

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA DE SISTEMAS**

**UNIDAD DE TITULACIÓN**

**Detección de incidentes automovilísticos usando técnicas de  
aprendizaje de máquina**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE  
MAGÍSTER EN COMPUTACIÓN**

**Mario Gerardo Moreno Pallares**

**mario.morneo01@epn.edu.ec**

**Director: Sang Guun Yoo, Ph.D.**

**sang.yoo@epn.edu.ec**

**Co-Director: Wilbert Geovanny Aguilar Castillo, Ph.D.**

**wilbert.aguilar@epn.edu.ec**

## **APROBACIÓN DEL DIRECTOR**

Como director del trabajo de titulación **DETECCIÓN DE INCIDENTES AUTOMOVILÍSTICOS USANDO TÉCNICAS DE APRENDIZAJE DE MÁQUINA**, del estudiante de la Maestría en Computación, habiendo supervisado la finalización de este trabajo y realizado las correcciones correspondientes, apruebo la redacción final del documento escrito para continuar con el correspondiente procedimiento para apoyar la defensa oral.

---

**Sang Guun Yoo, Ph.D**

**DIRECTOR**

## **APROBACIÓN DEL CO-DIRECTOR**

Como co-director del trabajo de titulación DETECCIÓN DE INCIDENTES AUTOMOVILÍSTICOS USANDO TÉCNICAS DE APRENDIZAJE DE MÁQUINA, del estudiante de la Maestría en Computación, habiendo supervisado la finalización de este trabajo y realizado las correcciones correspondientes, apruebo la redacción final del documento escrito para continuar con el correspondiente procedimiento para apoyar la defensa oral.

---

**Wilbert Geovanny Aguilar Castillo, Ph.D**

**CO-DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

Yo, Mario Gerardo Moreno Pallares, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que incluyen en este documento.

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



---

**Mario Gerardo Moreno Pallares**

## **DEDICATORIA**

Esta tesis se la dedico a DIOS quien inspira y guía mi vida. A mis padres quienes me dan su alegría, apoyo y consejos. A mis hermanos y sobrinas por ser fuente de alegría. A mi director y co-director por su ayuda incondicional. Y a todas las personas que de una u otra forma contribuyeron a este trabajo.

## **AGRADECIMIENTOS**

Al finalizar un trabajo arduo y lleno dificultades como el desarrollo de una tesis. Debo agradecer de manera especial a DIOS por todo.

De manera especial a mis padres Rodrigo y Lucila por ser el apoyo moral fundamental en el transcurso de mi vida. A mis hermanos Angelita y Rodrigo que me brindan sus consejos para llevar un camino correcto. A mis sobrinas Carito, Cami, Tina y Ale por ser un motor de alegría para seguir mi camino.

De igual manera mis más sinceros agradecimientos al Ph.D. Sang Guun Yoo y Ph.D. Wilbert Aguilar, directores del presente trabajo de investigación, por su guía científico-técnica, son personas apasionadas por su trabajo, con mucha vocación y servicio impulsando a que las personas alcancen sus metas. Gratitud por el incondicional apoyo brindado por el Ph.D. Marco Molina para el realce del presente trabajo.

También, agradecer a todas las autoridades de la Facultad de Sistemas y de toda la Escuela Politécnica Nacional por brindarme su apoyo.

# CONTENIDO

ÍNDICE DE FIGURAS.....	9
ÍNDICE DE TABLAS .....	11
ÍNDICE DE ECUACIONES .....	12
ÍNDICE DE ALGORITMOS.....	13
ÍNDICE DE DIAGRAMAS.....	14
RESUMEN.....	15
ABSTRACT .....	16
<b>1. INTRODUCCIÓN .....</b>	<b>17</b>
<b>1.1. ANTECEDENTES .....</b>	<b>17</b>
<b>1.2. ENFOQUE DE LA INVESTIGACIÓN Y OBJETIVOS.....</b>	<b>18</b>
1.2.1. OBJETIVO GENERAL.....	21
1.2.2. OBJETIVOS ESPECÍFICOS.....	21
<b>1.3. VALOR DE LA INVESTIGACIÓN.....</b>	<b>21</b>
<b>2. MARCO TEÓRICO .....</b>	<b>24</b>
<b>2.1. INCIDENTES DE TRÁNSITO Y EVENTOS INESPERADOS.....</b>	<b>24</b>
<b>2.2. APRENDIZAJE DE MÁQUINA.....</b>	<b>26</b>
2.2.1. CLASIFICACIÓN CON K – VECINOS MÁS CERCANOS (K-NEAREST NEIGHBORS O KNN) .....	29
2.2.2. CLASIFICACIÓN CON REGRESIÓN LOGÍSTICA.....	31
2.2.3. CLASIFICACIÓN CON MÁQUINA DE SOPORTE VECTORIAL.....	32
2.2.3.1. CLASIFICADOR DE MARGEN MÁXIMO .....	35
2.2.3.2. CLASIFICADOR DE MARGEN SUAVE.....	36
<b>2.3. INDICADORES DE RENDIMIENTO DEL MODELO DE CLASIFICACIÓN.....</b>	<b>40</b>

<b>2.4. VALIDACIÓN CRUZADA K-PLIEGUES .....</b>	<b>43</b>
<b>2.5. DETECCIÓN DE OBJETOS.....</b>	<b>43</b>
2.5.1. YOLO v3.....	45
2.5.2. SSD MOBILENET.....	47
<b>2.6. SEGUIMIENTO DE OBJETOS.....</b>	<b>50</b>
2.6.1. FILTRO DE CORRELACIÓN KERNELIZADO (KCF) .....	51
2.6.2. SUMA DE SALIDA MÍNIMA DE ERROR CUADRÁTICO (MOSSE) .....	53
2.6.3. MEDIANFLOW .....	54
2.6.4. CENTROIDE .....	56
2.6.5. ESTIMACIÓN DE ESCALA.....	57
2.6.6. DESBALANCE DE DATOS.....	58
2.6.7. MÉTODO DE BOOTSTRAP .....	59
<b>3. REVISIÓN DE LITERATURA.....</b>	<b>61</b>
<b>3.1. METODOLOGÍA .....</b>	<b>61</b>
<b>3.2. PREGUNTAS DE INVESTIGACIÓN .....</b>	<b>61</b>
<b>3.3. MÉTODO DE REVISIÓN .....</b>	<b>62</b>
3.3.1. FUENTES Y ESTRATEGIAS DE INVESTIGACIÓN.....	62
3.3.2. CADENAS DE BÚSQUEDA.....	62
3.3.3. CRITERIO DE SELECCIÓN DE ESTUDIOS.....	62
3.3.4. EXTRACCIÓN DE INFORMACIÓN .....	63
<b>3.4. ESTUDIOS INCLUIDOS Y EXCLUIDOS .....</b>	<b>64</b>
<b>3.5. RESULTADOS OBTENIDOS.....</b>	<b>64</b>
<b>3.6. DISCUSIÓN .....</b>	<b>85</b>
<b>3.7. CONCLUSIÓN DEL ESTUDIO DE LITERATURA .....</b>	<b>87</b>
<b>4. METODOLOGÍA .....</b>	<b>88</b>
<b>4.1. DISEÑO DE LA INVESTIGACIÓN PARA LA ACCIÓN.....</b>	<b>89</b>
4.1.1. ACTUACIÓN .....	90
4.1.1.1. Creación del Conjunto de Datos .....	92
4.1.1.2. Detección, seguimiento y estimación de velocidad .....	95
4.1.1.3. Entrenamiento y evaluación de modelos .....	98
<b>4.2. LIMITACIONES Y PROBLEMAS POTENCIALES .....</b>	<b>102</b>

<b>5. EXPERIMENTACIÓN.....</b>	<b>104</b>
<b>5.1. EXPERIMENTOS PILOTOS.....</b>	<b>104</b>
5.1.1. CREACIÓN DE MODELOS SIN BALANCEO DE DATOS.....	104
5.1.2. CREACIÓN DE MODELOS CON BALANCEO DE DATOS.....	109
<b>6. CONCLUSIONES .....</b>	<b>117</b>
<b>6.1. OBJETIVOS DE INVESTIGACIÓN: RESUMEN DE LOS HALLAZGOS Y CONCLUSIONES .....</b>	<b>117</b>
6.1.1. DETECCIÓN DE INCIDENTES AUTOMOVILÍSTICOS CON VIDEOS DE TRÁFICO Y APRENDIZAJE DE AUTOMÁTICO.....	117
6.1.2. EXPERIMENTACIÓN CON EL CONJUNTO DE DATOS PARA LA CREACIÓN DE LOS MODELOS DE CLASIFICACIÓN .....	118
6.1.3. MODELO CON MEJOR RENDIMIENTO DE CLASIFICACIÓN EN LA DETECCIÓN DE INCIDENTES AUTOMOVILÍSTICOS.....	118
<b>6.2. CONTRIBUCIONES CIENTÍFICAS .....</b>	<b>118</b>
<b>6.3. RECOMENDACIONES Y TRABAJOS FUTUROS .....</b>	<b>118</b>
<b>7. REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>120</b>
<b>8. ANEXOS.....</b>	<b>130</b>
<b>ANEXO 1 - PREGUNTAS DE INVESTIGACIÓN, CADENAS DE BÚSQUEDA Y CRITERIOS DE INCLUSIÓN Y EXCLUSIÓN DE ARTÍCULOS CIENTÍFICOS .....</b>	<b>130</b>
<b>ANEXO 2 - MATRICES DE CONFUSIÓN CON DESBALANCE DE DATOS.....</b>	<b>168</b>
<b>ANEXO 3 - MATRICES DE CONFUSIÓN CON BALANCEO DE DATOS.....</b>	<b>193</b>

## Índice de figuras

Figura 1: Ciclo en etapas de incidentes de tránsito propuesto .....	23
Figura 2: Etapas de incidentes de tránsito. ....	24
Figura 3: Aprendizaje supervisado .....	29
Figura 4: Clasificación con hiperplanos.....	34
Figura 5: Clasificación con hiperplanos.....	35
Figura 6: Hiperplano de Margen Máximo (MMH) .....	36
Figura 7: Hiperplano con vectores de soporte de margen suave.....	38
Figura 8: Clasificación de kernel polinomial .....	39
Figura 9: Clasificación SVM con kernel RBF Gaussiano .....	40
Figura 10: Matriz de confusión de dos clases .....	41
Figura 11: Validación cruzada proceso .....	43
Figura 12: Arquitectura Darknet-53 (YOLO v3) .....	47
Figura 13: Arquitectura SSD .....	48
Figura 14: Arquitectura MobileNet.....	49
Figura 15: Arquitectura SSD-MobileNet .....	50
Figura 16: Artículos seleccionados .....	64
Figura 17: Clasificadores utilizados en investigaciones escogidas.....	65
Figura 18: Tipos de conjuntos de datos .....	80
Figura 19: Conjuntos de datos públicos disponibles.....	80
Figura 20: Ciclo Action Research.....	89
Figura 21: Diagrama de selección de videos.....	93
Figura 22: Imagen con etiquetas.....	94
Figura 23: Imagen sin etiquetas .....	95
Figura 24: Diagrama cálculo de velocidades estimadas.....	98
Figura 25: Comportamiento de la exactitud en los modelos creados .....	105
Figura 26: Comportamiento de la Calificación AUC ROC.....	106
Figura 27: Comportamiento de la Precisión en los Modelos.....	107
Figura 28: Comportamiento de la Sensibilidad en los Modelos .....	108
Figura 29: Matriz de Confusión Modelo KNN .....	108
Figura 30: Comportamiento de AUC ROC del Modelo KNN.....	109
Figura 31: Exactitud de los modelos con balanceo de datos.....	110
Figura 32: Métrica AUC ROC de los modelos con datos balanceados.....	111
Figura 33: Sensibilidad de los modelos creados con balanceo de datos.....	112
Figura 34: Precisión de los modelos creados.....	113
Figura 35: Calificación F1 de los modelos creados con datos balanceados .....	114

Figura 36: Matriz de confusión primer modelo con mejor rendimiento.....	114
Figura 37: Curva ROC AUC primer modelo con mejor rendimiento .....	115
Figura 38: Matriz de confusión segundo modelo con mejor rendimiento .....	115
Figura 39: Curva AUC ROC segundo modelo con mejor rendimiento .....	116

## Índice de tablas

Tabla 1: SITs y CCT en etapas a seguir en un incidente .....	25
Tabla 2: Criterios de forma y fondo .....	63
Tabla 3: Estudios de investigación incluidos y excluidos.....	64
Tabla 4: Exactitud de clasificación del incidente .....	66
Tabla 5: Técnicas de detección y seguimiento de vehículos .....	73
Tabla 6: Conjuntos de datos hallados .....	80
Tabla 7: Métricas de desempeño utilizado en trabajos previos .....	84
Tabla 8: Descripción del conjunto de datos.....	94
Tabla 9: Descripción del nuevo conjunto de datos .....	95
Tabla 10: Número de tuplas en los conjuntos de datos.....	101
Tabla 11: Número de tuplas del conjunto de datos para entrenamiento.....	101
Tabla 12: Número de tuplas del conjunto de datos de evaluación.....	102
Tabla 13: Exactitud Score de los Modelos Creados .....	104
Tabla 14: AUC ROC Score de los Modelos Creados .....	105
Tabla 15: Precisión de los Modelos Creados .....	106
Tabla 16: Sensibilidad de los Modelos Creados.....	107
Tabla 17: Exactitud de los modelos creados con balanceo de datos .....	109
Tabla 18: AUC ROC de cada modelo creado.....	110
Tabla 19: Valores de sensibilidad de cada modelo creado.....	111
Tabla 20: Precisión de los modelos creados con balanceo de datos .....	112
Tabla 21: Calificación F1 de los modelos creados .....	113

## Índice de ecuaciones

Ecuación 1. Distancia euclidiana.....	30
Ecuación 2. Regresión Logística.....	31
Ecuación 3. Función sigmoidea .....	31
Ecuación 4. Predicción de la regresión logística .....	32
Ecuación 5. Costo de la ecuación .....	32
Ecuación 6. Pérdida logarítmica.....	32
Ecuación 7. Derivadas parciales de la función de costo.....	32
Ecuación 8. Hiperplano .....	33
Ecuación 9. Solución básica de problemas de clasificación lineal.....	34
Ecuación 10. Clasificador de Margen Máximo .....	36
Ecuación 11. Distancia perpendicular de un punto en el hiperplano .....	37
Ecuación 12. Vectores de soporte en el hiperplano .....	37
Ecuación 13. Hiperplano de soporte computacional.....	38
Ecuación 14. Métrica de Exactitud del modelo.....	41
Ecuación 15. Métrica de Sensibilidad.....	42
Ecuación 16. Métrica de Especificidad.....	42
Ecuación 17. Medida de Precisión .....	42
Ecuación 18. F1 score .....	42
Ecuación 19. Representación dinámica del movimiento.....	56
Ecuación 20. Representación de los centroides.....	57

## Índice de algoritmos

Algoritmo 1: Clasificación con kNN. ....	31
Algoritmo 2: Método bootstrap .....	60
Algoritmo 3: Cálculo de velocidades estimadas .....	97

## **Índice de diagramas**

Diagrama 1: Etapas generales de la detección de incidentes automovilísticos .....	90
---	----

## RESUMEN

El aumento de automóviles en las carreteras ha causado varios tipos de problemas entre los más graves están los incidentes de tránsito, producidos frecuentemente por el exceso de velocidad. Estos influyen directamente con el riesgo de colisión de impacto, así como la probabilidad de muerte y lesiones graves.

Cada año a nivel mundial se registran alrededor de 1.35 millones de fallecidos a causa de los siniestros de tránsito y alrededor de 50 millones sufren lesiones a causa de estos; originando un costo anual del 3% del PIB al gobierno de cada país.

En los últimos años la Inteligencia Artificial ha pasado a formar parte fundamental de los sistemas autónomos en la rama del transporte porque ayuda al control de los mismos con mínima intervención humana, como en el caso de la videovigilancia en las vías.

Es importante desarrollar herramientas de automatización que ayuden a los operadores de cámaras de seguridad a detectar incidentes de tránsito, debido a que por cada minuto ganado en detectar tempranamente los incidentes de tránsito, se puede ganar hasta cinco minutos en la fase de recuperación de la vía.

El presente trabajo ha creado y comparado 100 modelos de clasificación como kNN, regresión logística y SVM. Se creó un conjunto de datos a partir de videos de UCF y CADP el cual sirvió para calcular las velocidades estimadas mediante algoritmos de detección y seguimiento. Estas velocidades sirvieron para el entrenamiento y evaluación de los modelos, obteniendo un modelo con exactitud de 0.95 y AUC ROC de 0.96.

**Palabras clave:** aprendizaje de automático, aprendizaje supervisado, detección de objetos, seguimiento de objetos, balanceo de datos.

## **Abstract**

The increase of automobiles on the roads has caused several types of problems, among the most serious are traffic incidents, frequently caused by speeding. These directly influence the risk of an impact collision, as well as the probability of death and serious injury.

Every year around 1.35 million deaths are registered worldwide due to traffic accidents and around 50 million get injuries as a result of those accidents; originating an annual cost of 3% of GDP to the government of each country.

In recent years, Artificial Intelligence has become a fundamental part of autonomous systems in the field of transport because it helps control them with minimal human intervention, as in the case of video surveillance on the roads.

It is important to develop automation tools that help security camera operators to detect traffic incidents, because for every minute saved in early detection of traffic incidents, up to five minutes can be gained in the road recovery phase.

The present work has created and compared 100 classification models such as kNN, logistic regression and SVM. A dataset was created from videos from UCF and CADP which was used to calculate the estimated speeds using detection and tracking algorithms. These speeds were used for the training and evaluation of the models, obtaining a model with an accuracy of 0.95 and AUC ROC of 0.96.

**Keywords:** machine learning, supervised learning, object detection, object tracking, data balance.

# CAPÍTULO 1

## 1. INTRODUCCIÓN

### 1.1. ANTECEDENTES

El crecimiento de la cantidad de automóviles ha causado que aparezcan diversos tipos de problemas, uno de los más graves es el de los incidentes de tránsito [1], [2], siendo el más común por exceso de velocidad.

Un factor importante que hacen que existan los incidentes de tránsito es la velocidad con la que circula un vehículo ya que influye directamente al riesgo de colisión, así como la probabilidad de muerte y sufrir lesiones graves [3], [4], por ejemplo los ocupantes de un automóvil que circulan a velocidades superiores a 65 km/h tienen un riesgo mayor del 85% de morir en una colisión entre automóviles [5].

A nivel mundial cada año mueren alrededor de 1.35 millones de personas por consecuencia de siniestros viales y se estima que otros 50 millones sufren lesiones graves, lo que genera a los gobiernos un costo anual de aproximadamente del 3% de su Producto Interno Bruto (PIB) [6]. La segunda causa de muertes en jóvenes de 15 a 29 años son las lesiones de tránsito, ésta causa la muerte prematura de 117911 personas en Latinoamérica y el Caribe cada año. Además, se pronosticó que para el año 2020, se tendrá alrededor de 1.9 millones de personas que morirán a causa de accidentes de tránsito [7], y el 34% de las muertes estimadas serán causadas por vehículos de cuatro ruedas [6], [8].

Según cifras oficiales de la Secretaria General Subregional del Banco Interamericano de Desarrollo (BID), entre 2005 y 2014 el parque automotor de los países integrantes de la Comunidad Andina de Naciones (CAN) como lo son: Colombia, Ecuador, Perú y Bolivia, tuvieron un incremento anual del 11.7%, lo que representa alrededor de 19.19 millones de unidades. El incremento lo encabeza Colombia con el 12.7%, seguido de Bolivia con el 11.8%, Perú con 11.7% y Ecuador con 8.1%, a la vez otro informe de la CAN reveló que un total de 13479 personas murieron y 141175 resultaron heridas en 347018 accidentes de tránsito suscitados.

Mientras que en la región Andina hubo 127 fallecidos por cada millón de habitantes a causa de incidentes de tránsito y esta cifra está en incremento. Por otro lado, en la Unión Europea (EU), decayeron el número de fallecidos por incidentes de tránsito, llegando así a 50 fallecidos por cada millón de habitantes entre 2015 y 2016 gracias a diferentes factores siendo uno de los principales el despliegue de Sistemas Inteligentes de Transporte (SIT) [9].

En Ecuador, hubo un total de 24595 siniestros de transporte en 2019, que dejaron 19999 personas lesionadas y 2180 fallecidos [10]. Los autos particulares y motocicletas son los tipos de vehículos que más se presentan en los siniestros de tránsito, ocupando el 53% [11], de ello 4006 siniestros presentan como causa probable el exceso de velocidad [10].

En los últimos años se ha dado grandes avances en las tecnologías autónomas, en las cuales la Inteligencia Artificial (IA) es parte fundamental para el control de sistemas de forma autónoma con la mínima intervención humana.

Con el paso del tiempo los estudios del transporte han evolucionado en Sistemas Inteligentes de Transporte (SITs), con el fin de ayudar a tener soluciones efectivas como: la detección y seguimiento de objetos que interactúan en las vías [12], reconocimiento o clasificación de patrones que tienen una amplia cantidad de problemas, que requieren discriminar un objeto o colocarlo en su clase correcta, en las que figura la detección automática de incidentes [13].

En América del Sur, la mayoría de autopistas de alta velocidad tienen una cobertura casi total por Circuito Cerrado de Televisión (CCTV), por lo que los videos pueden ser posteriormente utilizados como una herramienta valiosa por ejemplo para la creación de nuevos algoritmos de inteligencia artificial y análisis de datos [14].

## **1.2. ENFOQUE DE LA INVESTIGACIÓN Y OBJETIVOS**

El presente trabajo se enfocó en la detección de incidentes de tránsito específicamente de colisiones de automóviles captados en videos de cámaras estáticas, mediante herramientas y algoritmos de detección de objetos como YOLO v3 y SSD MobileNet.

En cuanto a los rastreadores de los automóviles detectados se utilizaron el Filtro de Correlación Kernelizado (KCF), estimación de escala, Suma de Salida Mínima de Error Cuadrático (MOSSE), MedianFlow y centroide. Luego, para obtener las velocidades estimadas y realizar una clasificación binaria con algoritmos de aprendizaje supervisado como: k Vecinos más Cercanos (KNN), regresión logística y Máquina de Soporte Vectorial (SVM).

Para la evaluación del rendimiento de los modelos creados se tomó en cuenta la métrica de la exactitud de clasificación, sensibilidad, Área bajo la Curva ROC (AUC ROC). Los videos utilizados fueron tomados de cámaras estáticas y recogidos de las siguientes bases de datos: Anomaly Detection Dataset (UCF) y Car Accidents Detection and Prediction (CADP). También se utilizaron videos propios, filmados en vías de la ciudad de Riobamba.

Como se mencionó anteriormente, los SITs consisten en aplicaciones avanzadas de Tecnologías de la Información y Comunicación (TICs) aplicados en los sistemas de transporte que permiten atender de forma eficiente las distintas necesidades de usuarios, oferta disponible de servicios y mecanismos de gestión del tráfico. Estos permiten que los sistemas de transporte se vuelvan más sencillos de usar, sean más confiables y a la vez más seguros [15]; en efecto uno de los objetivos del BID es la aplicación de algoritmos de inteligencia artificial, con diferentes niveles de intervención humana en los sistemas de transporte [16].

Los incidentes de tránsito pueden ser: choques, vehículos averiados, derrames de carga, mantenimiento de vías y otros eventos inusuales, ya que son eventos que no ocurren con frecuencia y no dejan tener un flujo normal de los vehículos. Además, la congestión vehicular está vinculada con los incidentes de tránsito [14], [17], [18].

Al producirse incidentes de tránsito los SITs y los Centros de Control de Tránsito (CCT) toman un papel sumamente importante. Por ejemplo, al capturar imágenes por medio de CCTV algunos softwares detectan la anomalía [14], lo que conlleva a la actuación de la Gestión Integral de Incidentes de Tránsito (TIM), que tiene por objetivo principal dar respuesta de forma segura y rápida a los incidentes y otros eventos no planeados de tránsito que se producen en las carreteras para evitar nuevos incidentes y reanudar las condiciones de tránsito a la normalidad para ello requiere el despliegue de forma conjunta, ordenada, planificada de acciones y recursos [14], [19].

En este sentido, para la detección de incidentes de tránsito se están probando con algoritmos y técnicas de aprendizaje de máquina, éstas han existido durante décadas en aplicaciones especializadas. Por ejemplo, el aprendizaje supervisado toma decisiones mediante los datos proporcionados en el entrenamiento por medio de las etiquetas, una tarea típica es la clasificación. Algunos de los algoritmos más importantes de este aprendizaje son: K Nearest Neighbors (KNN), regresión logística, Máquinas de Soporte Vectorial (SVMs) [12], [20].

Como se mencionó anteriormente una parte importante es la detección de los objetos que intervienen en el tránsito, en este caso específico los automóviles. Hoy en día este problema se está resolviendo con modelos de aprendizaje profundo, basándose en configuraciones especiales de una red neuronal convolucional básica [21]. Uno de los modelos más utilizados en la computación visual en ambientes ligeros es SSD MobileNet por su compensación entre latencia y precisión. Otro modelo convolucional ligero es Solo

Mira Una Vez (YOLO) al cual lo describen como extremadamente rápido y preciso [22], [23], [24].

Los algoritmos de detección de objetos con redes convolucionales requieren del consumo de más recursos computacionales, por ello los algoritmos de seguimiento han tomado gran popularidad en la videovigilancia, ya que son rápidos y conservan la identidad. Además, tienen buenos resultados cuando la detección falla [25]. En este caso son de gran ayuda, porque se logra el cálculo de las velocidades estimadas de cada automóvil detectado.

Estos tienen algunos enfoques como: seguimiento bidireccional (adelante y atrás) que lo hace más confiable con movimiento constante como lo es MedianFlow, resistencia a la escala mediante filtro de correlación (MOSSE), extrayendo funciones de histograma (KCF) [26], manejando variaciones de escala en secuencia de imágenes complejas [27], teniendo mayor robustez con respecto a la oclusión y desplazamiento del contorno de las imágenes mediante centroides [28].

Una característica importante en cuanto a los sistemas de transporte es que estos operan en un dominio que no es simple, por ejemplo el modelar interacciones de factores, representar generalizaciones para luego resolver una tarea en particular representa una tarea desafiante ya que no son problemas triviales; en cuanto a los incidentes que se presentan en las vías los datos son muy escasos y el encontrar estructuras apropiadas o enfoques de modelado podría ayudar a resolver una deficiencia en el área de la detección de incidentes de tránsito [12].

La detección de incidentes automovilísticos en el estudio del transporte no es un tema nuevo de investigación, pero es de vital importancia ya que es un problema social. Algunas áreas de investigación clave en el transporte son: control de señales de tráfico, control autónomo de vehículos y el control del flujo del tráfico, ya que con ello se estaría cumpliendo con uno de los objetivos fundamentales de los SITs que es el de proporcionar sistemas de transporte seguros, efectivos y confiables para las personas [12], [29].

De hecho [29] en su estudio menciona un trabajo relacionado con incidentes de tránsito el cual modela un efecto de características específicas del incidente ocasionado, usa una Red Recurrente Memoria a Corto PLazo (LSTM) para modelar el flujo del tráfico y hacer predicciones, trabaja con datos extraídos de sensores de bucle facilitados por CalTrans , el Departamento de Transporte de la ciudad de Los Ángeles (Estados Unidos), la Patrulla de Carreteras de California y el Metro de Los Ángeles.

De acuerdo con los estudios de [30] y [31] en donde se tienen las limitaciones de las aplicaciones de detección de accidentes de tránsito mediante un teléfono inteligente por lo que expresan que un sistema de Detección Automática de Incidentes (DAI componente principal de los SITs que usa CCTV) bien diseñado es capaz de detectar incidentes y notificar automáticamente a las autoridades y así evitar falsas alarmas. Por los antecedentes mencionados junto con la posibilidad de encontrar nuevos métodos más para la detección de incidentes automovilísticos, constituyen razones suficientes para dar paso al planteamiento de una hipótesis:

*Es posible crear un modelo de detección de incidentes automovilísticos en videos de tráfico terrestre con el uso de técnicas de aprendizaje de automático, detección y seguimiento de objetos que se encuentran en la revisión de literatura.*

### **1.2.1. OBJETIVO GENERAL**

Detectar incidentes automovilísticos usando técnicas de aprendizaje de máquina sobre un conjunto de datos de videos de tráfico vehicular terrestre.

### **1.2.2. OBJETIVOS ESPECÍFICOS**

- Entender el estado del arte en la detección de incidentes automovilísticos mediante una revisión de literatura.
- Desarrollar un modelo de detección de incidentes automovilísticos usando algoritmos de detección y de seguimiento de objetos encontrados en el estudio del estado del arte.
- Evaluar el rendimiento del modelo de detección de incidentes automovilísticos.

## **1.3. VALOR DE LA INVESTIGACIÓN**

Aproximadamente 1.35 millones de personas mueren cada año a causa de accidentes de tránsito, son la principal causa de defunción de niños y jóvenes de 5 a 29 años de edad. En los países de ingresos bajos y medianos son donde se producen más del 90% de defunciones a causa de estos incidentes y en la mayoría de los países les cuesta el 3% de su PIB [32].

En la Unidad Europea (UE) la tasa de mortalidad por accidentes de tránsito ha bajado gracias a los despliegues de los SITs. Sin embargo la gran mayoría de sistemas de video vigilancia de tráfico se basan en la observación humana, por lo que es difícil el monitoreo de una gran cantidad de escenas de cámara de video al mismo tiempo y reconocer accidentes de tránsito [33].

Se debe tomar en cuenta que el período de recuperación de la vía puede ser de cuatro a cinco veces mayor que la duración del mismo incidente, pero puede ganarse hasta cinco minutos en la fase de recuperación para que el tránsito vuelva a circular con normalidad por cada minuto que se gana en detectar, verificar y terminar con los incidentes [14].

La detección de incidentes automovilísticos no es un tema nuevo de investigación, pero estos operan en un dominio que no es fácil de modelar y se torna un problema desafiante que involucra tanto aprendizaje automático como la computación visual y métodos de clasificación que ayuden a detectar si existe o no un incidente de tránsito mediante videos de tráfico.

En el presente trabajo se ha experimentado con métodos de aprendizaje de automático supervisado, los cuales ayudan a la clasificación en la detección de incidentes automovilísticos, por medio de las velocidades estimadas de los vehículos detectados con algunas técnicas de rastreo en los videos recogidos. Estos se encuentran en un ambiente controlado y con una ventana de tiempo específica de un segundo.

Los modelos que se crearon podrían servir para integrar un sistema DAI mediante el funcionamiento de aprendizaje por lotes el cual aprende por medio de la actualización de datos y el entrenamiento con la frecuencia que se necesite. Además, ayudaría a cerrar el ciclo entre las etapas que se siguen al presentarse un incidente de tránsito, en los que están involucrados los SITs y CCT, como se muestra en la Figura 1.

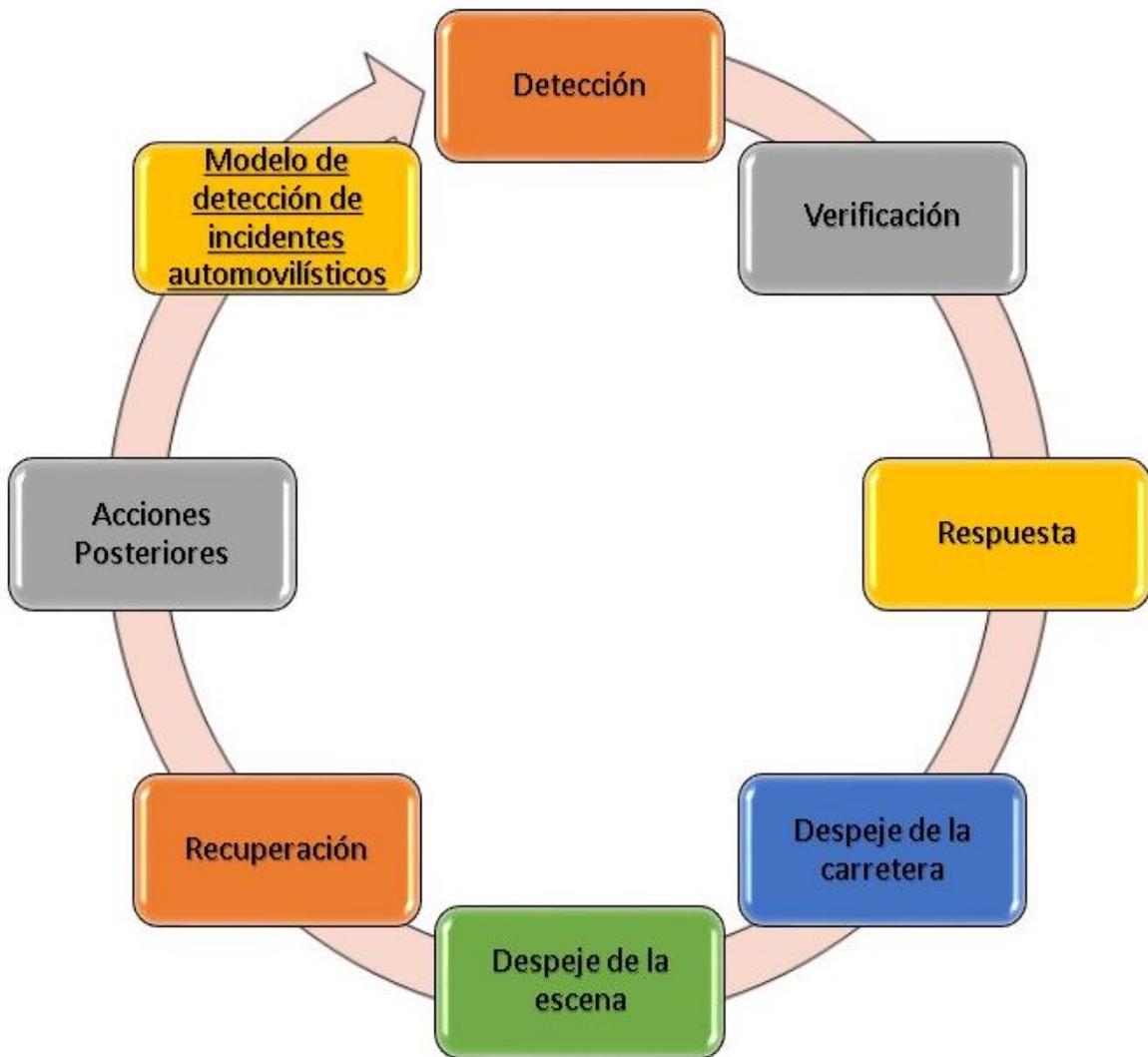


Figura 1: Ciclo en etapas de incidentes de tránsito propuesto

## CAPÍTULO 2

### 2. MARCO TEÓRICO

#### 2.1. INCIDENTES DE TRÁNSITO Y EVENTOS INESPERADOS

La gestión de incidentes de tránsito se da por medio de un ciclo de fases, que empieza con la notificación de forma inmediata de los posibles problemas que podrían surgir por el incidente provocado, con el fin de alertar a los conductores y prevenir accidentes. Los principales objetivos de la notificación del incidente y su gestión son: prevenir al mínimo el riesgo y consecuencias que causan los incidentes, gestionar y resolver los incidentes de una forma eficaz, segura y rápida. Los incidentes y su gestión tienen diferentes etapas en las cuales se evalúa la magnitud e identifican los recursos y acciones necesarias para restaurar las operaciones normales y brindar la respuesta adecuada en forma de control, información y ayuda [14], estas se muestran en la figura 2.

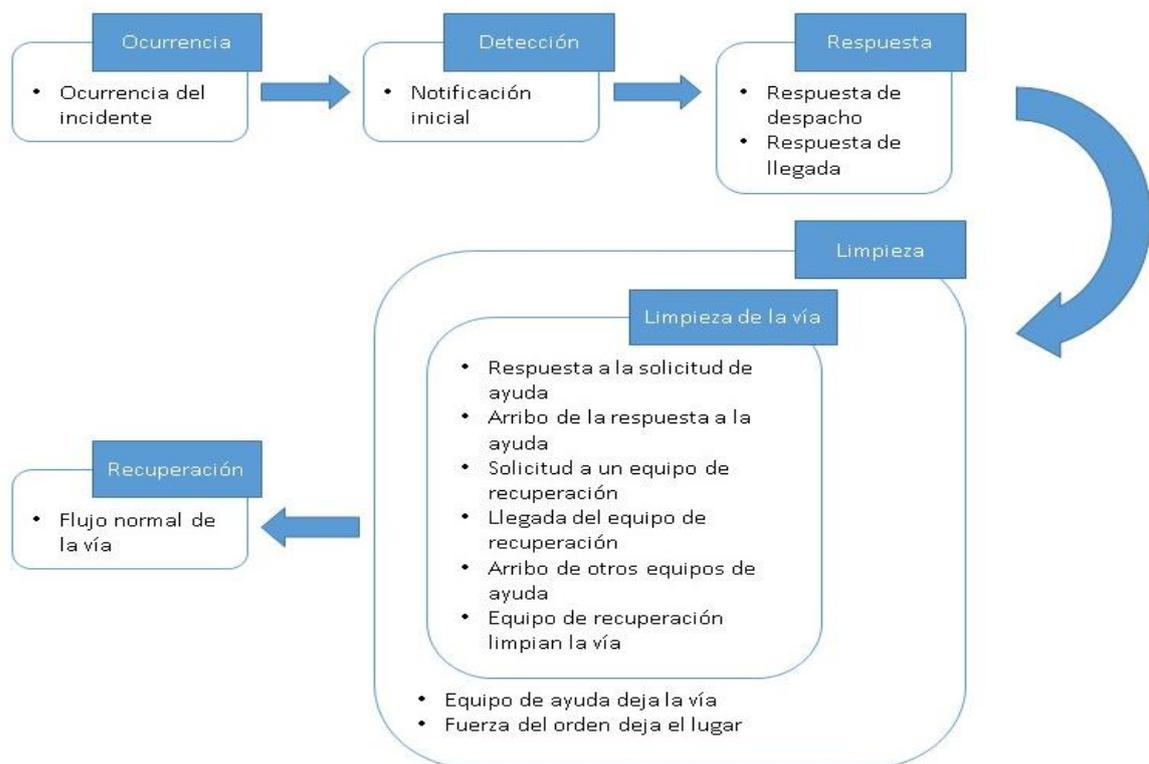


Figura 2: Etapas de incidentes de tránsito [14].

Lo anteriormente mencionado representa la secuencia típica de incidentes de tránsito desde lo más moderado a lo más grave, no todos los procedimientos se aplican en todos los incidentes y puede haber relaciones cruzadas de acciones. Tomando en cuenta que el periodo de recuperación puede ser de cuatro a cinco veces mayor que la duración del

mismo incidente, pero puede ganarse hasta cinco minutos en la fase de recuperación para que el tránsito vuelva a circular con normalidad, por cada minuto que se gana en detectar, verificar y terminar con los incidentes [14].

Al producirse incidentes de tránsito los SITs y los Centros de Control de Tránsito (CCT) toman un papel sumamente importante en cada una de las etapas. En la Tabla 1 se muestran las etapas y funciones que se siguen al producirse un incidente de tránsito.

Tabla 1: SITs y CCT en etapas a seguir en un incidente [14]

Etapa del Incidente	Funciones en incidentes de tránsito	
	Uso de los SITs	CCT
Detección	<ul style="list-style-type: none"> <li>• Sensores que detectan anomalías en el tránsito</li> <li>• Circuito Cerrado de Televisión (CCTV) que capturan imágenes</li> <li>• Vehículos de servicio de emergencias</li> </ul>	<ul style="list-style-type: none"> <li>• Varios softwares de CCT que detectan incidentes</li> <li>• Operadores que observan o reciben reportes telefónicos del público acerca de incidentes</li> </ul>
Verificación	<ul style="list-style-type: none"> <li>• Captura de imágenes por los CCTV</li> <li>• Sensores de tránsito que detectan inconsistencias</li> </ul>	<ul style="list-style-type: none"> <li>• Operadores observadores de incidentes</li> <li>• Operadores que notifiquen cambios de los sensores</li> </ul>
Respuesta	<ul style="list-style-type: none"> <li>• Alertas a través de mensajes en letreros de luces, radio y redes sociales</li> <li>• Establecimiento de cordones de seguridad</li> </ul>	<ul style="list-style-type: none"> <li>• Operadores y especialistas en incidentes</li> <li>• Disponibilidad de toda la red y desarrollo de alternativas</li> </ul>
Despeje de la carretera	<ul style="list-style-type: none"> <li>• Actividades de respuesta y cámara de video para supervisar el lugar</li> </ul>	<ul style="list-style-type: none"> <li>• Gestión de recursos que necesiten</li> </ul>

Despeje de la escena del incidente	<ul style="list-style-type: none"> <li>• Actividades de despeje de la zona del incidente</li> </ul>	<ul style="list-style-type: none"> <li>• Gestionar las llamadas a los recursos necesarios</li> </ul>
Recuperación	<ul style="list-style-type: none"> <li>• Actividades de despeje de la zona del incidente y cámaras de vigilancia para observar la cola de vehículos</li> </ul>	<ul style="list-style-type: none"> <li>• Cuidar de incidentes secundarios</li> <li>• Derivación de la red por desvíos existentes</li> <li>• Envío de mensajes sobre el incidente por medios tradicionales y redes sociales</li> </ul>
Acciones posteriores	<ul style="list-style-type: none"> <li>• Uso de datos e imágenes grabadas para su posterior análisis</li> </ul>	<ul style="list-style-type: none"> <li>• Revisión de los acontecimientos</li> <li>• Mejorar el plan de respuesta de ser necesario</li> </ul>

Durante cualquier incidente de tránsito es importante para evitar colisiones con el personal y equipo de respuesta o apoyo, el uso de uniformes de alta visibilidad, crear una zona de seguridad, tomando en cuenta el paso libre del tránsito de forma segura. Para facilitar información oportuna se pueden utilizar los medios de comunicación y de información del tránsito en forma precisa para que los viajeros conozcan acerca de los incidentes y las actividades de mantenimiento que se están realizando, por ejemplo por medio de una DAI [34].

## 2.2. APRENDIZAJE DE MÁQUINA

El aprendizaje de máquina o aprendizaje automático, más conocido como Machine Learning (ML), nació en 1959 cuando Arthur Samuel científico de Business Machines Corporation (IBM) escribió un programa para jugar damas, para mejorarlo hizo que jugara consigo mismo y así mejorar su rendimiento a través de la experiencia [35]. El ML es un método científico subcampo de las ciencias de la computación, una rama de la inteligencia artificial que se encarga del estudio de algoritmos informáticos que transforman los datos en acciones inteligentes, el principal objetivo es desarrollar técnicas que permitan a los

ordenadores aprender; es decir, generalizar comportamientos y conocimientos a partir de una información suministrada por muestras [36].

El ML requiere del entorno de la disponibilidad de datos, métodos estadísticos y la potencia de los ordenadores para realizar los cálculos de forma rápida y simultánea [35]. Un algoritmo de aprendizaje automático toma datos e identifica los patrones que se puedan usar para realizar algún tipo de acción, su aplicación se lo encuentra en: predecir resultados de elecciones, filtrar correos electrónicos, prever actividades delictivas, automatizar señales de tráfico, predecir el clima, pilotaje y conducción automática, etc [20], [37].

Según [37], dice que una máquina es capaz de aprender, si ésta puede adquirir conocimiento y utilizarla de modo que su rendimiento ayude a mejorar en comparación con experiencias similares en el futuro. El aprendizaje se puede dividir en tres aspectos fundamentales, de la siguiente manera [36]:

- Entrada de datos: usa el almacenamiento de memoria y su recuperación para suministrar una base fáctica para un razonamiento futuro.
- Abstracción: involucra la conversión de datos en representaciones más amplias.
- Generalización: utiliza datos extraídos (conocimiento) para formar una base y su futuro uso en eventos similares.

Las etapas del aprendizaje anteriormente mencionadas están vinculadas ya que sería imposible realizar una de ellas sin la otra, en las computadoras estos procesos deben hacerse de forma explícita, ya que todo el proceso es transparente, el conocimiento adquirido se puede inspeccionar y utilizar para futuras operaciones. En la fase de abstracción, es donde se brinda un significado a los datos a través de su interpretación. Las computadoras representan el conocimiento mediante entradas sin procesar en un modelo (ecuaciones, diagramas de árboles, reglas lógicas, etc.), por lo general este se lo deja a elección de la persona encargada, por otro lado el modelo domina el aprendizaje y los datos a analizar [36].

Al ajuste de un modelo a un conjunto de datos, se le conoce como la tarea del entrenamiento. Una vez terminada esta fase los datos se transforman en una forma abstracta que sintetiza la información original, lo importante de esto es que el modelo respuesta va a descubrir las relaciones entre los datos nunca antes identificadas a simple vista. Después de entrenado el modelo en un grupo de datos inicial se prueba en otro grupo, y se juzga en que métrica se generaliza a los nuevos datos, no todos los modelos se generalizan de forma correcta a cada caso, eso debido al ruido o a las variaciones de

los datos, estos son causados por errores de medición de sensores, de registro incorrecto (valores nulos, perdidos, dañados, etc.) [36].

El aprendizaje automático de cualquier tarea se puede dividir en una serie de pasos adaptables, según [36] propone cinco pasos que son los siguientes:

1. Recopilación de datos: los datos deben ser almacenados en un formato electrónico adecuado para su posterior análisis.
2. Exploración y preparación de datos: usualmente requiere de gran cantidad de intervención humana, la mayor parte del tiempo se lo dedica a aprender sobre los datos.
3. Entrenamiento del modelo: una vez que los datos se hayan preparado para su análisis, se procede al ajuste de ellos mediante un algoritmo de aprendizaje, el cual al final se presentará en forma de modelo.
4. Evaluación del rendimiento del modelo: cada modelo da como respuesta una solución sesgada, se debe evaluar la exactitud de este mediante un conjunto de datos de prueba.
5. Mejora del rendimiento: si el rendimiento del modelo creado no es el esperado entonces es necesario la utilización de estrategias más avanzadas (completar los datos, pre procesamiento, etc.), en algunos casos se cambia por completo a un tipo de modelo diferente.

Existen tres tipos de aprendizaje en ML, estos son el aprendizaje: supervisado (predicción y clasificación); no supervisado (análisis de clúster, reducción de dimensionalidad); y, por refuerzo [20], [35].

Con el aprendizaje supervisado se realizan modelos de predicción, este se utiliza para predecir un valor utilizando otros valores en el conjunto de datos, es decir, el algoritmo de aprendizaje trata de modelar y descubrir la relación entre las características que predice y las demás. El hecho de utilizar el término predicción implica que necesariamente se vayan a prever eventos futuros, por ejemplo, un modelo de predicción se puede usar para el control de semáforos durante las horas pico. Este tipo de aprendizaje resuelve problemas de clasificación y regresión [35], [38].

Los modelos predictivos toman conocimientos claros de lo que deben aprender y como se pretende que lo aprenda, es decir, estos modelos toman características etiquetadas (respuestas), que no es más que tener datos con sus valores objetivos. En otras palabras, dado un conjunto de datos el algoritmo de aprendizaje trata de optimizar una función, en este caso el modelo para encontrar la combinación de valores que dan como resultado la

salida destino [20], [38], la figura 3 muestra el típico procedimiento para este tipo de aprendizaje.

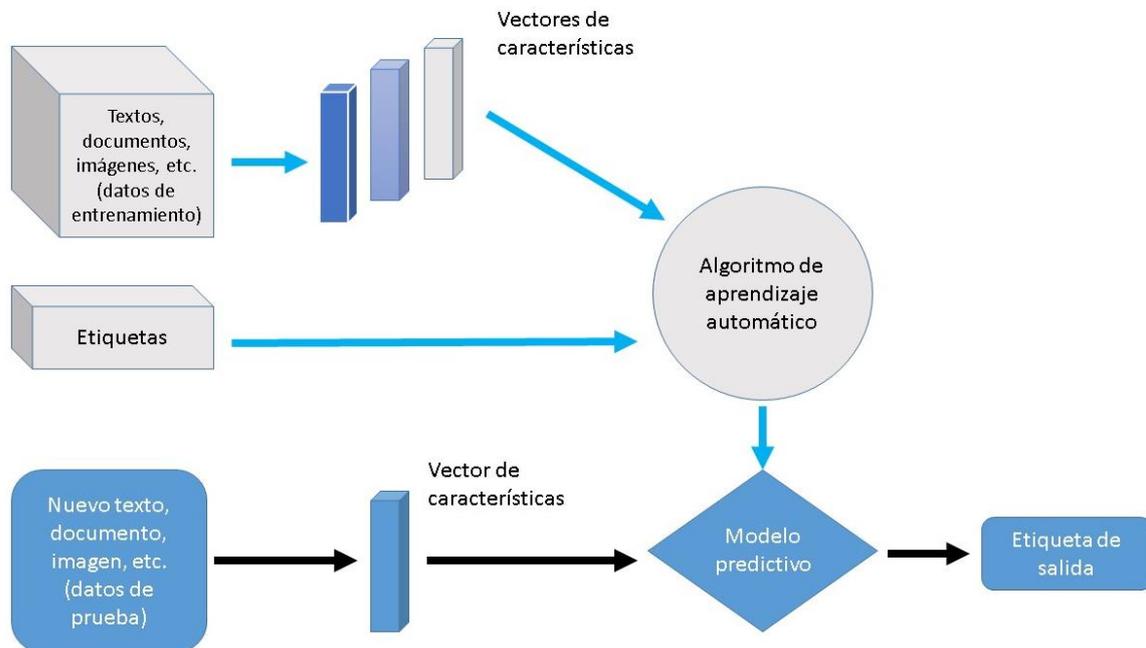


Figura 3: Aprendizaje supervisado[35]

Tomando en cuenta que la etiqueta de salida es una característica categórica conocida como clase, la cual puede tener dos o más niveles, se la conoce como clasificación, por el uso amplio de estos existen muchos tipos de algoritmos de clasificación, los más comunes son: k Vecinos más Cercanos (KNN), regresión logística, regresión lineal, Máquinas de Soporte Vectorial (SVM).

