UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

FACULTAD DE INGENIERÍA

Enfoque de aprendizaje activo basado en umbrales para la detección de violencia física en imágenes de vídeo utilizando modelos de redes neuronales profundas preentrenadas

T E S I S

Que para obtener el Grado Académico de:

Doctor en Ciencias de la Ingeniería

Presenta:

Itzel María Abundez Barrera

Director de Tesis:

Dr. Otniel Portillo Rodríguez

Co-director de Tesis:

Dr. Roberto Alejo Eleuterio

Toluca, México, Enero, 2025

Índice general

Índice de figuras III

1. Introducción 5

1.1. Planteamiento del problema 8

1.2. Justificación 9

1.3. Hipótesis 10

1.4. Objetivo General 10

1.5. Objetivos Particulares 11

1.6. Alcances y Limitaciones 11

2. Fundamentos teóricos y Estado del arte 13

2.1. Fundamento teórico 13

Violencia física 13

Aprendizaje Automático 14

Aprendizaje Activo 14

Redes Neuronales 16

Métricas de evaluación del rendimiento 23

2.2. Estado del Arte 24

3. Metodología Propuesta 29

3.1. Propuesta aprendizaje activo 31

3.2. Propuesta Umbral 35
3.3. Implementación 36

4. Artículo de investigación **39**

5. Discusión **67**

6. Conclusiones **77**

Índice de figuras

2.1. Arquitectura de una CNN. 17
2.2. Proceso de convolución la primer matriz corresponde a la imagen, la segunda al filtro o *kernel* a utilizar y la tercera a la matriz resultante. 18
2.3. Redes convolucionales preentrenadas. 20
2.4. Arquitectura de DenseNet [33]. 21
2.5. Arquitectura de EfficientNet [32]. 22
2.6. Compound Scaling [32]. 22

3.1. Metodología propuesta 29
3.2. Construcción del clasificador con un modelo preentrenado 30

Abstract

Nowadays, human security has played a key role in society, which involves the rapid detection of violent actions. This task is usually carried out by human visual inspection of videos taken by video surveillance cameras, which is tiresome. Due to this, several deep learning approaches have been implemented to eliminate the human eye in this task, achieving favorable results. One of the main difficulties in detecting violence in video is the diversity of scenarios that may arise, which has prompted the use of various models trained on datasets to detect violence in a single dataset or in a few types of videos. In the present research, we present an approach for violence identification based on active learning divided into two stages. In the initial stage, pre-trained neural network models are developed on individual datasets, in order to detect those images that the classifier cannot detect, the uncertainty sampling technique was implemented through a threshold μ, quantifying the uncertainty that the classifier has to classify an image. Consequently, those ambiguous images are included to the training data set. In the second stage, the model is evaluated with videos from other environments, and the threshold μ is reapplied to detect ambiguous images, which are then analyzed by a human expert to determine the actual class and thereby remove the ambiguity in them. The ambiguous images already labeled by the human expert are then added to the original training set and the classifier is retrained; this process is repeated as long as ambiguous images exist. During this active learning pro-

cess, the classifier can detect violence in a variety of environments. The model is a hybrid neural network that uses transfer learning to modify a feature extraction technique based on convolutional neural network architectures, which has been successfully applied to video violence identification. Experimental findings indicate that it is possible to use the proposed model to detect violence in various scenarios.

Resumen

Hoy, la seguridad humana ha desempeñado un papel clave en la sociedad, que implica la detección rápida de acciones violentas. Esta tarea suele llevarse a cabo mediante la inspección visual humana de los vídeos tomados por las cámaras de video-vigilancia, lo que resulta tedioso. Debido a esto, se han implementado varios enfoques de aprendizaje profundo para eliminar el ojo humano en esta tarea, lográndose resultados favorables. Una de las principales dificultades en la detección de violencia en vídeo es la diversidad de escenarios que pueden presentarse, lo que ha impulsado el uso de diversos modelos entrenados en un único conjunto de datos o en unos pocos tipos de vídeos. En la presente investigación, se propone un enfoque para la identificación de la violencia basado en el aprendizaje activo dividido en dos etapas: en la inicial se utilizan modelos de redes neuronales preentrenados para el entrenamiento de un modelo. Para identificar las imágenes que el clasificador no puede detectar, se utiliza la técnica de muestreo de incertidumbre por medio de un umbral μ, que cuantifica la incertidumbre que tiene el clasificador para clasificar una imagen. En consecuencia, esas imágenes ambiguas se incluyen en el conjunto de datos de entrenamiento. En la segunda etapa, el modelo se evalúa con vídeos de otros entornos y el umbral μ se vuelve a aplicar para detectar las imágenes ambiguas, que luego son analizadas por un experto humano para determinar la clase real y eliminar así la ambigüedad en ellas. Las imágenes ambiguas ya etiquetadas por el experto humano se añaden al conjunto

de entrenamiento original y se vuelve a entrenar el clasificador. Este proceso se repite mientras existan imágenes ambiguas. Durante este proceso de aprendizaje activo, el clasificador puede detectar la violencia en diversos escenarios.

El modelo es una red neuronal híbrida que utiliza el aprendizaje por transferencia para modificar una técnica de extracción de características basada en arquitecturas de redes neuronales convolucionales, que se ha aplicado con éxito a la identificación de la violencia en vídeo. Los resultados experimentales indican que es posible utilizar el modelo propuesto para detectar la violencia en diversos escenarios.

Capítulo 1

Introducción

En los últimos años la detección automática de la violencia física ha sido un reto en el reconocimiento de la actividad humana porque requiere un análisis continuo del comportamiento humano [1]. El análisis a nivel de fotograma de un vídeo ha sido ampliamente estudiado para detectar la violencia física. Se trata de un proceso mediante el cual se analiza y examina cada fotograma individual para identificar signos de comportamiento violento o agresivo. Este análisis busca indicadores visuales específicos, como movimientos bruscos, gestos, puñetazos, contactos físicos entre personas u otros patrones que puedan sugerir una acción violenta. Al realizar esto, fotograma a fotograma, el sistema intenta detectar y categorizar los casos de violencia a medida que se producen, lo que permite una comprensión detallada de la actividad en el vídeo sin extraer características temporales [2].

La técnica de la bolsa de palabras (BoW), que funciona a nivel de fotograma de un vídeo, se ha utilizado para representar de forma compacta y robusta las características locales extraídas, lo que permite captar la distribución de los patrones visuales locales que caracterizan la violencia física sin necesidad de una disposición espacial exacta. Wang [3] utilizó BoW para identificar la violencia en un conjunto de datos que contenía 500 imágenes con escenas violentas y 1500 imágenes sin violencia. Al aplicar BoW a la transformación de características in-

variante de escala (SIFT por sus siglas en ingles), alcanzaron una precisión del 85.7 % ± 1,4 %. Al utilizar características histograma de gradiente (HOG por sus siglas en ingles), la precisión fue del 84.3 % ± 1,6 %. En cambio, con las características Patrón Binario Local (LBP por sus siglas en ingles) obtuvieron una precisión del 90.1 % ± 1.5 %, lo que demuestra que la combinación de características espaciales robustas con una representación BoW eficaz, permite detectar con precisión la violencia física en las imágenes.

Sin embargo, a pesar del éxito alcanzado por el análisis a nivel de fotograma de vídeo, la detección de violencia física en vídeos sigue siendo un reto importante. Recientemente, se ha abordado utilizando técnicas de aprendizaje profundo (deep learning, DL), principalmente con redes neuronales artificiales (RNA). Este enfoque tiene como objetivo incrementar la eficiencia de la clasificación de escenas de violencia física extraídas de diversas fuentes de vídeo, como sistemas de circuito cerrado de televisión (CCTV), teléfonos inteligentes y cámaras digitales utilizadas para tareas de vigilancia. En consecuencia, el desarrollo de estas metodologías ha ampliado las herramientas disponibles, reduciendo el esfuerzo humano necesario para identificar con precisión las situaciones de violencia física. Aunque la literatura actual y el estado del arte de la investigación cuentan con varios trabajos centrados en lograr altas tasas de precisión en la detección de violencia dentro de los vídeos, suelen oscilar entre el 90 % y el 99 % (Ref. [1]), cabe señalar que muchos de estos modelos están adaptados a conjuntos de datos específicos, que sólo funcionan particularmente en los escenario establecidos.

Las redes neuronales convolucionales (CNN) destacan como los modelos predominantes empleados para la clasificación de la violencia en los vídeos a nivel de fotograma, y algunas de las redes preentrenadas más notables son VGG16 [2], VGG19, DenseNet18, 26, 50, 101, 152, ResNet3D [4], ResNet18, ResNet34, ResNet50 [5], EfficientNet-B7, InceptionB3 y MobileNetV2 [6]. Aunque esos modelos

se propusieron hace algunos años, siguen siendo el centro de la investigación en tareas de clasificación de imágenes.

