

Capítulo 3

Metodología

3.1. Metodología general

El objetivo principal de esta tesis es desarrollar un sistema capaz de detectar de forma automática distintos tipos de criaderos de mosquitos a partir de imágenes aéreas capturadas por un dron. La metodología general seguida para alcanzar este objetivo se resume en la Figura 3.1, la cual ilustra de manera esquemática las etapas que conforman el proceso completo.

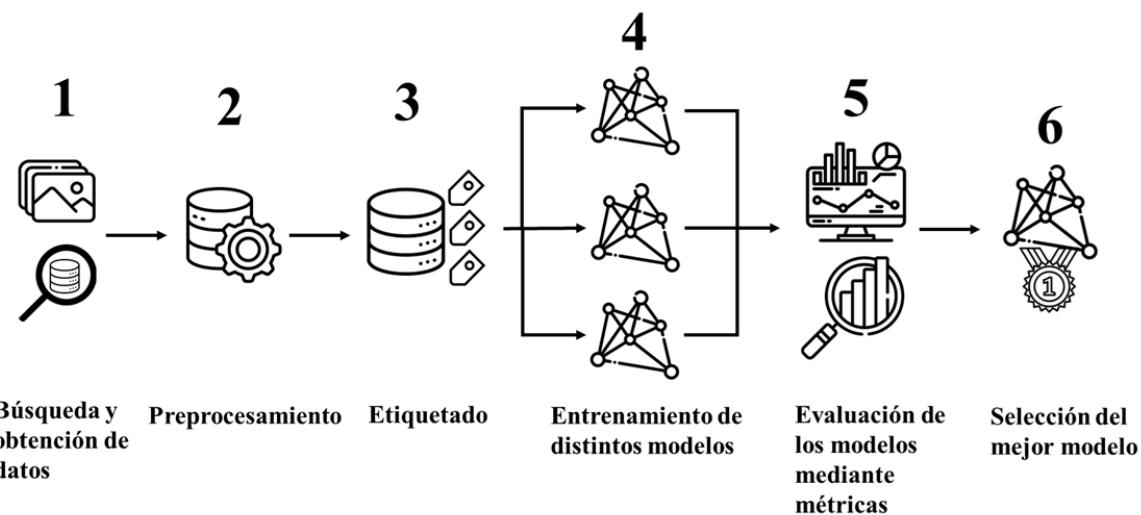


Figura 3.1: Metodología general. El esquema muestra las siete etapas principales llevadas a cabo para alcanzar los objetivos planteados en esta investigación.

A continuación, se describen brevemente las etapas mostradas en la figura:

1. **Búsqueda y obtención de datos:** Se recopilan los datos proporcionados por el Instituto Nacional de Salud Pública (INSP) y se realiza una búsqueda complementaria de conjuntos de datos públicos relacionados con la detección de criaderos de mosquitos disponibles en internet.
2. **Preprocesamiento:** Una vez obtenidos los conjuntos de datos, estos se someten a un proceso de preprocesamiento con el fin de estandarizar las imágenes, ajustar sus dimensiones, formatos y características según las necesidades del modelo. Las técnicas específicas pueden variar dependiendo del origen y formato de cada base de datos.
3. **Etiquetado:** Se lleva a cabo el etiquetado manual de las imágenes proporcionadas por el INSP, así como de aquellas obtenidas de fuentes externas en caso de no contar con anotaciones. En caso de que los datos ya estén etiquetados, se realiza la conversión de las etiquetas al formato requerido por el modelo seleccionado para el entrenamiento. Una vez preprocesados y etiquetados, los datos se encuentran listos para ser utilizados en la etapa de entrenamiento.
4. **Entrenamiento de distintos modelos:** Se entrena n distintos modelos de detección de objetos seleccionados bajo criterios comparables, como el número de parámetros y arquitectura, y utilizando configuraciones de hiperparámetros similares. Esto permite una evaluación equitativa del desempeño de los modelos.
5. **Evaluación de modelos mediante métricas:** Se evalúan los modelos entrenados mediante métricas estándar en detección de objetos, tales como *Precision*, *Recall*, *Average Precision* y *Mean Average Precision*. Estas métricas permiten cuantificar la capacidad de detección y exactitud de cada modelo.
6. **Selección del mejor modelo:** Finalmente, se selecciona el modelo con el mejor desempeño general, considerando tanto las métricas de evaluación como el costo computacional requerido para su implementación.

Cabe señalar que, si bien la metodología general se presenta de manera secuencial, se contempla el entrenamiento y evaluación de los modelos utilizando tres escenarios distintos: (i) únicamente con

el conjunto de datos proporcionado por el INSP, (ii) únicamente con los datos recolectados de fuentes en línea, y (iii) con la unión de ambos conjuntos. Esta estrategia tiene como propósito analizar el comportamiento y la capacidad de generalización de los modelos ante distintos tipos de datos.