Formalmente, la clasificación en el aprendizaje supervisado se la define como:

Sea el conjunto de entrenamiento,  $T = \{(x_1, y_1), \dots, (x_n, y_n)\}$ , donde dado un patrón de entrada  $x_i \in R^d$ , dado  $y_i \in Y$  que es su clase  $y_n$  indica varias instancias de entrenamiento, el modelo de aprendizaje supervisado aprenderá una hipótesis  $h: X \rightarrow Y$  del conjunto de datos  $T$  para asignar a una entrada  $x \in X$  su clase  $y \in Y$ .

Donde,  $X$  e  $Y$  representan los espacios de la entrada y salida respectivamente.

### 2.2.1. CLASIFICACIÓN CON K – VECINOS MÁS CERCANOS (K-NEAREST NEIGHBORS O KNN)

Los clasificadores de vecinos más cercanos se caracterizan por discriminar datos sin etiquetar, asignándoles la clase de las muestras etiquetadas más afines [36]. Este método es extremadamente poderoso, por lo que ha sido aplicado a varios ámbitos tales como: en visión por computadora en el reconocimiento óptico de caracteres, reconocimiento facial

en imágenes, predecir las preferencias de una persona e identificar patrones en datos genéticos para la detección de enfermedades específicas.

El algoritmo kNN empieza con un conjunto de datos de entrenamiento previamente seleccionado, estos deben estar clasificados en algunas categorías, etiquetados mediante una variable nominal, en otras palabras el conjunto de datos debe estar etiquetado. Para cada registro o vector de características en el conjunto de prueba, kNN halla los k registros en los datos de entrenamiento que son los más cercanos en similitud a través del cálculo de la distancia entre ellos, k es un número entero especificado anteriormente. A la instancia de prueba sin etiqueta se le asigna la clase mayoritaria en k vecinos más cercanos. Es fundamental la aplicación de este algoritmo cuando el conocimiento sobre la distribución de los datos es desconocida [39].

El cálculo de la distancia se lo hace para medir la similitud entre dos instancias, existen varias formas de medir la distancia, por ejemplo: euclidiana, euclidiana al cuadrado, City-block y Chebychev. Tradicionalmente el algoritmo kNN usa la euclidiana que es aquella que mediría dos puntos como si se usara una regla [36], ésta distancia se especifica mediante la Ecuación 1, donde:  $p$  y  $q$  son las muestras a comparar, las cuales tienen  $n$  características. El término  $p_1$  se refiere al valor de la primera característica de la muestra  $p$ , mientras que  $q_1$  se refiere al valor de la primera característica de la muestra  $q$  [36].

$$dist(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}$$

Ecuación 1. Distancia euclidiana

El número de vecinos, es decir, k a utilizar en este algoritmo determinará que tan bien se generalice el modelo a los futuros datos. El equilibrio existente entre el sobreajuste y el desajuste de los datos de entrenamiento es conocido como compensación sesgo-varianza, si se utiliza un k grande este reduciría el impacto en datos ruidosos, pero de lo contrario el ruido y casos atípicos influirían en la clasificación. Normalmente k se establece entre 3 y 10, sin embargo un enfoque diferente sugiere probar varios valores de k en múltiples conjuntos de datos de prueba que ofrezcan el mejor rendimiento de la clasificación [36].

El algoritmo de kNN para cualquier caso de clasificación, sigue los pasos del Algoritmo 1 [39].

Algoritmo 1: Clasificación con kNN.

Algoritmo de clasificación kNN.

**Para** Todas las muestras desconocidas  $U_i$  **Hacer**

**Para** Todas las muestras conocidas  $S_j$  **Hacer**

        Calcular la distancia entre  $U_i$  y  $S_j$

**Fin Para**

    Encontrar las  $k$  distancias más cortas

    Localizar las muestras correspondientes  $S_{j_1}, \dots, S_{j_k}$

    Asignar a  $U_i$  la clase que aparece con más frecuencia

**Fin Para**

### 2.2.2. CLASIFICACIÓN CON REGRESIÓN LOGÍSTICA

También llamada logit, ha tomado popularidad en la última década [40], apareció en el año de 1970, como una alternativa a la técnica del análisis de función de discriminante lineal, para la predicción de resultados dicotómicos, su principal objetivo es el de estimar la probabilidad de que una muestra (instancia o vector de características) pertenezca a una clase en particular [20], su especialidad es si la probabilidad de una instancia es mayor que 50% esta predice su pertenencia a la clase positiva, lo que le hace un clasificador binario, dicha técnica tiene acogida en muchos ámbitos como los son: educación superior, investigación educativa y ciencias naturales [41], [42].

El modelo de regresión logística calcula una suma ponderada de las muestras de entrada y genera la logística de este resultado, como se muestra en la Ecuación 2.

$$\hat{p} = h_{\theta}(x) = \sigma(\theta^t \cdot x)$$

Ecuación 2. Regresión Logística

Donde la logística  $\sigma(\cdot)$  es una función sigmoidea, la que genera un número entre 0 y 1. Esta función se define como en la Ecuación 3.

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

Ecuación 3. Función sigmoidea

Cuando el modelo de regresión logística ha estimado la probabilidad, es decir,  $h_{\theta}(x)$  de que una muestra  $x$  pertenece a la clase positiva, puede realizar la predicción  $\hat{y}$ , mediante la ecuación 4. Este modelo predice 1 si  $\theta^T \cdot x$  es positivo, de lo contrario será 0.

$$\hat{y} = \begin{cases} 0 & \text{si } \hat{p} < 0.5 \\ 1 & \text{si } \hat{p} \geq 0.5 \end{cases}$$

Ecuación 4. Predicción de la regresión logística

El principal funcionamiento del objetivo es el formar el vector de parámetros  $\theta$  de manera que el modelo estime probabilidades altas ya sea para instancias negativas ( $y = 0$ ) y positivas ( $y = 1$ ). Esto se lo hace mediante la ecuación de costo (Ecuación 5), en este caso solo para una muestra de entrenamiento.

$$c(\theta) = \begin{cases} -\log(\hat{p}) & \text{si } y = 1, \\ -\log(1 - \hat{p}) & \text{si } y = 0 \end{cases}$$

Ecuación 5. Costo de la ecuación

La función de costo de todo el conjunto de entrenamiento es el coste medio de todas las muestras de entrenamiento, esta técnica es conocida como pérdida logarítmica, como se muestra en la Ecuación 6.

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(\hat{p}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{p}^{(i)})]$$

Ecuación 6. Perdida logarítmica

Por la carencia de una ecuación para calcular el valor de  $\theta$  que minimice la función de costo, por lo que se aplica la técnica del gradiente descendiente que encuentre el mínimo global, las derivadas parciales de la función de costo del j-esimo parámetro del modelo  $\theta_j$ , está dada por:

$$\frac{\partial}{\partial \theta_j} J(\theta) = -\frac{1}{m} \sum_{i=1}^m (\sigma(\theta^T \cdot x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Ecuación 7. Derivadas parciales de la función de costo

### 2.2.3. CLASIFICACIÓN CON MÁQUINA DE SOPORTE VECTORIAL

Los SVMs son un conjunto de modelos muy populares de aprendizaje automático supervisado, muy potente capaz de realizar clasificaciones lineales o no lineales, el principal objetivo de este modelo es el de crear un límite plano, más conocido como hiperplano que conlleva a la partición de los datos del conjunto de entrada en grupos de diferentes categorías anteriormente descritas de forma homogénea [20], [43]. Existen varias aplicaciones en las que los SVMs han tenido éxito en la clasificación binaria, ya que es su forma tradicional de uso [36], tal es el caso de: clasificación de datos de

experimentación genética, categorización de texto, detección de eventos raros (fallas de motores de combustión, brechas de seguridad, terremotos) [44].

Un hiperplano es un conjunto de puntos que satisfacen  $w \cdot x + b = 0$  [44], se define como: en un espacio  $d$ -dimensional, un hiperplano es un subespacio semejante al plano de la dimensión del hiperplano ( $d - 1$ ), por ejemplo, en tres dimensiones, un hiperplano es un subespacio de dos dimensiones. Para dimensiones mayores a tres su visualización es muy compleja, pero su noción se mantiene [45], [46].

El hiperplano  $d$ -dimensional, matemáticamente se define por:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_d X_d = 0$$

$$\{x: f(x) = x^t \beta + \beta_0 = 0\}$$

Ecuación 8. Hiperplano

Donde  $\beta$  es una unidad del vector  $\|\beta\| = 1$  y  $x^T$  permite producir un hiperplano de tamaño  $d - 1$  [46].

La clasificación de hiperplanos dividen los datos en grupos de elementos similares, tal como lo indican los valores de las clases, tomando a [46], [47] en la teoría de clasificación con hiperplanos, formalmente el modelo lineal de regresión que participa en las combinaciones de las variables de entrada:

$$y(x, w) = w_0 + w_1 x_1 + \dots + w_i x_i$$

Donde  $x = (x_1, x_2, \dots, x_i)^T$  es la regresión lineal y  $w_0, w_1, \dots, w_i$  son los parámetros que forman parte de la función lineal de las variables  $x_i$ . Considerando las combinaciones no lineales de las variables de entrada tiene la forma:

$$y(x, w) = w_0 + \sum_{k=1}^{N-1} w_k \phi_k(x)$$

En la cual  $\phi_k(x)$  es la función base, el número total de parámetros en el modelo es  $N$  y el sesgo más conocido como bias es  $w_0$ , el cual permite el desplazamiento fijo de los datos, por lo general se lo suma a la función base. Por esta razón se obtiene:

$$y(x, w) = \sum_{k=0}^N w_k \phi_k(x) = w^T \phi(x)$$

Donde los parámetros de  $w$  y  $\phi$  se extienden hasta  $N - 1$ , es decir,  $w = (w_0, w_1, \dots, w_{N-1})^T$  y  $\phi = (\phi_0, \phi_1, \dots, \phi_{N-1})^T$ , por lo que los problemas de clasificación lineal, mediante este modelo, se representan por la Ecuación 9, donde  $\phi(x)$  es el espacio de muestras de entrenamiento y  $s$  es el sesgo.

$$y(x) = w^T \phi(x) + s$$

Ecuación 9. Solución básica de problemas de clasificación lineal

Tomando de inicio un conjunto de entrenamiento  $M$  con vectores de entrada  $x_1, x_2, \dots, x_i$ , los que tienen valores de etiqueta de la clase  $y_1, y_2, \dots, y_i$ , donde  $y_i \in \{-1, 1\}$ , los puntos  $x$  son clasificados según el signo de  $y(x)$ . La finalidad es el construir un clasificador con los datos de entrenamiento que encasillen exactamente las muestras de cada clase [44]–[46].

Tomando en cuenta que se puede construir un hiperplano que separe las muestras de entrenamiento de acuerdo con la clase objetivo, entonces es posible que estos hiperplanos clasifiquen las muestras posteriores. Por ejemplo en la Figura 4, se tienen etiquetadas las muestras, en este caso cuadrados  $y_i = -1$  y triángulos  $y_i = 1$ , se muestran los posibles hiperplanos que separan en grupos de figuras, en dos dimensiones, ya que estas se pueden dividir por una superficie plana o recta, se dice que son linealmente separables [44]–[46].

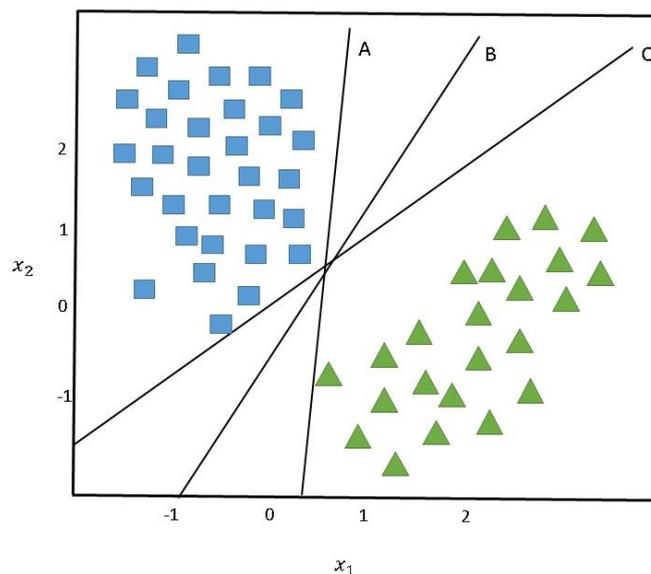


Figura 4: Clasificación con hiperplanos[20], [36], [46]

Un hiperplano que puede separar muestras, tienen la siguiente propiedad [46]:

$$x^T \beta + \beta_0 > 0, \text{ si } y(x_i) > 0 \text{ para } t_i = 1$$

$x^T \beta + \beta_0 < 0$ , si  $y(x_i) < 0$  para  $t_i = -1$

El hiperplano que separa de forma adecuada, es decir, el hiperplano separador tiene la siguiente propiedad:

$$y_i(x^T \beta + \beta_0) > 0, \forall i \geq 1$$

Con ello se puede clasificar las muestras  $x$  en clases, así que  $y(x_i)$  es positivo y se asigna la clase 1, de lo contrario es -1, la Figura 5 muestra lo anteriormente dicho.

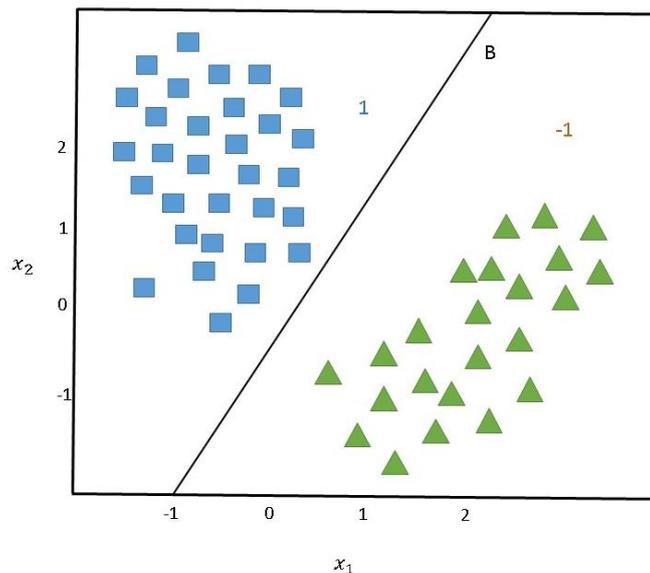


Figura 5: Clasificación con hiperplanos [20], [36], [46].

### 2.2.3.1. CLASIFICADOR DE MARGEN MÁXIMO

El hiperplano que crea una separación óptima entre las muestras de entrenamiento y esté lo más alejado posible de las muestras más cercanas de él, es conocido como Hiperplano de Margen Máximo (MMH), este conduce una mejor generalización para datos futuros, esto debido a que pueden existir variaciones en las posiciones de los puntos cercanos al límite y pueden caer en la línea por casualidad [20], [45].

Matemáticamente las muestras  $x$  del conjunto de datos de entrenamiento, se clasificarán mediante el signo de  $y(x_i)$ . La Figura 6 muestra un ejemplo de la construcción de un MMH, los puntos equidistantes son conocidos como vectores de soporte, los cuales son los puntos de cada clase más cercanos al MMH, al menos un vector de soporte pertenece a una de cada clase. Utilizando los vectores de soporte es posible encontrar el MMH que facilitan el almacenaje del modelo en forma compacta aún si el número de características es grande [44]–[46].

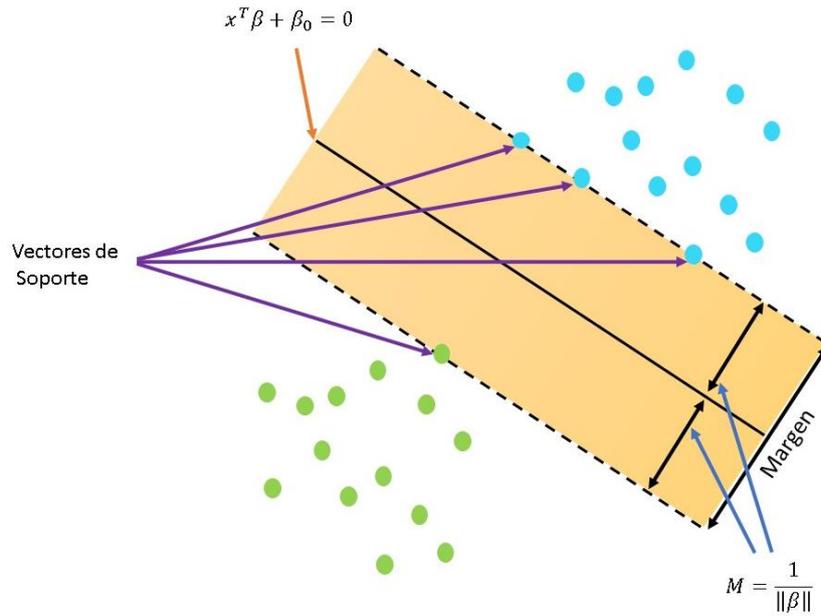


Figura 6: Hiperplano de Margen Máximo (MMH)[20], [36], [46]

En la construcción un MMH fundamentado en el conjunto de datos de entrenamiento de  $n$  muestras  $x_1, x_2, \dots, x_n \in \mathbb{R}^d$  donde  $d$  es la dimensión, cada una asociada a las etiquetas  $y_1, y_2, \dots, y_n \in \{-1, 1\}$ . se debe garantizar que cada muestra se encuentre perfectamente clasificada, para encontrar el MMH óptimo, esto se define en la Ecuación 10, donde  $M$  es el ancho del margen.

$$\max_{\beta, \beta_0, \|\beta\|=1} M$$

$$y_i(x^T \beta + \beta_0) \geq M, \forall i = 1, \dots, n$$

Ecuación 10. Clasificador de Margen Máximo

En algunos casos existen muestras que no son separables y no se cumple la Ecuación 9. Por lo tanto no es posible el uso del clasificador de margen máximo [20], [36].

### 2.2.3.2. CLASIFICADOR DE MARGEN SUAVE

Para evitar problemas al no tener muestras linealmente separables, es preferible utilizar un modelo más flexible como el clasificador por margen suave o de soporte vectorial que ayuda con el tratamiento en la clasificación de datos atípicos, para dar robustez a la clasificación de nuevos datos [45], [46].

Con base en [48], un modelo de distribución con vectores  $x$  para cada clase se usa el estimado de densidad de Parzen con núcleos Gaussianos con  $\sigma^2$  como parámetro común, que define con margen de error a un clasificador, lo que conlleva a la búsqueda del mejor

hiperplano que minimice la probabilidad de error, por lo que  $\lim_{\sigma \rightarrow 0} \sigma^2$  tiene el margen máximo.

La distancia de un punto  $x_i$  en el hiperplano está dado por la Ecuación 11.

$$\frac{t_i y(x_i)}{\|\beta\|} = \frac{t_i (x^T \beta + \beta_0)}{\|\beta\|}$$

Ecuación 11. Distancia perpendicular de un punto en el hiperplano

El margen se calcula por las distancias perpendiculares a los puntos más cercanos al conjunto de datos de entrenamiento, por esa razón se optimizan los parámetros como  $\beta$  y sesgo, maximizando la distancia. Esta viene dada por  $\arg \max_{w,b} \left\{ \frac{1}{\|\beta\|} \min_n [t_n (x^T \beta + \beta_0)] \right\}$ , la optimización del margen se lo tiene por la Ecuación 12.

$$\begin{aligned} & \min_{\beta, \beta_0} \|\beta\| \\ & y_i (x^T \beta + \beta_0) \geq 1, \forall i = 1, \dots, n \\ & y_i (x^T \beta + \beta_0) \geq M(1 - \xi_i), \xi_i \geq 0, \sum_{i=1}^n \xi_i \leq C \end{aligned}$$

Ecuación 12. Vectores de soporte en el hiperplano

Donde  $C$  es el parámetro de costo o de ajuste [36],  $\xi_i$  son las variables de holgura que admiten a las muestras ubicadas en el lado contrario se tomen en cuenta para el cálculo del margen máximo y  $M$  es el ancho del margen que se busca optimizar haciéndolo más grande. Para las variables de holgura se toman las siguientes consideraciones [45], [46]:

- Si  $\xi_i > 0$ , la muestra está en el lugar incorrecto del margen.
- Si  $\xi_i > 1$ , la muestra se encuentra en el lugar incorrecto del hiperplano.
- Si  $\xi_i = 0$ , la muestra se ubica en el lugar correcto.

La modificación del parámetro  $C$  ayuda al ajuste de la penalización de las muestras que se ubican en el lado equivocado del hiperplano, en la Figura 7 se muestra con mayor detalle la interacción de los parámetros, tomándose las siguientes consideraciones [45], [46]:

- Si  $C = 0$ , no existen violaciones al margen.
- Si  $C > 0$ , las muestras se encuentran en el lado incorrecto del hiperplano y se verifica que  $\xi_i > 1$  y se necesita que  $\sum_{i=1}^n \xi_i \leq C$ . Con ello:
  - a) Si  $C$  disminuye, el margen también lo hace y es menos tolerante a las muestras
  - b) Si  $C$  aumenta, el margen también lo hace y es mucho más tolerante a las muestras.

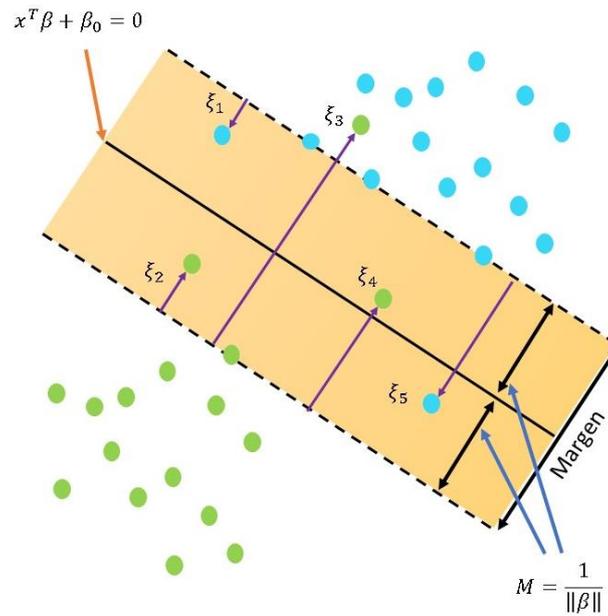


Figura 7: Hiperplano con vectores de soporte de margen suave [20], [36], [46].

Al aplicar un valor de costo  $C$  a los puntos que violan las restricciones, el algoritmo intenta minimizar dicho costo total, por lo tanto, se tiene la solución programable de optimización, tal como lo indica la Ecuación 13, la cual usa multiplicadores de Lagrange [45], [46].

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N \xi_i$$

$$\text{Donde } \xi \geq 0, y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i, \forall i$$

Ecuación 13. Hiperplano de soporte computacional

En muchos de los casos prácticos, las relaciones entre variables no son linealmente separables, sin embargo, la adición de variables de holgura ( $C$ ) no es la única solución para tener una clasificación no lineal, la principal característica de SVM es el poder mapear el problema en un espacio con mayor dimensión mediante el método o proceso del núcleo más conocido como kernel. Al utilizar kernels que no son lineales, estos adicionan dimensiones extras a los datos para su separación, es decir, agregan nuevas características que expresan relaciones matemáticas entre características medidas [36].

Los SVMs con kernels no lineales son altamente poderosos y eficaces al momento de realizar su trabajo de clasificación ya que permiten el conocimiento de medidas nuevas de forma directa en los datos originales de entrenamiento. Por tanto son realmente eficaces para problemas de clasificación porque no tienden al sobreajuste del modelo y no están afectados por los datos con ruido. Pero también para encontrar el mejor modelo depende

de varias combinaciones de kernels y parámetros del modelo, su entrenamiento puede ser lento; el modelo es muy complejo para interpretarlo [36],[45], [46].

Las funciones de kernels, por lo general son de la siguiente forma:

$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$$

Donde  $\phi(\vec{x})$  es un mapeo de los datos en otro espacio, algunos de los kernels más utilizados son polinomial, RBF Gaussiano y sigmoidea [36]. En la Figura 8 se muestra un ejemplo de la separación de datos con este kernel.

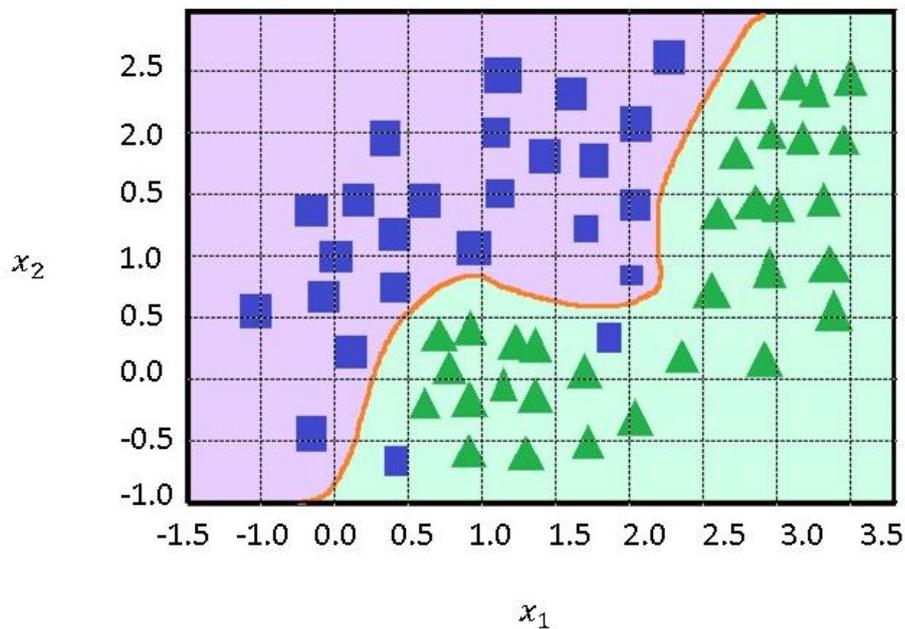


Figura 8: Clasificación de kernel polinomial [20], [36], [46]

El kernel lineal de grado  $g$  agrega una transformación no lineal simple de los datos, que hace posible el no obtener demasiado número de funciones combinatorias, este puede ser expresado de la siguiente manera:

$$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^g$$

La función de núcleo sigmoidea da como resultado un modelo de SVM muy parecido al de una red neuronal artificial con función de activación sigmoidea, se puede representar como:

$$K(\vec{x}_i, \vec{x}_j) = \tanh(k\vec{x}_i \cdot \vec{x}_j - \delta), \text{ donde } k \text{ y } \delta \text{ son utilizados como parámetros del kernel.}$$

El kernel RBF Gaussiano es un tipo de núcleo muy utilizado en muchos tipos de datos y la mayoría de los expertos lo tienen como un punto de partida razonable para las tareas de

aprendizaje, este funciona como un tipo de red neuronal artificial RBF y se lo puede expresar como:

$$K(\vec{x}_i, \vec{x}_j) = e^{-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{2\sigma^2}}$$

En la Figura 9 se presenta un ejemplo de la clasificación con kernel RBF Gaussiano.

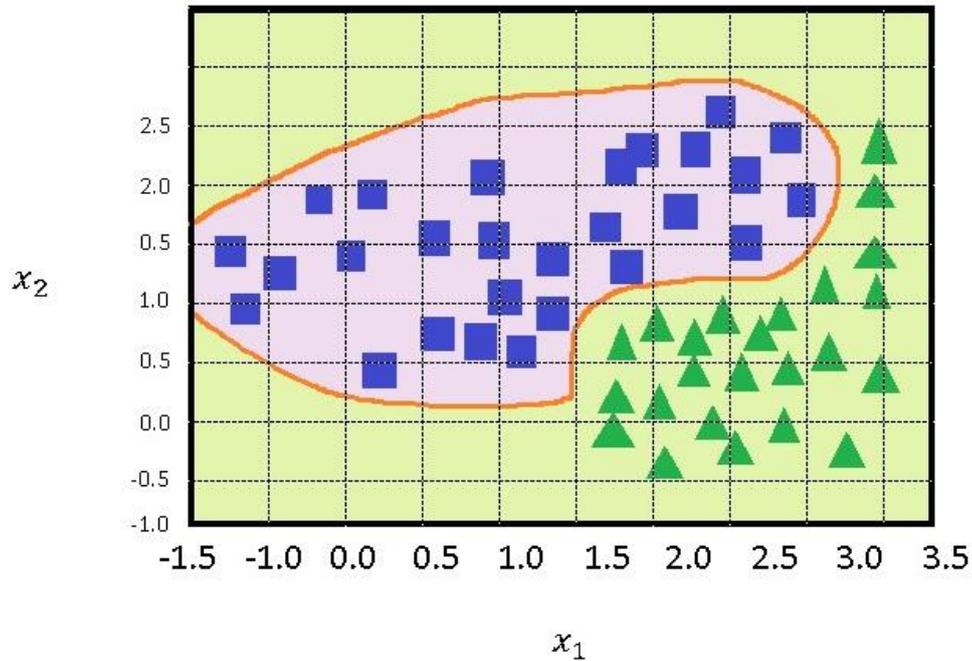


Figura 9: Clasificación SVM con kernel RBF Gaussiano [20], [36], [46]

Según [20], [36] recomienda el utilizar kernel RBF como un buen inicio, especialmente cuando el conjunto no es demasiado grande y sin demasiadas características, en la mayoría de los casos para la elección adecuada del kernel se requiere prueba y error al momento del entrenamiento y evaluación, en la mayor parte de los casos la elección del núcleo va a ser de forma arbitraria, cabe recalcar que al seguir utilizando otros núcleos valdría la pena experimentar con validación cruzada.

### 2.3. INDICADORES DE RENDIMIENTO DEL MODELO DE CLASIFICACIÓN

Para la medición del desempeño de un modelo de clasificación se usa la matriz de confusión, para comparar de forma visual el rendimiento de los valores de clasificación del conjunto de predicciones con los objetivos reales [36]. Normalmente las matrices de confusión se utilizan para estimar la exactitud general de la clasificación [49], así como la identificación del error que afectan a las categorías [50].

Una matriz de confusión es una tabla que clasifica las predicciones según coincida en la categoría, tomando en cuenta el valor real de los datos. Cada columna indica las categorías predichas, las filas muestran lo mismo para los valores reales, mientras que las celdas muestran el número de observaciones, se puede crear una matriz de confusión para cualquier número de clases [20], [36]; en la Figura 10 se muestra un ejemplo de matriz de confusión de  $2 \times 2$ , donde: TP significa Verdaderos Positivos, TN Verdaderos Negativos, FP Falsos Positivos y FN Falsos Negativos, en esta parte TP y TN son los correctamente clasificados.

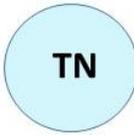
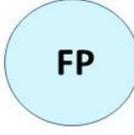
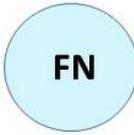
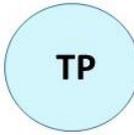
		Valores Predichos	
		Prediccion Negativa	Prediccion Positiva
Valores Reales	Condicion Negativa	 <b>TN</b> Verdaderos Negativos	 <b>FP</b> Falsos Positivos
	Condicion Positiva	 <b>FN</b> Falsos Negativos	 <b>TP</b> Verdaderos Positivos

Figura 10: Matriz de confusión de dos clases [36]

La matriz de confusión proporciona mucha información, pero en ciertas ocasiones es preferible saber una métrica en concreto, por ejemplo la métrica de la exactitud (AC) también llamada tasa de éxito, es la proporción que representa el número total de predicciones que fueron correctamente clasificadas, se lo calcula mediante la Ecuación 14, la tasa de error puede ser calculada por  $1 - AC$  [36].

$$AC = \frac{TP + TN}{TP + TN + FP + FN}$$

Ecuación 14. Métrica de Exactitud del modelo

Otra métrica conocida es la sensibilidad o también llamada tasa de verdaderos positivos, esta mide las muestras positivas que se calcularon de forma correcta, por lo que se calcula

como el número de  $TP$  dividido por el número total de positivos en los datos ( $TP + FN$ ), tal como se muestra en la Ecuación 15 [20], [36].

$$Sensibilidad = \frac{TP}{TP + FN}$$

Ecuación 15. Métrica de Sensibilidad

La especificidad es aquella que calcula la tasa de negativos verdaderos, es decir, mide la porción de muestras negativas que se clasificaron correctamente, se calcula con la Ecuación 16 [20], [36], [51].

$$Especificidad = \frac{TN}{TN + FP}$$

Ecuación 16. Métrica de Especificidad

A la precisión se la conoce también con el nombre de Exactitud Positiva Verdadera (EPV), esta expresa la porción de las muestras positivas predichas que corresponden a positivos reales. Esta es una medida que se concentra en los positivos predichos, la precisión se define en la ecuación 17 [52].

$$Precision = \frac{TP}{TP + FP}$$

Ecuación 17. Medida de Precisión

Existe otra medida importante como es  $F_1$  – score, pertenece a las medidas F. Es la media armónica de la precisión y la sensibilidad con pesos iguales. Se calcula con la ecuación 18 [53].

$$F_1 = 2 \cdot \frac{precision \cdot sensibilidad}{precision + sensibilidad} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

Ecuación 18.  $F_1$  score

### **Área Bajo la Curva ROC (AUC ROC)**

Proporciona una medición bidimensional de toda el área bajo la curva ROC (Características de Funcionamiento del Receptor), facilita una medida del rendimiento de los posibles umbrales de clasificación, es decir; es la probabilidad de que un modelo clasifique una muestra aleatoria positiva más alta que una negativa. Esta métrica es invariable con respecto a la escala y con respecto al umbral de clasificación [54]. Puede ser calculada por la siguiente ecuación.

$$AUC = \frac{1}{2}(1 + TPR - FPR)$$

Donde:

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN}$$

Con  $FPR$  es la Tasa de Falsos Positivos,  $TNR$  es la Tasa de Verdaderos Negativos,  $FP$  son las muestras que se posicionan en Falsos Positivos,  $TN$  Verdaderos Negativos,  $N$  Negativos y  $TPR$  es la sensibilidad.

## 2.4. VALIDACIÓN CRUZADA K-PLIEGUES

Es un método estadístico de evaluación y comparación de algoritmos de aprendizaje de automático, también es conocido como k-fold Cross Validation, apareció en el año de 1930 y con el tiempo ha ido mejorando. Se utiliza para mejorar la calidad en la predicción, este método consiste en tomar los datos de entrenamiento y dividirlo en dos partes de forma aleatoria: la una será usada para el entrenamiento y la otra para la validación. En el entrenamiento se utiliza  $k - 1$  partes y lo restante valida el modelo, este proceso se repite  $k$  veces y en cada iteración se selecciona un conjunto de pruebas diferente. Al finalizar la exactitud total se calcula como el promedio de las exactitudes de los  $k$  modelos entrenados, la validación más común que se utiliza es la de  $k = 10$  [55], [56], en la Figura 11 se muestra el procedimiento de validación cruzada k-pliegues con  $k = 3$ .

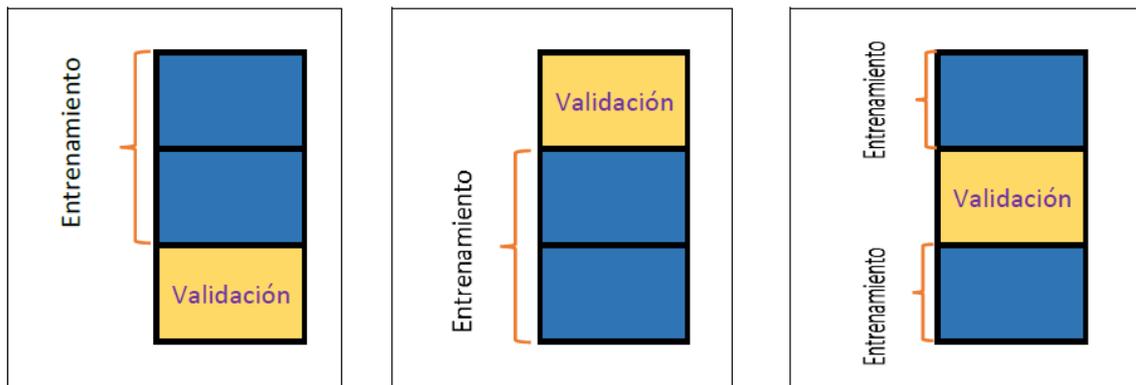


Figura 11: Validación cruzada proceso [56]

## 2.5. DETECCIÓN DE OBJETOS

Es una parte de la visión artificial, que se encarga del estudio de la detección de objetos en imágenes, a partir de su apariencia visual [57], ésta no solo se encarga de clasificar

varios tipos de imágenes que se encuentran, sino que también estiman con precisión la ubicación (localización) de los objetos [58], tiene varias aplicaciones como en la investigación científica y en la industria (reconocimiento de rostro, detección de imágenes médicas, texto, peatones). Por la existencia de varios puntos de vista como la iluminación, oclusiones, se puede dividir en tres etapas [23], [59]:

- Selección de la región de información: comúnmente se utiliza una ventana multi escala que se desliza por la imagen escaneada, esto se hace ya que los objetos pueden aparecer en diferentes posiciones y tamaños de la imagen, lo que ocasiona un costo computacional alto. Ya que, si se aplicara un número de plantillas fijo para las ventanas deslizantes, este podría dar regiones no apacibles.
- Extracción de característica: son comúnmente llamados descriptores, son la extracción u obtención de modelos matemáticos compactos, que resumen el contenido de la imagen, estos se usan para simplificar el aprendizaje y reconocer los diferentes objetos, los más conocidos son: HOG, características similares Haar, DIFT. De acuerdo a las propiedades de las imágenes no existe un descriptor robusto para describir de forma perfecta a todos los objetos [23].
- Clasificación: para distinguir entre las diferentes clases de objetos a detectar, se utilizan clasificadores los más populares son SVM, AdaBoost y Modelo Basado en Piezas Deformables (DPM) [23], por medio de las Redes Neuronales Profundas (DNN) se ha tenido un gran avance por medio de las CNN y R-CNN que tienen la capacidad de aprender características complejas. Existe un aporte considerable de modelos entre las más destacadas se tienen: Mira Solo una Vez (YOLO) y Single Shot Detection (SSD) Mobilenet [22], [23].

La detección de objetos con aprendizaje profundo se ha vuelto muy popular a partir del 2006, esto se debe a la aparición de una gran cantidad de datos etiquetados, desarrollo de sistemas informáticos en paralelo de rendimiento alto y el avance en las estructuras de redes [23]. El objetivo principal de la detección de objetos es el de situar y clasificar (tipo de categoría) objetos en una imagen por medio de cuadros delimitadores (Bounding Boxes o BB) para comprobar su existencia.

La clasificación de objetos se puede especificar en dos tipos:

- Propuesta de región: se basa en la exploración de todo el escenario y luego se concentra en las regiones de interés (RoI), se tienen algunos tipos de ellas en las DNN, las más destacadas son: R-CNN (mejora los BBs candidatos), SPP-Net (reutiliza mapas de características de la quinta capa), Fast R-CNN (se aplica la

perdida multitarea), Faster R-CNN (aplica métodos adicionales como búsqueda selectiva y Edgebox), R-FCN (se compone de dos subredes: una totalmente conectada convolucional compartida y la otra no compartida de Rol), FPN (utiliza pirámides de características de imágenes, su deficiencia es el tiempo de entrenamiento), Mask R-CNN (agrega una rama para predecir máscaras de segmentación píxel a píxel) [23], [59].

- Regresión / clasificación: mapea directamente los píxeles de la imagen, las probabilidades de la clase y las coordenadas de los BBs, existen varios modelos que sobresalen en este tipo como: Pioner Works (en la imagen de prueba se genera una máscara binaria y extrae detecciones con la inferencia simple de un BB), YOLO (usa el mapa de características para predecir la confianza de las categorías y los BBs), SSD (da un mapa de características fijo, fusiona varias predicciones) [23], [59].

### 2.5.1. YOLO v3

Apareció en 2015, fue propuesta por Redmon, tiene por significado Solo Miras una Vez (You Only Look Once), en su primera versión o más conocido como YOLO v1 propone una sola red neuronal artificial de extremo a extremo, la cual predice BBs y probabilidades de clase [59], [60], es decir; unifica los componentes que están separados en la detección de objetos en una sola red.

Este tipo de red utiliza una imagen, la que es dividida en cuadrículas de  $S \times S$ , si el centro de algún objeto cae dentro de una de las celdas de la cuadrícula, esta será responsable de la detección de dicho objeto, además cada celda predice  $B$  cantidad de BBs, puntajes de la confianza de clase de pertenencia y  $C$  probabilidades de clases condicionales [60]. Si no existen objetos en la celda esta dará como confianza el valor de cero. Cada BB consta de cinco predicciones:  $x, y, w, h$  y *confianza (confidense)*, donde [60]:

- $x$  y  $y$  son las coordenadas del centro de la celda de la cuadrícula.
- $w$  y  $h$  se refieren al ancho y largo, respectivamente, con relación a la imagen completa.
- *confianza* es la probabilidad de que se haya detectado un objeto.

YOLO v1 tiene una arquitectura de 24 capas convolucionales seguidas de dos capas completamente conectadas, con una entrada para imagen de  $488 \times 488$ , las primeras capas son aquellas que extraen características y las completamente conectadas predicen las coordenadas y probabilidades de salida, esta red fue entrenada con el conjunto de datos PASCAL VOC [60]. Este tipo de detector es sensible a objetos pequeños ya que una celda de la cuadrícula solo es capaz de predecir dos BBs y una sola clase tienen

dificultades para generalizar objetos en relaciones de aspecto inusuales y error en localizaciones incorrectas [23], [59], [60].

En YOLO v2 o YOLO9000 se centra en la mejora de la sensibilidad y la localización de la detección de objetos manteniendo la precisión de la clasificación, mediante [59], [61]:

- Normalización de lotes: ayuda a la regularización de la convergencia del tiempo y del modelo sin sobreajustar.
- Convolución con cajas de anclaje: usa Faster R-CNN para predecir los BBs utilizando alguno de ellos a priori a mano, se eliminan las dos capas completamente conectadas y se usan cajas de anclaje.
- Dimensiones de clústeres: se utiliza clusterización con k-means estándar con distancia euclidiana para seleccionar las dimensiones de los BBs, con el objetivo de obtener antecedentes que dirijan a puntuaciones altas, se utilizó un  $k = 5$ .
- Predicción de ubicación directa: predice cinco BBs para cada celda de la malla.

YOLO v2 fue entrenada con datos de clasificación de ImageNet y de detección de objetos COCO, esta red logra generalizar mejor los objetos que YOLO v1, además de brindar un espacio de salida más detallado [61].

En YOLO v3 se beneficia las ideas de YOLO v1 y YOLO v2 / 9000, en la predicción del BB predice la puntuación de este mediante regresión logística, será 1 si el BB anterior se superpone a un objeto verdadero, si un BB se asigna a un objeto verdadero anterior no incurre en pérdida de predicción de clase, solo objetividad. En la predicción de clase cada BB predice las clases de objeto que este puede contener, para ello se utiliza clasificadores logísticos que ayudan a tener datos más complejos o superpuestos [23], [59], [62].

YOLO v3 realiza predicciones de BB en tres escalas diferentes parecidas a las redes piramidales, se hace mediante la toma de mapas de características de las capas anteriores y lo aumentan en  $2 \times$ , de la misma forma toma dicho mapa de la red anterior, luego agregan varias capas convolucionales, el procedimiento anteriormente mencionado se repite con el fin de hallar el BB final, aún sigue manteniendo la clusterización por k-means con  $k = 9$ . La arquitectura de YOLO v3 está compuesta de 53 capas convolucionales que es una mezcla entre YOLO v2, Darknet-19 y la agregación anteriormente expuesta con entrada de imagen de  $416 \times 416$ , en la Figura 12 se muestra toda la arquitectura. Se usó en su entrenamiento imágenes completas de múltiple escala, aumento datos, normalización de lotes como lo hecho anteriormente (YOLO v2) [23], [59], [62].

	Tipo	Filtro	Tamaño	Salida
	Convolutacional	32	3 x 3	256 x 256
	Convolutacional	64	3 x 3 / 2	128 x 128
1x	Convolutacional	32	1 x 1	128 x 128
	Convolutacional	64	3 x 3	
	Residual			
	Convolutacional	128	3 x 3 / 2	
2x	Convolutacional	64	1 x 1	64 x 64
	Convolutacional	128	3 x 3	
	Residual			
	Convolutacional	256	3 x 3 / 2	32 x 32
8x	Convolutacional	128	1 x 1	32 x 32
	Convolutacional	256	3 x 3	
	Residual			
	Convolutacional	512	3 x 3 / 2	16 x 16
8x	Convolutacional	256	1 x 1	16 x 16
	Convolutacional	512	3 x 3	
	Residual			
	Convolutacional	1024	3 x 3 / 2	8 x 8
4x	Convolutacional	512	1 x 1	8 x 8
	Convolutacional	1024	3 x 3	
	Residual			
	Avgpool		Global	
	Conectada		1000	
	Softmax			

Figura 12: Arquitectura Darknet-53 (YOLO v3) [62]

### 2.5.2. SSD MOBILENET

SSD fue la primera red de detección de objetos en no usar propuesta de región y utilizó una red profunda [59], además es uno de los enfoques más populares en la detección de objetos [63], esta red necesita una sola imagen de entrada, usa un conjunto de cuadros de anclaje predeterminados con escalas y proporciones diferentes con el fin de discretizar la salida de los BBs.

Este modelo se basa en una red convolutacional de retro propagación que produce un conjunto de BBs y puntajes de instancia de clases, para la detección final se aplica la supresión no máxima. Las capas primeras se usan para la clasificación de las imágenes, además de ello se aumentan capas convolutacionales al final de la red para hacer

detecciones con: mapas de características de múltiple escala, predicciones convolucionales (como se presenta en la Figura 13), BBs predeterminados y relaciones de aspecto. Esta red ha sido entrenada con PASCAL, COCO e ILSVRC [64].

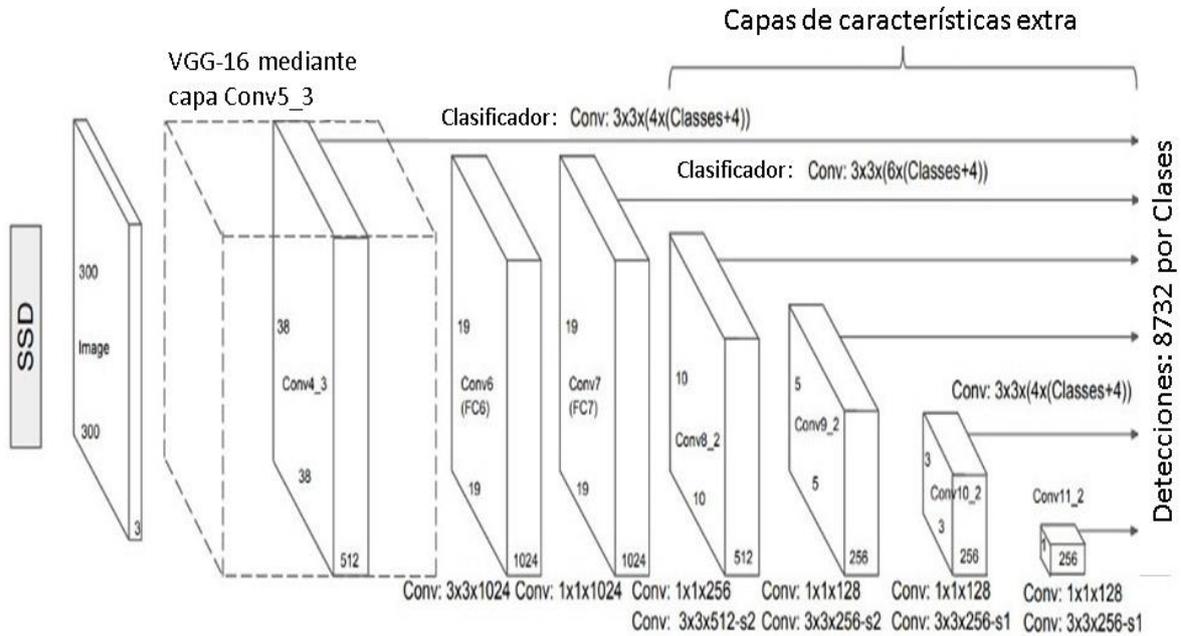


Figura 13: Arquitectura SSD [64]

MobileNet fue creada con el propósito de usarse en aplicaciones de visión integral y en tecnología móvil. Se basa en convoluciones separables en profundidad, excepto por la primera capa que es una convolución completa [65], la Figura 14 muestra la arquitectura de esta red. El entrenamiento de Mobilenet se la realizó en TensorFlow usando RMSprop con el método de descenso de gradiente, se usaron pocas técnicas de regularización y aumento de datos (data augmentation). MobileNet consta de 28 capas de profundidad con 4.2 millones de parámetros, este es el más pequeño de los modelos disponibles, además utiliza el multiplicador de ancho y de resolución para hacerlo mucho más pequeño [65].