Los avances en la detección de la violencia pueden atribuirse al uso de redes neuronales preentrenadas y a la aparición de modelos híbridos. Un enfoque predominante consiste en emplear estos modelos preentrenados para la extracción de características espaciales, seguidos de una red secundaria responsable de la clasificación. Estas metodologías han demostrado porcentajes de precisión de hasta el 96 % [7]. Otra implementación digna de mención integra ConvLSTM para la extracción de características temporales y arquitecturas 3D ResNet50, 3D ResNet101 y 3D ResNet152 para la extracción de características espaciales junto con el conjunto de datos UCF-Crime, logrando una tasa de precisión del 96 % [8]. Los últimos avances incluyen un módulo de atención convolucional (CBAM por sus siglas en ingles) para discernir la dinámica entre los individuos que requieren detección [9]. Otros métodos explorados para la detección de violencia implican la extracción de características a través del flujo óptico, como se demuestra en el trabajo presentado por Vieira [10], Rendón- Segador et. al.[11] emplean el flujo óptico como entrada, precediendo a la utilización de un codificador espacio-temporal con DenseNet121 y una capa bidireccional de LSTM (Long Short-Term Memory Network) convolucional 2D (BiConvLSTM2D), obteniendo como resultado una precisión del 99 %. De forma similar, Wang [3] experimentó con cuatro conjuntos de datos para detección de violencia pública: Películas, Hockey, Multitudes y escenas del mundo real utilizando el conjunto de datos RWF-2000 [12]. Los autores adoptaron un enfoque comparable, extrayendo dos flujos ópticos, concatenándolos e introduciéndolos en el clasificador, logrando precisiones entre 0.86 y 1.0, para los conjuntos de datos Movie y RWF-2000, respectivamente.

Varios estudios se han centrado en abordar el problema de la detección de la violencia mediante un enfoque multiclase, utilizando el conjunto de datos UCF-

Crime para distinguir con mayor precisión entre varios tipos de violencia [13]. Del mismo modo, Vosta [14] examina cuatro variantes (Binary, dataset UCF-Crime - que contiene 14 clases, 13 eventos anormales agrupados como una clase anómala-, 4MajorCat, y NREF) del mismo conjunto de datos (UCF-Crime). Por su parte, Yousaf [15] experimentó con clips de dibujos animados para clasificar la violencia estableciendo tres clases: segura, violencia de fantasía y desnudez sexual.

La utilización de datos procedentes de diversos canales, como redes sociales en línea, conjuntos de datos públicos y cámaras de videovigilancia (con o sin sonido), ha surgido como una tendencia destinada a implementar modelos más robustos en diversos entornos [7], [16], [17]. Sin embargo, dado que cada día se generan nuevos vídeos en diferentes escenarios, todavía es necesario mejorar la confianza en modelos robustos pero estáticos entrenados con diversos conjuntos de datos.

El volumen sin precedentes de datos disponibles presenta una oportunidad significativa para el desarrollo de modelos de aprendizaje profundo; sin embargo, también plantea retos como la adaptabilidad de los modelos tradicionales, la usabilidad y la adaptabilidad [18]. Además, un reto crítico asociado a esta abundancia de datos es convertir los datos brutos en datos etiquetados de alta calidad, lo que resulta esencial que los modelo predictivos incrementen su precisión. La creación de un conjunto de entrenamiento implica un proceso potencialmente intensivo desde el punto de vista computacional, tanto en términos de tiempo como de recursos [19].

1.1. Planteamiento del problema

En el aprendizaje automático o *machine learning*, los modelos para la detección automática de violencia han sido desarrollados o adaptados a conjuntos de datos limitados, que solo funcionan en los escenarios previamente establecidos,

no obstante, cada día se generan nuevos vídeos desde diferentes escenarios, y su análisis es un proceso costoso en horas hombre-máquina, por lo que es necesario desarrollar modelos robustos y automatizados capaces de aprender de escenarios limitados, que puedan operar relativamente bien en escenarios desconocidos, para reducir el alto costo en tiempo y recursos humanos necesarios para su análisis.

1.2. Justificación

La seguridad pública es una prioridad para los gobiernos y ciudadanos en general, en México el Consejo Nacional de Humanidades Ciencia y Tecnología (CONAHCYT) en los Programas Nacionales Estratégicos del Pronaces establece los Proyectos Nacionales de Investigación e Incidencia (PRONAII) siendo uno de ellos la **seguridad humana** que impulsa un nuevo modelo de aproximación a las problemáticas nacionales relacionadas con las violencias estructurales y con las movilidades humanas. En este ámbito, se focaliza en articular esfuerzos intersectoriales e innovadores encaminados a la búsqueda de soluciones efectivas que involucran atender las causas estructurales. Los proyectos que conforman el Pronaces en seguridad humana parten de la noción de generar e impulsar una ciencia para la vida, una ciencia honesta y cercana a las comunidades, a las juventudes, a las niñas, niños y adolescentes, a todas las personas que habitan, transitan, retornan a este país. Actualmente se han hecho esfuerzos importantes por mejorar este sector, los cuales van desde mejoras en la capacitación de los encargados de esta área, hasta el incremento de infraestructura de vídeo vigilancia. En este sentido, el análisis de los vídeos obtenidos por las cámaras de vídeo vigilancia son extenuante, al ser realizada en su mayor parte por humanos y cada día se generan más y más vídeos lo cual vuelve prácticamente imposible su análisis, desaprovechado todo su potencial. A pesar de que existen numerosas propuestas

para automatizar esta tarea, aún sigue siendo un reto porque la mayoría de estos trabajos están limitados a escenarios específicos o entornos preestablecidos.

Por lo tanto, la necesidad de contar con propuestas encaminadas a la detección de violencia física (a partir de imágenes de vídeo) en imágenes, sin importar el escenario en el que se encuentre, es fundamental para mejorar los sistemas de vídeo vigilancia, contribuyendo a la mejora de la seguridad pública.

Derivado de lo anterior es posible que el modelo propuesto se pueda implementar en sistemas de vídeo vigilancia y con ello automatizar la detección en la violencia en la población.

1.3. Hipótesis

Al desarrollar un modelo de clasificación de imágenes con violencia e incorporar un mecanismo de aprendizaje activo basado en un umbral dinámico (μ) que seleccione de manera óptima aquellas imágenes ambiguas, cuya predicción tienen baja confianza según el umbral (μ), para ser etiquetadas y retroalimentar al modelo de clasificación de violencia física utilizando redes neuronales profundas preentrenadas, se incrementara la efectividad del clasificador, medida en términos de la exactitud y el área bajo la curva ROC (Receiver Operating Characteristic), al adaptarse progresivamente a nuevos escenarios, incrementando la capacidad del modelo para generalizar en la detección de violencia en diferentes entornos y condiciones.

1.4. Objetivo General

Desarrollar un modelo de detección de violencia física que incorpore un mecanismo de aprendizaje activo con muestreo por incertidumbre utilizando un umbral

y redes convolucionales preentrenadas para el proceso de entrenamiento, con el fin de mejorar la precisión y generalización del clasificador en entornos desconocidos.

1.5. Objetivos Particulares

- Seleccionar conjuntos de datos de acceso público con imágenes de violencia previamente etiquetados.
- Utilizar modelos de redes preentrenadas ampliamente conocidas y estudiadas en la comunidad de aprendizaje profundo.
- Diseñar e implementar un modelo que incluya aprendizaje activo con muestreo por incertidumbre utilizando un umbral y redes convolucionales preentrenadas para el proceso de entrenamiento.
- Evaluar la efectividad de clasificación del modelo propuesto.

1.6. Alcances y Limitaciones

En esta investigación se establece como alcance el desarrollar un modelo de aprendizaje activo aplicado a conjuntos de datos de acceso publico previamente etiquetados en violencia y no violencia, con escenas tomadas en diferentes escenarios, el modelo se desarrollo en un software libre aplicando aprendizaje activo mediante muestreo por incertidumbre por umbral para seleccionar aquellas imágenes que la red neuronal considere ambiguas y sean enviadas a un experto humano para etiquetar e incorporar a la Base de Datos (BD). Simular un proceso en tiempo real con un nuevo conjunto de datos no etiquetados e identificar violencia física.

Capítulo 2

Fundamentos teóricos y Estado del arte

En este capítulo se presentan los fundamento teóricos que constituyen los pilares esenciales de esta investigación así como el estado del arte de cómo ha sido abordada la problemática de la identificación de violencia en vídeos.

2.1. Fundamento teórico

A continuación se presentan los fundamento teóricos de los temas que forman parte de la presente investigación.

Violencia física

Se dice que existe violencia física cuando una persona viola el espacio corporal de otra sin su consentimiento, por ejemplo golpeándola, tirando de ella o empujándola. También puede causar lesiones físicas con objetos (armas u otros objetos) o con el cuerpo del agresor (puñetazos, tirones, patadas, empujones)[20]. La violencia física tiene consecuencias emocionales, de salud física y mental, motivo por el cual es importante su identificación. En el caso de violencia contra una mujer: La Ley General de Acceso de las Mujeres a una Vida Libre de Violencia [21], en su artículo 6, fracción II, define la violencia física como: «Un tipo de violencia que se refiere a cualquier acto que infringe daño no accidental, utilizando

la fuerza física o algún tipo de arma u objeto que causa o no lesiones internas, externas o ambas».

