

**USS | UNIVERSIDAD  
SEÑOR DE SIPÁN**  
**FACULTAD DE INGENIERÍA, ARQUITECTURA Y  
URBANISMO**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**TESIS**

**COMPARACIÓN DE ALGORITMOS DE  
APRENDIZAJE SUPERVISADO PARA LA  
CLASIFICACIÓN AUTOMÁTICA DE MADERA  
ASERRADA MEDIANTE PROCESAMIENTO DE  
IMÁGENES DIGITALES.**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO  
DE SISTEMAS**

**Autor:**

**Bach. Mio Rojas Pool Cristopher**

**ORCID: <https://orcid.org/0000-0002-9800-5445>**

**Asesor:**

**Mg. Mejia Cabrera Heber Ivan**

**ORCID: <https://orcid.org/0000-0002-0007-0928>**

**Línea de Investigación:**

**Infraestructura, Tecnología y Medio Ambiente**

**Pimentel – Perú 2022**

**APROBACIÓN DEL JURADO**

**COMPARACIÓN DE ALGORITMOS DE  
APRENDIZAJE SUPERVISADO PARA LA  
CLASIFICACIÓN AUTOMÁTICA DE MADERA  
ASERRADA MEDIANTE PROCESAMIENTO DE  
IMÁGENES DIGITALES.**

---

**Bach. Mio Rojas Pool Cristopher**

**Autor**

---

**Mg. Mejía Cabrera Heber Ivan**

**Asesor**

---

**Mg. Tuesta Monteza Victor Alexci**

**Presidente de Jurado**

---

**Mg. Mejía Cabrera Heber Ivan**

**Secretario de Jurado**

---

**Mg. Bravo Ruiz Jaime Arturo**

**Vocal de Jurado**

## **Dedicatorias**

A mis padres que han sabido formarme con buenos sentimientos, hábitos y valores, porque ellos siempre estuvieron a mi lado brindándome su apoyo incondicional en la parte moral y económica para poder llegar a ser un profesional, asimismo con sus consejos para hacer de mí una mejor persona.

A mis hermanas por el apoyo que siempre me brindaron en el transcurso de mi carrera universitaria.

## **Agradecimientos**

Mi gratitud y agradecimiento sincero a mi asesor de tesis y a cada docente, por su apoyo y enseñanzas integran la base en mi trayecto universitario y han formado parte de mi desarrollo profesional.

## **Resumen**

La certeza referente a la tala de manera ilegítima perseverante, el lavado de madera ilícita, la depravación sistemática y las exportaciones peruanas de madera ilegal es algo agobiante, lo cual abarca la erradicación de una forma ilícita de leño tanto como especies maderables que no se encuentran autorizadas, modificación y mercantilizar de leño de procedencia prohibida.

En la actualidad el proceso para la identificación de madera es una actividad de suma importancia para las ciencias e industrias madereras, lo cual implica tanto a las propiedades orgánicas como propiedades físicas del leño, por lo tanto, el proceso de identificación se enfoca de manera exclusiva en la capacidad humana, que se basan en las diferentes características de la madera, como el color, la estructura y la textura.

Debido a ello en esta investigación nos motiva a desarrollar una comparación de los algoritmos para una mejor clasificación de especies de madera, debido que la clasificación de especies o reconocimiento de patrones es un tema que va mejorando al pasar del tiempo, para dicho proceso se realizó el proceso de adquisición de las imágenes el cual sigue un protocolo estructurado y teniendo una configuración adecuada de la cámara. Se construyó la base de datos con las imágenes obtenidas previamente para luego ser filtradas y obtener características importantes mediante el Filtro de Gabor. Estas características que se obtienen de cada imagen las cuales sirven como alimentación para los clasificadores supervisados, mediante su entrenamiento. Existen diversos algoritmos de clasificación, para ello se realizó una evaluación de los mismos para determinar cuál es el mejor algoritmo frente al problema enfrentado. Se evaluaron algoritmos como SVM, KNN y la red neuronal VGG16. Posterior a dicha evaluación se determinó que la red neuronal VGG16 es mejor en precisión con un 98.33%, exhaustividad con un 98.00%, pero con un mayor tiempo de respuesta de 250.3s, mayor porcentaje en el consumo de recursos frente a los algoritmos SVM y KNN.

### **Palabras Clave:**

Clasificación de imágenes, madera, especies, textura, aprendizaje supervisado, comparación, algoritmos.

## **Abstract**

The certainty of persistent illegitimate logging, illegal timber laundering, systematic depravation and Peruvian exports of illegal timber is overwhelming, which encompasses the eradication of an illicit form of timber as well as unauthorized timber species, modification and commercialization of illegally sourced timber.

Currently, the process of wood identification is a very important activity for the wood sciences and industries, which involves both the organic and physical properties of the wood, therefore the identification process is focused exclusively on human capacity, based on the different characteristics of the wood, such as color, structure and texture.

Because of this in this research we are motivated to develop a comparison of algorithms for better classification of wood species, because the classification of species or pattern recognition is a subject that is improving over time, for this process was performed the process of acquiring images which follows a structured protocol and having a proper configuration of the camera. The database was built with the previously obtained images and then filtered to obtain important characteristics through the Gabor Filter. These features are obtained from each image which serve as feed for the supervised classifiers, through their training. There are several classification algorithms, so an evaluation of them was carried out to determine which is the best algorithm for the problem faced. Algorithms such as SVM, KNN and the neural network VGG16 were evaluated. After this evaluation it was determined that the neural network VGG16 is better in accuracy with 98.33%, completeness with 98.00%, but with a higher response time of 250.3s, higher percentage in resource consumption compared to the SVM and KNN algorithms.

## **Keywords:**

Image classification, wood, species, texture, supervised learning, comparison, algorithms.

## Índice

<b>I. INTRODUCCIÓN</b> .....	9
<b>1.1. Realidad Problemática.</b> .....	9
<b>1.2. Trabajos previos.</b> .....	10
<b>1.3. Teorías relacionadas al tema.</b> .....	16
<b>1.4. Formulación del Problema.</b> .....	45
<b>1.5. Justificación e importancia del estudio.</b> .....	45
<b>1.6. Hipótesis.</b> .....	45
<b>1.7. Objetivos.</b> .....	45
<b>1.7.1. Objetivo general.</b> .....	45
<b>1.7.2. Objetivos específicos.</b> .....	45
<b>II. MATERIAL Y MÉTODO</b> .....	46
<b>2.1. Tipo y Diseño de Investigación.</b> .....	46
<b>2.2. Población y muestra.</b> .....	46
<b>2.3. Variables, Operacionalización.</b> .....	46
<b>2.3.1. Variable Independiente.</b> .....	46
<b>2.3.2. Variable Dependiente.</b> .....	46
<b>2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.</b> .....	49
<b>2.5. Procedimiento de análisis de datos.</b> .....	49
<b>2.6. Criterios éticos.</b> .....	51
<b>2.7. Criterios de Rigor Científico.</b> .....	51
<b>III. RESULTADOS.</b> .....	52
<b>3.1. Resultados en Tablas y Figuras.</b> .....	52
<b>3.2. Discusión de resultados.</b> .....	64
<b>3.3. Aporte práctico.</b> .....	66
<b>IV. CONCLUSIONES Y RECOMENDACIONES</b> .....	95

<b>4.1. Conclusiones.....</b>	<b>95</b>
<b>4.2. Recomendaciones.....</b>	<b>95</b>
REFERENCIAS.....	97
ANEXOS. ....	104

## I. INTRODUCCIÓN

### 1.1. Realidad Problemática.

La certeza referente a la tala de manera ilegítima perseverante, el lavado de madera ilícita, la depravación sistemática y las exportaciones peruanas de madera ilegal es algo agobiante. Es cierto que Estados Unidos emprendió a tomar acciones con contrariedad ante la comercialización de madera ilícita de Perú, no obstante otros importadores de gran importancia como lo son China y México siguen realizando esta actividad como si nada, señala EIA (Evaluación del Impacto Ambiental).(Urrunaga et al., 2018), según un reporte de un diario de circulación nacional, informa que la Dirección de Investigación de delitos de Alta Complejidad de la Policía Nacional (DIVIAC-PNP) descubrió e incautó presunto cargamento ilegal de madera, donde se presentaron guías de transporte forestal para la extracción de las especies Huayruro y Cachimbo, mas no para el cargamento en cuestión.(El Comercio, 2018) , la Ley Forestal y de Fauna Silvestre N° 29763, sanciona con multas superiores a 10 Unidades Impositivas Tributarias (UIT) equivalente a 40,500 soles, por el transporte de especímenes, artículos o sub artículos forestales, sin contar con los documentos que respalden dicha movilización.

Al adquirir madera de origen desconocido estamos contribuyendo con el deterioro ambiental de los bosques(Trujillo & Heinrichs, 2017), donde el Banco Mundial menciona que el 80 % de madera exportada del Perú proviene de tala ilegal. (Expreso, 2018), donde dicho término se define como «todo aprovechamiento(termino original: extracción) maderero que viola leyes nacionales e internacionales relevantes»(Brown et al., 2010), lo cual abarca la erradicación de una forma ilícita de leño tanto como especies maderables que no se encuentran autorizadas, erradicación de leño en zonas definidas como no autorizadas así mismo el traslado , modificación y mercantilizar de leño de procedencia prohibida.

No obstante, el proceso de reconocimiento de madera es una tarea muy importante para las ciencias e industrias madereras, ya que conlleva a la identificación de las propiedades orgánicas y propiedades físicas del leño, por

ende, el proceso de reconocimiento se basa casi exclusivamente en la capacidad de los humanos, que se enfocan en diversas características de la madera, como el color, la estructura y la textura. Sin embargo, existen numerosos tipos de especies de madera en la naturaleza que son difíciles de identificar incluso por científicos experimentados (Barmoutis et al., 2018)

No ha habido mucho desarrollo en el sistema automático de clasificación de madera debido a la gran dificultad para obtener una amplia gama de bases de datos de madera, falta de disponibilidad de técnicas probadas para la clasificación de la madera (Sundaram et al., 2015), lo cual nos motiva a desarrollar una comparación de los algoritmos para una mejor clasificación en especies de madera.

Así mismo la carencia de características discriminantes de las imágenes de textura es considerada uno de los principales inconvenientes dentro de los sistemas de detección de clases de madera (Yusof et al., 2010)

## **1.2. Trabajos previos.**

Kar Seng & Guniawan, (2018), realizó la investigación, An Experimental Study on the Use of Visual Texture for Wood Identification Using a Novel Convolutional Neural Network Layer, en Penang, Malaysia. Requerimiento de tecnología de identificación de la madera, que sea de manera fácil y rápida que sirva como medio para combatir la tala ilegal. Por esta razón, describen un enfoque para el desarrollo de una nueva capa de entrada para una red neuronal convolucional profunda (CNN) la cual no requiere del proceso de extracción de características de textura como GLCM. El modelo propuesto con 3 canales obtiene el 82.26% de precisión en el entrenamiento y el 85.12% de precisión en las pruebas, y los resultados con 6 canales son del 94.20% para la precisión de entrenamiento y el 93.94% para la precisión de pruebas. El enfoque propuesto es factible para la captura de la distribución estadística de los valores relacionales de los píxeles.

Nurthohari et al., (2019), realizó la investigación, Wood Quality Classification Based on Texture and Fiber Pattern Recognition using HOG Feature and SVM

Classifier, en Bali, Indonesia. La tecnología para el reconocimiento de la madera se lleva a cabo de manera tradicional, lo cual influye en la clasificación de manera correcta que solo corresponde al 55%. Por esta razón, se realiza la metodología la cual la madera será reconocida empleando el algoritmo HOG (Histograma de Gradiente Orientado) para reconocer su textura y patrones y para el método de clasificación se empleará el algoritmo SVM (Support Vector Machine) que realiza la comparación para hallar la mejor precisión y menor tiempo. El resultado de la clasificación mostró una tasa de rendimiento del 90% con un cálculo de tiempo de 1.40 empleando los algoritmos HOG y SVM. La metodología empleada para la clasificación es viable para posteriores trabajos de clasificación de manera para obtener buenos resultados en tanto a precisión y tiempo.

Asmara et al., (2017), realizó la investigación, Tropical hardwood species identification based on first order statistical moment of cross section images color and texture extraction using gray level co-occurrence matrix, en Surabaya. El proceso de reconocimiento de especies es susceptible, puesto que se basan en la peculiar forma de los expertos y de la falta de experiencia en dicho campo. Por esta razón, realizan un método para la identificación de las especies de madera dura, dicha identificación se enfoca en características de color y de textura para lo cual se aplicó la Matriz de Co-Ocurrencia de Niveles de Gris y para la clasificación se utilizó una red neuronal de retro propagación multicapa. El resultado del desarrollo del método propuesto logró un 90% de precisión. De acuerdo con los resultados nos evidencia que las técnicas empleadas son factibles para el desarrollo de un sistema de detección de maderas duras.

Wang et al., (2020), realizó la investigación, Research on Wood Panel Recognition Based on Digital Visual Features, en Guangzhou, China. El tiempo y el nivel de precisión en el control visual de forma manual por parte de los humanos. Por esta razón, proponen un método el cual se basa en el desarrollo de un histograma de colores y una matriz GLCM para la digitalización de las características ópticas de un tablón de madera y se aplica el clasificador de red neuronal de BackPropagation. El resultado obtenido por parte del clasificador

alcanza un porcentaje del 95.25% para el reconocimiento. El método propuesto demuestra un nivel de factibilidad buena para el reconocimiento de madera.

Sukrisdyanto et al., (2019), realizó la investigación, Wood Strength Classification Based on RGB Color and Image Texture Using KNN Method, en Surabaya, Indonesia. El alto costo para las pruebas de laboratorio para la realización del reconocimiento de la resistencia de una madera y por los métodos alternativos basados en la visión humana. Por esta razón, propone la realización de un método de procesamiento de imágenes que se basa en el color y la textura de una madera, se utilizó el histograma RGB para los canales de cada color, matriz GLCM (de co-ocurrencia de niveles de gris ) para la extracción de características, posteriormente se realizó la selección de las mismas con el método de RFECV(Validación Cruzada de Eliminación de Características Recursivas), y posteriormente se aplicaron 5 clasificadores: K-Nearest Neighbors(KNN), máquina de vectores de apoyo(SVM),Linear Discriminant lineal (LDA),bosque aleatorio(RF) y árbol de decisión(CART), evaluando sus resultados. Los mejores resultados se obtuvieron con el clasificador KNN, obteniendo un 95% de precisión. El método propuesto es realizable para la determinación de resistencia en una madera.

Liu et al., (2020), realizó la investigación, Real-time Classification of Rubber Wood Boards Using an SSR-based CNN, en China. El reconocimiento de las maderas es realizada por los típicos métodos realizados por expertos basados en la extraer características con descriptores caseros. Por esta razón, ejecutan el desarrollo de una red neuronal (CNN) basada en splith-shuffle-residual (SSR), donde se realiza la introducción de un módulo SSR lo cual reduce el tiempo computacional y obtener un nivel alto de precisión para la clasificación. El algoritmo propuesto indica un valor del 94.86% de precisión para el reconocimiento de maderas. El algoritmo propuesto se puede emplear para el reconocimiento de maderas debido al alto nivel de precisión y su tiempo de ejecución de 26, 55ms por imagen.

Barmpoutis et al., (2018), realizó la investigación, Wood species recognition through multidimensional texture analysis, en Grecia. La variedad de especies que no pueden ser reconocidas e identificadas fácilmente por los expertos. Por esta razón, proponen un enfoque nuevo para el reconocimiento de maderas basado en un proceso de análisis de manera multidimensional de su textura, en donde se implementa un descriptor espacial que toma a una imagen como un conjunto de señales multidimensionales, para su clasificación se emplea el algoritmo SVM (máquinas de vectores de soporte). El enfoque propuesto demuestra un potencial de buen nivel, donde se obtiene un 91.47% para la tasa de clasificación. El enfoque propuesto presenta un buen nivel para la clasificación de madera de manera automatizada.

Wang et al., (2019), realizó la investigación, Research on Wood Species Recognition Method Based on Hyperspectral Image Texture Features, en Hohhot, China. Integración del espectro y de la imagen, para poder aumentar la precisión en la clasificación de especies de madera. Por esta razón, proponen un nuevo método para la clasificación de especies de madera basada en las características de su textura obtenido con la matriz de co-ocurrencia y una reducción de dimensionalidad con el algoritmo PCA, para realizar la clasificación emplearon el algoritmo PNN (Probabilistic Neural Networks). El resultado obtenido con el algoritmo PNN (red neuronal probabilística) muestra un 96.5% de tasa de reconocimiento. El método propuesto es viable para llevar a cabo el proceso de reconocimiento de maderas.

Widyarto et al., (2017), realizó la investigación, Wood texture detection with conjugate gradient neural network algorithm, en Yogyakarta, Indonesia. La falta de mejora en la precisión de los sistemas de reconocimiento de madera con redes neuronales. Por esta razón, ejecutan la realización de pruebas combinando el método de propagación hacia atrás con la red neuronal artificial (ANN) para la clasificación. El resultado del experimento demuestra un 96% de precisión y es mucho mejor que trabajar solo con la red neuronal artificial (ANN). El método experimentado cuenta con buenos resultados por lo tanto es factible para la aplicarlo en el proceso de reconocimiento de maderas.

Taqyudin et al., (2019), realizó la investigación, Wood Classification Based on Fiber Texture Using Backpropagation Method, en Bandung, Indonesia. Los costos y el alto tiempo de demora para el reconocimiento de maderas debido al reconocimiento de manera visual manual de un experto. Por esta razón, elaboran el diseño de un sistema para el proceso de reconocimiento de las especies de las maderas, enfocándose en su textura, para lo cual emplearon dos métodos: HOG y Backpropagation. Los resultados que se obtuvieron con respecto a la tasa de precisión son del 90%. El sistema diseñado es realizable en el reconocimiento de clases de madera.

He et al., (2019), realizó la investigación, A Fully Convolutional Neural Network for Wood Defect Location and Identification, en China. El nivel de rendimiento en tiempo real es muy bajo y la dificultad para captar los defectos de la madera es muy alta. Por esta razón, proponen una nueva red neuronal que sea totalmente convolucional (Mix-FCN) para clasificar los defectos de la madera, partiendo del modelo VGG16. Los resultados obtenidos con la red neuronal propuesta demuestran un gran nivel de rendimiento, llegando a obtener un 99,14% de precisión. La red neuronal propuesta puede ser considerado como herramienta viable debido a su alto nivel para clasificar los defectos de la madera de forma automatizada

Ristiawanto et al., (2019), realizó la investigación, Wood Classification with Transfer Learning Method and Bottleneck Features, en Yogyakarta, Indonesia. Las empresas ejecutan la clasificación de madera de forma manual basados en la vista del humano logrando alcanzar solo un 55% como tasa correcta. Por esta razón, proponen un método enfocado en la red neuronal Convolucional Profunda (D-CNN), que se basa en características de cuello de botella y el aprendizaje por transferencia. El método empleado logra obtener un 95,69% de precisión como resultado de la clasificación. El método experimentado es viable para el reconocimiento de especies de madera.

Loke, (2018), realizó la investigación, Texture Recognition Using a Novel Input Layer for Deep Convolutional Neural Network, en Singapur. El desarrollo del

reconocimiento de texturas aún se lleva a cabo mediante la vista humana y esto conlleva a no tener una clasificación eficaz. Por esta razón, proponen un enfoque que incluye una nueva capa de entrada a la red neuronal convolucional, basándose en la arquitectura de la CNN para capturar propiedades relacionales de los píxeles. El resultado del enfoque propuesto alcanza un 83% de precisión. El enfoque propuesto es prometedor para el reconocimiento de diversas texturas.

Yang et al., (2019), realizó la investigación, Application of Deep Learning in Wood Classification, en Kunming, China. El proceso de reconocimiento se realiza de manera tradicional, basado en el juicio de los expertos y profesionales con experiencia en madera. Por esta razón, la metodología se basa en ejecutar un modelo de aprendizaje profundo con el modelo VGG19, el cual fue entrenado con el conjunto de datos de ImageNet para realizar la clasificación de 25 maderas. Los resultados del modelo obtuvieron un 93,63% de precisión. El método propuesto y realizado es muy viable para el proceso de clasificación de especies maderas.

Yusof et al., (2018), realizó la investigación, A Study of Feature Extraction and Classifier Methods for Tropical Wood Recognition System, en Jeju, Korea (South). La carencia de características que sean discriminantes entre especies de madera y así mismo no contar con un ambiente controlado para la mitigación de ruido y sombras. Por esta razón, la metodología propuesta se basa en la extracción de características mediante 3 técnicas (GLCM, BGLAM y SPPD), las mismas que serán combinadas con 3 clasificadores los cuales son LDA, BPNN y ResNet, para finalmente ser aplicadas a 3 dataset diferentes. Los resultados de realizar la combinación de BGLAM, SPPD, LDA y GA logró alcanzar un 100% de precisión con el dataset 1. La metodología propuesta es factible y muy viable para el proceso de reconocimiento de maderas

### **1.3. Teorías relacionadas al tema.**

#### **1.3.1. Madera.**

Se define madera como un material orgánico que tiene una textura fibrosa, heterogénea y anisotrópica producida a raíz del árbol, que es un organismo vivo (Protect Aqua, 2015), del mismo modo es un material orgánico de principio botánico el cual conforma un fragmento del tronco de los árboles y de su ocupación que se debe al transporte de agua y sustancias nutritivas de la superficie hacia las hojas, por lo tanto ofrece un sustento a las ramas que constituyen la copa y también asegura las sustancias de provisión de una manera que almacena los productos transformados en las hojas. (Pozzer & Guzowski, 2011).

##### **1.3.1.1. Características anatómicas.**

Se define Anatomía de una madera al estudio de sus características de manera general tanto como organolépticas y sub-microscópicas, esta a su vez se fracciona en dos partes, una de ellas es la Anatomía Sistemática que tiene por ocupación la identidad de la especie y por otro lado tenemos la Anatomía Aplicada la cual se basa en el estudio del dominio del sistema anatómico en base a sus propiedades tecnológicas, las propiedades que se pueden percibir por los distintos sentidos dentro de estas características encontramos a la dureza, grano, textura, color, olor peso, sabor, dureza, figura (Callo, 2014).

En tanto los anillos de desarrollo son capaces de poseer una alteración al pasado de un año esto depende de la acción de los factores externos así como la precipitación, humedad, disponibilidad de agua en el suelo, temperatura, luz, la permanencia del periodo vegetativo y la distancia del periodo de desarrollo (Callo, 2014).

##### **1.3.1.2. Características generales y organolépticas de la madera.**

Las características generales comparten relación con aspectos estéticos, estructura anatómica las cuales se le denominan características organolépticas de la madera las cuales son el color, veteado, textura (Callo, 2014).

#### **Brillo**

El brillo natural de una madera necesita de la disposición de la pared celular para así poder manifestar la luz (Callo, 2014).

## **Color**

El tono de un leño se determina por los componentes situados en la estructura celular o se adhiera en sus paredes, dichas sustancias son la goma, los taninos, el pigmento y las resinas. También nos dicen que el color o tono es una propiedad de mucha importancia para el reconocimiento de la madera asimismo desde un punto de vista atractivo, ya que el tono de la madera es debida a la impregnación de sustancias en la pared celular para así ocasionar la diversidad de coloración como por ejemplo: pardo, amarillo, negro, rosado, castaño, castaño oscuro, ocre y castaño rojizo(Callo, 2014).

## **Olor**

Indican la presencia de maderas las cuales cuentan con olores peculiares que proceden de las diversas sustancias alojadas al interior, dicho olor tiene mayor intensidad en cortes frescos, éste disminuye con el paso del tiempo. (Callo, 2014).

## **Sabor**

El sabor de un madero comparte relación con el olor, puesto que las mismas sustancias son responsables de ambos. (Callo, 2014).

## **Textura**

La textura de un madero es bastante particular por lo que se da debido a su tamaño , distribución y proporción concerniente a los componentes leñoso que poseen importancia para el terminado de la madera (Callo, 2014).

De acuerdo con los grados de uniformidad de la madera encontramos los siguientes tipos de textura:

**Madera de textura gruesa:** con un grosor de poros con una cantidad mayor a 250 micrómetros, un radio leñoso extenso y excesivo parénquima longitudinal. (Callo, 2014).

**Madera de textura media:** posee un grosor de poros desde 150 hasta 250 micrómetros con un radio leñoso extenso y excesivo parénquima longitudinal(Callo, 2014).

**Madera de textura fina:** posee un grosor de poros con una cantidad menor de 150 micrómetros, asimismo con un radio excesivamente fino y con parénquima longitudinal desprovisto(Callo, 2014).

## Veteado

Es una propiedad del madero provocada por el diseño de la franja que tiene como origen la superficie longitudinal pulimentada que se debe a la distribución de sus elementos componentes del leño fundamentalmente en el parénquima, los vasos, anillos de crecimiento y los radios medulares(Callo, 2014).

### 1.3.1.3. Características macroscópicas de la madera.

Callo, (2014) El parénquima es un tejido el cual asigna el transporte y depósito de sustancias de provisión, asimismo constituye parte del tejido longitudinal del tronco, ya que el parénquima es de suma importancia en el reconocimiento.

El parénquima se clasifica debido a su forma y disposición, tenemos:

Tabla 1.

*Clasificación del parénquima de acuerdo a su forma y disposición.*

<b>Forma apotraqueal</b>	<b>Forma paratraqueal</b>	<b>Forma marginal</b>
No se encuentran juntos a los poros.	Están aledaños de los poros.	Constituyen líneas anchas y angostas en el término de los anillos de desarrollo.

Fuente: Elaboración propia.

## Poros

Los poros se generan en el momento que un componente vascular es serrado de manera transversal, los mismos que pueden encontrarse de manera aislada o agrupada de una manera cambiante, para el primer caso se le designa poros solitarios y repetidamente es redondeado, en tanto el segundo caso se le designa poro múltiple radial esto hace referencia cuando diversos poros se encuentran próximos en una hilera radial. (Callo, 2014).

## Radios Medulares

Callo, (2014) Define a los radios medulares como líneas que van del interior hasta el exterior de un arbusto constituyendo el sistema transversal del tronco, por otro lado están conformados por células parénquimas de tal forma que los radios vienen a ser líneas o puntos débiles de madera porque mientras el

desarrollo de secado se propagan las grietas y fundamentalmente a raíz de los radios.

#### **1.3.1.4. Características microscópicas de la madera.**

Define a las propiedades microscópicas como la parte maderable de un arbusto, ésta a su vez cumple con tres funciones como el almacenamiento de sustancias de reserva, transporte del agua y la fortaleza (resistencia) mecánica.

De acuerdo a la función se distingue a la madera en tres tipos de tejidos:

Tabla 2.

*Clasificación de los tejidos de acuerdo a la función de características microscópicas.*

<b>Función de conducción</b>	<b>Función de almacenamiento</b>	<b>Función de resistencia</b>
Tejido vascular	Tejido parenquimático	Tejido fibroso

Fuente: Elaboración propia.

#### **1.3.1.5. Propiedades físicas.**

La conducta natural del leño está conformado por una secuencia de características como por ejemplo la densidad, humedad, pesos determinados que de una forma agrupada pueden definirse como características físicas del leño. (Callo, 2014).

#### **1.3.1.6. Densidad.**

La valoración de la consistencia de un leño y su variación necesita en gran medida a la altitud y sección del arbusto de donde se adquiere el muestrario , asimismo la consistencia se encuentra influenciada debido a la estructura genética del arbusto y esta a su vez varía de acuerdo al número y clases de componentes que posee como las ligninas y las resinas.(Callo, 2014).

#### **1.3.2. Madera Aserrada.**

La madera aserrada es madera que se corta de troncos en diferentes formas y tamaños, habitualmente se corta en distintos anchos y longitudes rectangulares, asimismo puede poseer forma de cuña. Los productos realizados en madera aserrada comunes incluyen vigas de madera maciza y más secciones de madera rectangulares(Wood Solutions, 2019).

### 1.3.3. Tipos de Madera.

El Perú disfruta de una diversidad muy amplia de especies forestales maderables en su estado, acción la cual que hace petición de un conocimiento diferente.

#### 1.3.3.1. Capinuri.

**Descripción:** Madera perteneciente a la especie *Naucleopsis ulei*( Warb.) Ducke, de familia denominada MORACEAE la cual su hábitat es en los ambientes pluviosos de forma perseverante, asimismo se caracteriza por ser una especie tolerante a las sombras, se encuentra vigente en el bosque de nivel primario, encima de suelos arenosos y con abundando lodo, fructosos, de sabor dulce. Su madera presenta un buen nivel de calidad, bastante grávido(OSINFOR, 2015).

**Usos:** Se utiliza para la carpintería y también en la elaboración de elementos estructurales de construcción(OSINFOR, 2015).



*Figura 1.* Imagen de la textura de la madera aserrada de especie Capinuri.

Fuente:(AITIM, 2016)

#### 1.3.3.2. Capirona.

**Descripción:** Madera diferenciada perteneciente a la especie *Calycophyllum spruceanum* de familia denominada RUBIACEAE que posee una albura blanca la cual se altera de forma progresiva a duramen de tono amarillento pálido, olor y sabor que no se diferencia, con una fina textura, un nivel medio de brillo, no presenta jaspeado, grano entrecruzado(Gonzales, 2011).

**Usos:** Es una madera pesada, dura, de un nivel alto en calidad y una textura sumamente fina la cual tiene una magnífica durabilidad asimismo es empleada ampliamente para construcciones rurales, igualmente es considerada como leña por el alto poder calorífico que posee (Reynel et al., 2003), del mismo modo es utilizada en carpintería, escultura, estructuras, manijas de herramientas, productos de deporte. (CEMU, 2017a).



*Figura 2.* Imagen de la textura de la madera aserrada de especie Capirona.

Fuente: (CEMU, 2017a)

### **1.3.3.3. Catahua.**

**Descripción:** Madera diferenciada perteneciente a la especie *Hura crepitans* L., de familia denominada EUPHORBIACEAE que cuenta con una albura blanca cremosa que varía de forma abrupta a duramen de tonalidad gris olivo, el sabor y olor no característico, textura de nivel medio, brillo de nivel medio a elevado, rayado arcos superpuestos, grano entrelazado (Gonzales, 2011).

**Usos:** los usos que se le asigna a esta madera son para el recubrimiento interior, mueblería barata y carpintería, contrachapado, cajonerías, empaquetado de productos efímeros, triplay en donde se necesita una madera ligera y de un manejo muy fácil para poder trabajar. (CEMU, 2017b).



*Figura 3.* Imagen de la textura de la madera aserrada de especie Catahua.

