agua en el Rio Gil González y tributarios más importantes, Rivas, Nicaragua. *Universidad y Ciencia*, *6*(9), 38–46. https://doi.org/10.5377/uyc.v6i9.1958

ANEXOS

Anexo 1: Variables “dummy” de las características extraídas de cada individuo de las familias de macroinvertebrados

Individuo	Clase	Orden	Labels	AlargTub	CilinSeg	Ovalado	ConvexAlargOval	Alargado	SegCorp	Npatas	CubPelos	Antenas	Alas	Longitud	Ancho	L/A
Chir1_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,29	0,677	0,093
Chir1_2	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	6,31	0,666	0,105
Chir2_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,48	0,555	0,074
Chir3_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,14	0,688	0,096
Chir3_2	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	6,88	0,699	0,102
Chir4_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,45	0,629	0,084
Chir5_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,27	0,610	0,084
Chir6_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,65	0,644	0,084
Chir7_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,52	0,610	0,081
Chir8_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,42	0,533	0,072
Chir9_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,82	0,555	0,071
Chir10_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	6,51	0,544	0,083
Chir11_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	6,07	0,677	0,112

Chir12_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	6,18	0,710	0,115
Chir13_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	6,20	0,544	0,088
Chir14_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	6,18	0,566	0,092
Chir15_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	6,51	0,577	0,089
Chir16_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,50	0,533	0,071
Chir17_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,44	0,766	0,103
Chir18_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	6,20	0,677	0,109
Chir19_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,44	0,644	0,086
Chir20_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	6,96	0,621	0,089
Chir21_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	5,15	0,566	0,110
Chir22_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,40	0,533	0,072
Chir23_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	5,74	0,433	0,075
Chir24_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	6,00	0,577	0,096
Chir25_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	6,46	0,667	0,103
Chir26_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,47	0,777	0,104
Chir27_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	5,50	0,544	0,099
Chir28_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	5,95	0,588	0,099

Chir29_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,21	0,577	0,080
Chir30_1	Insecta	Diptera	Chir	Si	No	No	No	No	12	4	Si	No	No	7,75	0,732	0,094
Olig1_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	14,135	2,230	0,158
Olig2_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	11,188	2,731	0,244
Olig3_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	12,665	2,304	0,182
Olig4_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	8,944	1,997	0,223
Olig5_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	13,503	2,366	0,175
Olig6_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	13,415	2,828	0,211
Olig7_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	14,533	2,839	0,195
Olig8_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	8,530	2,429	0,285
Olig9_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	11,368	1,848	0,163
Olig10_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	10,041	1,976	0,197
Olig11_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	11,569	1,444	0,125
Olig12_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	8,379	1,525	0,182
Olig13_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	7,331	1,536	0,209
Olig14_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	12,415	2,884	0,232
Olig15_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	8,917	1,634	0,183

Olig16_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	8,482	1,573	0,185
Olig17_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	12,890	1,508	0,117
Olig18_1		Olig	Lumbriculida	No	Si	No	No	No	3	0	No	No	No	8,033	1,405	0,175
Elmi1_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	1,863	0,766	0,411
Elmi2_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,857	0,899	0,315
Elmi3_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,485	0,899	0,362
Elmi4_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,729	1,010	0,370
Elmi5_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,213	0,782	0,353
Elmi6_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,968	1,060	0,357
Elmi7_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	1,836	0,838	0,456
Elmi8_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,091	0,832	0,398
Elmi9_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,241	0,871	0,389
Elmi10_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,524	0,926	0,367
Elmi11_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,363	0,932	0,394
Elmi12_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,069	0,876	0,424
Elmi13_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	1,892	0,782	0,413
Elmi14_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,102	0,870	0,414

Elmi15_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	1,926	0,841	0,437
Elmi16_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,147	0,693	0,323
Elmi17_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,086	0,876	0,420
Elmi18_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	1,814	0,721	0,398
Elmi19_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	1,875	0,666	0,355
Elmi20_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,757	0,932	0,338
Elmi21_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,234	0,804	0,360
Elmi22_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	1,220	0,521	0,427
Elmi23_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,064	0,688	0,333
Elmi24_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	1,636	0,632	0,386
Elmi25_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	1,625	0,716	0,440
Elmi26_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	1,903	0,743	0,391
Elmi27_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,202	0,971	0,441
Elmi28_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,080	0,904	0,435
Elmi29_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	2,696	1,076	0,399
Elmi30_1	Insecta		Coleoptera	No	Si	Si	No	No	2	6	No	Si	No	1,819	0,832	0,457
Ptil1_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	6,357	1,720	0,271

