

**Detección de mosquitos *Aedes Aegypti*
y *Aedes Albopictus* (vectores del virus
del dengue) en zonas urbanas utilizando
aprendizaje profundo y un framework
de adquisición de datos**

Protocolo de Tesis

Autor:

David Calderón ceja

Estudiante de la Licenciatura en Tecnologías para la
Información en Ciencias

418126455

dcc.david.calderon@gmail.com

Asesor:

Dr. Sergio Rogelio Tinoco Martínez

Técnico académico de la Licenciatura en Tecnologías
para la Información en Ciencias

Escuela Nacional de Estudios Superiores Unidad
Morelia

stinoco@enesmorelia.unam.mx

Coasesor:

Dr. Heberto Ferreira M.

Técnico Académico Titular B, TC, definitivo.

Instituto de Investigaciones en Ecosistemas y
Sustentabilidad, UNAM

hferreira@cieco.unam.mx

Índice

1. Resumen	1
2. Antecedentes	2
2.1. Mosquitos y la transmisión del virus Dengue	2
2.2. Aprendizaje automático	2
2.2.1. Aprendizaje supervisado	3
2.2.2. Aprendizaje no supervisado	4
2.2.3. Aprendizaje por refuerzo	4
2.3. Reconocimiento de objetos	5
2.4. Redes neuronales	5
2.4.1. Perceptrón simple	5
2.4.2. Aprendizaje profundo	6
2.4.3. Redes neuronales profundas	7
2.5. Estado del arte	8
2.5.1. Mosquito Larva Classification Method Base on Convolutional Neuronal Networks	8
2.5.2. Diseño e implementación de un sistema de detección de mosquitos basado en patrones creación de la plataforma Web y servicios seguros	8
3. Hipótesis	9
4. Justificación	9
5. Objetivo	9
5.1. Objetivo General	9
5.2. Objetivos particulares	9
6. Metodología	10

1. Resumen

Este trabajo forma parte del proyecto PAPIME con clave del proyecto: PE106021 Nombre del proyecto: "Propuesta de mejora a la enseñanza del aprendizaje automático aplicado a la Ciencia de Datos a gran escala; dirigido a académicos y estudiantes de la licenciatura en Tecnologías para la Información en Ciencias (LTICs) en la ENES Morelia."

Este trabajo se enfoca en la detección de casos de dengue en la zona urbana de Morelia, de los cuales los mosquitos de la familia Aedes Aegypti y Aedes Albopictus son los principales agentes de transmisión de las enfermedades del dengue, zika y chikungunya que se convierten en un problema de salud pública. Se propone mejorar un modelo de detección basado en aprendizaje profundo desarrollado previamente, para obtener una mejor precisión en la detección de

estos mosquitos (vectores de infección), en imágenes adquiridas con dispositivos móviles utilizando una aplicación desarrollada para tal fin.

Se desarrolla un modelo de detección utilizando un conjunto de datos de imágenes de mosquitos obtenidas en bases de datos y repositorios públicos. La finalidad es establecer una forma más eficiente de tratar los vectores de infección de las enfermedades producidas por el *Aedes Aegypti* y el *Aedes Albopictus* en la zona urbana de Morelia.

2. Antecedentes

2.1. Mosquitos y la transmisión del virus Dengue

El dengue es una enfermedad viral reconocida por la Organización Mundial de la Salud (OMS) que se transmite por medio de los mosquitos hembra principalmente de la especie *Aedes aegypti* y, en menor grado, de la especie *Aedes Albopictus*. Además, que son vectores de los virus de la fiebre Chikunguña, la fiebre amarilla y el Zika. La enfermedad esta fuertemente ligada a los trópicos cada una con variaciones locales.