	Tipo / paso	Forma de filtro	Tamaño de la entrada
	Conv / s2	3 x 3 x 3 x 32	224 x 224 x 3
	Conv dw / s1	3 x 3 x 32 dw	112 x 112 x 32
	Conv / s1	1 x 1 x 32 x 64	112 x 112 x 32
	Conv dw / s2	3 x 3 x 64 x dw	112 x 112 x 64
	Conv / s1	1 x 1 x 64 x 128	56 x 56 x 64
	Conv dw / s1	3 x 3 x 128 dw	56 x 56 x 128
	Conv / s1	1 x 1 x 128 x 128	56 x 56 x 128
	Conv dw / s2	3 x 3 x 128 dw	56 x 56 x 128
	Conv / s1	1 x 1 x 128 x 256	56 x 56 x 128
	Conv dw / s1	3 x 3 x 256 dw	28 x 28 x 256
	Conv / s1	1 x 1 x 256 x 256	28 x 28 x 256
	Conv dw / s2	3 x 3 x 256 dw	28 x 28 x 256
	Conv / s1	1 x 1 x 256 x 512	14 x 14 x 256
5 x	Conv dw / s1	3 x 3 x 512 dw	14 x 14 x 512
	Conv / s1	1 x 1 x 512 x 512	14 x 14 x 512
	Conv dw / s2	3 x 3 x 512 dw	14 x 14 x 512
	Conv / s1	1 x 1 x 512 x 1024	7 x 7 x 512
	Conv dw / s2	3 x 3 x 1024 dw	7 x 7 x 1024
	Conv / s1	1 x 1 x 1024 x 1024	7 x 7 x 1024
	AvgPool / s1	Pool 7 x 7	7 x 7 x 1024
	FC / s1	1024 X 1000	1 x 1 x 1024
	Softmax / s1	Clasificador	1 x 1 x 1000

Figura 14: Arquitectura MobileNet [65]

Al juntar SSD y MobileNet forman una potente red neuronal profunda, por lo que se deben descartar las 3 últimas capas de MobileNet que son Avg Pool / s1, FC / s1 y Softmax, en cambio en SSD se usa como un extractor de características. Se conectan las capas 12 y 14 de MobileNet a SSD, es decir; se conecta la capa separable con filtro 1 x 1 x 512 x 512 al mapa de características de SSD de profundidad 512 y se conecta la última capa de SSD con el mapa de características de 1024, se usó parámetros con pesos de 0 a 1 en el número de canales, de la misma forma se modificaron los canales de los encabezados de regresión y clasificación de SSD [63], como se presenta en la Figura 15.

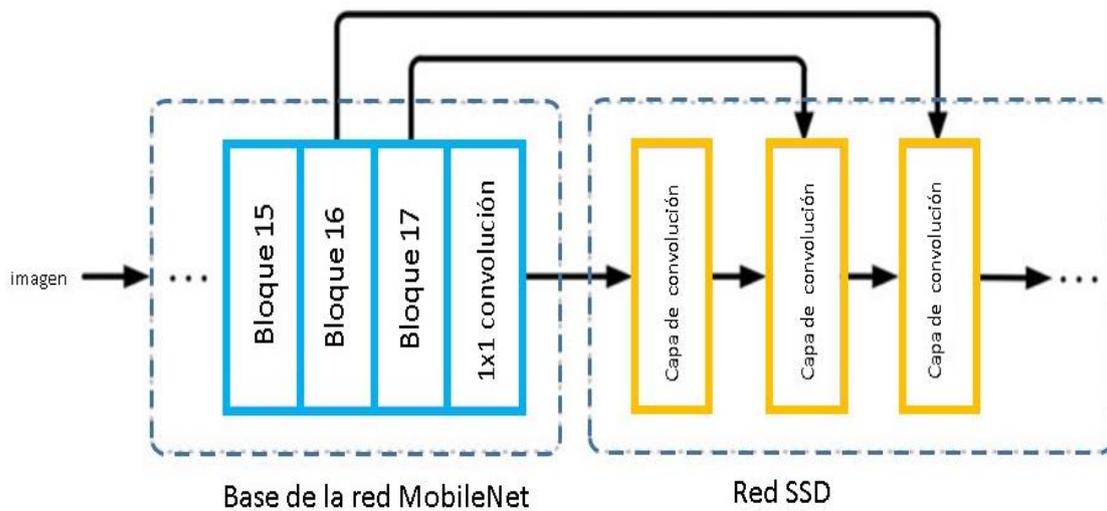


Figura 15: Arquitectura SSD-MobileNet [63]

## 2.6. SEGUIMIENTO DE OBJETOS

El seguimiento (tracking) de objetos está representado por un BB el objeto de interés, se estiman los BBs en cada fotograma del video, es decir; el seguimiento de objetos es la búsqueda de la ubicación de BBs de uno o varios objetos con la tarea de la actualizarlos en los fotogramas del video de entrada. Ya que los algoritmos de detección tardan en su ejecución, tienden a dominar los algoritmos de seguimiento por su rapidez, además que estos muestran buenos resultados al fallar los algoritmos de detección, da la impresión de que es omnipresente y tienen muchas aplicaciones como en la vigilancia [25].

Existen algoritmos de seguimiento de objetos múltiples y únicos, hay varios desafíos en este tipo de algoritmos como lo son las deformaciones del objetivo, oclusiones, variación de iluminación, etc. Una de las fortalezas es que al rastrear el objetivo se puede encontrar fácilmente la velocidad y dirección de éste [25], [66]. Los seguidores pueden clasificarse en rastreadores de modelos [67]:

- Generativos: describe la apariencia de un objeto, ayuda a la identificación de los objetos que tan similares son.
- Discriminativos: también conocidos como seguidores por detección, se concentran en los problemas de clasificación de objetos, su principal objetivo es el de obtener la posición actual del objetivo de la posición anterior y separar el fondo discriminativo.

### 2.6.1. FILTRO DE CORRELACIÓN KERNELIZADO (KCF)

Es un seguidor de objetos basado en filtros correlacionales de forma tradicional, usa el método de truco de kernel o núcleo y matrices circulantes, que mejora la velocidad del cálculo. La principal ventaja es que este rastreador puede aprender de un solo fotograma y adaptarse sin ningún inconveniente al siguiente [67].

KCF necesita que se defina una BB para el objetivo en cuestión, es decir; necesita de un modelo de entrenamiento con el parche de la imagen en la posición inicial del objetivo. Con ello se le entrega las coordenadas de la ubicación y tamaño del objeto, para garantizar la resolución y la calidad de los fotogramas KCF cambia el tamaño de la imagen mediante la interpolación binomial, si no es necesario este paso se guardan la ubicación y el tamaño para cálculos futuros. El parche de imagen extraído es más grande de lo que se define al inicio (ventana) [67], [68].

Para realizar cualquier cálculo KCF transforma la imagen a escala de grises, se inicializa un tamaño de la ventana mayor que el tamaño objetivo ya que se rellena la ventana, ésta se encuentra rodeada por filas y columnas de ceros, el borde proporciona espacio para anotaciones o se usa como límite para los filtros de correlaciones [67].

El filtro de correlación es un método que usa imágenes de muestras de entrada que capturan a los objetivos. Durante el seguimiento se estima un parche que contiene la ubicación del objeto en el cuadro anterior, los límites se suavizan aplicando filtro de coseno, debido a que la frecuencia se distorsiona en el paso de la Transformada Rápida de Fourier (FFT) más conocida como mapa de respuesta. Esta se calcula mediante la multiplicación por elementos del filtro y el parche del objeto estimado. La Transformada de Fourier puede considerarse de la siguiente manera [68]:

$$z = a \sin(hx + ky)$$

La Inversa de la Transformada de Fourier (IFFT) se aplica al mapa de respuestas para conseguir un mapa de confianza en el dominio del espacio, la nueva posición se estima mediante la puntuación máxima de confianza. En el resultado la ubicación del objetivo se actualiza extrayendo características y actualizando los filtros de correlación. El filtro de correlación tiene eficiencia gracias a que se reemplaza las convoluciones con multiplicaciones por elementos utilizando la Transformada de Fourier Discreta (DFT). La correlación se realiza de la manera siguiente, tomando en cuenta que  $*$  es la convolución y  $F\{f[n]\}$  es DFT [67], [68].

Convolución:  $x[n] * h[n] = \sum_{k=0}^{\infty} h[k]x[n+k]$ , también puede ser descrita como:  $x[n] * h[n] = F^{-1}\{F\{x[n]\} \cdot F\{h[n]\}\}$

Correlación:  $x[n] \cdot h[n] = \sum_{k=0}^{\infty} h[k]x[n-k]$

KCF utiliza funciones HOG, este extrae características de gradiente de una celda, en este caso un rango de píxeles. La mayor información de una imagen viene dada en los bordes y esquinas, HOG ayuda a esto ya que determina la distribución de gradientes, y el mayor cambio se produce en los bordes y esquinas. Antes de realizar la FFT se debe realizar la creación de ventanas para evitar la distorsión por la convolución al aplicar FFT se debe multiplicar una imagen con una función de ventana de suavizado, lo que reduce gradualmente los valores de píxeles a ceros en los bordes de la imagen, la ventana más popular es la de Hann que tiene forma de onda de coseno con agregado 1 que lo hace siempre positivo [67], [68].

Una vez que el espacio de característica es extraído, KCF utiliza correlación de kernel para obtener la nueva ubicación, este a la vez ayuda a medir la similitud entre los núcleos y  $\alpha$ , donde  $\alpha$  aprende mediante [67]:

$$\hat{\alpha}^* = \frac{\hat{y}}{\hat{k}^{xx} + \lambda}$$

KCF mediante la correlación Gaussiana en el espacio de características de HOG ayuda a identificar la ubicación del objeto.

Para cada nuevo fotograma se extrae un nuevo parche de la imagen y el mismo tamaño de la ventana. Del mismo modo actúa el extractor de características HOG y luego se transforma en el dominio de frecuencia 2D para el análisis de la respuesta. En vez de la correlación de Gauss se aplica la Transformada de Fourier de las características extraídas al parche actual, la matriz circular se usa para acelerar el proceso, una matriz circular se considera de la siguiente manera [68].

Dado un vector  $x \in \mathbb{R}^s$ , cada elemento denotado por  $x_i$ , se puede generar una matriz circulante  $C(x) \in \mathbb{R}^{s \times s}$  como:

$$C(x) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_s \\ x_s & x_1 & x_2 & \cdots & x_{s-1} \\ x_{s-1} & x_s & x_1 & \cdots & x_{s-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix}$$

El parche se usa como la muestra base para el cálculo de la respuesta en el dominio de Fourier, así:  $\hat{f}(z) = (\hat{k}^{\hat{a}z^*}) \odot \hat{\alpha}$ , donde  $\hat{\alpha}$  son los datos que debe aprender el modelo.

Cuando  $\hat{f}(z)$  se vuelve a transformar en el dominio espacial, es decir, IFFT, la transformación máxima es considerada como el movimiento del objeto, la ubicación se almacena, pero aún no se considera como nueva ubicación, con esto se produce la detección del objeto en cuestión. Por lo que se debe realizar el entrenamiento en la nueva ubicación, por ello se toma un parche con el mismo tamaño, pero diferente ubicación, se extraen sus características y se hace la Transformada de Fourier 2D, después se realiza una correlación Gaussiana usando FFT 2D para actualizar el valor de  $\alpha$ , el cual se interpola linealmente usando el  $\alpha$  anterior. Esta nueva posición se actualiza y almacena, creando un BB y la posición actual se usa para la estimación del siguiente fotograma [69].

### 2.6.2. SUMA DE SALIDA MÍNIMA DE ERROR CUADRÁTICO (MOSSE)

Este rastreador produce filtros de correlación estables con la inicialización de un solo fotograma. Se basa en el uso de filtros MOSSE que le hace resistente a variaciones de iluminación, oclusiones, escala, deformaciones no rígidas y escala; puede operar a 669 fotogramas por segundo [70].

Los seguidores que se basan en la correlación de filtros modelan la forma de los objetos mediante filtros entrenados en imágenes de muestra, los más común es inicializarlo a través de una pequeña ventana o BB en el primer fotograma. Después de ello el entrenamiento del seguidor y el filtrado funcionan de forma conjunta, en la salida el valor máximo de correlación de la ubicación indica la nueva posición del objeto en cuestión, para que el seguidor sea rápido se calcula en el dominio de FFT [70].

El seguidor MOOSE para concentrar su atención en el centro del objeto a rastrear, primero se transforman los valores de los píxeles que ayudan a la iluminación de bajo contraste, los valores se normalizan para tener valores de 0 a 1 y la imagen se multiplica con una ventana de coseno que tiende a reducir gradualmente los valores de píxeles cerca de cero [71].

MOOSE es un algoritmo para obtener filtros en base a un conjunto de imágenes de entrenamiento  $f_i$ , y sus respectivos objetivos o salidas  $g_i$ , este tiene un pico compacto en forma Gaussiana 2D centrado en la imagen objetivo de entrenamiento  $f_i$  [70]. El entrenamiento se lo lleva en el dominio de Fourier, puede representarse de la siguiente forma:

$$H_i^* = \frac{G_i}{F_i}$$

Donde  $H$  representa la transformada de Fourier y la división se realiza por elementos.

Para que las entradas de entrenamiento se asignen a las salidas deseadas MOOSE encuentra un filtro  $H$  que minimiza el error cuadrático entre las salidas reales y las deseadas de la convolución, este tiene la forma:

$$\min_{H^*} \sum_i |F_i \odot H^* - G_i|^2$$

Cuando MOOSE se entrena con una sola imagen este da como resultado un filtro exacto, que es el que produce cero de error ya que solo existe una imagen de entrenamiento  $f_i$  y una sola salida  $g_i$ . El denominador de MOOSE es la suma de las energías en más imágenes, por lo que no da como resultado números pequeños y por ende es más estable. la regularización (corrige espectros de baja energía) MOOSE lo realiza agregando un valor  $\epsilon$  a cada elemento en el espectro de energía [70], este puede ser denotado de la siguiente manera:

$$H^* = \frac{1}{N} \sum_i \frac{G_i \odot F_i^*}{F_i \odot F_i^* + \epsilon}$$

Durante el rastreo de un objeto este puede sufrir diferentes deformaciones, cambios de iluminación, escala, rotación, por estas razones los filtros se deben adaptar de una forma rápida para mantener el seguimiento. Para esto se usa el promedio móvil que se calcula de la siguiente manera en MOOSE:

$$H_i^* = \frac{A_i}{B_i}$$

$$A_i = \eta G_i \odot F_i^* + (1 - \eta) A_{i-1}$$

$$B_i = \eta F_i \odot F_i^* + (1 - \eta) B_{i-1}$$

Donde  $\eta$  es la tasa de aprendizaje, con el valor de 0.125 [70] se permite que el filtro se adapte rápidamente a los cambios de apariencia mientras tiene un filtro robusto. La ventaja al usar este rastreador es que en las oclusiones se detecta en función de la relación entre el pico y la onda lo que permite al seguidor hacer una pausa y reanudar donde reaparece el objeto.

### 2.6.3. MEDIANFLOW

Este tipo de seguidor tiene su principio en el error Avance - Retroceso (Forward-Backward o FB), realiza su rastreo hacia adelante y atrás en el tiempo, mide las diferencias entre estas trayectorias, si esta difiere mucho de dicha trayectoria se considera incorrecta [72].

El FB (secuencia avance - retroceso) requiere de una secuencia de imágenes  $S = (I_t, I_{t+1}, \dots, I_{t+k})$ , donde  $X_t$  representa una ubicación puntual en el tiempo  $t$ , dicho punto se rastrea hacia adelante para  $k$  pasos, obteniendo una trayectoria resultante:

$$T_f^k = (X_t, X_{t+1}, \dots, X_{t+k})$$

Donde  $f$  quiere decir adelante y  $k$  indica la longitud o los pasos que debe dar.

Para encontrar la confiabilidad de la trayectoria  $T_f^k$  dada la secuencia de imágenes  $S$ , se debe realizar la trayectoria de validación, es decir; el rastreo hacia atrás hasta el primer fotograma y se obtiene:

$$T_b^k = (\hat{X}_t, \hat{X}_{t+1}, \dots, \hat{X}_{t+k})$$

Donde  $\hat{X}_{t+k} = X_{t+k}$ .

El error FB es la distancia entre  $T_f^k$  y  $T_b^k$ , por lo que puede representarse:  $FB(T_f^k | S) = distance(T_f^k, T_b^k)$ , donde se utiliza la distancia euclidiana  $(T_f^k, T_b^k) = \|X_t - \hat{X}_{t-1}\|$ .

La selección de puntos a seguir en el seguimiento es importante y se lo toma en el primer fotograma, estos pueden ocluirse o desaparecer en la secuencia de fotogramas, por lo que el mecanismo de selección de fuerza bruta debe incorporar información de todo el video en el error FB [72], que lleva los siguientes pasos [72]:

- Rastrea cada píxel desde el primer fotograma a lo largo de toda la secuencia de imágenes (video)
- Evalúa las trayectorias por error FB
- Asignar a cada píxel el error de su trayectoria.

Al final la imagen resultante se denomina Mapa de Errores, lo que muestra los puntos que se siguen de manera confiable en todo el video, esta selección se lo realiza mediante el cálculo de umbrales. El rastreador acepta dos imágenes  $I_t, I_{t+1}$  y un BB  $\beta_t$ , donde se genera un BB  $\beta_{t+1}$ . Se inicializan puntos en el BB estos se rastrean por medio de Lucas Kanade, se estiman los puntos de las predicciones asignándole un error a cada uno, se eliminan el 50% de las predicciones que no cumplen, y las demás se usan para la estimación del desplazamiento del BB, todo el proceso anteriormente mencionado pertenece al seguidor MedianFlow. Para el cambio de escala se la define como la mediana de las razones entre los pares de puntos de la distancia entre el punto actual y el anterior [72].

#### 2.6.4. CENTROIDE

Este rastreador toma los centroides para realizar el seguimiento del objeto(s), dicho centroide se calcula de los BBs a través de sumas ponderadas de los puntos del borde del cuadro delimitador del objeto detectado, hace a este seguidor es robusto en la oclusión y desplazamiento ya que usa centroides de BB. Toma tres estrategias en la interacción de los datos con el modelo, estos son: asignación de puntos de características de los datos, con la utilización de un algoritmo de emparejamiento o con una función potencial la cual genera un campo de fuerza y la clasificación difusa de los rasgos representativos de los datos a través del aprendizaje competitivo [28].

El rastreo con centroides permite que el límite del objeto se lo aproxime mediante una curva paramétrica definida como una suma ponderada de funciones básicas  $d_1, \dots, d_n$ , puede representarse como:

$$z(s) = \sum_{k=1}^N z_k d_k(s)$$

Donde  $z(s) \in \mathbb{R}^2$ , y  $s$  es un parámetro definido en la ubicación de un punto  $z$  de la curva y este es un vector de dos dimensiones que combinan la forma del modelo. Como consecuencia del movimiento, la matriz  $z$  puede variar, por lo que se obtiene un modelo dinámico y puede ser representado como la Ecuación 19.

$$x(t) = Ax(t - 1) + w(t)$$

Ecuación 19. Representación dinámica del movimiento

Donde  $x(t) = (X^T(t), X^T(t))^T$ , es un vector que contiene a  $X$  y su derivada,  $A$  es una matriz dinámica y  $w(t)$  es ruido blanco con una distribución Gaussiana, de forma similar se usa la Ecuación 19 para el cálculo de la coordenada  $Y(t)$ .

Durante el rastreo en una secuencia de imágenes, se extrae un conjunto de características que asocia a los puntos del modelo resultantes muestreando el modelo del BB a este proceso se lo conoce como partición difusa, que puede ser denotado de la siguiente manera:

$$\xi_n = \frac{\sum_p p \vartheta_n(p)}{\mu_n}$$

Donde  $\vartheta_n(p)$  es el grado del peso de pertenencia del punto del borde,  $p \in \mathbb{R}^2$  de la  $k$ -ésima muestra del modelo, además  $\mu_n = \sum_p \vartheta_n(p)$ .

Las funciones de ponderación de  $\vartheta_n(p)$  son el corazón del algoritmo para la obtención de un buen rendimiento, el alcance de las funciones de ponderación debe estar acotada para fuerzas de que alcance grandes cantidades, para ello se multiplica por una ventana apropiada, en la que  $|p - z_n| \rightarrow \infty$  tiende a cero. Una función Gaussiana se utiliza para la modificación de los pesos, como lo es:

$$\hat{\vartheta}_n(p) = \vartheta_n(p) \exp\left(-\frac{|z_n - p|^2}{2\theta_\epsilon^2}\right)$$

Donde  $\theta_\epsilon$  es un parámetro de escala.

Los centroides están representados por la Ecuación 20, donde  $C_n = [B(s_n) \ 0]$  y  $v_n(t)$  es ruido blanco con una distribución Gaussiana de  $\mathcal{N}(0, \mu_n^{-1}, I)$ , donde  $I$  es una secuencia de imágenes.

$$\xi(s_n, t) = C_n Z(t) + v_n(t)$$

Ecuación 20. Representación de los centroides

A partir de las imágenes de entrada se puede estimar el vector de estado mediante filtros de Kalman los cuales ayudan a la predicción de dicho vector y la covarianza del error en el tiempo  $t$ , conociendo la secuencia hasta el instante  $t - 1$  y el filtrado de los datos hasta el instante  $t$ . El cálculo de los centroides se lo hace por cada instante de tiempo de forma recursiva, éstos asociados a muestras de curvas  $\hat{z}(s_1), \dots, \hat{z}(s_N)$  lo cual da como resultado la estimación actual. Estas estimaciones se actualizan hasta que los centroides convergen en ubicaciones estables, lo hace en menos de cinco iteraciones [28].

### 2.6.5. ESTIMACIÓN DE ESCALA

Este seguidor de objetos se concentra en la estimación de escala robusta mediante la detección previamente del objeto a rastrear, aprende a partir de filtros de correlación discriminativos por separado, a través de una representación de pirámide de escala, la información acerca de este rastreador fue sacada de la investigación original de [27].

Se utilizan filtros unidimensionales para estimar la escala, bidimensionales para la traducción y tridimensionales para la localización del espacio de escala del objeto a seguir, se representa a una imagen como el espacio de características  $d$ -dimensional, sea  $f$  un parche más conocido como BB de este espacio de características, denotando al número de dimensión de característica como:  $l \in \{1, \dots, d\}$  de  $f$  por  $f^l$ . El propósito es el encontrar un filtro de correlación  $h$  con un filtro  $h^l$  por la dimensión de características, esto se logra mediante la minimización de costo, como se muestra a continuación [27].

$$\varepsilon = \left\| \sum_{l=1}^d h^l * f^l - g \right\|^2 + \lambda \sum_{l=1}^d \|h^l\|^2$$

Donde  $g$  es la respuesta de correlación deseada asociada a la muestra de entrenamiento  $f$ ,  $\lambda \geq 0$  controla el impacto del plazo de regularización, para considerar más de una muestra se la representa de la siguiente manera:

$$H^l = \frac{\bar{G} F^l}{\sum_{k=1}^d F^k F^k + \lambda}$$

El tamaño del filtro se fija en  $M \times N \times S$ , donde son la altura, anchura y número de escalas respectivamente, para hallar el filtro se debe calcular una pirámide la que estima al objetivo con tamaño  $M \times N$ , seguido de esto va el entrenamiento de la muestra  $f$  establecida en un cubo rectangular de la pirámide de características, dicho cubo tiene la dimensión de  $M \times N \times S$ , además se encuentra en el centro de la ubicación y escala estimada. Al final se usa el filtro de seguimiento y para ubicar el objetivo actual se extrae un cubo  $z$  de dimensión  $M \times N \times S$  de la pirámide de características, las puntuaciones de correlación se la calculan con:

$$y = \mathfrak{F}^{-1} \left\{ \frac{\sum_{l=1}^d \bar{A}^l Z^l}{B + \lambda} \right\}$$

Donde  $y$  calcula la nueva ubicación y escala objetivo mediante su puntuación máxima.

### 2.6.6. DESBALANCE DE DATOS

Dentro de los problemas de clasificación, suele presentarse el desequilibrio de clases en el conjunto de datos, cuando existen más instancias de algunas clases que otras (existen muestras significativas menores a otras) [73]. Es decir, el desequilibrio sucede cuando la probabilidad real de una clase es difícil de registrar [74].

Este problema se da comúnmente porque las muestras mayoritarias son abundantes, por lo contrario las muestras de interés son raras o escasas ya sea por costo, dificultad o privacidad en la recopilación [75]. Por ejemplo las transacciones fraudulentas, clasificación de proteínas, detección de instrucciones, eventos anómalos, gestión de riesgos son aplicación en las que se podría encontrar desequilibrio de clases [73], [75].

Existe algunas soluciones o técnicas para remediar el inconveniente de desbalance de clases a nivel de datos, por ejemplo remuestreo o submuestreo aleatorio con reemplazo, sobre muestreo o submuestreo dirigido y combinación de las técnicas anteriores [73]. Se

debe tomar en cuenta que una eliminación aleatoria de muestras no es una buena opción ya que se pierden datos importantes.

Además al realizar el submuestreo de la clase mayoritaria aumenta su varianza de la estimación, por ello [74] recomienda realizar un sobre muestreo de la clase minoritaria, aunque este reduzca levemente la precisión del modelo de la clase mayoritaria.

El sobre muestreo es la duplicación o aumento de las muestras de la clase minoritaria, y el submuestreo es la eliminación aleatoria de muestras de la clase mayoritaria [76], [77].

### 2.6.7. MÉTODO DE BOOTSTRAP

Fue introducido en 1979 como un método de estimación del error estándar especialmente aplicado en las computadoras [78]. Es una técnica estadística de remuestreo o sobre muestreo, en la que se estiman cantidades de una población de datos o conjuntos de datos pequeños. El método de bootstrap depende del conocimiento de lo que es una muestra bootstrap.

Una muestra bootstrap, es una muestra aleatoria de datos con reemplazo, es decir, que después de seleccionar un punto de datos para el subconjunto, ésta aún sigue disponible para ser seleccionada. El tamaño de la muestra bootstrap es igual al del conjunto de datos origen. Las muestras que no se han seleccionado se las conocen como “fuera de la bolsa”, para una iteración de sobre muestreo bootstrap, se hace un modelo sobre las muestras seleccionadas para predecir las muestras que están fuera de la bolsa [79].

Lo referente a este método la mayor parte está tomada de [78]. Una muestra bootstrap se define como una muestra aleatoria de tamaño  $n$  que se extrae de  $F$ , donde  $F$  es la distribución de probabilidad de cada uno de los valores observados  $x_i, i = 1, 2, \dots, n$ . Se dice que:

$$X^* = (x_1^*, x_2^*, \dots, x_n^*)$$

$$\hat{F} \rightarrow (x_1^*, x_2^*, \dots, x_n^*)$$

Donde  $X^*$  es un conjunto de datos remuestreado.

En otras palabras  $X^*$  es una replicación bootstrap de  $\hat{\theta}$ ,

$$\hat{\theta}^* = s(X^*)$$

Donde la  $s(X^*)$  es la cantidad resultado de aplicar la función  $s(\cdot)$  a  $X^*$  que se aplicó a  $X$ . Computacionalmente el método bootstrap se lo hace seleccionando números enteros

aleatorios  $i_1, i_2, \dots, i_n$  que pueden tener el valor entre 1 y  $n$  con probabilidad  $1/n$ . La muestra de inicio o arranque consta de los miembros de  $X$ .

$$x_1^* = x_{i_1}, x_2^* = x_{i_2}, \dots, x_n^* = x_{i_n}$$

En el Algoritmo 2, se presenta los pasos que sigue el método bootstrap, donde  $B$  representa al número de muestras bootstrap utilizadas.

Algoritmo 2: Método bootstrap [78].

Algoritmo bootstrap para estimar los errores estándar.

1. Seleccionar  $B$  muestras bootstrap independientes  $x^{*1}, x^{*2}, \dots, x^{*B}$ , de las cuales cada uno consta de  $n$  valores de datos extraídos con reemplazo ( $F \rightarrow (x_1^*, x_2^*, \dots, x_n^*)$ ) o ( $x_1^* = x_{i_1}, x_2^* = x_{i_2}, \dots, x_n^* = x_{i_n}$ ).
2. Evaluar, la replicación bootstrap correspondiente a cada muestra bootstrap,  $\hat{\theta}^*(b) = s(X^{*b})$ , donde  $b = 1, 2, \dots, B$ .
3. Estimar el error estándar  $se_F(\hat{\theta})$  por la desviación estándar muestral de  $B$ .

$$\widehat{se}_B = \left\{ \sum_{b=1}^B [\hat{\theta}^*(b) - \hat{\theta}^*(\cdot)]^2 / (B - 1) \right\}^{1/2},$$

$$\text{Donde } \hat{\theta}^*(\cdot) = \sum_{b=1}^B \hat{\theta}^*(b) / B.$$

## **CAPÍTULO 3**

### **3. REVISIÓN DE LITERATURA**

Los incidentes automovilísticos son un problema social muy importante que debe ser tratado, ya que la mayoría dejan consecuencias graves como pérdidas humanas, económicas y enfrentamientos legales. En este sentido los avances tecnológicos de las ciencias de la computación han ayudado a los CCT a detectar incidentes de tránsito que han permitido entregar una intervención y respuesta temprana. Cabe destacar lo informado en [33], donde resalta, que la mayor parte de centros de videovigilancia se basan en la observación humana de varias cámaras, lo que dificulta operar con facilidad y a la vez produce cansancio.

Por lo anteriormente mencionado, esta revisión de literatura tiene como objetivo identificar el estado del arte en la detección de incidentes automovilísticos mediante técnicas de ML como la clasificación.

#### **3.1. METODOLOGÍA**

El esquema de desarrollo de la presente revisión está basada en la metodología de Bárbara Kitchenham [80]. La selección y extracción de información se especifica de la siguiente forma:

- a) Preguntas de investigación
- b) Método de revisión
  - Fuentes y estrategias de búsqueda
  - Cadena de búsqueda
  - Criterios de selección de estudios
  - Extracción de información
- c) Estudios incluidos y excluidos

#### **3.2. PREGUNTAS DE INVESTIGACIÓN**

Las preguntas que se plantearon para la presente revisión de literatura son las siguientes:

- P1: ¿Qué métodos de clasificación han sido utilizados para la detección de incidentes de vehículos en videos de tráfico?
- P2: ¿Qué métodos de detección y seguimiento de objetos han sido aplicados en videos de tráfico?
- P3: ¿Qué bases de datos públicas de videos de incidentes de tráfico existen?

- P4: ¿Cuál es la técnica utilizada para evaluar el desempeño del algoritmo?

### **3.3. MÉTODO DE REVISIÓN**

#### **3.3.1. FUENTES Y ESTRATEGIAS DE INVESTIGACIÓN**

En la presente investigación para la recolección de la información se utilizaron las siguientes bases de datos: IEEE, ScienceDirect, Springer Link, Association for Computing Machinery (ACM).

#### **3.3.2. CADENAS DE BÚSQUEDA**

Basada en las preguntas de investigación, las cadenas de búsqueda que se definieron son: (vehicle OR car OR traffic) AND (incident OR accident OR collision) AND (detection OR classification)

En el estudio la búsqueda fue ejecutada con los siguientes criterios de inclusión:

- La fecha de publicación se consideró a partir del año 2014
- Los resultados de la investigación son solo del área de Ciencias e Informática
- Las producciones científicas son estudios primarios (artículos de revistas y de congresos)
- Los estudios deben tener información de importancia para las preguntas de investigación.
- Los estudios científicos deben ser en idioma inglés debido a su relevancia científica.

Los criterios de exclusión del estudio son:

- No disponibles
- Incompletos
- Los que no tengan relevancia a las preguntas de investigación
- Duplicados

#### **3.3.3. CRITERIO DE SELECCIÓN DE ESTUDIOS**

Una vez conseguidos los artículos científicos por medio de las cadenas de búsqueda, la selección de los estudios primarios se basó en considerar los criterios de fondo y forma de cada artículo. Los artículos fueron valorizados por una escala que va del 1 al 3, donde 3 es completo, 2 parcialmente completo y 1 incompleto.

Los criterios de forma y fondo son los que se presentan en la Tabla 2.

Tabla 2: Criterios de forma y fondo

<b>Criterios de forma</b>			
<i>1. El artículo científico consta de: abstract, introducción, método, resultados, conclusión, referencias bibliográficas</i>			
Ítem	Completa	Parcial	Incompleta
Valor	3	2	1
<i>2. Numero de referencias bibliográficas</i>			
Ítem	Mayor a 8 referencias	8 referencias	Menor a 8 referencias
Valor	3	2	1
<b>Criterios de fondo</b>			
<i>1. Metodología reproducible</i>			
Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3	2	1
<i>2. Detalle del modelo propuesto</i>			
Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3	2	1
<i>3. Tipo de base de datos usada</i>			
Ítem	Menciona	No menciona	
Valor	1	0	

La mayor calificación que un artículo científico puede obtener es 13, por lo tanto, se selecciona si cumple con una calificación mayor al 80% de la calificación máxima, en este caso mayor a 10.

### **3.3.4. EXTRACCIÓN DE INFORMACIÓN**

Para cada estudio científico seleccionado, se extrajo al menos uno de los siguientes elementos mencionados a continuación:

- Métodos de clasificación de incidentes automovilísticos con técnicas de aprendizaje automático
- Resultados
- Conclusiones relevantes

### 3.4. ESTUDIOS INCLUIDOS Y EXCLUIDOS

En el primer proceso de extracción, se obtuvieron 34 estudios. Después aplicando los criterios de fondo y forma 28 artículos relevantes son los que se consiguieron, tal como se muestra en la Tabla 3 su distribución, la calificación con la aplicación a cada uno de los criterios se encuentran en el Anexo 1.

Tabla 3: Estudios de investigación incluidos y excluidos

Fuente	Resultados	
	Primer proceso de extracción	Aplicando criterios de fondo y forma
IEEE	29	23
ScienceDirect	1	1
Springer Link	3	3
ACM	1	1
Total	34	28

En Figura 16 se observa que el mayor número de artículos recolectados y seleccionados pertenecen a la IEEE.

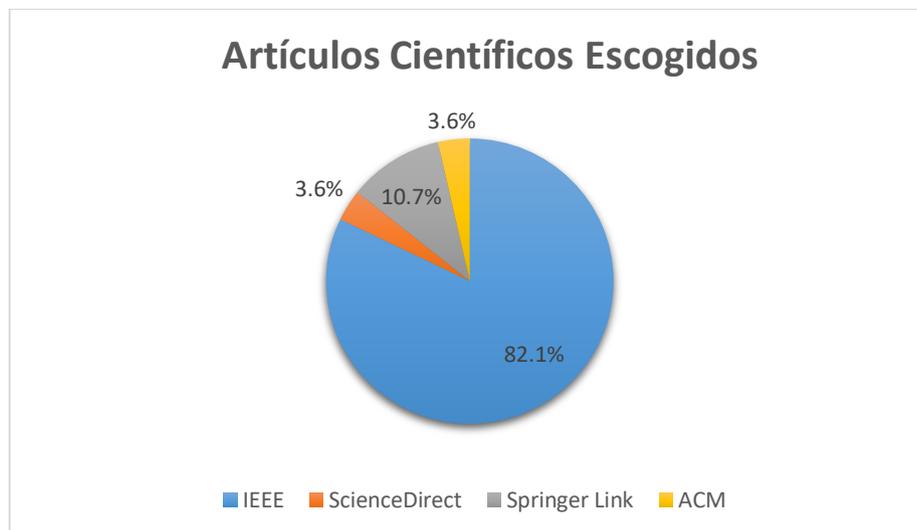


Figura 16: Artículos seleccionados

### 3.5. RESULTADOS OBTENIDOS

De los estudios seleccionados, se encontró evidencia relevante para responder las preguntas de investigación anteriormente planteadas, como se describe a continuación.

**P1: ¿Qué métodos de clasificación han sido utilizados para la detección de incidentes de vehículos en videos de tráfico?**

En la Figura 17, se muestran los clasificadores que usan los trabajos de investigación escogidos. La solución más común es la creación de una función la cual utiliza un umbral para determinar si existe un incidente, le sigue el SVM y ELM como clasificadores supervisados y como no supervisado está el de k-Means.



Figura 17: Clasificadores utilizados en investigaciones escogidas

En la Tabla 4 se muestran los modelos usados de cada una de las investigaciones incluidas, valor de la exactitud y una descripción. Cabe recalcar que en algunos de ellos no se evalúa el desempeño del algoritmo por su exactitud, más bien usan otras métricas que son muy comunes, tal es el caso de: curva ROC, AUC ROC, precisión, sensibilidad, etc.

Se hallan propuestas interesantes como el entrenamiento de clasificadores con características como la orientación y velocidad de los objetos [81], o en redes neuronales probabilísticas se usa la distancia entre vehículos y velocidad [82]. El SVM es el clasificador más utilizado cuando de entrenamiento supervisado se trata, además de una exactitud de 89.87% como la más alta [83].

Tabla 4: Exactitud de clasificación del incidente

Código de referencia	Modelo de clasificación	Exactitud	Descripción
RL01 [84]	Función	-	Toma tres parámetros para detectar un accidente vehicular (aceleración, dirección, cambios de posición), se realiza la suma de las funciones de estado de los parámetros y si ésta sobrepasa un umbral (si es muy grande es 1 de otra forma 0) este se considera como un accidente. No tiene exactitud, se valúa en un video de 347 fotogramas
RL02 [85]	Función	-	Usa una cámara de UAV donde el reconocimiento de un accidente se lo hace por las coordenadas de GPS enviadas con anterioridad, para que el UAV se dirija a reconocer la escena mediante un sensor ultrasónico, un mapa 3D del automóvil es creado por lo que el operador del UAV debe decidir acerca del accidente ocurrido.
RL03 [86]	SVM	81%	El clasificador SVM con kernel RBF, el aprendizaje se lo hace por volúmenes de espacio temporales de la intensidad de los píxeles sin procesar, de videos de incidentes de la ciudad de Hyderabad utiliza autoencoders para la eliminación de ruido y así extraer una representación profunda
RL04 [82]	PPNN	85%	Una Red Neuronal Probabilística de Parzen (Parzen Probabilistic Neural Network o PNN) con núcleo de Ventana de Parzen, fue entrenado por medio de características como velocidad y distancia entre vehículos.
RL05 [87]	Función	-	No se usa ningún método de clasificación se evalúa en videos para obtener los vectores de

			desplazamiento que deben ser monitoreados para su posterior análisis.
RL06 [1]	Función	-	Utilizan una función de costo con la que determina las colisiones de los vehículos, utiliza la exactitud de verdaderos positivos detectados como lo es de 82% y de falsos positivos de 10%.
RL07 [83]	ELM	89.87%	Usa Máquina de Aprendizaje Extremo (Extreme Learning Machine o ELM) para su aprendizaje y lo evalúa utilizando características de nivel bajo de videos de cámaras de seguridad, con distintos fotogramas no etiquetados en una ventana de tiempo deslizante.
RL08 [88]	TTC	-	Una cámara colocada en el vehículo, para calcular una colisión potencial mediante información de movimiento del vehículo, ya que se divide al fotograma en líneas horizontales, verticales y de profundidad con otros vehículos, se llega a determinar una posible colisión por medio del cálculo de una función denominada Time To Collision (TTC) la cual da como resultado el tiempo del flujo de los objetos en movimiento y si esta llegara a ser cero se la toma como potencial colisión, de otra forma toma valores como atención o seguro.
RL09 [2]	Función	-	Usa la tecnología de reconocimiento binocular, con el fin de extraer las líneas de la carretera desde las imágenes tomadas por las dos cámaras para determinar las áreas que pueden ser conducidas, después el vehículo objetivo es detectado en el área transitable. La distancia entre el vehículo controlado y el objetivo es calculado por el principio de alcance binocular que provee de la información del ambiente para el sistema de alerta, el cual halla la distancia

			segura mediante la suma de las distancia recorrida por el control del vehículo durante la fase de reacción, margen de seguridad, recorrido por el vehículo objetivo, tiene una tasa de error promedio de 8.44.
RL10 [33]	Función	-	Propone la detección de accidentes en la carretera mediante modelación de las interacciones de los objetos, como el Campo de Interacción de Movimiento (Motion Interaction Field o MIF) a través de las propiedades simétricas, se obtiene en su métrica Área Bajo la Curva ROC (AUC ROC) 89.50%
RL11 [89]	SVM	73.33%	Es un sistema en tiempo real que utiliza una cámara trasera de un automóvil para alertar de colisiones en retroceso, obteniendo como un costo computacional de 0.63 segundos en cada fotograma, usa un SVM que es entrenado por los vectores de características del movimiento del flujo óptico de los objetos.
RL12 [90]	SVM	-	Propone la detección de accidentes de tránsito en varios niveles mediante la detección, seguimiento y luego la verificación de las áreas afectadas de los vehículos para decidir su gravedad, usa un clasificador SVM entrenado con las características de Histogramas de Gradientes Orientados (Histogram of Oriented Gradients o HOG), de ello se obtiene en su evaluación una pérdida del 16% en la tasa de aciertos, pero tienen 0.7% en falsas alarmas.
RL13 [18]	NAD-DBSCAN	68.70%	Detecta la anomalía mediante un enfoque no supervisado a través de la Agrupación Espacial de Aplicaciones con Ruido Basada en la Densidad de Detección de Anomalías Novedosa (Novel Anomaly Detection-Density Based Spatial Clustering of Applications with

			Noise o NAD-DBSCAN), agrupa las trayectorias de los objetos en movimiento con características como: área, centroide y ángulo, además el principio de que una trayectoria es anómala si esta no encaja con el modelo entrenado.
RL14 [81]	k-Means	-	La detección de eventos fuera de un tráfico normal se lo hace mediante clusterización del comportamiento con k-Means y GMM para abstraer el fondo de los fotogramas, no se presenta un valor para su evaluación ya que esta plantilla presentada es todo un reto.
RL15 [91]	Función	-	Para distinguir si un video tiene neblina o nieve se lo hace con el clasificador de Biltzman, para la detección de las colisiones se lo hace con la siguiente regresión: $collision = 0.35472snowy + 0.29486foggy + 0.0373rainy + 0.00723sunny$ Obtiene como desviación estándar un promedio de 86.5%, y de tasa de aprendizaje 96.5% (para la detección del clima).
RL16 [26]	Función	-	No usa ningún tipo de clasificador, su objetivo es el de detectar bicicletas desde cámaras de 360 grados ya sean por cámaras montadas en cascos y en UAVs. Obtiene como exactitud de seguimiento excelente a MOSSE, KCF.
RL17 [92]	ELM		Para la recolección de las características se utiliza el aprendizaje profundo con: cuatro capas convolucionales y al final con zancada y un Softmax, después de ello se usa una Mezcla de Expertos (Expert Mixture o ME) con clasificadores Máquina de Aprendizaje Extremo (Extreme Learning Machine o ELM) con kernel base, Kernel ELM, On-line Sequential SELM, y

			Constrine ELM, obtiene 92.73% en la precisión de clasificación.
RL18 [93]	Clasificador propio (clasificador no supervisado)	73.19%	Propone un algoritmo que detecta anomalías de objetos en movimiento como: oclusiones, un objeto entrando o saliendo de un fotograma, alteraciones de la velocidad, colisiones. Usa para la segmentación de objetos GMM, Reconstrucción Morfológica luego aplica una operación AND y finalmente una OR. Se conoce que existe alguna anomalía mediante una función la cual tiene por umbral que depende de las secuencias de los videos en este caso solo se realiza para videos normales, pero si pasara algo anómalo este tendría un valor no logarítmico en el error de mínimos cuadrados, su exactitud es de 73.19%, sensibilidad 80.77%, y especificidad de 71.66%.
RL19 [94]	Función	-	Se usan dos cámaras dentro del automóvil además de tarjetas de video nvidia Jetson TX2, toma datos del Sistema interno de Navegación (INS) y de la Unidad de Medida Inercial (IMU), las cámaras se calibran por el método de Zhang. Se centran en la evasión de obstáculos, su promedio de correlación en diez grabaciones de cuatro minutos es 31%. No se usa ningún clasificador.
RL20 [95]	LGMD	-	Propone el uso del Detector de Movimiento Gigante Lobular (Lobula Giant Movement Detector o LGMD), el que permite percibir señales fuera de lo común contra un fondo dinámico de rejillas.
RL21 [96]	Función	-	Propone un algoritmo que se basa en la técnica del modelado de flujo del tráfico en movimiento, la detección de accidentes lo hace mediante un

			método heurístico estadístico, mediante el flujo de tránsito normal. Obtiene F1 0.19, precisión 68% y sensibilidad 91%. No usa ningún clasificador para realizar el aprendizaje.
RL22 [97]	pLSA, LDA, STC, FSTM	-	Utiliza modelos temáticos para la detección de accidentes, por medio de los patrones de movimiento de los objetos aprendidos, los cuales son adquiridos mediante la asignación de una palabra que más tarde sirven para ser clusterizadas, y así aplicar Análisis Semántico Latente Probabilístico (Probabilistic Latent Semantic Analysis o pLSA), Asignación de Dirichlet Latente (Latent Dirichlet Asignation o LDA), Codificación Tópica Escasa (Sparse Topical Coding o STC) y Modelos de Temas Completamente Dispersos (Fully Sparse Topics Models o FSTM) para la detección de accidentes de tráfico. Obtiene como AUC ROC más alto de 0.89 perteneciente a STC.
RL23 [98]	LCU		Mediante la codificación del video en Codificación de Video de Alta Eficiencia (High Efficient Video Coding o HEVC), se puede predecir con poco costo computacional anomalías, a través de vectores de movimiento. Se detectaron todos los accidentes, pero tubo complicaciones en videos normales con un 0.1%. Una anomalía se la detecta si las Unidades de Código Largas (Large Code Units o LCU) sobre pasan un umbral.
RL24 [99]	Aproximación de Matrices	-	Para detectar anomalías en las vías, usa un procedimiento de aprendizaje adaptativo mediante desviaciones de movimiento, con la aproximación de matrices [100], en cuatro escenas en sus detecciones son de: 149, 185, 158, 155 milisegundos sobre fotograma respectivamente.

RL25 [101]	SVM	-	Propone la detección de incidentes de tráfico mediante el análisis de características de distribución del tráfico, los estados encontrados se juzgan mediante identificación difusa, se entrena un clasificador SVM para detectar el incidente, obtiene en sensibilidad 96.60%, tasa de falsos 0.72% y de F1 score 97.93%. El algoritmo creado se evaluó en un segmento de video sobre tráfico extraído del sitio web NextGeneration.
RL26 [102]	Función	-	Propone la detección de accidentes de tránsito mediante la Hidrodinámica de Partículas Suavizadas (Smoothed Particles Hydrodynamic o SPH), utilizando la medida de entropía para aislar partículas que contribuyan a la detección, estas se adjuntan a SPH y se lo detecta mediante el cambio de impulso de la escena. Se obtiene como mejor F1 score en la secuencia 14 con 0.92 de 20 escenas.
RL27 [103]	SVM	-	Plantea un modelo por medio de dispersión conjunta el cual involucra múltiples objetos, acompañado de un algoritmo voraz, por último se introduce la linealidad para que exista una separabilidad de clases. Se crea un diccionario de entrenamiento de eventos y objetos diferentes, esto hecho en un SVM con un núcleo propio. En su curva ROC obtiene un valor de 0.9.
RL28 [104]	k-Prototipe, LOF	-	Expone su algoritmo mediante el aprendizaje no supervisado de las características como: el conjunto de secuencias periódicas y de enlaces, se utiliza el algoritmo k-Prototipe para agrupar características históricas, luego se caracteriza Factor de Parte Aislada Local (Local Outlier Factor o LOF) y se calcula el umbral, si este se sobrepasa se tiene un incidente.

**P2: ¿Qué métodos de detección y seguimiento de objetos han sido aplicados en videos de tráfico?**

Se utilizan diversos métodos de detección de vehículos. Algunos de ellos son el GMM, el cual substraee el fondo de los fotogramas y Cascada Haar en el que se debe realizar un entrenamiento previo.

De la misma forma existen varias técnicas para realizar el seguimiento de objetos en videos de tráfico, por ejemplo: KCF, MedianFlow, MOSSE, centroide, en algunos trabajos se utilizan modificaciones de estos, tal es el caso de [93] que fusiona el algoritmo del vecino más cercano con la técnica del centroide del cuadro delimitador del objeto. También existen investigaciones que utilizan el mismo tipo de técnica para la detección y para el seguimiento, ya que toman en consideración el movimiento de los objetos que interactúan.

Todas estas tecnologías tienen un fin común que es el de obtener características, ya sea velocidad, dirección, etc. Con el fin de usarlos para el aprendizaje de un clasificador o para obtener un umbral y así detectar si ocurrió un incidente de tránsito.

En algunos artículos científicos seleccionados no se realiza ninguna detección y seguimiento, ya que estos trabajan solamente con fotografías de incidentes de tránsito en un momento dado, a su vez otros artículos utilizan datos recolectados de sensores del vehículo como: GPS y otros sensores. En la Tabla 5 se muestran los detectores y seguidores de objetos que interactúan en el tránsito, que usan los trabajos de investigación escogidos.