Aprendizaje Automático

Es considerado el diseño y análisis de las herramientas informáticas que utilizan la experiencia anterior para tomar decisiones futuras [22]; el objetivo primordial es generalizar, o inducir una regla desconocida a partir de ejemplos en los que esa regla se aplica. En este proceso de aprendizaje automático se fusionan conceptos y técnicas de diferente áreas del conocimiento, como las matemáticas, estadística y las ciencias computacionales; debido a que existen diversas formas de aprender la disciplina.

Aprendizaje Activo

En la actualidad la generación de grandes volúmenes de datos de los cuales se puede extraer conocimiento, permiten desarrollar modelos de aprendizaje automático en busca de su eficiencia, precisión; y con ellos adaptarse a diferentes ambientes, para lo cual se es necesario disponer de BD, previamente etiquetadas lo que conlleva a un proceso que realiza uno o varios seres humanos, dicho proceso es costoso en tiempo y recursos. El aprendizaje activo se establece como una solución convincente al buscar los puntos de datos más informativos para etiquetar a partir de un conjunto de muestras sin etiquetar con el fin de maximizar el rendimiento de la predicción. Si el proceso de etiquetado se realiza con ejemplos relevantes o ambiguos de forma selectiva y repetida, se construyen modelos con un rendimiento similar o superior reduciendo la cantidad de datos en comparación con enfoques tradicionales, disminuyendo el costo y la intervención humana, a diferencia de estrategias estadísticas no es necesario el etiquetado de el conjunto de datos completo. Lo anteriormente mencionado es la razón por lo

que se propone aplicar aprendizaje activo como una estrategia híbrida en la que intervienen expertos humanos y algoritmos para aprovechar las capacidades de ambas partes.

Al aplicar el aprendizaje activo es posible disminuir el costo temporal y espacial del proceso de etiquetado teniendo como resultado el incremento en el desempeño del clasificador al entrenar el modelo basado en un conjunto de entrenamiento robusto con datos etiquetados por un experto humano, en un proceso iterativo. Al respecto Cui *et al.* [23] señalan que no es posible comprender la decisión que un experto humano toma al momento de asignar etiquetas, por lo que no es posible aplicar una métrica que evalué el desempeño del experto humano. Sin embargo, se ha demostrado que el esfuerzo cognitivo y físico impacta en la tarea del experto humano.

Para etiquetar un dato es necesario que el clasificador, realice la selección secuencialmente de las instancias relevantes, consultando el conjunto de datos no etiquetados, los cuales se considera están representados por todas las clases [24]. Los enfoques establecidos para el desarrollo de consultas [25] son:

1. *Muestreo por incertidumbre*, basado en la selección de datos, en la probabilidad y un umbral, empleando una métrica de cuantización de la incertidumbre. Esta técnica es de las mayormente utilizadas.

2. *Consulta por comisión*, consiste en mantener un comité de n modelos, los cuales representan hipótesis contrapuestas, cada miembro del comité realiza su votación sobre el etiquetado de la consulta, la consulta que tiene el mayor numero de discrepancias corresponde a la instancia de mayor relevancia.

3. *Cambio del modelo previsto*, consiste en elegir la instancia que tenga el mayor cambio en el modelo actual si conociéramos su etiqueta.

4. *Reducción de la varianza y coeficiente de información de Fisher*, selecciona del conjunto de datos no etiquetados aquellos datos con mayor variabilidad.

5. *Reducción de la estimación de error*, reduce al máximo el error de los datos no etiquetados al seleccionar el dato con mayor probabilidad.

6. *Métodos de ponderación de la densidad*, establece que las instancias informativas no solo son las inciertas sino también las representativas de la distribución de entrada (aquellos datos localizados en áreas densas del espacio de entrada).

Muestreo por incertidumbre

Se considera un dato como relevante de acuerdo al enfoque de consulta, por lo que el clasificador selecciona las instancias en las que su predicción resulta tener una mayor incertidumbre *i.e.*, y que podrían estar localizadas en áreas ambiguas en las que su probabilidad de pertenencia es mayor o igual a dos clases [24]. Lo anterior se interpreta como el desconocimiento por parte del algoritmo de clasificación, este desconocimiento disminuye conforme se agregan más datos al conjunto de entrenamiento [26]. El aprendizaje activo ofrece la posibilidad de integrar el conocimiento humano, con el propósito de incrementar la eficacia del clasificador.

Redes Neuronales

Una RNA consta de numerosas capas interconectadas, cada una de ellas compuesta por varias neuronas. Hay dos tipos principales de RNA: las de avance (feed-forward, FNN por sus siglas en ingles) y las recurrentes. La diferencia radica en sus patrones de conectividad. En las redes neuronales recurrentes, las neuronas pueden estar conectadas a otras neuronas de la misma capa o a neuro-

nas de capas que no son ni anteriores ni posteriores. Por el contrario, en las redes neuronales FNN, las neuronas de una determinada capa sólo están conectadas a las neuronas de la capa posterior a través de pesos sinápticos (w) de la capa l, que incluye el peso de sesgo b_l. La salida de la neurona corresponde a una transformación espacial de s_i^l (Ecuación 2.1) por la función de activación (Ecuación (2.2)) [27], [28].

$$s_i^l = \sum_h w_{hi}^l \varphi_h^{(l-1)}(s_h^{(l-1)}), \tag{2.1}$$

$$\varphi^{(l)}(s_i^l) = \varphi^{(l)}(\mathbf{w}^l, s_i^l). \tag{2.2}$$

Redes Neuronales Convolucionales

El reconocer patrones visuales es un problema que en la actualidad se trabaja en inteligencia artificial utilizando aprendizaje profundo basado en redes neuronales convolucionales (CNN por sus siglas en ingles). La arquitectura de una CNN se muestra en la Figura 2.1. Estas CNN toman su nombre del proceso que realizan llamado convolución, el cual es la operación matemática lineal entre matrices ver (Ecuación. 2.3). La Figura 2.2 muestra un ejemplo del proceso de convolución.

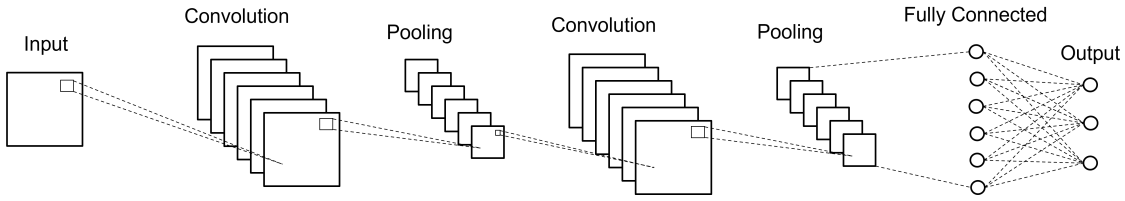


Figura 2.1: Arquitectura de una CNN.

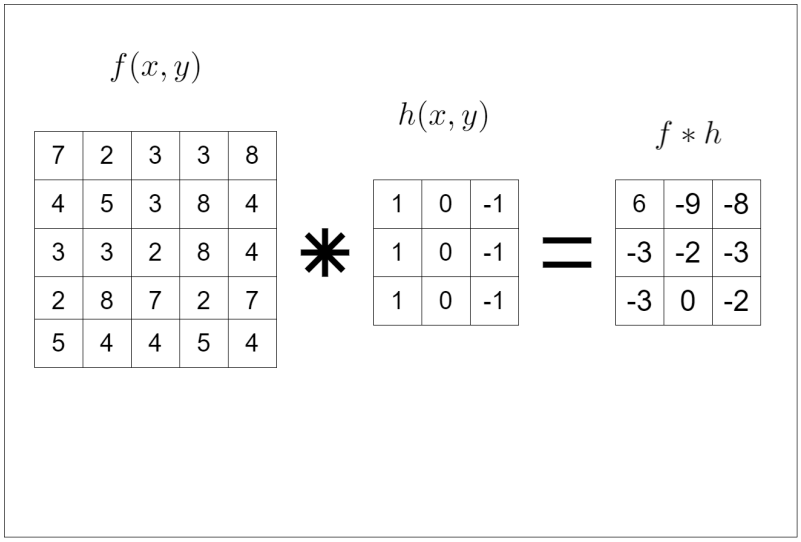


Figura 2.2: Proceso de convolución la primer matriz corresponde a la imagen, la segunda al filtro o *kernel* a utilizar y la tercera a la matriz resultante.

$$F(x,y) = \sum_i \sum_j h(i,j) f(x+i, y+j) \tag{2.3}$$

Las CNN, son una clase de perceptrón multicapa, con conexiones hacia todas las neuronas de la anterior capa. Estas redes tienen una estructura formada por capas de convolución (esta se encarga de extraer las características de los datos de entrada) y capas de agrupación (esta extrae invariancia traslacional de los patrones) a cada capa le corresponde una neurona diferente. Las dos capas tienen un campo excitatorio lo cual corresponde a un subconjunto de entrada en el caso de reconocimiento visual, este campo es una región rectangular de $n \times m$ píxeles. Todas las neuronas convolucionales de un mismo mapa de características presentan el mismo numero de conexiones y el mismo conjunto de pesos sinápticos. Al conjunto de pesos sinápticos se denomina *kernel*.