Fuente: (CEMU, 2017b)

#### **1.3.3.4. Cedro.**

**Descripción:** Madera diferenciada perteneciente a la familia MELIACEAE, que se ubica de manera distribuida en los departamentos de San Martín, Ucayali, Loreto, Madre de Dios y Huánuco, en tanto a su color de la madera recién cortada es de una pigmentación rosado a pardo rojizo que posteriormente se tornad de color rojo a pardo rojizo oscuro debido a la exposición del aire, su textura es nivel medio y presenta veteados en forma de arcos superpuestos y líneas verticales no muy bien pronunciados. (Confederación Peruana de la Madera, 2008)



*Figura 4.* Imagen de la textura de la madera aserrada de especie Cedro.

Fuente: (Perú Forestal, 2020a)

#### 1.3.3.5. Copaiba.

**Descripción:** Madera diferenciada perteneciente a la especie *Copaifera paupera* (Herzog) Dwyer, de familia denominada FABACEAE que posee una albura de tono marrón exageradamente pálida que cambia abruptamente a duramen de tono rojo amarillento, sabor y olor ausentado, textura a nivel medio, brillo de nivel medio a elevado, rayado arcos superpuestos y grano entrelazado(Gonzales, 2011).

**Usos:** Se usa para la realización de trabajos de ebanistería de interiores y en general, durmientes, cajonerías, torneados, empaquetados, parihuela , postes de cerco, cajas, molduras y maderamen o armazón.(CEMU, 2017c).



*Figura 5.* Imagen de la textura de la madera aserrada de especie Copaiba.

Fuente:(CEMU, 2017c)

#### 1.3.3.6. Cumala.

**Descripción:** Madera diferenciada perteneciente a la especie de *Otoba glycarpa*, de familia denominada MYRISTICACEAE, que posee una albura de tono marrón pálido que varía de forma manera gradual de sabor y olor ausente, a un duramen de tono marrón, textura media, brillo de nivel alto con un rayado satinado y de grano entrelazado (Gonzales, 2011).

**Usos:** esta madera posee una excelente calidad, es suave y ligera de grano recto y una textura de nivel medio, de tono amarillo rojizo cuando seca. (Reynel et al., 2003).



*Figura 6.* Imagen de la textura de la madera aserrada de especie Cumala.

Fuente: *(CEMU, 2017d)*

#### **1.3.3.7. Huayruro.**

**Descripción:** Es una madera perteneciente a la especie *Ormonia coccinea* Jacks, familia denominada FABACEAE, donde presenta las siguientes características, color de la albura (capas externas) de un color marrón de tonalidad clara a una tonalidad óxido y el duramen (capas internas) de un crema de tonalidad oscura a un rosado de tonalidad pálido, de una textura de nivel grueso con unos veteados en forma de arcos superpuestos con líneas verticales. *(Castillo, 2010)*



*Figura 7.* Imagen de la textura de la madera aserrada de especie Huayruro.

Fuente: *(Perú Forestal, 2020b)*

#### 1.3.3.8. Lupuna.

**Descripción:** Es una madera perteneciente a la especie de Ceiba pentrada, familia denominada MALVACEAE con una distribución muy amplia al interior de los bosques amazónicos, en el Perú se logra hallar en diferentes departamentos tales como: Ucayali (Pucallpa), San Martín (Tarapoto), Loreto (Yurimaguas) y en los distintos bosques donde esta especie es preponderante. (CEMU, 2017e).

**Usos:** De manera limitada para la realización de trabajos de ebanistería de forma general asimismo cuenta con una aptitud papelera de manera aceptable que viene a ser la pulpa y con triplay que vendría siendo el alma(CEMU, 2017e).



*Figura 8.* Imagen de la textura de la madera aserrada de especie Lupuna.

Fuente:(CEMU, 2017e)

#### 1.3.3.9. Moena.

**Descripción:** Es una madera perteneciente a la familia LAURACEAE, donde su corteza externa presenta las siguientes características: color de tonalidad marrón claro y es lenticelada y su corteza interna es de tonalidad amarillento que se oxida de manera inmediata a un tono marrón y es homogénea; tanto a su textura es de nivel medio a gruesa y dentro de los usos de esta especie tenemos: la carpintería y en la construcción.(Reynel et al., 2003)



*Figura 9.* Imagen de la textura de la madera aserrada de especie Moena.

Fuente: *(Perú Forestal, 2020c)*

#### **1.3.3.10. Tornillo.**

**Descripción:** Es una madera perteneciente a la especie *Cedrelinga cateniformis*, de familia denominada FABACEAE que posee un albor de tono marrón bastante pálido, verduzca que varía progresivamente a duramen de tono marrón blanquecino verduzco, su sabor y olor no se diferencian asimismo cuenta con un brillo nivel medio, textura gruesa y un grano entrelazado. (Gonzales, 2011).

**Usos:** Esta madera posee una gran durabilidad y calidad con un grano recto a entrelazado, es semi-dura y semi-pesada, textura gruesa y de tono blanquecino a rosado, de igual manera es una madera muy fácil de trabajar y cuenta con un extenso mercado dentro del Perú en trabajos basados en carpintería, construcciones. Dicha madera se distribuye con otro nombre : "Pino peruano" (Reynel et al., 2003).



*Figura 10.* Imagen de la textura de la madera aserrada de especie Tornillo.

Fuente: *(CEMU, 2017f)*

### 1.3.4. Clasificación de las maderas.

Los troncos se pueden catalogar de diversas maneras dependiendo las pautas a ocuparse, principalmente se usa el criterio de las propiedades que se encuentran en destino a la estructura que hace referencia a la textura.(Presa, 2014).

### 1.3.5. Imagen.

Se define imagen como una función de tipo bidimensional  $f(x,y)$  en el cual  $x$  e  $y$  viene siendo coordenadas espaciales es decir planas asimismo la extensión de  $f$  en cualquiera de los pares de coordenadas a lo cual se le denomina de gran magnitud o grado de gris de la imagen en dicho punto.(Gonzalez & Woods, 2009).

#### 1.3.5.1. Tipos de Imagen.

##### 1.3.5.1.1. TIFF.

Formato de archivo utilizado y flexible que fue diseñada para satisfacer necesidades de profesionales de variados campos, este formato es compatible con una amplia gama de escala de grises, indexado, color vivo así mismo los tipos de imágenes especiales con número entero a gran profundidad(Burger & Burge, 2016).

##### 1.3.5.1.2. GIF.

Archivo de imagen con formato de 8 bits que fue diseñada para color lo cual no le permite soportar imágenes de color real(Burger & Burge, 2016).

##### 1.3.5.1.3. PNG.

Formato diseñado como imagen general para ser usada en internet de una forma en particular, y es compatible con 3 tipos de diferentes imágenes que son imágenes en color verdadero(Burger & Burge, 2016).

Tabla 3.

*Compatibilidad Tipos de Imagen.*

<b>Imagen de color verdadero</b>	<b>Imagen en escala de grises</b>	<b>Imagen en color indexado</b>
Máximo de 3 x 16 bits/pixel	Con hasta 16 bits/pixel	Máximo de 256 colores

Fuente: Elaboración propia.

#### **1.3.5.1.4. JPEG.**

El objetivo de este formato es obtener el acortamiento del valor de la media de los datos de un factor de 1:16, este tipo de formato es el más frecuente debido a su uso(Burger & Burge, 2016).

#### **1.3.5.1.5. BMP.**

Formato de archivo simple que es muy utilizado en Windows el cual admite escala de grises, indexado y verdadero, como imágenes en color, presenta compatibilidad con imágenes de tipo binarias, pero no de una forma competente(Burger & Burge, 2016).

#### **1.3.5.1.6. PBM**

Este formato es ampliamente utilizado en Unix y admite los siguientes formatos: PBM (mapa de bits portátil) para mapas de bits binarios, PGM (mapa gris portátil) para imágenes en escala de grises y PNM (portátil cualquier mapa) para imágenes en color(Burger & Burge, 2016).

#### **1.3.6. Procesamiento de Imagen.**

Es la agrupación de prácticas que transforman las imágenes digitales para un incremento de perceptibilidad de algunas propiedades de las cosas actuales en una imagen, y posteriormente realizar un análisis o un incremento de visualización de dicha imagen de igual modo no incrementa la información la cual se pueda extraer de las imágenes, es decir se basa en el realce de dichas características de la misma.(Viera, 2017).

El procesamiento de las imágenes digitales puede realizarse a través de métodos visuales o por métodos digitales en un computador. (Mantilla, 2019).

##### **1.3.6.1. Filtro.**

La técnica de aplicar filtro a una determinada imagen( $f$ ), se establece en la aplicación de una variación( $T$ ) para tener como objetivo una imagen vigente( $g$ ) por tanto determinadas propiedades sean resaltadas o mitigadas:

$$g(x, y) = T[f(x, y)]$$

### 1.3.6.1.1. Técnicas de filtrado.

El filtrado es un método para mejorar o alterar una imagen(Araki et al., 2015), existen principalmente dos tipos de filtrado:

- Filtrado de tipo Espacial.
- Filtrado de tipo frecuencia.

#### 1.3.6.1.1.1. Filtro espacial.

En el proceso de filtrado espacial el valor del píxel ya procesado para el píxel evidente necesita de sí mismo y de los píxeles vecinos(Gonzalez & Woods, 2009), de modo que el filtrado espacial es un proceso vecino, en donde el valor de cualquier píxel en específico en la salida de imagen se calcula a través del uso de determinado algoritmos con los valores de cada píxel cercano del debido píxel que funciona como entrada, la cercanía de un pixel se determina por un grupo de pixeles circundantes en vínculo con ese pixel(Chaki & Dey, 2018).

Algunos tipos de filtrado espacial se discuten a continuación.

#### 1.3.6.1.1.1.1. Filtro Lineal (convolución).

El producto del procedimiento de filtrado lineal es la adición del proceso de multiplicar los coeficientes de máscara con los pixeles correspondientes de manera exacta debajo de la máscara, como se muestra en la Figura 11.(Chaki & Dey, 2018)

El filtro lineal se puede expresar por la siguiente ecuación:

$$\begin{aligned} I(x, y) = & [M(-1,1) * I(X - 1, Y + 1)] \\ & + [M(0,1) * I(x, y + 1)] + [M(1,1) * I(x + 1, y + 1)] \\ & + [M(-1,0) * I(x - 1, y)] + [M(0,0) * I(x, y)] \\ & + [M(1,0) * I(x + 1, y)] + [M(-1, -1) * I(x - 1, y - 1)] \\ & + [M(0, -1) * I(x, y - 1)] + [M(1, -1) * I(x + 1, y - 1)] \end{aligned}$$

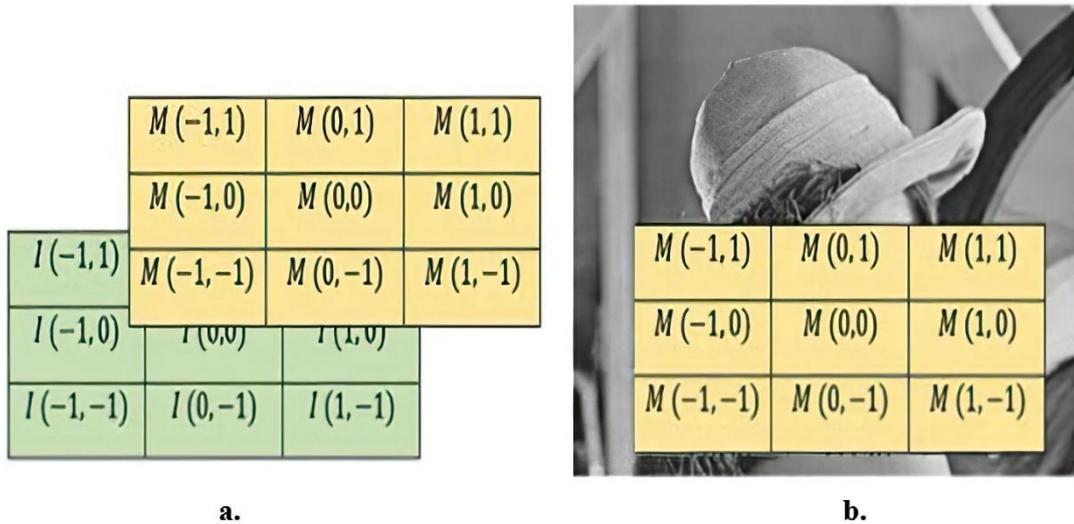


Figura 11. Ecuación con la que se puede expresar el filtro lineal. a. I:posiciones de la imagen de píxeles y M:coeficiente de la máscara. b. máscara de píxeles de la imagen. Fuente: (Chaki & Dey, 2018).

El coeficiente de máscara  $M(0, 0)$  se solapa con el valor de píxel de imagen  $I(x, y)$ , que simboliza que el centro máscara se ubica en  $(x, y)$  en el momento que el cálculo de la suma de productos ocurre, no obstante para una máscara de dimensión  $p \times q$ ,  $q$  y  $p$  son números impares y simboliza a  $p = 2m + 1$ ,  $q = 2n + 1$  en el cual  $m$  y  $n$  no son negativos enteros, en donde el filtrado lineal de una imagen ( $I$ ) de tamaño  $p \times q$ , con una máscara de filtro de tamaño  $p \times q$  se torna dada por la siguiente ecuación(Chaki & Dey, 2018):

$$LF(x, y) = \sum_{a=-m}^m \sum_{b=-n}^n M(a, b) * I(x + a, y + b).$$

### 1.3.6.1.1.1.2. Filtro no lineal.

El filtrado espacial no lineal de igual manera se desempeña en los barrios, como se analiza en el caso del filtrado lineal, la única desigualdad es que el filtrado no lineal se fundamenta condicionalmente en los valores de los píxeles cercanos de un píxel relativo(Chaki & Dey, 2018).

### 1.3.6.1.1.1.3. Filtro de nitidez.

El propósito primordial de dicho filtro se basa en el enriquecimiento de los elementos más refinados, los mismos que están comprendidos en alguna imagen, asimismo realizar la predominación de la parte más imprecisa, todo esto puede llevarse a cabo empleando derivadas de tipo espacial, estas a su vez podrían ser suministradas en zonas llanas o que tengan un grado persistente de gris.(Chaki & Dey, 2018),se pueden expresar empleando la siguiente ecuación:

$$\frac{\partial I}{\partial x} = I(x + 1, y) - I(x, y) \text{ y } \frac{\partial I}{\partial y} = I(x, y + 1) - I(x, y)$$



**Imagen Original**

a.



**Imagen afilada**

b.

*Figura 12.* Técnica de Filtro de nitidez a través del afilado de imágenes. a. Imagen Original. b. Imagen afilada. Fuente: (Chaki & Dey, 2018)

### **Frecuencia de filtro**

Los filtros de frecuencia son empleados para el tratado de una imagen en el dominio de la frecuencia en donde hay una modificación de la imagen al dominio de la frecuencia a través del uso de una función de transformación de Fourier, posteriormente del procedimiento dominio de la frecuencia la imagen retorna a convertirse en el dominio espacial por la transformada de Fourier inversa, la disminución de las frecuencias altas en los convertidos dominio espacial de la imagen en uno con mayor suavidad, entretanto que la reducción de bajas frecuencias más destacado de los bordes de la imagen; en su totalidad los filtros

de frecuencia igualmente se pueden desarrollar en el dominio espacial, asimismo los filtros de frecuencias no son computacionalmente caro para llevar a cabo el filtrado en el dominio espacial, de igual manera es más apropiada si es que no hay un centro que puede ser directamente creado en el dominio espacial, por lo cual de igual manera pueden ser más eficientes; todas las imágenes espaciales poseen una representación de frecuencia equivalente, los de alta frecuencia pertenecen a los valores de pixeles que cambian de manera rápida a través de la imagen como el texto, las hojas, la textura y los de baja frecuencia corresponde con la parte homogénea de la imagen.(Chaki & Dey, 2018)

El Filtrado de frecuencia se fundamenta en la transformada de Fourier, donde el operador toma de forma general una función de filtro y una imagen en el dominio de Fourier, además esta imagen se multiplica por lo tanto de una manera píxel a píxel con la función de filtro y se puede expresar por la siguiente ecuación:

$$F(u, v) = \frac{1}{PQ} \sum_{x=0}^{P-1} \sum_{y=0}^{Q-1} I(x, y) e^{-j2\pi(\frac{ux}{p} + \frac{vy}{Q})}$$

Aquí  $I(x, y)$  es la imagen de entrada de dimensión  $P \times Q$  en el dominio de Fourier y  $F(u, v)$  es la imagen filtrada [ $u = 0, \dots, P - 1$  y  $v = 0, \dots, Q - 1$ ]. Para transformar,  $F(u, v)$  se vuelve a transformar mediante el uso de la transformada de Fourier inversa, como se visualiza en la siguiente ecuación:

$$I(x, y) = \frac{1}{PQ} \sum_{x=0}^{P-1} \sum_{y=0}^{Q-1} F(u, v) e^{-j2\pi(\frac{ux}{p} + \frac{vy}{Q})}$$

De manera que la multiplicación en el espacio de Fourier es semejante a la convolución en el dominio espacial, en su totalidad los filtros de frecuencia se pueden implementar en teoría como un filtro espacial, de manera que los diferentes tipos de filtros de frecuencia se discuten en las siguientes subsecciones.

### **Filtro de paso bajo**

Hace referencia al filtro que recorre o permite que las señales de baja frecuencia y las señales de Suprime con frecuencias más altas que el punto de corte o

umbral de frecuencia basado en el diseño del filtro específico, la cantidad real de supresión varía para cada frecuencia. Un filtro de paso bajo se utiliza generalmente para suavizar una imagen, las formas estándar de filtros de paso bajo son ideales, Butterworth, y filtros de paso bajo gaussianos (Chaki & Dey, 2018).

### **Filtro ideal de paso bajo (ILP)**

Este es el filtro de paso bajo simple que suprime todos los elementos que cuentan con un nivel alto de frecuencia con respecto a la transformada de Fourier que son mayores que un especificado  $F_0$  frecuencia de corte. Esta función de transferencia del filtro puede definirse por la ecuación (Chaki & Dey, 2018):

$$P(u, v) = \begin{cases} 1 & \text{si } F(u, v) \leq F_0 \\ 0 & \text{si } F(u, v) > F_0 \end{cases}$$

### **Filtro Butterworth de paso bajo (BLP)**

Este filtro se utiliza para eliminar el ruido de alta frecuencia con la menor pérdida de datos de imagen en la banda de paso especificado con el fin de la función de transferencia de orden  $d$  y con  $F_0$  frecuencia de corte (Chaki & Dey, 2018), se puede expresar utilizando la ecuación:

$$P(u, v) = \frac{1}{1 + \left[ \frac{F(u, v)}{F_0} \right]^{2d}}$$

### **Filtro de paso bajo gaussiano (GLP)**

La función de transferencia de un filtro de GLP se expresa en la ecuación:

$$P(u, v) = e^{-F^2(u,v)/2\sigma^2}$$

Aquí,  $\sigma$  es la desviación estándar y una medida de la extensión de la curva de Gauss. Si  $\sigma$  se sustituye con el  $F_0$  radio de corte, a continuación, la función de transferencia de GLP se expresa como en la siguiente ecuación:

$$P(u, v) = e^{-F^2(u,v)/2F_0^2}$$

### **Filtro de paso alto**

Realiza la anulación de frecuencias más bajas que la frecuencia de corte, pero permite o pasa altas frecuencias bien, un filtro de paso alto se utiliza generalmente para perfilar una imagen y para resaltar los bordes y los detalles

finos asociados a la imagen; los diferentes tipos de filtros de paso alto son ideales, Butterworth y el filtro de paso alto de Gauss asimismo todos los filtros de paso alto (HPF) se pueden representar por su relación con los filtros de paso bajo (LPF) (Chaki & Dey, 2018), como se muestra en la ecuación:

$$HPF = 1 - LPF$$

### **Filtro ideal de alto paso (IHP)**

La función de transferencia de un filtro de IHP se puede expresar por la ecuación(Chaki & Dey, 2018):

$$P(u, v) = \begin{cases} 0 & \text{si } F(u, v) \leq F_0 \\ 1 & \text{si } F(u, v) > F_0 \end{cases}$$

Donde  $F_0$  es la frecuencia de corte o el radio de corte.

### **Filtro Butterworth paso alto (BHP)**

La función que realiza la propagación del filtro de BHP se puede definir por la ecuación(Chaki & Dey, 2018):

$$P(u, v) = 1/(F_0)$$

Donde  $p$  es el orden y  $F_0$  es la frecuencia de corte o el radio de corte:

### **Filtro de paso de banda**

Un pase de banda suprime frecuencias muy altas y muy bajas, pero conserva una banda de gama intermedia de frecuencias, el filtrado de paso de banda se puede usar para bordes de luz (atenuantes bajas frecuencias), mientras que la disminución de la cantidad de ruido al mismo tiempo (supresión de las altas frecuencias) para obtener la función de filtro de paso de banda, la función de filtro de paso bajo se multiplica con la función de filtro de paso alto en el dominio de la frecuencia, donde la frecuencia de corte de la de paso alto es menor que la del paso bajo, por lo tanto, en teoría, una función de filtro de paso de banda puede desarrollarse si la función de filtro de paso bajo está disponible, los diferentes tipos de filtro de paso de banda son ideales paso de banda, paso de banda Butterworth, y de paso de banda Gaussiano(Chaki & Dey, 2018).

### **Filtro ideal pasa bandas (IBP)**

El IBP permite que la frecuencia dentro de la banda de paso y elimina la frecuencia muy alta y muy baja, por otro lado un filtro de IBP dentro de un rango de frecuencia  $F_L, \dots, F_H$  (Chaki & Dey, 2018), se define por la ecuación que se muestra a continuación:

$$P(u, v) = \begin{cases} 1 & \text{si } F_L \leq F(u, v) \leq F_H \\ 0 & \text{de distinta manera} \end{cases}$$

### **Filtro Butterworth pasa bandas (BBP)**

Este filtro se puede obtener multiplicando la función que realiza la propagación de un filtro de alta y baja Butterworth, si  $F_L$  es la frecuencia de corte baja,  $F_H$  es el alto frecuencia de corte, y  $P$  es el orden del filtro entonces el filtro BBP (Chaki & Dey, 2018), se puede definir por la ecuación:

$$B_{LP}(u, v) = \frac{1}{1 + \left[ \frac{F(u, v)}{F_L} \right]^{2p}}$$

$$B_{HP}(u, v) = 1 - 1/(1 + [F(u, v)/F_H])^{2p}$$

$$B_{BP}(u, v) = B_{LP}(u, v) * B_{HP}(u, v)$$

Donde la gama de frecuencia depende de la orden del filtro.

### **Filtro Gaussiano de paso de banda (GBP)**

Este filtro se puede obtener multiplicando la función de transferencia de un bajo y alta filtro de Gauss, si  $F_L$  es la frecuencia de corte baja,  $F_H$  es la frecuencia de corte alta, y  $p$  es el orden del filtro entonces el filtro GBP (Chaki & Dey, 2018), se puede definir por la siguiente ecuación:

$$G_{LP}(u, v) = e^{-F \frac{2(u,v)}{2F_0^2}}$$

$$G_{HP}(u, v) = 1 - e^{-F \frac{2(u,v)}{2F_0^2}}$$

$$G_{BP} = G_{LP}(u, v) * G_{HP}(u, v) \text{ donde } F_L > F_H$$

### **Banda de filtro de rechazo**

Band-filtro de rechazo (filtro también llamada banda-stop) es sólo el opuesto del filtro de paso de banda, se atenúa las frecuencias dentro de un rango de un superior y reducir la frecuencia de corte, los diferentes tipos de filtros de rechazo de banda son banda Ideal rechazan, rechazan banda Butterworth, y la banda de Gauss rechazan(Chaki & Dey, 2018).

### **Ideal rechazo de banda de filtro (IBR)**

En este filtro, las frecuencias dentro de la banda de paso se atenúan y las frecuencias fuera del intervalo dado se hacen pasar sin atenuación, se define un filtro de IBR con una frecuencia de corte  $F_0$ , que es el centro de la banda de frecuencia y donde  $W$  es la anchura de la banda de frecuencia(Chaki & Dey, 2018):

$$P(u, v) = \begin{cases} 0 & \text{si } F_0 - \frac{W}{2} \leq F(u, v) \leq F_0 + W/2 \\ 1 & \text{de otra manera} \end{cases}$$

### **Butterworth rechazo de banda de filtro (BBR)**

En un filtro de BBR, las frecuencias en el centro de la banda están completamente bloqueados. Las frecuencias en el borde de la banda de frecuencias se suprimen por una fracción del valor máximo, si  $F_0$  es el centro de la frecuencia,  $W$  es la anchura de la banda de frecuencia, y  $p$  es el orden del filtro, a continuación, un filtro de BBR (Chaki & Dey, 2018), se puede definir por la siguiente ecuación:

$$P(u, v) = 1 / (1 + [F(u, v)W / (F(u, v)^2 - F_0^2)]^p)$$

### **Rechazo Gaussiano de banda del filtro (GBR)**

Aquí, la transición entre la frecuencia filtrada y sin filtrar es muy suave, si  $F_0$  es el centro de la frecuencia y  $W$  es la anchura de la banda de frecuencia, entonces filtro GBR (Chaki & Dey, 2018), se puede definir por la ecuación:

$$P(u, v) = e^{-[(F(u, v)^2 - F_0^2) / (F(u, v)W)]^2}$$

### **Segmentación**

Es un procedimiento donde la imagen es separada en regiones o componentes las cuales pueden pertenecer a objetos o segmento de objetos asimismo se ocupa de determinar si cada uno de los pixeles de la imagen corresponde al objeto de interés. (Guevara & GIL, 2018).

El curso de la segmentación cuenta con una envergadura de nivel muy alto, la misma que apunta al estudio de datos que contiene una imagen que fue tratada, por ende, el propósito esencial es seccionar la imagen que se está trabajando en fragmentos, los mismos que contienen un nivel de analogía muy alto con los elementos del mundo real y la cual se encuentra comprendida en dicha figuración. El proceso de segmentación de forma total de una figura  $R$  es un conjunto finito de las regiones  $R_1, \dots, R_S$  (Sonka et al., 1993).

$$R = \bigcup_{i=1}^S R_i, \quad R_i \cap R_j = \emptyset, \quad i \neq j.$$

### **1.3.6.1.2. Umbralización.**

Umbral es un procedimiento de transformación de la imagen en gradación gris de entrada en una imagen binarizada o la imagen con una nueva gama de nivel de gris, mediante el uso de un valor umbral particular, su objetivo es extraer algunos píxeles de la imagen, mientras que la eliminación de otros. El propósito de umbral es a los píxeles de la marca que pertenecen al primer plano píxeles con la misma intensidad y de fondo píxeles con diferentes intensidades, el umbral no sólo está relacionada con el campo de procesamiento de imágenes(Chaki & Dey, 2018).

Un umbral es básicamente un valor que tiene dos conjunto de regiones en su cada lado, es decir, por encima del umbral o por debajo del umbral, cualquier función puede tener un valor umbral; la función tiene diferentes expresiones para debajo del valor umbral y para encima del valor umbral entonces para una imagen, si el valor de píxel de la imagen original es inferior o por debajo de un valor umbral particular, se sigue una transformación específica o función de conversión, si no, va a seguir otro, el umbral puede ser global o local, por lo cual umbral global significa el umbral se selecciona toda la imagen umbral adaptativo local o se utiliza cuando la imagen tiene una iluminación irregular, que hace que sea difícil de segmento usando un único umbral, en ese caso la imagen original se divide en sub-imágenes, y para cada sub-imagen se utiliza un umbral particular para la segmentación (Chaki & Dey, 2018).

#### **1.3.6.1.2.1. Umbral basado en la forma del histograma.**

Los presume método de histograma que hay algún valor medio para el primer plano o de objetos de píxeles y fondo, pero la realidad es que el verdadero costoso. Una técnica relativamente sencilla es el método iterativo para encontrar un umbral imagen específica, que también es robusto frente al ruido. Los pasos del método iterativo es el siguiente:

Paso 1: Un umbral inicial (T) se seleccionar arbitrariamente por cualquier otro método deseado.

Paso 2: La imagen  $I(x, y)$  se procede a segmentar en primer plano o de objetos de píxeles y los píxeles del fondo:

$$\text{Píxeles del objeto (OP)} \rightarrow \{I(x, y): I(x, y) \geq T\}$$

$$\text{píxeles de fondo (BP)} \leftarrow \{I(x, y) : I(x, y) < T\}$$

Paso 3: La media de cada conjunto de píxeles se calcula.

$$A_{OP} \leftarrow \text{promedio de OP}$$

$$A_{BP} \leftarrow \text{promedio de BP}$$

Paso 4: se forma un nuevo umbral el cual es el promedio de AOP y ABP:

$$T_{nuevo} \leftarrow \frac{(A_{OP} + A_{BP})}{2}$$

Paso 5: En el paso 2, utilice el nuevo umbral obtenido en la etapa 4. Repetir hasta que el nuevo umbral coincida con la anterior.

Supongamos que la imagen de nivel de gris  $I(x, y)$  se compone de un objeto de luz en un fondo oscuro, de tal manera que el fondo y el objeto, o en primer plano píxeles de niveles de gris, se pueden agrupar de dos formas preponderantes. Una forma clara para realizar la extracción de los píxeles objeto del fondo es seleccionar un umbral  $T$ , que divide estos dos modos, entonces, cualquier píxel  $(x, y)$  donde  $I(x, y) \geq T$  se llama un píxel de objeto, de lo contrario, el píxel se llama un píxel de fondo (Chaki & Dey, 2018).

#### 1.3.6.1.2.2. Umbral basado en agrupamiento.

**K-means Umbral Método:** los pasos de K-means algoritmo para la selección del umbral es la siguiente (Chaki & Dey, 2018):

Paso 1: centros de clases (K) se inicializan:

$$C_{j0} = G_{min} + \left[ \frac{\left( j - \left( \frac{j}{2} \right) \right) (G_{max} - G_{min})}{k} \right]$$

Donde  $j = 1, 2, \dots, k$ ;  $C_{j0}$  es el primer centro de clase de la clase  $j$ ;  $G_{min}$  y  $G_{max}$  son los valores mínimo y máximo de gris el espacio muestral.

Paso 2: Asignar cada punto del espacio de muestra a su centro clase más cercana basado en la distancia euclídea:

$$D_{j,i} = \text{abs}(G_i - C_j)$$

Donde  $j = 1, 2, \dots, k$ ;  $i = 1, 2, \dots, P$ ;  $D_{j,i}$  es la distancia desde un punto a  $j$ th la clase  $j$ -ésima, y  $P$  es el número total de puntos en el espacio de la muestra.