Tabla 5: Técnicas de detección y seguimiento de vehículos

Código de referencia	Detección de automóvil	Seguimiento	Descripción
RL01 [84]	GMM	Desplazamiento Medio	Toma parámetros como: dirección, aceleración y cambios de posición.
RL02 [85]	Haar Cascade	SAR	Radar de Apertura Sintética (Synthetic Aperture Radar o SAR) funciona mediante radares colocados en grandes altitudes que pueden rastrear vehículos independientemente de su instrumentación,

			en este caso los radares se encuentran en un UAV, en la detección su exactitud es de 95.81%.
RL03 [86]	STVVs	Autoencoders Profundos	Se utiliza para la detección de objetos en movimiento la generación de Volúmenes Espacio-Temporales de Video (Spatio-Temporal Video Volumes o STVVs) por medio de Flujo Óptico (Optical Flow) y para el seguimiento un Aprendizaje de Representación Profunda usando Autoencoders Profundos.
RL04 [82]	Horn Schunck	Húngaro	El algoritmo de Horn Schunck obtiene los vectores de flujo óptico y si cada vector es más grande que la media se confirma movimiento, en el seguimiento se usa el algoritmo Húngaro calcula la velocidad aproximada del vehículo y crea un nuevo identificador para un nuevo ingreso.
RL05 [87]	SURF	SSD, SAD, NCC	Para encontrar los vehículos se lo hace mediante detectores de características como Características Robustas Aceleradas (Speeded Up Robust Features o SURF) y para su seguimiento se hace con técnicas de emparejamiento basadas en áreas como: Diferencia de Suma de Cuadrados (Sum of Square Differences o SSD), Suma de Diferencias Absolutas (Sum of Absolute Differences o SAD), Correlación Cruzada Normalizada (Normalizaed Cross Correlation o NCC).
RL06 [1]	ACF	ACF	Característica de Canal Agregado (Aggregated Channel Feature o ACF) detectan los objetos y su movimiento, en la detección de la colisión se utiliza una función de costo que es la suma de las actividades

			(prominencia, colisión y costo), el costo es cero cuando no existe actividad. El costo de la actividad está definida como la suma de las áreas superpuestas de los objetos detectados.
RL07 [83]	OF-SIFT	OF-SIFT	Las magnitudes de flujo con que luego se usa el descriptor de características Transformación de Entidad Invariante de Escala (Scale Invariant Feature Transform o SIFT) como emparejador, con la diferencia que se construyen los histogramas por acumulación de flujo de orientación, así obteniendo un vector dimensional que hace al descriptor más robusto, a este nuevo algoritmo lo llaman OF-SIFT.
RL08 [88]	Método propio	Método propio	Se usa la captura del movimiento de los objetos mediante la división del fotograma con líneas horizontales, verticales y de profundidad en la vista lateral del vehículo y marcas de carriles adyacentes.
RL09 [2]	Método del valor de gris promedio	Método del valor de gris promedio	La detección de la posición del obstáculo se lo hace por el método del valor de gris promedio.
RL10 [33]	No se usa detección	No se usa seguimiento	No se utiliza ninguna técnica de seguimiento de objetos en este caso propone obtener la información de la dirección y velocidad de los objetos que interactúan en el video a través de [46] esta información es usada para generar MIF, luego filtrarla mediante un núcleo que tiene por referencia la velocidad y se centra en la región de los accidentes ya que esta rompe la simetría de los datos al tener una colisión.
RL11 [89]	SaDPM	No se usa seguimiento	La detección de los objetos del método de mapas de características en múltiples

			escalas (SaDPM) que estima la escala y la ubicación de los objetos en función de la distancia (3 y 6 metros) desde la cámara.
RL12 [90]	Haar Cascade	KCF, Median Flow	Se hace modificación en lo que propone [47] para la detección de automóviles en lo que destaca la reducción del Área de Interés (Region OF Interest o ROI), la selección del rectángulo más pequeño y el uso de detector en Cascada Haar (Haar Cascade) en esa región. En el seguimiento de los automóviles se usan dos: Filtros de Correlación Kernelizados (Kernelized Correlation Filters o KCF) y Flujo Medio (Median Flow) para luego extraer su HOG.
RL13 [18]	EM	Centroide	La sustracción del fondo de los fotogramas mediante GMM para realizar la detección de los vehículos con Maximización Esperada (Expected Maximization o EM), el seguimiento se lo hace con la técnica del cálculo de los centroides de los cuadros delimitadores del vehículo detectado.
RL14 [81]	No se usa detección	No se usa seguimiento	No se presenta ningún tipo de detección y seguimiento de vehículos ya que los datos con los que trabajaron son extensos, entre ellos se presentan datos de GPS, velocidad, aceleración, distancia, etc. De vehículos propios y diferentes.
RL15 [91]	No se usa detección	No se usa seguimiento	No se presenta ningún tipo de detección y seguimiento de vehículos ya que su objetivo es el de detectar el tipo de clima de los videos.
RL16 [26]	CNN	MedianFlow, KCF, MOSSE, TLD, CSRT, MIL, GOTURN	Se usa para la detección de vehículos una Red Neuronal Convolutiva (Convolutional Neural Network o CNN) utiliza el modelo estándar Caffe, para el seguimiento utiliza

			MedianFlow, KCF, MOSSE, Multiple Instance Learning (MIL), TLD, Generic Object tracking Using Regression Networks (GOTURN), Channel and Spatial Reliability Tracker (CSRT)
RL17 [92]	No se usa detección	No se usa seguimiento	El modelo se concentra en la clasificación de accidentes de vehículos en imágenes por lo que no se realizan detección y seguimiento de vehículos.
RL18 [93]	No se usa detección	Vecino más cercano con centroide.	Se propone extraer características como masa y energía cinética (velocidad), la masa se lo hace mediante el punto de fuga en el fotograma RGB con líneas paralelas, luego para detectar los objetos en movimiento se calcula el centroide de cada objeto y se lo compara mediante el cálculo del vecino más cercano, del fotograma actual con el anterior.
RL19 [94]	Flujo optico	No se usa seguimiento	Los obstáculos son detectados por medio de algoritmos de flujo óptico que usa niveles de pirámide para remover estimaciones incorrectas.
RL20 [95]	LGMD	LGMD	Para la detección del movimiento de los objetos se usa LGMD, su entrada son imágenes en escala de grises, la luminancia se captura por los fotoreceptores, estos se transmiten por rutas de encendido y apagado, dentro de Redes Neuronales Parciales (PNN), se ajusta la latencia de excitación local de cada fotograma.
RL21 [96]	Farneback	Farneback	Para la detección de movimiento de los objetos se utiliza el algoritmo de Farneback de flujo óptico, para la extracción de la velocidad y orientación de los píxeles de los fotogramas.

RL22 [97]	Shi y Thomas, Lucas Kanade	Shi y Thomas, Lucas Kanade	Se define un vocabulario para caracterizar la escena, con ello se crea la construcción de los videos. Para la creación de palabras visuales se segmenta la escena en celdas cuadradas (10X10), se usa Shi y Tomas para encontrar puntos clave, se extrae de ello mediante Lucas Kanade la velocidad y posición, estas palabras se acumulan y se crean un histograma de palabras para luego realizar su clusterización.
RL23 [98]	MIC	MIC	Cada fotograma se divide en cuadrados de igual tamaño de forma recursiva, cada unidad representa una predicción que será parte del vector de movimiento calculado por Recuento de Intensidad de Movimiento (Motion Intensity Count o MIC), es decir, el movimiento de los objetos se calculan con MIC.
RL24 [99]	AMVA	AMVAcau	El movimiento de los objetos los detecta mediante la construcción de matrices de movimiento, cada segmento de video pertenece a una escena específica de los cuales se obtienen los vectores de velocidad que forman la matriz de movimiento. El movimiento de los objetos se lo calcula por Amplitud Media del Vector de Movimiento (Average Motion Vector Amplitude o AMVA)
RL25 [101]	GMM	Kalman	Para detectar a los objetos de los videos se lo hace con GMM con tasas de actualización adaptativas, luego para el seguimiento de cada uno de ellos se utiliza filtros de Kalman, estos parámetros se extraen cada 30 segundos.
RL26 [102]	Farneback, TDP	Farneback, TDP	El movimiento de los objetos se lo realiza mediante la extracción de flujo óptico denso

			de Farneback, después un método de difusión térmica (Thermal Diffusion Process o TDP) convierte el movimiento en algo coherente.
RL27 [103]	GMM, B-spline	GMM, B-spline	La detección de movimiento se lo hace mediante GMM y su seguimiento lo hace por el cálculo del centroide, además esta trayectoria es aproximada por B-spline.
RI28 [104]	No se usa detección	No se usa seguimiento	Utiliza como factor la velocidad para la extracción de características, ya que cuenta con ese factor en sus datos, además propone que el incidente pasa en un tiempo $t-1$ , en el enlace $l$ y la carretera esta despejada antes del evento de tráfico.

### **RQ3: ¿Qué bases de datos públicas de videos de incidentes de tráfico existen?**

En cada trabajo de investigación analizado hay un conjunto de datos con el que trabajaron para crear su algoritmo. La gran parte de ellos son videos de cámaras de seguridad colocados en vías concurridas. En el caso de [82], los autores proponen videos de tráfico normal de cámaras recogidas en carreteras, pero para los incidentes de tránsito ellos crean simulaciones mediante el programa Adobe Flash.

Varios trabajos utilizan los videos de YouTube, mientras que otros usan videos proporcionados por empresas privadas. En el análisis realizado se pudo verificar que los trabajos previos han utilizado 16 conjuntos de datos de fuentes privadas y 12 de fuentes públicas, es decir; 57% y 43% respectivamente. Por ejemplo, TRIMARC y NavInfo pertenecen a empresas privadas y no liberan sus datos. En la Figura 18 se presenta la información acerca de lo expuesto anteriormente.

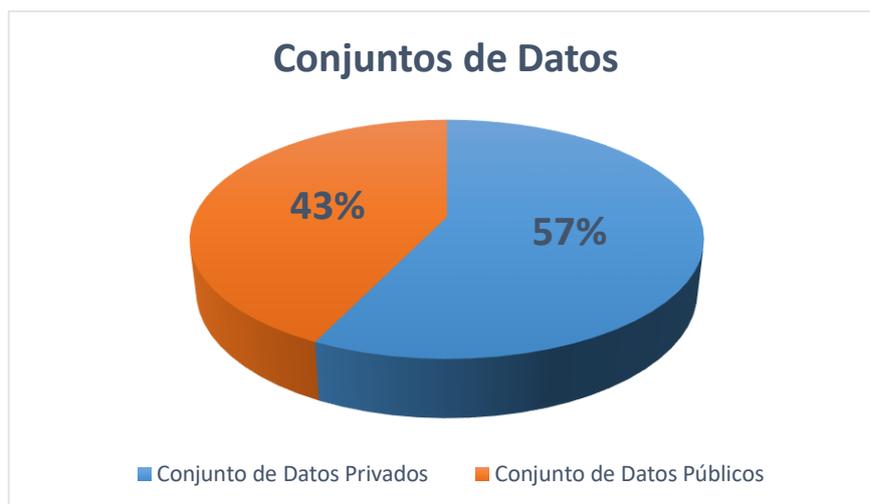


Figura 18: Tipos de conjuntos de datos

De los conjuntos de datos públicos encontrados en los trabajos analizados, existen algunas anomalías con sus enlaces o a su vez las páginas que la proporcionaban dejaron de funcionar. En la Figura 19, se presentan los resultados de los datos que están disponibles para su descarga y revisión.

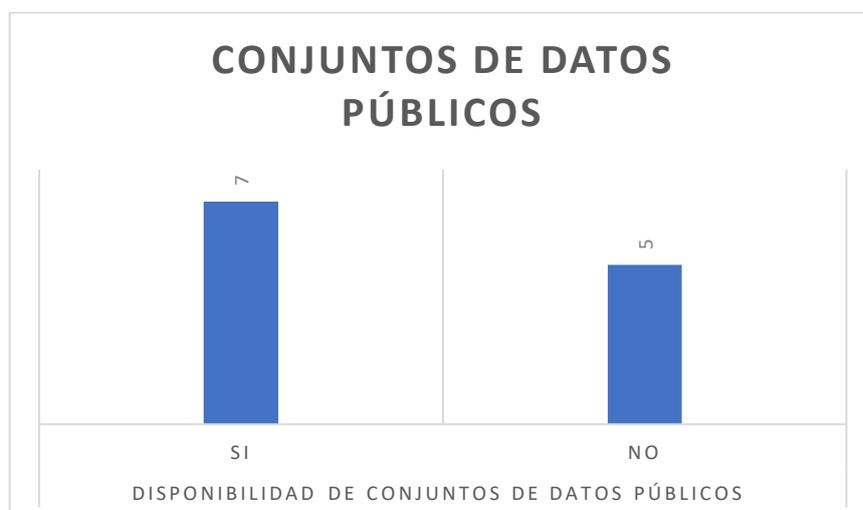


Figura 19: Conjuntos de datos públicos disponibles

Para mayor explicación, la Tabla 6 muestra los diferentes conjuntos de datos utilizados en los trabajos previos.

Tabla 6: Conjuntos de datos encontrados

Código de referencia	Conjunto de datos	Descripción

RL01 [84]	Privado	Explica que el video con el que se evaluó el algoritmo se llama accident.avi con 347 fotogramas.
RL02 [85]	Privado	El modelo de detección de vehículos se lo hace con la base de datos Urbana-Champaign de la Universidad de Illinois, pero no nombra un conjunto de datos sobre incidentes.
RL03 [86]	IITH_Accident Dataset	Trabajaron el conjunto de datos IITH_Accident, el cual tiene videos reales de accidentes proporcionados por CCTVs pertenecientes a la India, los videos son capturados a 30 fotogramas por segundo en distintas horas de día, este tiene 94720 fotogramas para entrenamiento y 33280 para test entre ellos 863 pertenecientes a incidentes.
RL04 [82]	Privado (Videos Reales y creados)	Tiene videos reales del Área Residencial de Bashundara, las cámaras ubicadas en la parte lateral de la vía y videos creados en Adobe Flash para recrear los accidentes, teniendo así un total de 18590 fotogramas, dicho conjunto de datos no está disponible para su revisión.
RL05 [87]	TRIMARC	Tiene videos capturados de la cabina de transporte de Kentucky que no está disponible para países diferentes a Estados Unidos.
RL06 [1]	YouTube-8M, <u>UrbanTracker</u>	Se usaron dos tipos de bases de datos YouTube-8M y UrbanTracker ambos contienen videos de tráfico, algunos con cámaras colocadas en automóviles, motocicletas y cámaras estáticas.
RL07 [83]	YouTube y videos reales	Crearon su propio conjunto de datos con videos de Youtube, tiene 324 ejemplos de entrenamiento que cubren seis tipos de colisiones etiquetando fotogramas de la región de interés y para prueba del modelo utiliza videos de tráfico real con distintos fotogramas sin etiquetas.
RL08 [88]	Privado	Se recogen videos de conducción de cámaras montadas en vehículos mirando hacia delante con fotogramas de 1280 × 720 píxeles, capturados a 30 fotogramas por segundo.
RL09 [2]	Privado	Las pruebas lo realizaron en tiempo real mediante la conducción.

RL10 [33]	Conjunto de datos propio	Usa el conjunto de datos propuestos por [48] el cual facilita un enlace para su descarga, pero no está disponible. <a href="http://www.car.accidents.com/pages/stas.html">http://www.car.accidents.com/pages/stas.html</a>
RL11 [89]	VOC 2009	Crearon un conjunto de datos tomados en diferentes localizaciones y condiciones, estos videos están capturados a 30 fotogramas por segundo, contienen información acerca de peatones caminando por la parte trasera, simulación de colisiones cercana y retroceso de vehículos, el conjunto de datos tiene por nombre VOC 2009.
RL12 [90]	Privado	Usa una cámara web conectada en una carretera, el modelo se lo ejecuta en un servidor y se simula un video en tiempo real con una aplicación de escritorio a un video grabado anteriormente.
RL13 [18]	Narrow traffic dataset	El conjunto de datos Narrow, contiene videos de cámaras de seguridad de accidentes vehiculares, peatonales y enfrentamientos.
RL14 [81]	EuroFOT	Tiene datos de varios sensores de automóviles del grupo Volvo, entre ellos videos, el cual tiene aproximadamente 30 TB, todos estos datos son recolectados de viajes que son más de 170000, entre los grupos de datos están: posición, relación con otros vehículos y el área de descripción, movimiento del vehículo, otras medidas internas.
RL15 [91]	Traffic Accident Surveillance and Analysis System (TASAS)	Son videos colectados de 41 CCTVs de Korea, de aproximadamente 3 kilómetros de distancia, del año 2010 a 2015. Consta de datos como: tiempo, velocidad, flujo y ocupación, información metodológica. Se proporciona el enlace <a href="http://www.its.go.kr/opTraffic/information.jsp">http://www.its.go.kr/opTraffic/information.jsp</a>
RL16 [26]	360-degree videos	Contiene videos de 360 grados, con resolución de 3840X2160, tres canales de colores, con formato MP4. Proporciona el link para su descarga <a href="https://drive.google.com/open?id=1Ybp0G6yWXYCsP06nzEMRJR-exK0DSos8">https://drive.google.com/open?id=1Ybp0G6yWXYCsP06nzEMRJR-exK0DSos8</a> .
RL17 [92]	Conjunto de datos propio	El modelo utiliza uniones de distintos conjuntos de datos de fotos como: dígitos, polución, tumores cerebrales, vehículos,

		accidentes de vehículos de UAVs este último tiene 1225 de accidentes de alto riesgo.
RL18 [93]	Privado	El algoritmo creado se evalúa en cuatro videos reales, el cual tiene 3079 fotogramas con normalidad y 1059 con anomalías.
RL19 [94]	Privado	Usa datos como video, acelerómetro y giroscopio del vehículo. No se nombra un conjunto de datos el cual pueda ser revisado.
RL20 [95]	Privado	Las simulaciones se las hicieron con videos en escala de grises, con resolución de $38 \times 334$ , $540 \times 270$ , $352 \times 288$ , no se muestra el conjunto de datos para su revisión.
RL21 [96]	Privado	Se tiene 40 videos diferentes colectados de YouTube con ejemplos de diferente calidad, orientación y condiciones. Tiene resolución $1280 \times 720$ a 30 fotogramas por segundo.
RL22 [97]	Privado	Su conjunto de datos son videos del centro de control de Teherán, los cuales son tomados de intersecciones, carreteras concurridas, túneles. En los cuales existen fotogramas con accidentes, pero no existe la posibilidad de su revisión ya que es privado.
RL23 [98]	MIT traffic dataset	Contiene 16 videos de tráfico de accidentes y 20 normales, en total suman 120 minutos.
RL24 [99]	Privado	Tiene videos de 15 fotogramas por segundo, con un tamaño de $480 \times 272$ píxeles, está dividido en normales y accidentes.
RL25 [101]	Privado	Los videos tiene una resolución de $640 \times 480$ , las carreteras son de tres carriles, consta de dos escenas con un total de 12 horas.
RL26 [102]	Privado	Son veinte secuencias de videos que se obtienen de YouTube.
RL27 [103]	CAVIAR, Xerox stop sign	Son la unión de dos conjuntos de datos CAVIAR y Xerox stop sign dataset, el primero con 27 videos cortos y el segundo 39.
RI28 [104]	NavInfo Corporation	Videos, número de enlace, hora y velocidad son los datos recolectados de las carreteras de Xian, facilitados por la

		corporación NavInfo, pertenecen al mes de septiembre y octubre 2017.
--	--	--

**RQ4: ¿Cuál es la técnica utilizada para evaluar el desempeño del algoritmo?**

Se halló que la métrica más común es la exactitud de clasificación. Además, se encontraron otras técnicas como la precisión, sensibilidad, especificidad, curva ROC, AUC ROC. En algunos casos, se indican solamente si hubo o no la detección de incidente; en estos casos se optó por colocar un guion medio “-” en la columna Métrica, en la Tabla 7 se listan las métricas utilizadas en los trabajos previos.

Tabla 7: Métricas de desempeño utilizado en trabajos previos

Código de referencia	Métrica
RL01 [84]	-
RL02 [85]	Exactitud
RL03 [86]	Exactitud y AUC
RL04 [82]	Exactitud
RL05 [87]	-
RL06 [1]	Exactitud de verdaderos positivos
RL07 [83]	Exactitud, precisión y sensibilidad
RL08 [88]	Exactitud
RL09 [2]	Tasa de error
RL10 [33]	AUC ROC
RL11 [89]	Exactitud, tiempo
RL12 [90]	Tasa de aciertos, falsas alarmas
RL13 [18]	Exactitud
RL14 [81]	-
RL15 [91]	Desviación estándar, tasa de aprendizaje profundo
RL16 [26]	Exactitud
RL17 [92]	Precisión en detección, clasificación.
RL18 [93]	Exactitud, sensibilidad, especificidad.
RL19 [94]	Promedio de correlación
RL20 [95]	Tasa de picos sobre tiempo(indica el cambio de LGMD)

RL21 [96]	F1, precisión, sensibilidad
RL22 [97]	AUC ROC
RL23 [98]	Porcentaje de error en videos
RL24 [99]	Tiempo de ejecución
RL25 [101]	Sensibilidad, tasa de descubrimiento de falsos, F1 score
RL26 [102]	F1 score
RL27 [103]	Porcentaje de tráfico normal y anormal, curva ROC
RI28 [104]	Gráfica de eventos actuales vs detectados

### 3.6. DISCUSIÓN

La revisión de literatura realizada proporciona una visión general de los métodos actuales de la detección de incidentes de tránsito. En los trabajos previos seleccionados, la utilización de una función propia del autor es el método más utilizado, la cual depende de un límite para determinar la incidencia de los objetos en el video de tráfico. Después, le sigue el modelo basado en SVM y ELM. Cabe destacar que también existen trabajos que usan el aprendizaje no supervisado como el algoritmo de k-Means [81]. Sin embargo, aunque estas técnicas han demostrado resultados alentadores al momento de detectar incidentes, aún existen varios desafíos que permanecen abiertos.

De acuerdo con los modelos existentes en los estudios analizados los resultados obtenidos demuestran que hay una gran variedad en cuanto a la creación de modelos para la detección de incidentes de tránsito, tal es el caso de [1], [84], [86], [82], [88], [33], [91], [93] y [99] coinciden en extraer de los videos, características como la velocidad o en su defecto ésta es tomada por otros sensores (velocímetros, acelerómetros, GPS). Por otro lado [81] en su conjunto de datos tiene muchas características entre ellas la velocidad del auto en cuestión y de otros con los que interactúa.

Además [83] propone usar la densidad de flujo óptico utilizando cada muestra como una ventana fija. Algo semejante ocurre con [86] que coincide en ocupar una ventana de tiempo, pero en este caso es muy larga para la extracción de características, lo que facilita la adaptación del algoritmo de flujo óptico.

En [91] concentra su objetivo en la detección del tipo de clima que se presentan en los videos para luego realizar un análisis con su conjunto de datos y decidir la existencia de algún incidente de tránsito.

Para la detección de incidentes de tránsito por medio de clasificadores destaca [90] y [101], que proponen una plantilla con los siguientes pasos: detección, seguimiento y clasificación (mediante aprendizaje supervisado) del accidente. En [52] se segmenta temporalmente los videos, que tienen la misma longitud de 1 segundo. Además [82], utiliza en su Red Neuronal Probabilística como núcleo, el método de ventana de Parzen que es un método parecido a KNN. Cuando se utilizan descriptores de características como propone [87], se usa una técnica de seguimiento de emparejamiento basada en el área de los vehículos detectados por SURF.

También en los resultados de esta revisión se tomaron consideraciones de acuerdo a las desventajas de los algoritmos creados, por ejemplo: existe una vulnerabilidad en errores de técnicas de segmentación en [1]; en [88] se propone un método en tiempo real para detectar una colisión potencial, la cual necesita de una cámara colocada de forma precisa en el tablero del vehículo para evitar problemas. Según lo propuesto en [94], se sacrifica exactitud por rapidez para poner en uso algoritmos de flujo óptico.

De acuerdo con [95], el umbral para el disparo de alarma es puesto de forma manual para objetos que se acercan o superen al automóvil, esto altera el rendimiento generando falsas alertas. La desventaja de [98], [102] y [103] se presenta cuando existe congestión de vehículos, oclusiones y al encontrar el umbral para determinar si hay anomalía.

En cuanto a los conjuntos de datos no disponibles para su descarga o revisión destacados en [18], [91], [92], [98], [99] y [101], la mayoría de ellos tienen enlaces expirados o que redirigen a páginas con distintos datos. No obstante [104] da a conocer su conjunto de datos, pero es privado. En [83], [96] y [102] se hablan de videos de YouTube con los que evaluaron su algoritmo, pero no se presenta ningún conjunto de datos para su revisión. También existen trabajos como [89] en donde los videos que evalúan su modelo son de una cámara en la parte trasera de un vehículo, estacionario y en movimiento. Por otra parte [26] proporciona videos de 360 grados que no contienen específicamente videos de incidentes de tránsito.

Para evaluar el desempeño de los algoritmos creados la gran mayoría de trabajos utilizan la métrica de la exactitud de clasificación, pero en otros los autores prefieren analizar su modelo por otras métricas. Por ejemplo en [101] se utiliza la sensibilidad, tasa de falsos detectados y calificación F1. En [104] se presenta una gráfica la cual da a conocer los

eventos con los cuales detecta los incidentes, algo semejante ocurre en [86], [33], [97] y [103] donde utilizan la gráfica AUC ROC, como métrica.

Por los resultados obtenidos mediante el análisis de los trabajos de investigación escogidos se puede notar claramente que existen distintas métricas para evaluar el desempeño del algoritmo de detección de incidentes de tránsito al igual que para la creación de modelos de clasificación.

### **3.7. CONCLUSIÓN DEL ESTUDIO DE LITERATURA**

En el presente trabajo, se realizó una revisión sistemática de los métodos de detección de incidentes de tránsito (colisiones de automóviles) y se analizó el estado del arte actual, sus desafíos y retos. Con los resultados obtenidos, se concluye que existen muchas formas para detectar anomalías en videos de tráfico que recaen la mayor parte en métodos de detección basada en región. De igual forma los datos con los que se entrena y evalúa el modelo creado provienen de múltiples fuentes ya sean públicas o privadas. Además, se pudo determinar que las métricas más usadas para medir el desempeño del algoritmo son la exactitud de clasificación, sensibilidad, AUC ROC, precisión y calificación F1.

## CAPÍTULO 4

### 4. METODOLOGÍA

Según [45] y [105], Computer Science (CS) es una ciencia que cumple con los criterios como: proponer una hipótesis y probarlas empíricamente, en informática la mayor parte de los casos se enfocan en el estudio de un programa, en el campo de la IA la CS estudia programas informáticos que realizan tareas en entornos [106].

A la ciencia se la puede considerar como una colección de conocimiento sistemático [107], tomando como núcleo de todas las ciencias a la lógica y la matemática. Según [108] la investigación en la parte académica se le puede considerar como la acción de indagar siguiendo las normas y procedimientos de forma ordenada en un área, con el objetivo de realizar descubrimiento y difundirlos. Al necesitar de flexibilidad y la realización de cambios de forma rápida e integral en una investigación, es imprescindible contar con la metodología de la Investigación para la Acción o Action Research [109].

Este tipo de metodología fue propuesta por Kurt Lewin, con el pasar del tiempo acuñó otros nombres como: investigación participativa, colaborativa o contextual. La investigación para la acción puede considerarse como el proceso en que una o varias personas identifican un problema y hacen algo para resolverlo, hasta estar satisfechos con sus resultados, si no es así vuelven a intentarlo [109]. Lo que lo convierte en una dirección deseable [110].

Lo que separa a esta metodología de las prácticas profesionales generales, es el énfasis en el estudio científico. Además tiene mucha afinidad con el paradigma de la praxis (práctica), siendo este un proceso continuo en el conocimiento, se deriva de la práctica y la práctica informada por el conocimiento [111]. Las herramientas que se pueden usar en esta metodología de investigación son diversas, por ejemplo; se puede llevar un diario de investigación, grabaciones de observaciones, encuestas, entrevistas, recopilación y análisis de documentos y casos de estudio [109].

En el presente trabajo se utilizó la investigación para la acción de tipo tradicional, ya que ha presentado resultados muy favorables para las tecnologías de la información [109]. Este tipo de metodología presenta cuatro pasos [112]:

1. Planificar: se indaga acerca del problema para su comprensión, también se elaboran planes con el fin de dar solución a dicho problema.
2. Actuar: se debe llevar a cabo un solo plan.
3. Observar: se recopilan los diversos resultados obtenidos de diversas formas.

4. Reflexionar y revisión: se deliberan las observaciones conseguidas, y si estas no son las esperadas se sigue con otros planes, repitiendo así de forma cíclica hasta lograr los resultados deseados.

Para dar a conocer los pasos anteriormente expuestos, se presenta la Figura 20.

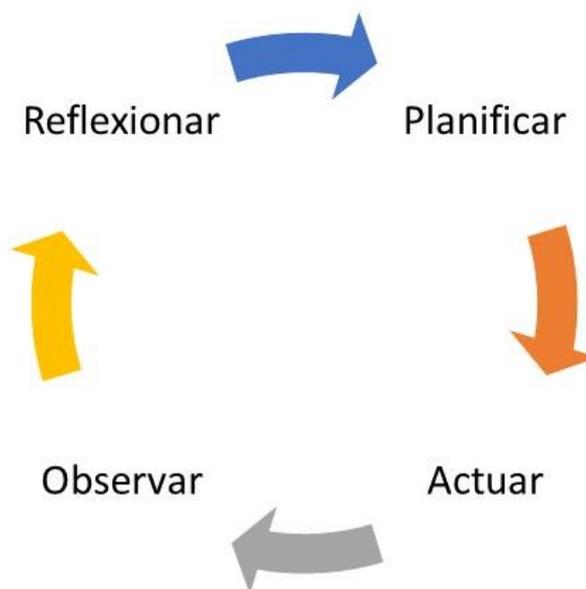


Figura 20: Ciclo Action Research [113]

#### 4.1. DISEÑO DE LA INVESTIGACIÓN PARA LA ACCIÓN

En la siguiente sección, se describe el diseño que se siguió para lograr la obtención de los objetivos y metas propuestas, mediante la metodología de la investigación para la acción.

La metodología consta de cuatro pasos, como lo son: planificación, en esta fase se realiza una ardua investigación para la comprensión del problema, además se presentan los procedimientos tentativos a seguir. En el presente trabajo la indagación y los planes posibles se lo hizo previamente, en el Capítulo 2 y Capítulo 3 correspondientes al marco teórico y revisión de literatura respectivamente.

La fase de actuación corresponde a llevar a cabo un único plan, dicho plan se describe en el presente apartado. Lo referente a la observación, que es el paso donde se recopilan los resultados obtenidos al aplicar un plan, se lo describe en el Capítulo 5 perteneciente a la experimentación.

Finalmente, el paso de reflexión que es aquel donde se deliberan las observaciones encontradas, se lo presenta en el Capítulo 6 correspondiente a las conclusiones.

#### 4.1.1. ACTUACIÓN

La presente investigación tiene como objetivo el crear modelos de clasificación y encontrar el mejor modelo para detectar incidentes automovilísticos (colisiones entre automóviles) en videos de tráfico usando la velocidad estimada de los automóviles detectados, por medio de un conjunto de datos de videos de incidentes y no incidentes automovilísticos. Para lograr lo anteriormente expuesto, se llevaron a cabo tres bloques principales, los cuales constan de subactividades. Los bloques son:

- Creación del conjunto de datos
- Detección, seguimiento y estimación de velocidades
- Entrenamiento y evaluación de modelos

Una visión general de todas las fases del presente trabajo se presenta en el Diagrama 1.

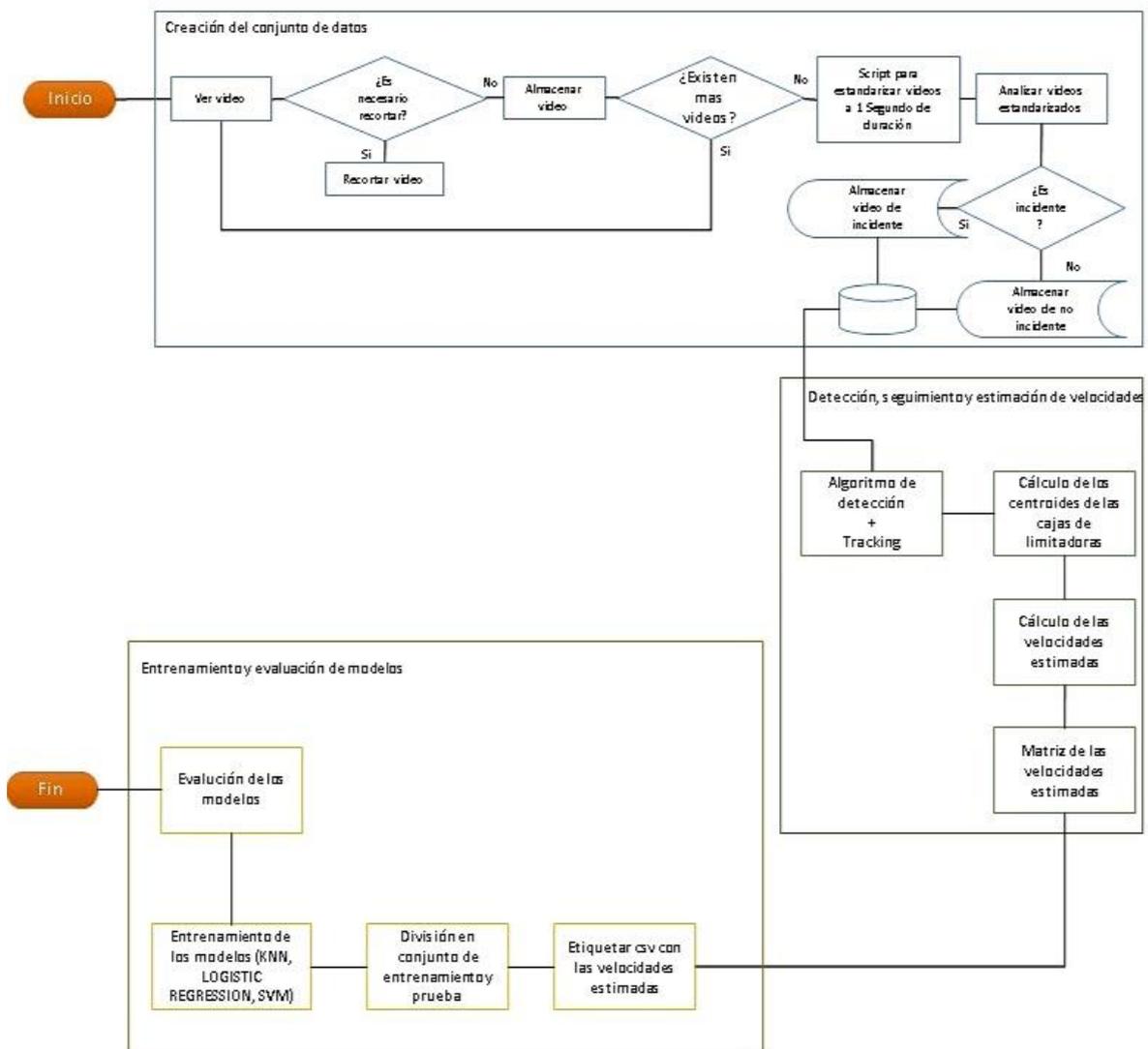


Diagrama 1: Etapas generales de la detección de incidentes automovilísticos

Para la creación de los modelos anteriormente mencionados, se necesitó contar con la detección de automóviles. Para esto, se tomó en consideración YOLO v3 y SSD MobileNet. El primero se encuentra en la API de Tensorflow [114] (definiciones de red con pesos pre entrenados o previamente entrenados) de la biblioteca de aprendizaje automático Tensorflow versión 1.13.2 [115]. El segundo algoritmo se lo utilizó por medio del modelo Caffe [116], que utiliza dos tipos de archivos como son de extensión prototxt que contiene la arquitectura del modelo y caffemodel contiene los pesos. También se utilizó el módulo de Redes Neuronales Profundas dnn que contiene la librería de computación visual de uso libre OpenCv [117].

Después de detectar el automóvil en cuestión, se lo rastreó con: Centroide, MedianFlow, MOSSE, KCF y por escala. La mayoría de estas funciones se encuentran en la librería OpenCv (en la API Tracking), a excepción del rastreo por escala el cual se encuentra en el kit de herramientas escala [118], que es una librería de ML para resolver problemas del mundo real.

El algoritmo de clasificación KNN se encuentra en el módulo de aprendizaje automático de última generación Scikit-learn [119]. Para los demás algoritmos como regresión logística y SVM se utilizó el conjunto de bibliotecas y APIs de código abierto RAPIDS [120], el cual usa CUDA para la optimización y paralelismo de las unidades de procesamiento gráfico (GPU), específicamente con la API cuML [120] dedicada al aprendizaje automático.

Para el ajuste de cada modelo de clasificación con la tarea objetivo, en este caso la detección de un incidente automovilístico en videos de tráfico se requirió de un conjunto de videos (secuencia de imágenes) de incidentes y no incidentes. Para ello, en el presente trabajo se utilizaron dos conjuntos de datos de datos i.e. Anomaly Detection Dataset (UCF) [121] y Car Accidents Detection and Prediction (CADP) [122], pertenecientes a eventos anómalos los cuales contienen videos de cámaras estáticas de seguridad y de cámaras colocadas en el tablero del vehículo y cascos.

Los videos utilizados en este trabajo fueron aquellos filmados por cámaras de seguridad estáticas.

### **Anomaly Detection Dataset (UCF)**

Contiene un conjunto de videos de vigilancia no recortados que cubren trece tipos de anomalías como son: abuso, arrestos, incendios premeditados, robos, asaltos, accidentes de tráfico, explosiones, peleas, disparos, robo de locales comerciales, y

vandalismo. Los videos pertenecen a YouTube y LiveLeak, no existen videos que sean editados manualmente, tomados por noticieros, videos de broma, capturados por cámaras de celulares, así como videos que no se noten claramente la anomalía y se toman todos los capturados por cámaras CCTV.

Por los criterios anteriormente mencionados, este conjunto de datos recopila 950 videos de anomalías y 950 normales, en el conjunto de entrenamiento se tienen 800 videos normales y 810 anómalos, por consiguiente, el conjunto de pruebas tiene 150 y 140 respectivamente. En lo concerniente a los accidentes de tránsito que es el corazón de esta investigación, se encuentran 150 videos entre normales y anómalos. El link de descarga es el siguiente: <https://webpages.uncc.edu/cchen62/dataset.html>

### **Car Accidents Detection and Prediction (CADP)**

Es un conjunto de datos que recopila videos de YouTube capturados con diferentes tipos de cámaras, calidad y editados, además de las distintas condiciones climáticas. Contiene 1416 videos de accidentes de carretera con vehículos como: autobuses, automóviles, motocicletas, de dos, tres ruedas y otros, se recopilan datos como: una lista de los identificadores de los videos en YouTube, videos cortos que funcionan como vista previa y un conjunto de fotogramas extraídos de los videos originales.

Para la recolección de estos videos utilizaron cadenas de búsqueda como: “car accidents traffic camera”, para la selección de los videos, expertos en el área recomendaron rechazos por su calidad. El link de descarga es el siguiente: [https://drive.google.com/drive/folders/1ozfZbz8wRoNqKknC59IPXP\\_twKgmCOaA](https://drive.google.com/drive/folders/1ozfZbz8wRoNqKknC59IPXP_twKgmCOaA)

#### **4.1.1.1. Creación del Conjunto de Datos**

Para la creación del conjunto de datos a utilizar en la presente investigación se han utilizado los videos de los conjuntos de datos mencionados anteriormente, tomando en cuenta que el video será recopilado si:

- Pertenece a un incidente de tráfico específicamente, colisiones entre automóviles.
- Pertenece a tráfico con fluido normal.
- No debe presentar ninguna interferencia (clima lluvioso, neblina, noche, etc.)
- Debe ser captado por una cámara CCTV estática.

Se revisaron todos los videos pertenecientes a las dos bases de datos expuestas. En cuanto a UCF-Anomaly Detection se verificaron los videos de las carpetas, Training\_Normal\_Videos\_Anomaly así como de Test\_Normal\_Videos\_Anomaly, esto para clasificar los videos de tráfico normal. Para este mismo conjunto de datos de la carpeta

RoadAccidents y del conjunto de datos CADP del archivo list\_youtube\_videos.txt, se reunieron los videos de incidentes de tráfico que cumplan con las condiciones anteriores, siguiendo los pasos tal como se muestra en la Figura 21 (bloque de creación del conjunto de datos).

Cabe recalcar que los incidentes no tienen mucho tiempo de duración, por lo que todos los videos se ajustan a 1 segundo, esto se logró mediante un script realizado en Matlab [123], por esta razón existen videos en los que el tráfico es normal hasta que el incidente se presente. Por ello hay más muestras (videos) de tráfico normal. En CADP algunos casos los videos son una recopilación de incidentes de tránsito y es necesario el uso de BeeCut como herramienta de corte de videos.

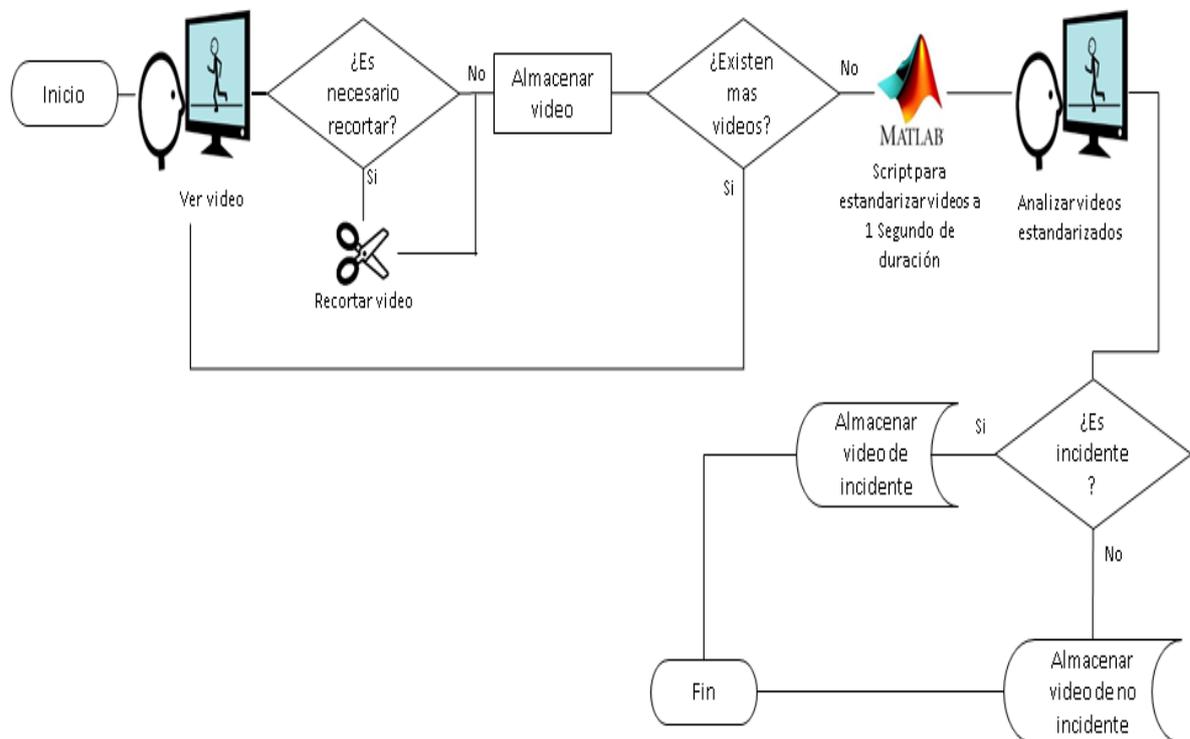


Figura 21: Diagrama de selección de videos

Con lo mencionado anteriormente se obtuvo un total de 216 videos, de los cuales 68 son de incidentes y 148 de no incidentes. Añadiendo los videos recogidos por cuenta propia en la ciudad de Riobamba, que son videos de trafico normal, el conjunto de datos creado tiene una duración total de 9 horas, 55 minutos y 15 segundos, tal como se explica en la Tabla 8.

Tabla 8: Descripción del conjunto de datos

Descripción	Número de videos	Duración
Incidentes	68	1011.84 segundos (16 minutos, 51 segundos)
No Incidentes	148	34698.76 segundos (9horas, 38 minutos, 24 segundos )
Total	216	9 horas 55 minutos 15 segundos

**Pre procesamiento:** en la búsqueda de videos para formar el conjunto de datos que servirá de soporte en la presente investigación, se encontraron videos con etiquetas de hora y fecha. Tal como se presenta en la Figura 22.



Figura 22: Imagen con etiquetas

La hora y fecha que están registrados en los videos no son indispensables en este trabajo, se procede a realizar el corte de dichas etiquetas mediante la herramienta de edición de video BeeCut, una vez realizado esto las imágenes de los videos quedan como se presentan en la Figura 23. Este tipo de corte no afecta en la detección o rastreo de algún automóvil, ya que se encuentran en los límites del borde de los fotogramas.



Figura 23: Imagen sin etiquetas

Luego de lo descrito previamente los videos fueron sometidos a un script, el cual divide a éstos en videos de 1 segundo, es decir,  $D^n = \{d_1, d_2, \dots, d_n\}$ , donde  $d_i$  representa una secuencia de imágenes (videos), se realizó una función la cual da una nueva duración de los videos, obteniendo así videos de 1 segundo de duración, en otras palabras,  $H^p = \varphi(D^n)$ , donde  $\varphi(\cdot)$  es la función que divide a los videos y se llega a tener un nuevo conjunto de videos  $H^p = \{h_1, h_2, \dots, h_p\}$ , donde  $p$  es el nuevo número de videos.

El nuevo conjunto de datos con videos de 1 segundo de duración (ventana de tiempo), se obtuvieron 26474 videos normales y 808 de incidentes, pero estos últimos no todos presentan una colisión de automóviles, mediante la revisión de cada uno de ellos se obtiene de forma definitiva que: 353 son incidentes y 27990 son normales, tal como se describen en la Tabla 9.

Tabla 9: Descripción del nuevo conjunto de datos

Descripción	Número de videos
No incidentes	27990
Incidentes	353
Total	28343

#### 4.1.1.2. Detección, seguimiento y estimación de velocidad

Una vez que ya se tuvo el nuevo conjunto de datos, se procedió a encontrar las velocidades estimadas de los automóviles detectados, mediante los seguidores: centroide, escala, KCF, medianFlow y MOSSE. Para ello cada uno de los videos pasaron por scripts (siguen la forma del Algoritmo 3) hechos en Python [124], que realizaban los siguientes pasos:

- Detecta los automóviles que aparecen en el video ya sea por SSD MobileNet o YOLO v3.
- Cada automóvil detectado es rastreado por el algoritmo de seguimiento correspondiente ya sea: centroide, escala, KCF, MOSSE, MedianFlow. esto para obtener las coordenadas que cada detección sigue en el video. Cabe recalcar que por cada algoritmo de detección de objetos se ejecutará solo un tipo de seguidor.
- Después se procede al cálculo de la velocidad estimada de cada vehículo por medio de [125], el cual lo hace mediante  $Velocidad = Distancia/Tiempo$ ; la distancia se la obtiene por medio del desplazamiento Euclidiano entre dos puntos  $(x_1, y_1)$  y  $(x_2, y_2)$ , es decir,  $D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ . En este trabajo, se ha realizado la estimación de la velocidad para cada automóvil detectado, ya que todos los videos tienen 30 fotogramas por 1 segundo, se obtuvo 29 velocidades estimadas porque el primer fotograma no cuenta. El tiempo en demorarse el paso de un fotograma a otro es 0.033 segundos, es decir,  $t_f = \frac{1 \text{ fotogramas}}{30 \text{ fotogramas/segundo}} = 0.033 \text{ segundo}$ , donde  $t_f$  es el tiempo del paso de un fotograma a otro, 30 *fotogramas/segundo* es la tasa de captura de imágenes y 1 *fotogramas* es el salto del fotograma a otro.
- Una vez calculadas las velocidades estimadas, éstas pasan a escribirse en un archivo de extensión csv, con el objetivo de adaptarlo en un formato idóneo para los pasos de entrenamiento, validación y prueba.

### Algoritmo 3: Cálculo de velocidades estimadas

Algoritmo de cálculo de velocidades estimadas de cada automóvil detectado.

**Entrada:** video de incidente automovilístico ( $h_p$ )

**Salida:** velocidades estimadas de cada automóvil detectado del video de entrada

```
rastreador ← metodo_de_rastreo()
net ← detector_de_objetos()
mientras True hacer
    fotograma ← leer_video( $h_p$ )
    si fotograma es None después
        break
    sino
        fotograma ← cambiar_de_tamaño(fotograma)
        detecciones_de_net ← detecciones(net)
        para i en detecciones_de_net hacer
            si net == 'car'
                caja ← coordenadas_de_cajas_delimitadoras(i)
                t ← rastreador(fotograma, caja)
                lista_de_cajas ← adjuntar(caja)
            fin si
        fin para
        cajas_totales ← adjuntar(lista_de_cajas)
    fin si
fin mientras
velocidades_estimadas ← velocidades(cajas_totales)
```

Los pasos mencionados anteriormente pueden ser descritos en un diagrama tal como se presenta en la Figura 24 (bloque de detección, seguimiento y estimación de velocidades).

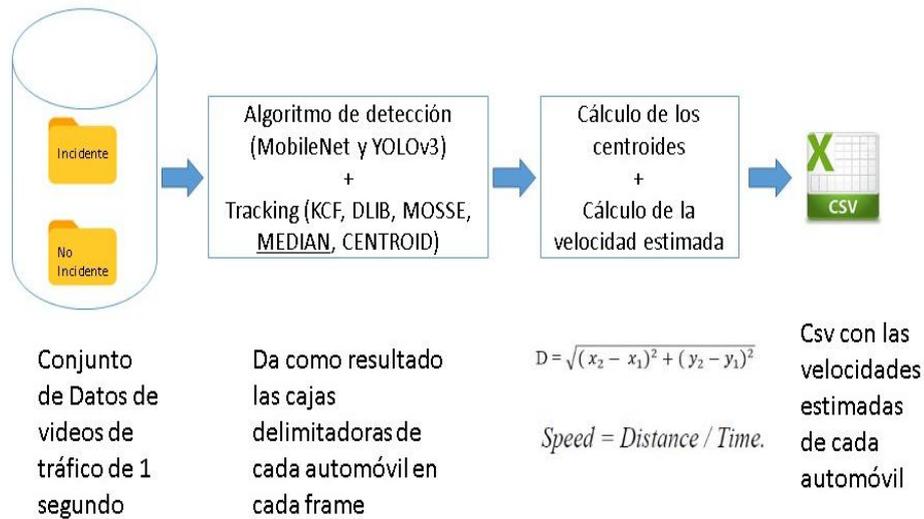


Figura 24: Diagrama cálculo de velocidades estimadas

Las velocidades obtenidas de los videos de tráfico normal se etiquetaron con el número 0 y los que presentan colisión de automóviles con 1, lo que conlleva a formar un nuevo conjunto de datos con 29 columnas (características) pertenecientes a las velocidades estimadas de cada paso de fotograma, es decir; se tendría que cada fila pertenece a un automóvil detectado en el video, lo cual puede ser representado en  $automovil_i = \{v_1, v_2, \dots, v_{29}\}$ , donde  $v_i$  es la velocidad perteneciente al  $automovil_i$

Con este antecedente, el conjunto de datos a usar sería  $G = \{automovil_1, automovil_2, \dots, automovil_n\}$ . Cabe recalcar que  $G$ , es un conjunto de datos generado por cada algoritmo de detección y de rastreo.