Las CNN pueden ser aplicadas a problemas donde no se cuenta con características que dependan del espacio, es decir que una característica puede encontrarse

en diferente lugar siendo independiente de la posición donde se encuentre el objeto a identificar. La arquitectura básica de una CNN está compuesta por:

- las capas convolucionales.
- funciones de activación.
- capas pooling.
- capas totalmente conectadas o full conected.

Aprendizaje por transferencia

El entrenar una red convolucional desde cero tiene un alto costo computacional, por lo que surge la trasferencia de aprendizaje, esta técnica trata de ajustarse a como los seres humanos utilizan el conocimiento en diferentes tareas. Aplicando el conocimiento previo en una nueva tarea. Existen diferentes modelos preentrenados, como: AlexNet [29], VGG16 [30], Resnet50 [31] , EfficientNet [32], DenseNet [33]. Cada modelo cuenta con su propia estructura; por lo que son utilizadas en diferentes problemas. La Figura 2.3 muestra la precisión de clasificación frente al tiempo de predicción de las diferentes redes preentrenadas.

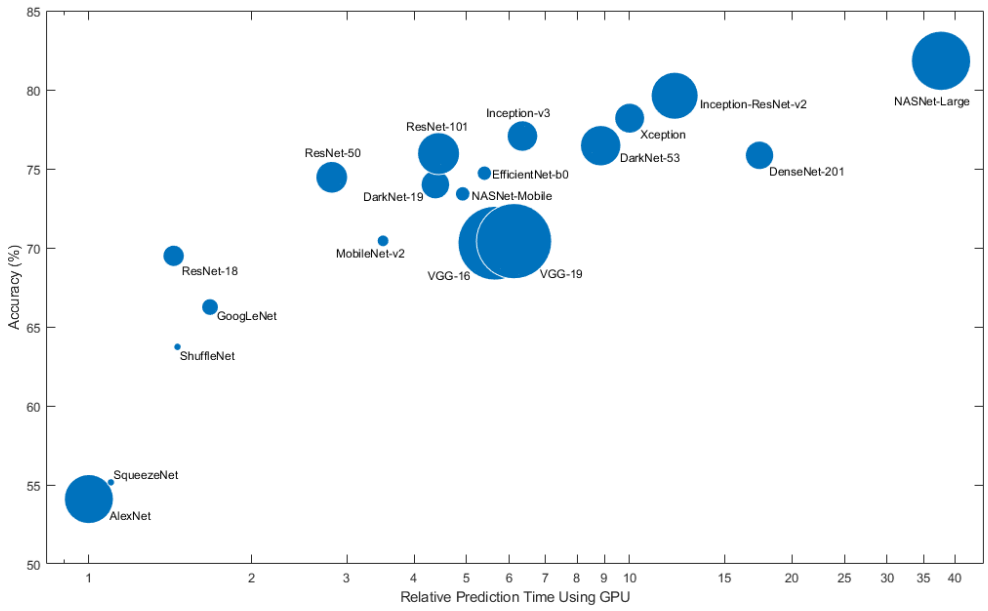


Figura 2.3: Redes convolucionales preentrenadas.

A continuación se describen tres modelos de redes preentrenadas utilizadas en esta investigación:

- *MobileNetV2*, la arquitectura de MobileNetV2 consta de dos tipos principales de bloques de construcción: bloques residuales con stride y bloques con un stride de 2 para reducir el tamaño. Cada bloque consta de tres capas: la primera capa es una capa convolucional de 1×1 seguida de una función de activación (ReLU6) que recorta la entrada al rango [0, 6], la segunda capa es una capa convolucional en profundidad, y la tercer capa es otra capa convolucional de 1×1 utilizada para combinar linealmente los canales de salida de la convolución en profundidad con una función de activación ReLU6 [34].

- *DenseNet*, es una red convolucional de 121 capas densamente conectadas

ver Figura 2.4. DenseNet utiliza las capas de reducción MaxPool2D y AveragePool3D con un tamaño de pool de (2,2) y (7,7). Las capas de reducción Maxpool3D y MaxPool2D se utilizan con un tamaño de pool(2,2,2) y (7,7,7). La arquitectura de esta red es una de sus ventajas por ser robusta, requiere un menor numero de filtros y parámetros para obtener resultados eficientes.

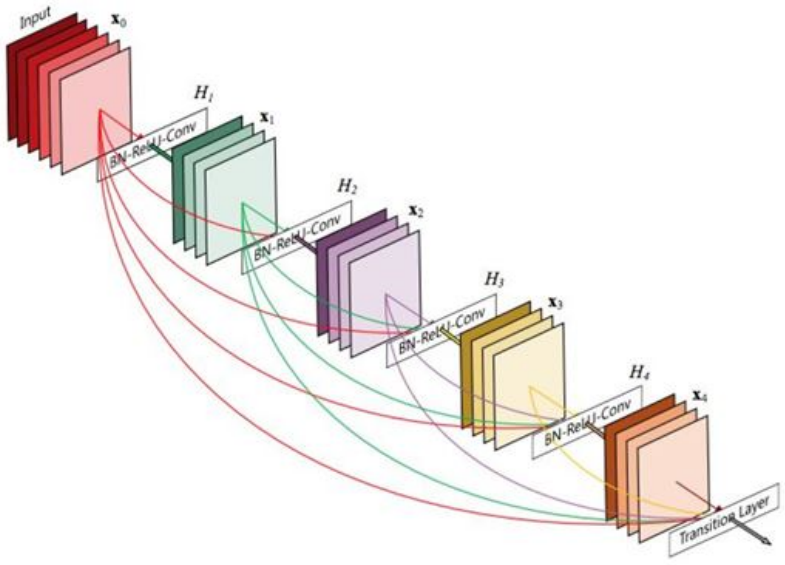


Figura 2.4: Arquitectura de DenseNet [33].

- *EfficientNet*, el modelo de red neuronal convolucional EfficientNet [32] es un modelo preentrenado con la base de datos de ImageNet [35] con 1000 categorías de objetos, en la Figura 2.5 se muestra su arquitectura. EfficientNet utiliza "Compound Scaling" ver Figura 2.6 tiene como función el incrementar la resolución de dimensión de las imágenes de entrada, es necesario disponer de una red más ancha (anchura) y larga (profundidad), para aprovechar los nuevos detalles que aportan las imágenes con resoluciones mayores. Lo que permite ser utilizado en sistemas poco potentes como es el

caso del uso de equipos de cómputo personales. La arquitectura de EfficientNet fue desarrollada para encontrar un método adecuado para escalar CNNs y lograr una mejor precisión (mayor rendimiento del modelo) y eficiencia proponiendo un método de escalado compuesto que utiliza un conjunto fijo de coeficientes para escalar de forma uniforme la anchura, la profundidad y la resolución. Siendo el primer modelo EfficientNet-B0, existiendo ocho arquitecturas de CNN EfficientNets B1-B7 basadas en el conjunto de datos ImageNet, permitiendo imágenes con mayor información y características complejas.

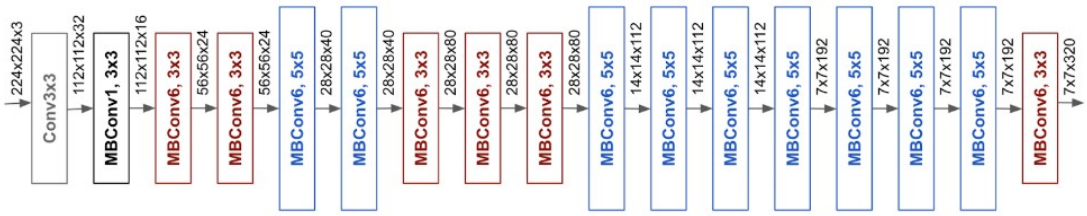


Figura 2.5: Arquitectura de EfficientNet [32].

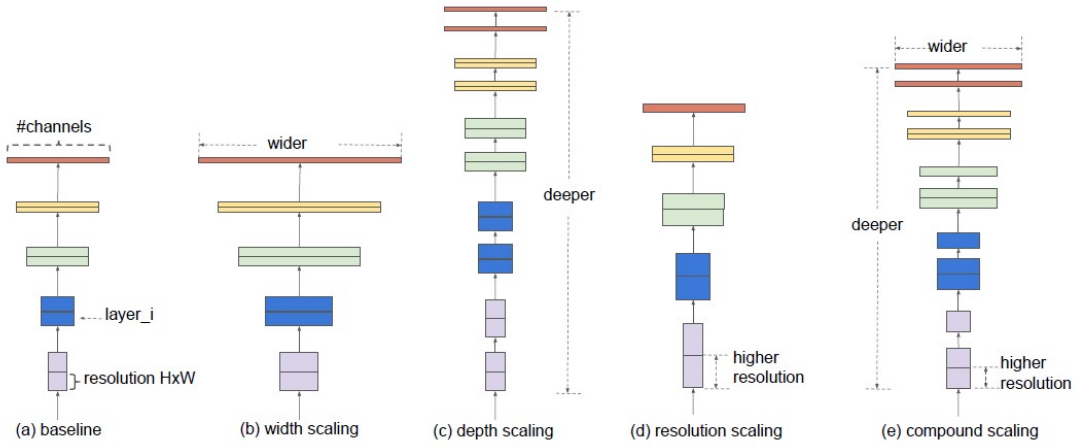


Figura 2.6: Compound Scaling [32].