Paso 3: Calcular la (K) nuevos centros de clase a partir de la media de los puntos que se asignan a la misma:

$$C_{nuevo} = \frac{2}{P_i} \sum G_j$$

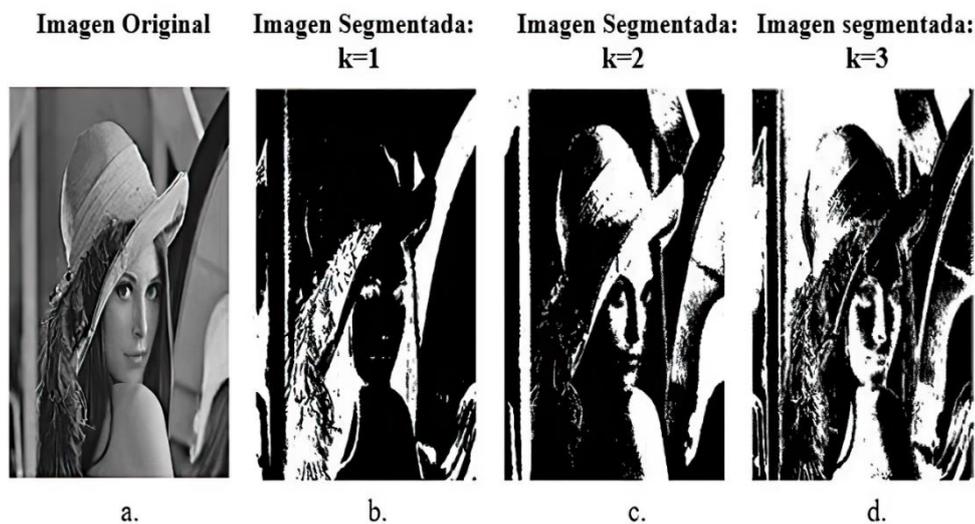
Donde  $j = 1, 2, \dots, k$  y  $P_i$  es el número total de puntos que fueron asignados a la clase  $i$ -ésima en el paso 2.

Paso 4: Repetir el paso 2 para el cambio en el centro de la clase; de lo contrario detener la iteración.

Paso 5: El umbral se calcula la media del centro de la clase  $k$ -ésimo y  $(k - 1)$  centro de clase:

$$T = 1/2(C_k + C_{k-1})$$

El resultado del proceso de segmentar la imagen se visualiza a continuación:



*Figura 13.* Proceso de segmentación de la imagen con los distintos valores de  $k$ . a. Imagen Original. b. Imagen segmentada con el valor de  $k = 1$ . c. Imagen segmentada con el valor de  $k = 2$ . d. Imagen segmentada con el valor de  $k = 3$ . Fuente: (Chaki & Dey, 2018).

### Otsu- Agrupación método umbral

Este método se usa para la selección de un valor de umbral al minimizar el plazo de clase varianzas de dos clusters; la varianza dentro de la clase puede ser expresada por la siguiente ecuación(Chaki & Dey, 2018):

$$\sigma_w^2(T) = P_b(T)\sigma_b^2(T) + P_f(T)\sigma_f^2(T)$$

Donde  $P_f$  y  $P_b$  son la probabilidad de primer plano y de fondo de clase ocurrencias, donde  $T$  es el valor umbral inicial, que se selecciona al azar por algún algoritmo, y  $\sigma_f^2$  y  $\sigma_b^2$  son las varianzas de primer plano y fondo clusters.

### 1.3.6.1.2.3. Umbral basado en entropía.

Este método se creó basado en la función de distribución de probabilidad del histograma de niveles de gris donde dos entropías pueden ser calculados: uno para los píxeles negros y el otro para los píxeles blancos(Chaki & Dey, 2018):

$$\sum_{i=0}^{255} g(i) = 1$$

$$E_b(t) = - \sum_{i=0}^t \frac{g(i)}{\sum_{j=0}^t g(j)} * \log \frac{g(i)}{\sum_{j=0}^t g(j)}$$

$$E_w(t) = - \sum_{i=t+1}^t \frac{g(i)}{\sum_{j=t+1}^{255} g(j)} * \log \frac{g(i)}{\sum_{j=t+1}^{255} g(j)}$$

Donde g (i) es el histograma normalizado.

El valor óptimo único umbral se selecciona mediante la maximización de la entropía de píxeles blancos y negros y puede ser representada por la ecuación siguiente:

$$T = Arg_{t=0} \max E_b (t) + E_w(t).$$

Los valores de umbral óptimo se pueden encontrar con la siguiente ecuación:

$$\{T_1, \dots, T_p\} = Arg_{t_1 < \dots < t_p} \max E(-1, t_1) + E(t_1 + t_2) + \dots + E(t_p, 255)$$

**Imagen Original**



a.

**Imagen Segmentada**



b.

*Figura 14. Segmentación de imagen utilizando el método basado en la entropía. a. Imagen original. b. Imagen segmentada mediante el método de entropía.*

Fuente:(Chaki & Dey, 2018)

### 1.3.6.1.3. Segmentación basada en bordes.

La segmentación de borde es un área vital de la investigación ya que ayuda a la exploración de la imagen de orden superior, es una herramienta importante para la segmentación de imágenes, disminuye de manera significativa la cantidad de datos a procesar, sin embargo, contiene información vital acerca de las formas de los objetos en la escena. Los bordes son básicamente las variaciones locales de intensidad de la imagen con enfoques de detección de bordes convertir las imágenes originales en las imágenes de bordes en función de las variaciones de tonos grises en la figura, el reconocimiento de bordes de la figura también es utilizada por muchas aplicaciones como la identificación forma del objeto, procesamiento de imágenes médicas, datos biométricos, y así sucesivamente(Chaki & Dey, 2018).

#### 1.3.6.1.3.1. Detector de bordes Roberts.

La técnica de detección de bordes Roberts se utiliza para resaltar regiones de frecuencia espacial de alta de la imagen, que corresponde a los bordes, la entrada para el operador es una imagen en escala de grises, asimismo una máscara se utiliza para calcular la salida la cual opera en cada valor asignado por píxel de la figura de entrada(Chaki & Dey, 2018).

**Imagen Original**



a.

**Detector de Bordes Roberts**



b.

*Figura 15.* Detección de bordes en una imagen utilizando el detector de bordes Roberts. a. Imagen original. b. Imagen procesada con el detector de bordes Roberts. Fuente: (Chaki & Dey, 2018).

#### **1.3.6.1.3.2. Detector de bordes Sobel.**

El método de detección de bordes de Sobel utiliza la aproximación Sobel al derivado a los bordes de relieve, conduce los bordes en aquellos puntos donde el gradiente es más alto. El operador consta de un par de 3 x 3 núcleos, o máscaras(Chaki & Dey, 2018).

#### **1.3.6.1.3.3. Detector de bordes Prewitt.**

La detección de bordes Prewitt se utiliza para evaluar la orientación y la magnitud de un borde, el operador consta de un par de 3 x 3 núcleos, o máscaras(Chaki & Dey, 2018).

#### **1.3.6.1.3.4. Detector de bordes Kirsch.**

Detección de bordes Kirsch utiliza una sola máscara y lo gira a ocho direcciones: Norte, Oeste, Este, Sur, Noroeste, Suroeste, Sureste, y Nordeste(Chaki & Dey, 2018).

#### **1.3.6.1.3.5. Detector de bordes Robinson.**

El método Robinson se implementa mediante el uso de coeficientes de 0, 1, y 2, donde las máscaras son simétricos alrededor de su eje direccional, la cual se compone de ceros, la magnitud de borde es el valor máximo obtenido mediante la convolución de la máscara con la zona de imagen de píxeles y el ángulo de borde se puede obtener por el ángulo de la línea de ceros en la máscara que contiene la respuesta máxima(Chaki & Dey, 2018).

#### **1.3.6.2. Extractores de Características.**

El desarrollo para extraer características está basada en la identificación y descripción de las propiedades de los elementos vigentes dentro de una figura como el nivel del color, su grado de textura, zona, entre otros, también depende de la ejecución de manera contundente y adecuada con aplicaciones eficientes(Viera, 2017).

#### **1.3.7. Clasificación Supervisada.**

La clasificación supervisada persigue la adquisición de un modelo permitido para así poder clasificar en próximos casos basándose del aprendizaje de casos antes ya trabajados o conocidos, es decir que a raíz de un conjunto de objetos referidos por un vector de características y de la cual tenemos conocimiento de su clase perteneciente, se procede a la elaboración del entrenamiento o el conjunto de datos de aprendizaje(Giron, 2013).

### **1.3.8. Aprendizaje Supervisado.**

El aprendizaje supervisado tiene por requerimiento la disposición de una agrupación de patrones de los que se tiene conocimiento su clase, dicha agrupación se le nombra conjunto de entrenamiento(Guevara & GIL, 2018).

#### **1.3.8.1. Algoritmo KNN(K- Nearest Neighbor).**

K-Nearest Neighbor es un algoritmo de aprendizaje con supervisión, de manera que partiendo de un conjunto de datos de pruebas busca como objetivo principal clasificar de manera precisa todas las peticiones, el conjunto de datos se encuentra conformado por distintas características y a una sola clase. Este algoritmo computa una instancia para realizar pruebas (d) el producto de este procedimiento el algoritmo realiza una selección la cantidad de instancias denominadas K, que están ubicadas más cerca y a dicha instancia se le estipula la clase más cercana(Giron, 2013).

#### **1.3.8.2. Algoritmo SVM (Support Vector Machine).**

El algoritmo SVM puede desarrollar problema de reconocimiento de tipo binario su proceso se basa en la construcción de un hiperplano o recta en el plano , a raíz de un conjunto de datos para entrenamiento, este hiperplano podrá dividir los puntos en dos clases o grupos(Giron, 2013).

### **1.3.9. Lenguajes de Programación.**

#### **1.3.9.1. Python.**

Es un lenguaje de programación el cual fue desarrollado para mantenerse de manera simple y permite soportar más de un paradigma, desarrollado, de un nivel muy alto el cual se destaca por su gestionamiento automático de los recursos, un sistema para gestionar las excepciones y un nivel muy alto de introspección(Chazallet, 2016).

#### **1.3.9.1.1. Librerías de Python.**

##### **1.3.9.1.1.1. Numpy.**

Paquete esencial que permite realizar computación de forma científica en Python, asimismo es una librería que puede ser empleada en Python el cual otorga, múltiples objetos derivados como por ejemplo las matrices y las matrices enmascaradas, donde incluye lógica, selección, clasificación, entre muchas más operaciones rápidas en matrices(Numpy, 2021).

#### **1.3.9.1.1.2. Matplotlib.**

Es un paquete de Python que permite el trazado 2D que produce gráficos con calidad de producción, también permite el trazado de manera interactiva y no interactiva y puede realizar el almacenamiento de imágenes en distintos formatos de salida como, por ejemplo: PS, PNG, entre otros. También nos otorga gráficos de gran variedad como líneas, gráficos circulares, barras, histogramas(Tosi, 2009).

#### **1.3.9.1.1.3. SciPy**

Es un conjunto de procesos de conveniencia así como también, de algoritmos matemáticos elaborados en la extensión Numpy de Python, lo cual añade de manera significativa un dominio a la sesión que se lleva a cabo de manera interactiva de Python, al otorgarle clases de nivel muy alto y comandos al usuario para que pueda representar y manipular datos(SciPy, 2021).

#### **1.3.9.1.1.4. Keras**

Es una librería para el aprendizaje profundo desarrollado con el fin de permitir una experiencia de manera rápida, es decir obtener resultados de manera rápida y fue desarrollado en el lenguaje de Python(KerasSIG, 2020).

### **1.3.10. Entorno de desarrollo.**

#### **1.3.10.1. Visual Studio Code.**

Visual Studio Code es un programa que permite la edición de código fuente ligero pero a su vez potente, que se desempeña en su escritorio, el cual está disponible para distintos sistemas operativos como Windows, Linux y macOS, este a su vez trae consigo un soporte integrado para Node.js, JavaScript, TypeScript, así como extensiones para múltiples lenguajes como : Python, Java, C++, PHP(Clou, 2018).

#### **1.4. Formulación del Problema.**

¿Cuál es el mejor algoritmo de aprendizaje supervisado para la clasificación de especies de madera aserrada mediante el procesamiento de imágenes digitales?

#### **1.5. Justificación e importancia del estudio.**

Este proyecto se lleva a cabo para identificar el algoritmo más eficiente para la clasificación e identificación de especies de madera la cual será diseñada para ayudar a disminuir el alto índice de suplantación de especies de madera, así como a disminuir el tráfico ilegal de la misma contando un dispositivo el cual tenga la capacidad de brindar una imagen de manera adecuada.

#### **1.6. Hipótesis.**

KNN es el mejor algoritmo de aprendizaje supervisado para la clasificación de especies de madera aserrada mediante el procesamiento de imágenes digitales.

#### **1.7. Objetivos.**

##### **1.7.1. Objetivo general.**

Analizar comparativamente algoritmos de aprendizaje supervisado para la clasificación automática de madera aserrada mediante procesamiento de imágenes digitales.

##### **1.7.2. Objetivos específicos.**

- a) Construir un dataset de imágenes de los distintos tipos de madera.
- b) Implementar algoritmos de extracción de características.
- c) Implementar los algoritmos de aprendizaje supervisado.
- d) Evaluar resultados.

## **II. MATERIAL Y MÉTODO**

### **2.1. Tipo y Diseño de Investigación.**

#### **Tipo de Investigación**

La presente investigación pertenece a una investigación tecnológica aplicada de tipo cuantitativa la cual se basa en analizar diversos datos que pueden ser medidos y cuantificados.

#### **Diseño de la Investigación**

De acuerdo al tipo de investigación el diseño empleado es de tipo cuasi-experimental, con post prueba y será aplicada a un solo grupo porque se analizará las diferentes especies de madera empleando de manera deliberada la variable independiente, la cual es, algoritmos de aprendizaje supervisado de clasificación, para así observar sus efectos en la variable independiente planteada.

### **2.2. Población y muestra.**

#### **Población**

En esta investigación se encontraron 15 algoritmos supervisados para la clasificación, los cuales son los más utilizados por diversos autores según la investigación realizada, el listado de dichos algoritmos se puede visualizar en el Anexo 4.

#### **Muestra**

La muestra ha sido determinada por la técnica no probabilística denominada técnica por conveniencia, para el desarrollo de la investigación se considera 3 algoritmos supervisados: SVM, KNN y la red neuronal VGG16 para la clasificación de madera aserrada mediante textura del total de la población.

### **2.3. Variables, Operacionalización.**

#### **2.3.1. Variable Independiente.**

Algoritmos de aprendizaje supervisado.

#### **2.3.2. Variable Dependiente.**

Clasificación de especies de madera.

### 2.3.3. Operacionalización de Variables.

<b>Variables Independiente</b>	<b>Indicadores</b>	<b>Fórmula</b>	<b>Técnica de recolección de datos</b>
	Consumo de CPU	$TCPU = \text{Uso total de CPU del proceso}$ $NCPU = \text{No uso del proceso}$ $CPU = TCPU / NCPU$	
Algoritmos de aprendizaje supervisado de clasificación.	Consumo de Memoria RAM	$TM = \text{Total de Memoria}$ $UM = \text{Uso de memoria}$ $\text{Utilización de Memoria} = TM - UM / TM$	Técnica de Observación. Ficha de Observación.
	Tiempo de respuesta	$TI = \text{tiempo de Inicio}$ $TF = \text{tiempo Final}$ $TR = TI - TF$	
<b>Variable Dependiente</b>	<b>Indicadores</b>	<b>Fórmula</b>	<b>Técnica de recolección de datos</b>
Clasificación de especies de madera por textura.	Precisión	$\text{Precisión} = VP / FP + VP$	Técnica de observación.
	Exhaustividad	$\text{Exhaustividad} = VP / VP + FN$	Técnica de observación.

ROC-área	$1 - \text{especificidad} = \text{FP} / \text{VN} + \text{FP}$	Técnica de observación.
----------	--	-------------------------

---

## **2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.**

### **Observación**

Analizar de una manera atenta para verificar los rasgos, patrones que presenta cada especie de madera para su clasificación y/o diferenciación de otras especies, así como el comportamiento de los algoritmos.

### **Ficha Técnica**

Evaluar de forma documentada de las características que presentará el software de acuerdo a su desempeño y funcionalidad para la obtención de resultados.

## **2.5. Procedimiento de análisis de datos.**

### **Promedio**

Se realizó la suma de la lista de resultados de cada algoritmo de clasificación de aprendizaje supervisado dividida por el número de veces que el algoritmo haya sido ejecutado para así determinar la tendencia general del conjunto de datos.

### **Consumo de CPU**

Se realizó para la evaluación de rendimiento de los algoritmos (evaluación comparativa). La fórmula a emplearse es:

$$\boxed{CPU = TCPU / NCPU}$$

Donde:

$$TCPU = \text{Uso total de CPU}$$

$$NCPU = \text{No uso de proceso}$$

### **Consumo de memoria RAM**

Basado en el consumo que ejecuta el algoritmo respecto a la memoria RAM. La fórmula a emplearse es:

$$\boxed{\text{Utilización de Memoria} = TM = TI - TF}$$

Donde:

$$TM = \text{Total de Memoria}$$

$$UM = \text{Uso de Memoria}$$

### **Tiempo de respuesta**

Tiempo que toma ejecutarse el clasificador. La fórmula a emplearse es:

$$TR = TI - TF$$

Donde:

*TI = Tiempo de Inicio*

*TF = Tiempo Final*

### **Precisión**

Es una fracción de sesiones que se determinan de manera correcta. La fórmula es la siguiente:

$$precisión = \frac{VP}{VP + FP}$$

Donde:

VP = Son el número verdaderos positivos, es decir, de predicciones correctas.

FP = Son el número de falsos negativos, es decir, la predicción es negativa cuando realmente el valor es positivo.

### **Exhaustividad**

Demuestra la proporción de los casos positivos que se detectaron de manera correcta.

$$exhaustividad = \frac{VP}{VP + FN}$$

Donde:

VN: Son el número de verdaderos negativos, es decir, de predicciones correctas.

VP: Son el número verdaderos positivos, es decir, de predicciones correctas.

FP: Son el número de falsos negativos, es decir, la predicción es negativa cuando realmente el valor es positivo.

FN: Son el número de falsos negativos, es decir, la predicción es negativa cuando realmente debería ser negativo.

### **ROC- área**

Medición del rendimiento del modelo de clasificación.

$$1 - especificidad = \frac{FP}{VN + FP}$$

## **2.6. Criterios éticos.**

**Derechos de autor:** Todo el buen uso de la información para el desarrollo del trabajo de investigación estará adecuadamente referenciado y citado con sus respectivos autores de desarrollo.

**Confidencialidad:** Se asocia a un código ético, a la seguridad y protección de la identidad de las personas y además a los informantes de la investigación, basándose en el reconocimiento de la dignidad del ser humano y respeto a la autonomía.

## **2.7. Criterios de Rigor Científico.**

**Fiabilidad:** La investigación cumplió con su implementación de acuerdo con la estándar y política para su desarrollo.

**Validez:** La operacionalización de las variables y la definición de las dimensiones están escritas de una manera clara y concisa. Estas dimensiones se evaluaron según los indicadores establecidos y las técnicas de recolección de datos definida, de esa manera el proyecto de investigación será evaluado y analizados.

**Originalidad:** El desarrollo del trabajo de investigación fue citado por fuentes bibliográficas, para reflejar la ausencia del plagio.

### III. RESULTADOS.

#### 3.1. Resultados en Tablas y Figuras.

En este trabajo de investigación se evaluaron algoritmos de clasificación de lenguaje supervisado como Support Vector Machine y K-Nearest Neighbor y la red neuronal VGG16 que mediante el Filtro de Gabor y la extracción de 3 características realizaron la clasificación de las especies de madera. Se realizó las pruebas correspondientes con cada algoritmo, los resultados obtenidos con el algoritmo SVM se muestran a continuación en la siguiente tabla.

Tabla 4.

*Cuadro con valores con respecto al comportamiento del algoritmo SVM.*

<b>especie</b>	<b>precisión</b>	<b>recall</b>	<b>roc-area</b>
capinuri	75.70%	78.00%	96.20%
capirona	34.70%	34.00%	71.90%
catahua	18.80%	09.00%	62.30%
cedro	45.90%	45.00%	84.40%
copaiba	29.40%	32.00%	78.80%
cumala	33.90%	43.00%	84.00%
huayruro	45.60%	41.00%	88.60%
lupuna	22.70%	15.00%	80.40%
moena	11.40%	04.00%	64.60%
tornillo	31.40%	71.00%	81.40%

*Nota:* Los valores de rendimiento fueron calculados usando un 80 % de entrenamiento y un 20% de prueba de la base de datos. Fuente: Elaboración propia.

La precisión denota la proporción de las detecciones positivas por cada clase que fueron realizadas de manera correcta. El cuadro estadístico de la figura 16 muestra los porcentajes obtenidos, el porcentaje más bajo con un 11.40 % de precisión para la madera de tipo moena y el nivel más alto con un 75.70% lo obtuvo la madera de tipo capinuri.

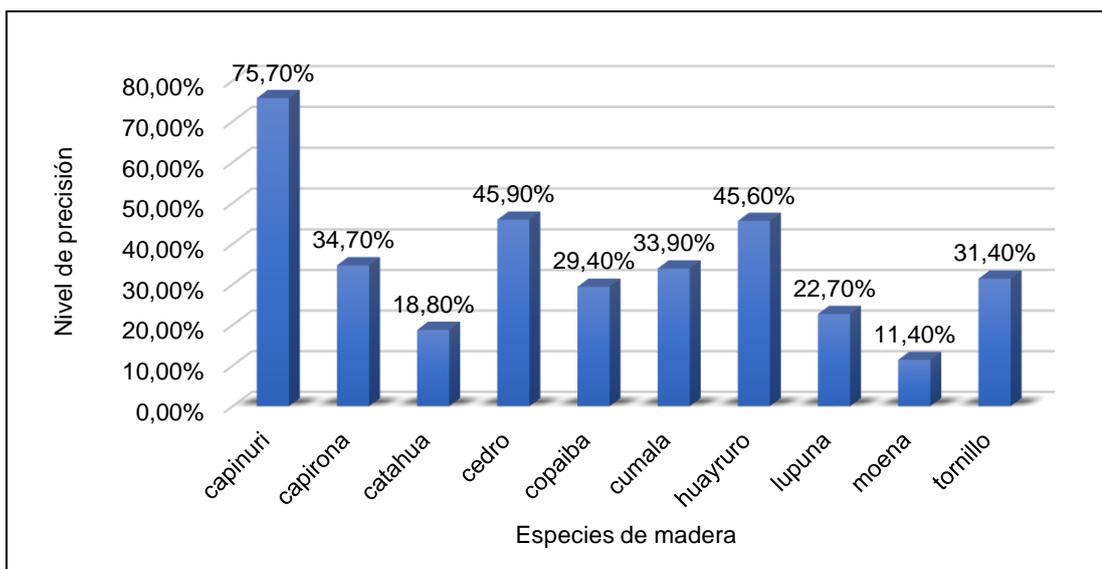


Figura 16. Cuadro estadístico de la precisión con el algoritmo SVM.  
Fuente:Elaboración propia.

La métrica de exhaustividad realiza el cálculo del porcentaje de las especies de madera que han sido clasificadas, el comportamiento de algoritmo con respecto a esta métrica demuestra que la especie de madera moena obtiene el porcentaje más bajo con un 4% y la especie de madera capinuri obtiene el nivel más alto con 78%, como se observa a continuación.

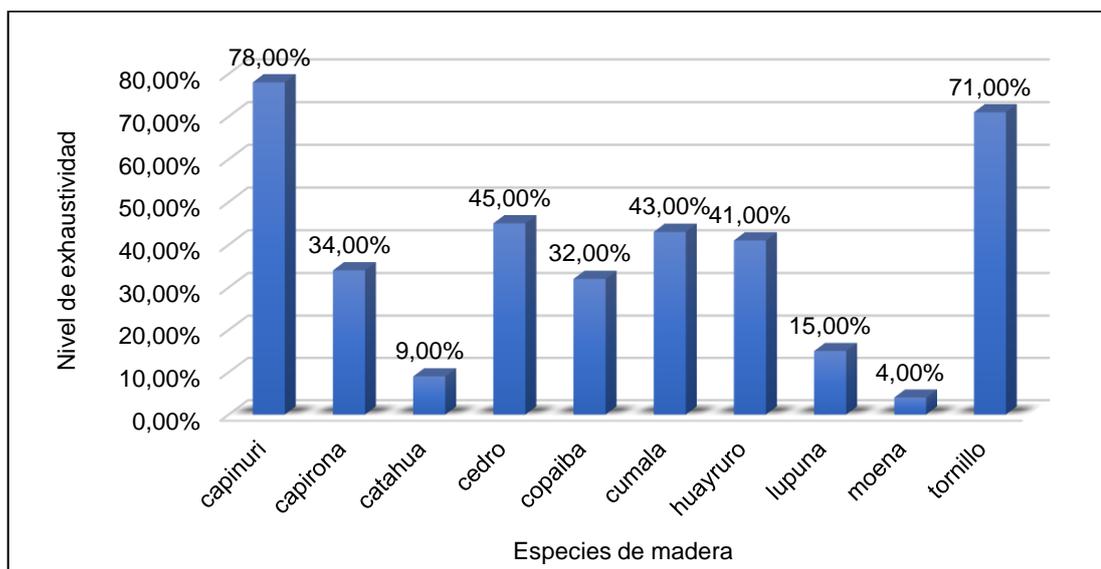


Figura 17. Cuadro estadístico de la exhaustividad con el algoritmo SVM.  
Fuente:Elaboración propia.

La métrica del análisis ROC permitió identificar el porcentaje del rendimiento de las pruebas realizadas a cada especie de madera obteniendo como el nivel más bajo a la especie de madera catahua con un 62.30 % y el nivel más alto a la especie capinuri con un 96.20%.

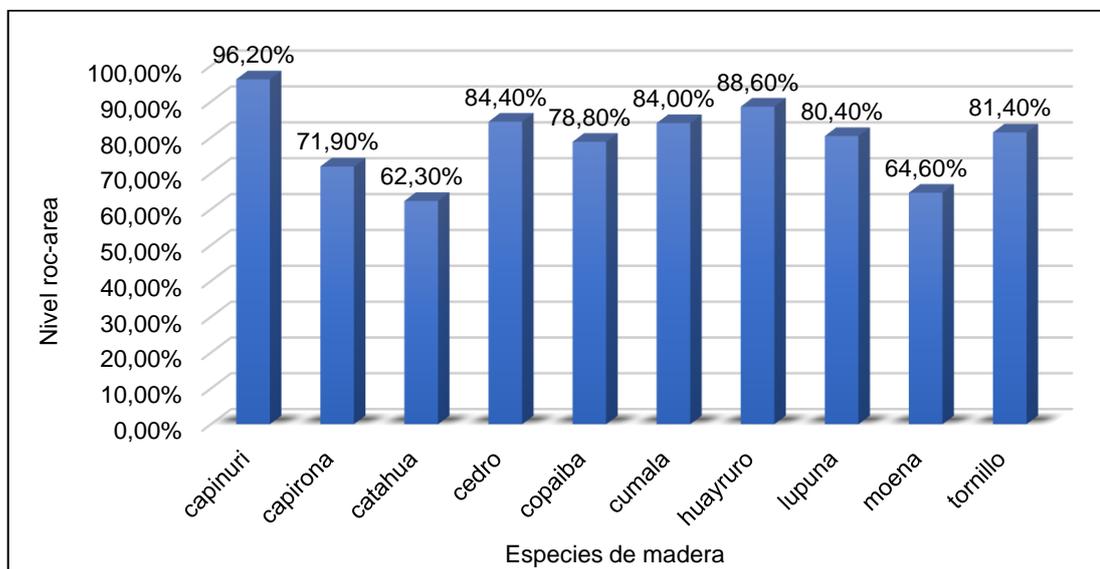


Figura 18. Cuadro estadístico del nivel roc-area con el algoritmo SVM. Fuente: Elaboración propia

El comportamiento del segundo algoritmo con respecto a las pruebas realizadas se detalla en la siguiente tabla.

Tabla 5.

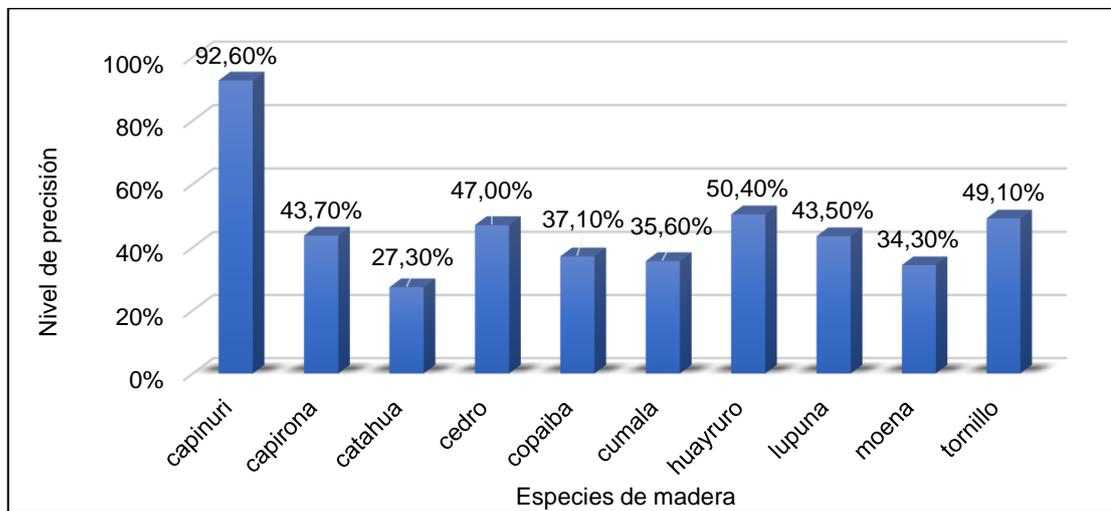
Cuadro con valores con respecto al comportamiento del algoritmo KNN.

Especie	precisión	recall	roc-area
capinuri	92.60%	75.00%	94.50%
capirona	43.70%	31.00%	79.10%
catahua	27.30%	18.00%	73.30%
cedro	47.00%	54.00%	89.20%
copaiba	37.10%	23.00%	75.10%
cumala	35.60%	31.00%	83.10%
huayruro	50.40%	68.00%	93.50%
lupuna	43.50%	27.00%	85.00%
moena	34.30%	71.00%	88.10%

tornillo	49.10%	56.00%	91.20%
----------	--------	--------	--------

*Nota:* Los valores de rendimiento fueron calculados usando un 80 % de entrenamiento y un 20% de prueba de la base de datos. Fuente: Elaboración propia.

La precisión denota la proporción de las detecciones positivas por cada clase que fueron realizadas de manera correcta. El cuadro estadístico de la figura 19 muestra los porcentajes obtenidos, el porcentaje más bajo con un 27.30 % de precisión para la madera de tipo catahua y el nivel más alto con un 92.60% lo obtuvo la madera de tipo capinuri.