3.2. Descripción de los datos

En esta sección se describe el conjunto de datos obtenidos: los datos de **El Vergel**, Tapachula Chiapas que fueron proporcionados por el INSP y el conjunto de datos **MBG-V2** que fue encontrado en la web, también se muestra el procesamiento de estos y el etiquetado realizado para dejar los conjuntos de entrenamiento estandarizados y listos.

3.2.1. Datos del Vergel, Tapachula Chiapas

El área de estudio donde se llevó acabo la recolección de datos y fotografías fue El Vergel, una colonia ubicada en la ciudad de Tapachula, Chiapas, México. Se trata de una zona residencial que forma parte del área urbana de Tapachula (véase la Figura 3.2).

Para la toma de fotografías utilizaron drones RTK de la marca DJI de 1,391 g, con una cámara con sensor CMOS de 1 pulgada y 20 megapixeles con obturador mecánico (espectro visible), con una resolución de $(H/36.5)$ cm/pixel, donde H es la altitud de la aeronave relativa a la escena grabada (expresada en metros), además de un sistema de transmisión de video HD Lightbridge. Este modelo de drone posee un sistema Flight Autonomy, con posicionamiento por satélite de doble banda (GPS y GLONASS), con control remoto y un alcance máximo de transmisión de 7 km. Posee un tiempo de vuelo máximo de 30 minutos, una velocidad máxima de vuelo de 50 km/h y una temperatura de funcionamiento de -20 °C a 40 °C. Las imágenes fueron capturadas durante la época de lluvia y simultáneo a las inspecciones entomológicas, socioeconómicas y demográficas realizadas por el INEGI. En cada una de las zonas de estudio donde realizaron los sobrevuelos llevaron a cabo los protocolos de vuelo elaborados por el CENAPRED, a fin de asegurar el éxito de las misiones y no violar ninguna norma.



Figura 3.2: Área de vuelo en El Vergel, Tapachula, Chiapas para la obtención de datos El Vergel. El Vergel se encuentra en la región del Soconusco, caracterizada por su clima cálido y húmedo durante la mayor parte del año. La zona es propensa a lluvias intensas, especialmente durante la temporada de huracanes, lo que puede generar riesgos de inundaciones. La colonia cuenta con una infraestructura urbana en desarrollo. Se han reportado deficiencias en servicios básicos, los habitantes han solicitado mejoras en los servicios públicos. Cuenta con una población cercana a 2,590 habitantes distribuidos en 592 hogares y con una edad promedio de 11 años [42]. Figura tomada de [43].

Datos proporcionados

El conjunto de datos proporcionado consistió en un total de 1,122 imágenes en alta resolución, cada una con dimensiones de $4,608 \times 3,456$ píxeles, en formato *JPG* (véase la figura 3.3).

Preprocesamiento

Debido a la altura desde la cual fueron tomadas las imágenes, se identificaron dos inconvenientes significativos:

- Las dimensiones elevadas de las imágenes hacen que los objetos de tamaño reducido, como los criaderos de mosquitos, se visualicen a una escala muy pequeña. Esto dificulta su identificación



Figura 3.3: Ejemplo de una imagen del conjunto de datos sin preprocesamiento. Las imágenes fueron capturadas a diferentes horas del día desde una altura aproximada de 100 metros. Debido a esta altitud, se obtiene una vista panorámica amplia, permitiendo abarcar alrededor de dos cuadras completas y partes de otras dos.

durante el proceso de etiquetado manual, ya que obliga al uso de herramientas externas para aplicar zoom y localizar dichos objetos, lo que además incrementa el riesgo de errores.

- Si las imágenes se utilizan con su resolución original, durante el entrenamiento del modelo estas deben ser redimensionadas para adaptarse al tamaño de entrada requerido por las redes neuronales, que comúnmente son de mucho menor resolución (por ejemplo 640×640). Este cambio implica una reducción significativa de la información visual, lo que puede provocar la desaparición de objetos pequeños en la imagen redimensionada y, en consecuencia, dificultar su detección por parte del modelo, afectando negativamente su rendimiento predictivo.

Con el fin de mitigar estos problemas, se decidió dividir cada imagen original en nueve subimágenes sin superposición (véase la Figura 3.4), (véase Algoritmo 1). Esta estrategia permite reducir la resolución de cada subimagen, facilitando la identificación de objetos pequeños y, al mismo tiempo, incrementar el tamaño total del conjunto de datos.

Algorithm 1 División de imágenes en 9 subimágenes sin intersección

Require: Carpeta de entrada `input_folder`, carpeta de salida `output_folder`

Ensure: Para cada imagen, 9 subimágenes guardadas en la carpeta de salida

```

1: Crear carpeta output_folder si no existe
2: for all archivo en input_folder do
3:   if archivo tiene extensión válida (.jpg, .png, etc.) then
4:     Abrir imagen y obtener dimensiones ( $W, H$ )
5:     Calcular  $w \leftarrow \lfloor W/3 \rfloor$ ,  $h \leftarrow \lfloor H/3 \rfloor$ 
6:     for  $i = 0$  to  $2$  do
7:       for  $j = 0$  to  $2$  do
8:          $x \leftarrow j \cdot w$ 
9:          $y \leftarrow i \cdot h$ 
10:        Recortar subimagen desde  $(x, y)$  hasta  $(x + w, y + h)$ 
11:        Guardar subimagen
12:      end for
13:    end for
14:  end if
15: end for
16: return Total de subimágenes generadas

```

Como resultado del preprocesamiento, se obtuvo un nuevo conjunto de datos compuesto por 10,098 imágenes en formato *JPG*, cada una con dimensiones de $1,536 \times 1,152$ píxeles (véase la Figura 3.5).