El dengue puede causar una gran gama de patologías, que puede abarcar desde una enfermedad asintomática hasta llevar síntomas graves para las personas que contraen el virus. Existe una variante del dengue conocida como Dengue hemorrágico su origen está ligado a los primeros años de la década de 1950 a 1960 que tuvo lugar en Tailandia y Filipinas aunque hoy en días se puede encontrar alrededor de los países asiáticos y en América latina convirtiéndose en una de las principales causas de hospitalización y muerte de niños y adultos de estas regiones. [1]

El causante del dengue es un virus de la familia *Flaviviridae* que tiene cuatro serotipos distintos, aunque estrechamente emparentados: DENV-1, DENV-2, DENV-3 y DENV-4. Existe las creencias que la recuperación de la infección del virus del dengue otorga inmunidad de por vida contra el tipo de serotipo que causo la infección, sin embargo, la inmunidad cruzada a los otros serotipos tras la recuperación es temporal o parcial. La reincidencia de la infección causada por los otros serotipos aumenta considerablemente el riesgo de padecer Dengue hemorrágico [1]

2.2. Aprendizaje automático

El aprendizaje automático es una rama de la inteligencia artificial (IA) y la informática que se centra en el uso de datos y algoritmos para imitar la forma en que aprenden los humanos, mejorando gradualmente su precisión además el aprendizaje automático es un componente del campo de la ciencia de datos, Mediante el uso de métodos estadísticos, se entrenan algoritmos para realizar

clasificaciones o predicciones.

El algoritmo del aprendizaje automático se consta de tres partes principales.

- Un proceso de decisión: en general, los algoritmos de aprendizaje automático se utilizan para hacer una predicción o clasificación. Según algunos datos de entrada, que pueden estar etiquetados o no, su algoritmo producirá una estimación sobre un patrón en los datos.
- Una función de error: una función de error sirve para evaluar la predicción del modelo. Si hay ejemplos conocidos, una función de error puede hacer una comparación para evaluar la precisión del modelo.
- Un proceso de optimización del modelo: si el modelo puede ajustarse mejor a los puntos de datos del conjunto de entrenamiento, los pesos se ajustan para reducir la discrepancia entre el ejemplo conocido y la estimación del modelo. El algoritmo repetirá este proceso de evaluación y optimización, actualizando los pesos de forma autónoma hasta que se haya alcanzado un umbral de precisión.

Los sistemas de aprendizaje automático se pueden clasificar según su entrenamiento. Hay tres categorías principales: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo.

2.2.1. Aprendizaje supervisado

El aprendizaje supervisado, también conocido como aprendizaje automático supervisado, es una subcategoría del aprendizaje automático y la inteligencia artificial. Se define por el uso de conjuntos de datos etiquetados para entrenar algoritmos que clasifiquen datos o predigan resultados con precisión. A medida que los datos de entrada se introducen en el modelo, este ajusta sus pesos hasta que el modelo se ha ajustado adecuadamente, lo que ocurre como parte del proceso de validación cruzada.

El aprendizaje supervisado utiliza un conjunto de datos para enseñar a los modelos a producir el resultado deseado. Este conjunto de datos de entrenamiento incluye entradas y salidas correctas, que permiten que el modelo aprenda con el tiempo. El algoritmo mide su precisión a través de la función de pérdida, ajustándose hasta que el error se haya minimizado lo suficiente.

El aprendizaje supervisado se puede dividir en dos tipos de problemas durante la minería de datos: clasificación y regresión. La clasificación utiliza un algoritmo para asignar con precisión los datos de prueba en categorías específicas. Reconoce entidades específicas dentro del conjunto de datos e intenta sacar algunas conclusiones sobre cómo esas entidades deben etiquetarse o definirse. La regresión se utiliza para comprender la relación entre variables dependientes e independientes. [2]

los algoritmos de aprendizaje supervisado mas importantes son:

- K-Vecinos más cercanos
- Regresión lineal
- Regresión logística
- Máquinas de vectores de soporte (SVM)
- Árboles de decisión y bosques aleatorios
- Redes neuronales

2.2.2. Aprendizaje no supervisado

El aprendizaje no supervisado, también conocido como aprendizaje automático no supervisado, utiliza algoritmos de aprendizaje automático para analizar y agrupar conjuntos de datos no etiquetados. Estos algoritmos descubren patrones ocultos o agrupaciones de datos sin necesidad de intervención humana. Su capacidad para descubrir similitudes y diferencias en la información lo convierte en la solución ideal para el análisis exploratorio de datos, estrategias de venta cruzada, segmentación de clientes y reconocimiento de imágenes. Los modelos de aprendizaje no supervisados se utilizan para tres tareas principales: agrupación, asociación y reducción de la dimensionalidad.[3]

los algoritmos de aprendizaje supervisado mas importantes son:

- Agrupación(Clustering)
- Clustering exclusivo y superpuesto(Exclusive and Overlapping Clustering)
- Agrupación jerárquica(Hierarchical clustering)
- Reglas de asociación(Association Rules)
- Algoritmos a priori(Apriori algorithms)
- Reducción de dimensionalidad(Dimensionality reduction)
- Análisis de componentes principales(Principal component analysis)
- Valor singular de descomposición(SVD)

2.2.3. Aprendizaje por refuerzo

El aprendizaje por refuerzo es diferente de los otros tipos de aprendizaje, pero se alinea más con el extremo del aprendizaje supervisado. En el aprendizaje por refuerzo, el mapeo del estado a la acción se aprende a través de una recompensa o castigo acumulativo por sus acciones. El mapeo se lleva a cabo en línea, a través de un equilibrio de exploración (probar nuevas acciones para un estado dado) y explotación (utilizando el conocimiento existente del mapeo de estado/acción).[4]

2.3. Reconocimiento de objetos

El reconocimiento de objetos en imágenes o videos por medio de la visión computacional sigue siempre la misma estructura de trabajo:

- Adquisición de la imagen.
- Digitalización.
- Extracción de características: es la técnica que consiste en la búsqueda de patrones en una imagen, ya sea de tipo geométrico, topológico, estadístico, etc.
- Método de clasificación: se compone de asignar a cada clase de objeto que se busca reconocer, un patrón o conjunto de patrones características, de manera que se pueda asociar y clasificar cada uno de ellos.

Así, cada vez que se disponga de una nueva imagen a clasificar, se obtendrán sus características principales y éstas serán comparadas con aquellas que ya se habían estudiado para cada caso de objeto. En función de la similitud encontrada, la imagen se asocia a una clase u otra, definiendo así la probabilidad que tendría de pertenecer a dicha clase. A mayor similitud, mayor probabilidad se consigue.

Cuando se asocian patrones de características a cada objeto, es usual que se defina un vector de características. Se define de esta manera un patrón como el conjunto de características de una imagen.

$$x = [x_1, x_2, x_3, \dots, x_n]^T$$

x representa el patrón que se estudia mientras x_i son las características que los componen.

Para cada clase hay que encontrar una función de decisión (clasificador) de manera que si x pertenece a w y no a la clase v entonces $d_W(x) > d_V(x)$. [5]

2.4. Redes neuronales

Las redes neuronales, también conocidas como redes neuronales artificiales (ANN) o redes neuronales simuladas (SNN), son un subconjunto del aprendizaje automático y están en el corazón de los algoritmos de aprendizaje profundo. Su nombre y estructura están inspirados en el cerebro humano, imitando la forma en que las neuronas biológicas se envían señales entre sí. [6]

2.4.1. Perceptrón simple

El Perceptrón Simple es una de las arquitecturas ANN más simples, inventada en 1957 por Frank Rosenblatt. Se basa en una neurona artificial ligeramente diferente [3] denominada unidad lógica de umbral (TLU) o, a veces, unidad de

umbral lineal (LTU). Las entradas y salidas son números (en lugar de valores binarios de encendido/apagado), y cada conexión de entrada está asociada con un peso. La TLU calcula una suma ponderada de sus entradas definidas como

$$Z = w_1x_1 + w_2x_2 + \dots + w_nx_n = x^T w$$

luego aplica una función de paso a esa suma y genera la salida, La función de paso más común utilizada en los perceptrones es la función de paso de Heaviside. A veces se usa la función de signo en su lugar.[7]

$$Heaviside(z) = \begin{cases} z = 0, & \text{si } z < 0 \\ z = 1, & \text{si } z \geq 0 \end{cases}$$

$$sgn(z) = \begin{cases} z = -1, & \text{si } z < 0 \\ z = 0, & \text{si } z = 0 \\ z = +1, & \text{si } z > 0 \end{cases}$$