Ptil2_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	7,100	1,564	0,220
Ptil3_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	6,146	1,564	0,255
Ptil4_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	7,681	1,544	0,201
Ptil5_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	6,102	1,553	0,255
Ptil6_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	12,769	1,675	0,131
Ptil7_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	8,814	1,996	0,226
Ptil8_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	7,775	1,599	0,206
Ptil9_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	9,315	1,351	0,145
Ptil10_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	9,495	1,432	0,151
Ptil11_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	9,489	1,840	0,194
Ptil12_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	7,932	1,553	0,196
Ptil13_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	7,215	1,828	0,253
Ptil14_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	14,103	1,950	0,138
Ptil15_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	14,582	1,972	0,135
Ptil16_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	7,921	1,764	0,223
Ptil17_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	9,659	1,866	0,193
Ptil18_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	10,195	1,808	0,177

Ptil19_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	11,597	2,721	0,235
Ptil20_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	11,300	1,989	0,176
Ptil21_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	11,906	1,820	0,153
Ptil22_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	9,674	1,738	0,180
Ptil23_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	8,416	1,753	0,208
Ptil24_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	11,259	1,602	0,142
Ptil25_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	14,236	1,925	0,135
Ptil26_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	10,732	1,881	0,175
Ptil27_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	9,142	1,590	0,174
Ptil28_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	12,394	1,524	0,123
Ptil29_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	13,449	1,522	0,113
Ptil30_1	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	11,384	1,436	0,126
Ptil30_2	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	9,562	1,326	0,139
Ptil30_3	Insecta		Coleoptera	No	Si	No	Si	No	4a11	6	Si	Si	No	9,064	1,111	0,123
Perl1_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	6,268	2,108	0,336
Perl2_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	5,225	1,797	0,344
Perl3_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	4,959	1,520	0,306

Perl4_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	5,847	1,753	0,300
Perl5_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	5,259	1,697	0,323
Perl6_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	5,003	1,520	0,304
Perl7_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	4,815	1,609	0,334
Perl8_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	5,702	1,542	0,270
Perl9_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	5,292	1,642	0,310
Perl10_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	3,883	1,254	0,323
Perl11_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	4,138	1,320	0,319
Perl12_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	4,537	1,431	0,315
Perl13_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	5,835	1,631	0,279
Perl14_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	4,915	1,531	0,312
Perl15_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	5,292	1,586	0,300
Perl16_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	5,835	1,753	0,300
Perl17_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	5,370	1,642	0,306
Perl18_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	4,870	1,620	0,333
Perl19_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	4,981	1,442	0,290
Perl20_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	5,259	1,897	0,361

Perl21_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	4,881	1,686	0,345
Perl22_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	5,769	2,186	0,379
Perl23_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	6,412	1,831	0,285
Perl24_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	5,159	2,019	0,391
Perl25_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	6,701	2,064	0,308
Perl26_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	4,937	1,642	0,333
Perl27_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	6,068	1,709	0,282
Perl28_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	3,162	1,609	0,509
Perl29_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	5,747	1,709	0,297
Perl30_1	Insecta		Plecoptera	No	Si	No	No	Si	3	6	Si	Si	No	6,335	2,008	0,317
Blep1_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	4,323	1,459	0,338
Blep2_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	4,108	1,475	0,359
Blep3_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	4,316	1,703	0,395
Blep4_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,301	1,326	0,402
Blep5_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,167	1,126	0,356
Blep6_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,722	1,641	0,441
Blep7_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	2,696	1,098	0,407

Blep8_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	1,079	1,387	1,286
Blep9_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	4,177	1,603	0,384
Blep10_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	4,182	1,686	0,403
Blep11_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,894	1,498	0,385
Blep12_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	4,238	1,747	0,412
Blep13_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,661	1,953	0,533
Blep14_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,201	1,525	0,477
Blepl15_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	4,367	1,525	0,349
Blep16_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,833	1,731	0,452
Blep17_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,955	1,975	0,499
Blep18_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,167	1,054	0,333
Blep19_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,694	1,548	0,419
Blep20_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	4,282	1,653	0,386
Blep21_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,872	1,548	0,400
Blep22_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	2,435	0,871	0,358
Blep23_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,567	1,659	0,465
Blep24_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,705	1,243	0,335

Blep25_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,151	1,137	0,361
Blep26_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,822	1,464	0,383
Blep27_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,628	1,442	0,398
Blep28_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	2,441	0,871	0,357
Blep29_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	2,108	0,971	0,461
Blep30_1	Insecta		Diptera	No	Si	No	No	Si	6	12	No	No	No	3,822	1,564	0,409

PERMISO DEL AUTOR DE TESIS PARA SUBIR AL REPOSITORIO INSTITUCIONAL

Yo, **Paola Soledad Castro Calle** portadora de la cédula de ciudadanía Nº 0150090447. En calidad de autora y titular de los derechos patrimoniales del trabajo de titulación **"Diseño de un algoritmo computacional de identificación de macroinvertebrados basados en parámetros de taxonomía"** de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos, Así mismo; autorizo a la Universidad para que realice la publicación de éste trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Cuenca, 17 de marzo de 2021

Paola Soledad Castro Calle
C.I. 0150090447