Este procedimiento resolvió los problemas previamente identificados: ahora es posible observar con mayor claridad los objetos de menor tamaño, lo que facilita su correcta anotación y reduce la probabilidad de omitir criaderos de mosquito. Asimismo, al contar con imágenes de menor tamaño, la pérdida de información durante el redimensionamiento para el entrenamiento de los modelos es considerablemente menor. En consecuencia, el preprocesamiento permitió estandarizar el conjunto de imágenes, dejándolas listas para el proceso de etiquetado posterior.

Categorías

Una vez concluido el proceso de preprocesamiento de los datos, fue necesario definir las categorías a considerar para este proyecto. Aunque existe una gran variedad de objetos que pueden funcionar como criaderos de mosquitos, las categorías a utilizar fueron determinadas por la frecuencia de aparición en el conjunto de datos.

Para identificar los tipos de criaderos con mayor frecuencia, se realizó una exploración visual del

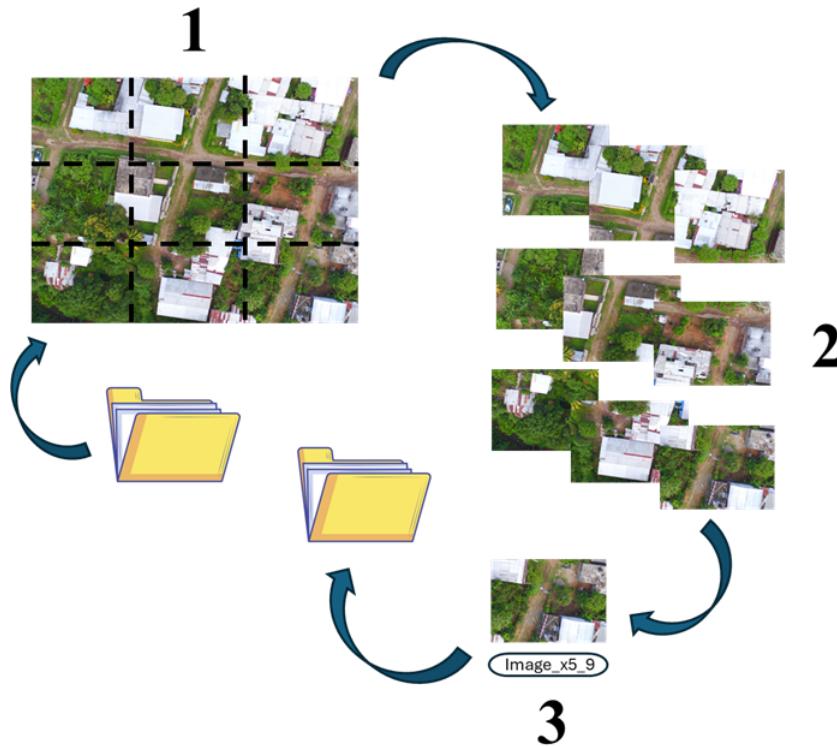


Figura 3.4: Recortes sin superposición. A partir de cada imagen original se generaron nueve recortes, los cuales fueron renombrados y almacenados en una carpeta separada para su posterior procesamiento.



Figura 3.5: Ejemplo de una imagen del conjunto de datos preprocesados. La reducción de escala permite una mejor visualización de objetos de menor tamaño, lo cual facilita el proceso de etiquetado.

conjunto de imágenes preprocesadas. A partir de este análisis, se seleccionaron las cinco categorías más representativas, las cuales fueron: charco, cubeta, llanta, tinaco y maceta (véase la Figura 3.6).



Figura 3.6: Ejemplos representativos de cada categoría de criadero en la base de datos El Vergel.

A continuación, se describen algunas características relevantes de los objetos incluidos en cada una de las categorías:

- **Charco:** Los charcos no presentan una forma o tamaño definidos; pueden variar considerablemente en ambas dimensiones. En ocasiones, el agua estancada es turbia. Además, se consideraron como charcos aquellas zonas sobre techos donde se observan marcas de acumulación de agua.
- **Cubeta:** Son objetos de tamaño reducido, sin un color específico predominante. Generalmente presentan forma redonda.
- **Llanta:** Se incluyeron únicamente llantas en condición de desecho, usualmente de color negro. Estas se encontraban comúnmente en exteriores.
- **Tinaco:** Los tinacos, por lo general, tienen forma circular. Se consideraron tanto tinacos con tapa como sin tapa. A pesar de existir variantes en color y diseño (alrededor de tres tipos principales), todos se agruparon dentro de una misma categoría.
- **Maceta:** Esta categoría, al igual que la de cubetas, se compone de objetos pequeños. En la mayoría de los casos, contienen una planta visible en el centro, aunque el tamaño y la forma de las macetas pueden variar significativamente entre sí.