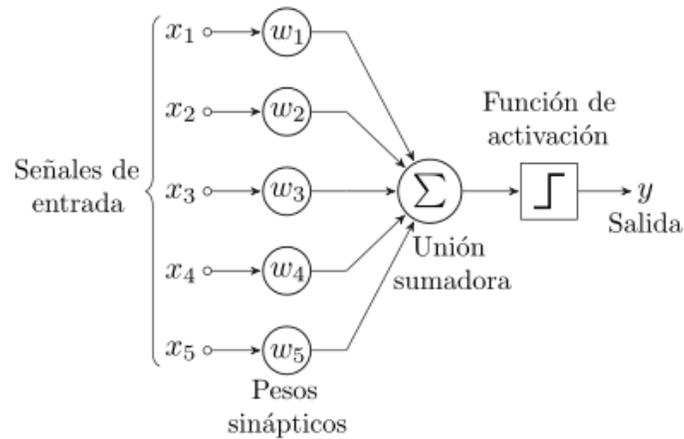


Figura 1: Perceptrón simple

Se puede utilizar una única TLU para la clasificación binaria lineal simple. Calcula una combinación lineal de las entradas y , si el resultado supera un umbral, genera la clase positiva. De lo contrario, genera la clase negativa (al igual que una regresión logística o un clasificador SVM lineal)[7]

2.4.2. Aprendizaje profundo

Los primeros esfuerzos en el desarrollo de algoritmos de aprendizaje profundo vinieron de Alexey Grigoryevich Ivakhnenko (desarrolló el *Group Method of Data Handling* [8]) y Valentin Grigorevich Lapa (autor de *Cybernetics and Forecasting Techniques* [9]) en 1965. Utilizaron modelos con funciones de activación polinomiales, que luego fueron analizados estadísticamente. De cada capa,

las mejores características elegidas estadísticamente se reenviaron a la siguiente capa (un proceso manual y lento).

El aprendizaje profundo es un subconjunto del aprendizaje automático, que es esencialmente una red neuronal con tres o más capas. Estas redes neuronales intentan simular el comportamiento del cerebro humano, aunque lejos de igualar su capacidad, lo que le permite .aprender.^a partir de grandes cantidades de datos. Si bien una red neuronal con una sola capa aún puede hacer predicciones aproximadas, las capas ocultas adicionales pueden ayudar a optimizar y refinar la precisión.[10]

2.4.3. Redes neuronales profundas

Las redes neuronales profundas consisten en múltiples capas de nodos interconectados, cada uno de los cuales se basa en la capa anterior para refinar y optimizar la predicción o categorización. Esta progresión de cálculos a través de la red se llama propagación directa. Las capas de entrada y salida de una red neuronal profunda se denominan capas visibles. La capa de entrada es donde el modelo de aprendizaje profundo ingiere los datos para su procesamiento, y la capa de salida es donde se realiza la predicción o clasificación final.

Otro proceso llamado retro-propagación usa algoritmos, como el descenso de gradiente, para calcular los errores en las predicciones y luego ajusta los pesos y sesgos de la función moviéndose hacia atrás a través de las capas en un esfuerzo por entrenar el modelo. Juntas, la propagación directa y la retro-propagación permiten que una red neuronal haga predicciones y corrija cualquier error en consecuencia. Con el tiempo, el algoritmo se vuelve gradualmente más preciso.

Lo anterior describe el tipo más simple de red neuronal profunda en los términos más simples. Sin embargo, los algoritmos de aprendizaje profundo son increíblemente complejos y existen diferentes tipos de redes neuronales para abordar problemas o conjuntos de datos específicos. Por ejemplo, Redes neuronales convolucionales (CNN), Redes neuronales recurrentes (RNN).[10]

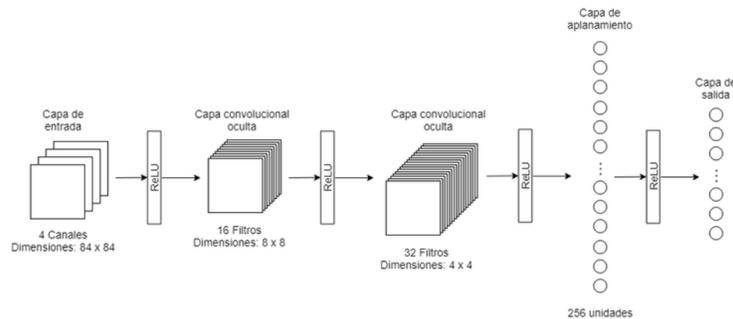


Figura 2: Arquitectura de una Red neuronal convolucional

2.5. Estado del arte

2.5.1. Mosquito Larva Classification Method Base on Convolutional Neuronal Networks

este trabajo plantea un sistema para la detección de larvas de mosquitos de las especies Aedes basado en Redes Neuronales Convolucionales. Con un conjunto de datos de larvas se procede a entrenar un modelo que distinga entre dos géneros de larvas, “Aedes” y “Otros géneros”.