Métricas de evaluación del rendimiento

Para obtener el rendimiento de cada modelo se calcula la eficiencia por clasificador, al ser un problema binario la clasificación es 0 o 1, con lo cual se puede construir la matriz de confusión ver tabla 2.1, esta matriz es la base para calcular las métricas utilizadas.

Tabla 2.1: Matriz de confusión

	1	**0**
1	**VP**	**FP**
0	**FN**	**VN**

- **VP** (Verdaderos Positivos) – corresponde a los valores que realmente son positivos y son clasificados como positivos.
- **VN** (Verdaderos Negativo) – valores que realmente son negativos y son clasificados como negativos.
- **FP** (Falsos Positivos) – valores que realmente son negativos y son clasificados como positivos.
- **FN** (Falsos Negativos) – valores clasificados como negativos y realmente son positivos.

Las métricas utilizadas para evaluar el rendimiento del modelo propuesto, se describen a continuación:

1. *Precisión*, calcula el porcentaje de predicciones correctas realizadas por el modelo (Ecuación 2.4).

$$Precision = \frac{1}{N}\sum_{c=0}^{N-1}\frac{(V_p^c + V_n^c)}{(V_p^c + V_n^c + F_p^c + F_n^c)} * 100\,\% \tag{2.4}$$

2. *Especificidad*, obtiene la relación entre las predicciones correctas y las predicciones generales, calculando el porcentaje de valores positivos clasificados correctamente (Ecuación 2.5).

$$Especificidad = \frac{1}{N}\sum_{c=0}^{N-1}\frac{V_p^c}{(V_p^c + F_p^c)} * 100\,\% \tag{2.5}$$

3. *Sensibilidad*, calcula la relación de predicciones correctas con respecto al numero total de casos positivos del conjunto de datos, es decir cuantos valores positivos son clasificados correctamente (Ecuación 2.6).

$$Sensibilidad = \frac{1}{N}\sum_{c=0}^{N-1}\frac{V_p^c}{(V_p^c + F_n^c)} * 100\,\% \tag{2.6}$$

4. *F1 Score*, calcula el promedio entre precisión y sensibilidad lo cual permite evaluar conjuntos de datos desbalanceados (Ecuación 2.7).

$$F1Score = 2 * (\frac{Precision * Sensibilidad}{Precisin + Sensibidad}) \tag{2.7}$$

2.2. Estado del Arte

El reto inherente a cualquier sistema de identificación de imágenes es intentar lograr una eficacia del 100 %, un objetivo a menudo difícil de alcanzar debido a complejidades intrínsecas. Esto es especialmente cierto en tareas como la identificación de la violencia física, donde los falsos negativos son lamentablemente frecuentes. Para hacer frente a este reto, nuestra investigación adopta un enfoque de aprendizaje activo, en el que se mezclan los esfuerzos de colaboración de expertos humanos y algoritmos de aprendizaje automático. El aprendizaje activo es un marco estratégico destinado a perfeccionar los clasificadores para que generalicen bien en instancias que no se encuentran en el dominio del problema [24]. El

aprendizaje activo encuentra su aplicación en varios dominios, incluyendo la clasificación de imágenes y la detección de objetos. Li y Guo [36] proponen un método que emplea una medida de incertidumbre, junto con la densidad de información y un marco de información adaptativo, para seleccionar instancias de conjuntos no etiquetados en función de su informatividad. Combinando medidas de incertidumbre con densidad de información adaptativa, este enfoque guía eficazmente la selección de instancias para el etiquetado, enriqueciendo así el conjunto de datos etiquetados.

Beluch et al. [37] investigan y comparan diferentes métodos de aprendizaje activo en el contexto de la clasificación de imágenes utilizando datos de alta dimensión y CNNs. El artículo evalúa métodos basados en conjuntos frente a enfoques geométricos y de abandono de Monte Carlo, encontrando que los conjuntos funcionan mejor y proporcionan incertidumbres predictivas más calibradas, que son cruciales para los algoritmos de aprendizaje activo. El estudio incluye experimentos con varios conjuntos de datos, como MNIST, CIFAR-10 e ImageNet, así como un gran conjunto de datos de retinopatía diabética con clases desbalanceadas. Los resultados demuestran que el aprendizaje activo basado en conjuntos es especialmente eficaz para gestionar el desbalance de clases durante la fase de adquisición.

En [38], Sener y Savarese introducen un enfoque de selección del conjunto central para el aprendizaje activo en CNNs, proporcionando un marco teórico para evaluar el rendimiento del subconjunto y demostrando mejoras significativas en tareas de clasificación de imágenes sobre los métodos existentes. Este método consiste en elegir un subconjunto de puntos de datos, el modelo entrenado en este subconjunto cuenta con un rendimiento competitivo en los datos restantes.

Pan et al. [39] producen resultados significativamente diferentes para una muestra dada basándose en el principio de que si dos redes neuronales profundas

(DNN) de la misma estructura se entrenan en el mismo conjunto de datos, esa muestra debe seleccionarse para un entrenamiento adicional. El enfoque de muestreo activo dual sobre aprendizaje activo incremental por lotes ha demostrado su utilidad. Simplifica la implementación y reduce los requisitos computacionales en comparación con otros métodos del estado de la técnica, lo que conduce a mejores resultados en el conjunto de datos CIFAR-10 con un tiempo computacional reducido en comparación con el método de conjunto central.

Carbonneau et al. [40] introducen nuevos métodos para la agregación a nivel de bolsa en instancias múltiples de aprendizaje activo (MIAL por sus siglas en ingles), una técnica utilizada en problemas de clasificación de instancias. El estudio se centra en reducir los costes de etiquetado mediante la identificación de las instancias más informativas y consulta al experto humado por las etiquetas. El método propuesto supera a los métodos existentes en varios dominios de aplicación, incluyendo la información agregada y el muestreo agregado, basado en clusters. Esta investigación es significativa, por aborda las limitaciones de los métodos de aprendizaje activo de instancia única en problemas de aprendizaje de múltiples instancias.

Los esfuerzos recientes, ejemplificados por Chen et al. [41], hacen hincapié en el uso de grandes conjuntos de datos sin etiquetado completo. Chen et al. proponen un método de detección de objetos que combina técnicas de aprendizaje activo y supervisado. Inicialmente, los datos etiquetados se utilizan para entrenar un modelo de detección de forma semisupervisada. Posteriormente, se evalúa la estabilidad de los datos no etiquetados y se etiquetan manualmente los casos de baja estabilidad, mientras que los de alta incertidumbre se pseudoetiquetan utilizando predicciones del detector. La integración del etiquetado manual y el pseudoetiquetado contribuye a refinar el modelo de detección, lo que arroja resultados prometedores con una precisión media (mAP) del 79.2 %. Otro enfoque

notable, presentado por Li et al. [42], utiliza el aprendizaje activo basado en la competencia para clasificar imágenes no etiquetadas por su complejidad. Utilizando una función de competencia junto con probabilidades de predicción, este método identifica eficazmente imágenes con distintos grados de dificultad, aumentando iterativamente el conjunto de datos. Li et al., que entrenaron el modelo con descenso estocástico de gradiente (SGD) y una tasa de aprendizaje de 0.1, evaluaron tres métodos de aprendizaje activo, la selección basada en la similitud, la selección de similitud basada en la probabilidad y el aprendizaje activo basado en la competencia, y alcanzaron una precisión final del 92 %.

Además, Mohammadi et al. [43] introducen un marco que utiliza transformadores de visión preentrenados y entrenados en el conjunto de datos RWF como expertos junto con un sistema de enrutamiento y un clasificador basado en el aprendizaje por refuerzo. Este enfoque activa dinámicamente a los expertos en relación con cada clip de vídeo, determinando su categoría (pelea o no pelea) a través de varias selecciones y categorizaciones. El módulo de enrutamiento ejecuta el proceso de clasificación, decidiendo si selecciona inmediatamente la categoría de un clip o si recaba más información activando transformadores de visión, concluyendo así el proceso de clasificación con la selección de la categoría adecuada.

En [42] utilizan el aprendizaje activo basado en competencias, para la clasificación de imágenes sin etiquetar en función de su complejidad utilizando la probabilidad de predicción y la función de competencias para establecer el numero de imágenes con dificultad, estas se añaden al conjunto de datos en cada iteración, dicho modelo es entrenado con SGD y una tasa de aprendizaje del 0.1. Evalúan tres métodos basados en aprendizaje activo: selección basada en similitud, similitud basada en la probabilidad de predicción y aprendizaje activo basado en competencia, alcanzando un 0.92 % de eficiencia.