Etiquetado de datos

Una vez preprocesadas las imágenes y definidas las categorías, se procedió con la tarea de etiquetado. La metodología empleada para el etiquetado de los datos se presenta en la Figura 3.7. Dado que los modelos seleccionados para la comparación son de la familia YOLO, se utilizó el software de etiquetado

YOLOLabel v1.2.1 [44]. YOLOLabel es una herramienta que permite etiquetar manualmente imágenes para tareas de detección de objetos, siguiendo el formato requerido por el modelo YOLO (véase la Figura 3.8).

La Tabla 3.1 muestra la correspondencia entre las categorías definidas y su identificador numérico asignado para el etiquetado.

Tabla 3.1: Categorías de etiquetado y sus identificadores

ID	Categoría
0	Charco
1	Cubeta
2	Llanta
3	Tinaco
4	Maceta

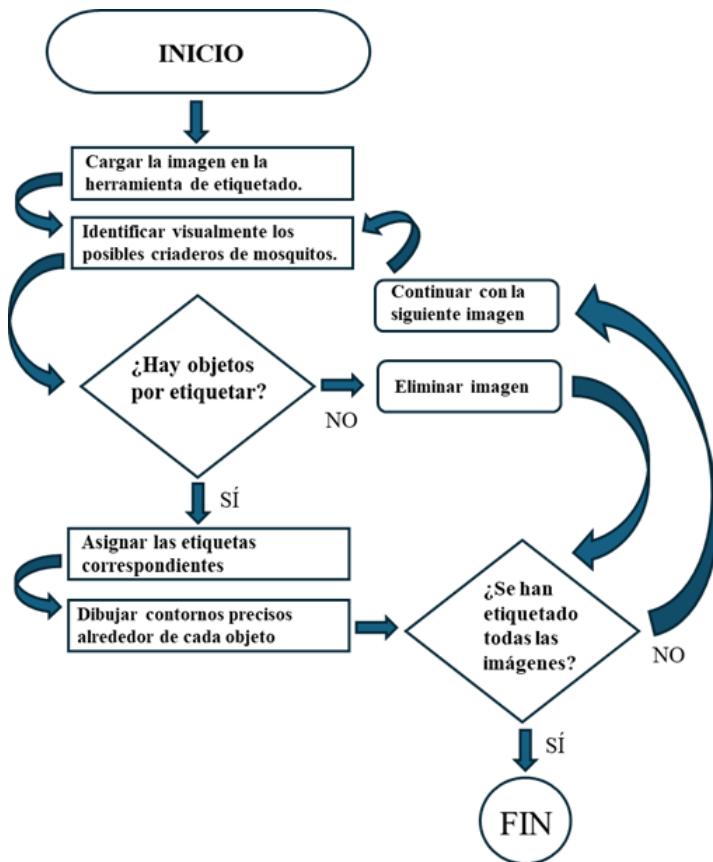


Figura 3.7: Metodología empleada para el etiquetado de imágenes.

El criterio adoptado para el etiquetado de criaderos de mosquitos fue que el objeto correspondiente

debía ser visible al menos en un 30 % de su totalidad; de lo contrario, no se procedía con el etiquetado. Asimismo, si en una imagen no se identificaban criaderos, esta era descartada, ya que no aportaba información útil para el entrenamiento del modelo.

Una vez realizado este preprocesamiento y etiquetado del conjunto de datos El Vergel. Se dejó listo y estandarizado para el entrenamiento de los modelos.



Figura 3.8: Interfaz de la herramienta de etiquetado YOLOLabel. Aunque existen diversas herramientas para el etiquetado de imágenes en formato YOLO, se seleccionó esta debido a la sencillez de su interfaz gráfica. En la parte derecha de la interfaz se muestran las categorías definidas, cada una asociada a un color. El usuario selecciona la categoría de interés y encierra los objetos correspondientes. La herramienta genera un archivo *txt* por cada imagen etiquetada, con el mismo nombre de la imagen, donde se especifica la categoría y las coordenadas normalizadas en el formato (x_c, y_c, w, h) para cada objeto etiquetado.

3.2.2. Datos MBG-V2

Además de contar con los datos proporcionados por el INSP, se realizó una búsqueda de bases de datos relacionadas con la detección de criaderos de mosquitos. Como resultado, se identificó únicamente una base de datos altamente relacionada con el objetivo de nuestro trabajo: la detección de criaderos de mosquitos mediante algoritmos de aprendizaje automático utilizando imágenes aéreas de alta resolución capturadas por drones. Esta base de datos, denominada *Mosquito Breeding Grounds V2* (MBG-V2) [45], representa una segunda versión mejorada de un conjunto de datos previamente existente.