Las imágenes del conjunto de datos de larvas se procesan usando un algoritmo de aprendizaje automático para la tarea de clasificación, con el método desarrollado proponen que el proceso de identificación de larvas es más eficiente, automático y rápido que los métodos tradicionales.

Los resultados demuestran que desempeña bien la tarea propuesta lo que demuestra que se puede aplicar al mundo real.[11]

Las aportaciones de este proyecto es un nuevo sistema de clasificación de imágenes de larvas que consta de 4 pasos:

- Adquisición de datos
- Pre-procesamiento de imágenes
- Entrenamiento de una Red Neuronal Convolutiva
- Clasificación en tiempo real

2.5.2. Diseño e implementación de un sistema de detección de mosquitos basado en patrones creación de la plataforma Web y servicios seguros

Este trabajo consiste en el desarrollo de un sistema de software REST API basado en una aplicación web con la finalidad de detectar la presencia de mosquitos Aedes Aegypti y Aedes Albopictus en poblaciones urbanas debido a que generan enfermedades como el Dengue, Zika y Chikunguña.

Implementa una red neuronal convolutiva (CNN) para hacer uso del Aprendizaje profundo (DL) para mediante un servicio web desarrollar 3 tareas:

- Publicar el servicio para enviar imágenes
- Almacenar y clasificar las imágenes
- Analizar la similitud de las especies de mosquitos

Los resultados son guardados en una base de datos para su posterior consulta a través del servicio web, se proporciona el informe de los análisis de similitud de la imagen. El desarrollo del proyecto consta de 5 fases:

1. Almacenamiento de imágenes
2. Tratamiento de imágenes utilizando aprendizaje profundo
3. Uso de una Red Neuronal Convolutiva
4. Uso del modelo de aprendizaje para establecer similitud
5. Reportar resultado

El espectro de similitud que logra el proyecto esta en el rango de 60 % y 90 % por lo cual se puede determinar que es viable el proyecto.[12]

3. Hipótesis

Utilizar un modelo de Deep Learning permitirá la detección de mosquitos Aedes Aegypti y Aedes Albopictus con un porcentaje de precisión mayor al 95 % en imágenes adquiridas con una aplicación móvil, que permita tomar decisiones sanitarias adecuadas

4. Justificación

Se ha presentado un incremento de casos de dengue en la zona urbana de Morelia [13], de los cuales los mosquitos de la familia Aedes Aegypti y Aedes Albopictus son los principales vectores de transmisión de la enfermedad para el cual se implementó un servicio web para su detección y clasificación [12]. Se propone mejorar el modelo para obtener una precisión mas alta en su detección. Con esto se tendrá una forma más eficiente de aplicar fumigaciones en áreas detectadas y evitar así el impacto innecesario en la salud humana, debido a que para erradicar el problema de los mosquitos se rocían pesticidas los cuales pueden causar daños por exposición directa o prolongada.

5. Objetivo

5.1. Objetivo General

Desarrollar un modelo de Deep Learning que detecte con una precisión mayor al 95 % la presencia en imágenes del mosquito vector del dengue(Aedes Aegypti y Aedes Albopictus) utilizando servicios web para la toma de decisiones sanitarias.

5.2. Objetivos particulares

1. investigar y analizar el estado del arte con respecto a la detección de mosquitos(Aedes Aegypti y Aedes Albopictus) utilizando modelos de Deep Learning.

2. Obtener conjuntos de imágenes por diferentes medios (Bases de datos, repositorios, etc.) correspondientes a los mosquitos de las familias *Aedes Aegypti* y *Aedes Albopictus* en un orden de cientos o miles. especialmente por medio del Instituto de la Salud pública (Cuernavaca, Morelos) en el área de enfermedades infecciosas, parasitarias y transmitidas por vector.
3. Seleccionar y etiquetar las imágenes para el entrenamiento del modelo.
4. Establecer un modelo Deep Learning apropiado para la detección de mosquitos *Aedes Aegypti* y *Aedes Albopictus*.
5. Implementar y entrenar el modelo realizando pruebas que determinen los meta parámetros adecuados para lograr una precisión mayor al 95 %.
6. Acoplar el modelo en el framework de trabajo montado en un servicio web.
7. Reportar el desempeño resultante del modelo.
8. Ajustar el conjunto de datos para asegurar consistencia y, de ser necesario, evaluar nuevamente las métricas de desempeño del modelo.