*Figura 19.* Cuadro estadístico de la precisión con el algoritmo KNN. Fuente:Elaboración propia

La métrica de exhaustividad realiza el cálculo del porcentaje de las especies de madera que han sido clasificadas, el comportamiento de algoritmo con respecto a esta métrica demuestra que la especie de madera catahua obtiene el porcentaje más bajo con un 18% y la especie de madera capinuri obtiene el nivel más alto con 75%, como se observa a continuación en la figura 20.

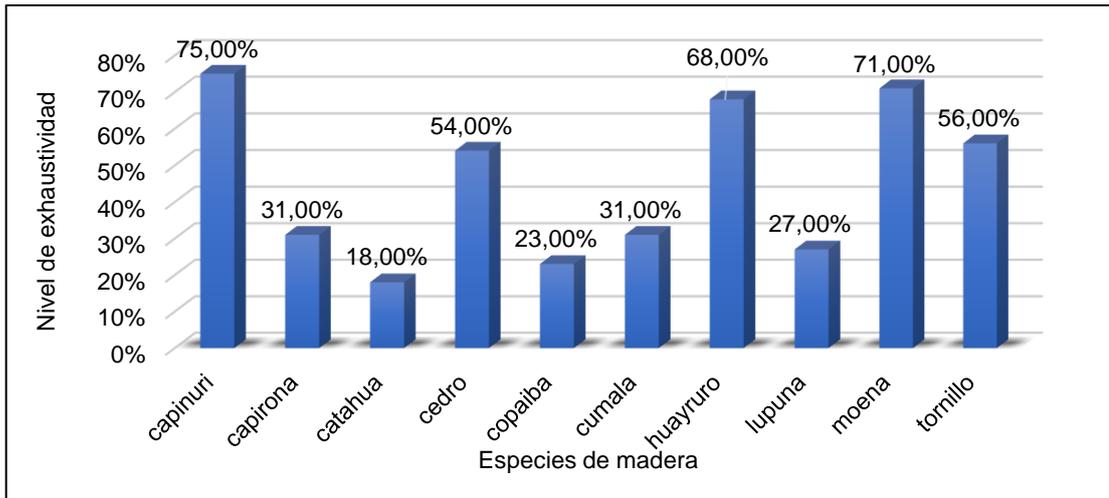


Figura 20. Cuadro estadístico de la exhaustividad con el algoritmo KNN. Fuente: Elaboración propia.

La métrica del análisis ROC permite identificar el porcentaje del rendimiento de las pruebas realizadas a cada especie de madera obteniendo como el nivel más bajo a la especie de madera catahua con un 73.30 % y el nivel más alto a la especie capinuri con un 94.50%.

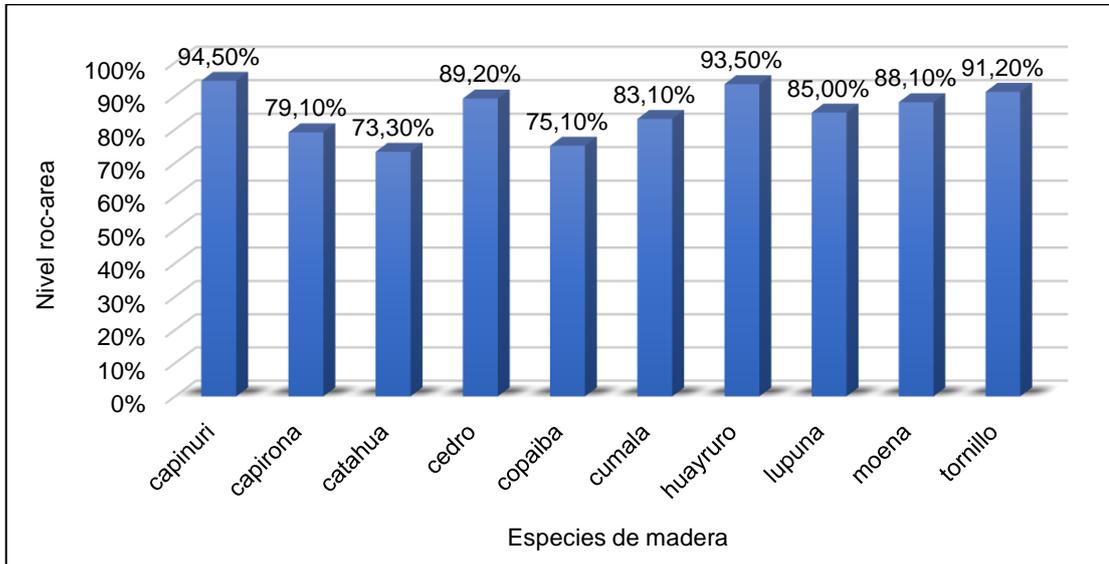


Figura 21. Cuadro estadístico del nivel roc-area con el algoritmo KNN. Fuente: Elaboración propia.

El comportamiento de la red neuronal VGG16 con respecto a las pruebas realizadas se detalla en la siguiente tabla.

Tabla 6.

Cuadro con valores con respecto al comportamiento de la red neuronal VGG16.

especie	Precisión	Recall	Roc-area
Capinuri	100%	100%	96.65%
Capirona	100%	100%	97.77%
Catahua	100%	80.00%	98.32%
Cedro	100%	100%	98.10%
Copaiba	100%	100%	98.90%
Cumala	100%	100%	97.20%
Huayruro	100%	100%	98.20%
Lupuna	100%	100%	98.88%
Moena	83.33%	100%	97.90%
Tornillo	100%	100%	95.10%

*Nota:* Los valores de rendimiento fueron calculados usando un 90 % de entrenamiento, 10% de validación y un 10% de prueba de la base de datos.

Fuente: Elaboración propia.

La precisión denota la proporción de las detecciones positivas por cada clase que fueron realizadas de manera correcta. El cuadro estadístico de la figura 22 muestra los porcentajes obtenidos, el porcentaje más bajo con un 83.33 % de precisión para la madera de tipo moena y el nivel más alto con un 100% lo obtuvo el resto de especies.

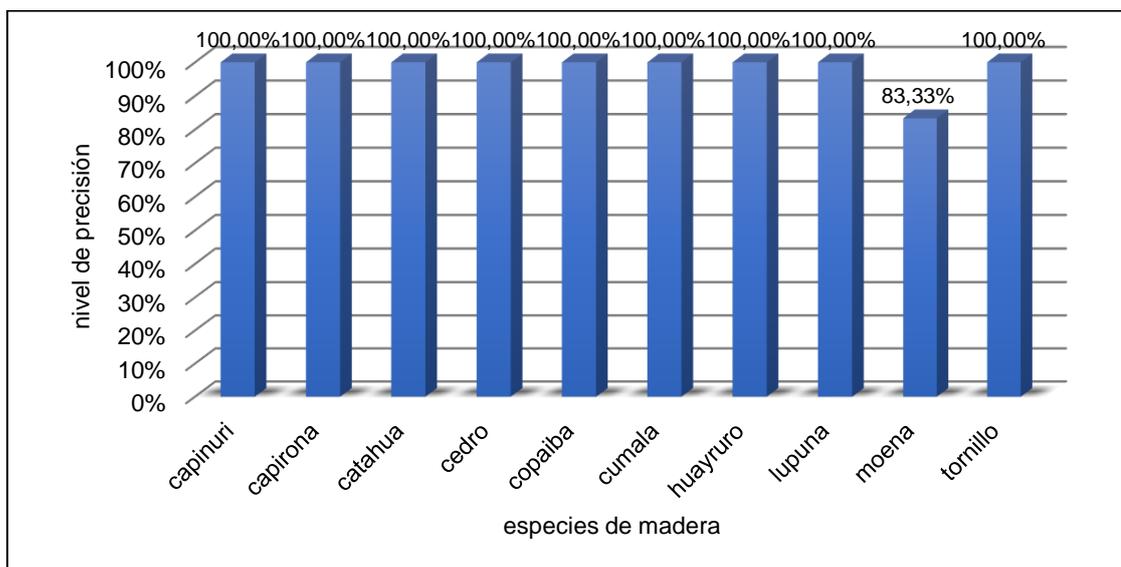


Figura 22. Cuadro estadístico de la precisión con la red neuronal VGG16.Fuente: Elaboración propia.

La métrica de exhaustividad realiza el cálculo del porcentaje de las especies de madera que han sido clasificadas, el comportamiento de algoritmo con respecto a esta métrica demuestra que la especie de madera catahua obtiene el porcentaje más bajo con un 80% mientras que el resto de especies de madera obtienen el nivel más alto con un 100%, como se observa a continuación en la figura 23.

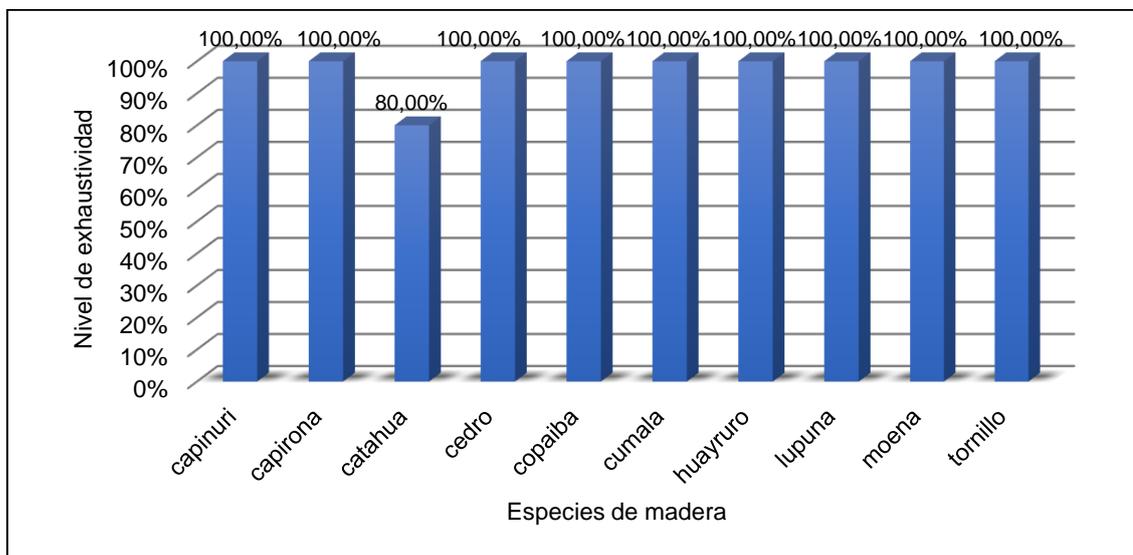
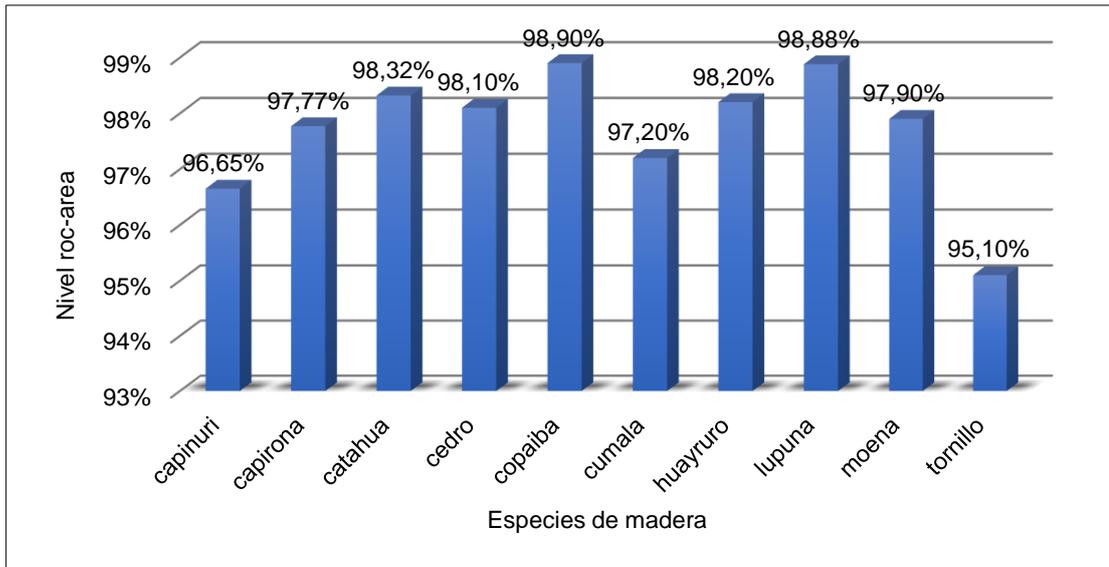


Figura 23. Cuadro estadístico de la exhaustividad con la red neuronal VGG16.Fuente: Elaboración propia.

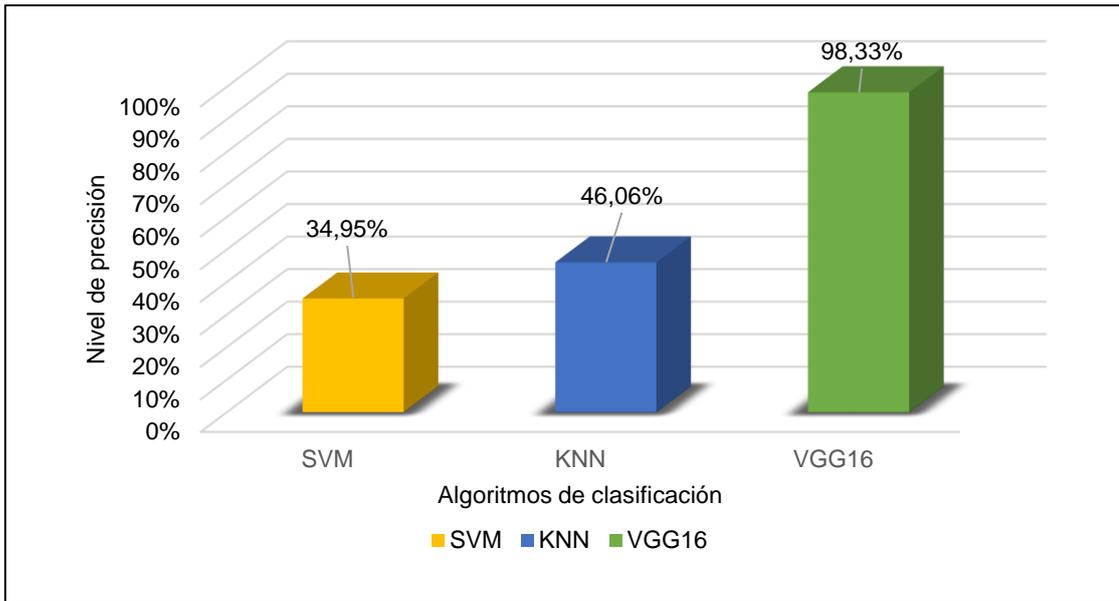
La métrica del análisis ROC permite identificar el porcentaje del rendimiento de las pruebas realizadas a cada especie de madera obteniendo como el nivel más bajo a la especie de madera tornillo con un 95.10% y el nivel más alto a la especie copaiba con un 98.90%.



*Figura 24.* Cuadro estadístico del nivel roc-area con la red neuronal VGG16. Fuente: Elaboración propia.

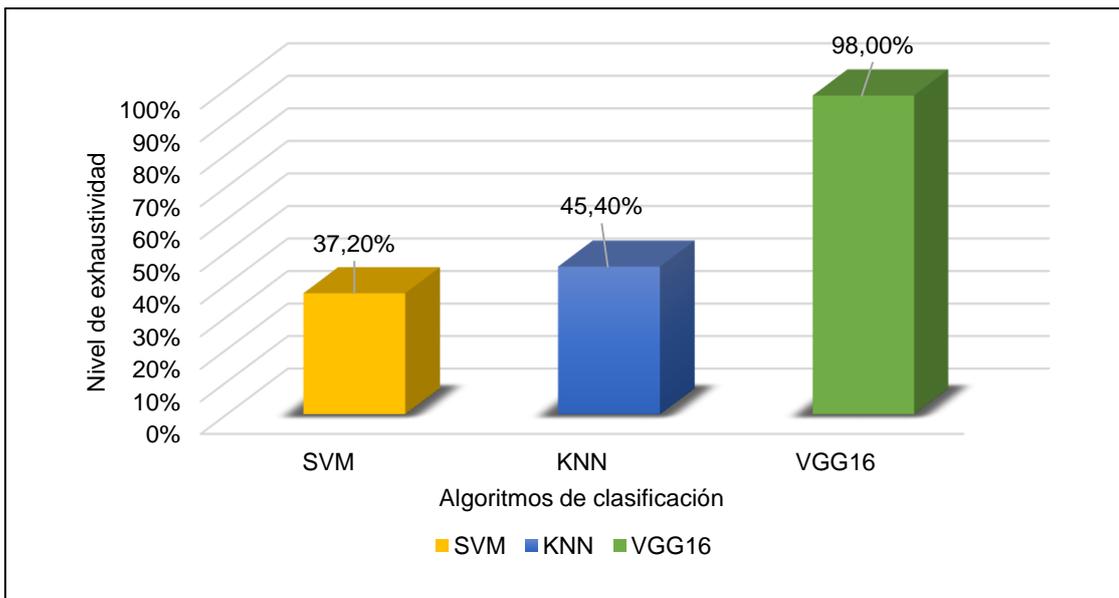
Los algoritmos de clasificación fueron evaluados por la precisión, la exhaustividad, el análisis ROC – consumo de CPU, consumo de memoria RAM y el tiempo de respuesta.

La evaluación y comparación de algoritmos con respecto a la precisión demuestra la clasificación positiva de maderas que se realizaron de manera correcta en las pruebas realizadas, en el cuadro comparativo de la figura 25 muestra al algoritmo SVM con un 34.95% de precisión siendo el nivel más bajo frente al algoritmo KNN que presenta un 46.06% de precisión y a la red neuronal VGG16 que presenta un 98.33% de precisión siendo el nivel más alto.



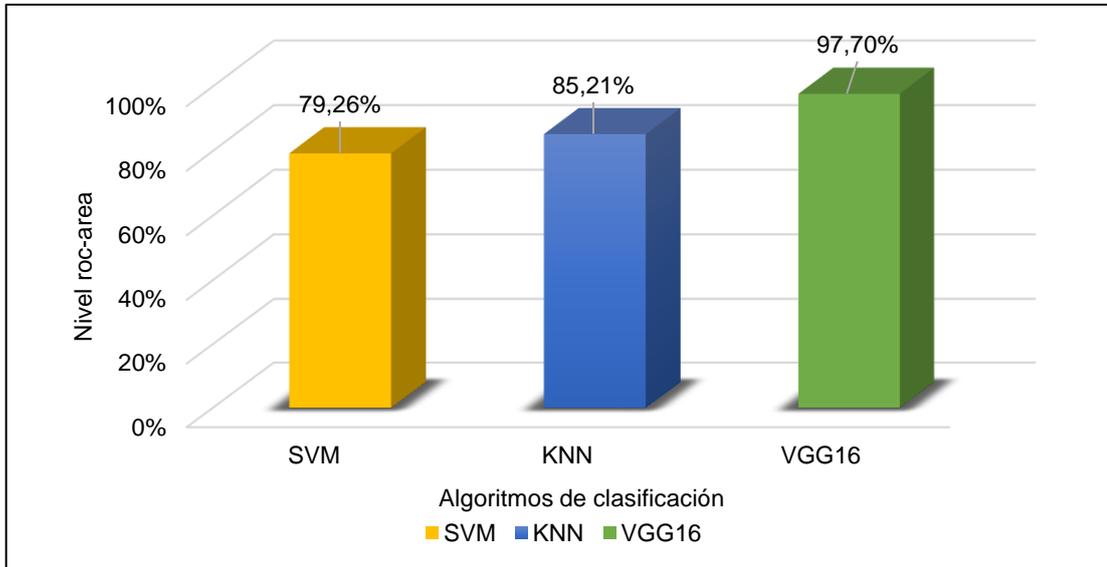
*Figura 25.* Comparación de precisión de los algoritmos de clasificación.  
Fuente:Elaboración propia.

En tanto la exhaustividad nos muestra el porcentaje de maderas reales que fueron clasificadas en las pruebas realizadas, donde la red neuronal VGG16 detectó la mayoría de maderas según su especie obteniendo el 98.00% de exhaustividad, el algoritmo KNN obtuvo el 45.40% de exhaustividad, mientras que el algoritmo SVM obtuvo un menor porcentaje con un 37.20% de exhaustividad como se muestra a continuación.



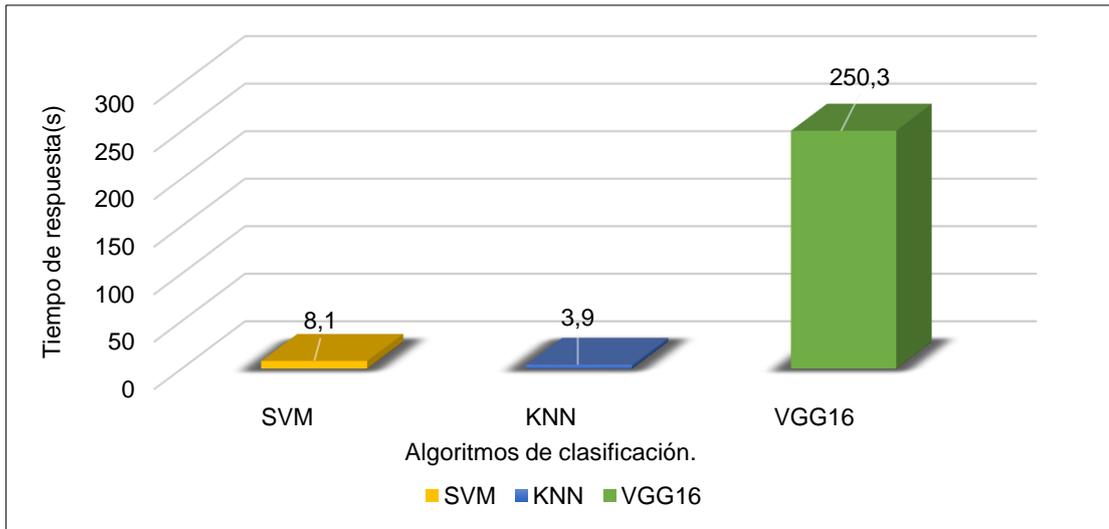
*Figura 26.* Comparación de exhaustividad de los algoritmos de clasificación.  
Fuente: Elaboración propia.

Mediante la comparación de la métrica del análisis ROC permite identificar el porcentaje del rendimiento de las pruebas realizadas a cada modelo de clasificación de madera. La figura 27 nos muestra que la red neuronal VGG16 presenta un mejor resultado con un 97.70% frente al algoritmo KNN que obtuvo un 85.21% y al algoritmo SVM que obtuvo un 79.26 en tanto al análisis ROC.



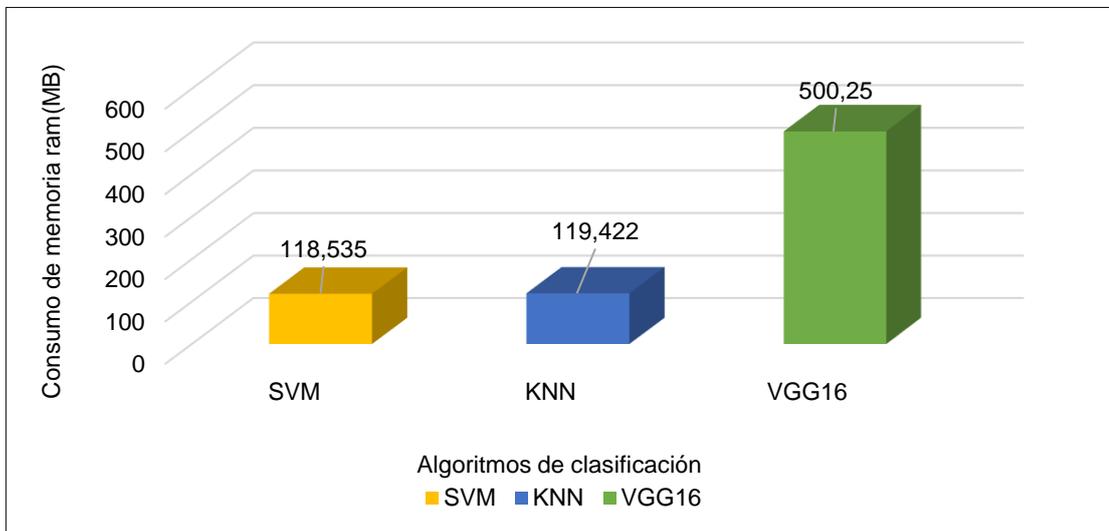
*Figura 27.* Comparación del análisis ROC de los algoritmos de clasificación.  
Fuente: Elaboración propia.

El tiempo de respuesta demuestra la rapidez de un algoritmo al ejecutar la clasificación de especies de madera, los tiempos que se obtuvieron con los algoritmos son los siguientes un tiempo de 250.3s para la red neuronal VGG16, un 8.1 s para el algoritmo SVM, mientras que la aplicación del algoritmo KNN obtuvo un 3.9 s. Teniendo un mejor resultado el algoritmo KNN.



*Figura 28.* Comparación de tiempo de respuesta de los algoritmos de clasificación. Fuente: Elaboración propia.

En la figura 29 se puede observar la comparación referente al consumo de memoria ram de los algoritmos evaluados, donde se registra un consumo del 118.535 MB de ram para el algoritmo SVM, un 119.422 MB de ram para el algoritmo KNN y 500.25 MB de ram para la red neuronal VGG16.



*Figura 29.* Comparación del consumo de memoria ram de los algoritmos de clasificación. Fuente: Elaboración propia.

En la figura 30 se observa un cuadro estadístico comparativo del consumo de cpu para los algoritmos evaluados, donde el algoritmo SVM obtiene un valor del 1.08% de consumo de cpu, frente a ello el algoritmo KNN obtiene un 2.13% de

consumo y un 10.20% obtenido por la red neuronal VGG16. Se registra que el algoritmo con menor porcentaje de consumo es SVM.

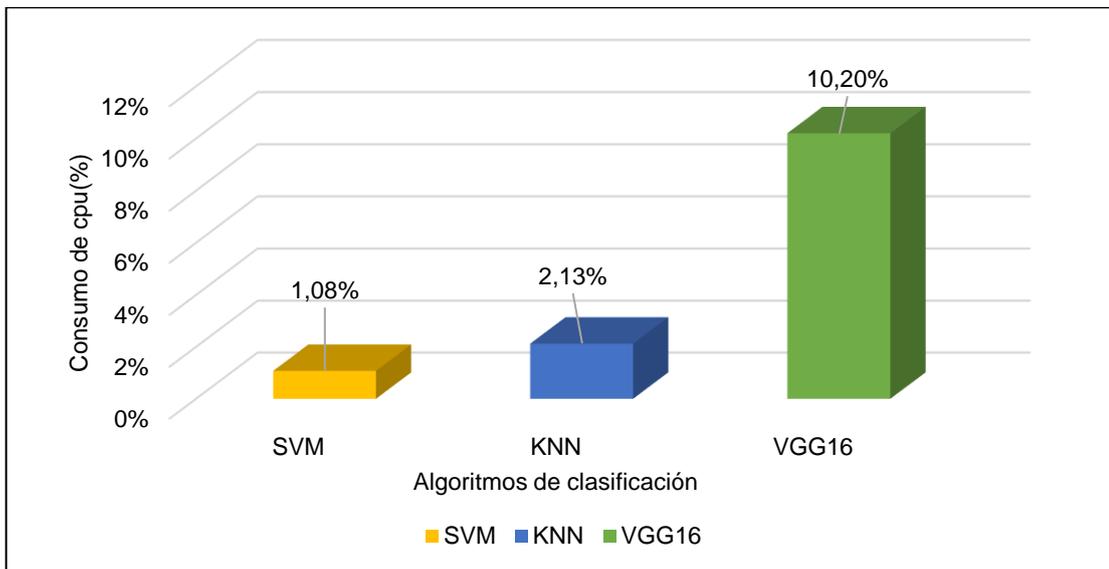


Figura 30. Comparación del consumo de cpu de los algoritmos de clasificación.

Fuente: Elaboración propia.

### 3.2. Discusión de resultados.

En el proceso de clasificación se emplearon dos algoritmos de clasificación, los cuales fueron alimentados por los resultados obtenidos mediante un extractor de características Gabor, los resultados de la red neuronal VGG16 fueron del 98.33% de precisión frente al clasificador KNN que obtuvo un 46.06% de precisión y el clasificador SVM que obtuvo el 34.95%, siendo esto una diferencia significativa en este indicador, mientras que en el indicador de exhaustividad se obtuvieron los siguientes resultados un 37.20% para el clasificador SVM ,un 45.40% para KNN y un 98.00% para la red neuronal VGG16 siendo mejor el último clasificador, en la investigación previa desarrollada por Sukrisdyanto, donde realiza la comparación de 5 clasificadores, señala que el clasificador KNN obtuvo los mejores resultados con un 95% de precisión frente a los demás algoritmos(Sukrisdyanto et al., 2019).

Para el indicador roc el clasificador VGG16 presenta un mejor porcentaje con un 97.70% frente a un 85.21% obtenido por KNN y un 79.26% obtenido por el clasificador SVM.

En tanto los tiempos de respuesta donde el clasificador KNN realizó la clasificación en 3.9 s. frente a 8.1s obtenido por el clasificador SVM y un tiempo de 250.3s obtenido por la red neuronal VGG16 ,mostrando una diferencia considerable en tiempo, resultados contrarios fueron obtenidos por Nurthohari en su investigación, donde señala que el algoritmo SVM obtuvo un cálculo de tiempo de 1.40s y una tasa de rendimiento del 90% en la clasificación de especies de madera(Nurthohari et al., 2019).

Con respecto a los resultados obtenidos del consumo de memoria ram el clasificador SVM obtuvo un menor consumo con un 118.535 MB frente a un 119.422 MB de consumo por parte del algoritmo KNN y un total de 500.25 MB de consumo por la red neuronal VGG16, finalmente en el indicador de consumo de cpu de denota un menor porcentaje en el clasificador SVM con un 1.08% frente al clasificador KNN que obtuvo el 2.13% y la red neuronal VGG16 que obtuvo un 10.20%.

Entonces podríamos decir que la red neuronal VGG16 es mejor en precisión, exhaustividad, pero con un mayor tiempo de respuesta, mayor porcentaje en el consumo de recursos frente a los algoritmos SVM y KNN.