MBG-V2 consiste en un conjunto de videos anotados, especializados en la detección de criaderos del mosquito *Aedes aegypti* de grabaciones aéreas. A diferencia de su versión anterior, esta nueva incluye una

mayor cantidad de videos, así como una clasificación más amplia de tipos de criaderos, lo que incrementa su complejidad y utilidad para el entrenamiento de modelos más robustos. Cabe destacar que todos los videos fueron capturados en escenarios reales y que el conjunto de datos está disponible públicamente para su descarga.

Los videos fueron obtenidos de trece regiones distintas del municipio de Río de Janeiro, Brasil (véase la Figura 3.9).

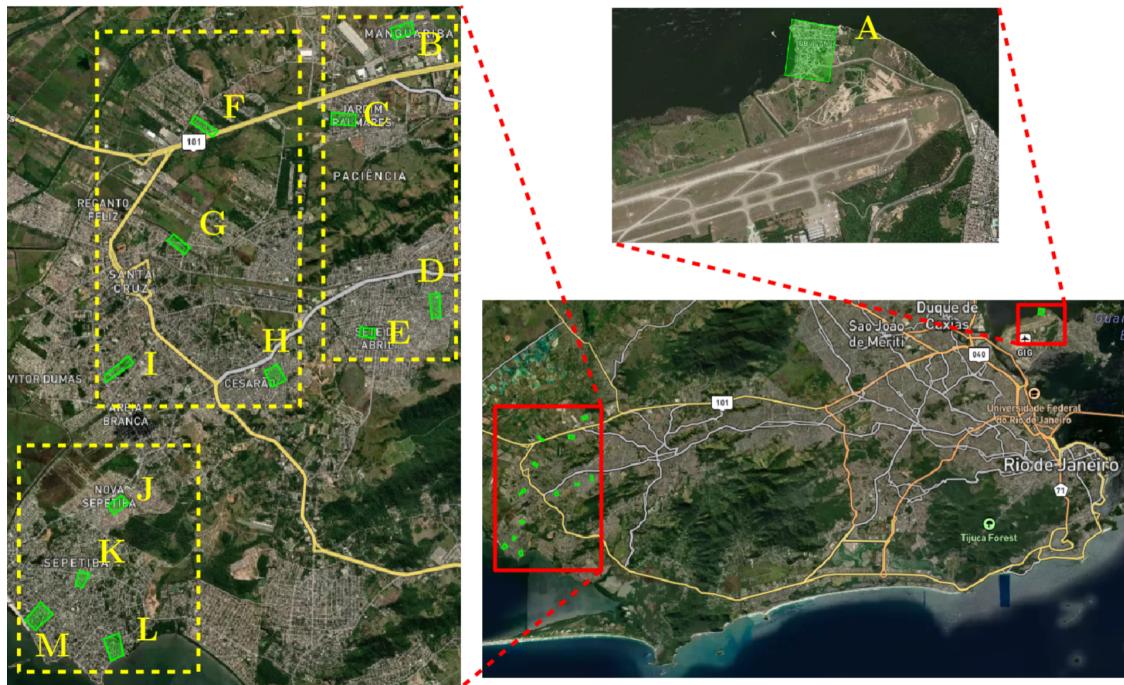


Figura 3.9: Áreas de vuelo en Río de Janeiro para la obtención de datos MBG-V2. Se realizaron vuelos con drones en trece regiones diferentes, identificadas con letras de la A a la M.

Para la captura de los videos, los drones los programaron para realizar vuelos automáticos siguiendo un patrón serpenteante, como se ilustra en la Figura 3.10. Las grabaciones se realizaron a una altitud constante de 40 metros.