#### 4.1.1.3. Entrenamiento y evaluación de modelos

Este bloque es el paso final para poder recolectar los resultados obtenidos al aplicar el plan del Diagrama 1; consta de cuatro fases (etiquetación de conjunto de datos, división del conjunto de datos, entrenamiento de los modelos y medición del rendimiento de los modelos).

**Fase 1:** Etiquetación de conjunto de datos: parte indispensable en el aprendizaje supervisado. En esta fase se agregan etiquetas informativas y significativas para darle sentido para que los modelos de aprendizaje automático aprendan de ello [126]. Cabe recalcar que el rendimiento de los modelos de clasificación dependen de la calidad de los datos etiquetados que se usan para el entrenamiento [127]. En el presente trabajo se colocó como etiquetas, los números 0 y 1 correspondiente a no incidente e incidente respectivamente.

**Fase 2:** División del conjunto de datos, en este caso se obtuvieron 10 conjuntos de datos. Cinco pertenecientes a SSD MobileNet con todos los rastreadores y cinco restantes con los mismos seguidores, pero con YOLO v3.

Con lo mencionado anteriormente se dividió a cada uno de los conjuntos de datos en 70% destinado para entrenamiento, etapa que se realizó con la validación cruzada de 10 pliegues mediante la función GridSearchCV de la librería Scikit-learn [119], esto para obtener el modelo con mejor exactitud. La partición restante está dirigida al conjunto de prueba con el 30%, conjunto crucial con el que se halló la exactitud del modelo creado con datos no entrenados.

Esta división se lo realizó por medio de la una partición de tipo estratificada, es decir, que la división de los datos tenga la misma proporción de valores en los conjuntos producidos (entrenamiento y prueba).

**Fase 3:** Entrenamiento de los modelos: se han utilizado los siguientes algoritmos de clasificación: KNN, regresión logística y SVM. El entrenamiento se realizó por medio de validación cruzada de 10 pliegues con ajustes de hiper parámetros utilizando la función GridSearchCV de Scikit-learn lo cual ayuda a obtener un modelo con mejor exactitud. Cabe indicar que para el entrenamiento se utilizó seis núcleos.

A continuación se describen los hiper parámetros que se utilizaron para cada modelo con desbalance de datos.

KNN utiliza un rango que va de 1 a 10 como número de vecinos. El tipo de algoritmos usados son: auto, ball\_tree, kd\_tree y brute. Además, el tipo de cálculo de distancia fueron manhattan, eculidean y mincowski,

Regresión logística usa como tipo de penalización los tipos l1 y l2; como inverso de la fuerza de regularización se usaron cuatro cantidades diferentes: 1, 10, 100 y 1000. De algoritmos de solución se utilizó newton-cg, lbfgs, liblinear, sag y saga.

En lo referente a SVM el entrenamiento se lo hace mediante GPUs utilizando la librería cuml del API de Rapids [128]. Con núcleo RBF gamma utilizó los dos únicos tipos que existe scale y auto.

Con núcleo polinomial se probó con grado 2 y 3; en el hiperparámetro gamma se usó las opciones scale y auto, y; en C cuatro cantidades: 1, 10, 100, 1000. Con núcleo sigmoideo sus hiper parámetros son los mismos que el anteriormente expuesto, excepto en la parte del grado ya que éste se suprime.

Cabe mencionar que la función de GridSearchCV es probar todos los hiper parámetros introducidos; por ejemplo en KNN al tener vecinos de rango de 1 a 10, se probará el entrenamiento con 1 vecino con el algoritmo auto con distancia manhattan, realiza validación cruzada con 10 pliegues, y revisa su exactitud de clasificación; de esta forma se puede evidenciar el costo computacional que se necesita para obtener el mejor modelo.

En cuanto a los conjuntos de datos balanceados con los que se crearon los modelos correspondientes se tienen diferentes configuraciones en los hiper parámetros, esto porque se hizo sobre muestreo con Bootstrap para que los datos de incidentes sean de igual cantidad que los no incidentes y se realiza validación cruzada de 5 pliegues.

Para el modelo KNN el número de vecinos es en el rango de 1 a 5, únicamente con el algoritmo auto, con pesos uniform y métrica de cálculo de vecinos Euclidiana.

En el modelo de regresión logística se usa como hiper parámetros los mismos que se emplearon con datos desbalanceados, debido a que este tipo de modelo es muy ligero y no requiere de tantos recursos para su entrenamiento, validación y evaluación.

En lo referente a los modelos SVM, con núcleo polinomial se utiliza grado tres en el regularizador de parámetros (C) que es igual a 1; gamma de tipo escala el cual define cuanta influencia tiene una sola muestra de entrenamiento. Con núcleo sigmoideo se usó coef0 con el valor de 0 ya que es un parámetro que se usa para escalar los datos y sin límite de iteraciones. Con núcleo RBF gamma utilizó el tipo auto.

También, a lo anteriormente expuesto se realizó el paso del sobre muestreo, en el cual se incrementan los datos de las velocidades estimadas de cada automóvil detectado en videos de incidentes, es decir, del conjunto de datos  $G$ , el valor a incrementar es el número de datos de no incidentes esto para tener un balance exacto de datos (igual número de incidentes y no incidentes), este paso se lo hace mediante el algoritmo de bootstrap, que lo tiene Scikit-learn [119].

Después del balanceo de datos se procede de forma normal hacia el entrenamiento, pero teniendo en cuenta que ahora se tiene un nuevo conjunto de datos balanceado, que puede denotarse como  $G'$ .

Se tienen 100 modelos creados (50 con datos desbalanceados y 50 con datos balanceados), esto debido al tipo de algoritmo de detección de automóviles y el seguidor usado como se explicó anteriormente.

**Entrenamiento:** esta etapa corresponde al uso del 70% del conjunto de datos  $G$  y  $G'$  para el entrenamiento de cada clasificador, tomando en cuenta el ajuste de hiper parámetros

anteriormente mencionados. En la Tabla 10, se menciona el número de tuplas (filas) que existen en cada conjunto de datos generado por el algoritmo de detección y su rastreador correspondiente.

Tabla 10: Número de tuplas en los conjuntos de datos

<b>Conjuntos de Datos Generados</b>									
<b>Datos Desbalanceados</b>									
<b>SSD MobileNet</b>					<b>YOLOv3</b>				
<b>Centroide</b>	<b>KCF</b>	<b>MOSSE</b>	<b>MedianFlow</b>	<b>Escala</b>	<b>Centroide</b>	<b>KCF</b>	<b>MOSSE</b>	<b>MedianFlow</b>	<b>Escala</b>
87294	44315	144609	142450	128805	15635	196231	198013	249201	363442
<b>Datos Balanceados</b>									
<b>SSD MobileNet</b>					<b>YOLOv3</b>				
<b>Centroide</b>	<b>KCF</b>	<b>MOSSE</b>	<b>MedianFlow</b>	<b>Escala</b>	<b>Centroide</b>	<b>KCF</b>	<b>MOSSE</b>	<b>MedianFlow</b>	<b>Escala</b>
283184	84974	283184	279096	251644	24908	377440	380126	483304	711762

Como se mencionó anteriormente el entrenamiento se realizó con el 70% del conjunto de datos. En la Tabla 11, se menciona cuantos datos han sido destinados para el entrenamiento (número de filas).

Tabla 11: Número de tuplas del conjunto de datos para entrenamiento

<b>Número de Datos Destinados al Entrenamiento</b>									
<b>Datos Desbalanceados</b>									
<b>SSD MobileNet</b>					<b>YOLOv3</b>				
<b>Centroide</b>	<b>KCF</b>	<b>MOSSE</b>	<b>MedianFlow</b>	<b>Escala</b>	<b>Centroide</b>	<b>KCF</b>	<b>MOSSE</b>	<b>MedianFlow</b>	<b>Escala</b>
61106	31021	101226	99715	90164	10945	137362	138609	174441	254409
<b>Datos Balanceados</b>									
<b>SSD MobileNet</b>					<b>YOLOv3</b>				
<b>Centroide</b>	<b>KCF</b>	<b>MOSSE</b>	<b>MedianFlow</b>	<b>Escala</b>	<b>Centroide</b>	<b>KCF</b>	<b>MOSSE</b>	<b>MedianFlow</b>	<b>Escala</b>
198228	59481	198228	195367	176150	17435	264208	266088	338312	498233

**Fase 4:** Evaluación: una vez terminada la fase de entrenamiento, se procedió a realizar la evaluación de cada uno de los modelos de clasificación del conjunto de datos  $G$  y  $G'$ . Las métricas de evaluación son exactitud de clasificación y AUC ROC. En la Tabla 12 se presentan el número de tuplas de los conjuntos de datos destinadas a evaluarse.

Tabla 12: Número de tuplas del conjunto de datos de evaluación

Número de Datos Destinados a la Evaluación									
Datos Desbalanceados									
SSD MobileNet					YOLOv3				
Centroide	KCF	MOSSE	MedianFlow	Escala	Centroide	KCF	MOSSE	MedianFlow	Escala
26818	13294	43383	42735	38641	4690	58869	59404	74760	109033
Datos Balanceados									
SSD MobileNet					SSD MobileNet				
Centroide	KCF	MOSSE	MedianFlow	Escala	Centroide	KCF	MOSSE	MedianFlow	Escala
84956	25493	84956	83729	75494	7473	113232	114038	144992	213529

### Métricas y medidas

Uno de los estudios más frecuentes es la clasificación binaria en lo que se refiere al aprendizaje automático por ejemplo podemos mencionar las aplicaciones en las ramas de la medicina, biología, meteorología, etc. Para determinar el desempeño de cada modelo de clasificación entrenado es necesario aplicar métricas. Según [53], propone la distinción entre métricas y medidas para la clasificación binaria, por lo que plantea una hoja de ruta o tabla periódica, donde se menciona utilizar las mismas dependiendo del tipo de problema a resolver. Por ejemplo, en medicina son de vital importancia tomar en cuenta a los falsos positivos y falsos negativos.

En este caso en particular la clasificación de incidentes de tránsito (colisiones de automóviles) mediante las velocidades estimadas de los automóviles detectados al tener falsos positivos se podría recurrir a realizar una revisión manual y comprobarlo [129], lo que no causa mayor problema. Para este caso es de vital importancia el calcular la exactitud de clasificación y verificar el comportamiento de cada modelo de clasificación, por ello se usó la métrica ROC AUC ya que se la considera como la mejor para establecer el rendimiento de la clasificación [130], así mismo se estarán tomando las métricas adicionales como la sensibilidad, precisión y F1 score. Las ecuaciones de las métricas anteriormente expuestas se presentan en la Sección 2.4.

### 4.2. LIMITACIONES Y PROBLEMAS POTENCIALES

Esta sección se discuten las posibles amenazas y problemas a la validez del presente trabajo de investigación.

La primera limitación podría estar relacionada con la cantidad de modelos de clasificación creados, que utilizan la librería Scikit-learn; considerando este problema, se decidió utilizar varios modelos en algunos de ellos con diferente núcleo que son similares a redes neuronales, se utilizaron para su entrenamiento validación cruzada con 10 pliegues y se tomó el modelo con la mejor calificación de exactitud usando GridSearchCV, ajustando previamente algunos hiper parámetros como se indicó en la sección 4.1.1.3.

En la revisión de literatura realizada no se ha encontrado ningún otro estudio comparativo de tantos modelos de clasificación como se ha propuesto en la presente investigación.

Otra amenaza, es el uso de los videos de tránsito del conjunto de datos anteriormente descritos, ya que directamente éstos se utilizan para el cálculo de las velocidades estimadas de cada automóvil detectado, las cuales son el corazón de esta investigación y pasan a formar parte del conjunto de datos con los que se entrena y evalúa cada modelo de clasificación.

En el caso de entrenamiento y evaluación de los modelos SVM con diferentes núcleos, fue imprescindible utilizar GPUs ya que ayudan a obtener una respuesta en un menor tiempo, por esta razón fue necesario usar la librería cuML juntamente con GridSearchCV, y al calcular el área bajo la curva ROC se utilizó la misma librería.

Los demás modelos de clasificación fueron entrenados con CPUs lo que requirió más tiempo de ejecución en entrenamiento y evaluación.

En el caso de que los modelos presenten varias limitaciones es debido al entrenamiento de estos, ya que, por el tiempo y condicionamiento de recursos informáticos, fueron sometidos al mejor tratamiento al alcance disponible, sin embargo, para trabajos futuros se tomarán todas las mejores prácticas de esta investigación para el entrenamiento con otros modelos de clasificación.

## CAPÍTULO 5

### 5. EXPERIMENTACIÓN

Una parte importante del presente trabajo es la experimentación, debido a que es la ruta que se debe seguir para la reproducibilidad de los experimentos realizados y su posible mejoramiento de los resultados de esta propuesta.

En el presente capítulo, se presenta la configuración que se usó para los experimentos y sus respectivos resultados, lo que es piedra angular para la evaluación del desempeño de los modelos propuestos.

#### 5.1. EXPERIMENTOS PILOTOS

Los experimentos de esta sección se llevaron a cabo para confirmar las diferentes calificaciones que se le dan a cada uno de los modelos creados con  $G$  y  $G'$ , dependiendo de su algoritmo de detección y seguimiento.

##### 5.1.1. CREACIÓN DE MODELOS SIN BALANCEO DE DATOS

En la Tabla 13, se muestran los resultados de los modelos creados sin un balance de datos. Se puede observar que la exactitud más alta es 0.98, perteneciente a varios modelos por ejemplo KNN con MedianFlow y SSD Mobilinet.

Tabla 13: Exactitud de los Modelos Creados

		YOLO v3					SSD Mobilenet				
		KCF	ESCALA	MOSSE	MEDIAN	CENTROID	KCF	ESCALA	MOSSE	MEDIAN	CENTROID
SVM	KNN	0.96	0.98	0.96	0.97	0.93	0.96	0.98	0.98	0.98	0.97
	REGRESIÓN LOGÍSTICA	0.96	0.98	0.96	0.97	0.8	0.96	0.97	0.97	0.98	0.97
	POLY	0.96	0.98	0.96	0.97	0.81	0.96	0.98	0.98	0.98	0.97
	RBF	0.96	0.98	0.96	0.97	0.83	0.96	0.98	0.98	0.98	0.97
	SIGMOID	0.95	0.97	0.95	0.96	0.67	0.93	0.97	0.97	0.97	0.96

En la Figura 25 se muestran los datos de la tabla anterior que muestra de mejor manera el comportamiento de la exactitud de cada modelo.

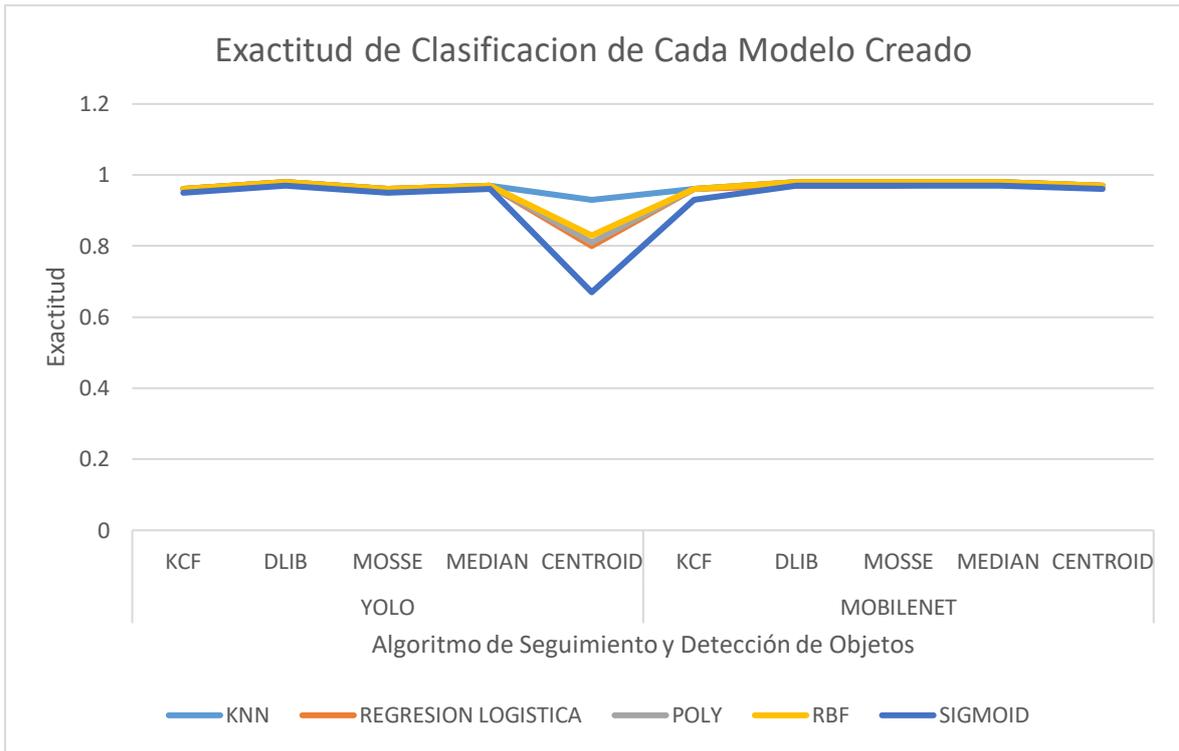


Figura 25: Comportamiento de la exactitud en los modelos creados

De la misma forma, se propuso determinar la métrica de AUC ROC de todos los modelos, las cuales se presentan en la Tabla 14. La mejor calificación es 0.91 perteneciente al modelo KNN con YOLO v3 y centroide.

Tabla 14: AUC ROC Score de los Modelos Creados

		YOLO v3					SSD Mobilenet				
		KCF	ESCALA	MOSSE	MEDIAN	CENTROID	KCF	ESCALA	MOSSE	MEDIAN	CENTROID
<b>SVM</b>	<b>KNN</b>	0.77	0.83	0.73	0.80	0.91	0.78	0.80	0.76	0.79	0.81
	<b>REGRESIÓN LOGISTICA</b>	0.43	0.76	0.50	0.43	0.51	0.63	0.60	0.60	0.62	0.60
	<b>POLY</b>	0.50	0.50	0.50	0.50	0.53	0.50	0.50	0.50	0.50	0.50
	<b>RBF</b>	0.50	0.50	0.50	0.57	0.59	0.53	0.52	0.51	0.52	0.52
	<b>SIGMOID</b>	0.50	0.50	0.50	0.50	0.49	0.51	0.50	0.50	0.50	0.50

Para una mejor vista en el comportamiento en la calificación de AUC ROC, se presenta la Figura 26.

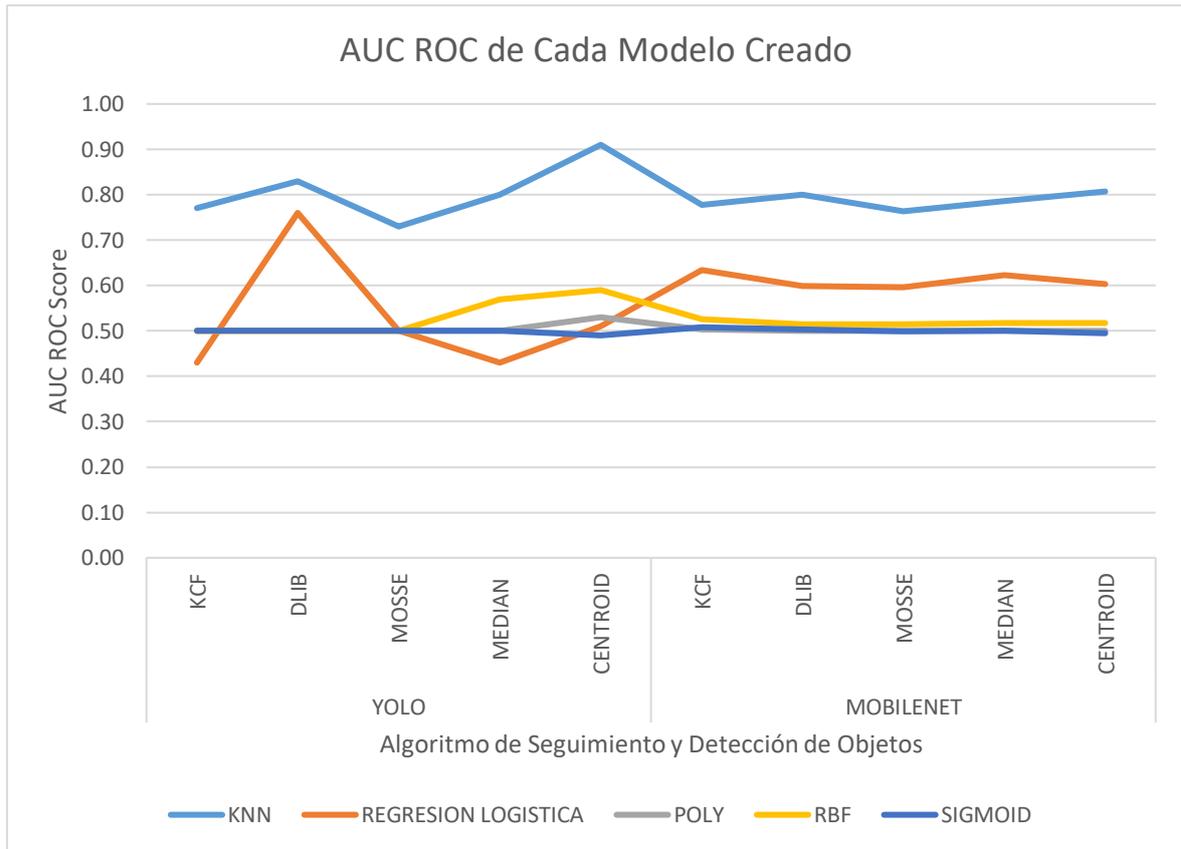


Figura 26: Comportamiento de la Calificación AUC ROC

Otra métrica que se analizó fue la precisión. La Tabla 15 muestra la tasa de los datos clasificados como verdaderos. En este caso, se tiene la mejor precisión de 0.95 perteneciente al modelo KNN con YOLO v3 y centroide.

Tabla 15: Precisión de los Modelos Creados

		YOLO v3					SSD Mobilenet				
		KCF	ESCALA	MOSSE	MEDIAN	CENTROID	KCF	ESCALA	MOSSE	MEDIAN	CENTROID
SVM	<b>KNN</b>	0.73	0.74	0.48	0.9	0.95	0.48	0.99	0.49	0.69	0.49
	<b>REGRESIÓN LOGISTICA</b>	0.48	0.49	0.48	0.48	0.52	0.48	0.49	0.49	0.49	0.49
	<b>POLY</b>	0.48	0.49	0.98	0.48	0.78	0.85	0.49	0.49	0.49	0.49
	<b>RBF</b>	0.95	0.92	0.6	0.99	0.87	0.98	0.99	0.99	0.99	0.99
	<b>SIGMOID</b>	0.5	0.51	0.49	0.5	0.49	0.51	0.51	0.51	0.5	0.49

El comportamiento de los modelos creados en la parte de precisión se presenta en la Figura 27.

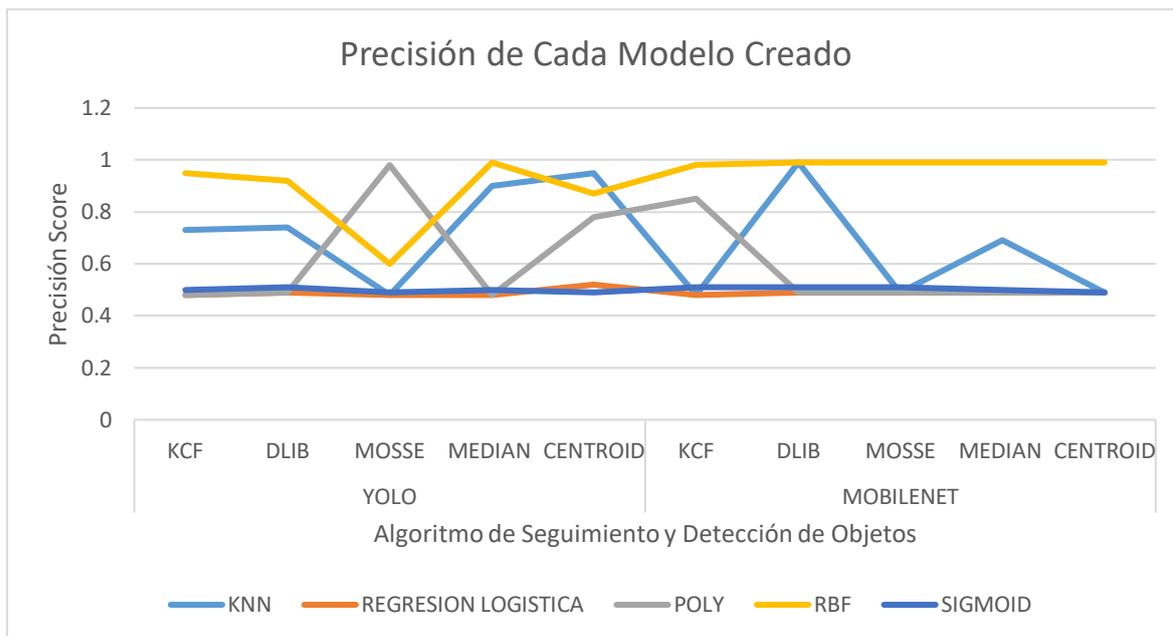


Figura 27: Comportamiento de la Precisión en los Modelos

En cuanto a la sensibilidad la cual toma en cuenta los falsos negativos y verdaderos positivos, se muestra en la Tabla 16, las calificaciones que se encontraron en los modelos creados, donde nuevamente aparece como mejor modelo KNN con YOLO v3 y centroeide con una sensibilidad de 0.82.

Tabla 16: Sensibilidad de los Modelos Creados

		YOLO v3					SSD Mobilenet				
		KCF	ESCALA	MOSSE	MEDIAN	CENTROID	KCF	ESCALA	MOSSE	MEDIAN	CENTROID
<b>SVM</b>	<b>KNN</b>	0.5	0.5	0.5	0.5	0.82	0.5	0.5	0.5	0.5	0.5
	<b>REGRESIÓN LOGÍSTICA</b>	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	<b>POLY</b>	0.5	0.5	0.5	0.5	0.54	0.5	0.5	0.5	0.5	0.5
	<b>RBF</b>	0.5	0.51	0.5	0.57	0.59	0.53	0.52	0.51	0.52	0.52
	<b>SIGMOID</b>	0.5	0.5	0.5	0.5	0.49	0.51	0.5	0.5	0.5	0.5

Para una mejor comprensión en la Figura 28 se describe el comportamiento de la sensibilidad de los modelos creados.

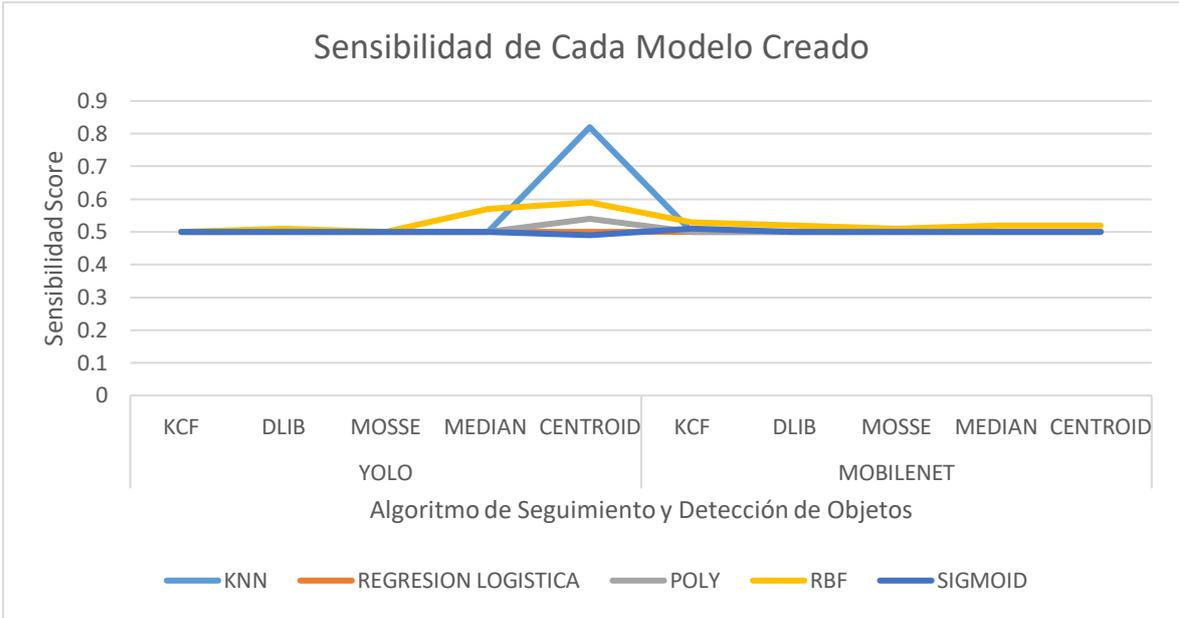


Figura 28: Comportamiento de la Sensibilidad en los Modelos

También, se presenta la matriz de confusión (Figura 29) del modelo KNN con algoritmo de detección YOLO v3 y de seguimiento, el método del centroide que tiene como exactitud 0.93, en AUC ROC 0.91, precisión 0.95 y sensibilidad 0.82.

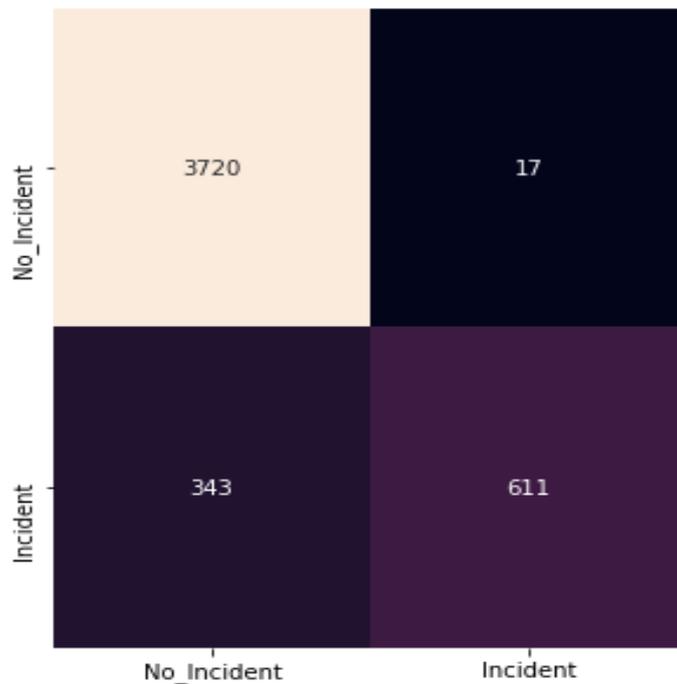


Figura 29: Matriz de Confusión Modelo KNN

También la Figura 30 muestra el comportamiento de AUC ROC de dicho modelo.

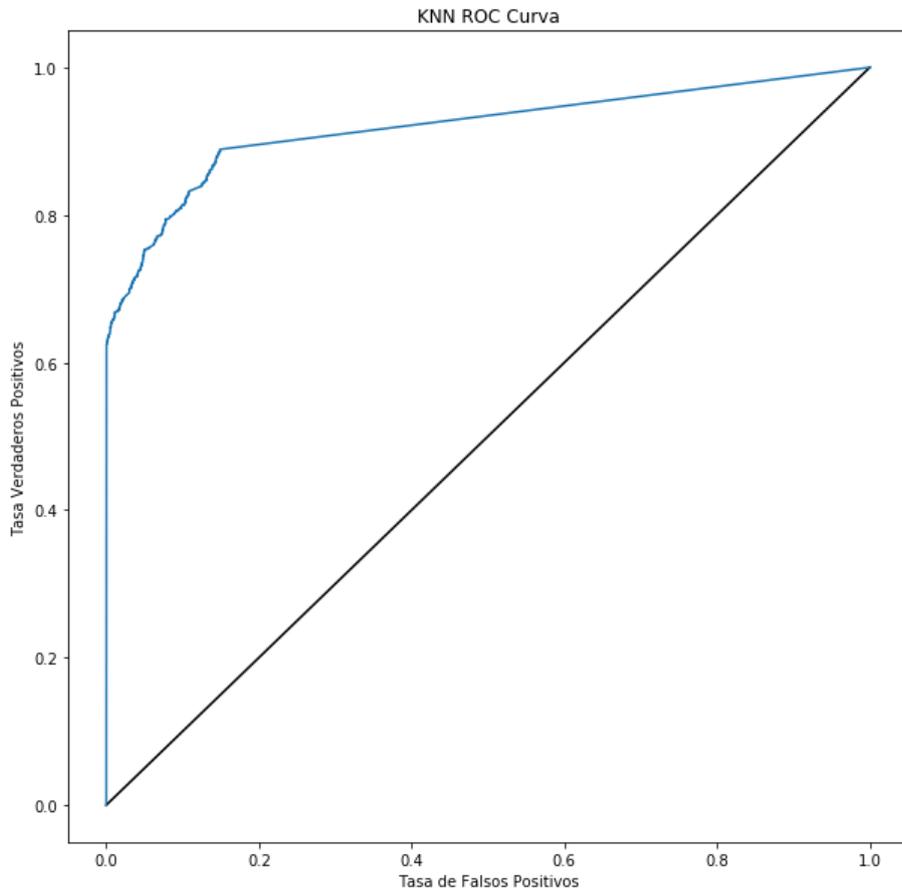


Figura 30: Comportamiento de AUC ROC del Modelo KNN

En el Anexo 2 se presentan las matrices de confusión pertenecientes a todos los modelos creados.

### 5.1.2. CREACIÓN DE MODELOS CON BALANCEO DE DATOS

En la Tabla 17, se presentan los resultados en cuanto a la exactitud de los modelos creados con balanceo de datos. En este caso, cuatro modelos tuvieron la mejor exactitud y estos fueron: KNN con SSD Mobilenet como detector de objetos, centroid, medianFlow y escala.

Tabla 17: Exactitud de los modelos creados con balanceo de datos

		YOLO v3					SSD Mobilenet				
		KCF	ESCALA	MOSSE	MEDIAN	CENTROID	KCF	ESCALA	MOSSE	MEDIAN	CENTROID
SVM	<b>KNN</b>	0.9	0.95	0.84	0.93	0.85	0.94	0.95	0.9	0.95	0.95
	<b>REGRESIÓN LOGÍSTICA</b>	0.64	0.65	0.48	0.56	0.57	0.59	0.6	0.57	0.58	0.56
	<b>POLY</b>	0.55	0.77	0.53	0.56	0.56	0.66	0.68	0.64	0.64	0.64
	<b>RBF</b>	0.64	0.77	0.58	0.68	0.65	0.73	0.75	0.72	0.72	0.72

	<b>SIGMOID</b>	0.46	0.67	0.46	0.47	0.5	0.46	0.49	0.52	0.57	0.47
--	----------------	------	------	------	------	-----	------	------	------	------	------

En la Figura 31, se muestran de una manera más dinámica los datos de la tabla anteriormente expuesta, perteneciente a la exactitud de los modelos con balanceo de datos.

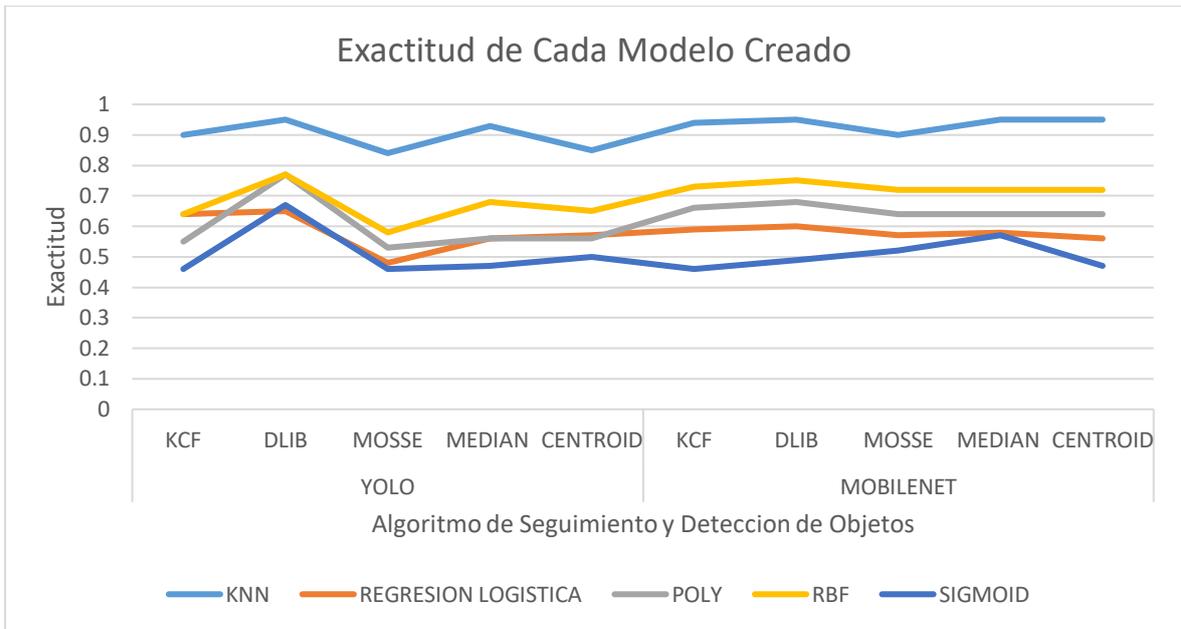


Figura 31: Exactitud de los modelos con balanceo de datos

De la misma forma se analizó la métrica de AUC ROC, con los modelos creados con balanceo de datos. La Tabla 18, muestra todas las calificaciones; en este caso, el mejor valor fue de 0.97 perteneciente al modelo KNN con SSD Mobilenet como detector de objetos y MOSSE como rastreador.

Tabla 18: AUC ROC de cada modelo creado

		YOLO v3					SSD Mobilenet				
		KCF	ESCALA	MOSSE	MEDIAN	CENTROID	KCF	ESCALA	MOSSE	MEDIAN	CENTROID
SVM	KNN	0.92	0.96	0.84	0.95	0.8	0.95	0.96	0.97	0.95	0.95
	REGRESIÓN LOGISTICA	0.58	0.77	0.49	0.62	0.59	0.62	0.62	0.59	0.62	0.61
	POLY	0.55	0.77	0.53	0.65	0.56	0.66	0.68	0.64	0.64	0.64
	RBF	0.65	0.77	0.56	0.68	0.65	0.73	0.75	0.72	0.72	0.72
	SIGMOID	0.46	0.67	0.46	0.47	0.5	0.49	0.49	0.52	0.57	0.47

Para una mejor representación de los valores de la Tabla 18, se muestra la Figura 32.

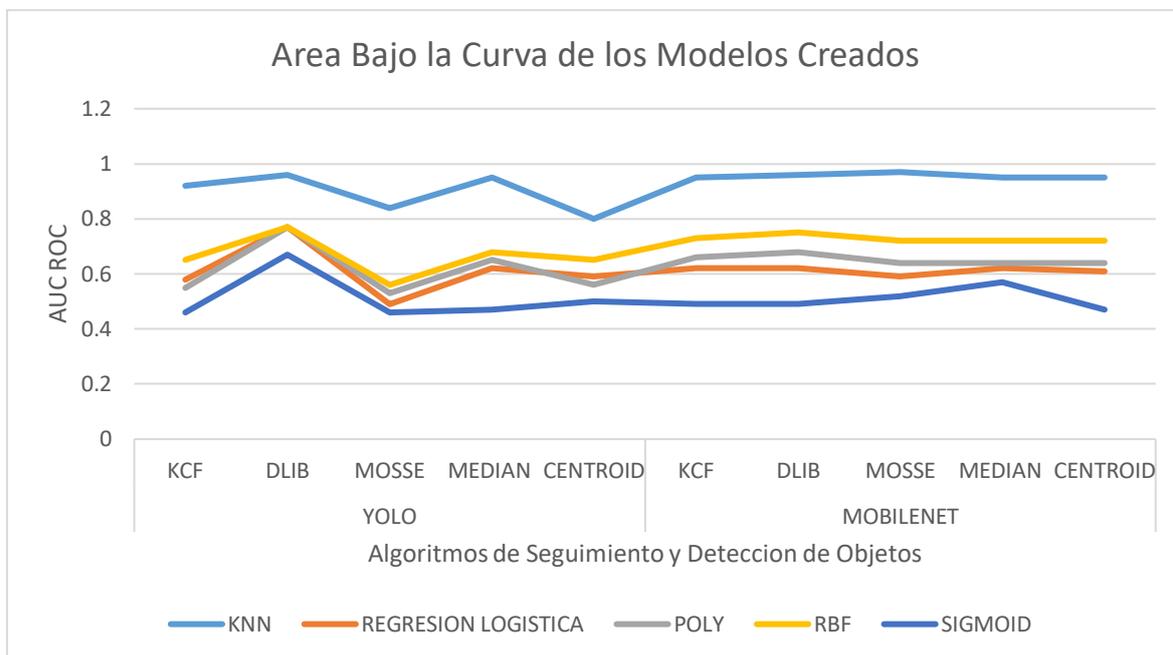


Figura 32: Métrica AUC ROC de los modelos con datos balanceados

Otra métrica estudiada en este trabajo fue la sensibilidad, la cual toma en cuenta los valores de los falsos negativos y verdaderos positivos de la clasificación realizada. La Tabla 19, muestra las calificaciones que se obtuvieron en los modelos creados con datos balanceados.

Tabla 19: Valores de sensibilidad de cada modelo creado

		YOLO v3					SSD Mobilenet				
		KCF	ESCALA	MOSSE	MEDIAN	CENTROID	KCF	ESCALA	MOSSE	MEDIAN	CENTROID
<b>SVM</b>	<b>KNN</b>	0.92	0.96	0.84	0.95	0.8	0.95	0.96	0.97	0.95	0.95
	<b>REGRESIÓN LOGISTICA</b>	0.58	0.77	0.49	0.62	0.59	0.62	0.62	0.59	0.62	0.61
	<b>POLY</b>	0.55	0.77	0.53	0.65	0.56	0.66	0.68	0.64	0.64	0.64
	<b>RBF</b>	0.65	0.77	0.56	0.68	0.65	0.73	0.75	0.72	0.72	0.72
	<b>SIGMOID</b>	0.46	0.67	0.46	0.47	0.5	0.49	0.49	0.52	0.57	0.47

En el caso de la sensibilidad el mejor valor fue de 0.97, perteneciente al modelo KNN con algoritmos SSD Mobilenet y MOSSE. Para una mejor comprensión del comportamiento se muestra la Figura 33.

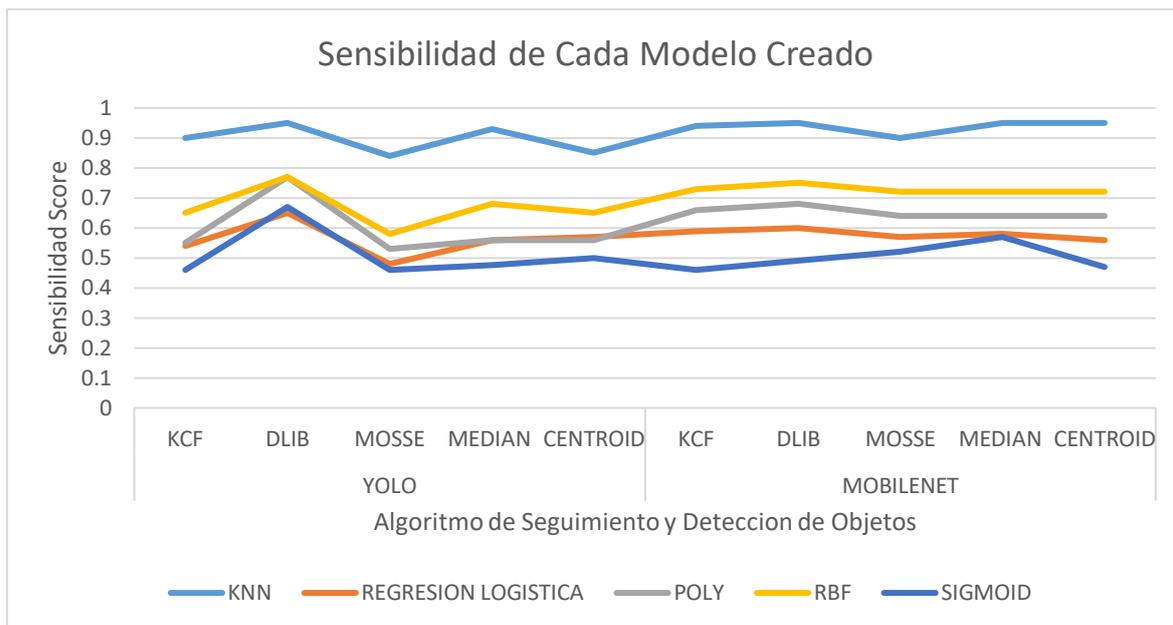


Figura 33: Sensibilidad de los modelos creados con balanceo de datos

En cuanto a la métrica de la precisión de los modelos creados con balanceo de datos se los presenta en la Tabla 20. Los modelos con SSD Mobilenet con centroide, medianFlow y escala obtuvieron el valor de 0.95, de igual manera el modelo con YOLO v3 con rastreador de escala.

Tabla 20: Precisión de los modelos creados con balanceo de datos

		YOLO v3					SSD Mobilenet				
		KCF	ESCALA	MOSSE	MEDIAN	CENTROID	KCF	ESCALA	MOSSE	MEDIAN	CENTROID
SVM	<b>KNN</b>	0.9	0.95	0.87	0.93	0.86	0.94	0.95	0.91	0.95	0.95
	<b>REGRESIÓN LOGISTICA</b>	0.56	0.66	0.48	0.57	0.58	0.6	0.61	0.58	0.59	0.57
	<b>POLY</b>	0.7	0.78	0.74	0.73	0.69	0.76	0.76	0.75	0.75	0.75
	<b>RBF</b>	0.65	0.78	0.58	0.68	0.67	0.75	0.77	0.74	0.75	0.73
	<b>SIGMOID</b>	0.46	0.67	0.46	0.47	0.5	0.46	0.49	0.52	0.57	0.47

En la Figura 34, se muestran los valores de la métrica de la precisión de cada uno de los modelos creados con datos balanceados.

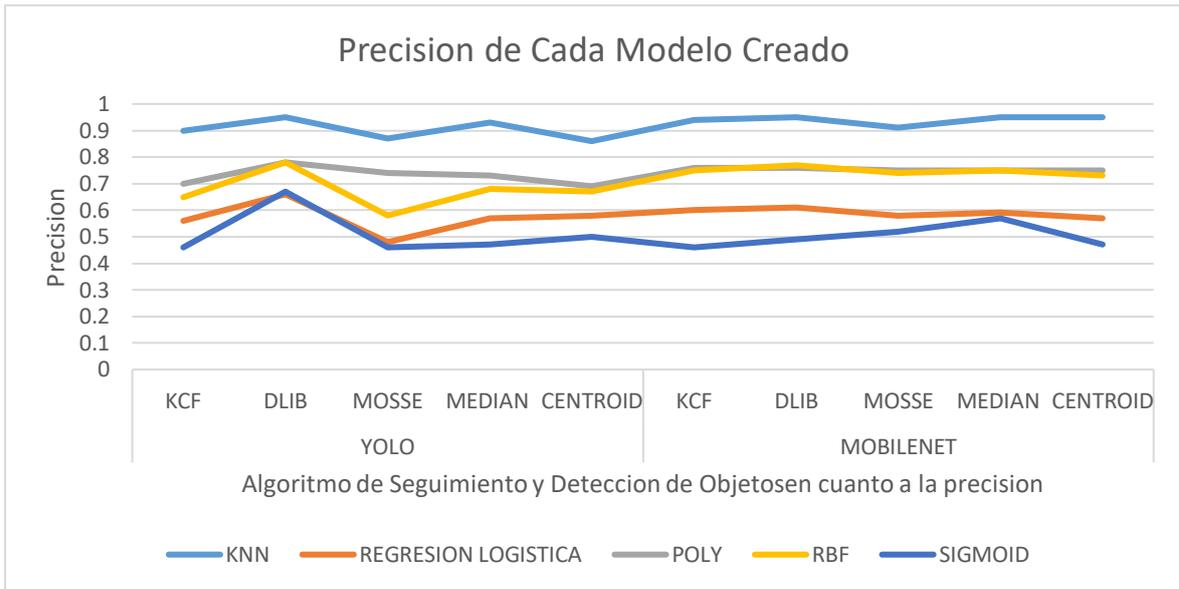


Figura 34: Precisión de los modelos creados

Una de las métricas que ayuda a la evaluación del desempeño de los modelos creados es la calificación F1, en la Tabla 21 se muestran todos los valores que se obtuvieron. El valor más alto es de 0.95, con los modelos iguales al del caso de la precisión.