### 3.3. Aporte práctico.

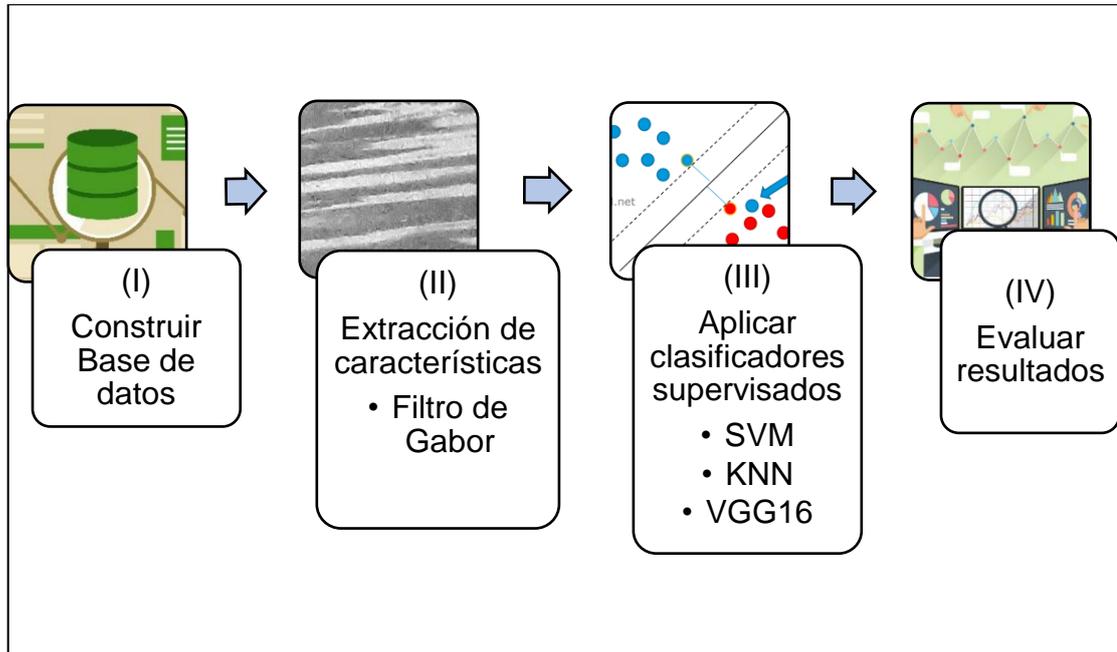


Figura 31. Método general propuesto. Fuente: Elaboración propia.

#### I. Construir Base de Datos.

Para la construcción de la base de datos se planteó un protocolo de 3 pasos, como se ilustra a continuación en la siguiente figura:

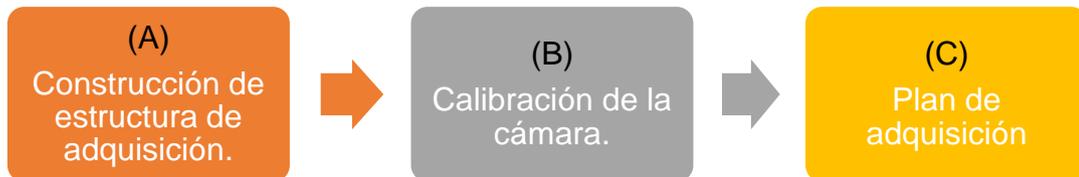


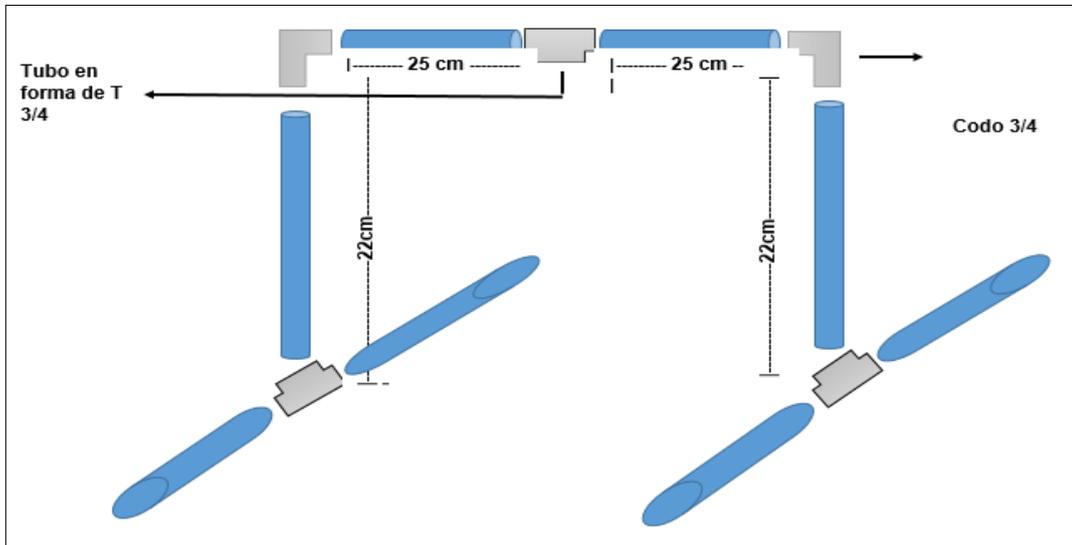
Figura 32. Pasos a seguir para el protocolo de obtención de imágenes digitales.

Fuente: Elaboración propia.

#### A. Construcción de estructura de adquisición.

Para la construcción de la estructura se empleó tubos de material de PVC con las siguientes medidas, 2 piezas de 22 centímetros cada una las cuales conformaron los postes paralelos, 2 piezas de 25 centímetros c/u, que unidos por 1 tubo de  $\frac{3}{4}$  en forma de T formaron el poste superior, 2 piezas de tamaño reducido que sirve para cambiar de dirección las tuberías denominados codos de PVC de  $\frac{3}{4}$  los mismos que sirven de unión entre el poste superior y los postes

paralelos, 2 tubos en forma de T de  $\frac{3}{4}$ , los cuales estaban unidos a la parte inferior de los postes paralelos para que sirvan de soporte como se aprecia en la siguiente figura.

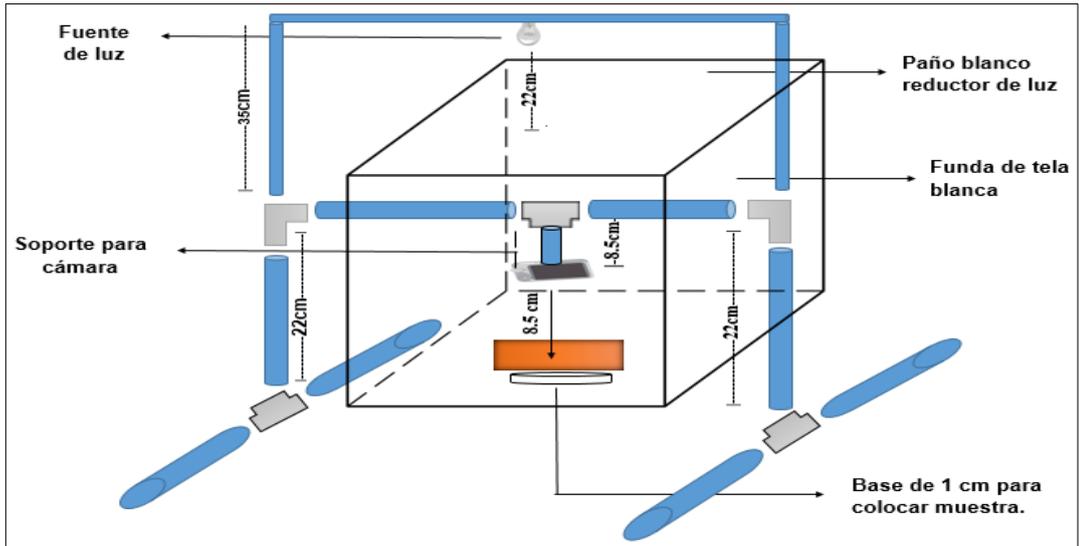


*Figura 33.* Estructura realizada en material tubo de PVC para la adquisición de imágenes digitales con las medidas correspondientes. Fuente: Elaboración propia.

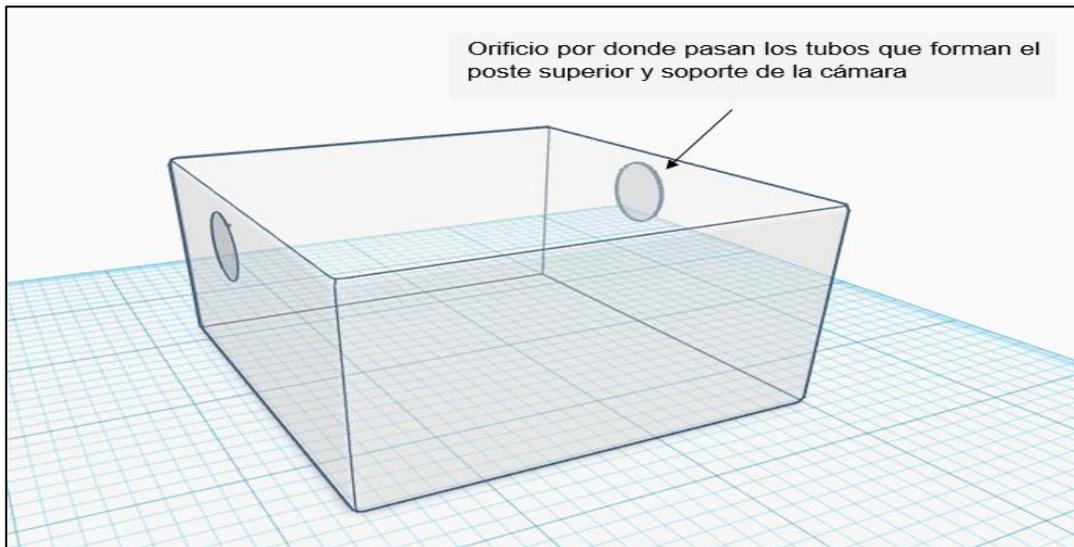
## **B. Calibración de la cámara.**

Para la calibración del dispositivo de adquisición de imágenes digitales dentro de un ambiente controlado, se optó por situar el soporte en el centro de la estructura, posterior a ello se realizó el ajuste de la cámara regulando el soporte, el cual se encuentra sujeto al tubo de PVC en forma de T del poste superior a una distancia de 8.5 cm de altura de la muestra de la madera, que se encontró situada encima de una plataforma y/o base de 1 cm de altura.

Para el control de la iluminación y sombras se construyó una estructura de forma cuadrada, la cual se encuentra forrada por tela de color blanco, y en la parte superior está cubierta por un paño de color blanco para la reducción de luz.



*Figura 34.* Calibración de la cámara empleada, detallando distancias y ubicación con respecto a la muestra utilizada para la captura de imágenes digitales controlando la iluminación y sombras. Fuente: Elaboración propia.



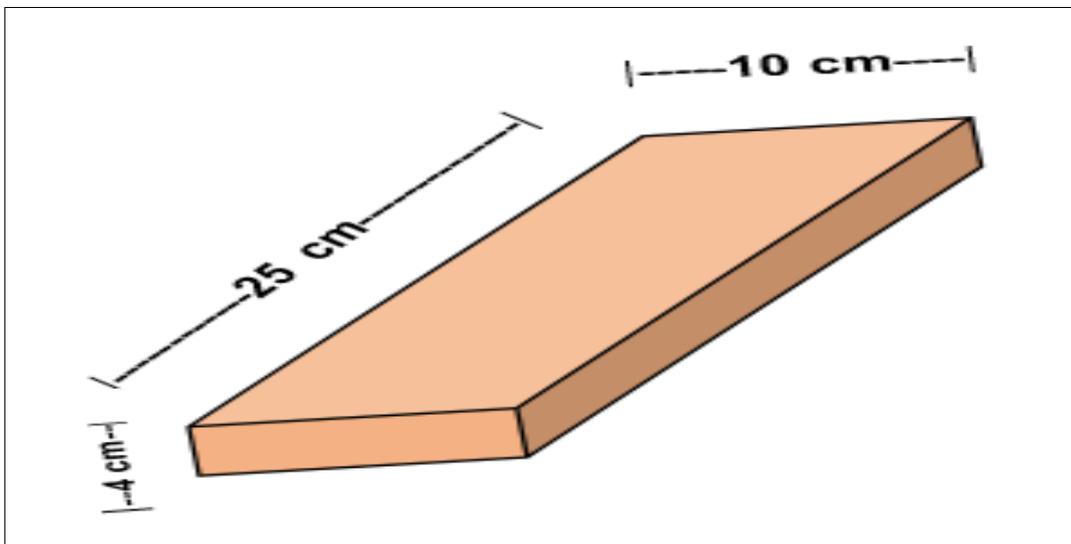
*Figura 35.* Diseño 3D de la estructura para el control de iluminación y sombras para la adquisición de imágenes. Fuente: Elaboración propia.



*Figura 36.* Imagen real de la estructura para la adquisición de imágenes digitales de la madera. Fuente: Elaboración propia.

### **C. Plan de adquisición.**

La madera empleada en la investigación fue recortada de 25cm de largo x 10cm de ancho y 4cm de alto como se muestra en la siguiente figura.



*Figura 37.* Medidas de las muestras de madera empleadas para la adquisición de imágenes digitales. Fuente: Elaboración propia.

En la siguiente tabla se detalla las especificaciones del dispositivo de adquisición de imágenes digitales que se utilizó en esta investigación.

Tabla 7.

*Especificaciones del dispositivo de adquisición de imágenes digitales.*

<b>Característica</b>	<b>Detalle</b>
Resolución	12 Mpx
Pixel de la cámara	4032x3024
Pixel total de la cámara	12192768
Apertura focal	f/1.7
Distancia focal	28mm(lente)
Tamaño del sensor	1/3" pulgadas
Zoom digital	X5

*Nota:* Las especificaciones técnicas del dispositivo de adquisición de imágenes fueron obtenidas de la información de cámara del mismo. Fuente: Elaboración propia.

Luego fueron establecidos los parámetros de configuración del dispositivo de adquisición para asegurar imágenes digitales con buenas características.

Tabla 8.

*Configuración del dispositivo de adquisición de imágenes digitales.*

<b>Característica</b>	<b>Detalle</b>	<b>Comentario</b>
Apertura focal	f/1.7	Debido al control de la iluminación
Velocidad de obturación	1/24s	Para la regular la cantidad de luz que ingresa al sensor de la cámara
Balance de blancos	5300k	Debido al control de los niveles RGB
Velocidad ISO	400	Controlar el nivel de exposición y ruido de la imagen

Compensación de exposición	0.0	Debido al control de imágenes brillantes u oscuras
----------------------------	-----	--

*Nota:* Las configuraciones fueron aplicadas al dispositivo de adquisición de imágenes para poder controlar una buena iluminación y resolución de imagen.

Fuente: Elaboración propia.

Después de haber aplicado el protocolo diseñado anteriormente fueron adquiridas 50 imágenes de las 10 especies de maderas diferentes cada una de ellas con un tamaño de 3024 x 4032 como se ve en las siguientes figuras.



*Figura 38.* Imágenes digitales adquiridas de especies de madera. a). Capinuri. b) Cumala. c) Catahua. d) Tornillo. Fuente: Elaboración propia.

### Proceso para generar imágenes

Para contar con una base de datos más amplia conformada por 100 imágenes por las 10 especies de madera que hacen un total de 1000 imágenes, se realizó el proceso de multiplicación de imágenes. Este proceso se realizó con la herramienta ImageJ en donde a cada imagen original se le aplicó una selección de tipo específica con los siguientes parámetros.

Tabla 9.

*Parámetros para la especificación de selección de la imagen con variación en la coordenada X*

<b>Ancho</b>	<b>Alto</b>	<b>Coordenada X</b>	<b>Coordenada Y</b>
403	15212	0	0
403	15212	403	0
403	15212	806	0
403	15212	1209	0
403	15212	1612	0
403	15212	2015	0
403	15212	2418	0
403	15212	2821	0
403	15212	3224	0
403	1512	3627	0

*Nota:* Los parámetros de selección de la imagen fueron establecidos de manera proporcional, estableciendo un valor fijo de alto y de ancho y la variación de la coordenada X la cual hace referencia a la posición superior de la imagen.

Fuente: Elaboración propia.

Tabla 10.

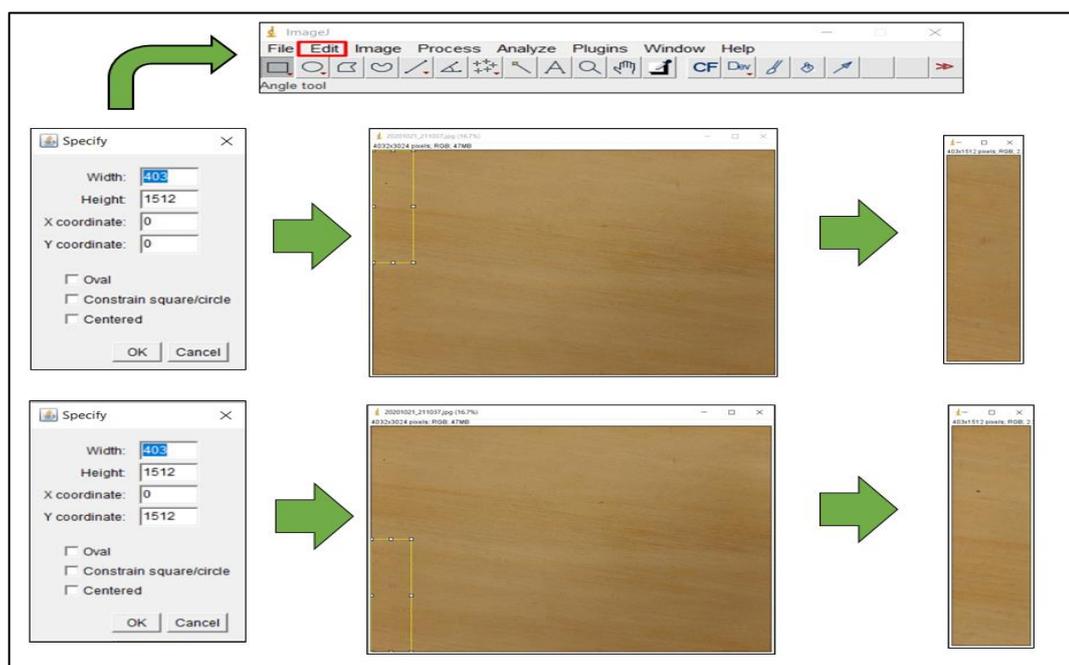
*Parámetros para la especificación de selección de la imagen con variación en la coordenada Y.*

<b>Ancho</b>	<b>Alto</b>	<b>Coordenada X</b>	<b>Coordenada Y</b>
403	15212	0	15212
403	15212	403	15212

403	15212	806	15212
403	15212	1209	15212
403	15212	1612	15212
403	15212	2015	15212
403	15212	2418	15212
403	15212	2821	15212
403	15212	3224	15212
403	1512	3627	1512

*Nota:* Los parámetros de selección de la imagen fueron establecidos de manera proporcional, estableciendo un valor fijo de alto y de ancho y la variación de la coordenada Y, la cual hace referencia a la posición inferior de la imagen. Fuente: Elaboración propia.

La representación visual del proceso se presenta en la siguiente imagen.



*Figura 39.* Representación gráfica del proceso de multiplicación de imágenes a raíz de una imagen original. Fuente: Elaboración propia.

Las imágenes quedaron organizadas en distintitas carpetas nombradas con cada una de las especies de madera y cada una de las imágenes fue nombrada con la numeración en la que fue tomada como se visualiza a continuación.

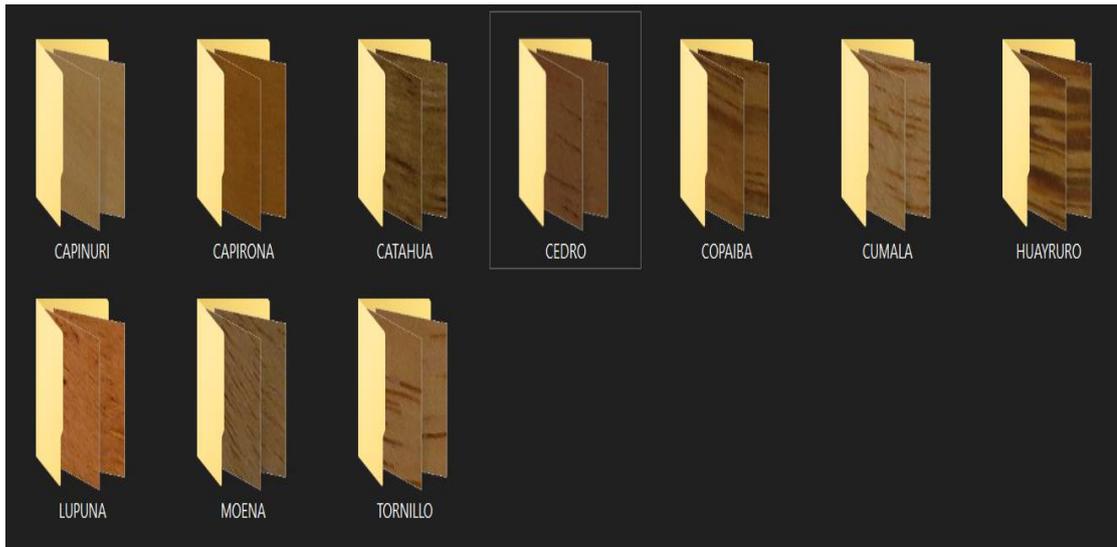


Figura 40. Representación de la organización de imágenes adquiridas, de acuerdo a su especie en específico. Fuente: Elaboración propia.

## II. Extracción de características.

Se realizó una búsqueda de trabajos de investigación anteriores enfocados en el estudio de la extracción de características por textura, esta búsqueda se realizó en repositorios científicos como IEEE Xplore, Scopus y ScienceDirect donde se obtuvieron trabajos relacionados a la ejecución de diversos métodos y algoritmos, basado en ellos se elaboró un cuadro comparativo de los resultados del rendimiento de cada algoritmo evaluados según cada investigador.

Tabla 11.

Cuadro representativo de los resultados de los trabajos de investigación de extracción de características por textura.

Investigaciones	Matriz de covarianza	GLCM	Filtros de Gabor	Filtro de Gabor + GLCM	wavelets	Packets wavelets	Bispectral features
(Tou et al., 2009)	x						
(Yusof et al., 2010)				x			

(Vautrot et al., 1996)	x	x	x
(Elunai et al., 2005)	x		x
(Wang et al., 2019)	x		
(Grigorescu et al., 2002)	x		
(Li et al., 2010)	x		

*Nota:* La comparación y selección de técnicas de extracción se realizó mediante el análisis de los resultados de trabajos de investigación previos.

Fuente: Elaboración propia.

Para la selección del algoritmo de extracción de características por textura se elaboró un cuadro comparativo de los diferentes algoritmos y/o métodos de extracción de características por textura, que se aplicaron de diversas investigaciones ya desarrolladas, en donde el cuadro comparativo nos muestra que los investigadores aplicaron el Filtro de Gabor con mayor frecuencia y obteniendo buenos resultados. A continuación, se muestra la función real de Gabor.

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right)$$

Donde:

$\gamma$  es la relación de aspecto espacial.

$\lambda$  es la longitud de onda de la función coseno en la función del núcleo de Gabor.

$\theta$  es la dirección de las bandas que están en forma paralela en el núcleo de Gabor.

$\varphi$  es la representación del parámetro de la fase de función coseno en la función del núcleo de Gabor.

Para iniciar la implementación del algoritmo seleccionado se utilizó con una base de datos de 1000 imágenes de maderas elaborada previamente, a las cuales se filtraron mediante el filtro de Gabor; para ello se importaron las siguientes librerías:

- Importación de la librería OpenCV y del módulo cv2 para realizar el llamado de las imágenes a las cuales se realizó el filtrado.
- Importación de la librería numpy para la realización de los cálculos numéricos y poder analizar los datos en gran volumen.
- Importación de la librería matplotlib.pyplot para poder trabajar con gráficos.
- Para la obtención de la ruta de ubicación de la base de datos y poder analizar imagen por imagen se aplicó la importación de la librería listdir y la librería isfile.

```

1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from os import listdir
5 from os.path import isfile, join

```

*Figura 41.* Importación de librerías para la implementación del Filtro de Gabor. Posterior a ello se procedió a la creación de las funciones; la primera función que se realizó fue la conversión de la imagen a escala de grises.

```

1 def BGR2GRAY(img):
2     # Escala de grises
3     gray = 0.2126 * img[..., 2] + 0.7152 * img[..., 1] + 0.0722 * img[..., 0]
4     return gray

```

*Figura 42.* Función para escala de grises.

La función def\_Gaborfilter permitió la creación del filtro con los siguientes parámetros de K\_size= 111, sigma =10, gamma=1.2, lamba=10 y el ángulo; en la línea número tres y cuatro se obtiene el tamaño medio del kernel y se prepara el kernel del Gabor en un array de ceros dependiendo del tamaño de X y de Y, y lo convierte en un valor flotante32, la línea número cuatro realiza el recorrido de cada valor mediante un bucle que recorre el tamaño de X y de Y en donde se calcula el centro de dicho valores para luego ser devueltos para el llenado

del Kernel y es ahí en donde se implementa la función real del filtro de Gabor y finalmente se realiza la normalización al Kernel obtenido.

```
1 def Gabor_filter(K_size=111, Sigma=10, Gamma=1.2, Lambda=10, Psi=0,
angle=0):
2     d = K_size // 2
3     gabor = np.zeros((K_size, K_size), dtype=np.float32)
4     for y in range(K_size):
5         for x in range(K_size):
6             px = x - d
7             py = y - d
8             theta = angle / 180. * np.pi
9             # obtener kernel x
10            _x = np.cos(theta) * px + np.sin(theta) * py
11            # obtener kernel y
12            _y = -np.sin(theta) * px + np.cos(theta) * py
13
14            # llenado del Kernel(parte real del filtro de Gabor)
15            gabor[y, x] = np.exp(-(_x**2 + Gamma**2 * _y**2) / (2 *
16            Sigma**2)) * np.cos(2*np.pi*_x/Lambda + Psi)
17
18            # Normalización del Kernel
19            gabor /= np.sum(np.abs(gabor))
20
21            return gabor
```

Figura 43. Función para definir el Filtro de Gabor.

La función `Gabor_filtering` se realizó para efectuar el filtrado de la imagen el cuál recibió los mismos parámetros de la función mencionada anteriormente, en la línea número 3 y 4 se obtiene la forma de la imagen en escala de grises y realizado el llenado del array, en la línea número 5 se prepara la imagen resultante o de salida mediante la creación de un array de ceros que es llenada con valores y los establece de tipo `float32`, para así poder procesarla. La quinta línea realiza el llamado a la función `def_Gabor_filter` definida anteriormente, a

partir de la línea 6 realiza el recorrido de la imagen mediante un bucle con los rangos de su alto y ancho producto de ello es la imagen filtrada (out), para finalmente obtener los valores y realizar un recorte o un límite de los valores dentro de la matriz en un intervalo de [0, 255].

```
1 def Gabor_filtering(gray, K_size=111, Sigma=10, Gamma=1.2,
Lambda=10, Psi=0, angle=0):
3 H, W = gray.shape
4 gray = np.pad(gray, (K_size//2, K_size//2), 'edge')
5 out = np.zeros((H, W), dtype=np.float32)
6 gabor = Gabor_filter(K_size=K_size, Sigma=Sigma, Gamma=Gamma,
Lambda=Lambda, Psi=0, angle=angle)
8 for y in range(H):
9     for x in range(W):
10         out[y, x] = np.sum(gray[y : y + K_size, x : x + K_size] * gabor)
11
12 out = np.clip(out, 0, 255)
13 out = out.astype(np.uint8)
14
15 return out
```

*Figura 44.* Función Gabor\_filtering para el proceso de filtrado a la imagen.

Función Gabor\_process fue definida para la extracción de características mediante 6 filtros de Gabor, cada una de ellas con diferentes orientaciones (0, 30, 60, 90, 120, 150) que se puede visualizar en la línea número 7, donde agrega los resultados del filtrado para la creación de un mapa de características de la imagen resultante.

```
1 def Gabor_process(img):
2     #obtener forma de la imagen
3     H, W, _ = img.shape
4     # escala de grises
5     gray = BGR2GRAY(img).astype(np.float32)
6     # definir ángulos
7     As = [0,30,60,90,120,150]
```

```

8     plt.subplots_adjust(left=0, right=1, top=1, bottom=0, hspace=0,
wspace=0.2)
9
10    out = np.zeros([H, W], dtype=np.float32)
11
12    # para cada ángulo
13    for i, A in enumerate(As):
14        # filtrado gabor
15        _out = Gabor_filtering(gray, K_size=11, Sigma=1.5, Gamma=1.2,
Lambda=3, angle=A)
16        # añadir imagen filtrada por cada gabor
17        out += _out
18
19    # escala de normalización
20    out = out / out.max() * 255
21    out = out.astype(np.uint8)
22
23    return out

```

*Figura 45.* Función `Gabor_process` para la creación de un mapa de características de una imagen resultante.

Finalmente definimos una función que pueda realizar el llamado a todas las imágenes y estas sean procesadas de manera secuencial es decir una a una, mediante el Filtro de Gabor y producto de ese proceso se generó un mapa de características de cada una de ellas. De cada mapa de características se le extrajo 3 características fundamentales: la media, la varianza y la desviación las cuales se fueron utilizadas en nuestros clasificadores.

```

1 def listar_directorio(ruta):
2     #lista el contenido de archivos de un directorio en específico
3     archivos = [a for a in listdir(ruta) if isfile(join(ruta, a))]
4     return archivos
5
6     ruta = r'C:\Users\USUARIO\Desktop\TESIS_PRUEBAS\BDMADERA'

```

```

7  listado_archivos = listar_directorio(ruta)
8  length = len(listado_archivos)
9  i = 0
10
11 while i < length:
12  listado_archivos[i] = listado_archivos[i]
13  img = cv2.imread("BDMADERA"+ listado_archivos[i]).astype(np.float32)
14
15  # hacemos el llamado a la imagen resultante del proceso de Gabor(mapa
de características)
16  out = Gabor_process(img)
17  print("Imagen ", i , " : "+listado_archivos[i])
18  i += 1
19
20  #Obtenemos las 3 características para nuestro clasificador
21  print("MEDIA")
22  media = out.mean()
23  print(media)
24
25  #varianza de los valores de la lista
26  print("Varianza")
27  varianza = out.var()
28  print(varianza)
29
30  #desviación de los valores de la lista
31  print("Desviación")
32  desviacion = out.std()
33  print(desviacion)
34  print("\n")

```

*Figura 46.* Función para que cada imagen sea procesada mediante el Filtro de Gabor y obtener 3 características del mapa de características.