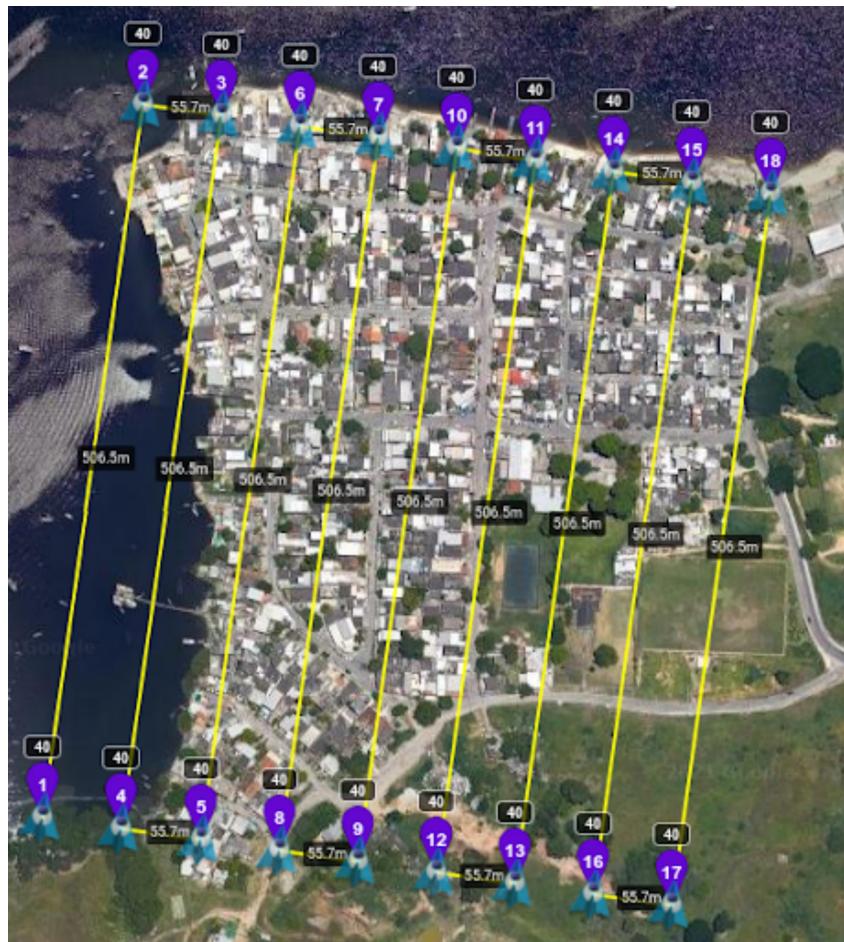


Figura 3.10: Planificación de vuelo para la captura de datos MBG-V2. El dron inicia en el punto 1, continúa hasta el punto 2, realiza un giro, y regresa por el trayecto del punto 3 al punto 4, repitiendo este patrón hasta cubrir toda el área.

Debido a las limitaciones en el tamaño de los archivos y en la extensión de las áreas cubiertas, en cada región se realizaron dos grabaciones complementarias, como se detalla en la Tabla 3.2.

Región	Ubicación	Área (m ²)	Fecha de grabación	Videos	Duración (min:seg)
A	Tubiacanga	239,317	01-06-2019	01	5:27
				02	5:27
B	Paciencia	97,483	28-10-2020	03	3:44
				04	3:29
C	Paciencia	113,623	25-02-2021	05	4:18
				06	4:02
D	Paciencia	90,199	03-02-2021	07	4:07
				08	3:54
E	Paciencia	55,823	03-02-2021	09	2:39
				10	2:28
F	Santa Cruz	81,596	25-02-2021	11	4:25
				12	2:19
G	Santa Cruz	77,431	25-02-2021	13	3:31
				14	3:43
H	Santa Cruz	107,751	03-02-2021	15	4:43
				16	4:29
I	Santa Cruz	100,844	17-03-2021	17	5:37
				18	5:25
J	Sepetiba	75,038	17-03-2021	19	3:08
				20	3:22
K	Sepetiba	53,269	27-01-2021	21	2:35
				22	2:48
L	Sepetiba	131,356	27-01-2021	23	6:03
				24	5:49
M	Sepetiba	149,687	27-01-2021	25	6:15
				26	6:01

Tabla 3.2: Videos por región con duración y fecha de grabación para la captura de datos MBG-V2.

El peso total de toda la base de datos es de 211.6 GB. Cuenta con un total de 26 carpetas (2 grabaciones por las 13 regiones) que contienen un archivo de etiqueta *XML*, el video tomado por el dron, una carpeta de frames extraídas del video y un archivo de excel de la telemetría del dron.

Categorías

El conjunto de datos MBG-V2 contempla doce categorías distintas de objetos considerados como posibles criaderos de mosquitos, tal como se muestra en la Figura 3.11.