Tabla 21: Calificación F1 de los modelos creados

		YOLO v3					SSD Mobilenet				
		KCF	ESCALA	MOSSE	MEDIAN	CENTROID	KCF	ESCALA	MOSSE	MEDIAN	CENTROID
SVM	<b>KNN</b>	0.9	0.95	0.84	0.93	0.85	0.94	0.95	0.9	0.95	0.95
	<b>REGRESIÓN LOGISTICA</b>	0.52	0.64	0.45	0.5	0.55	0.58	0.6	0.56	0.57	0.55
	<b>POLY</b>	0.45	0.76	0.4	0.47	0.47	0.62	0.66	0.6	0.6	0.64
	<b>RBF</b>	0.65	0.76	0.56	0.67	0.63	0.72	0.74	0.71	0.72	0.71
	<b>SIGMOID</b>	0.46	0.67	0.46	0.47	0.5	0.46	0.49	0.52	0.57	0.47

De igual manera que en las métricas anteriores, se presenta en la Figura 34, los valores de F1 de cada modelo creado con datos balanceados.

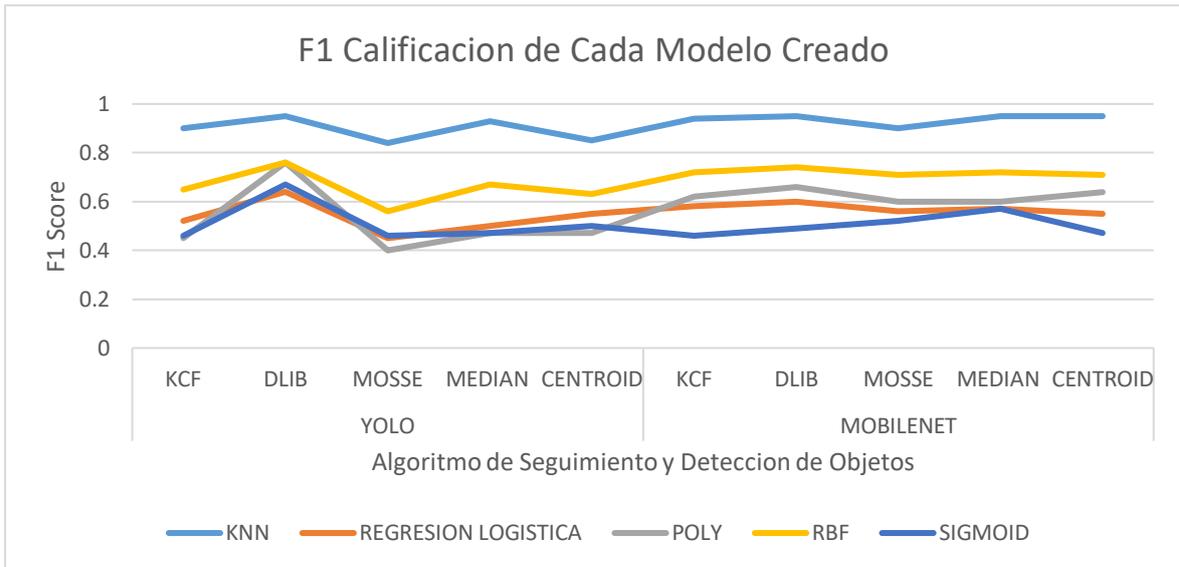


Figura 35: Calificación F1 de los modelos creados con datos balanceados

También, se presentan las matrices de confusión de los modelos con mejor rendimiento, en este caso son dos modelos.

Y kNN con algoritmo de detección SSD Mobilenet, de seguimiento el método de escala que tiene como exactitud 0.95, en AUC ROC 0.96, precisión 0.95 y sensibilidad 0.96 (Figura 36 y Figura 37).

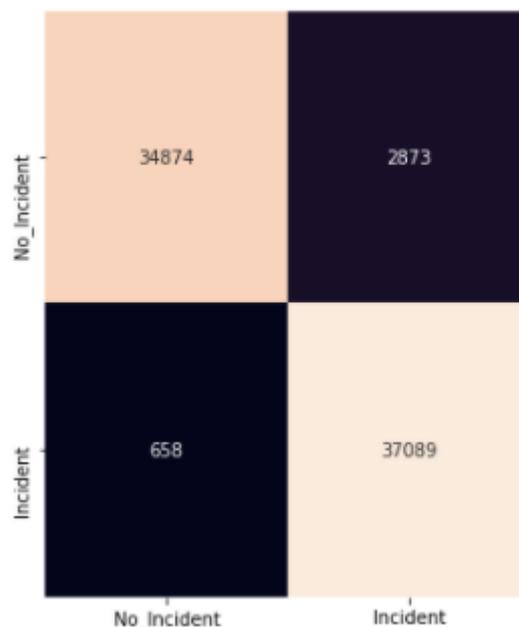


Figura 36: Matriz de confusión primer modelo con mejor rendimiento

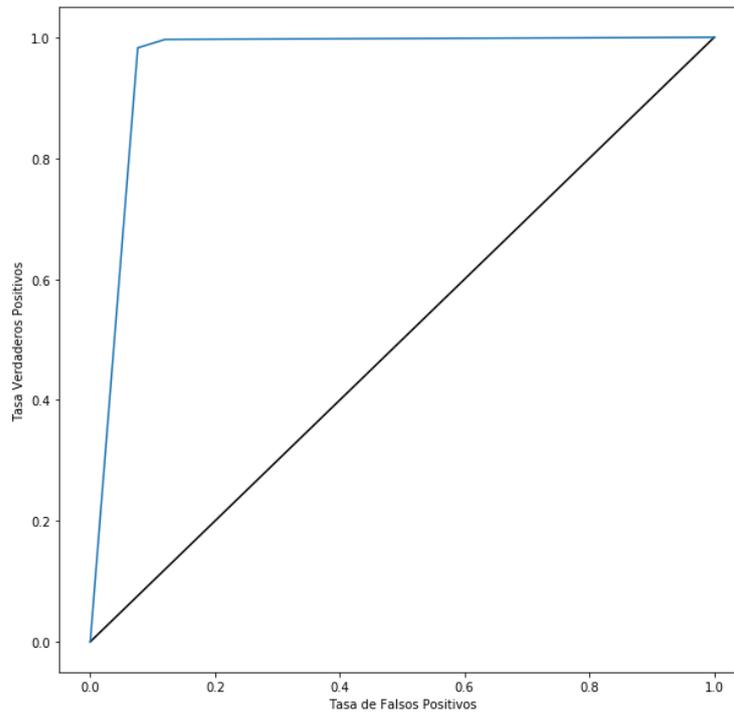


Figura 37: Curva ROC AUC primer modelo con mejor rendimiento

kNN con algoritmo de detección YOLO v3, seguimiento el método de escala que tiene como exactitud 0.95, en AUC ROC 0.96, precisión 0.95 y sensibilidad 0.96 (Figura 38 y Figura 39).

No_Incident	100077	6688
Incident	4246	102518
	No_Incident	Incident

Figura 38: Matriz de confusión segundo modelo con mejor rendimiento

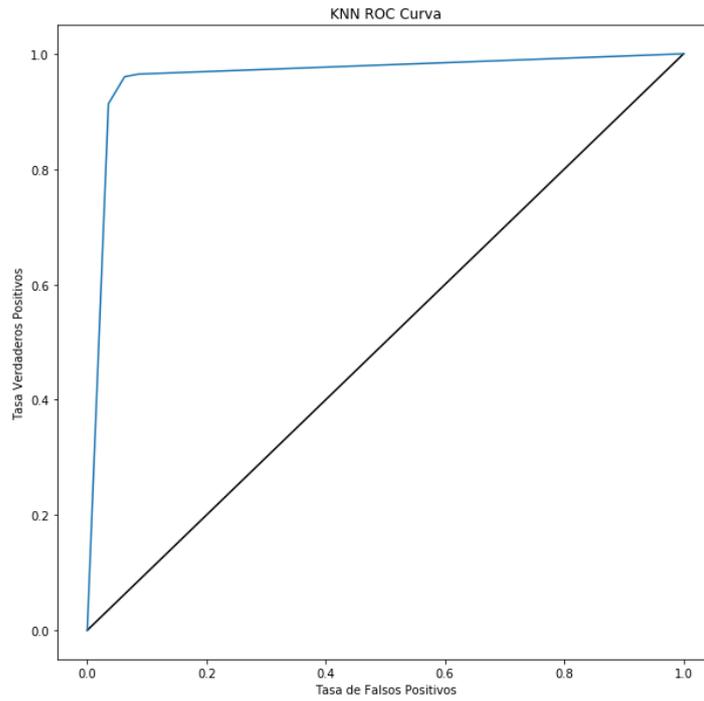


Figura 39: Curva AUC ROC segundo modelo con mejor rendimiento

En el Anexo 3 se presentan las matrices de confusión pertenecientes a todos los modelos creados.

## **CAPÍTULO 6**

### **6. CONCLUSIONES**

#### **6.1. OBJETIVOS DE INVESTIGACIÓN: RESUMEN DE LOS HALLAZGOS Y CONCLUSIONES**

Existen varias preguntas de investigación que se han ilustrado a través de la presente investigación. Primero: ¿Qué métodos de clasificación se han utilizado para la detección de incidentes automovilísticos? En segundo lugar: ¿Qué métodos de detección y seguimiento se han aplicado en videos de tráfico? Tercero: ¿Qué bases de datos públicas con videos de incidentes de tránsito hay disponibles? Finalmente ¿Qué métricas se han utilizado para evaluar el desempeño del algoritmo o modelo creado?.

En las próximas subsecciones se aborda estas preguntas alineadas a los objetivos de investigación planteados. Se inicia con lo relevante a la revisión de literatura realizada. Luego, se evalúan los descubrimientos experimentales y se proponen los trabajos futuros tentativos.

##### **6.1.1. DETECCIÓN DE INCIDENTES AUTOMOVILÍSTICOS CON VIDEOS DE TRÁFICO Y APRENDIZAJE DE AUTOMÁTICO.**

###### **Conclusiones**

Los métodos de seguimiento o rastreo cumplen un proceso de búsqueda de la ubicación de un objeto de interés en los fotogramas siguientes, su modelo de movimiento puede ser representado matemáticamente y ayuda a la estimación de la velocidad del objeto en cuestión.

Un modelo de clasificación es parte del aprendizaje supervisado que tiene como fin predecir las etiquetas de la clase de forma discreta de las nuevas muestras.

La revisión de literatura realizada reveló que la mayoría de las investigaciones utilizan una función en la que tienen un umbral para determinar un incidente, de ahí la necesidad de una investigación empírica para estudiar la detección de incidentes automovilísticos mediante técnicas de detección y rastreo de objetos con aprendizaje supervisado. Además, como se indica en las revisiones sistemáticas de literatura [131], [132], no existen métricas estándar en el estado del arte. Así mismo, indican las revisiones mencionadas

que en la mayor parte de trabajos existe la falta de datos de incidentes de tráfico que son muy escasos y no se tiene un conjunto de datos estándar público.

### **6.1.2. EXPERIMENTACIÓN CON EL CONJUNTO DE DATOS PARA LA CREACIÓN DE LOS MODELOS DE CLASIFICACIÓN**

#### **Conclusiones**

Los modelos con balance de datos con sobre muestreo aleatorio de las velocidades estimadas pertenecientes a incidentes automovilísticos logran una puntuación más alta en exactitud, sensibilidad, precisión, calificación F1 y AUC ROC, a los modelos con desbalance de datos.

### **6.1.3. MODELO CON MEJOR RENDIMIENTO DE CLASIFICACIÓN EN LA DETECCIÓN DE INCIDENTES AUTOMOVILÍSTICOS**

#### **Conclusiones**

La matriz de confusión que se presenta en la Figura 35, mejora claramente con respecto a la Figura 37, perteneciente al modelo kNN con SSD Mobilenet y seguidor de escala que tuvo el mejor desempeño de todos y también se mejora el resultado en la clasificación mejorando el estado del arte en AUC ROC = 0.96 y exactitud = 0.95.

## **6.2. CONTRIBUCIONES CIENTÍFICAS**

Este trabajo presenta dos puntos clave como contribución al conocimiento. En primer lugar, hasta donde sabemos, no existe otro estudio que compare cien modelos de clasificación de incidentes automovilísticos en el estado del arte. Además, de indicar que no solo un tipo de clasificador sea la mejor técnica para la discriminación de incidentes automovilísticos, así que se probó con regresión logística y SVM con distintos núcleos.

El segundo punto es la creación de un conjunto de datos que contienen videos de incidentes y no incidentes automovilísticos por medio de cámaras de seguridad, lo que ayudará a futuras investigaciones, ya que es contiene 68 videos completos de accidentes automovilísticos específicamente choques.

## **6.3. RECOMENDACIONES Y TRABAJOS FUTUROS**

Una de las principales limitaciones al desarrollar el presente trabajo fue la falta de conjuntos de datos públicos disponibles, especialmente de videos de incidentes automovilísticos. Por lo que se debería seguir contribuyendo a la recolección de videos pertenecientes a los incidentes anteriormente mencionados.

Aunque algunos de los modelos propuestos creados han logrado buenos resultados de clasificación, se debe obtener un resultado real reduciendo la necesidad del sobre muestreo aleatorio. Este trabajo muestra que los cien modelos creados pueden resolver este problema con alrededor de 216 videos. Sería deseable proponer un trabajo unificado para recopilar por lo menos esa cantidad de videos que sean totalmente de incidentes automovilísticos con sus diferentes variaciones.

Existen algunos aspectos del presente trabajo que pueden ser cuestionables y, por lo tanto, se podría realizar un estudio más profundo.

Es importante en este punto, estudiar métodos que aborden el procesamiento de todo tipo de incidentes de tránsito en diferentes ambientes (lluvia, neblina, nieve, etc.), a través de otros modelos de clasificación o con los mismos.

Finalmente, se podría desarrollar otro trabajo a futuro utilizando técnicas de aprendizaje profundo para resolver esta problemática, así como el de crear varios conjuntos de datos de entrenamiento y repetir la metodología propuesta en este trabajo.

## 7. REFERENCIAS BIBLIOGRÁFICAS

- [1] S. S. Thomas, S. Gupta, and V. K. Subramanian, "Event Detection on Roads Using Perceptual Video Summarization," pp. 1–11, 2017.
- [2] Z. Han, J. Liang, and J. Li, "Design of Intelligent Road Recognition and Warning System for Vehicles Based on Binocular Vision," *IEEE Access*, vol. 6, pp. 62880–62889, 2018.
- [3] MINISTERIO DE TRANSPORTE Y OBRAS PUBLICAS DEL ECUADOR, "Consecuencias del exceso de velocidad | Servicio Público para Pago de Accidentes de Tránsito." [Online]. Available: <https://www.protecciontransito.gob.ec/servicios/consecuencias-del-exceso-de-velocidad/>. [Accessed: 04-Feb-2020].
- [4] A. Vadeby and Å. Forsman, "Traffic safety effects of new speed limits in Sweden," *Accid. Anal. Prev.*, vol. 114, pp. 34–39, 2018.
- [5] C. Jurewicz, A. Sobhani, J. Woolley, J. Dutschke, and B. Corben, "Exploration of Vehicle Impact Speed - Injury Severity Relationships for Application in Safer Road Design," *Transp. Res. Procedia*, vol. 14, pp. 4247–4256, 2016.
- [6] S. Dalve and A. Pinto, "Estrategia de Seguridad Vial," *Banco Interam. Desarro.*, p. 201, 2015.
- [7] Organización Panamericana de la Salud (OPS), *La seguridad vial en la región de las américas*. 2016.
- [8] Organización Panamericana de la Salud and Organización Mundial de la Salud, *Estado de la seguridad vial en la Región de las Américas*. 2019.
- [9] Banco Interamericano de desarrollo, "Conceptos prácticos para la implementación de," vol. 1, p. 22, 2019.
- [10] Agencia Nacional de Transito, "reportes\_nacionales\_junio\_2020 v2." .
- [11] Agencia Nacional de Transito, "Cifras de siniestralidad Enero-Diciembre 2019," pp. 1–1, 2019.
- [12] A. Haydari and Y. Yilmaz, "Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–22, 2020.
- [13] K. Kyamakya, "Artificial intelligence in Transportation Telematics," *OGAI J. (Oesterreichische Gesellschaft fuer Artif. Intell.*, vol. 25, no. 3, pp. 2–4, 2006.

- [14] C. D. E. L. A. Red, "CONTROL DE LA RED," pp. 1–9, 2016.
- [15] I. Granada G, "Sistemas inteligentes de transporte para la movilidad universal," *Bid*, 2018.
- [16] S. Henao Pérez, "Sistemas Inteligentes De Transporte (Its)," *Anales de Ingeniería*, vol. 894. p. 28, 2005.
- [17] A. Guin, "an Incident Detection Algorithm Based on a Discrete State Propagation Model of Traffic Flow," no. July, 2004.
- [18] R. Ranjith, J. J. Athanesious, and V. Vaidehi, "Anomaly Detection using DBSCAN Clustering Technique for Traffic Video Surveillance," pp. 1–6, 2019.
- [19] N. Owens *et al.*, "Traffic Incident Management Handbook," *Sci. Appl. Int. Corp. Am. Transp. Res. Inst.*, no. January 2010, pp. 1–116, 2010.
- [20] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. 2017.
- [21] C. Szegedy *et al.*, "Going Deeper with Convolutions," *Des. Track. Knowl. Manag. Metrics*, pp. 163–182, 2019.
- [22] W. Kurdthongmee, "A comparative study of the effectiveness of using popular DNN object detection algorithms for pith detection in cross-sectional images of parawood," *Heliyon*, vol. 6, no. 2, p. e03480, 2020.
- [23] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [24] G. Cheng, P. Zhou, and J. Han, "Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, 2016.
- [25] A. Gautam and S. Singh, "Trends in Video Object Tracking in Surveillance: A Survey," *Proc. 3rd Int. Conf. I-SMAC IoT Soc. Mobile, Anal. Cloud, I-SMAC 2019*, pp. 729–733, 2019.
- [26] T. W. Mi and M. T. Yang, "Comparison of tracking techniques on 360-degree videos," *Appl. Sci.*, vol. 9, no. 16, 2019.
- [27] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate Scale Estimation for Robust Visual Tracking," *BMVC 2014 - Proc. Br. Mach. Vis. Conf. 2014*, 2014.

- [28] J. C. Nascimento, A. J. Abrantes, and J. S. Marques, "Algorithm for centroid-based tracking of moving objects," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 6, no. 1, pp. 3305–3308, 1999.
- [29] M. Veres and M. Moussa, "Deep Learning for Intelligent Transportation Systems: A Survey of Emerging Trends," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3152–3168, 2019.
- [30] U. Khalil, T. Javid, and A. Nasir, "Automatic road accident detection techniques: A brief survey," *2017 Int. Symp. Wirel. Syst. Networks, ISWSN 2017*, vol. 2018-Janua, pp. 1–6, 2017.
- [31] M. S. Shehata *et al.*, "Video-based automatic incident detection for smart roads: The outdoor environmental challenges regarding false alarms," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 349–360, 2008.
- [32] Organización Mundial de la Salud, "Accidentes de tránsito," 2008. .
- [33] K. Yun, H. Jeong, K. M. Yi, S. W. Kim, and J. Y. Choi, "Motion Interaction Field for Accident Detection in Traffic Surveillance Video," 2014.
- [34] U.S. Department of Transportation, "Traffic Incident Response Practices in Europe," no. February. p. 59, 2006.
- [35] A. Ramiro and V. Alvarado, "Machine Learning para Todos Aldo Valdez Alvarado," no. January 2019, 2020.
- [36] K. A. Gaurav and L. Patel, *Machine Learning With R*. 2020.
- [37] T. Mitchell, *Machine Learning*. 2019.
- [38] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*, vol. 9781107057. 2013.
- [39] S. B. Imandoust and M. Bolandraftar, "Application of K-Nearest Neighbor ( KNN ) Approach for Predicting Economic Events : Theoretical Background," *Int. J. Eng. Res. Appl.*, vol. 3, no. 5, pp. 605–610, 2013.
- [40] C. Y. J. Peng, K. L. Lee, and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting," *J. Educ. Res.*, vol. 96, no. 1, pp. 3–14, 2002.
- [41] A. F. Cabrera, "Logistic Regression Analysis in Higher Education: An Applied Perspective \*," no. 814, 2007.

- [42] P. D. Cleary and R. Angel, "The analysis of relationships involving dichotomous dependent variables.," *J. Health Soc. Behav.*, vol. 25, no. 3, pp. 334–348, 1984.
- [43] T. Wen and A. Edelman, "Support Vector Machines: Algorithms and Applications," pp. 249–257, 2003.
- [44] A. Kowalczyk, "Support Vector Machines Succinctly, Syncfusion," *Succinctly E-b. Ser.*, p. 114, 2017.
- [45] A. B. Tucker, *Computer science handbook*. 2004.
- [46] C. Bishop, *Pattern Recognition and Machine Learning*, vol. 27, no. 1. 2004.
- [47] N. H. Farhat, "Photonit neural networks and learning mathines the role of electron-trapping materials," *IEEE Expert. Syst. their Appl.*, vol. 7, no. 5, pp. 63–72, 1992.
- [48] S. Tong and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," *Am. Q.*, vol. 61, no. 2, pp. 417–421, 2009.
- [49] G. H. Rosenfield and K. Fitzpatrick-Lins, "A coefficient of agreement as a measure of thematic classification accuracy.," *Photogramm. Eng. Remote Sens.*, vol. 52, no. 2, pp. 223–227, 1986.
- [50] F. Spoto, P. Martimort, and M. Drusch, "Sentinel - 2: ESA's optical high-resolution mission for GMES operational services," *Eur. Sp. Agency, (Special Publ. ESA SP*, vol. 707 SP, 2012.
- [51] Ramaswami M, "Validating Predictive Performance of Classifier Models for Multiclass Problem in Educational Data Mining," *Int. J. Comput. Sci.*, vol. 11, no. 5, pp. 86–90, 2014.
- [52] D. M. W. Powers, "Evaluation : From Precision , Recall and F-Factor to ROC , Informedness , Markedness & Correlation," no. December, 2007.
- [53] G. Canbek, T. T. Temizel, S. Sagiroglu, and N. Baykal, "Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights," *2nd Int. Conf. Comput. Sci. Eng. UBMK 2017*, pp. 821–826, 2017.
- [54] GOOGLE, "Clasificación: ROC y AUC | Curso intensivo de aprendizaje automático."
- [55] R. Delgado, "Introducción a la Validación Cruzada (k-fold Cross Validation) en R," *R Pubs*. 2018.

- [56] V. Gulisano, M. Papatrantafileou, and A. V Papadopoulos, "Cross Validation," p. 2019, 2019.
- [57] J. Howse, P. Joshi, and M. Beyeler, *OpenCV: Computer Vision Projects with Python*. 2016.
- [58] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, "Object Detection With Partbase," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.
- [59] Y. Xiao *et al.*, *A review of object detection based on deep learning*, vol. 79, no. 33–34. Multimedia Tools and Applications, 2020.
- [60] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016.
- [61] J. Redmon and A. Farhadi, "Yolo V2.0," *Cvpr2017*, no. April, pp. 187–213, 2017.
- [62] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018.
- [63] J. D. Peter and S. L. Fernandes, *Advances in Intelligent Systems and Computing 1167 Intelligence in Big Data Technologies — Beyond the Hype*. 2019.
- [64] W. Liu *et al.*, "SSD: Single shot multibox detector," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 2016.
- [65] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017.
- [66] RSIP VISION, "Seguimiento de objetos en videos." [Online]. Available: <https://www.rsipvision.com/object-tracking-in-videos/#:~:text=Object tracking in videos is,objects%2C both static and dynamic.>
- [67] S. Yadav and S. Payandeh, "Understanding Tracking Methodology of Kernelized Correlation Filter," *2018 IEEE 9th Annu. Inf. Technol. Electron. Mob. Commun. Conf.*, pp. 1330–1336, 2018.
- [68] João F. Henriques, Caseiro Rui, Martins Pedro, and Batista Horge, "Kernelized Correlation Filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, 2015.
- [69] Y. Yang and G. A. Bilodeau, "Multiple Object Tracking with Kernelized Correlation

- Filters in Urban Mixed Traffic,” *Proc. - 2017 14th Conf. Comput. Robot Vision, CRV 2017*, vol. 2018-Janua, pp. 209–216, 2018.
- [70] Y. Lu, Z. Zhou, and J. Zhao, “Visual Object Tracking Using Adaptive Correlation Filters,” pp. 2544–2550, 2018.
- [71] D. S. Bolme, B. A. Draper, and J. R. Beveridge, “Average of synthetic exact filters,” *2009 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. CVPR Work. 2009*, vol. 2009 IEEE, pp. 2105–2112, 2009.
- [72] Z. Kalal, K. Mikolajczyk, and J. Matas, “Forward-backward error: Automatic detection of tracking failures,” *Proc. - Int. Conf. Pattern Recognit.*, pp. 2756–2759, 2010.
- [73] N. V. Chawla, N. Japkowicz, and A. Kotcz, “Editorial: Special Issue on Learning from Imbalanced Data Sets,” *ACM SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 1–6, 2004.
- [74] F. Provost, “Learning with Imbalanced Data,” *AAAI’2000 Work. Imbalanced Data Sets*, no. June, 2018.
- [75] S. Ertekin, J. Huang, L. Bottou, and C. Lee Giles, *Learning on the border: active learning in imbalanced data classification*. 2007.
- [76] B. W. Yap, K. A. Rani, H. A. Abd Rahman, S. Fong, Z. Khairudin, and N. N. Abdullah, “An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets,” *Lect. Notes Electr. Eng.*, vol. 285 LNEE, pp. 13–22, 2014.
- [77] R. Mohammed, J. Rawashdeh, and M. Abdullah, “Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results,” *2020 11th Int. Conf. Inf. Commun. Syst. ICICS 2020*, no. April, pp. 243–248, 2020.
- [78] B. Efron, *An Introduction to the Bootstrap*. Washington, DC: Chapman & Hall/CRC, 1967.
- [79] M. Kuhn and K. Johnson, *Applied Predictive Modeling with Applications in R*. 2013.
- [80] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, “Systematic literature reviews in software engineering - A systematic literature review,” *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, 2009.
- [81] M. Riveiro, M. Lebram, and M. Elmer, “Anomaly Detection for Road Traffic : A Visual Analytics Framework,” pp. 1–11, 2017.
- [82] M. M. L. Elahi, R. Yasir, M. A. Syrus, S. Q. Z. Nine, I. Hossain, and N. Ahmed,

- “Computer Vision Based Road Traffic Accident and Anomaly Detection in the Context of Bangladesh,” 2014.
- [83] Y. Chen, Y. Yu, and T. Li, “A Vision based Traffic Accident Detection Method Using Extreme Learning Machine,” 2016.
- [84] F. Jiansheng, “Vision-based real-time traffic accident detection,” pp. 1035–1038, 2014.
- [85] V. R. C, N. S. B, and R. Madhavan, “Vision based Accident Vehicle Identification and Scene Investigation,” 2017.
- [86] D. Singh, S. Member, and C. K. Mohan, “Deep Spatio-Temporal Representation for Detection of Road Accidents Using Stacked Autoencoder,” pp. 1–9, 2018.
- [87] A. Saini, S. Suregaonkar, N. Gupta, V. Karar, and S. Poddar, “Region and Feature Matching based Vehicle Tracking for Accident Detection,” no. August, pp. 10–12, 2017.
- [88] M. Kilicarslan, S. Member, J. Y. Zheng, and S. Member, “Direct Vehicle Collision Detection from Motion in Driving Video,” no. Iv, pp. 1558–1564, 2017.
- [89] C. Jun, W. Yilin, and L. Baoxin, “REAL-TIME VEHICLE BACK-UP WARNING SYSTEM WITH A SINGLE CAMERA 4600 S . Dobson Road,” pp. 2275–2279, 2015.
- [90] Y. Akram, A. Jarouf, and M. Sc, “A Hybrid Method To Detect And Verify Vehicle Crash With Haar-Like Features And SVM Over The Web,” *2018 Int. Conf. Comput. Appl.*, pp. 177–182.
- [91] I. J. Lee, “Big data processing framework of learning weather information and road traffic collision using distributed CEP from CCTV video: Cognitive image processing,” *Proc. 2017 IEEE 16th Int. Conf. Cogn. Informatics Cogn. Comput. ICCI\*CC 2017*, pp. 400–406, 2017.
- [92] A. Pashaei, M. Ghatee, and H. Sajedi, “Convolution neural network joint with mixture of extreme learning machines for feature extraction and classification of accident images,” *J. Real-Time Image Process.*, vol. 17, no. 4, pp. 1051–1066, 2020.
- [93] P. A. Mateus and C. L. Nino, “Adaptive algorithm to identify anomalies in moving objects using computer vision,” *Proc. 2016 IEEE ANDESCON, ANDESCON 2016*, 2017.
- [94] D. Pęszor, M. Paszkuta, M. Wojciechowska, and K. Wojciechowski, “Optical Flow for

- Collision Avoidance in Autonomous Cars,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10752 LNAI, pp. 482–491, 2018.
- [95] G. Lucarelli and M. Borrotti, “A Visual Neural Network for Robust Collision Perception in Vehicle Driving Scenarios,” vol. 2, no. 691154, pp. 247–258, 2019.
- [96] B. Maaloul, A. Taleb-Ahmed, S. Niar, N. Harb, and C. Valderrama, “Adaptive video-based algorithm for accident detection on highways,” *2017 12th IEEE Int. Symp. Ind. Embed. Syst. SIES 2017 - Proc.*, 2017.
- [97] R. Kaviani, P. I. Ahmadi, and I. Gholampour, “Automatic accident detection using topic models,” *ICEE 2015 - Proc. 23rd Iran. Conf. Electr. Eng.*, vol. 10, pp. 444–449, 2015.
- [98] H. Li, Y. Zhang, M. Yang, Y. Men, and H. Chao, “A rapid abnormal event detection method for surveillance video based on a novel feature in compressed domain of HEVC,” *Proc. - IEEE Int. Conf. Multimed. Expo.*, vol. 2014-Septe, no. September, 2014.
- [99] S. Xia, J. Xiong, Y. Liu, and G. Li, “Vision-based traffic accident detection using matrix approximation,” *2015 10th Asian Control Conf. Emerg. Control Tech. a Sustain. World, ASCC 2015*, 2015.
- [100] L. Wang and D. Ming, “REAL-TIME DETECTION OF ABNORMAL CROWD BEHAVIOR USING A MATRIX APPROXIMATION-BASED APPROACH Lijun Wang and Ming Dong Department of Computer Science Wayne State University Detroit , MI 48202 , USA,” *Science (80-. )*, pp. 2701–2704, 2012.
- [101] J. Ren, Y. Chen, L. Xin, J. Shi, B. Li, and Y. Liu, “Detecting and positioning of traffic incidents via video-based analysis of traffic states in a road segment,” *IET Intell. Transp. Syst.*, vol. 10, no. 6, pp. 428–437, 2016.
- [102] H. Ullah, U. Mohib, A. Hina, C. Nicola, and N. Francesco, “TRAFFIC ACCIDENT DETECTION THROUGH A HYDRODYNAMIC LENS,” *Int. Conf. Image Process.*, pp. 2470–2474, 2015.
- [103] X. Mo, V. Monga, R. Bala, and Z. Fan, “Adaptive sparse representations for video anomaly detection,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 4, pp. 631–645, 2014.
- [104] Z. Zhao, X. Lang, W. Wang, and G. Xiong, “Traffic detection algorithm based on outlier mining,” *ACM Int. Conf. Proceeding Ser.*, no. 37, pp. 40–43, 2018.

- [105] P. J. Denning, "Is Computer Science Science?," pp. 27–31, 2005.
- [106] D. Kibler, "Paul R. Cohen's Empirical Methods for Artificial Intelligence," *Artif. Intell.*, vol. 113, no. 1–2, pp. 281–284, 1999.
- [107] R. Jain, H. C. Triandis, and C. Weick, *MANAGING RESEARCH, DEVELOPMENT, AND INNOVATION*, vol. 51, no. 4. 2008.
- [108] M. M. Ayash, "Research Methodologies in Computer Science and Information Systems," *Comput. Sci.*, vol. 2014, pp. 1–4, 2014.
- [109] R. O' Brien, "Overview of Action Research Methodology," 1998. [Online]. Available: [https://www.web.ca/~robrien/papers/arfinal.html#\\_edn5](https://www.web.ca/~robrien/papers/arfinal.html#_edn5).
- [110] D. Gabel, "An Introduction to Action Research," *National Association for Research in Science Teaching*, no. 24 April. pp. 1–4, 1995.
- [111] K. Lewin, "Action Research and Minority Problems.," *J. Soc. Issues*, no. 2, pp. 34–46, 1946.
- [112] T. Gilmore, R. Ramirez, and J. Krantz, "Action-Based Modes of Inquiry and the Host-Researcher Relationship," in *Action Based Modes of Inquiry and the Host-Researcher Relationship*, vol. 5, no. 3, 1986, p. 9.
- [113] M. Robson, *Action research: principles and practice*, vol. 13, no. 3. 2016.
- [114] T. Lee, "GitHub - taehoonlee\_tensornets\_ High level network definitions with pre-trained weights in TensorFlow." .
- [115] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," 2016.
- [116] Y. Jia *et al.*, "Caffe: Convolutional Architecture for Fast Feature Embedding," pp. 675–678, 2014.
- [117] OpenCv, "OpenCV." [Online]. Available: <https://opencv.org/>.
- [118] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, 2009.
- [119] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 39, no. 2014, pp. i–ii, 2014.
- [120] D. Arnette, S. Ranganathan, A. Borulkar, S.-H. Lin, and S. Rao, "NetApp ONTAP AI , Powered by NVIDIA Scalable AI Infrastructure : Designing for Real-World Deep

- Learning Use Cases,” no. August. 2018.
- [121] W. Sultani, C. Chen, and M. Shah, “Real-world Anomaly Detection in Surveillance Videos,” Jan. 2018.
- [122] A. Shah, J. B. Lamare, T. N. Anh, and A. Hauptmann, “CADP: A Novel Dataset for CCTV Traffic Camera based Accident Analysis,” Sep. 2018.
- [123] T. Davis, “Desktop Tools and Development Environment,” *MATLAB Prim. Eighth Ed.*, pp. 165–167, 2010.
- [124] B. Cannon, “Python Developer’s Guide Documentation.” 2020.
- [125] M. S. Rajib, J. Ahmed, M. M. Rahman, W. M. Abdullah, and N. S. Chauhan, “A proposed system to estimate the velocity of vehicles from traffic video,” *Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC 2017*, pp. 320–324, 2017.
- [126] Amazon Web Services, “What is data labeling?” [Online]. Available: <https://aws.amazon.com/sagemaker/groundtruth/what-is-data-labeling/>.
- [127] T. Kulesza, S. Amershi, R. Caruana, D. Fisher, and D. Charles, “Structured labeling to facilitate concept evolution in machine learning,” *Conf. Hum. Factors Comput. Syst. - Proc.*, pp. 3075–3084, 2014.
- [128] RAPIDS, “gpuCI Usage Docs - RAPIDS Docs.” [Online]. Available: <https://docs.rapids.ai/>.
- [129] S. Y. Yerima, S. Sezer, and G. McWilliams, “Analysis of Bayesian classification-based approaches for Android malware detection,” *IET Inf. Secur.*, vol. 8, no. 1, pp. 25–36, 2014.
- [130] S. Schaal, S. Vijayakumar, and C. G. Atkeson, *Encyclopedia of Machine Learning and Data Mining*. 1998.
- [131] B. S. Shobha and R. Deepu, “A Review on Video Based Vehicle Detection, Recognition and Tracking,” *Proc. 2018 3rd Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. CSITSS 2018*, pp. 183–186, 2018.
- [132] J. R. B. Del Rosario, A. A. Bandala, and E. P. Dadios, “Multi-view multi-object tracking in an intelligent transportation system: A literature review,” *HNICEM 2017 - 9th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag.*, vol. 2018-Janua, pp. 1–4, 2017.

## 8. ANEXOS

### Anexo 1 - Preguntas de investigación, cadenas de búsqueda y criterios de inclusión y exclusión de artículos científicos

- Preguntas de investigación

General

¿ Qué métodos de clasificación han sido utilizados para la detección de incidentes de vehículos en videos de tráfico?

Específicas

¿Qué métodos de detección y seguimiento de objetos han sido aplicados en videos de tráfico?

¿Qué bases de datos públicas de videos de tráfico de vehículos existen?

¿Cuál es la técnica utilizada para evaluar el desempeño del algoritmo?

- Cadenas de búsqueda

Cadena 1:

“vehicle” AND “accident” AND “detection”

Cadena 2:

“car” AND “collision” AND “detection”

Cadena 3:

“traffic” AND “accident” AND “video” AND “detection”

- Lista de criterios de inclusión y exclusión

Inclusión	Exclusión
Proponen algoritmos o modelos de detección de choques de vehículos, por medio de la lectura del título y el abstract	Trabajos publicados con años anteriores a 2014
Journal/congreso/capítulo revisado por pares	Trabajos que no están disponibles

El modelo se realiza con videos reales ya sea en tiempo real o no, por medio de la lectura del título y el abstract	Trabajos que no estén en ingles
---	---------------------------------

- Lista de criterios de evaluación de calidad, con el universo de valores para cada criterio. Ayudarse en la Escala de Likert para definir los valores

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0

- Bases de datos donde va a realizar la búsqueda de trabajos

IEEE Xplore

Science Direct

Springer Link

ACM

- Aplicación de cada cadena de búsqueda a cada base de datos definida y resultados por cadena y por base de datos

Consideraciones en las búsquedas en las bases de datos

En SpringerLink se ha buscado en la opción: Advanced Search -> Find Resources -> with all of the words -> Ejemplo de cadena: 'car AND collision AND detection'

En Science Direct se ha buscado en la opción: Advanced Search -> Title, abstract or author-specified keywords -> Ejemplo de cadena: car collision detection

En IEEE Xplore se ha buscado en la opción: Advanced Search -> Ejemplo de cadena: (((car) AND collision) AND detection)

En ACM se ha buscado en la opción: Advanced Search -> Ejemplo de cadena: (+car +collision +detection)

Cadena 1: "vehicle" AND "accident" AND "detection"

<b>Nombre de la bases de datos</b>	<b>Resultados</b>
IEEE Xplore	1947
Science Direct	173
Springer Link	15573
ACM	65
<b>Total</b>	<b>17758</b>

Cadena 2: "car" AND "collision" AND "detection"

<b>Nombre de la bases de datos</b>	<b>Resultados</b>
IEEE Xplore	325

Science Direct	55
Springer Link	12847
ACM	20
<b>Total</b>	<b>13247</b>

Cadena 3: “traffic” AND “accident” AND “video” AND “detection”

<b>Nombre de la bases de datos</b>	<b>Resultados</b>
IEEE Xplore	264
Science Direct	21
Springer Link	4353
ACM	3
<b>Total</b>	<b>4641</b>

- Aplicación de los criterios de inclusión y exclusión a las cadenas de búsqueda y a las bases de datos

Cadena 1: “vehicle” AND “accident” AND “detection”

<b>Nombre de la bases de datos</b>	<b>Resultados</b>
IEEE Xplore	21
Science Direct	1
Springer Link	1
ACM	0
<b>Total</b>	<b>23</b>

Cadena 2: “car” AND “collision” AND “detection”

<b>Nombre de la bases de datos</b>	<b>Resultados</b>
IEEE Xplore	1
Science Direct	0
Springer Link	2
ACM	0
<b>Total</b>	<b>3</b>

Cadena 3: “traffic” AND “accident” AND “video” AND “detection”

Nombre de la bases de datos	Resultados
IEEE Xplore	9
Science Direct	0
Springer Link	0
ACM	1
<b>Total</b>	<b>10</b>

Se tiene un total de 37 papers que pasaron los criterios de inclusión y exclusión y pasan a la etapa de criterios de calidad (fondo, forma)

- Calificación tomando en cuenta los criterios de calidad a cada uno de los artículos que pasaron la fase de los criterios de inclusión / exclusión.

Cadena 1: "vehicle" AND "accident" AND "detection"

IEEE Xplore

Vision-based real-time traffic accident detection

Zu hui ; Xie yaohua ; Ma lu ; Fu Jiansheng

Proceeding of the 11th World Congress on Intelligent Control and Automation

Year: 2014 | Conference Paper | Publisher: IEEE

### **Criterios de calidad**

#### Criterios de forma:

El artículo científico consta por lo menos de las siguientes secciones: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

#### Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3	2 x	1

#### Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 11**

An Implementation of Deep Learning based IoV System for Traffic Accident Collisions Detection with an Emergency Alert Mechanism  
 Liang-Bi Chen ; Ke-Yu Su ; Yu-Ching Mo ; Wan-Jung Chang ; Wei-Wen Hu ; Jing-Jou Tang ; Chao-Tang Yu  
 2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3	2 x	1

(porque no tiene conclusión)

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3	2	1 x

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3	2 x	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3	2 x	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 7**

Vision based accident vehicle identification and scene investigation

C. Vipin Raj ; B. Nithya Sree ; Raj Madhavan

2017 IEEE Region 10 Symposium (TENSYMP)

Year: 2017 | Conference Paper | Publisher: IEEE

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
------	----------------------------	---------------------------	-----------------

Valor	3	2 x	1
-------	---	-----	---

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3	2 x	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 11**

Intelligent traffic accident detection system based on mobile edge computing  
 Chunxiao Liao ; Guochu Shou ; Yaqiong Liu ; Yihong Hu ; Zhigang Guo  
 2017 3rd IEEE International Conference on Computer and Communications (ICCC)

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3	2 x	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3	2 x	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 10**

Deep Spatio-Temporal Representation for Detection of Road Accidents Using Stacked Autoencoder

Dinesh Singh ; Chalavadi Krishna Mohan

IEEE Transactions on Intelligent Transportation Systems

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
------	-------------------------	------------------------	---------------

Valor	3 x	2	1
-------	-----	---	---

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 13**

Computer vision based road traffic accident and anomaly detection in the context of Bangladesh

Mirza M. Lutfe Elahi ; Rahat Yasir ; Minhaz Ahmed Syrus ; Md. S. Q. Zulkar Nine ; Ishtiaque Hossain ; Nova Ahmed

2014 International Conference on Informatics, Electronics & Vision (ICIEV)

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3	2 x	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 12**

Collision detection and avoidance system for vehicle

Saurav Agrawal ; S. W. Varade

2017 2nd International Conference on Communication and Electronics Systems (ICCES)

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3	2	1 x

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3	2 x	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3	2 x	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 8**

Region and feature matching based vehicle tracking for accident detection  
 Abhinav Saini ; Sameer Suregaonkar ; Neena Gupta ; Vinod Karar ; Shashi Poddar  
 2017 Tenth International Conference on Contemporary Computing (IC3)

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3	2 x	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 12**

Event Detection on Roads Using Perceptual Video Summarization  
Sinnu Susan Thomas ; Sumana Gupta ; Venkatesh K. Subramanian  
IEEE Transactions on Intelligent Transportation Systems

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 13**

A vision based traffic accident detection method using extreme learning machine

Yu Chen ; Yuanlong Yu ; Ting Li

2016 International Conference on Advanced Robotics and Mechatronics (ICARM)

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 13**

Direct vehicle collision detection from motion in driving video

Mehmet Kilicarslan ; Jiang Yu Zheng

2017 IEEE Intelligent Vehicles Symposium (IV)

**Criterios de calidad**Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

## Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

## Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

## Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

## Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 12****Design of Intelligent Road Recognition and Warning System for Vehicles Based on Binocular Vision**

Zidong Han ; Junyu Liang ; Jianbang Li

IEEE Access

Year: 2018 | Volume: 6 | Journal Article | Publisher: IEEE

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 12**

**Motion Interaction Field for Accident Detection in Traffic Surveillance Video**

Kimin Yun ; Hawook Jeong ; Kwang Moo Yi ; Soo Wan Kim ; Jin Young Choi

2014 22nd International Conference on Pattern Recognition

Year: 2014 | Conference Paper | Publisher: IEEE

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 13**

**Real-time vehicle back-up warning system with a single camera**

Jun Cao ; Yilin Wang ; Baoxin Li

2015 IEEE International Conference on Image Processing (ICIP)

Year: 2015 | Conference Paper | Publisher: IEEE

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 13**

**A Hybrid Method to Detect and Verify Vehicle Crash with Haar-Like Features and SVM Over the Web**

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3	2 x	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 11**

**Anomaly detection using DBSCAN clustering technique for traffic video surveillance**

R. Ranjith ; J. Joshan Athanesious ; V. Vaidehi

2015 Seventh International Conference on Advanced Computing (ICoAC)

Year: 2015 | Conference Paper | Publisher: IEEE

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
------	----------------------	---------------	-----------------------

Valor	3 x	2	1
-------	-----	---	---

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 13**

**Anomaly Detection for Road Traffic: A Visual Analytics Framework**

Maria Riveiro ; Mikael Lebram ; Marcus Elmer

IEEE Transactions on Intelligent Transportation Systems

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 12**

**Automatic incident detection (AID) with thermal cameras**

Eddy Vermeulen

Road Transport Information and Control Conference 2014 (RTIC 2014)

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3	2 x	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3	2	1 x

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
------	----------------------------	---------------------------	-----------------

Valor	3	2	1 x
-------	---	---	-----

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3	2	1 x

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 5**

**Big data processing framework of learning weather information and road traffic collision using distributed CEP from CCTV video: Cognitive image processing**

In Jung Lee

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 12**

Science direct

Comparison of Tracking Techniques on 360-Degree Videos

Principles, Algorithms, Applications, Learning 2018, Pages 663-720

Tzu-Wei Mi and Mau-Tsuen Yang

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 12**

Springer link

Para la búsqueda en esta base de datos se lo hizo de la siguiente manera con los filtros que proporciona la página web:

Refine Your Search

Content Type: Article

Discipline: Computer Science

Subdiscipline: Image Processing and Computer Vision

**Convolution neural network joint with mixture of extreme learning machines for feature extraction and classification of accident images**

Ali Pashaei, Mehdi Ghatee, Hedieh Sajedi in Journal of Real-Time Image Processing (2019)

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 13**

ACM

No existe artículos que interesen al tema.

Cadena 2: “car” AND “collision” AND “detection”

IEEE Xplore

**Adaptive algorithm to identify anomalies in moving objects using computer vision**

Paola A. Mateus ; Cesar L. Nino

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 12**

Science direct

No existen artículos de interés para esta cadena

Springer link

Se realizó la siguiente búsqueda con los siguientes filtros.

Refine Your Search

**Content Type: Chapter, Conference Paper**

**Discipline: Computer Science**

**Subdiscipline: Image Processing and Computer Vision**

**Language: English**

**Optical Flow for Collision Avoidance in Autonomous Cars**

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3	2 x Por el hardware	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 12**

## A Visual Neural Network for Robust Collision Perception in Vehicle Driving Scenarios

### Criterios de calidad

#### Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

#### Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

#### Criterios de fondo:

#### Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

#### Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

#### Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 13**

ACM

No existen papers de interés para esta cadena de búsqueda

Cadena 3: "traffic" AND "accident" AND "video" AND "detection"

IEEE Xplore

**Vision-based real-time traffic accident detection**

Zu hui ; Xie yaohua ; Ma lu ; Fu Jiansheng

Proceeding of the 11th World Congress on Intelligent Control and Automation

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3	2 x	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 11**

**Adaptive video-based algorithm for accident detection on highways**

Boutheina Maaloul ; Abdelmalik Taleb-Ahmed ; Smail Niar ; Naim Harb ; Carlos Valderrama

2017 12th IEEE International Symposium on Industrial Embedded Systems (SIES)

**Criterios de calidad**Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

## Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

## Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3	2 x	1

## Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3	2 x	1

## Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 11**

**Deep Spatio-Temporal Representation for Detection of Road Accidents Using Stacked Autoencoder**

Dinesh Singh ; Chalavadi Krishna Mohan

IEEE Transactions on Intelligent Transportation Systems

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 13**

**Automatic accident detection using topic models**

Razie Kaviani ; Parvin Ahmadi ; Iman Gholampour

2015 23rd Iranian Conference on Electrical Engineering

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 13**

**A rapid abnormal event detection method for surveillance video based on a novel feature in compressed domain of HEVC**

Huang Li ; Yihao Zhang ; Ming Yang ; Yangyang Men ; Hongyang Chao

2014 IEEE International Conference on Multimedia and Expo (ICME)

**Criterios de calidad**Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

## Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

## Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

## Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

## Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 13****Vision-based traffic accident detection using matrix approximation**

Siyu Xia ; Jian Xiong ; Ying Liu ; Gang Li

2015 10th Asian Control Conference (ASCC)

**Criterios de calidad**Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 12**

**Detecting and positioning of traffic incidents via video-based analysis of traffic states in a road segment**

Jianqiang Ren ; Yangzhou Chen ; Le Xin ; Jianjun Shi ; Baotong Li ; Yinan Liu

IET Intelligent Transport Systems

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 12**

### **Traffic accident detection through a hydrodynamic lens**

Habib Ullah ; Mohib Ullah ; Hina Afridi ; Nicola Conci ; Francesco G. B. De Natale  
2015 IEEE International Conference on Image Processing (ICIP)

#### **Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas			
Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3	2 x Se necesitan los mismos videos del dataset	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 12**

**Adaptive Sparse Representations for Video Anomaly Detection**

Xuan Mo ; Vishal Monga ; Raja Bala ; Zhigang Fan

IEEE Transactions on Circuits and Systems for Video Technology

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas			
Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1 x	0

**Total: 13**

Science direct

No existen papers de interes para esta cadena de busqueda

Springer link

Para esta base de datos se lo hizo con los siguientes filtros:

Refine Your Search

**Content Type: Article**

**Discipline: Computer Science**

**Subdiscipline: Computer Science, general**

**Language: English**

No existen papers de interés para esta cadena de búsqueda

ACM

Traffic detection algorithm based on outlier mining  
Zairang Zhao, Xianmei Lang, Wenzhi Wang, Guixi Xiong  
May 2018 ICBDT '18: Proceedings of 2018 International Conference on Big Data Technologies

**Criterios de calidad**

Criterios de forma:

El artículo científico consta de: abstract, introducción, método(s), resultados, conclusión, referencias bibliográficas.