En el estado del arte los diferentes métodos y enfoques, han obtenido re-

sultados favorables entre 84-99 % de efectividad, sin embargo, los modelos son desarrollados para conjuntos de datos limitados que solo funcionan en los escenarios previamente establecidos, no obstante, cada día se generan nuevos vídeos desde diferentes escenarios, y su análisis es un proceso costoso en horas hombre-máquina, por lo que es necesario desarrollar modelos robustos y automatizados capaces de aprender de escenarios limitados que puedan operar relativamente bien en escenarios desconocidos, para reducir el alto costo en tiempo y recursos humanos necesarios para su análisis.

Capítulo 3

Metodología Propuesta

En esta investigación, se propone un enfoque de aprendizaje activo basado en un parámetro de umbral (μ) utilizado en incrementar la eficiencia de los clasificadores, construir modelos robustos capaces de funcionar eficazmente en diversos entornos para detectar la violencia física a partir de vídeos mediante el análisis de cada fotograma, y crear conjuntos de datos de alta calidad con la intervención de un experto humano. La metodología propuesta se muestra en la Figura 3.1.

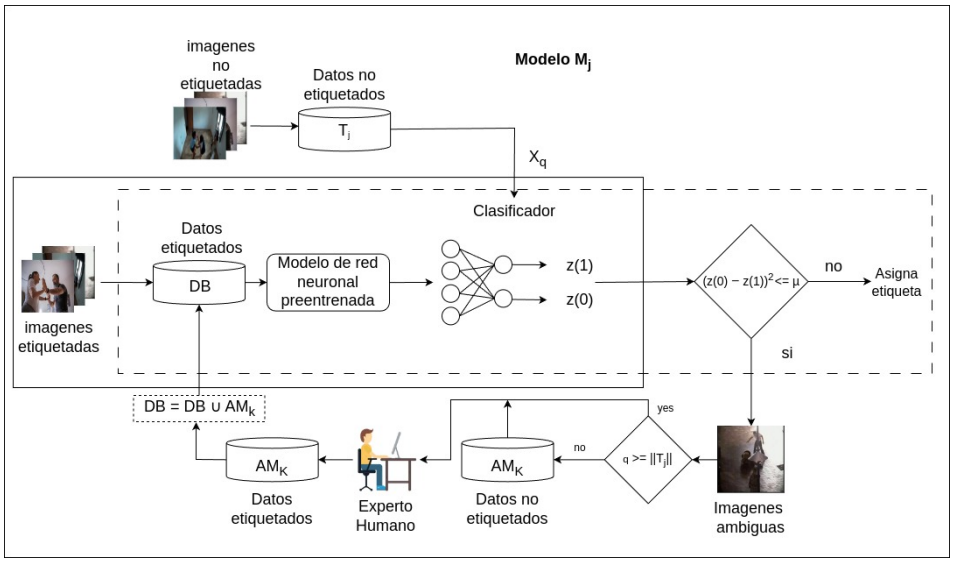


Figura 3.1: Metodología propuesta

A continuación se hacen referencia a los acrónimos de la Figura 3.1, como

primer etapa se construyó un modelo inicial utilizando el conjunto de datos previamente etiquetados (AIRTLab), la Figura 3.2 muestra el proceso del clasificador con un modelo preentrenado. Con la salida de la red se calcula el parámetro umbral μ. En esta etapa el valor inicial de μ es cero; es decir, en la primera etapa, el entrenamiento del modelo no está sesgado por el valor de μ.

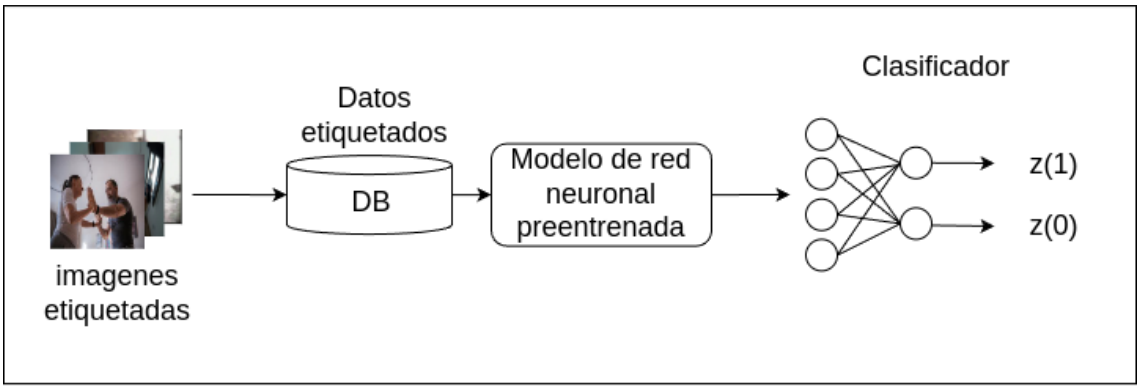


Figura 3.2: Construcción del clasificador con un modelo preentrenado

Entrenada la red se procede a mostrar al clasificador imágenes de un entorno desconocido, al clasificador llegan conjuntos de datos sin etiquetar (T_j) ver Figura 3.1, las imágenes o muestras de datos se reenvían al modelo, el cual asigna una etiqueta y calcula el valor de incertidumbre, si el valor de incertidumbre es mayor, el modelo considera esta muestra como ambigua y se añade a la base de datos AM_k, en caso contrario, se asigna a la imagen la etiqueta obtenida por el modelo. Al existir suficientes imágenes ambiguas en la base de datos AM_k, se envía a un experto humano para que las etiquete. Al no existir más imágenes en T_j, las imágenes ambiguas etiquetadas por el experto humano AM_k se agrega a la BD la cuál ahora contiene los datos de entrenamiento incluyendo los ahora etiquetados por el experto humano, el modelo se vuelve a entrenar con la BD actualizada.

3.1. Propuesta aprendizaje activo

Una red neuronal artificial sirve como un modelo matemático que mapea la entrada $\mathbf{x}$ a la salida $\mathbf{z}$. El vector $\mathbf{z}$ contiene las salidas de la red neuronal, y su dimensionalidad corresponde al número de clases C. En este trabajo, la dimensionalidad de $\mathbf{z}$ son dos clases: violencia física (VF) o no violencia física (NVF). Cada posición c en $\mathbf{z}_q$ se asigna a una clase concreta; en este caso, $\mathbf{z}_q[0]$ corresponde a la clase VF y $\mathbf{z}_q[1]$ a la clase NVF. Así, $\mathbf{z}_q[c]$ es la probabilidad de una muestra $\mathbf{x}_q \in c$, donde $c = 0, 1, 2, 3, ..., C$. El clasificador asigna la clase c a la muestra $\mathbf{x}_q$ si $\mathbf{z}_q[c]$ contiene el valor máximo de $\mathbf{z}_q$ [27].

En tareas de clasificación binaria ($C = 2$), si ($\mathbf{z}_q[0] << \mathbf{z}_q[1]$) o ($\mathbf{z}_q[0] >> \mathbf{z}_q[1]$), está clara la clase que el clasificador debe asignar a $\mathbf{x}_q$, pero cuando $\mathbf{z}_q[0] \approx \mathbf{z}_q[1]$, la decisión del clasificador podría ser errónea. Cuando las salidas $\mathbf{z}_q[0]$ y $\mathbf{z}_q[1]$ están muy sesgados hacia una clase, la decisión del clasificador es sencilla. Sin embargo, cuando estos resultados están próximos, el clasificador puede tener dificultades para tomar una decisión definitiva. Esta incertidumbre motiva el uso de un umbral para identificar muestras ambiguas, en las que la diferencia entre los resultados es insignificante, lo que puede dar lugar a clasificaciones erróneas.

El algoritmo 1 inicia el proceso de aprendizaje activo construyendo primero el clasificador inicial mostrado en la Figura 3.2, seguido de la ejecución del enfoque de aprendizaje activo.

Algoritmo 1 Propuesta de aprendizaje activo

Input: DB, $\mathrm{M}_j, T[]$; */* DB conjunto de datos del entrenamiento, M_0 modelo inicial, y T el conjunto de base de datos sin etiquetar. */*

1: **STAGE 1:**
2: $j \leftarrow 0$;
3: $\mu = 0$; */*Valor por default μ es cero.*/*
4: M_j =**OPTIMIZE**(M_j, DB, μ); */* Algoritmo 3 para construir el modelo inicial M_0 usando DB, conjunto de datos etiquetados. */*
5: μ = **THRESHOLD**(M_j, DB); */*Algoritmo 4 se emplea para obtener el mejor valor μ umbral.*
6: **STAGE 2:**
7: $j \leftarrow 1$;
/ T_j es un conjunto de datos no etiquetado y desconocido */*
8: **mientras** new unlabeled datasets T_j exist **hacer**
9: M_j =**ACTIVE-LEARNING** $(\mathrm{M}_{j-1}, DB, T_j, \mu)$; */* Algoritmo 2 se realiza devolviendo el modelo optimizado M_j */*
10: μ = **THRESHOLD**(M_j, DB); */* Re-calcular el umbral μ usando el Algoritmo 4. */*
11: $j \leftarrow j + 1$;
12: **fin mientras**

El algoritmo 2 delinea los pasos clave del enfoque de aprendizaje activo basado en un umbral. Incluye la clasificación por un modelo de red neuronal y la intervención de un experto humano, lo que facilita la mejora iterativa del clasificador mediante el etiquetado selectivo de los datos.