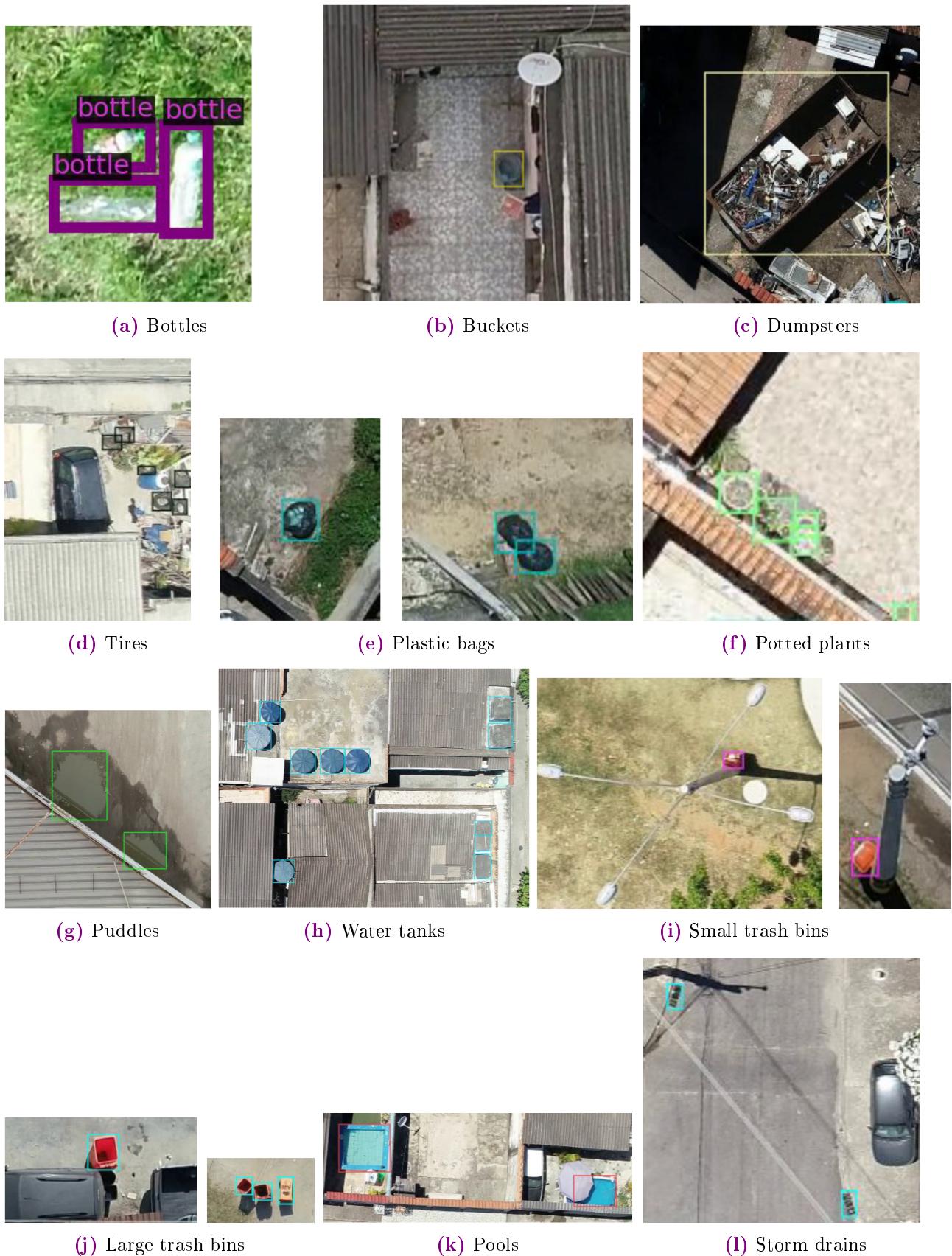


Figura 3.11: Categorías de objetos etiquetados como criaderos potenciales en el conjunto de datos MBG-V2.

Dentro de estas doce categorías se encuentran incluidas las que definimos en el conjunto de datos El Vergel, estableciéndose las siguientes correspondencias: *Tinaco* con *Water tanks*, *Llanta* con *Tires*, *Charco* con *Puddles*, *Maceta* con *Potted plants* y *Cubeta* con *Buckets*.

Etiquetado de datos

El conjunto de datos MBG-V2 incluye anotaciones proporcionadas en formato *XML*, generadas mediante la herramienta *CVAT* (Computer Vision Annotation Tool) para los videos. Este formato sigue una estructura jerárquica definida por el lenguaje *XML*, y se emplea para representar las anotaciones realizadas sobre secuencias de imágenes o videos.

Cada archivo *XML* contiene múltiples elementos denominados `<track>`, los cuales representan individualmente a cada objeto etiquetado. Dentro de cada `<track>` se especifica la categoría del objeto mediante el atributo `label`, así como un identificador único `id`. Los objetos se describen a lo largo de distintos cuadros (*frames*) mediante los elementos `<box>`, los cuales contienen información detallada sobre la ubicación del objeto en cada cuadro, mediante las coordenadas de las esquinas superior izquierda (`xtl`, `ytl`) e inferior derecha (`xbr`, `ybr`). Además, ciertas categorías incluyen atributos adicionales que describen propiedades específicas del objeto, como su material o si se encuentra abierto o cerrado en el caso de los tinacos, etc.

A continuación se presenta un ejemplo:

```

1  <track id="0" label="watertank" source="file">
2    <box frame="243" keyframe="1" outside="0" occluded="0" xtl="1320.41" ytl="0.00" xbr="1446.77" ybr
3      = "46.46" z_order="0">
4      <attribute name="is_open">no</attribute>
5      <attribute name="material">plastic</attribute>
6      </box>
7    <box frame="244" keyframe="1" outside="0" occluded="0" xtl="1320.92" ytl="0.78" xbr="1447.28" ybr
8      = "75.51" z_order="0">
9      <attribute name="is_open">no</attribute>
10     <attribute name="material">plastic</attribute>
11     </box>
12   </track>

```

Código 3.1: Ejemplo de anotación XML de CVAT. Anotación para un objeto de la categoría `watertank`, identificado con `id=0`, visible en los cuadros 243 y 244, en los que se indican sus coordenadas y atributos específicos como el tipo de material (`plastic`) y el estado de la tapa (`is_open=no`).