6. Metodología

La metodología que se plantea en este proyecto consiste en dividir las siguientes actividades en un lapso no mayor a un año, con inicio el mes de enero del 2023.

1. Buscar diferentes modelos para la detección de mosquitos para incorporar un modelo óptima con las metodologías adecuadas.
2. Obtener imágenes de alta calidad por diferentes medios (Bases de datos, repositorios, etc.) correspondientes a los mosquitos de las familias *Aedes Aegypti* y *Aedes Albopictus*
 - Investigar los modelos previamente desarrollados en el área.
 - Se determinará como parámetro de selección aquellas imágenes que cumplan con una resolución mayor o igual a 800x600 pixeles de largo por ancho.
 - Extraer imágenes de la base de datos de la Red Europea de Entomología Digital Mosquito Alert [14] y descargar imágenes del repositorio de Kaggle “*Aedes Mosquitos Dataset*” [15].
3. Seleccionar y etiquetar las imágenes útiles para el entrenamiento del modelo.
 - Descartar las imágenes que no cumplan el criterio de resolución previamente establecido.

- Utilizar el software de LabelImg para generar las cajas y la etiqueta correspondiente a la familia del mosquito *Aedes Aegypti* o *Aedes Albopictus*.
 - Generar un archivo XML de cada imagen con sus meta parámetros de los cuales se almacenan: nombre de la imagen, ubicación del archivo, Ancho de la imagen, largo de la imagen, numero de capas (RGB o blanco y negro), nombre de la etiqueta de clase, largo de la caja, ancho de la caja, Pixel x de inicio, Pixel y de inicio.
4. Establecer un modelo Deep Learning apropiado para la detección de mosquitos *Aedes Aegypti* y *Aedes Albopictus*.
- Seleccionar y descargar un modelo pre-entrenado en lenguaje de programación Python del repositorio de TensorFlow llamado “TensorFlow 2 Detection Model Zoo” que consta de una colección de modelos de detección entrenados previamente en el conjunto de datos COCO-2017. Estos modelos pueden ser útiles para modelos de detección de caja ya que estos fueron entrenados con dichos datasets.
5. Implementar y entrenar el modelo realizando pruebas que determinen los meta parámetros adecuados.
- Instalar la API (interfaz de programación de aplicaciones) TensorFlow 2 Object Detection API para un sistema operativo Windows 10 con una versión de Python 3.9, CUDA Toolkit 11.2 y TensorFlow 2.5 en un ambiente de Anaconda Python.
 - a) Crear un ambiente de Anaconda con las versiones de las librerías necesarias.
 - b) Instalar la API de COCO como dependencia de la API de TensorFlow 2 Object Detection.
 - c) Comprobar la build de la instalación corriendo los archivos de prueba del API
 - Entrenar la parte de Detección de objetos personalizados del modelo pre-entrenado previamente descargado.
 - a) Colocar el modelo pre-entrenado en la carpeta del API.
 - b) Dividir el dataset en las dos carpetas que corresponden al Train, Test del API con las imágenes y archivos XML de cada registro.
 - c) Configurar y ejecutar el archivo del Data Loader para especificar el numero de clases y darle una etiqueta a cada una tal que el modelo entienda cual pertenece a cada individuo.
 - d) Usar el archivo del PipeLine del Modelo pre-entrenado para determinar los meta parámetros (Numero de clases, épocas, función de perdida, optimizador, etc.) para el ajuste de la parte de detección de objetos personalizados.
 - e) Entrenar el modelo ya con todos los pasos configurados.