Las 3 características obtenidas (media, varianza, desviación estándar) y la clase a la que pertenece la madera, se llenaron en un Excel en formato .csv el cual le enviamos a nuestros clasificadores.

```
1 archivo = open("bdmadera.csv","a")
2     archivo.write(str(media))
3     archivo.write(";")
4     archivo.write(str(varianza))
5     archivo.write(";")
6     archivo.write(str(desviacion))
7     archivo.write(";")
8     archivo.write("capinuri")
9     archivo.write("\n")
10  archivo.close()
```

Figura 47. Fragmento de código para almacenar todos los valores obtenidos en un archivo.csv

Al ejecutar todo el código del Filtro de Gabor se obtuvo lo siguiente:

```
In [1]: runfile('C:/Users/USUARIO/Desktop/TESIS_PRUEBAS/Gabor2.py', wdir='C:/
Users/USUARIO/Desktop/TESIS_PRUEBAS')
Imagen 0 : 20201021_211037-1.jpg
MEDIA
27.433985518662936
Varianza
135.14121366124473
Desviación
11.625025318735643

Imagen 1 : 20201021_211037-10.jpg
MEDIA
46.14263723134691
Varianza
454.60993741987136
Desviación
21.321583839383774

Imagen 2 : 20201021_211037-11.jpg
MEDIA
56.69923162261872
Varianza
637.0603894735344
Desviación
25.24005525892395
```

Figura 48. Características obtenidas del mapa de características de cada imagen. Fuente: Elaboración propia.

Tabla 12.

*Modelo del archivo .csv creado con las 3 características de cada imagen y la clase a la que pertenece. Ver tabla completa Anexo 3.*

<b>Imagen</b>	<b>Media</b>	<b>Varianza</b>	<b>Desviación</b>	<b>Clase</b>
Imagen 1	27.4339855	135.141214	11.6250253	capinuri
Imagen 2	46.1426372	454.609937	21.3215838	capinuri
Imagen 3	56.6992316	637.06039	25.2400553	capinuri
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
Imagen 1000	27.1545912	321.532724	17.9313336	tornillo

*Nota:* Las características fueron calculada usando Filtro de Gabor y la base de datos como entrada. Fuente: Elaboración propia.

### **III. Aplicar Algoritmos Supervisados.**

Para la selección de los clasificadores supervisados empleados se realizó una búsqueda de los algoritmos con los mejores resultados anteriormente utilizados en trabajos de investigación, esta búsqueda se realizó en los repositorios científicos como IEEE xplora, ScienceDirect, Scopus, logrando obtener trabajos relacionados a la implementación de algoritmos supervisados de clasificación, enfocado en ello se elaboró un cuadro comparativo de los resultados del rendimiento de cada algoritmo que fue evaluado por los investigadores.

Tabla 13.

*Cuadro representativo de los resultados de los trabajos de investigación con algoritmos supervisados.*

Investigaciones	CNN	SVM	KNN	BP	PNN	ANN	LDA	RF	CART	VGG16
(Kar Seng & Guniawan, 2018)	X									
(He et al., 2019)										X
(Nurthohari et al., 2019)		X								
(Wang et al., 2020)				X						
(Sukrisdyanto et al., 2019)		X	X				X	X	X	
(Tou et al., 2009)			X							
(Barmpoutis et al., 2018)		X								
(Wang et al., 2019)					X					
(Widyarto et al., 2017)						X				
(Cengil & Cinar, 2019)										X

*Nota:* La comparación y selección de algoritmos de clasificación se realizó mediante el análisis de los resultados de trabajos de investigación previos.

Fuente: Elaboración propia.

Para la selección de los algoritmos de clasificación se realizó un cuadro comparativo de los algoritmos de clasificación de imágenes aplicados en anteriores trabajos de investigación, de acuerdo con el cuadro comparativo los investigadores obtuvieron buenos resultados aplicando los clasificadores SVM (Support Vector Machine), Knn (Vecinos más cercano) y la red neuronal VGG16.

Para implementar el algoritmo de clasificación SVM se realizó la clasificación lineal, donde se muestra la función a continuación.

$$f(X) = W^T X + b = \langle W, X \rangle + b$$

Donde  $WT XJ + b = \pm 1$  actúan como los vectores de soporte y a su vez separación de clases. Para la implementación se llamaron a las librerías necesarias como Pandas para el manejo especializado de estructuras de datos, Numpy para realizar cálculos numéricos en grandes volúmenes de datos, también se importaron módulos de Sklearn, para entrenamiento de datos y las métricas para la evaluación de algoritmos y módulos de reporte de clasificación. La funcionalidad del algoritmo SVM se detalla a continuación, en la línea número ocho se realiza la importación de la base de datos en formato .csv, la línea diez y once realiza la división de los datos en atributos y etiquetas mientras que en la línea doce divide los datos en datos de entrenamiento de un 80% y datos de prueba de un 20%, la línea catorce y quince realizan la creación y entrenamiento de nuestro modelo que tiene como parámetro un kernel de tipo linear para realizar la clasificación de las clases, la línea diecisiete realiza una predicción y finalmente la línea dieciocho y diecinueve realiza una evaluación del algoritmo mediante la matriz de confusión y un reporte con algunas métricas de evaluación.

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from sklearn.model_selection import train_test_split
5 from sklearn.svm import SVC
6 from sklearn.metrics import classification_report
7
8 bankdata = pd.read_csv("maderas.csv")
9
10 X = bankdata.drop('clase', axis=1)
11 y = bankdata['clase']
12 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
13
```

```

14 svcclassifier = SVC(kernel='linear')
15 svcclassifier.fit(X_train,y_train)
16
17 y_pred = svcclassifier.predict(X_test)
18 print(confusion_matrix(y_test,y_pred))
19 print(classification_report(y_test, y_pred))

```

Figura 49. Algoritmo de Clasificación SVM (Support Vector Machine)

Como resultado final se obtuvo la matriz de confusión con respecto a la clasificación mediante el algoritmo SVM, como se muestra a continuación.

Tabla 14.

Matriz de confusión generada con algoritmo SVM.

Matriz de confusión SVM									
18	0	0	1	0	3	0	0	2	0
0	8	4	0	1	5	0	2	3	0
0	1	2	0	0	4	0	2	5	0
0	0	2	10	0	1	5	1	0	4
4	3	0	0	7	1	0	5	3	0
3	1	3	0	0	6	0	5	2	0
0	0	0	8	2	0	11	0	0	3
0	3	0	0	0	4	0	3	5	0
5	2	0	0	0	0	0	8	2	0
0	0	2	0	5	4	0	1	0	5

Nota: Los valores de la matriz de confusión fueron calculados usando la misma base de datos de entrenamiento y prueba. Fuente: Elaboración propia.

Se procedió con la implementación del segundo algoritmo de clasificación KNN (Vecinos más cercanos), donde se aprecia la función matemática de clasificación a continuación.

$$\hat{f}(\chi) \leftarrow \arg \max_{v \in V} \sum_{i=1}^k [v = f(x_i)]$$

A partir de la línea 1 a la línea número 7 se realizó la importación de las librerías y módulos necesarios, la línea 9 realiza la importación de la base de datos en formato .csv, la línea once y doce realizan la división de los datos en atributos y etiquetas, mientras que en la línea trece divide los datos en datos de entrenamiento en un 80% y datos de prueba en un 20%, luego se procede a la creación y entrenamiento de nuestro modelo con el parámetro de KNN =10 debido al número de clases todo ello en la línea veinte y veintiuno, con la línea veintitrés se procede a realizar las predicciones para finalmente con la línea veinticinco y veintiséis evaluar el algoritmo mediante la matriz de confusión y un reporte con las métricas de evaluación.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 from sklearn.model_selection import train_test_split
5 from sklearn.preprocessing import StandardScaler
6 from sklearn.neighbors import KNeighborsClassifier
7 from sklearn.metrics import classification_report, confusion_matrix
8
9 dataset = pd.read_csv("maderas.csv")
10
11 X = dataset.iloc[:, :-1].values
12 y = dataset.iloc[:, 3].values
13 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
14
15 scaler = StandardScaler()
16 scaler.fit(X_train)
17 X_train = scaler.transform(X_train)
18 X_test = scaler.transform(X_test)
19
20 classifier = KNeighborsClassifier(n_neighbors=10)
21 classifier.fit(X_train,y_train)
22
```

```

23 y_pred = classifier.predict(X_test)
24
25 print(confusion_matrix(y_test, y_pred))
26 print(classification_report(y_test,y_pred))

```

Figura 50. Algoritmo de clasificación KNN (Vecinos más cercanos).

Con la ejecución del algoritmo KNN se obtuvo como resultado final la matriz de confusión la cual se muestra a continuación.

Tabla 15.

Matriz de confusión generada con algoritmo KNN.

Matriz de confusión KNN									
19	0	0	0	0	2	0	0	1	0
0	10	2	0	0	3	0	8	1	0
2	0	5	0	0	8	0	1	4	0
0	0	0	8	1	0	3	0	0	6
3	0	1	6	8	0	0	4	0	0
1	0	4	0	0	5	0	1	3	0
0	0	0	8	0	0	14	2	0	1
3	2	0	5	1	0	0	8	0	0
3	4	0	0	0	4	0	2	6	0
0	0	0	6	0	0	3	0	0	8

*Nota:* Los valores de la matriz de confusión fueron calculados usando la misma base de datos de entrenamiento y prueba. Fuente: Elaboración propia.

Para la implementación del tercer algoritmo de clasificación de la red neuronal VGG16 se realizó lo siguiente, desde la línea 1 de código a la línea 16 se importó las librerías, modelos y funciones de Keras necesarias para la implementación del modelo.

```

1 from keras.models import Sequential, Model
2 from keras.layers import Conv2D, MaxPool2D, Dense, Flatten, Dropout,
BatchNormalization, Input

```

```

3 from keras.optimizers import Adam
4 from keras.callbacks import TensorBoard, ModelCheckpoint
5 from keras.utils import np_utils
6 import os
7 import numpy as np
8 from keras.preprocessing import image
9 from keras.applications.imagenet_utils import preprocess_input,
decode_predictions
10 from keras.applications.vgg16 import VGG16
11 from keras.preprocessing.image import ImageDataGenerator
12 from sklearn.utils import shuffle
13 from sklearn.model_selection import train_test_split
14 import cv2
15 import matplotlib.pyplot as plt
16 %matplotlib inline

```

*Figura 51.* Importación de librerías, paquetes para la implementación de la red neuronal VGG16.

Posterior a ello se realizó la configuración de parámetros de entrenamiento y del sistema donde la línea uno y dos configuran el ancho y alto de la imagen, la línea tres configura el número de clases, la línea cuatro y cinco el número de épocas y el tamaño del `batch_size` del entrenamiento del sistema. Se procede a definir los directorios de entrenamiento y validación como se detalla en la línea siete y ocho del código.

```

1 width_shape = 224
2 height_shape = 224
3 num_classes = 10
4 epochs = 50
5 batch_size = 32
6 train_data_dir='C:/Users/USUARIO/Desktop/TESIS_PRUEBAS/
BDMADERA /train'
7 validation_data_dir = ' C:/Users/USUARIO/Desktop/TESIS_PRUEBAS/
BDMADERA /valid'

```

*Figura 52.* Configuración de parámetros de entrenamiento y del sistema y asignación de directorios de la base de datos.

Se realizó la configuración del generador de imágenes para los datos de imágenes de entrenamiento y validación asignándoles los parámetros de zoom, desplazamiento horizontal y vertical y la función del pre procesamiento todo ello detallado en la línea uno hasta la línea once, posterior a ello en la línea trece hasta la línea veintitrés se realizó la asignación de parámetros anteriormente definidos como son la ubicación de las imágenes, tamaño de las imágenes , tamaño de batch\_size y el modelo de clase categorial ya que se contó con 10 clases.

```
1 train_datagen = ImageDataGenerator(  
2     zoom_range=0.2,  
3     horizontal_flip=True,  
4     vertical_flip=True,  
5     preprocessing_function=preprocess_input)  
6  
7 valid_datagen = ImageDataGenerator(  
8     zoom_range=0.2,  
9     horizontal_flip=True,  
10    vertical_flip=True,  
11    preprocessing_function=preprocess_input)  
12  
13 train_generator = train_datagen.flow_from_directory(  
14    train_data_dir,  
15    target_size=(width_shape, height_shape),  
16    batch_size=batch_size,  
17    class_mode='categorical')  
18  
19 validation_generator = valid_datagen.flow_from_directory(  
20    validation_data_dir,  
21    target_size=(width_shape, height_shape),  
22    batch_size=batch_size,
```

```
23 class_mode='categorical')
```

Figura 53. Configuración del generador de imágenes para los datos de imágenes de entrenamiento y validación.

Creación de nuestro modelo VGG16 para entrenar nuestro sistema en la línea uno se definió la estructura de imagen de entrada , en la línea tres se carga el modelo VGG16 con los pesos del dataset="imagenet", posteriormente en la línea siete se toma la capa "block5\_pool" ,con la línea ocho hasta la línea diez se agregó 2 capas densas de 128 con activación "relu" y con la línea once se agrega una capa densa que tiene como salida el número de clases =10,luego con la línea dieciséis y diecisiete recorreremos las capas para solo entrenar las anteriormente creadas, finalmente la línea veintiuno define la función para entrenamiento, optimizador ="adam" y las métricas.

```
1 image_input = Input(shape=(width_shape, height_shape, 3))
2
3 model2 = VGG16(input_tensor=image_input, include_top=True, weights='imagenet')
4
5 model2.summary()
6
7 last_layer = model2.get_layer('block5_pool').output
8 x = Flatten(name='flatten')(last_layer)
9 x = Dense(128, activation='relu', name='fc1')(x)
10 x = Dense(128, activation='relu', name='fc2')(x)
11 out = Dense(num_classes, activation='softmax', name='output')(x)
12 custom_model = Model(image_input, out)
13 custom_model.summary()
14
15 # freeze all the layers except the dense layers
16 for layer in custom_model.layers[:-3]:
17     layer.trainable = False
18
19 custom_model.summary()
```

```

20
21 custom_model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
22
23 model_history = custom_model.fit_generator(
24     train_generator,
25     epochs=epochs,
26     validation_data=validation_generator,
27     steps_per_epoch=nb_train_samples//batch_size,
28     validation_steps=nb_validation_samples//batch_size)

```

Figura 54. Creación, compilación y entrenamiento del modelo VGG16.

El siguiente fragmento de código se realizó para la predicción, en la línea uno y dos se importaron las librerías necesarias para la predicción y la carga de datos del modelo, en la línea 4 se realizó la creación del vector con cada una de las clases que contiene el dataset, en la línea seis se cargó el modelo ya entrenado, la línea ocho importa el dataset de prueba mediante la ubicación, en la línea 9 cargamos la imagen y posterior a ello se le cambia el espacio de color de BGR a RGB con la línea 10, la línea 11 redimensiona la imagen para que tenga la misma dimensión que las imágenes de entrenamiento, la línea 12 y 13 pasa las imágenes a un array y son procesadas, en la línea 14 se realizó el shape del array para que se ajuste al de la entrada del modelo para finalmente realizar la predicción con las líneas de código restantes.

```

1 from keras.applications.imagenet_utils import preprocess_input, decode_predictions
2 from keras.models import load_model
3
4 names = ['CAPINURI', 'CAPIRONA', 'CATAHUA', 'CEDRO', 'COPAIBA', 'CUM
ALA', 'HUAYRURO', 'LUPUNA', 'MOENA', 'TORNILLO']
5
6 modelt = load_model("/content/drive/MyDrive/MODELO2_VGG16/modeloo
_VGG16.h5")
7

```

```

8 imaget_path = "/content/drive/MyDrive/NUEVA_BASE_MADERAS/PRUEB
AS/CATAHUA/1.jpg"
9 image = cv2.imread(imaget_path)
10 imaget = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
11 imaget=cv2.resize(imaget, (width_shape, height_shape), interpolation = c
v2.INTER_AREA)
12 xt = np.asarray(imaget)
13 xt=preprocess_input(xt)
14 xt = np.expand_dims(xt,axis=0)
15 preds = modelt.predict(xt)
16
17 print(names[np.argmax(preds)])
18 plt.imshow(cv2.cvtColor(np.asarray(image),cv2.COLOR_BGR2RGB))
19 plt.axis('off')
20 plt.show()

```

*Figura 55.* Código para la predicción con el modelo VGG16.

El siguiente fragmento de código se emplearon métricas para poder generar la matriz de confusión con diversos indicadores.

```

from sklearn.metrics import confusion_matrix, f1_score, roc_curve, precision
_score, recall_score, accuracy_score, roc_auc_score
from sklearn import metrics
from mlxtend.plotting import plot_confusion_matrix
from keras.models import load_model
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

test_data_dir = '/content/drive/MyDrive/NUEVA_BASE_MADERAS/PRUEB
AS'

test_datagen = ImageDataGenerator(preprocessing_function=preprocess_in
put)

```

```

test_generator = test_datagen.flow_from_directory(
    test_data_dir,
    target_size=(width_shape, height_shape),
    batch_size = 1,
    save_to_dir='/content',
    class_mode='categorical', shuffle = False)

predictions = modelt.predict_generator(generator=test_generator)

y_pred = np.argmax(predictions, axis=1)
y_real = test_generator.classes

matc=confusion_matrix(y_real, y_pred)

plot_confusion_matrix(conf_mat=matc, figsize=(10,10), show_normed=False)
plt.tight_layout()

print(metrics.classification_report(y_real,y_pred, digits = 4))

```

*Figura 56.* Fragmento de código para generar matriz de confusión y métricas de evaluación del modelo VGG16.

Con la ejecución de la red neuronal VGG16 se obtuvo como resultado final la matriz de confusión la cual se muestra a continuación.

Tabla 16.

*Matriz de confusión generada con la red neuronal VGG16.*

<b>Matriz de confusión</b>									
5	0	0	0	0	0	0	0	0	0
0	5	0	0	0	0	0	0	0	0
0	0	4	0	0	0	0	0	1	0
0	0	0	5	0	0	0	0	0	0
0	0	0	0	5	0	0	0	0	0
0	0	0	0	0	5	0	0	0	0
0	0	0	0	0	0	5	0	0	0
0	0	0	0	0	0	0	5	0	0
0	0	0	0	0	0	0	0	5	0
0	0	0	0	0	0	0	0	0	5

*Nota:* Los valores de la matriz de confusión fueron calculados usando la misma base de datos de entrenamiento y prueba. Fuente: Elaboración propia

## **IV. CONCLUSIONES Y RECOMENDACIONES**

### **4.1. Conclusiones.**

- a) Se elaboró un dataset siguiendo un protocolo para la adquisición de las imágenes digitales, siendo necesario realizar un proceso de multiplicación de imágenes, debido que al inicio se contaba con una muestra muy pequeña. Todo se realizó para aumentar el tamaño de muestras y conseguir mejor información para el proceso de clasificación.
- b) Se seleccionaron los filtros de extracción en base a los trabajos previos descritos en la tabla 11 y finalmente se decidió optar por la implementación del Filtro de Gabor, la evidencia de este acápite fue la obtención de características de textura por cada imagen. Lo mencionado se detalla en la tabla 12.
- c) Se seleccionaron los algoritmos de clasificación en base a los trabajos previos descritos en la tabla 13 y finalmente se decidió optar por la implementación de los algoritmos SVM, KNN y la red neuronal VGG16, la evidencia de este acápite fue la matriz de confusión de cada clasificador. Lo mencionado se detalla en la tabla 14, tabla 15 y la tabla 16.
- d) Se evaluaron los algoritmos mediante un dataset de entrenamiento y a la vez de prueba, donde se logró evaluar el desempeño de cada clasificador. La red neuronal VGG16 es el algoritmo con mayor nivel de precisión con un 98.33%, exhaustividad con un 98.00%, pero con un mayor tiempo de respuesta de 250.3s y que presenta un mayor consumo de recursos frente a los algoritmos SVM y KNN.

### **4.2. Recomendaciones.**

- a) Se recomienda realizar una base de datos a base de imágenes estandarizada para que los algoritmos de clasificación tengan los mismos valores y/o condiciones y obtener resultados mucho más confiables.

- b) Se recomienda aplicar y/o añadir diferentes extractores de características para así brindarle información más consistente a los clasificadores para aumentar la precisión en tanto a los resultados.

## REFERENCIAS.

- AITIM. (2016). *Especies de madera*. AITIM. <https://infomadera.net/modulos/maderas.php?id=177&madera=Muiratinga>
- Araki, T., Ikeda, N., Dey, N., Acharjee, S., Molinari, F., Saba, L., Godia, E. C., Nicolaidis, A., & Suri, J. S. (2015). Shape-Based Approach for Coronary Calcium Lesion Volume Measurement on Intravascular Ultrasound Imaging and Its Association With Carotid Intima-Media Thickness. *Journal of Ultrasound in Medicine*, 34(3), 469–482. <https://doi.org/10.7863/ultra.34.3.469>
- Asmara, R., Hamidi, Y., & Puspitasari, D. (2017). Tropical hardwood species identification based on first order statistical moment of cross section images color and texture extraction using gray level co-occurrence matrix. *2017 11th International Conference on Information & Communication Technology and System (ICTS)*, 7–12. <https://doi.org/10.1109/ICTS.2017.8265598>
- Barmpoutis, P., Dimitropoulos, K., Barboutis, I., Grammalidis, N., & Lefakis, P. (2018). Wood species recognition through multidimensional texture analysis. *Computers and Electronics in Agriculture*, 144, 241–248. <https://doi.org/10.1016/j.compag.2017.12.011>
- Brown, D., Schreckenber, K., Bird, N., Cerutti, P., Del Gato, F., Diaw, C., Fomété, T., Luttrell, C., Navarro, G., Oberndorf, R., Thiel, H., & Wells, A. (2010). *Madera legal: Verificación y gobernanza en el sector forestal* (CATIE (ed.); 1st ed.).
- Burger, W., & Burge, M. J. (2016). Digital Image Processing an Algorithmic Introduction Using Java. In D. Gries & F. Barry (Eds.), *Springer* (2nd ed., Vol. 24, Issue 11). <https://doi.org/10.1007/978-1-4471-6684-9>
- Callo, J. C. (2014). Determinación de las Propiedades Físicas y Características Anatómicas de la Especie *Aspidosperma subincanum* MART. & Zucc. (Quillobordon) y *Beilschmiedia towarensis* (Meisn.) Sa. Nishida (Palta Moena) Provenientes del Distrito las Piedras - Provincia de Tamb. *Tesis Título*, 1–139.
- Castillo, A. (2010). Manual dendrológico de las principales especies de interés comercial actual y potencial de la zona del Alto Huallaga. In W. Nalvarte (Ed.), *Camara Nacional Forestal* (Vol. 06). <http://www.cnf.org.pe/enero011/MD.pdf>
- CEMU. (2017a). *Especie de madera Capirona*. <http://www.cemuperu.com/categoria/tornillo/>

- CEMU. (2017b). *Especie de madera Catahua*. CEMU.  
<https://www.cemuperu.pe/#especies>
- CEMU. (2017c). *Especie de madera Copaiba*. CEMU.  
<https://www.cemuperu.pe/#especies>
- CEMU. (2017d). *Especie de madera Cumala*. CEMU.  
<https://www.cemuperu.pe/#especies>
- CEMU. (2017e). *Especie de madera Lupuna*. CEMU.  
<https://www.cemuperu.pe/#especies>
- CEMU. (2017f). *Especie de madera Tornillo*. CEMU.  
<https://www.cemuperu.pe/#especies>
- Cengil, E., & Cinar, A. (2019). Multiple classification of flower images using transfer learning. *2019 International Conference on Artificial Intelligence and Data Processing Symposium, IDAP 2019*.  
<https://doi.org/10.1109/IDAP.2019.8875953>
- Chaki, J., & Dey, N. (2018). *A BEGINNER'S GUIDE TO IMAGE PREPROCESSING TECHNIQUES* (CRC Press (ed.); 1st ed., Vol. 1, Issue 4). CRC Press.
- Chazallet, S. (2016). *Python 3: los fundamentos del lenguaje* (eni (ed.); 2nd ed.). Recursosinformáticos. <https://www.ediciones-eni.com/open/mediabook.aspx?idR=9103588b3f4c184b8f2497bcf75605a9>
- Clow, M. (2018). Visual Studio Code. In *Angular 5 Projects* (pp. 57–68).  
[https://doi.org/10.1007/978-1-4842-3279-8\\_5](https://doi.org/10.1007/978-1-4842-3279-8_5)
- Confederación Peruana de la Madera. (2008). *Compendio de Información Técnica de 32 especies forestales Tomo I* (2nd ed.). CITEmadera.  
<https://cdn.www.gob.pe/uploads/document/file/1458588/1>. Compendio de información técnica de 32 especies Forestales Tomo I.pdf.pdf
- El Comercio. (2018). *Incautan presunto cargamento ilegal de madera en Ucayali Ucayali*. El Comercio. <https://elcomercio.pe/peru/ucayali/incautan-cargamento-ilegal-madera-ucayali-noticia-498205-noticia/>
- Elunai, R., Chandran, V., & Sridharan, S. (2005). Texture classification using Gabor energy features and higher order spectral features: A comparative study. *Proceedings - 8th International Symposium on Signal Processing and Its Applications, ISSPA 2005*, 2(4), 659–662.  
<https://doi.org/10.1109/ISSPA.2005.1581024>

- Expreso. (2018). *Perú: tala ilegal y tráfico de madera - Expreso*. Palacios, Alfredo. <https://www.expreso.com.pe/opinion/alfredo-palacios-dongo/peru-tala-ilegal-y-trafico-de-madera/>
- Giron, J. (2013). *Algoritmos (Ed. Business Analytics)* (1st ed.). UOC.
- Gonzales, I. (2011). *Atlas de maderas selva central*. Universidad Nacional del Centro del Perú.
- Gonzalez, R., & Woods, R. (2009). Digital Image Processing, Third Edition. In *Journal of Biomedical Optics* (3rd ed., Vol. 14, Issue 2). Pearson International Edition. <https://doi.org/10.1117/1.3115362>
- Grigorescu, S. E., Petkov, N., & Kruizinga, P. (2002). Comparison of texture features based on Gabor filters. *IEEE Transactions on Image Processing* ( Volume: 11, Issue: 10, Oct. 2002), 11(10), 8. <https://doi.org/10.1109/TIP.2002.804262>
- Guevara, C., & GIL, J. (2018). *Análisis Y Determinación De Enfermedades En El Fruto Del Mango Utilizando Técnicas De Procesamiento Digital De Imágenes* [Pedro Ruiz Gallo]. <http://repositorio.unprg.edu.pe/bitstream/handle/UNPRG/2571/BC-TES-TMP-1444.pdf?sequence=2&isAllowed=y>
- He, T., Liu, Y., Xu, C., Zhou, X., Hu, Z., & Fan, J. (2019). A fully convolutional neural network for wood defect location and identification. *IEEE Access*, 7, 123453–123462. <https://doi.org/10.1109/ACCESS.2019.2937461>
- Kar Seng, L., & Guniawan, T. (2018). An Experimental Study on the Use of Visual Texture for Wood Identification Using a Novel Convolutional Neural Network Layer. *2018 8th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 4. <https://doi.org/10.1109/ICCSCE.2018.8685006>
- KerasSIG. (2020). *About Keras*. Keras Special Interest Group. <https://keras.io/about/>
- Li, W., Mao, K. Z., Zhang, H., & Chai, T. (2010). Designing compact gabor filter banks for efficient texture feature extraction. *11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010, December*, 1193–1197. <https://doi.org/10.1109/ICARCV.2010.5707806>
- Liu, S., Jiang, W., Wu, L., Wen, H., Liu, M., & Wang, Y. (2020). Real-Time Classification of Rubber Wood Boards Using an SSR-Based CNN. *IEEE Transactions on Instrumentation and Measurement*, 69(11), 9.