Dado que la estructura de anotación utilizada en los archivos *XML* proporcionados por *CVAT* difiere del formato requerido por *YOLO*, fue necesario realizar una conversión considerando únicamente las cinco categorías de criaderos de mosquitos definidas en la base de datos de El Vergel. Cada carpeta correspondiente a una región contiene el video de grabación del dron y una subcarpeta con imágenes extraídas a razón de un fotograma por segundo. A partir del primer fotograma correspondiente al primer segundo, los siguientes se obtienen como múltiplos de dicho fotograma. Por tanto, se tomaron en cuenta únicamente estos fotogramas, omitiendo los intermedios.

Durante la conversión, se consideraron las siguientes particularidades de las imágenes:

- Aunque todas las imágenes son de alta resolución, sus dimensiones varían entre regiones.
- El número de fotograma correspondiente al primer segundo varía en cada región; los siguientes fotogramas utilizados son múltiplos de este número.
- El formato de las imágenes no es uniforme entre regiones, pudiendo ser *JPG* o *PNG*.
- Los nombres de archivo de las imágenes no siguen un mismo formato en todas las regiones.

Considerando estas características, se desarrolló el Algoritmo 2 para convertir las anotaciones de *CVAT* al formato requerido por *YOLO*. Este procedimiento se aplicó por carpeta de cada región. En la primera etapa del algoritmo, se definen variables que describen las propiedades de las imágenes, como sus dimensiones, el número de fotograma inicial, el tipo de imagen y variables auxiliares para la estandarización del nombre (por ejemplo, un identificador del video y la región).

El algoritmo realiza primero la estandarización de los nombres de las imágenes. Luego, genera un archivo `txt` vacío por cada imagen. Posteriormente, rellena cada archivo con las etiquetas correspondientes extraídas del archivo *XML*, transformando las coordenadas de las cajas delimitadoras al formato *YOLO*. Finalmente, aquellos archivos `txt` que no contienen ninguna etiqueta, junto con sus imágenes asociadas, son eliminados del conjunto.

Algorithm 2 Conversión de etiquetas XML a formato YOLO con 5 clases

```

1: Definir xml_file, folder_path, id, IMAGE_WIDTH, IMAGE_HEIGHT, n_frame, n_digitos, tipo_img
2: Definir class_map con las 5 clases de interés
3: for all imagen en folder_path do
4:   if imagen válida then
5:     Renombrar imagen agregando id como prefijo
6:   end if
7: end for
8: for all imagen en folder_path do
9:   if imagen válida then
10:    Crear archivo .txt vacío con el mismo nombre
11:   end if
12: end for
13: Leer archivo XML con ElementTree
14: for all track en XML do
15:   Obtener label
16:   if label no está en class_map then continuar
17:   end if
18:   Obtener class_id desde class_map[label]
19:   for all box en track do
20:     Obtener frame
21:     if frame no es múltiplo de n_frame then continuar
22:     end if
23:     Obtener xtl, ytl, xbr, ybr
24:     Calcular centro y dimensiones normalizadas: x_center, y_center, width, height
25:     Escribir línea YOLO en el archivo correspondiente
26:   end for
27: end for
28: Imprimir "Etiquetas YOLO generadas correctamente."
29: for all archivo .txt en folder_path do
30:   if archivo vacío then
31:     Eliminar archivo .txt y su imagen correspondiente
32:   end if
33: end for

```

Objetos por región

De acuerdo a las etiquetas en los archivos XML, el número de criaderos de mosquitos por región se observa en la Tabla 3.3.

Región	bottle	bucket	pool	puddle	tire	water tank	dumpster	large trash bin	plastic bag	potted plant	small trash bin	storm drain
A	8	35	13	0	74	681	1	8	1	210	3	75
B	0	12	36	0	17	438	0	0	3	107	0	50
C	0	17	55	0	3	186	0	4	3	37	1	50
D	6	52	95	0	25	359	0	1	1	70	2	94
E	0	38	36	1	7	270	0	1	5	29	0	30
F	0	10	28	2	85	95	0	4	2	104	0	49
G	38	23	30	1	9	155	0	1	7	82	3	28
H	1	48	99	0	86	357	2	1	4	52	4	78
I	4	75	40	5	29	408	1	6	14	44	14	55
J	11	73	66	29	106	183	0	1	15	62	8	41
K	0	52	59	0	36	241	2	9	20	68	5	120
L	1	19	105	1	15	329	0	0	9	211	0	115
M	0	2	101	0	17	582	0	0	6	127	0	162
Total	69	456	763	39	509	4284	6	36	90	1203	40	947

Tabla 3.3: Conteo de criaderos por región y tipo en la base de datos MBG-V2, incluyendo el total general por tipo.

De acuerdo a la tabla anterior se obtienen un total de 8442 criaderos de mosquitos auténticos de los cuales las categorías con menor objetos son *bottle*, *puddle*, *dumpster*, *large trash bin*, *plastic bag* y *small trash bin*. Por otro lado, las categorías con mayor cantidad de objetos son *bucket*, *pool*, *tire*, *water tank*, *potted plant* y *storm drain*.