- Visualizar los resultados del modelo para determinar si los resultados son satisfactorios.
 - Ajuste del conjunto de datos para asegurar consistencia. Se ajustará la calidad de las imágenes obtenidas de repositorios que superan la calidad de imágenes obtenidas con dispositivos móviles a través de la App para Android, con la finalidad de igualar la distribución de probabilidad objetivo(imágenes de baja calidad de los dispositivos móviles) con la de entrenamiento(imágenes de alta calidad de repositorios).
6. Integrar el modelo entrenado con el servicio web para la detección de mosquitos.
 - Congelar el modelo una vez finalizado el entrenamiento usando los puntos de verificación (checkpoints) que se generan durante el entrenamiento.
 - Incorporar el modelo entrenado en el framework de la aplicación “No mosquito”, trabajo previamente realizado en la tesis “Diseño e implementación de un sistema de detección de mosquitos basado en patrones creación de la plataforma Web y servicios seguros” [12]
 7. Medir la eficiencia en la detección por medio de la visualización de métricas tales como la función de pérdida y la exactitud a través de cada época de entrenamiento.

Cronograma de actividades												
2023												D
Actividad												N
E	F	M	A	M	J	J	A	S	O	N	D	
Investigar y analizar el estado del arte con respecto a la detección de mosquitos (Aedes Aegypti y Aedes Albopictus) utilizando modelos de Deep Learning.												
Investigar los modelos previamente desarrollados en el área.												
Determinar como parámetro de selección aquellas imágenes que cumplan con el criterio de selección.												
Extraer imágenes de la base de datos de la Red Europea de Entomología Digital Mosquito Alert y descargar imágenes del repositorio de Kaggle "Aedes Mosquitos Dataset".												
Descartar las imágenes que no cumplan el criterio de resolución previamente establecido.												
Utilizar el software de Labelmg para generar las cajas y la etiqueta correspondiente a la familia del mosquito Aedes Aegypti o Aedes Albopictus.												
Generar un archivo XML de cada imagen con sus meta parámetros adecuados												
Seleccionar y descargar un modelo pre-entrenado del repositorio "TensorFlow 2 Detection Model Zoo".												
Instalar la API (interfaz de programación de aplicaciones) TensorFlow 2 Object Detection.												
Entrenar la parte de Detección de objetos personalizados del modelo pre-entrenado.												
Visualizar los resultados del modelo.												
Congelar el modelo una vez finalizado el entrenamiento usando los puntos de verificación (checkpoints).												
Incorporar el modelo entrenado en el framework de la aplicación "No mosquito"												
Medir la eficiencia en la detección por medio de la visualización de métricas tales como la función de pérdida y la exactitud a través de cada época de entrenamiento												
Publicación del proyecto a manera de modo de titulación.												
Redactar del documento correspondiente a la tesis Defensa y exposición de la tesis.												

Figura 3: Cronograma de actividades

Referencias

- [1] Organización Mundial de la Salud, “Dengue y dengue grave,” 2022.
- [2] G. D. Greenwade, “The Comprehensive Tex Archive Network (CTAN),” *TUGBoat*, vol. 14, no. 3, pp. 342–351, 1993.
- [3] IBM Cloud Education, “What is unsupervised learning?.”
- [4] M. T. Jones, “Train a software agent to behave rationally with reinforcement learning.”
- [5] V. R. Marina, “Reconocimiento de objetos usando deep learning,” 2016.
- [6] IBM Cloud Education, “Neural networks.”
- [7] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras Tensor-Flow*. Canada: O’Reilly Media, Inc., 2 ed., 2019.
- [8] A. G. Ivakhnenko, “The Group Method of Data Handling-A Rival of the Method of Stochastic Approximation.,” *Soviet Automatic Control*, vol. 1, no. 3, pp. 43–55, 1968.
- [9] A. Ivakhnenko and V. Lappa, *Cybernetics and Forecasting Techniques*. New York, NY: American Elsevier Pub. Co., 1967.
- [10] IBM Cloud Education, “Deep learning.”
- [11]
- [12] M. C. ULISES, “Diseño e implementación de un sistema de detección de mosquitos basado en patrones creación de la plataforma web y servicios seguros,” 2021.
- [13] Coordinación General de Comunicación Social, “Sin tregua al dengue en morelia,” 2020.
- [14] Institución Catalana de Investigación y Estudios Avanzados, “Mosquito alert image dataset,” 2014.
- [15] P. ISAWASAN, “Aedes mosquitos dataset,” 2020.