- <https://doi.org/10.1109/TIM.2020.3001370>
- Loke, K. S. (2018). Texture recognition using a novel input layer for deep convolutional neural network. *2018 IEEE 3rd International Conference on Communication and Information Systems, ICCIS 2018*, 14–17. <https://doi.org/10.1109/ICOMIS.2018.8645055>
- Mantilla, L. (2019). Classification of satellite images using Rp fuzzy c means for unsupervised classification algorithm. *2019 IEEE Colombian Conference on Applications in Computational Intelligence (CoCACI)*, 1–5. <https://doi.org/10.1109/CoCACI.2019.8781988>
- Numpy. (2021). *What is Numpy?* Numpy. <https://numpy.org/doc/stable/user/whatisnumpy.html>
- Nurthohari, Z., Murti, M. A., & Setianingsih, C. (2019). Wood quality classification based on texture and fiber pattern recognition using HOG feature and SVM classifier. *Proceedings - 2019 IEEE International Conference on Internet of Things and Intelligence System, IoTaIS 2019*, 2011, 123–128. <https://doi.org/10.1109/IoTais47347.2019.8980414>
- OSINFOR. (2015). *Fichas De Identificacion De Especies Forestales Maderables De La Selva Central* (R. Rojas (ed.); 1st ed.). OSINFOR. [www.osinfor.gob.pe](http://www.osinfor.gob.pe)
- Perú Forestal. (2020a). CEDRO. Perú Forestal. <https://www.peruforestal.org/cedro/>
- Perú Forestal. (2020b). *HUAYRURO*. Perú Forestal. <https://www.peruforestal.org/huayruro/>
- Perú Forestal. (2020c). *MOENA*. Perú Forestal. <https://www.peruforestal.org/mohenablanca/>
- Pozzer, J., & Guzowski, E. (2011). *Madera Materiales y materias primas* (M. de Educación & I. N. de E. Tecnológica (eds.); 3rd ed.). <http://www.inet.edu.ar/wp-content/uploads/2012/11/madera.pdf>
- Presa, J. (2014). *Cedria Lasures y barnices* (CEDRIA (ed.); 1st ed.). CEDRIA.
- Protect Aqua. (2015). *What is wood?* Protect Aqua. <https://protectaqua.com/Page/7/What-is-wood>
- Reynel, C., Pennington, R., Pennington, T., Flores, C., & Daza, A. (2003). *Árboles útiles de la Amazonía Peruana*. Lima Tarea Gráfica Educativa.
- Ristiawanto, V., Irawan, B., & Setianingsih, C. (2019). Wood classification with transfer learning method and bottleneck features. *2019 International*

- Conference on Information and Communications Technology, ICOIACT 2019*, 111–116. <https://doi.org/10.1109/ICOIACT46704.2019.8938428>
- SciPy. (2021). *Introduction SciPy. SciPy.* <https://docs.scipy.org/doc/scipy/reference/tutorial/general.html>
- Sonka, M., Hlavac, V., & Boyle, R. (1993). *Image Processing, Analysis and Machine Vision* (1st ed.). Springer. <https://doi.org/10.1007/978-1-4899-3216-7>
- Sukrisdyanto, O. D., Purnama, I. K. E., & Nugroho, S. M. S. (2019). Wood Strength Classification Based on RGB Color and Image Texture Using KNN Method. *2019 International Seminar on Intelligent Technology and Its Application, ISITIA*, 360–365. <https://doi.org/10.1109/ISITIA.2019.8937239>
- Sundaram, M., Abitha, J., Mal Mathan Raj, R., & Ramar, K. (2015). Wood species classification based on local edge distributions. *Optik*, 126(21), 2884–2890. <https://doi.org/10.1016/j.ijleo.2015.07.044>
- Taqyudin, M. I., Irawan, B., & Setianingsih, C. (2019). Wood Classification Based on Fiber Texture Using Backpropagation Method. *ICSECC 2019 - International Conference on Sustainable Engineering and Creative Computing: New Idea, New Innovation, Proceedings*, 245–250. <https://doi.org/10.1109/ICSECC.2019.8907197>
- Tosi, S. (2009). *Matplotlib for Python Developers*. In L. D'Silva (Ed.), *Packt Publishing* (1st ed.). <http://www.amazon.com/Matplotlib-Python-Developers-Sandro-Tosi/dp/1847197906>
- Tou, J. Y., Tay, Y. H., & Lau, P. Y. (2009). A Comparative Study for Texture Classification Techniques on Wood Species Recognition Problem. *2009 Fifth International Conference on Natural Computation*, 5, 8–12. <https://doi.org/10.1109/ICNC.2009.594>
- Trujillo, M., & Heinrichs, W. (2017). *COMPRAS RESPONSABLES DE MADERA EN EL PERÚ*. FSC Perú.
- Urrunaga, J., Johnson, A., & Orbegozo Sánchez, I. D. (2018). *El momento de la verdad oportunidad o amenaza para la amazonía peruana en la lucha contra el comercio de la madera ilegal* (Enviroment). <https://www.actualidadambiental.pe/la-evidencia-sobre-tala-ilegal-corrupcion-y-lavado-de-madera-es-abrumadora/>
- Vautrot, P., Bonnet, N., & Herbin, M. (1996). Comparative study of different

- spatial/spatial-frequency methods (Gabor filters, wavelets, wavelets packets) for texture segmentation/classification. *IEEE International Conference on Image Processing*, 3, 145–148. <https://doi.org/10.1109/icip.1996.560391>
- Viera, G. (2017). Procesamiento de imágenes usando OpenCV aplicado en Raspberry Pi para la clasificación del cacao. In *Thesis*. [https://pirhua.udel.edu.pe/bitstream/handle/11042/2916/IME\\_218.pdf?sequence=1&isAllowed=y](https://pirhua.udel.edu.pe/bitstream/handle/11042/2916/IME_218.pdf?sequence=1&isAllowed=y)
- Wang, H., Wang, H., Yu, W., & Li, H. (2019). Research on wood species recognition method based on hyperspectral image texture features. *Proceedings - 2019 4th International Conference on Mechanical, Control and Computer Engineering, ICMCCE 2019*, 413–416. <https://doi.org/10.1109/ICMCCE48743.2019.00099>
- Wang, H., Yukun, C., Wang, H., & Hui, L. (2020). Research on Wood Panel Recognition Based on Digital Visual Features. *2020 International Conference on Computer Engineering and Application (ICCEA)*, 1, 0–3. <https://doi.org/10.1109/ICCEA50009.2020.00096>
- Widyarto, S., Suryasa, I. N., Fajarianto, O., bin Abdullah, K. A., Rahim, M. S. M., Priyandoko, G., & Budaya, G. A. (2017). Wood texture detection with conjugate gradient neural network algorithm. *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 1–6. <https://doi.org/10.1109/EECSI.2017.8239118>
- Wood Solutions. (2019). *Madera aserrada*. Wood Solutions. <https://www.woodsolutions.com.au/wood-product-categories/sawn-timber>
- Yang, J., Huang, P., Dai, F., Sun, Y., Wang, L., & Bi, H. (2019). Application of Deep Learning in Wood Classification. *2019 IEEE International Conference on Computer Science and Educational Informatization, CSEI 2019*, 124–129. <https://doi.org/10.1109/CSEI47661.2019.8938960>
- Yusof, R., Khairuddin, U., Rosli, N. R., Abdul Ghafar, H., Aizuddin Nik Azmi, N. M., Ahmad, A., & Khairuddin, A. S. (2018). A Study of Feature Extraction and Classifier Methods for Tropical Wood Recognition System. *IEEE Region 10 Annual International Conference, Proceedings/TENCON, 2018-October(October)*, 2034–2039. <https://doi.org/10.1109/TENCON.2018.8650411>
- Yusof, R., Rosli, N. R., & Khalid, M. (2010). Using Gabor Filters as Image Multiplier for Tropical Wood Species Recognition System. *2010 12th International*

*Conference on Computer Modelling and Simulation*, 289–294.  
<https://doi.org/10.1109/UKSIM.2010.61>

## ANEXOS.

### Anexo 1. Resolución de aprobación del proyecto de investigación



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO  
RESOLUCIÓN N°0669-2021/FIAU-USS**

Pimentel, 14 de julio de 2021

**VISTO:**

El Acta de reunión N° 2206-2021 del Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS, remitido el 24 de junio de 2021 mediante oficio N° 02-41-2021/FIAU-IS-USS, para la modificación del Tema de la Tesis: "COMPARACIÓN DE ALGORITMOS DE APRENDIZAJE NO SUPERVISADO PARA LA CLASIFICACIÓN AUTOMÁTICA DE MADERA ASERRADA MEDIANTE PROCESAMIENTO DE IMÁGENES DIGITALES", presentada por el(los) testista(s) MIO ROJAS POOL CRISTOPHER, del Programa de estudios INGENIERÍA DE SISTEMAS, y;

**CONSIDERANDO:**

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y tesis con aprobados por el Comité de Investigación y derivados a la facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El periodo de vigencia de los mismos será de dos años, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; es individual o en pares para obtener un título profesional. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C."

Que, según documento de Visto, el Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS, acuerda aprobar la modificación del tema de la tesis aprobada mediante Resolución N° 1817-2019/FIAU-USS del 09 de diciembre de 2019, denominada "COMPARACIÓN DE ALGORITMOS DE APRENDIZAJE NO SUPERVISADO PARA LA CLASIFICACIÓN AUTOMÁTICA DE MADERA ASERRADA MEDIANTE PROCESAMIENTO DE IMÁGENES DIGITALES" por "COMPARACIÓN DE ALGORITMOS DE APRENDIZAJE SUPERVISADO PARA LA CLASIFICACIÓN AUTOMÁTICA DE MADERA ASERRADA MEDIANTE PROCESAMIENTO DE IMÁGENES DIGITALES" de la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE, a cargo de MIO ROJAS POOL CRISTOPHER en condición de estudiante(s), del Programa de estudios INGENIERÍA DE SISTEMAS.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

**SE RESUELVE:**

**ARTÍCULO 1°:** MODIFICAR, el tema de la Tesis denominada "COMPARACIÓN DE ALGORITMOS DE APRENDIZAJE NO SUPERVISADO PARA LA CLASIFICACIÓN AUTOMÁTICA DE MADERA ASERRADA MEDIANTE PROCESAMIENTO DE IMÁGENES DIGITALES" por "COMPARACIÓN DE ALGORITMOS DE APRENDIZAJE SUPERVISADO PARA LA CLASIFICACIÓN AUTOMÁTICA DE MADERA ASERRADA MEDIANTE PROCESAMIENTO DE IMÁGENES DIGITALES", perteneciente a la línea de investigación de INFRAESTRUCTURA,

**FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO  
RESOLUCIÓN N°0669-2021/FIAU-USS**

Pimentel, 14 de julio de 2021

**TECNOLOGÍA Y MEDIO AMBIENTE**, a cargo de MIO ROJAS POOL CRISTOPHER en condición de estudiante(s), del Programa de estudio: **INGENIERÍA DE SISTEMAS**.

**ARTÍCULO 2°: MODIFICAR**, la Resolución de Facultad con la que se asignó Aceror especialista y la Resolución de Facultad con la que se asignó Jurado evaluador, en el extremo del tema de la tesis quedando tal como se indica en el artículo 1° de la presente Resolución.

**ARTÍCULO 3°: DEJAR SIN EFECTO**, toda Resolución emitida por la Facultad que se oponga a la presente Resolución.

**REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE**

  
  
Dr. Mario Fernando Torres Morón  
Decano - Facultad de Ingeniería,  
Arquitectura y Urbanismo  
UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.

  
  
MSc. María Soledad Gallo Rivera  
Secretaria Académica / Facultad de Ingeniería,  
Arquitectura y Urbanismo  
UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.

Cc: Interesado, Archivo

Anexo 2. Formato de ficha de evaluación de algoritmos para clasificación de especies de madera.

	Algoritmo SVM						Algoritmo KNN						algoritmo VGG16					
	precisión	recall	roc-area	Tiempo de respuesta	Consumo de Ram	Consumo CPU	precisión	recall	roc-area	Tiempo de respuesta	Consumo de Ram	Consumo CPU	precisión	recall	roc-area	Tiempo de respuesta	Consumo de Ram	Consumo CPU
capinuri																		
capirona																		
catahua																		
cedro																		
copaiba																		
cumala																		
huayruro																		
lupuna																		
moena																		
tornillo																		

Anexo 3. Base de datos obtenida mediante filtro de Gabor en formato .csv.

Tabla 17.

Base de datos maderas.csv.

media	varianza	desviación	clase
27.4339855	135.141214	11.6250253	capinuri
46.1426372	454.609937	21.3215838	capinuri
56.6992316	637.06039	25.2400553	capinuri
53.0823519	641.39392	25.3257561	capinuri
58.0897699	863.063088	29.3779354	capinuri
54.1512679	820.076969	28.636986	capinuri
45.758698	643.250467	25.3623829	capinuri
54.8773649	996.399454	31.5657956	capinuri
51.8594503	939.602963	30.6529438	capinuri
58.53106	1196.88064	34.5959628	capinuri
49.4858732	856.361105	29.2636482	capinuri
66.480871	879.600052	29.6580521	capinuri
45.6825118	600.428162	24.5036357	capinuri
54.5159452	648.376743	25.463243	capinuri
63.7452539	954.562542	30.8959956	capinuri
40.1203851	393.881652	19.8464519	capinuri
35.4111213	321.503753	17.9305257	capinuri
53.1575026	735.553767	27.1210945	capinuri
63.8662758	1052.55527	32.4431082	capinuri
59.0210606	909.220582	30.1532848	capinuri
33.1436154	194.098291	13.9319163	capinuri
47.4920898	484.08269	22.0018792	capinuri
56.1188261	631.475372	25.1291737	capinuri
56.6548784	745.653261	27.3066523	capinuri
59.8056507	920.446762	30.3388655	capinuri
60.12682	1018.04241	31.9067769	capinuri
52.0611338	811.266854	28.4827466	capinuri
56.411351	1054.1848	32.4682122	capinuri
47.5531579	793.120335	28.1623922	capinuri
53.0400141	1004.8708	31.6996971	capinuri
53.1195055	991.266304	31.4843819	capinuri
71.8064089	1031.42352	32.115783	capinuri
48.8626718	694.194033	26.3475622	capinuri
64.6268496	923.183878	30.3839411	capinuri
60.7519808	871.721856	29.5249362	capinuri
60.5629308	889.462653	29.8238605	capinuri
36.5988207	349.891754	18.7053937	capinuri
58.2257113	898.046259	29.96742	capinuri
56.0992917	825.97695	28.7398147	capinuri
38.2256768	390.495739	19.760965	capinuri
48.4806281	430.446114	20.7471953	capinuri

56.4881002	703.812093	26.5294571	capinuri
53.7679359	599.892253	24.492698	capinuri
49.1464791	567.968656	23.832093	capinuri
53.1389923	739.70258	27.1974738	capinuri
62.3742352	1108.40967	33.2927871	capinuri
53.8354537	884.987179	29.7487341	capinuri
50.1189492	842.145362	29.0197409	capinuri
47.3278438	791.978985	28.1421212	capinuri
33.1877257	401.206698	20.0301447	capinuri
55.2736389	1105.07584	33.2426811	capinuri
50.5996675	526.496349	22.9455083	capinuri
46.6673297	650.478834	25.5044866	capinuri
67.0126384	1016.64572	31.8848823	capinuri
48.1607799	551.492033	23.4838675	capinuri
40.539351	405.632552	20.1403215	capinuri
43.4499636	496.129726	22.2739697	capinuri
53.7585027	774.476099	27.8294107	capinuri
55.782025	840.281803	28.9876146	capinuri
48.5536633	634.354877	25.1864026	capinuri
34.0012735	202.941715	14.2457613	capinuri
37.1860665	300.176394	17.3255994	capinuri
60.8706313	753.673942	27.4531226	capinuri
61.4942725	881.588569	29.6915572	capinuri
65.1203425	1093.93639	33.0747092	capinuri
62.7575	1112.66214	33.3565907	capinuri
39.4454767	480.010386	21.9091393	capinuri
50.3983057	845.219154	29.072653	capinuri
51.3424416	928.599422	30.4729293	capinuri
47.1975232	801.582153	28.3122262	capinuri
45.2122835	733.089502	27.0756256	capinuri
58.5597782	697.879069	26.4174009	capinuri
55.8644196	907.296768	30.1213673	capinuri
67.4017258	1013.6204	31.8374056	capinuri
49.671764	589.306102	24.2756277	capinuri
33.2685907	273.123656	16.5264532	capinuri
41.5317854	446.531422	21.1312901	capinuri
31.982522	272.259275	16.5002811	capinuri
58.4355298	899.380554	29.9896741	capinuri
45.2948849	557.526885	23.6120072	capinuri
28.7463354	147.31262	12.1372411	capinuri
36.1982289	285.044285	16.8832546	capinuri
49.9121158	513.108037	22.6518882	capinuri
48.9371037	559.449428	23.6526833	capinuri
64.6351028	1091.31112	33.0349984	capinuri
44.0453182	556.380218	23.5877133	capinuri
58.1525628	1037.8756	32.2160768	capinuri
53.965405	984.372154	31.3747056	capinuri

53.0175732	995.965454	31.5589204	capinuri
52.6814812	993.012642	31.5121031	capinuri
52.8543152	987.07007	31.4176713	capinuri
68.298694	942.573827	30.7013652	capinuri
29.0045328	249.439277	15.7936467	capinuri
66.9796795	1014.50024	31.8512204	capinuri
60.4340725	877.785031	29.6274371	capinuri
40.5892234	411.205537	20.2782035	capinuri
44.6955095	517.38035	22.7459963	capinuri
60.205509	960.82771	30.997221	capinuri
58.094621	886.261975	29.7701524	capinuri
63.393914	1054.30077	32.469998	capinuri
37.2625333	497.508388	22.3048961	capirona
35.3766937	476.495663	21.8287806	capirona
32.4637917	425.383121	20.6248181	capirona
30.242697	411.075825	20.2750049	capirona
38.4647058	700.797325	26.4725769	capirona
27.1436498	366.970097	19.1564636	capirona
32.8152546	576.161948	24.0033737	capirona
26.2866645	398.649543	19.96621	capirona
38.6106713	874.836	29.5776267	capirona
28.0116192	473.501716	21.7600946	capirona
34.5585096	732.199178	27.0591792	capirona
32.5777289	405.415181	20.1349244	capirona
31.6099804	575.373275	23.9869397	capirona
42.1163562	732.78177	27.0699422	capirona
39.6125274	671.073415	25.9050847	capirona
36.3971979	584.315498	24.1726188	capirona
39.312084	699.814035	26.4539985	capirona
42.097452	834.875071	28.8942048	capirona
44.3204094	894.308978	29.9049992	capirona
38.3827757	665.665299	25.8004903	capirona
31.0287034	447.419404	21.1522908	capirona
25.5695019	346.343237	18.6102992	capirona
29.3786646	460.65183	21.4628011	capirona
29.0664248	471.754222	21.7199038	capirona
36.6165876	763.849596	27.6378291	capirona
29.3457698	514.638701	22.6856497	capirona
29.1162741	550.605758	23.46499	capirona
19.7884648	274.89182	16.5798619	capirona
28.4838004	582.071488	24.1261578	capirona
28.7629666	622.844328	24.9568493	capirona
24.3223739	443.458287	21.0584493	capirona
34.0113862	551.924726	23.4930783	capirona
23.1236855	422.615405	20.5576119	capirona
34.2508206	588.813686	24.2654834	capirona
37.3991952	724.837827	26.9228124	capirona

27.7090686	410.609608	20.2635043	capirona
27.7638364	425.469193	20.6269046	capirona
33.6921026	639.113112	25.2806866	capirona
33.1192954	618.430195	24.8682568	capirona
29.9010004	498.148086	22.3192313	capirona
25.0867551	315.678625	17.7673472	capirona
33.9563246	607.688813	24.6513451	capirona
29.4054856	458.719837	21.4177459	capirona
34.3374165	676.28912	26.0055594	capirona
31.2324498	561.255278	23.6908269	capirona
28.6431837	498.753016	22.332779	capirona
27.0309353	476.861887	21.8371676	capirona
30.2325318	623.671745	24.9734208	capirona
29.376948	623.623576	24.9724564	capirona
16.8403377	212.549701	14.5790844	capirona
28.8011655	632.415559	25.1478738	capirona
24.5434194	308.011739	17.5502632	capirona
23.4012023	438.224996	20.9338242	capirona
22.7137868	270.658196	16.4516928	capirona
28.3702112	432.880752	20.8057865	capirona
28.1016746	428.096684	20.6904974	capirona
30.756215	530.367537	23.0297099	capirona
31.7204892	577.197889	24.0249431	capirona
25.8071261	385.265193	19.6281734	capirona
33.2718697	632.004595	25.1397016	capirona
15.029862	119.208133	10.9182477	capirona
27.6669818	395.863494	19.8963186	capirona
31.2115056	524.927104	22.9112877	capirona
23.5274315	320.729597	17.9089251	capirona
30.6867902	541.41535	23.2683336	capirona
31.1317319	572.161256	23.9198925	capirona
27.2612089	484.355323	22.008074	capirona
23.0768935	371.035093	19.2622712	capirona
28.711742	590.219407	24.2944316	capirona
31.2201249	704.618399	26.5446492	capirona
23.4523432	423.170862	20.5711172	capirona
30.402914	463.273321	21.523785	capirona
21.3505833	351.703819	18.7537681	capirona
31.7345356	522.631389	22.8611327	capirona
27.7577051	413.475854	20.3341057	capirona
33.4369297	612.876341	24.7563394	capirona
28.5894876	467.914409	21.6313293	capirona
31.1423861	552.306992	23.5012126	capirona
33.7970266	649.910603	25.4933443	capirona
25.6046286	362.916932	19.0503788	capirona
39.5488778	607.277306	24.6429971	capirona
38.6944822	660.630649	25.7027362	capirona

43.6806196	896.710799	29.9451298	capirona
43.6806196	896.710799	29.9451298	capirona
43.2634868	894.838547	29.9138521	capirona
35.2887766	632.063141	25.140866	capirona
28.0472662	451.626488	21.2515056	capirona
24.9604898	392.082962	19.8010849	capirona
32.7636312	695.077157	26.364316	capirona
34.1285252	780.858498	27.9438454	capirona
30.4466304	648.005909	25.4559602	capirona
35.0436853	473.389208	21.7575092	capirona
31.4550527	738.392383	27.1733764	capirona
32.4119665	429.549069	20.7255656	capirona
26.518855	300.435847	17.3330853	capirona
37.2208486	598.915221	24.4727445	capirona
29.9383033	403.067038	20.0765295	capirona
37.8306288	673.001996	25.942282	capirona
40.3815301	769.531795	27.7404361	capirona
46.7176435	1011.54207	31.8047492	capirona
38.4510287	404.554651	20.113544	catahua
31.0445665	289.465135	17.013675	catahua
39.0990422	773.168486	27.8059074	catahua
35.7408392	705.171574	26.5550668	catahua
34.9866658	700.870844	26.4739654	catahua
27.8898473	462.613872	21.5084605	catahua
23.3765066	376.119522	19.3938011	catahua
23.1140931	382.292195	19.5522939	catahua
22.5322942	365.408222	19.1156539	catahua
28.6032796	546.555212	23.3785203	catahua
32.8348415	716.313938	26.7640419	catahua
33.6522329	302.357512	17.3884304	catahua
30.9993353	644.431202	25.3856495	catahua
42.9374631	508.152239	22.5422324	catahua
43.8412502	569.070805	23.855205	catahua
43.0890084	578.320977	24.0483051	catahua
38.5183561	530.742094	23.0378405	catahua
37.2429776	476.457693	21.8279109	catahua
39.6307489	500.849225	22.379661	catahua
40.2120423	480.331902	21.9164756	catahua
43.1338211	495.342097	22.2562822	catahua
28.1588811	225.421656	15.0140486	catahua
38.8748572	758.719996	27.5448724	catahua
34.0864334	634.99989	25.1992042	catahua
32.3902395	588.233384	24.2535231	catahua
27.1123042	425.741502	20.6335044	catahua
21.3962051	320.001059	17.8885734	catahua
21.4052969	322.292415	17.9525044	catahua
19.8965448	268.913088	16.3985697	catahua

28.3736083	531.260185	23.0490821	catahua
29.9387317	593.66556	24.3652531	catahua
42.7600158	484.740117	22.0168144	catahua
33.7040943	747.77553	27.3454846	catahua
46.5040963	593.656527	24.3650678	catahua
41.0098566	493.62817	22.2177445	catahua
32.2817411	328.394973	18.1216714	catahua
28.7792466	291.881344	17.0845352	catahua
35.8563978	429.864204	20.7331668	catahua
44.4977828	612.21241	24.7429265	catahua
42.4384806	516.514655	22.7269588	catahua
38.2009187	381.580795	19.5340931	catahua
37.7927154	405.720267	20.142499	catahua
39.0183577	759.608224	27.560991	catahua
35.5377181	681.860913	26.1124666	catahua
31.5003742	548.590981	23.4220192	catahua
34.6521722	685.737783	26.1865955	catahua
24.827445	415.71669	20.3891317	catahua
26.4346846	485.88458	22.0427898	catahua
29.8971947	601.091971	24.5171771	catahua
22.6506558	344.273031	18.554596	catahua
26.9754881	477.182917	21.8445169	catahua
40.6813876	422.975419	20.5663662	catahua
33.9449122	757.752928	27.5273124	catahua
44.6798187	548.697735	23.424298	catahua
40.896315	476.051378	21.8186016	catahua
41.3428043	516.802987	22.7333013	catahua
41.9171163	590.758813	24.3055305	catahua
36.970824	459.068772	21.4258902	catahua
26.4963747	215.629945	14.6843435	catahua
37.7665131	407.660246	20.190598	catahua
45.9622885	535.51934	23.1412908	catahua
31.3036896	288.091958	16.9732719	catahua
29.7066331	459.200417	21.4289621	catahua
36.8379285	743.782206	27.2723708	catahua
29.5475698	502.095887	22.4074962	catahua
27.0579516	443.66664	21.0633957	catahua
24.5591152	402.731643	20.0681749	catahua
27.866279	550.751259	23.4680902	catahua
25.7363507	460.188706	21.4520094	catahua
24.6889614	417.652889	20.4365577	catahua
26.9297251	491.65829	22.1733689	catahua
41.6747082	437.336637	20.9125952	catahua
26.8243071	479.275685	21.8923659	catahua
42.0725511	474.677755	21.7871007	catahua
34.8435083	356.371722	18.8778103	catahua
45.8952647	640.029335	25.2988011	catahua

34.4085841	395.488282	19.8868872	catahua
38.9489773	498.66983	22.3309165	catahua
32.5740659	329.52417	18.1528006	catahua
38.7455739	431.99522	20.7844947	catahua
34.4950733	321.219674	17.9226023	catahua
38.7864249	449.795552	21.208384	catahua
22.4371989	265.41823	16.2916614	catahua
32.1422483	560.46051	23.6740472	catahua
29.5826473	513.907053	22.6695181	catahua
21.5279468	278.297721	16.6822577	catahua
24.8372261	427.240779	20.6698036	catahua
25.5471628	468.972486	21.6557726	catahua
25.3429635	448.391671	21.1752608	catahua
27.9369904	538.833988	23.2127979	catahua
29.3900771	586.514841	24.2180685	catahua
38.5472301	389.716286	19.7412331	catahua
24.2973762	396.019229	19.9002319	catahua
34.5845822	326.941899	18.0815347	catahua
43.9724339	575.650177	23.9927109	catahua
31.6052703	312.881601	17.6884596	catahua
37.497064	502.55024	22.4176323	catahua
41.7932635	595.383258	24.4004766	catahua
37.2491975	441.067043	21.0015962	catahua
34.0285278	341.519279	18.4802402	catahua
30.4821166	582.284912	24.1305804	cedro
22.1265443	305.935754	17.4910192	cedro
32.0772776	650.534154	25.505571	cedro
29.0282389	537.483909	23.1836992	cedro
22.4697835	318.27119	17.8401567	cedro
30.1372724	566.87904	23.8092218	cedro
28.6375727	570.458538	23.8842739	cedro
20.9603404	327.507135	18.0971582	cedro
30.9570434	729.257068	27.0047601	cedro
29.8930869	707.15906	26.5924625	cedro
28.7724326	688.898173	26.2468698	cedro
32.5663099	653.039595	25.5546394	cedro
28.4420484	715.175449	26.7427644	cedro
31.0233943	579.919827	24.0815246	cedro
32.4083412	613.265566	24.7641993	cedro
32.5960652	641.701749	25.3318327	cedro
32.3360117	644.29498	25.3829663	cedro
27.9127591	493.24051	22.2090187	cedro
35.9327727	827.1596	28.7603825	cedro
29.6372412	546.824381	23.3842764	cedro
30.1148545	570.294381	23.8808371	cedro
20.1866573	258.877275	16.0896636	cedro
32.8134576	691.221899	26.2910992	cedro

33.8457222	714.64971	26.7329331	cedro
35.2300652	772.059584	27.7859602	cedro
40.3279127	1011.72975	31.8076995	cedro
28.1017862	543.878276	23.321198	cedro
28.6677465	603.430449	24.5648214	cedro
25.3114177	486.849003	22.0646551	cedro
35.2765125	990.270041	31.4685564	cedro
32.4765811	897.608636	29.9601174	cedro
38.6076615	915.762901	30.2615747	cedro
29.1887021	766.322455	27.6825298	cedro
33.0332608	658.79168	25.6669375	cedro
36.7181982	796.814899	28.2279099	cedro
21.8955552	302.110604	17.3813292	cedro
37.9677616	893.097826	29.8847424	cedro
27.3470581	472.757282	21.7429824	cedro
35.0150886	791.178931	28.1279031	cedro
36.9433629	870.017613	29.496061	cedro
26.5263484	453.090707	21.2859274	cedro
29.7610661	556.818741	23.597007	cedro
30.6606372	611.3889	24.7262795	cedro
26.5094365	447.095103	21.1446235	cedro
29.0051696	527.564463	22.9687715	cedro
31.2770458	621.594317	24.9317933	cedro
22.0895975	343.187767	18.5253277	cedro
23.4727014	419.697461	20.486519	cedro
24.8334088	476.365754	21.8258048	cedro
23.1624998	430.638342	20.7518274	cedro
26.3281999	586.308354	24.213805	cedro
33.0868289	685.123292	26.1748599	cedro
29.6483287	787.488541	28.0622262	cedro
29.6359939	540.060566	23.2392032	cedro
21.9848392	294.877627	17.1720013	cedro
22.134402	309.537406	17.5936752	cedro
30.3200139	598.769353	24.4697641	cedro
27.5589232	479.513327	21.8977927	cedro
33.2368168	720.560266	26.8432536	cedro
33.1417395	701.417063	26.4842795	cedro
31.8571888	648.304723	25.4618287	cedro
26.2840551	426.721115	20.6572291	cedro
32.0641699	652.184407	25.5379014	cedro
34.4666457	743.557085	27.2682432	cedro
25.4097871	403.474672	20.086679	cedro
27.9268679	504.611169	22.463552	cedro
19.3253131	269.027709	16.4020642	cedro
24.2977159	446.222699	21.123984	cedro
25.802805	509.42644	22.5704772	cedro
28.7020609	651.790004	25.5301783	cedro

34.4382278	1000.58901	31.6320884	cedro
20.9027368	273.836165	16.5479958	cedro
26.8357721	634.33826	25.1860727	cedro
32.5252422	638.389555	25.266372	cedro
25.0371897	376.80082	19.411358	cedro
23.3803895	336.89617	18.3547315	cedro
27.7301522	489.717895	22.1295706	cedro
28.5399435	529.0111	23.0002413	cedro
33.0158779	708.354352	26.6149272	cedro
34.1591995	743.875856	27.2740876	cedro
30.7831279	588.223057	24.2533102	cedro
25.4233773	412.46689	20.3092809	cedro
29.5682858	555.019691	23.5588559	cedro
31.8066124	635.368272	25.2065125	cedro
37.9413165	875.165022	29.5831882	cedro
29.2883877	531.111279	23.0458517	cedro
25.8685947	459.275281	21.4307088	cedro
20.8683239	316.362749	17.786589	cedro
31.1557105	731.522185	27.0466668	cedro
24.459085	473.260605	21.7545537	cedro
30.444451	782.393589	27.9712994	cedro
25.2446548	402.989931	20.0746091	cedro
20.849441	383.675106	19.5876264	cedro
28.0455693	479.40977	21.8954281	cedro
31.4733448	580.781374	24.0994061	cedro
29.9733398	547.812005	23.4053841	cedro
25.9440489	420.651965	20.5098017	cedro
25.7745694	406.064246	20.1510358	cedro
27.9597907	504.524636	22.4616259	cedro
23.8488929	366.175926	19.1357238	cedro
29.9292312	412.629949	20.3132949	copaiba
17.5493225	188.524584	13.7304255	copaiba
26.7822417	438.70137	20.9451992	copaiba
26.1991299	440.947991	20.9987617	copaiba
23.5330097	368.463492	19.1954029	copaiba
20.5376656	299.721063	17.312454	copaiba
19.2769047	272.645628	16.5119844	copaiba
22.207434	374.540739	19.353055	copaiba
25.7052135	526.622506	22.9482571	copaiba
25.457905	529.918814	23.0199656	copaiba
25.4452027	566.994982	23.8116564	copaiba
27.1770288	351.480454	18.747812	copaiba
23.7711968	507.295638	22.5232244	copaiba
27.6736316	367.209743	19.1627175	copaiba
28.6831141	427.257468	20.6702073	copaiba
27.5720621	426.474244	20.6512528	copaiba
28.9751106	476.272612	21.8236709	copaiba