Ítem	Completa	Parcialmente completa	Incompleta
Valor	3 x	2	1

Numero referencias bibliográficas

Ítem	Más de 8 referencias	8 referencias	Menor a 8 referencias
Valor	3 x	2	1

Criterios de fondo:

Metodología reproducible

Ítem	Completamente reproducible	Parcialmente reproducible	No reproducible
Valor	3 x	2	1

Detalle del modelo propuesto

Ítem	Completamente detallado	Parcialmente detallado	No se detalla
Valor	3 x	2	1

Tipo de base de datos usada para las pruebas

Ítem	Menciona	No menciona
Valor	1	0 x

**Total: 12**

La mayor calificación que un paper puede obtener es de 13, por tanto un paper pasara los criterios de calidad si este tiene una calificación mayor al 80% de la calificación máxima, en este caso mayor a 10.4 o mayor a 10.

Con lo anteriormente expuesto se obtiene un total de 28 papers para su posterior lectura.

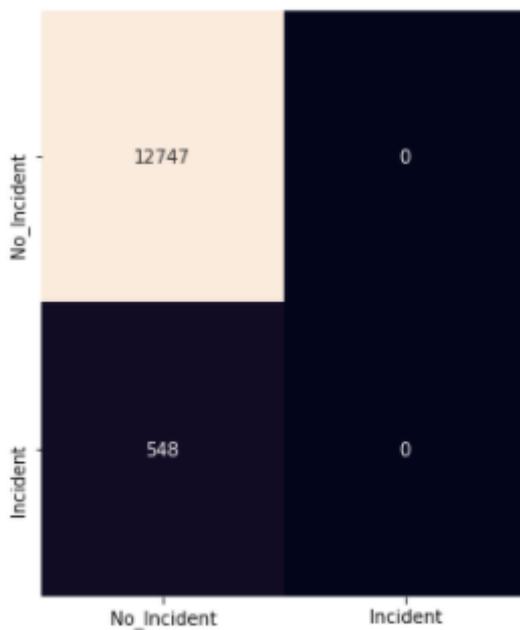
Nota: estas búsquedas se realizaron en el mes de mayo del 2019 y se obtuvieron los datos anteriormente mencionados.

## **Anexo 2 - Matrices de confusión con desbalance de datos**

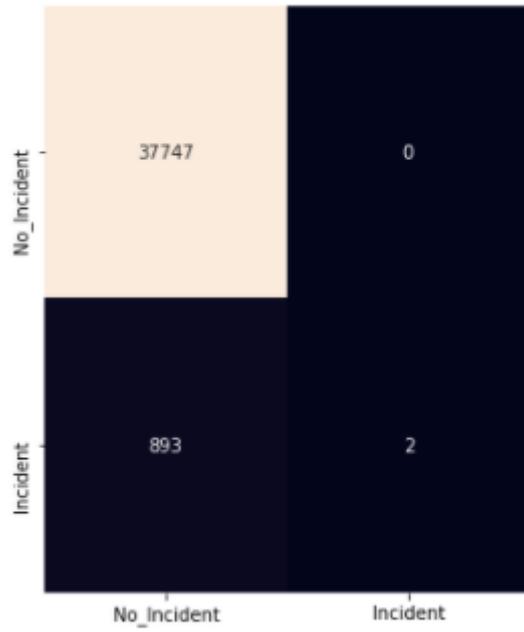
Algoritmo de detección de objetos: SSD Mobilinet

Modelo kNN

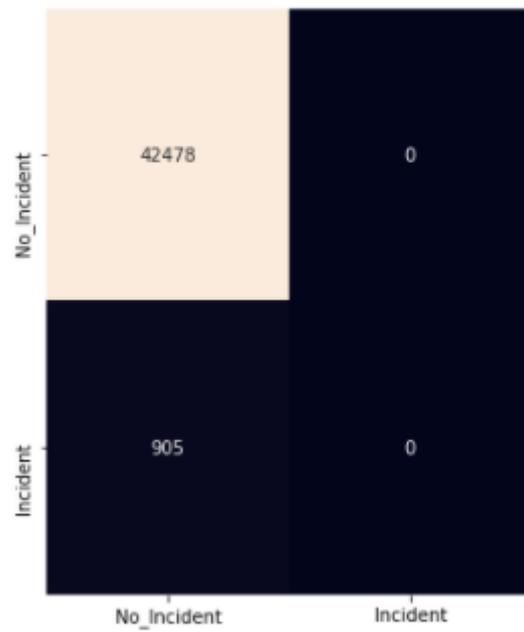
Algoritmo de seguimiento: KCF



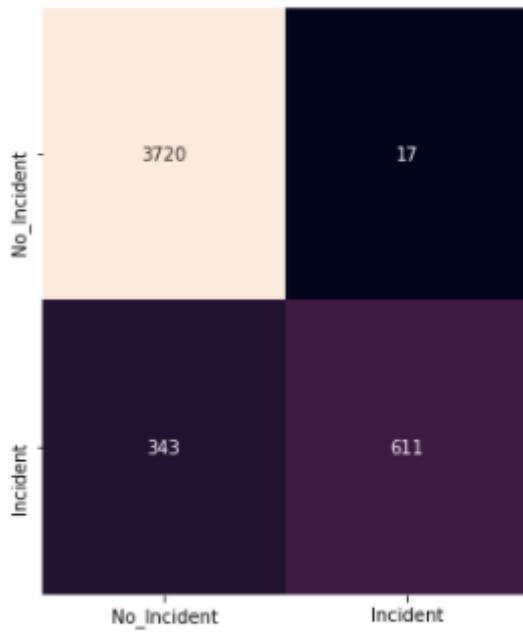
Algoritmo de seguimiento: DLIB



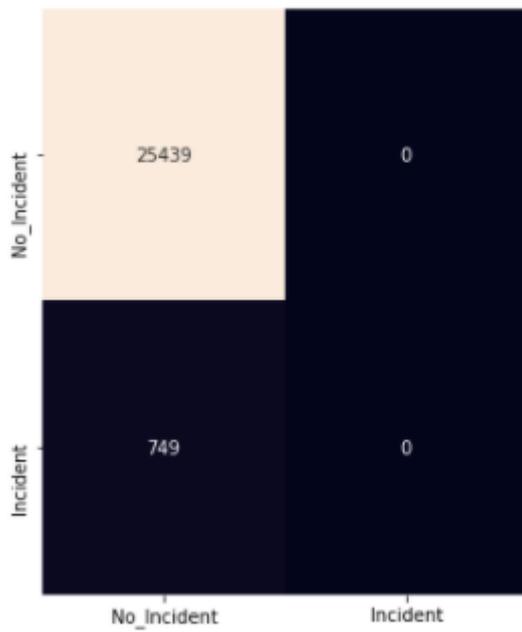
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

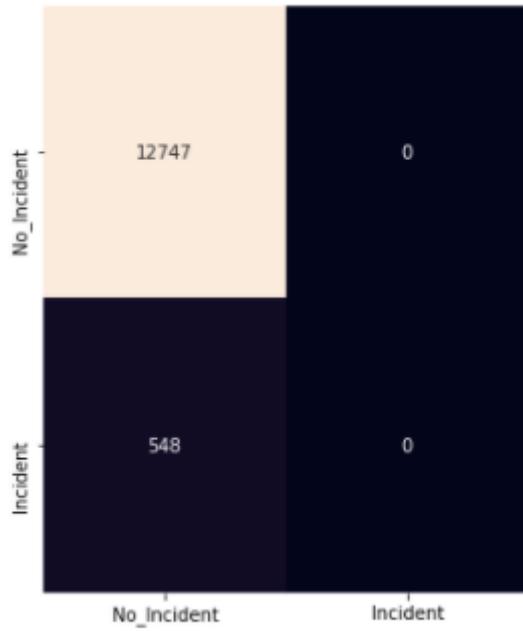


Algoritmo de seguimiento: centroide

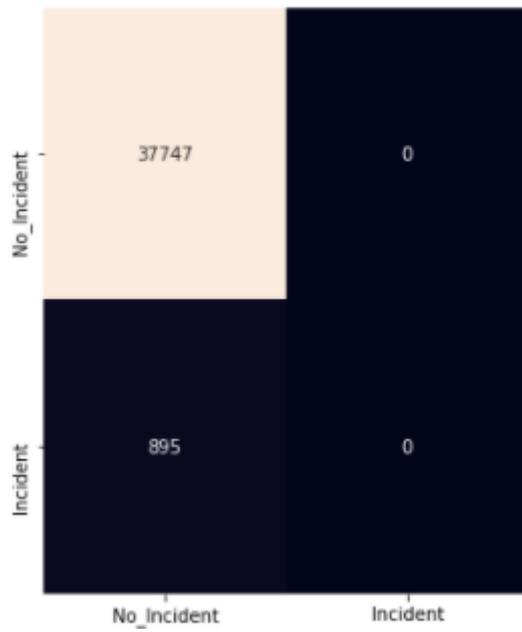


Modelo Regresión Logística

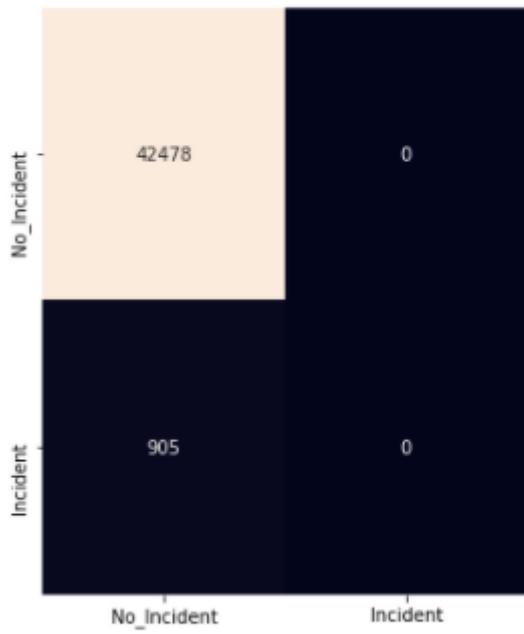
Algoritmo de seguimiento: KCF



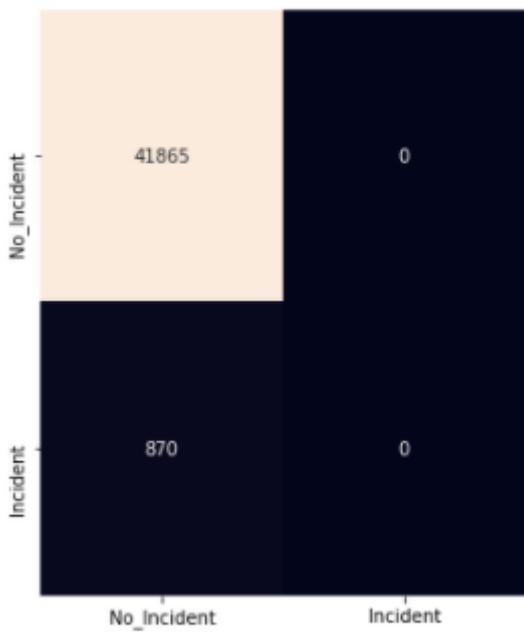
Algoritmo de seguimiento: DLIB



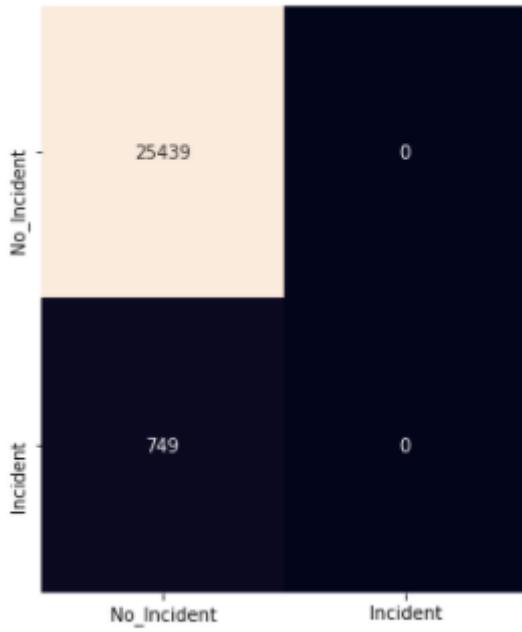
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

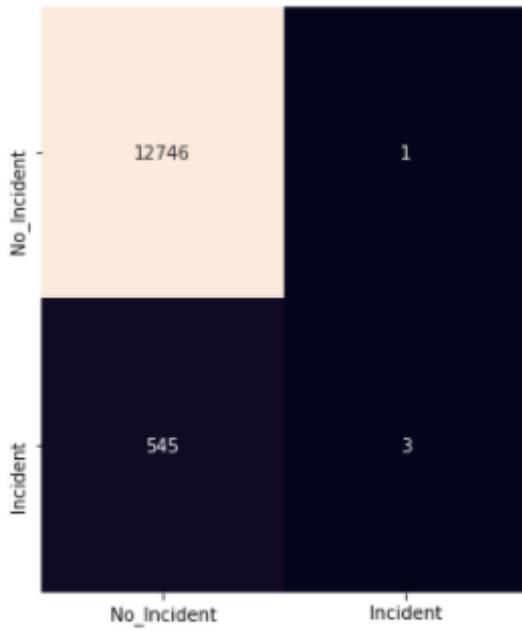


Algoritmo de seguimiento: centroide

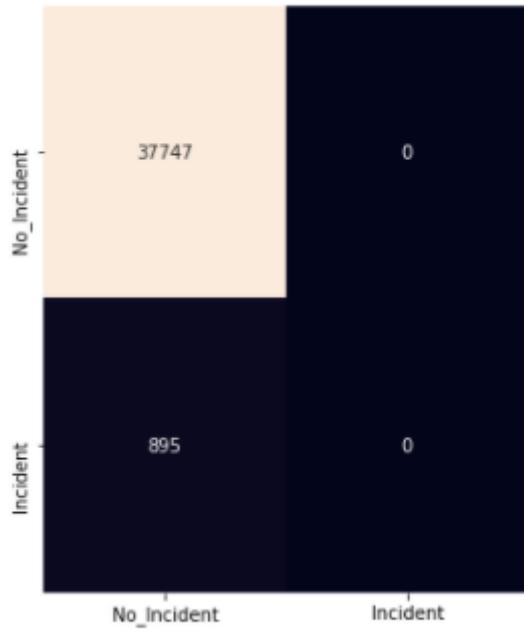


Modelo SVM (núcleo: POLY)

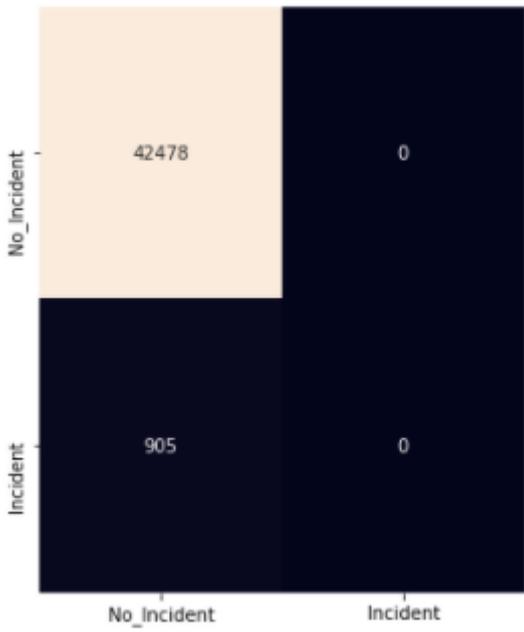
Algoritmo de seguimiento: KCF



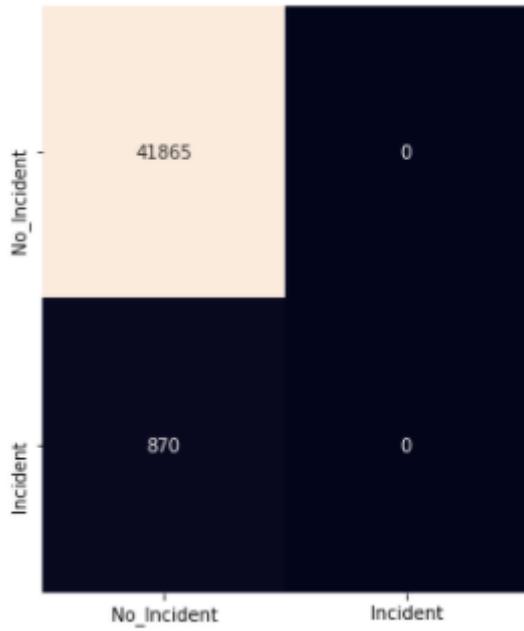
Algoritmo de seguimiento: DLIB



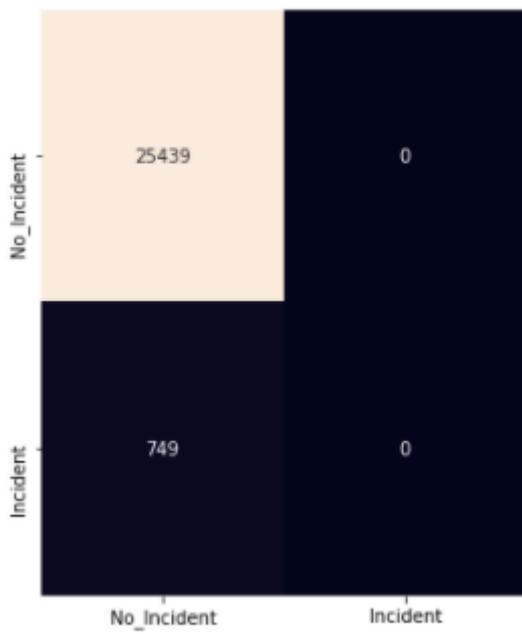
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

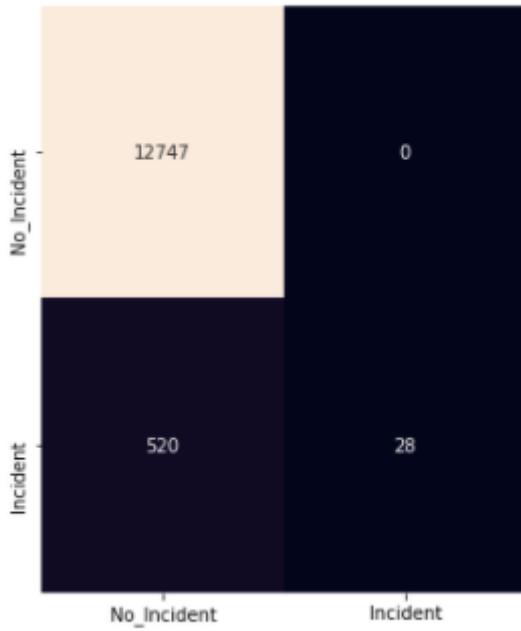


Algoritmo de seguimiento: centroide

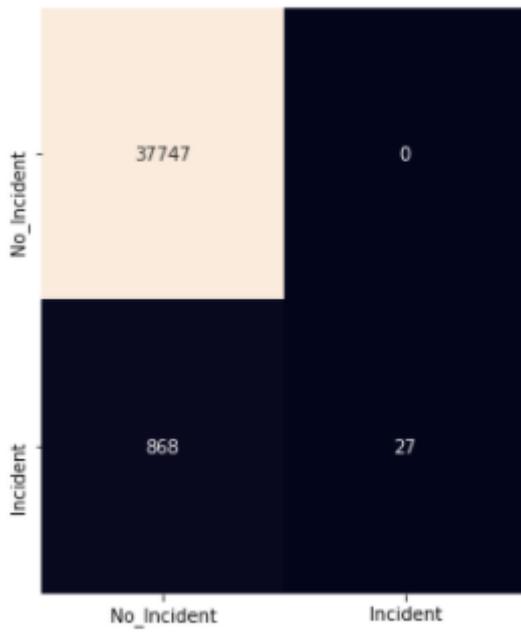


Modelo SVM (núcleo: RBF)

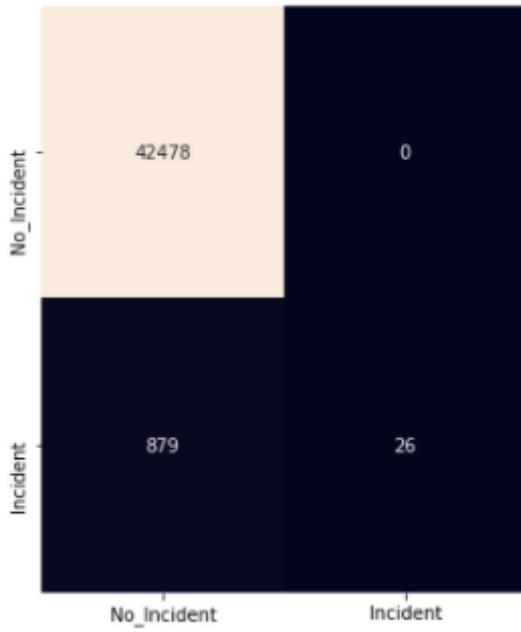
Algoritmo de seguimiento: KCF



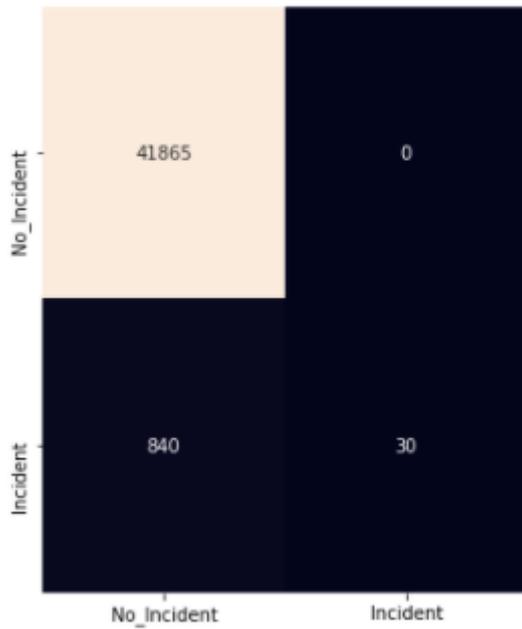
Algoritmo de seguimiento: DLIB



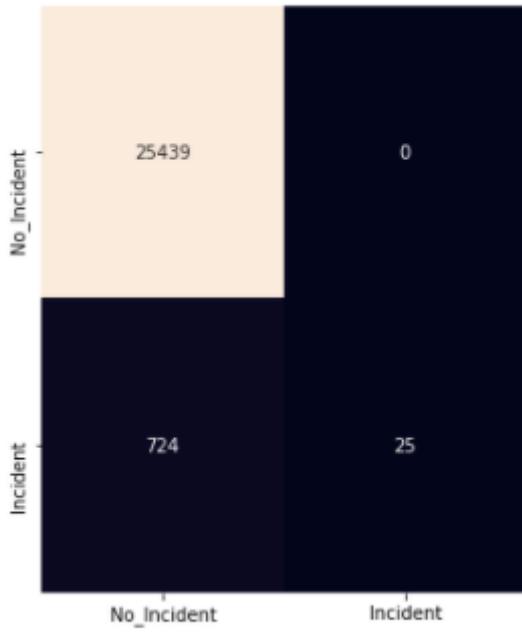
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

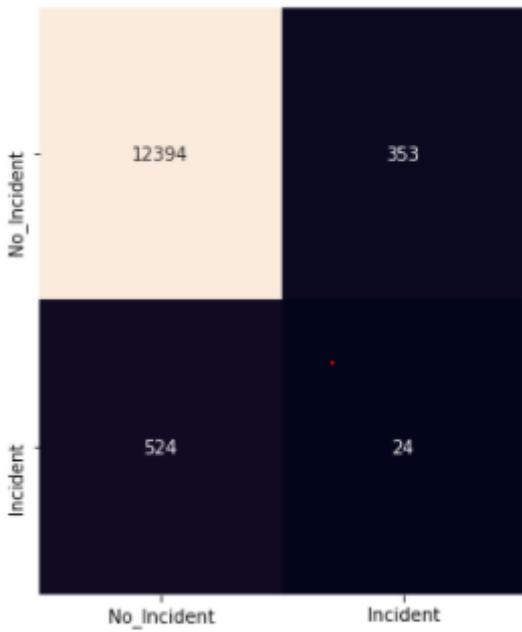


Algoritmo de seguimiento: centroide

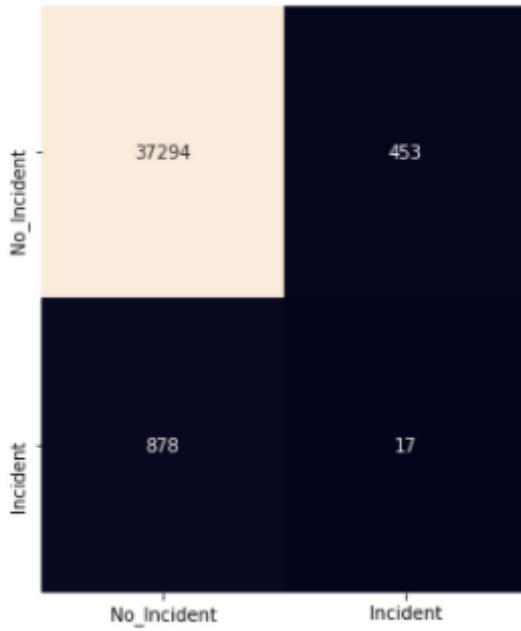


Modelo SVM (núcleo: SIGMOID)

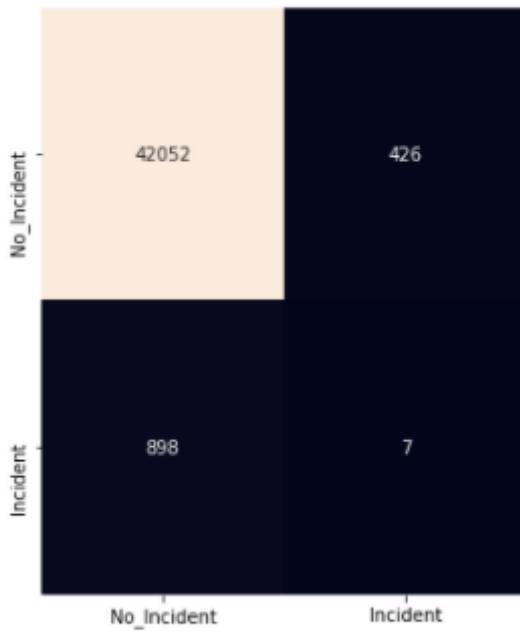
Algoritmo de seguimiento: KCF



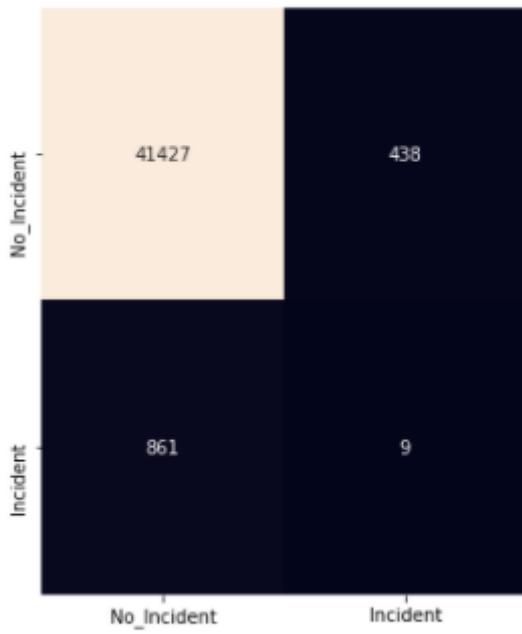
Algoritmo de seguimiento: DLIB



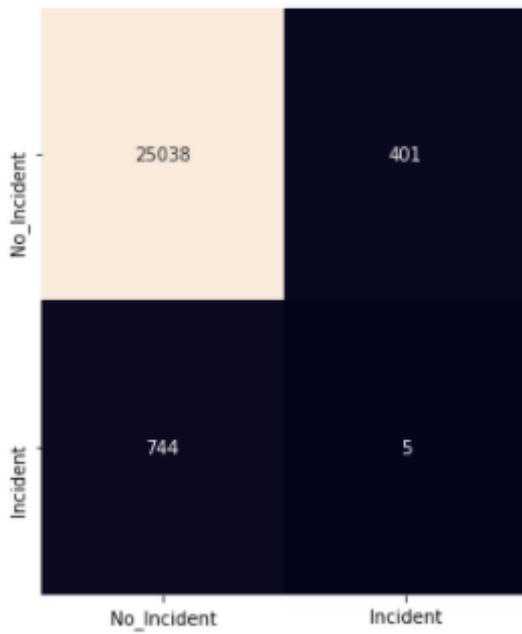
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow



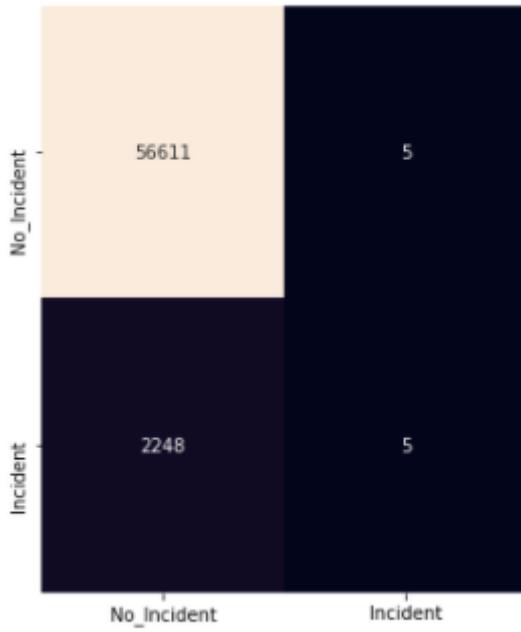
Algoritmo de seguimiento: centroide



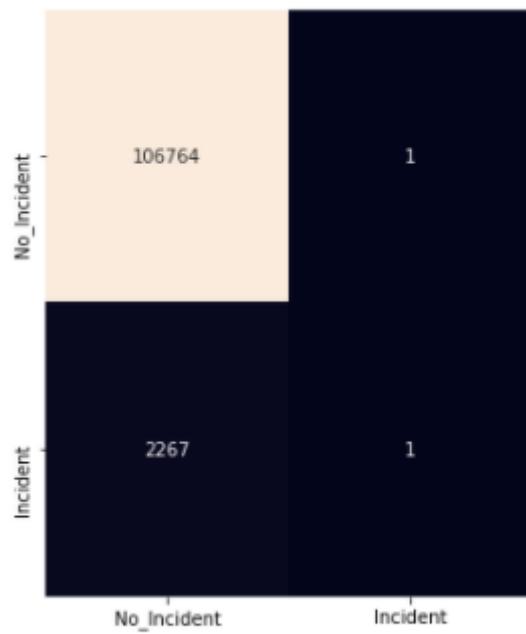
Algoritmo de deteccion de objetos: YOLO v3

Modelo kNN

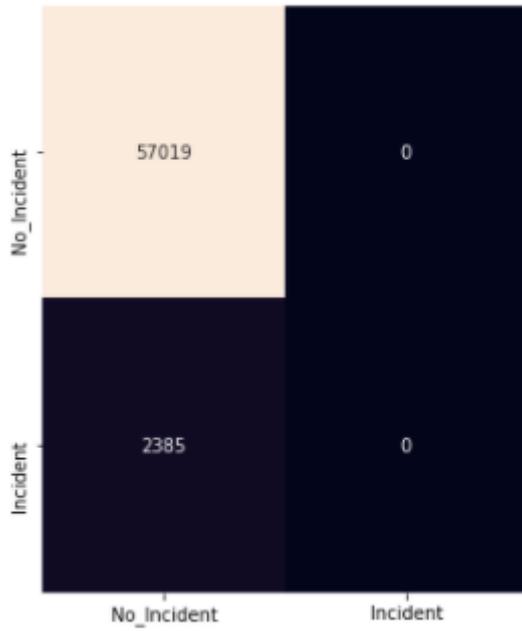
Algoritmo de seguimiento: KCF



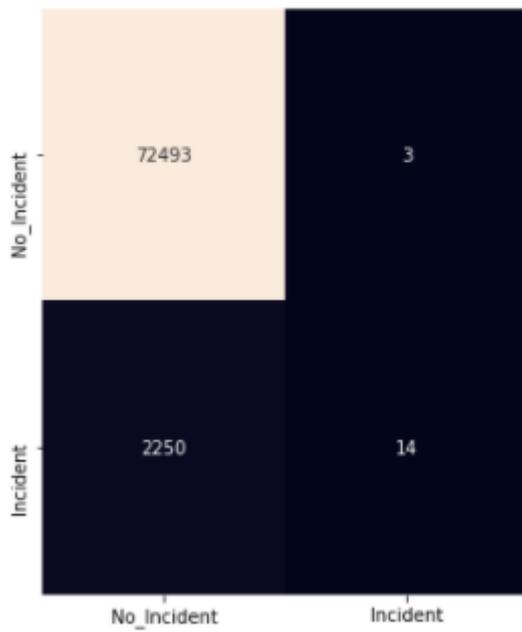
Algoritmo de seguimiento: DLIB



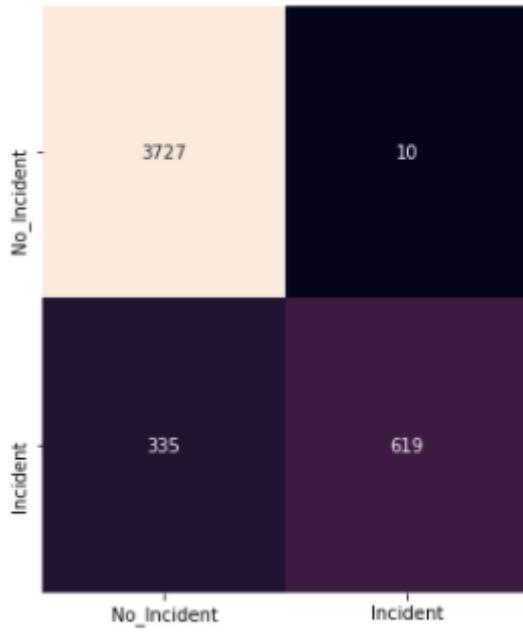
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

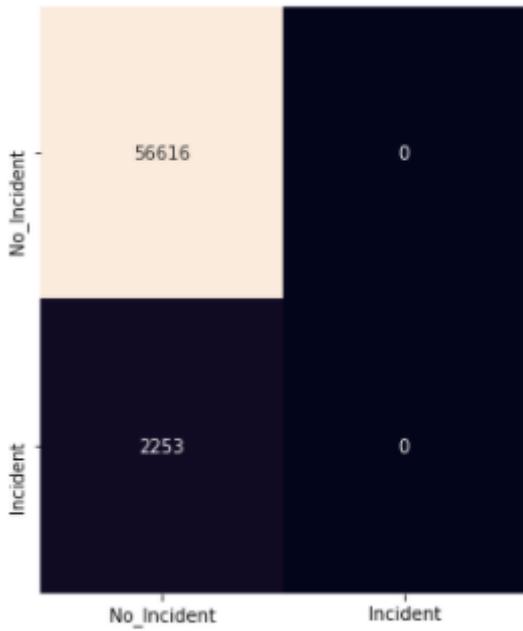


Algoritmo de seguimiento: centroide

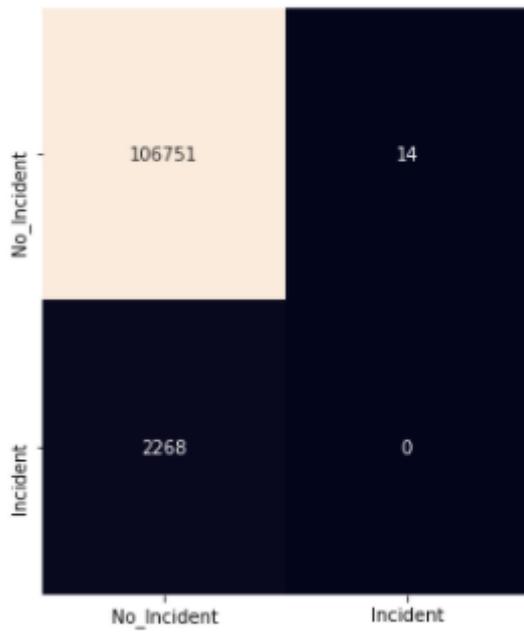


Modelo Regresión Logística

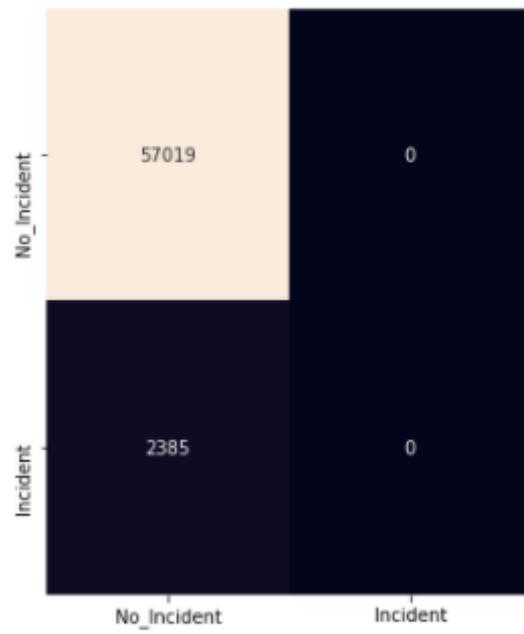
Algoritmo de seguimiento: KCF



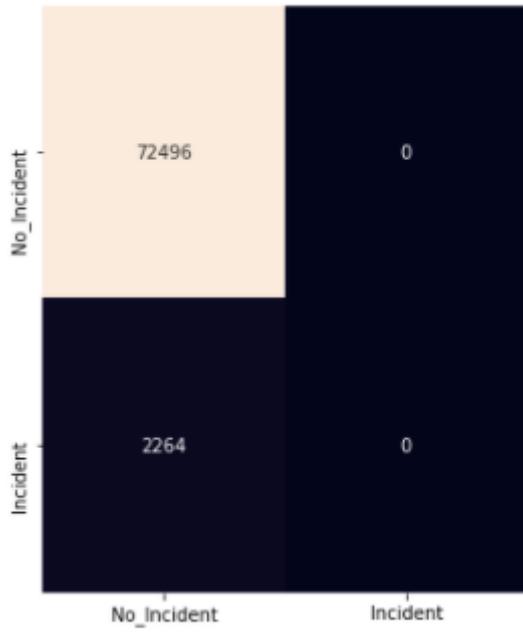
Algoritmo de seguimiento: DLIB



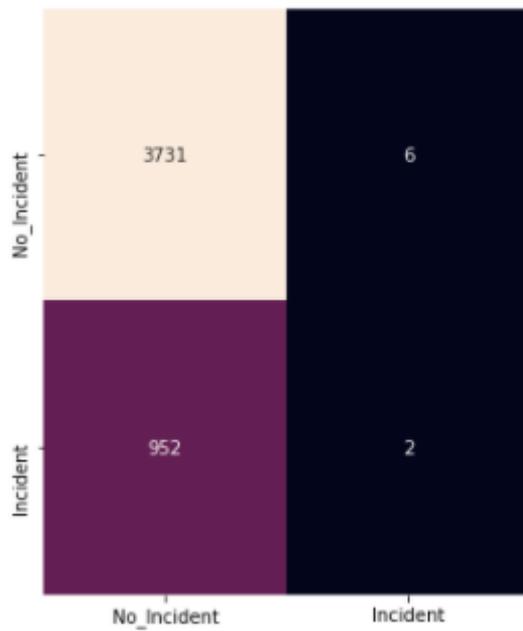
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

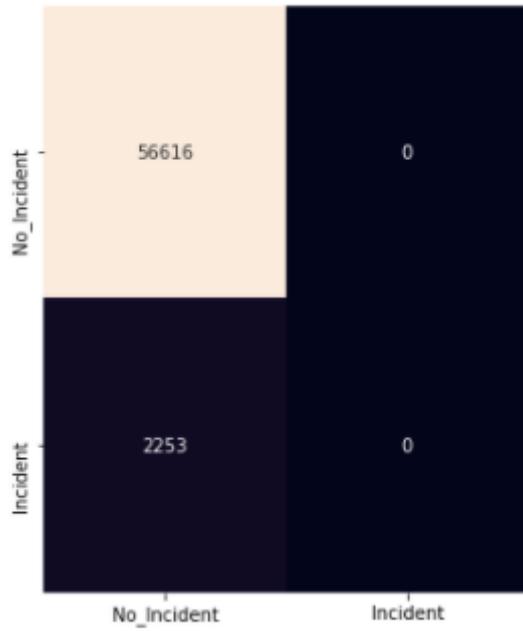


Algoritmo de seguimiento: centroide

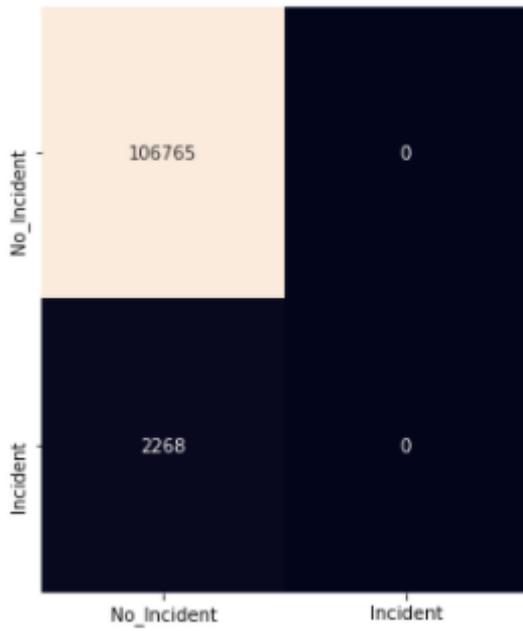


Modelo SVM (núcleo: POLY)

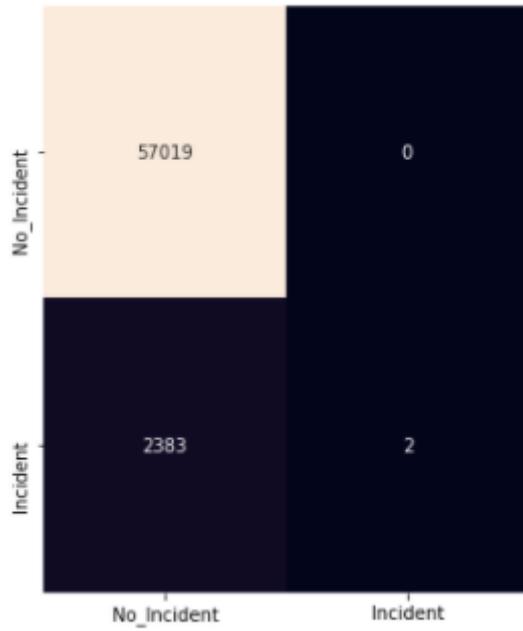
Algoritmo de seguimiento: KCF



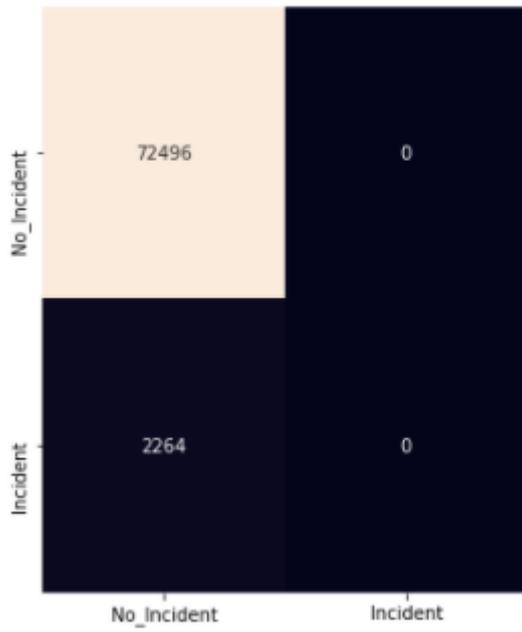
Algoritmo de seguimiento: DLIB



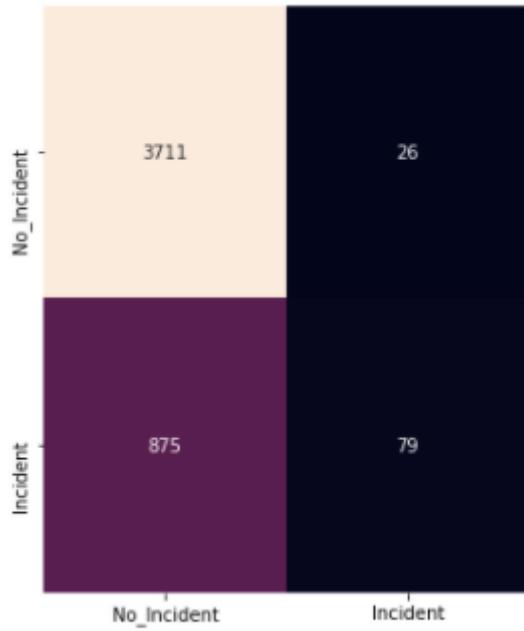
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

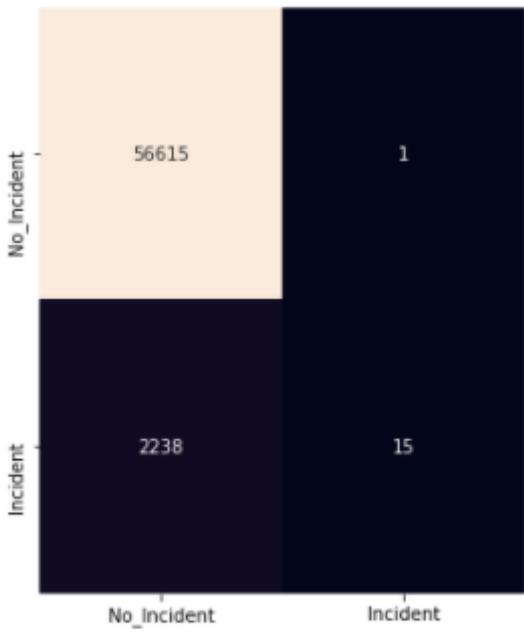


Algoritmo de seguimiento: centroide

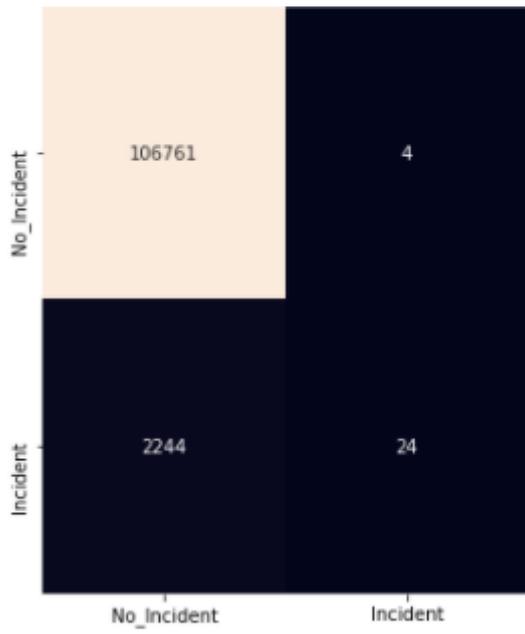


Modelo SVM (núcleo: RBF)

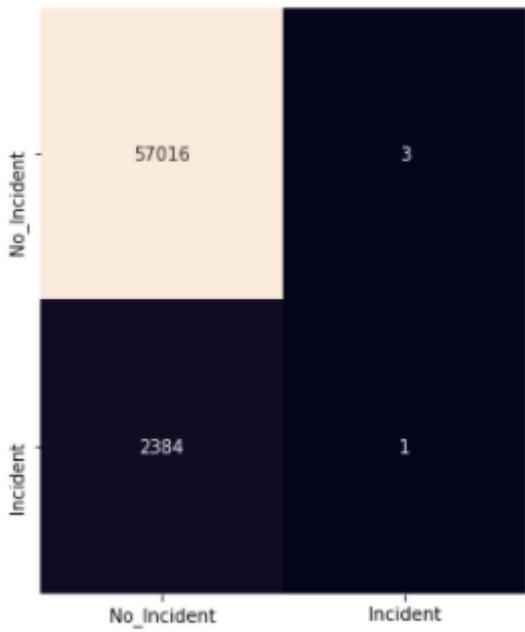
Algoritmo de seguimiento: KCF



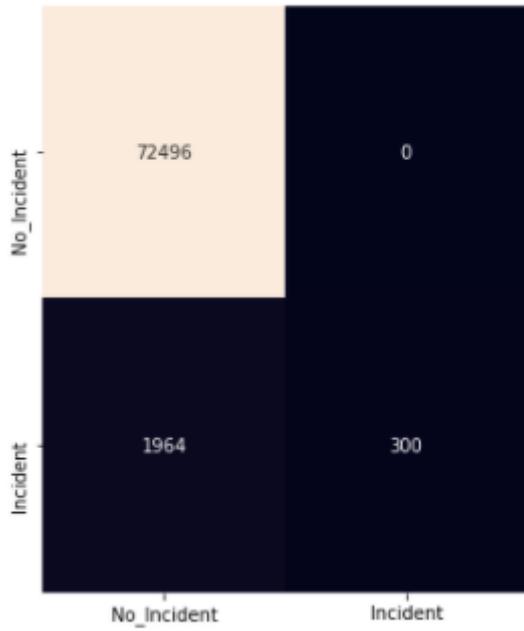
Algoritmo de seguimiento: DLIB



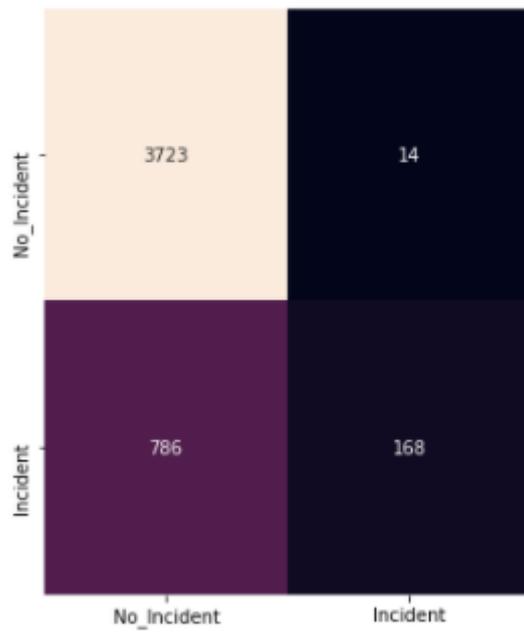
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