Algoritmo 2 Aprendizaje activo basado en umbral μ

Input: $\mathrm{M}_j, DB, T_j, \mu$;

Output: M_j; */* El modelo optimzado mediante aprendizaje activo */*

ACTIVE-LEARNING ( ):

1: **para** $q = 0$ **to** $\|T_j\|$ **hacer**
2: $\mathbf{z} \leftarrow \mathrm{M}_j(x_q)$; */* $\boldsymbol{z}$ es la salidad de la red neuronal a la muestra x_q */*
3: **si** $(\mathbf{z}(0) - \mathbf{z}(1))^2 \leq \mu$ **entonces**
4: $AM_k \leftarrow AM_k \cup x_q$; */* x_q es considerada ambigua, por lo que se incluye en AM_k. */*
5: **si no**
6: $x_q \leftarrow$ **MODEL_LABEL**(x_q); */* Establecer la etiqueta generada por el clasificador en x_q. */*
7: **fin si**
8: **fin para**
9: **si** $\|AM_k\| \neq 0$ **entonces**
10: $AM_k \leftarrow$ **HUMAN_LABEL**(AM_k); */* AM_k se envía al experto humano para que la etiquete; */*
11: $DB \leftarrow DB \cup AM_k$; */* Actualización DB con muestras ambiguas etiquetadas. */*
12: $\mathrm{M}_j =$ **OPTIMIZE** (M_j, DB, μ); */* Volver a optimizar con los datos actualizados DB. */*
13: **fin si**
14: **return** M_j; */* Se vuelve el modelo optimizado tras el aprendizaje activo. */*

El algoritmo 3 se enfoca en entrenar y evaluar el modelo utilizado en el conjunto de datos original y el conjunto de datos actualizado (DB), incorporando muestras ambiguas que han sido etiquetadas y añadidas al DB.

Algoritmo 3 Procedimiento para construir el modelo de red neuronal artificial M_j

Input: M_j, DB, μ; /* M_j *es el modelo de red neuronal artificial,* DB *es el conjunto de entrenamiento.* */

Output: M_j; /* *Retorna el modelo optimizado*/

OPTIMIZE ( ):

1: $k \leftarrow 0$; /* $k = 0, 1, 2, ..., K$, *donde* K *numero máximo de épocas* */
   /* *Obtiene el entrenamiento (*$DBTrain$*) y prueba (*$DBTest$*) conjunto de datos, donde* $DB = DBTrain \cup DBTest$; *and* $DBTrain \cap DBTest = \varnothing$; */
2: $DBTrain =$ **getRamdomly**$(DB, 70\,\%)$;
3: $DBTest =$ **getRamdomly**$(DB, 30\,\%)$;
4: **mientras** $(AUC_{k-2} \leq AUC_{k-1}$ **and** $k > 1)$ **or** $(k < K)$ **hacer**
5: **LEARNING**$(\mathrm{M}_j, DBTrain)$; /* *Entrena el modelo* M_j *con* DB */
6: $AUC_k \leftarrow$ **TEST**$(\mathrm{M}_j, DBTest)$; /* *probar el modelo* M_j *con* $DBTest$ */
7: **para** $q = 0$ **to** $\|DB\|$ **hacer**
8: $\mathbf{z} \leftarrow \mathrm{M}_j(x_q)$; /* $\boldsymbol{z}$ *es la salida de la red neuronal para la muestra* x_q */
9: **si** $(\mathbf{z}(0) - \mathbf{z}(1))^2 \leq \mu$ **entonces**
10: $AM_k \leftarrow AM_k \cup x_q$; /* x_q *es considerado ambiguo, por lo que se incluye en subconjunto* AM_k */
11: **fin si**
12: **fin para**
13: **si** $\|AM_k\| \neq 0$ **entonces**
14: $DBTrain \leftarrow DBTrain \cup AM_k$; /* *El conjunto de datos se actualiza con muestras ambiguas*/
15: **fin si**
16: $k \leftarrow k + 1$;
17: **fin mientras**
18: **return** M_j; /* *Optimización del modelo* */

3.2. Propuesta Umbral

El uso de un umbral μ (Ecuación. 3.2) permite detectar imágenes ambiguas; es decir aquellas imágenes en las que el clasificador podría clasificar de forma errónea. Cada conexión neuronal puede apreciarse como la probabilidad de que una imagen o muestra específica pertenezca a una clase determinada. En consecuencia, si las probabilidades de cada salida son muy diferentes entre ellas, la probabilidad de error del clasificador puede reducirse (Ecuación. 3.1).

1. El modelo M_j es construido a partir de DB.

2. M_j es probado DB.

3. Se eligen las salidas de las muestras de DB clasificadas incorrectamente por M_j, utilizando la Ecuación (3.1) para encontrar el umbral inicial μ_{ini}.

$$\mu_{ini} = \frac{\sum_{q=0}^{Q}(\mathbf{z}_q[0] - \mathbf{z}_q[1])^2}{Q}, \tag{3.1}$$

   Donde $Q = |DB|$.

4. La ecuación (3.2) obtiene el umbral final μ, en el que μ_{ini} inicial se incrementa en una constante $\triangle$ (obtenida por un experto humano), para dar un nivel de tolerancia al umbral μ. El proceso de fijación del valor $\triangle$ se realiza una sola vez, donde $\triangle$ es un valor pequeño ($0 < \triangle$ y $\triangle << 1$).

$$\mu = \mu_{ini} + \triangle, \tag{3.2}$$

La variable μ se calcula mediante los siguientes criterios: las salidas objetivo de la RNA suelen codificarse como 0 y 1. Para un problema de dos clases (clase A y clase B), codificando las salidas requeridas de la RNA como ((1, 0) y (0,

1)) respectivamente. Siendo estos valores las salidas objetivo de la RNA, y los valores finales esperados son emitidos por la RNA después del entrenamiento. En consecuencia, los posibles valores para μ son:

- $\mu \approx 1.0$ para muestras seguras, debido a que se espera que la RNA clasifique con alto nivel de precisión, se prevé que la salida de la RNA a todas las neuronas sean valores cercanos a (0,1) o (1,0).
- $\mu \approx 0.0$ para las muestras que se encuentran en la frontera de decisión, ya que se prevé que el clasificador no clasifique de manera adecuada, las salidas esperadas de la RNA para todas las neuronas son valores cercanos a (0.5, 0.5).
- $\mu \approx 0.5$ para las muestras medias, porque se espera que la RNA clasifique con menor precisión. Las muestras medias se encuentran entre las muestras seguras ($\mu \approx 1.0$) y la frontera de decisión ($\mu \approx 0.0$).

El proceso para obtener el mejor valor del umbral μ se describe en el Algoritmo 4.

3.3. Implementación

El modelo propuesto fue programado en Python 3.10.11 con pandas 2.1.0, numpy 1.23.5, y ejecutado en una estación de trabajo equipada con un procesador Intel Xeon E-2186G (12) a 4,700 GHz y una tarjeta gráfica NVIDIA GeForce RTX 3060 con 6 GB de memoria de vídeo y 64 GB de memoria RAM. Con NVIDIA-SMI 535.86.05 y la versión 12.2 de CUDA para la aceleración en la GPU.

Algoritmo 4 Obtención del umbral μ.

/ M_j es el modelo reciente, y DB es la muestra de entrenamiento */*

Input: M_j,DB;

Output: μ;

THRESHOLD( ):

1: $\mu_{ini} = 0$;
2: **para** $q = 0$ **to** $\|DB\|$ **hacer**
3: $\quad \mathbf{z}_q \leftarrow \mathrm{M}_j(x_q)$; */* $\boldsymbol{z}$ es la salida de la red neuronal x_q */*
4: $\quad \mathbf{d}_q \leftarrow$ **LABEL**(x_q, DB); */* $\boldsymbol{d}$ Corresponde a la etiqueta correcta x_q */*
5: $\quad$ **si** $(\mathbf{z}_q <> \mathbf{d}_q)$ **entonces**
6: $\quad\quad \mu_{ini} += (\mathbf{z}_q(0) - \mathbf{z}_q(1))^2$;
7: $\quad$ **fin si**
8: **fin para**
9: $\mu = \frac{\mu_{ini}}{\|DB\|} + \triangle$;
10: **return** μ;

Capítulo 4

Artículo de investigación

En este capítulo se anexa el artículo publicado en la revista Algorithms 2024 Volumen 17, Issue 7, 316 de MDPI con un factor de impacto en JCR de 1.8 (2023), como resultado la presente investigación.

algorithms

MDPI

Article

Threshold Active Learning Approach for Physical Violence Detection on Images Obtained from Video (Frame-Level) Using Pre-Trained Deep Learning Neural Network Models