24.4383493	335.931778	18.3284418	copaiba
32.9468224	574.186626	23.9621916	copaiba
28.2294251	438.270934	20.9349214	copaiba
25.7680803	312.928894	17.6897963	copaiba
18.5247039	204.456921	14.2988433	copaiba
26.0742645	417.206329	20.4256292	copaiba
22.2085368	321.622534	17.9338377	copaiba
23.5448636	378.325376	19.4505881	copaiba
24.1797416	413.685417	20.339258	copaiba
19.5549155	286.531819	16.9272508	copaiba
19.0421639	283.437428	16.8356	copaiba
19.4317848	314.280283	17.727952	copaiba
27.0972501	601.517861	24.5258611	copaiba
27.5048676	667.860077	25.8429889	copaiba
31.6218408	473.325923	21.7560549	copaiba
15.3857609	220.567007	14.8514985	copaiba
25.6038508	324.55999	18.0155486	copaiba
29.6526219	444.892917	21.0924849	copaiba
26.9287717	406.216138	20.1548043	copaiba
28.3686521	457.965024	21.4001174	copaiba
27.0211378	420.567184	20.5077347	copaiba
27.1044924	392.185615	19.8036768	copaiba
22.0530676	279.677547	16.7235626	copaiba
24.0419637	252.978953	15.9053121	copaiba
24.7172201	350.790058	18.7293902	copaiba
29.1612181	511.569523	22.6179027	copaiba
22.4744394	319.407542	17.8719764	copaiba
24.4203888	390.200609	19.7534961	copaiba
25.5687798	449.385012	21.1987031	copaiba
21.4363225	340.692687	18.4578625	copaiba
24.0165984	445.209599	21.0999905	copaiba
20.2939577	325.530997	18.0424776	copaiba
25.1290372	514.945395	22.6924083	copaiba
28.4783765	706.190012	26.5742359	copaiba
30.6049109	410.057777	20.2498834	copaiba
19.6328003	349.209063	18.6871363	copaiba
27.300217	347.374955	18.6379976	copaiba
32.2231281	503.893411	22.4475703	copaiba
28.8928883	446.619947	21.1333847	copaiba
31.4643514	552.662813	23.5087816	copaiba
27.7575886	422.820732	20.5626052	copaiba
29.187555	443.820138	21.0670391	copaiba
27.1687903	398.686462	19.9671345	copaiba
28.6742979	353.956944	18.8137435	copaiba
21.430956	251.669239	15.8640864	copaiba
32.6419119	626.651369	25.0330056	copaiba
23.4228406	334.797344	18.2974682	copaiba

28.3608206	511.993666	22.627277	copaiba
27.5090722	500.878185	22.380308	copaiba
21.1478872	322.564208	17.9600726	copaiba
24.6046812	463.03994	21.5183629	copaiba
26.9394193	556.556976	23.5914598	copaiba
25.8975721	537.249245	23.1786377	copaiba
31.2011944	819.316199	28.6237	copaiba
31.113709	429.355903	20.720905	copaiba
26.7256834	636.773171	25.2343649	copaiba
31.4324445	444.91446	21.0929955	copaiba
31.1821442	468.213848	21.6382497	copaiba
29.8586576	470.914031	21.7005537	copaiba
28.3660247	434.737539	20.8503606	copaiba
28.5093331	427.692244	20.6807215	copaiba
29.294235	437.528321	20.9171777	copaiba
27.8749754	415.244518	20.3775494	copaiba
25.9048834	295.086083	17.1780698	copaiba
24.9199752	339.878687	18.4357991	copaiba
26.1816354	397.609231	19.9401412	copaiba
25.5618559	394.260015	19.8559819	copaiba
32.3681581	662.841028	25.7456992	copaiba
27.6075499	505.301804	22.4789191	copaiba
27.3549011	537.765527	23.189772	copaiba
20.8412419	329.168983	18.1430147	copaiba
25.9560833	527.120552	22.9591061	copaiba
24.4848474	475.874444	21.8145466	copaiba
28.8588595	701.795413	26.4914215	copaiba
29.5582043	386.181223	19.6514942	copaiba
24.8292551	546.613203	23.3797605	copaiba
28.7325876	361.72695	19.0191206	copaiba
30.5958404	450.568516	21.2265993	copaiba
26.5327701	371.656025	19.2783823	copaiba
24.8912915	336.359515	18.3401067	copaiba
30.634878	486.716549	22.0616534	copaiba
29.4521528	438.157071	20.9322018	copaiba
28.0479719	416.924437	20.4187276	copaiba
32.6593538	316.297748	17.7847617	cumala
27.508977	235.316149	15.3400179	cumala
33.622668	314.161673	17.7246064	cumala
37.8912981	428.532127	20.7010175	cumala
34.7813948	425.866929	20.6365435	cumala
31.7230001	388.959701	19.7220613	cumala
27.9699279	340.062985	18.4407968	cumala
28.4448711	375.024374	19.3655461	cumala
26.3268049	340.3642	18.448962	cumala
20.5404096	209.336141	14.4684533	cumala
29.0031526	423.520952	20.5796247	cumala

35.8493606	393.810362	19.8446558	cumala
30.3936859	397.049147	19.9260921	cumala
25.8270708	224.344933	14.9781485	cumala
32.0707541	372.144233	19.2910402	cumala
26.9548788	299.078114	17.293875	cumala
32.1554495	415.24469	20.3775536	cumala
30.0030279	331.953913	18.2196024	cumala
32.9813847	399.977687	19.9994422	cumala
32.9970854	369.424568	19.2204206	cumala
27.9648174	236.448842	15.3768931	cumala
24.9102613	216.086967	14.6998968	cumala
28.0027046	227.850647	15.0947225	cumala
39.4152947	485.992533	22.0452383	cumala
31.2739786	351.692973	18.753479	cumala
29.8036059	359.161493	18.9515565	cumala
28.1313249	346.184841	18.6060431	cumala
29.2024811	392.340825	19.8075951	cumala
27.9644334	389.709968	19.7410731	cumala
25.6053606	332.496668	18.2344912	cumala
24.1237019	301.75275	17.3710319	cumala
29.7642302	273.494357	16.5376648	cumala
24.7773938	277.017894	16.6438545	cumala
31.6251953	328.355268	18.1205758	cumala
29.9373843	334.179893	18.2805879	cumala
24.630936	257.31165	16.0409367	cumala
25.8377956	274.650671	16.5725879	cumala
32.3633808	407.437415	20.185079	cumala
29.3047941	323.115445	17.9754123	cumala
26.717353	264.317857	16.2578552	cumala
32.5304643	299.784281	17.3142797	cumala
26.675683	238.780169	15.4525134	cumala
35.4352738	344.95089	18.5728536	cumala
33.4736713	346.574572	18.6165134	cumala
36.7658369	477.332031	21.8479297	cumala
30.9865247	375.783732	19.3851421	cumala
31.133227	428.680342	20.7045971	cumala
27.8528234	366.704057	19.1495185	cumala
24.8129029	295.551379	17.1916078	cumala
25.4715182	323.041231	17.9733478	cumala
23.9180108	289.912926	17.0268296	cumala
34.3630591	341.589696	18.4821453	cumala
27.0065596	325.373992	18.0381261	cumala
36.6364108	441.672666	21.0160098	cumala
29.1509955	309.682127	17.5977876	cumala
31.6754057	387.606802	19.6877323	cumala
31.2393737	379.577862	19.4827581	cumala
28.5052959	310.282174	17.6148283	cumala

32.414095	384.374587	19.6054734	cumala
37.5725298	494.949408	22.2474585	cumala
31.7330685	284.752567	16.8746131	cumala
31.102917	297.480361	17.247619	cumala
32.2205614	277.903524	16.6704386	cumala
30.4572174	278.204906	16.6794756	cumala
34.6659774	421.280255	20.5251128	cumala
31.6804407	389.791007	19.7431256	cumala
30.4703628	396.217655	19.9052168	cumala
28.9580675	380.849173	19.5153574	cumala
30.9004195	454.533157	21.3197832	cumala
29.6156259	420.92883	20.5165502	cumala
31.4795072	478.739559	21.8801179	cumala
41.6287533	501.307241	22.3898915	cumala
27.2341861	326.513884	18.0696952	cumala
26.4852774	230.040244	15.1670776	cumala
28.530827	292.901731	17.1143721	cumala
32.4975662	419.482595	20.4812743	cumala
31.9017078	395.845724	19.895872	cumala
31.8538475	374.755403	19.3586002	cumala
30.0568586	319.988712	17.8882283	cumala
31.842243	349.488391	18.6946086	cumala
31.9175644	300.898001	17.3464118	cumala
32.4921423	353.864401	18.8112839	cumala
35.9028155	349.012888	18.6818866	cumala
33.1451071	326.619915	18.0726289	cumala
34.5605265	414.255264	20.3532618	cumala
29.940486	350.968126	18.7341433	cumala
25.139931	276.464282	16.6272151	cumala
27.0924088	336.432419	18.3420942	cumala
28.0811933	383.839493	19.5918221	cumala
28.8769201	413.916608	20.3449406	cumala
32.1526219	510.206767	22.587757	cumala
33.2832838	332.301847	18.2291483	cumala
23.4265627	244.174025	15.6260688	cumala
31.9175217	328.549706	18.1259402	cumala
34.7308234	446.540511	21.1315052	cumala
24.8125911	257.364272	16.0425768	cumala
32.5838569	429.928091	20.7347074	cumala
30.4159938	346.680491	18.619358	cumala
32.1325525	369.760682	19.2291623	cumala
32.4733825	377.057785	19.4179758	cumala
24.0493094	449.537849	21.2023076	huayruro
17.8883227	254.566406	15.9551373	huayruro
25.8880749	496.183877	22.2751852	huayruro
26.4736533	494.210775	22.2308519	huayruro
23.6824757	399.070304	19.9767441	huayruro

26.5628044	559.981407	23.6639263	huayruro
19.0295108	303.649086	17.4255297	huayruro
24.5796884	496.11659	22.2736748	huayruro
24.9283351	507.47822	22.5272772	huayruro
23.1188687	476.498906	21.8288549	huayruro
23.5730418	485.058958	22.0240541	huayruro
28.45191	593.361497	24.3590126	huayruro
24.0757776	534.796705	23.125672	huayruro
26.9198915	525.31307	22.9197092	huayruro
25.5410447	434.412947	20.8425753	huayruro
27.4727966	510.734552	22.599437	huayruro
25.3034992	438.040684	20.9294215	huayruro
20.4770537	305.806296	17.4873182	huayruro
25.1039377	487.534684	22.0801876	huayruro
22.310003	347.665652	18.6457945	huayruro
21.8795968	345.341632	18.5833698	huayruro
18.5986254	237.237477	15.4025153	huayruro
28.564759	559.869229	23.6615559	huayruro
27.915961	515.291748	22.7000385	huayruro
27.3781756	517.555206	22.7498397	huayruro
25.4603946	483.543691	21.9896269	huayruro
22.734232	407.669885	20.1908367	huayruro
18.6245618	276.667937	16.6333381	huayruro
21.7883582	351.633929	18.7519047	huayruro
27.1980434	595.978344	24.4126677	huayruro
30.9907211	738.616132	27.1774931	huayruro
29.0269966	565.099662	23.771825	huayruro
29.6028923	695.343267	26.3693623	huayruro
29.8661543	598.776854	24.4699173	huayruro
19.2683889	239.13355	15.4639436	huayruro
26.3433016	442.748605	21.0415923	huayruro
24.9936915	404.120134	20.1027395	huayruro
26.8250112	470.575772	21.6927585	huayruro
28.1648844	535.233251	23.1351086	huayruro
30.6250279	592.986236	24.3513087	huayruro
25.0206979	473.459915	21.7591341	huayruro
23.7292774	391.882883	19.796032	huayruro
25.8222672	476.685774	21.8331348	huayruro
25.3241332	437.974516	20.9278407	huayruro
23.7645339	397.443069	19.9359742	huayruro
23.5949214	436.991587	20.9043437	huayruro
19.7386664	318.418556	17.8442864	huayruro
21.98436	388.248574	19.7040243	huayruro
25.0411858	485.084972	22.0246447	huayruro
21.6215044	384.902724	19.6189379	huayruro
27.9613514	624.936974	24.9987394	huayruro
20.9610921	300.699321	17.340684	huayruro

25.794235	540.649306	23.2518667	huayruro
27.0015788	505.702375	22.4878273	huayruro
25.8683879	432.338892	20.7927606	huayruro
22.4406534	340.409639	18.4501935	huayruro
21.992628	330.18842	18.1710875	huayruro
23.2282764	365.552031	19.119415	huayruro
26.1472898	492.723824	22.1973833	huayruro
30.3112733	587.380891	24.2359421	huayruro
19.9238105	309.138154	17.582325	huayruro
21.2054449	313.406056	17.7032781	huayruro
23.1236756	380.84319	19.5152041	huayruro
30.3596046	611.391327	24.7263286	huayruro
26.6541022	486.180485	22.0495008	huayruro
21.7234449	356.920864	18.8923493	huayruro
21.8630608	382.971496	19.5696575	huayruro
18.2425378	263.959824	16.2468404	huayruro
17.9363323	248.51124	15.7642393	huayruro
22.8629705	435.035704	20.8575095	huayruro
22.3062908	406.223105	20.1549772	huayruro
28.2579037	555.446595	23.5679145	huayruro
22.9970443	430.722251	20.7538491	huayruro
21.6865884	337.279918	18.3651822	huayruro
23.2870928	349.375614	18.6915921	huayruro
25.6106729	437.062932	20.9060501	huayruro
27.7675289	518.567192	22.7720705	huayruro
25.9731232	470.024234	21.6800423	huayruro
22.0563269	345.196688	18.5794695	huayruro
23.4949371	354.171108	18.8194343	huayruro
21.3179937	343.2126	18.525998	huayruro
23.792077	395.040825	19.875634	huayruro
22.4697556	359.9405	18.9720979	huayruro
20.4329418	294.328151	17.1559946	huayruro
26.9235529	500.779115	22.3780945	huayruro
25.258465	491.265275	22.1645049	huayruro
20.8992132	366.328439	19.1397084	huayruro
19.0724412	298.201221	17.2685037	huayruro
22.5802644	391.80235	19.7939978	huayruro
22.8762309	435.509231	20.8688579	huayruro
25.0790894	503.866127	22.4469625	huayruro
23.7012912	399.122512	19.9780508	huayruro
27.0169824	582.550705	24.1360872	huayruro
23.7814539	400.457617	20.0114372	huayruro
25.9367213	433.428106	20.8189362	huayruro
25.8585362	447.5078	21.1543802	huayruro
24.0704816	393.587554	19.8390412	huayruro
23.7220335	388.752528	19.7168083	huayruro
17.4973922	218.531464	14.7828097	huayruro

28.4346272	524.093902	22.8930973	huayruro
32.8873101	326.744903	18.0760865	lupuna
33.2654627	360.726998	18.9928144	lupuna
37.5754789	364.343339	19.0877798	lupuna
37.5068025	379.986539	19.4932434	lupuna
35.7550055	367.691156	19.1752746	lupuna
32.0997611	324.022198	18.0006166	lupuna
30.5763323	367.463261	19.1693313	lupuna
27.3823999	307.239446	17.5282471	lupuna
33.6610392	467.518085	21.6221665	lupuna
27.7854041	325.407966	18.0390678	lupuna
34.3902363	520.158639	22.8069866	lupuna
32.1071544	321.55836	17.9320484	lupuna
32.0817398	426.297622	20.6469761	lupuna
31.484214	335.882663	18.3271019	lupuna
36.8404969	469.656352	21.6715563	lupuna
29.7917701	314.927848	17.7462066	lupuna
23.5217745	223.199215	14.9398533	lupuna
29.7361554	321.891012	17.9413214	lupuna
34.7374667	418.339598	20.4533518	lupuna
40.899356	586.076788	24.2090229	lupuna
26.735819	243.702185	15.6109636	lupuna
25.2426756	206.686748	14.3766042	lupuna
41.9609854	457.10859	21.380098	lupuna
34.478493	327.263861	18.0904356	lupuna
30.3335516	276.754862	16.6359509	lupuna
33.2916716	371.986993	19.2869643	lupuna
27.5984859	311.767543	17.6569404	lupuna
27.6505163	322.300533	17.9527305	lupuna
31.5878481	421.92217	20.5407442	lupuna
29.6665157	382.854626	19.5666713	lupuna
35.2621772	553.788416	23.5327095	lupuna
25.8155369	221.556501	14.8847742	lupuna
39.3950169	644.01631	25.3774764	lupuna
29.412723	312.384163	17.6743929	lupuna
29.6560436	316.719201	17.7966064	lupuna
32.7620328	408.683922	20.2159324	lupuna
25.5369008	269.117653	16.4048058	lupuna
31.9335457	388.757645	19.716938	lupuna
35.6959001	442.24822	21.0296985	lupuna
34.6061812	417.938957	20.4435554	lupuna
26.816666	245.317952	15.6626291	lupuna
26.9826139	241.388682	15.5366883	lupuna
30.1724467	244.351661	15.6317517	lupuna
37.8304515	399.172256	19.9792957	lupuna
36.1146445	388.275947	19.7047189	lupuna
30.8009752	323.951161	17.9986433	lupuna

25.7280942	269.364906	16.41234	lupuna
29.9518919	388.797356	19.717945	lupuna
28.3787845	340.86429	18.4625104	lupuna
28.0224917	335.559095	18.3182722	lupuna
31.5135754	447.805404	21.1614131	lupuna
27.4160627	249.524336	15.7963393	lupuna
31.5135754	447.805404	21.1614131	lupuna
31.9839973	364.842447	19.1008494	lupuna
29.2945944	314.464442	17.7331453	lupuna
28.7390914	325.388813	18.0385369	lupuna
24.742441	271.788942	16.4860226	lupuna
30.8715963	364.580351	19.0939873	lupuna
31.9832785	358.290967	18.9285754	lupuna
24.0377247	203.816508	14.2764319	lupuna
36.6230536	435.756291	20.8747764	lupuna
27.0156465	244.768467	15.6450781	lupuna
33.241343	299.797319	17.3146562	lupuna
35.5034923	364.74233	19.0982285	lupuna
36.3046677	404.368196	20.1089084	lupuna
35.1440273	422.878663	20.5640138	lupuna
30.1123452	374.963001	19.3639614	lupuna
27.5692295	328.252185	18.1177312	lupuna
31.6582001	434.272585	20.8392079	lupuna
28.687235	358.335688	18.9297567	lupuna
35.4459412	568.974069	23.8531773	lupuna
38.1453582	475.191696	21.7988921	lupuna
30.576178	397.427265	19.9355779	lupuna
32.4052772	371.198339	19.2665082	lupuna
29.5323828	313.234162	17.6984226	lupuna
29.0080038	321.791727	17.9385542	lupuna
28.0382121	338.320789	18.3934986	lupuna
25.9038773	262.823531	16.2118331	lupuna
34.2407818	407.926734	20.1971962	lupuna
34.7815031	426.582348	20.6538701	lupuna
26.1217621	230.858013	15.1940124	lupuna
28.444282	278.39312	16.6851167	lupuna
34.4528224	327.652933	18.101186	lupuna
45.3497758	571.006569	23.8957438	lupuna
38.9488509	457.97804	21.4004215	lupuna
30.3202814	313.922852	17.7178682	lupuna
26.8611866	303.704335	17.4271149	lupuna
25.9527863	296.786398	17.2274896	lupuna
29.9418794	398.746476	19.9686373	lupuna
38.0527115	634.509348	25.189469	lupuna
30.4629285	427.210407	20.6690689	lupuna
31.3545138	325.170416	18.0324823	lupuna
22.967522	225.369043	15.0122964	lupuna

34.6255449	429.118174	20.7151677	lupuna
30.0987353	332.161	18.2252846	lupuna
28.4045043	321.419885	17.9281869	lupuna
23.4595954	237.895216	15.4238522	lupuna
26.0435277	272.362904	16.503421	lupuna
38.4716167	540.143952	23.2409972	lupuna
39.6583543	565.870715	23.7880372	lupuna
37.2315865	336.678578	18.3488032	moena
27.6192167	344.07496	18.5492577	moena
51.9046487	605.885027	24.6147319	moena
46.144664	566.971784	23.8111693	moena
29.5166919	276.714437	16.6347359	moena
37.3026081	517.340853	22.7451281	moena
32.3657867	425.753929	20.6338055	moena
30.705806	448.256386	21.1720662	moena
26.9446611	379.417369	19.4786388	moena
33.9264117	652.989369	25.5536567	moena
31.5314408	593.044435	24.3525037	moena
39.6015614	410.712137	20.2660341	moena
26.638111	420.774402	20.5127863	moena
45.6508938	605.985965	24.6167822	moena
44.5493767	633.027841	25.1600445	moena
42.0000295	582.829759	24.1418674	moena
33.4752714	424.429535	20.6016877	moena
27.0225229	291.346103	17.0688636	moena
30.5325535	416.952848	20.4194233	moena
32.6620699	505.887371	22.4919401	moena
37.2315865	336.678578	18.3488032	moena
27.6192167	344.07496	18.5492577	moena
51.9046487	605.885027	24.6147319	moena
46.144664	566.971784	23.8111693	moena
29.5166919	276.714437	16.6347359	moena
37.3026081	517.340853	22.7451281	moena
32.3657867	425.753929	20.6338055	moena
30.705806	448.256386	21.1720662	moena
26.9446611	379.417369	19.4786388	moena
33.9264117	652.989369	25.5536567	moena
31.5314408	593.044435	24.3525037	moena
39.6015614	410.712137	20.2660341	moena
26.638111	420.774402	20.5127863	moena
45.6508938	605.985965	24.6167822	moena
44.5493767	633.027841	25.1600445	moena
42.0000295	582.829759	24.1418674	moena
33.4752714	424.429535	20.6016877	moena
27.0225229	291.346103	17.0688636	moena
30.5325535	416.952848	20.4194233	moena
32.6620699	505.887371	22.4919401	moena

37.2315865	336.678578	18.3488032	moena
27.6192167	344.07496	18.5492577	moena
51.9046487	605.885027	24.6147319	moena
46.144664	566.971784	23.8111693	moena
29.5166919	276.714437	16.6347359	moena
37.3026081	517.340853	22.7451281	moena
32.3657867	425.753929	20.6338055	moena
30.705806	448.256386	21.1720662	moena
26.9446611	379.417369	19.4786388	moena
33.9264117	652.989369	25.5536567	moena
31.5314408	593.044435	24.3525037	moena
39.6015614	410.712137	20.2660341	moena
26.638111	420.774402	20.5127863	moena
45.6508938	605.985965	24.6167822	moena
44.5493767	633.027841	25.1600445	moena
42.0000295	582.829759	24.1418674	moena
33.4752714	424.429535	20.6016877	moena
27.0225229	291.346103	17.0688636	moena
30.5325535	416.952848	20.4194233	moena
32.6620699	505.887371	22.4919401	moena
37.2315865	336.678578	18.3488032	moena
27.6192167	344.07496	18.5492577	moena
51.9046487	605.885027	24.6147319	moena
46.144664	566.971784	23.8111693	moena
29.5166919	276.714437	16.6347359	moena
37.3026081	517.340853	22.7451281	moena
32.3657867	425.753929	20.6338055	moena
30.705806	448.256386	21.1720662	moena
26.9446611	379.417369	19.4786388	moena
33.9264117	652.989369	25.5536567	moena
31.5314408	593.044435	24.3525037	moena
39.6015614	410.712137	20.2660341	moena
26.638111	420.774402	20.5127863	moena
45.6508938	605.985965	24.6167822	moena
44.5493767	633.027841	25.1600445	moena
42.0000295	582.829759	24.1418674	moena
33.4752714	424.429535	20.6016877	moena
27.0225229	291.346103	17.0688636	moena
30.5325535	416.952848	20.4194233	moena
32.6620699	505.887371	22.4919401	moena
37.2315865	336.678578	18.3488032	moena
27.6192167	344.07496	18.5492577	moena
51.9046487	605.885027	24.6147319	moena
46.144664	566.971784	23.8111693	moena
29.5166919	276.714437	16.6347359	moena
37.3026081	517.340853	22.7451281	moena
32.3657867	425.753929	20.6338055	moena

30.705806	448.256386	21.1720662	moena
26.9446611	379.417369	19.4786388	moena
33.9264117	652.989369	25.5536567	moena
31.5314408	593.044435	24.3525037	moena
39.6015614	410.712137	20.2660341	moena
26.638111	420.774402	20.5127863	moena
45.6508938	605.985965	24.6167822	moena
44.5493767	633.027841	25.1600445	moena
42.0000295	582.829759	24.1418674	moena
33.4752714	424.429535	20.6016877	moena
27.0225229	291.346103	17.0688636	moena
30.5325535	416.952848	20.4194233	moena
32.6620699	505.887371	22.4919401	moena
20.9221497	181.74508	13.4812863	tornillo
22.3645066	214.797714	14.6559788	tornillo
23.1213977	230.250328	15.1740017	tornillo
17.2602046	135.962825	11.6603098	tornillo
23.9496649	263.063366	16.2192283	tornillo
20.9096229	221.605434	14.8864178	tornillo
20.882431	241.434434	15.5381606	tornillo
27.8641735	425.199763	20.6203725	tornillo
19.0762617	208.044342	14.4237423	tornillo
26.6755255	385.758716	19.6407412	tornillo
27.3659853	390.80039	19.7686719	tornillo
30.4770061	385.070496	19.6232132	tornillo
23.4585237	238.440453	15.4415172	tornillo
25.9740652	288.526309	16.9860622	tornillo
25.5203123	297.58426	17.2506307	tornillo
26.6544895	339.34027	18.4211908	tornillo
21.5786709	256.160264	16.0050075	tornillo
28.1225071	379.613621	19.4836758	tornillo
28.281856	349.13431	18.6851361	tornillo
26.9282957	317.116332	17.8077604	tornillo
22.3654191	222.490639	14.9161201	tornillo
27.9055989	311.129245	17.6388561	tornillo
23.2384497	233.326904	15.2750419	tornillo
25.6362893	291.660118	17.0780596	tornillo
32.293828	472.680759	21.7412226	tornillo
23.0334397	271.198145	16.4680948	tornillo
21.423479	252.818629	15.9002714	tornillo
24.4162383	341.823849	18.4884788	tornillo
30.1356165	484.768123	22.0174504	tornillo
27.4055283	394.140027	19.8529602	tornillo
34.4768108	598.816539	24.4707282	tornillo
30.1785977	387.143054	19.6759512	tornillo
25.3581275	283.726551	16.8441845	tornillo
22.9056022	235.060594	15.331686	tornillo

26.5404079	321.247192	17.92337	tornillo
25.6465103	340.039368	18.4401564	tornillo
24.5659193	337.701896	18.3766672	tornillo
26.0582848	325.020283	18.0283189	tornillo
30.7421078	401.170781	20.0292481	tornillo
25.1537083	272.610553	16.5109222	tornillo
28.8713961	354.175787	18.8195586	tornillo
23.2359733	227.031226	15.0675554	tornillo
24.1580606	258.412818	16.0752237	tornillo
22.0795784	222.569543	14.9187648	tornillo
26.2680344	317.715925	17.8245877	tornillo
29.8787401	438.312255	20.9359083	tornillo
24.9540861	335.913484	18.3279427	tornillo
28.4609591	452.610818	21.274652	tornillo
23.9887238	317.111107	17.8076137	tornillo
28.1315498	429.145606	20.7158299	tornillo
26.7763598	368.2886	19.1908468	tornillo
28.2649195	346.927026	18.6259772	tornillo
27.4401069	329.716692	18.1581027	tornillo
27.8952417	339.893825	18.4362096	tornillo
21.8799447	219.843938	14.8271352	tornillo
29.2795436	425.943909	20.6384086	tornillo
22.5390441	286.940915	16.9393304	tornillo
23.4330205	272.82821	16.5175122	tornillo
28.8436183	376.661401	19.4077665	tornillo
30.9769749	424.342312	20.5995707	tornillo
25.6101625	284.633039	16.8710711	tornillo
23.4397508	225.078006	15.0026	tornillo
27.5719554	327.100094	18.0859087	tornillo
23.3121562	244.483778	15.635977	tornillo
26.4689334	328.376495	18.1211615	tornillo
22.3966596	254.63974	15.9574353	tornillo
23.9962353	314.332426	17.7294226	tornillo
26.8068635	423.316547	20.5746579	tornillo
21.7520317	257.65282	16.0515675	tornillo
27.5337285	397.12199	19.9279199	tornillo
34.6085001	602.15307	24.5388074	tornillo
24.1459474	244.013273	15.6209242	tornillo
33.5075147	491.233467	22.1637873	tornillo
28.8192574	370.294611	19.2430406	tornillo
26.7697018	331.720393	18.2131928	tornillo
23.9669066	298.026507	17.2634442	tornillo
22.994064	297.568585	17.2501764	tornillo
22.1709763	245.383446	15.6647198	tornillo
34.1864046	488.19329	22.0950965	tornillo
26.9062488	313.162558	17.6963996	tornillo
24.8367272	261.30549	16.1649464	tornillo

25.6991529	273.449904	16.5363208	tornillo
20.1340722	181.301426	13.4648218	tornillo
26.9564526	327.064197	18.0849163	tornillo
26.9273488	334.03049	18.276501	tornillo
22.0299392	241.192673	15.5303791	tornillo
27.1262965	393.48925	19.8365635	tornillo
24.3472649	335.970958	18.3295106	tornillo
19.7404618	214.73985	14.6540046	tornillo
26.2048443	368.318886	19.1916358	tornillo
35.3505472	631.251017	25.1247093	tornillo
21.1255596	190.536351	13.8034905	tornillo
24.1864242	258.632285	16.0820485	tornillo
29.9023527	398.854704	19.9713471	tornillo
25.9694471	313.772648	17.7136289	tornillo
24.6439091	306.05704	17.494486	tornillo
26.7013339	390.908386	19.7714032	tornillo
22.6968569	256.545976	16.0170527	tornillo
23.1638045	232.703721	15.2546295	tornillo
27.1545912	321.532724	17.9313336	tornillo

Anexo 4. Lista de algoritmos supervisados tomados para la población.

<b>N°</b>	<b>Algoritmos Supervisados de Clasificación</b>
1	CNN (red neuronal convolucional)
2	SVM (Support Vector Machine)
3	KNN (K- Nearest Neighbors)
4	BP (BackPropagation)
5	PNN (Probabilistic Neural Network)
6	ANN (Red neuronal artificial)
7	LDA (Linear Discriminat Analysis)
8	RF (Random Forest)
9	CART (Árbol de decisión)
10	VGG16 (Red Neuronal de Convolución de 16 capas)
11	VGG19(Red Neuronal de Convolución de 19 capas)
12	MLP (Multi-Layer Perceptron)
13	D-CNN (Deep Convolutional Neural Network)
14	NAIVE BAYES
15	RESNET

*Nota:* Elaboración Propia.