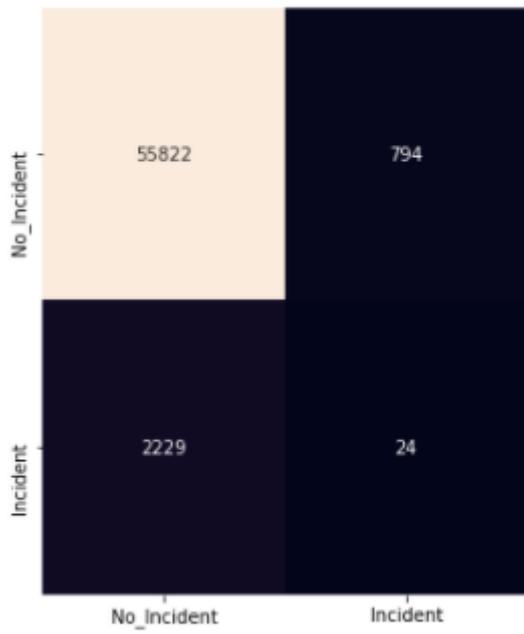


Algoritmo de seguimiento: centroide

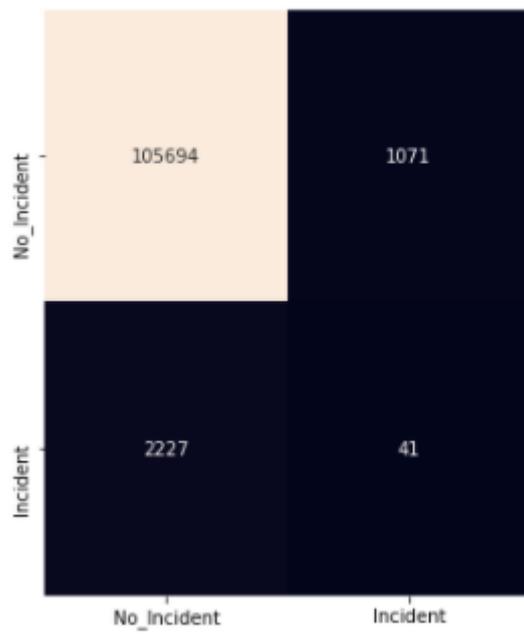


Modelo SVM (núcleo: SIGMOID)

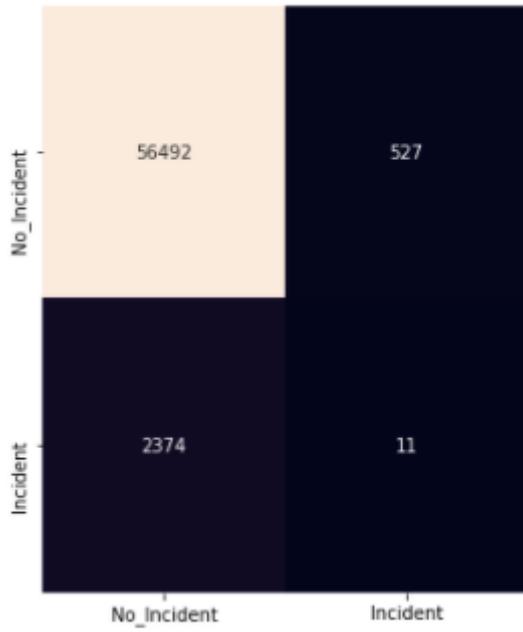
Algoritmo de seguimiento: KCF



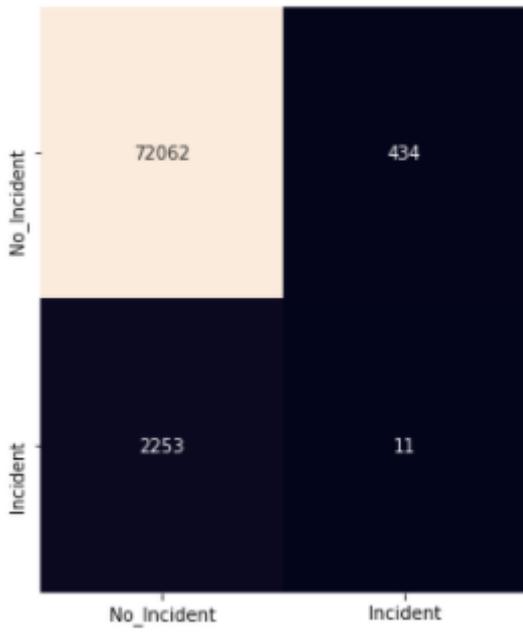
Algoritmo de seguimiento: DLIB



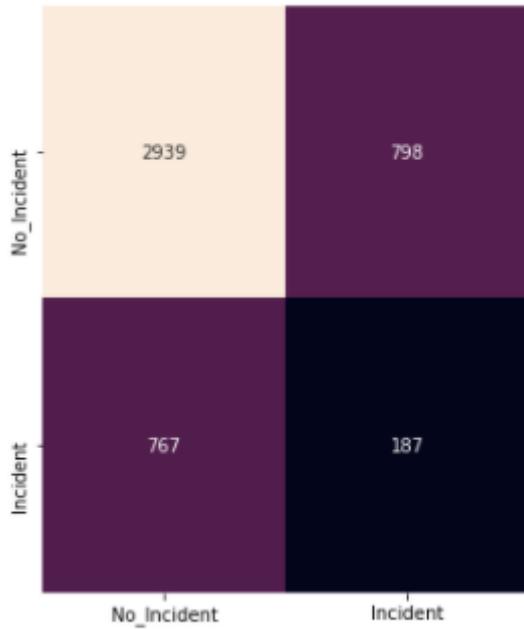
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow



Algoritmo de seguimiento: centroide



### Anexo 3 - Matrices de confusión con balanceo de datos

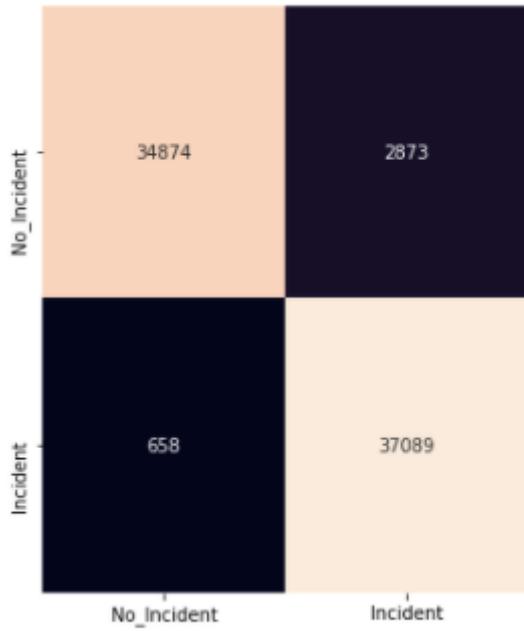
Algoritmo de detección de objetos: SSD Mobilenet

Modelo kNN

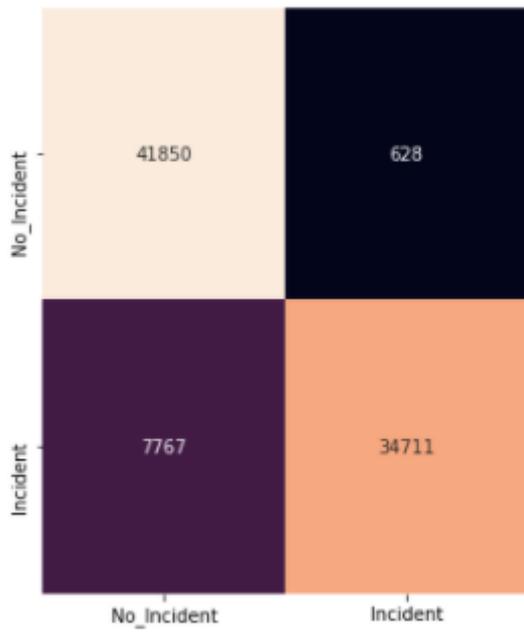
Algoritmo de seguimiento: KCF



Algoritmo de seguimiento: DLIB



Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

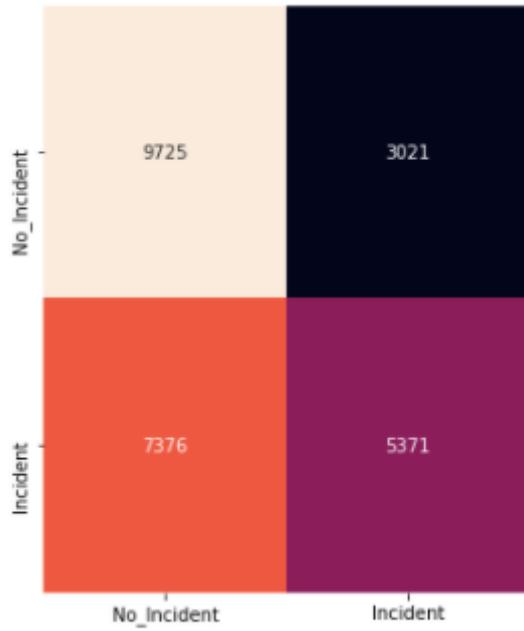
No_Incident	38233	3632
Incident	855	41009
	No_Incident	Incident

Algoritmo de seguimiento: centroide

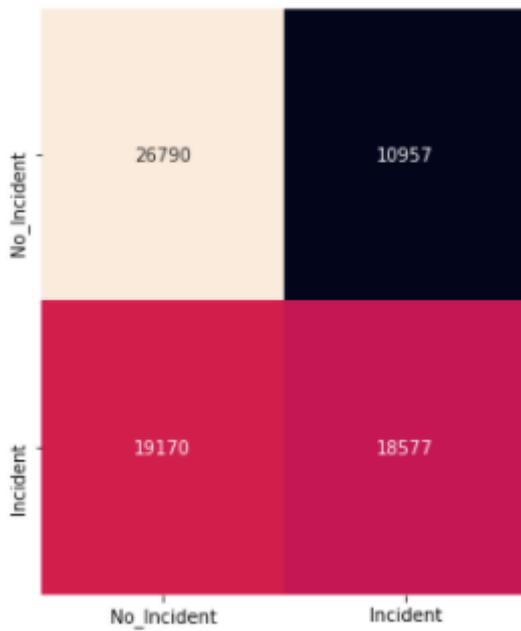
No_Incident	24824	615
Incident	2047	23391
	No_Incident	Incident

Modelo Regresión Logística

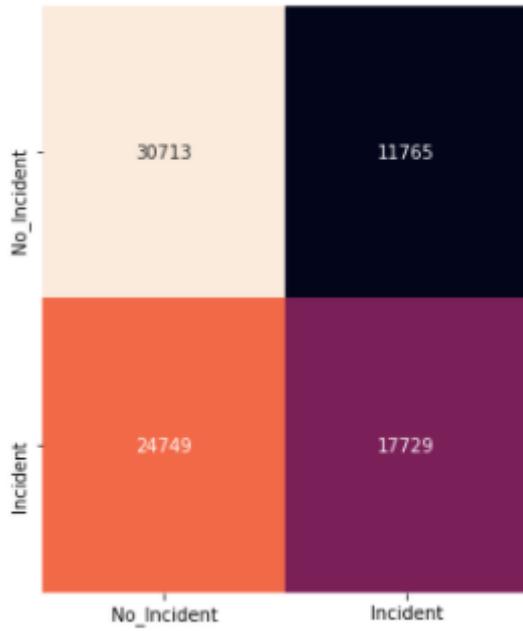
Algoritmo de seguimiento: KCF



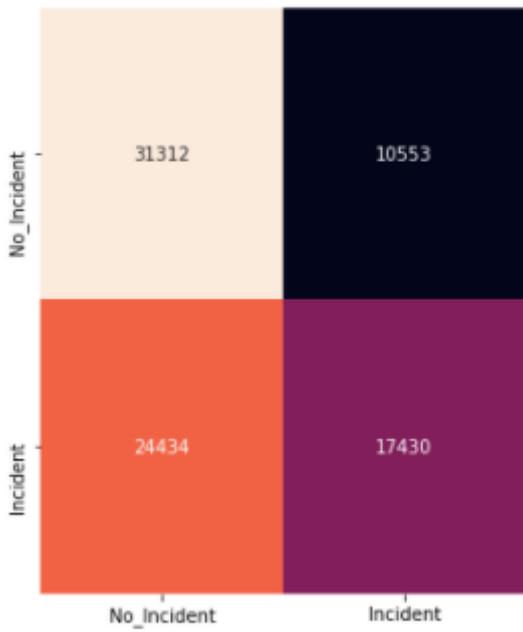
Algoritmo de seguimiento: DLIB



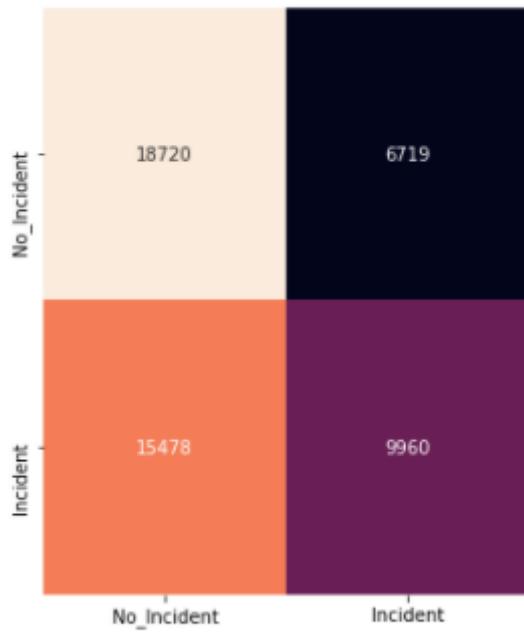
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

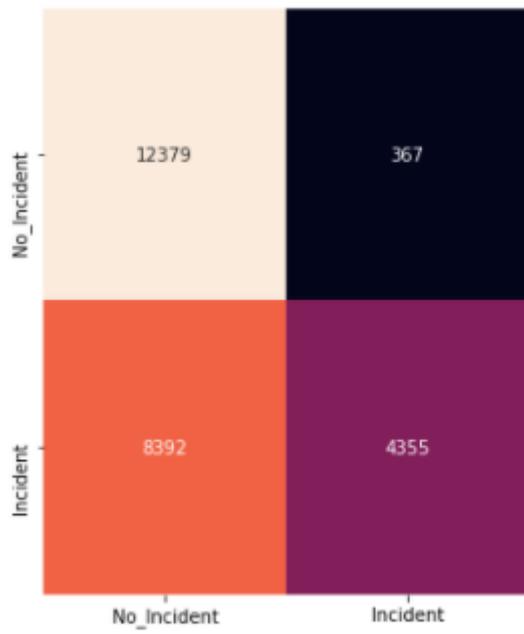


Algoritmo de seguimiento: centroide

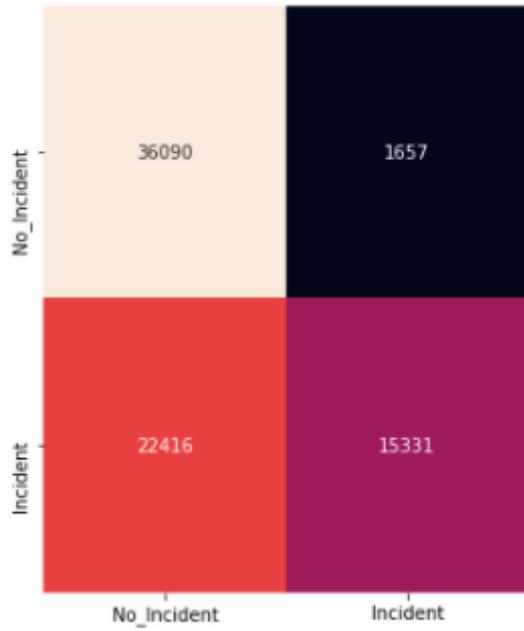


Modelo SVM (núcleo: POLY)

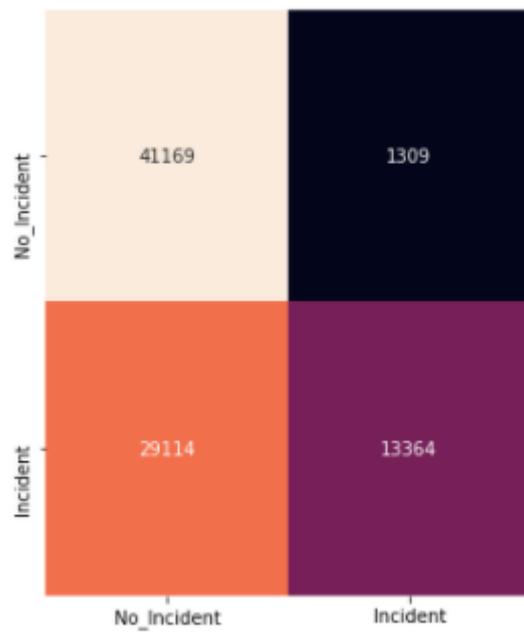
Algoritmo de seguimiento: KCF



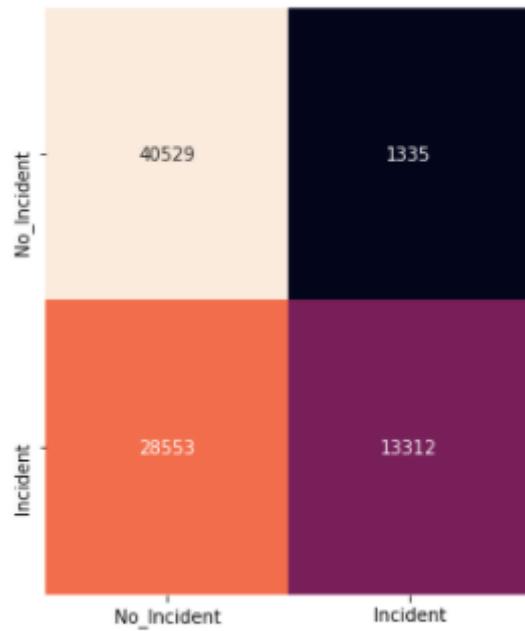
Algoritmo de seguimiento: DLIB



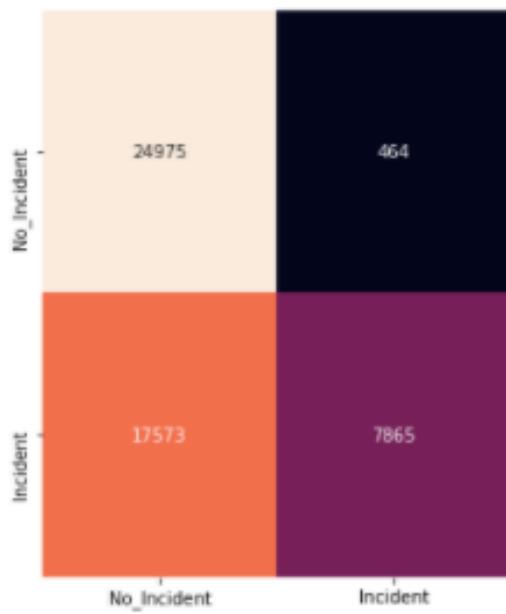
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

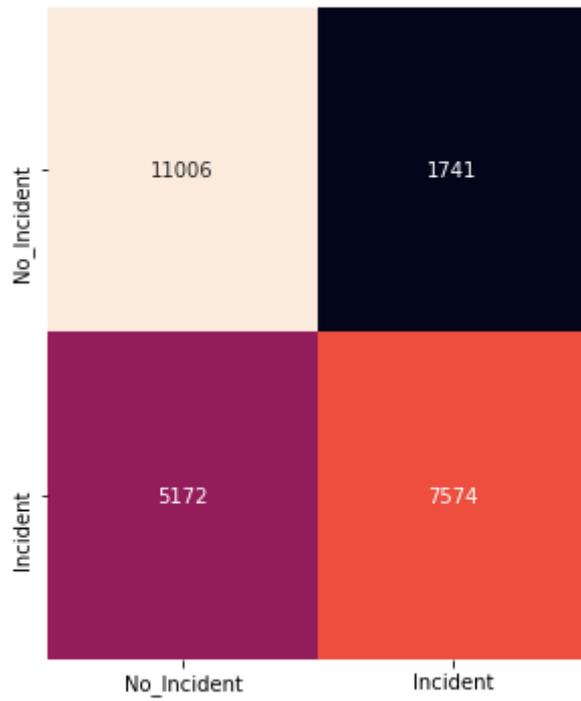


Algoritmo de seguimiento: centroide

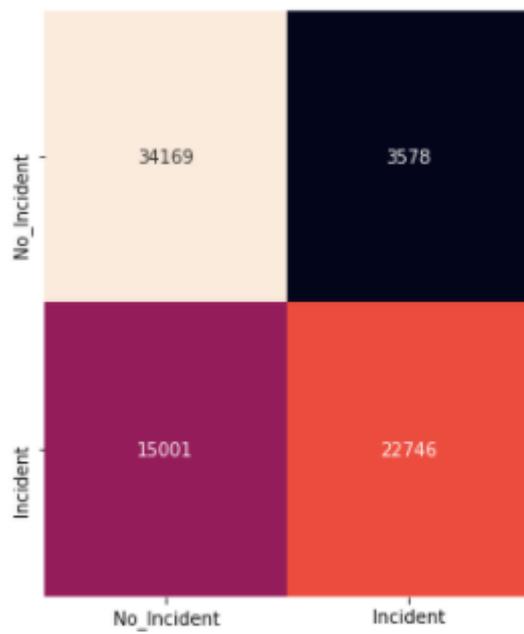


Modelo SVM (núcleo: RBF)

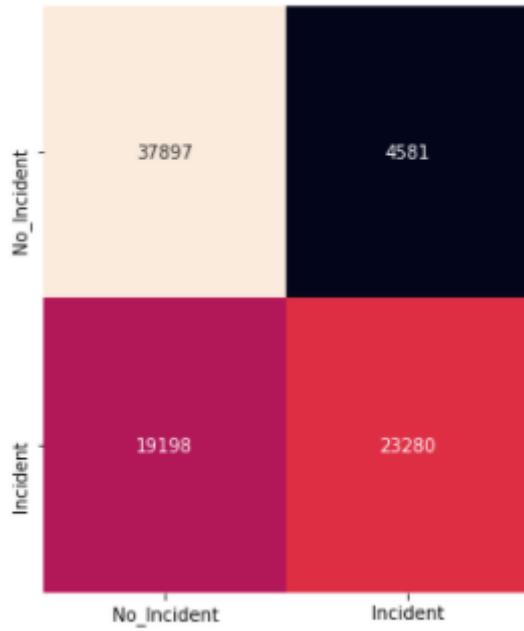
Algoritmo de seguimiento: KCF



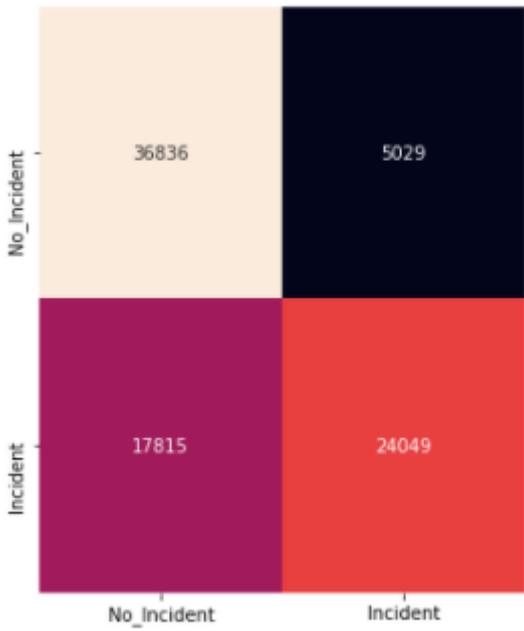
Algoritmo de seguimiento: DLIB



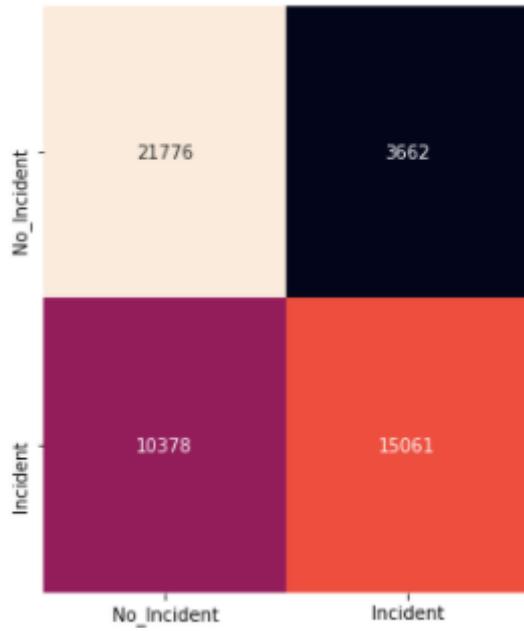
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

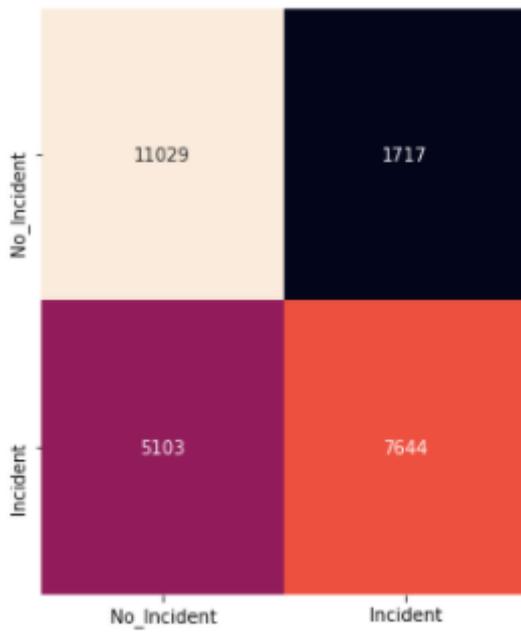


Algoritmo de seguimiento: centroide

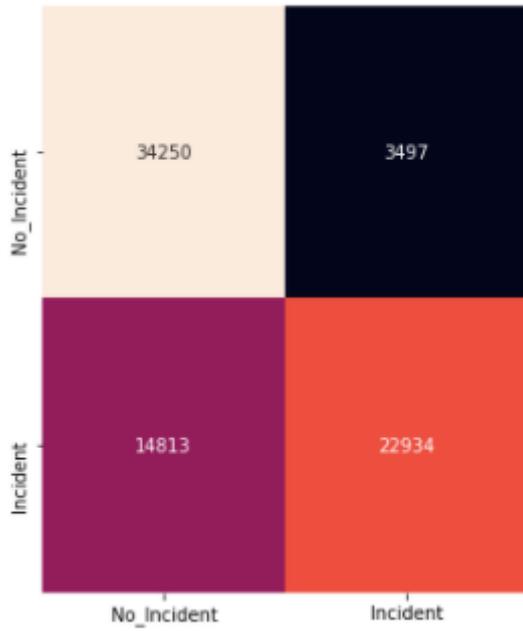


Modelo SVM (núcleo: SIGMOID)

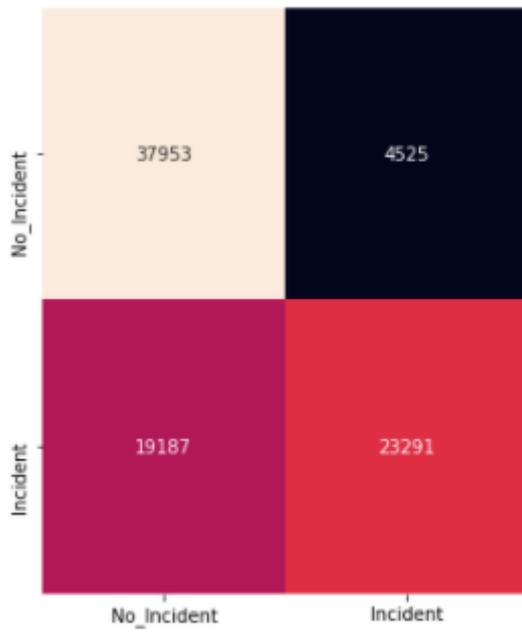
Algoritmo de seguimiento: KCF



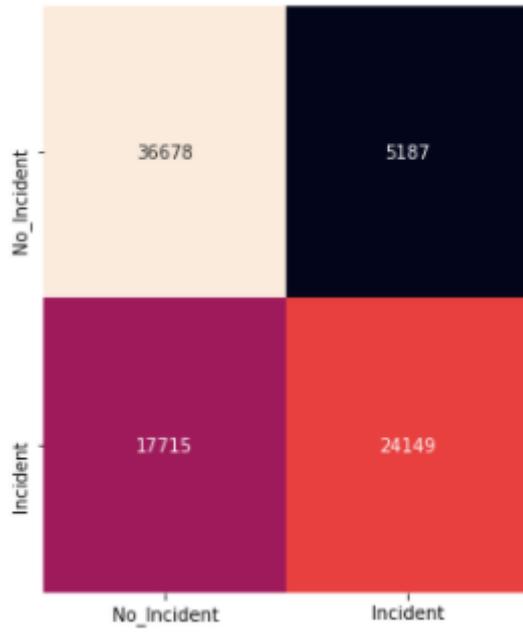
Algoritmo de seguimiento: DLIB



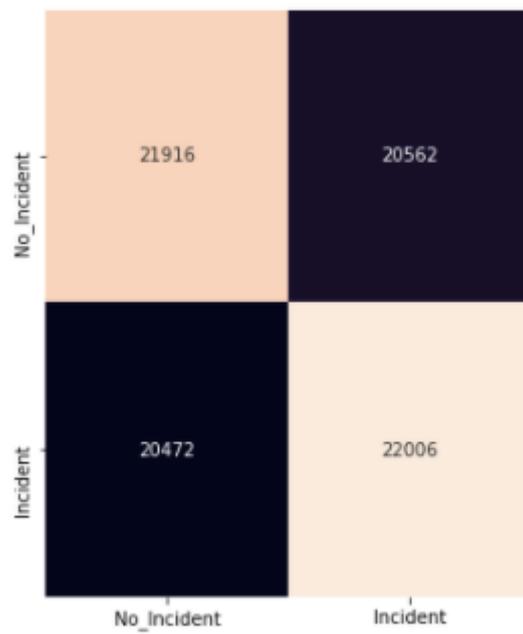
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow



Algoritmo de seguimiento: centroide



Algoritmo de detección de objetos: YOLO v3

Modelo kNN

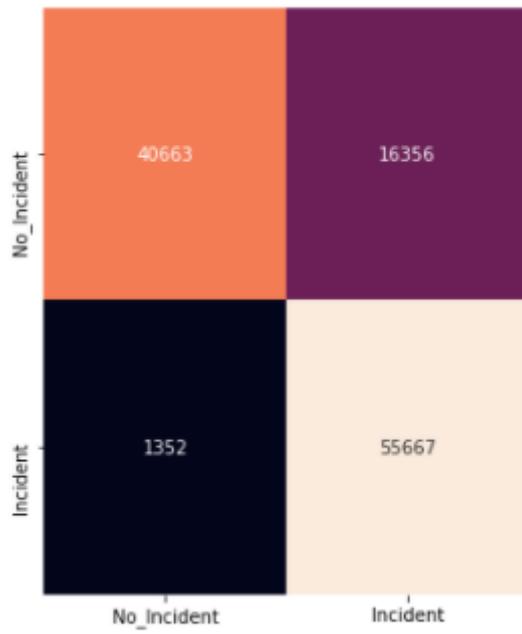
Algoritmo de seguimiento: KCF

No_Incident	52108	4508
Incident	7286	49330
	No_Incident	Incident

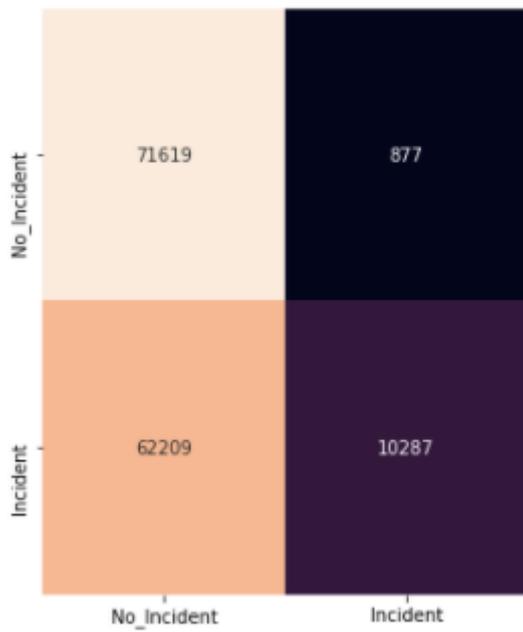
Algoritmo de seguimiento: DLIB

No_Incident	100077	6688
Incident	4246	102518
	No_Incident	Incident

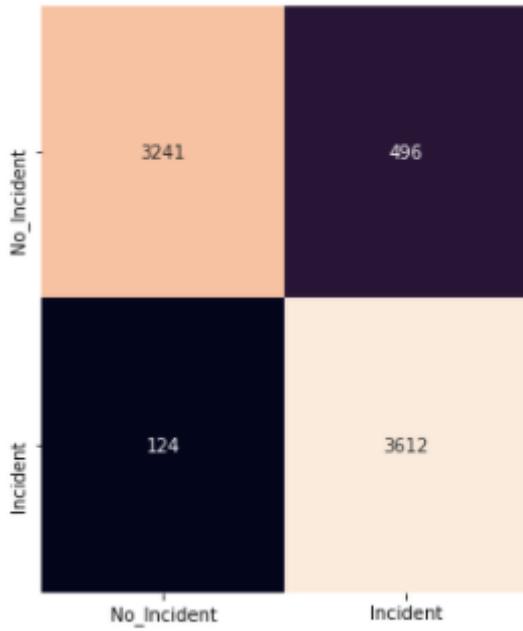
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

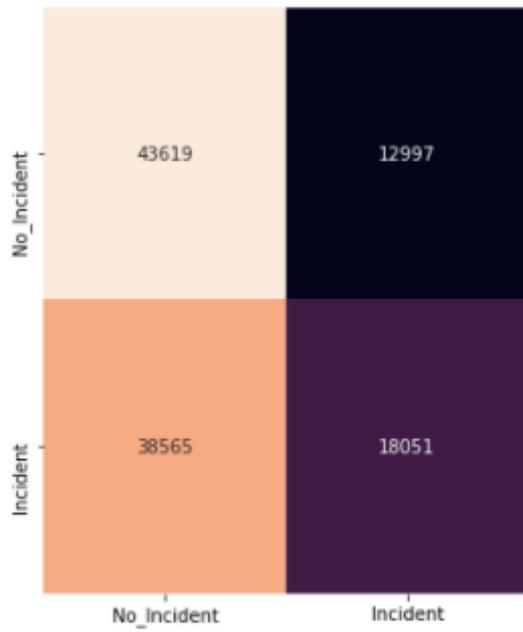


Algoritmo de seguimiento: centroide

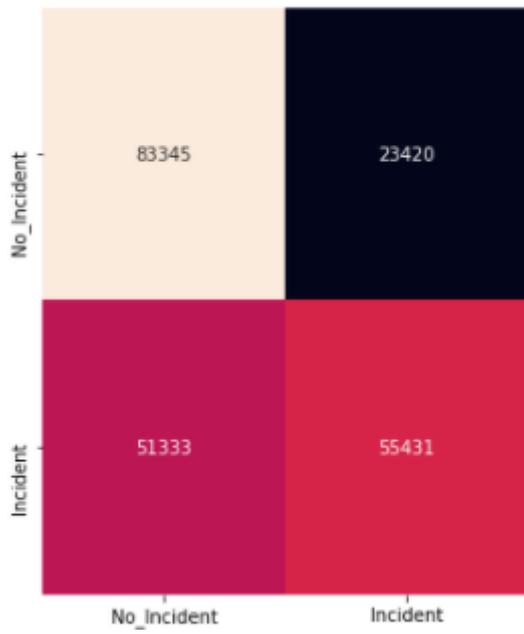


Modelo Regresión Logística

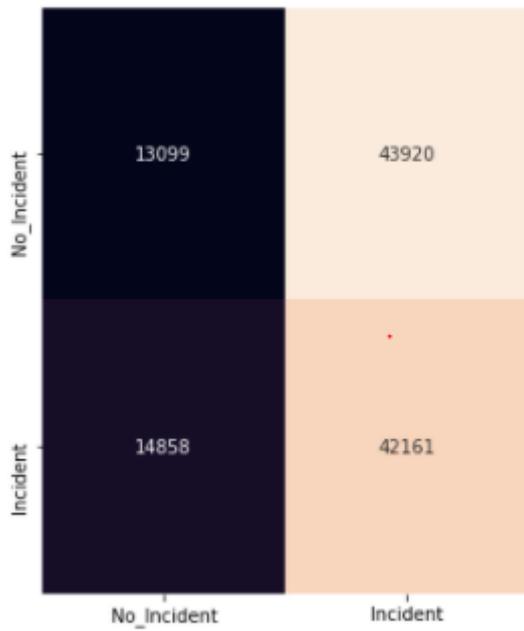
Algoritmo de seguimiento: KCF



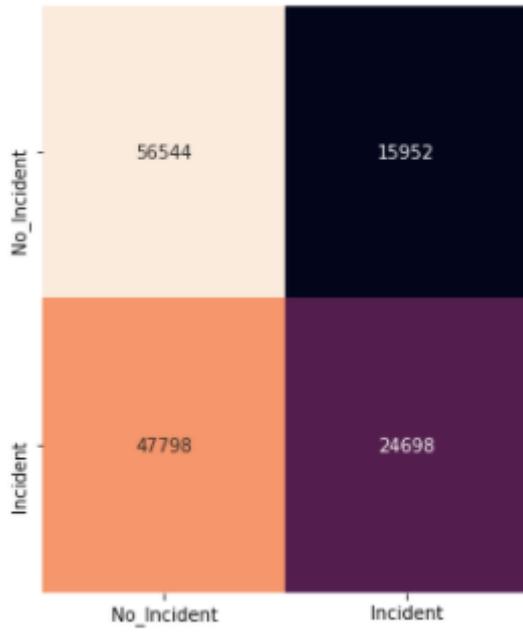
Algoritmo de seguimiento: DLIB



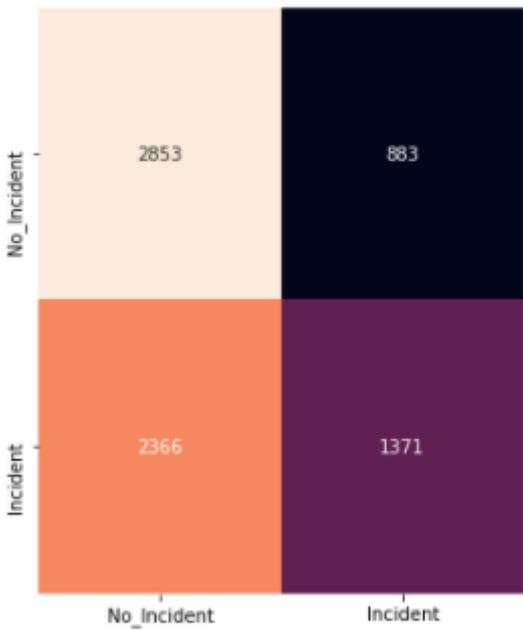
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

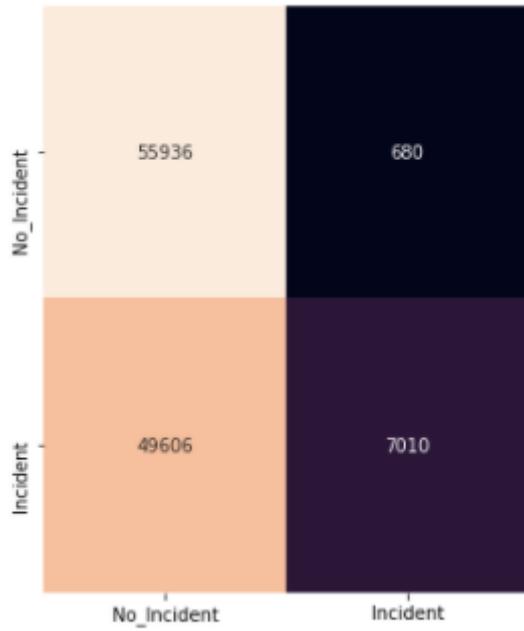


Algoritmo de seguimiento: centroide



Modelo SVM (núcleo: POLY)

Algoritmo de seguimiento: KCF



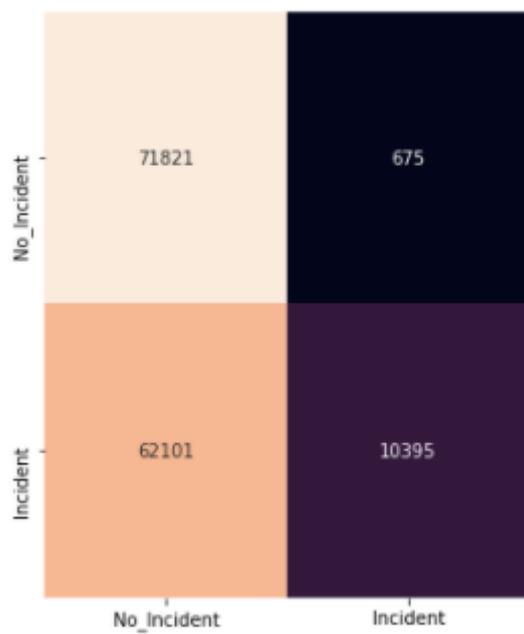
Algoritmo de seguimiento: DLIB



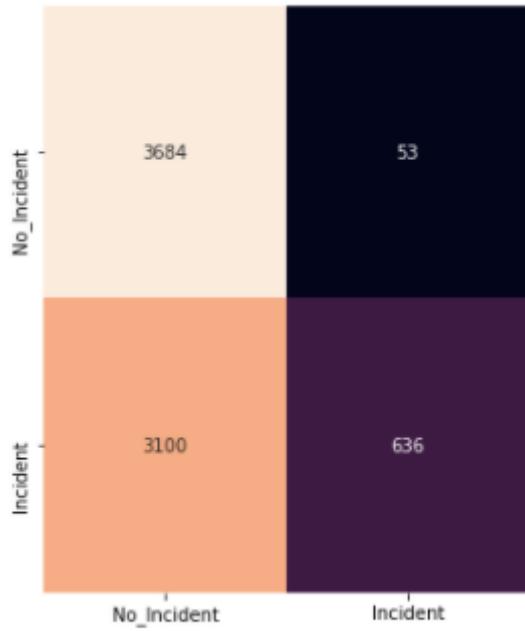
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

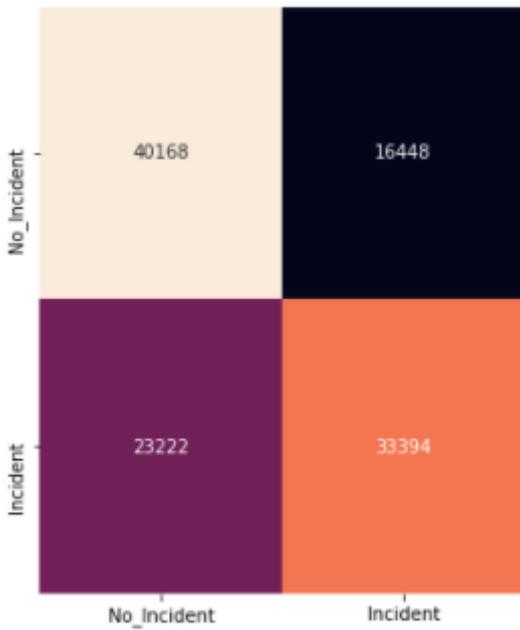


Algoritmo de seguimiento: centroide



Modelo SVM (núcleo: RBF)

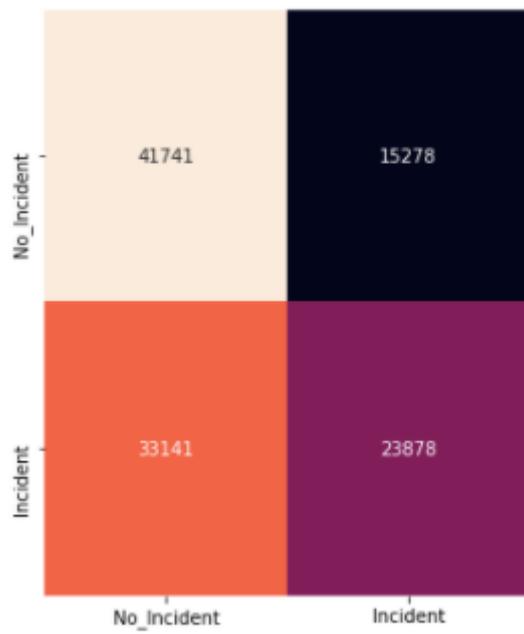
Algoritmo de seguimiento: KCF



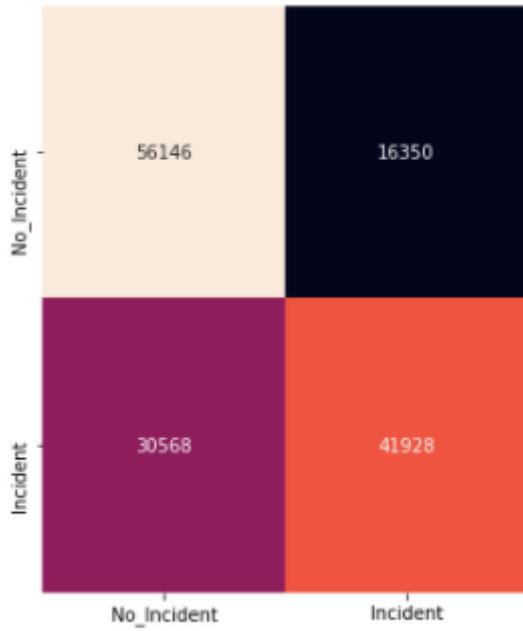
Algoritmo de seguimiento: DLIB



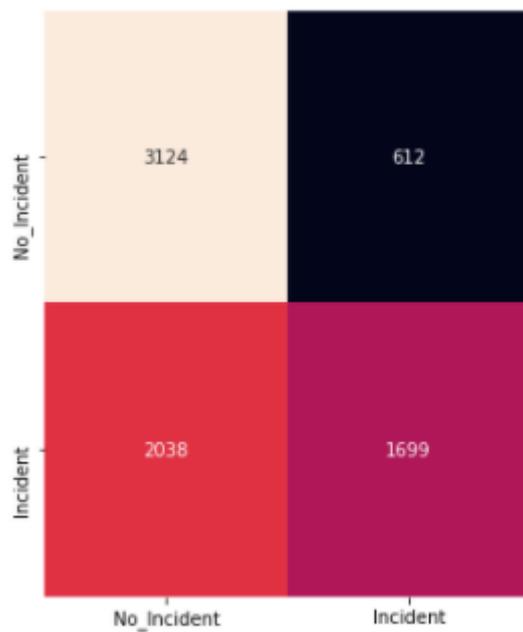
Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow

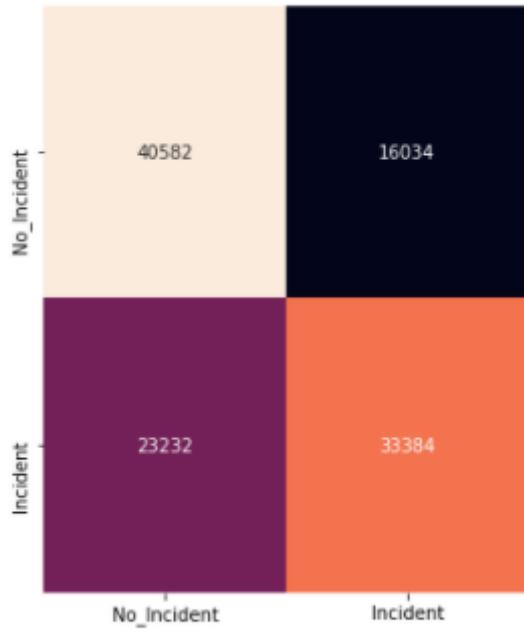


Algoritmo de seguimiento: centroide

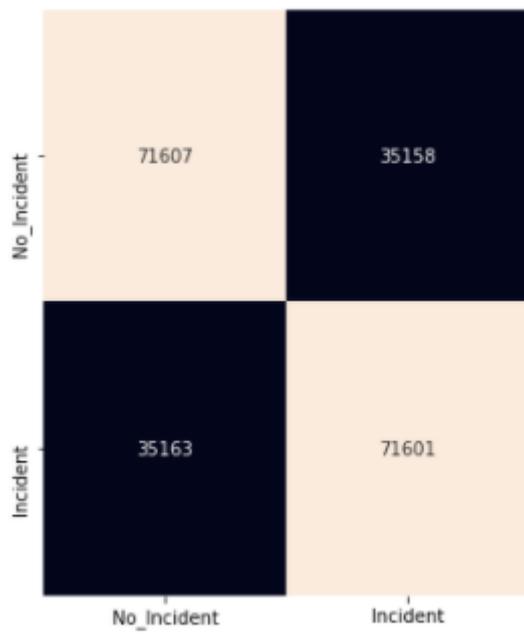


Modelo SVM (núcleo: SIGMOID)

Algoritmo de seguimiento: KCF



Algoritmo de seguimiento: DLIB



Algoritmo de seguimiento: MOSSE



Algoritmo de seguimiento: medianFlow



Algoritmo de seguimiento: centroide