Itzel M. Abundez [1], Roberto Alejo [1], Francisco Primero Primero [1], Everardo E. Granda-Gutiérrez [2], Otniel Portillo-Rodríguez [3,*] and Juan Alberto Antonio Velázquez [4]

[1] Tecnológico Nacional de México, Instituto Tecnológico de Toluca, Av. Tecnológico s/n, Colonia Agrícola Bellavista, Metepec 52149, Mexico; iabundezb@toluca.tecnm.mx (I.M.A.); ralejoe@toluca.tecnm.mx (R.A.); mm23281646@toluca.tecnm.mx (F.P.P.)
[2] Centro Univertsitario UAEM Atlacomulco, Universidad Autónoma del Estado de México, KM 60 Carretera Toluca-Atlacomulco, Atlacomulco 50450, Mexico; eegrandag@uaemex.mx
[3] Facultad de Ingeniería, Universidad Autónoma del Estado de México, Instituto Literario No. 100 Oriente, Toluca 50130, Mexico
[4] Ingeniería en Sistemas Computacionales, Instituto de Estudios Superiores de Jocotitlán, Carretera Toluca-Atlacomulco KM 44.8, Ejido de San Juan y San Agustín, Jocotitlán 50700, Mexico; juan.antonio@tesjo.edu.mx
* Correspondence: oportillor@uaemex.mx

Abstract: Public authorities and private companies have used video cameras as part of surveillance systems, and one of their objectives is the rapid detection of physically violent actions. This task is usually performed by human visual inspection, which is labor-intensive. For this reason, different deep learning models have been implemented to remove the human eye from this task, yielding positive results. One of the main problems in detecting physical violence in videos is the variety of scenarios that can exist, which leads to different models being trained on datasets, leading them to detect physical violence in only one or a few types of videos. In this work, we present an approach for physical violence detection on images obtained from video based on threshold active learning, that increases the classifier's robustness in environments where it was not trained. The proposed approach consists of two stages: In the first stage, pre-trained neural network models are trained on initial datasets, and we use a threshold (μ) to identify those images that the classifier considers ambiguous or hard to classify. Then, they are included in the training dataset, and the model is retrained to improve its classification performance. In the second stage, we test the model with video images from other environments, and we again employ (μ) to detect ambiguous images that a human expert analyzes to determine the real class or delete the ambiguity on them. After that, the ambiguous images are added to the original training set and the classifier is retrained; this process is repeated while ambiguous images exist. The model is a hybrid neural network that uses transfer learning and a threshold μ to detect physical violence on images obtained from video files successfully. In this active learning process, the classifier can detect physical violence in different environments, where the main contribution is the method used to obtain a threshold μ (which is based on the neural network output) that allows human experts to contribute to the classification process to obtain more robust neural networks and high-quality datasets. The experimental results show the proposed approach's effectiveness in detecting physical violence, where it is trained using an initial dataset, and new images are added to improve its robustness in diverse environments.

Keywords: physical violence detection; active learning; video images processing; deep learning; convolutional neural network

Citation: Abundez, I.M.; Alejo, R.; Primero Primero, F.; Granda-Gutiérrez, E.E.; Portillo-Rodríguez, O.; Antonio Velázquez, J.A. Threshold Active Learning Approach for Physical Violence Detection on Images Obtained from Video (Frame-Level) Using Pre-Trained Deep Learning Neural Network Models. *Algorithms* **2024**, *17*, 316. https://doi.org/10.3390/a17070316

Academic Editors: Frank Werner and Paolo Spagnolo

Received: 2 May 2024
Revised: 1 July 2024
Accepted: 16 July 2024
Published: 18 July 2024

1. Introduction

In recent years, the automatic detection of physical violence has been a challenge in human activity recognition because it involves a continuous analysis of human behavior [1]. Analysis at the frame level of a video has been widely studied for detecting physical violence. It is a process by which each individual frame of a video is analyzed and examined to identify signs of violent or aggressive behavior. This analysis involves looking for specific visual indicators, such as sudden movements, punching gestures, physical contact between people, or other patterns that may suggest a violent action. In performing this on a frame-by-frame basis, the system attempts to detect and categorize instances of violence as they occur over time, allowing for a detailed and granular understanding of the activity in the video without extracting temporal features [2].

Since the early 2000s, the analysis of video at frame level has demonstrated that it is possible to detect violence in images using only spatial feature extraction, using techniques such as LPB [3], HOG [4], FAST [5], SURF [6], SIFT [7], BRISK [8], and ORB [9], because of their ability to capture robust and distinctive spatial features that are invariant to common transformations such as scale, rotation, and illumination. These techniques identify key points in images that reflect crucial local details, such as edges, corners, and textures, essential for recognizing complex visual patterns associated with violence.

The bag-of-words (BoW) technique, which works at the frame level of a video, has been used to compactly and robustly represent the extracted local features, allowing capture of the distribution of local visual patterns that characterize physical violence without the need for exact spatial arrangement. Wang [10] used BoW to identify violence in a dataset containing 500 images with violent scenes and 1500 images without violence. When applying BoW to SIFT features, they achieved an accuracy of 85.7% ± 1.4%. When using HOG features, the accuracy was 84.3% ± 1.6%. In contrast, with LBP features, they achieved an accuracy of 90.1% ± 1.5%, demonstrating that combining robust spatial features with effective BoW representation enables accurate detection of physical violence in images.

However, despite the success achieved by analysis at the frame level of video, detecting physical violence in videos is still a significant challenge. Recently, it has been addressed using deep learning (DL) techniques, mainly artificial neural networks (ANNs). This approach aims to enhance the efficiency of classifying physical violence scenes extracted from various video sources, such as closed-circuit television systems (CCTV), smartphones, and other digital cameras used for surveillance tasks. Consequently, developing such methodologies has expanded the available tools, reducing the human effort required to identify physical violence situations accurately. Although the current literature and state-of-the-art research boast various works focused on achieving high accuracy rates in violence detection within videos, typically ranging between 90% and 99% (see Ref. [11]), it is worth noting that many of these models are tailored to specific datasets, which work only in those particular scenarios.

Convolutional neural networks (CNNs) stand out as the predominant models employed for violence classification in videos at the frame level, with some of the most notable pre-trained networks being VGG16 [12], VGG19, DenseNet18, 26, 50, 101, 152, ResNet2 38 [13], ResNet18, 34, ResNet50 [14], EfficientNet-B7, InceptionB3, and MobileNetV2 [15]. Although those models were proposed some years ago, they are still the focus of research on image classification tasks, as discussed above.

Strides in violence detection can be attributed to using pre-trained neural networks and the emergence of hybrid models. One prevalent approach involves employing these pre-trained models for spatial feature extraction, followed by a secondary network responsible for classification. Such methodologies have demonstrated precision percentages reaching up to 96% [16]. Another noteworthy implementation integrates ConvLSTM for temporal feature extraction and 3D ResNet50, 3D ResNet101, and 3D ResNet152 architectures for spatial feature extraction in conjunction with the UCF-Crime dataset, achieving a precision rate of 96% [17]. Recent advancements include a convolutional attention module (CBAM) to discern the dynamics among individuals requiring detection [18]. Other methods explored

for violence detection involve feature extraction through optical flow, as demonstrated in the work presented in Ref. [19]. Rendón-Segador et al. [20] employ optical flow as input, preceding the utilization of a spatial–temporal codifier with DenseNet121 and a bidirectional layer of 2D convolutional LSTM (BiConvLSTM2D), resulting in an accuracy of 99%. Similarly, Wang [21] experimented on four public violence detection datasets: Movies, Hockey, Crowd, and RWF-2000. The authors adopt a comparable approach, extracting two optical flows, concatenating them, and feeding them into the classifier, achieving precision rates of 0.86 and 1.0 for the Movies and RWF-2000 datasets, respectively.

Furthermore, several studies have focused on addressing the violence detection challenge through a multi-class approach, utilizing the UCF-Crime dataset to more precisely distinguish between various types of violence [22]. Similarly, Vosta [23] examines four variants (Binary, UCF-Crime original dataset containing 14 classes, 13 abnormal events grouped as one anomalous class, 4MajorCat, and NREF) of the same dataset (UCF-Crime). Likewise, Yousaf [24] experimented with cartoon animated clips to classify violence into three classes: safe, fantasy violence, and sexual nudity.

The utilization of data sourced from a variety of channels, including online social networks, public datasets, and video surveillance cameras (with or without sound), has emerged as a trend aimed at cultivating more robust models across diverse environments [2,16,25]. Nevertheless, as new videos in different scenarios are generated daily, relying solely on robust yet static models trained with diverse datasets still needs to be improved.

The unprecedented volume of available data presents a significant opportunity for developing deep learning models; however, it also poses challenges such as the scalability of traditional models, usability, and adaptability [26]. Moreover, a critical challenge associated with this abundance of data is converting raw data into high-quality labeled data, which is essential for enhancing predictive models' accuracy. It is crucial to recognize that constructing a training set entails a potentially computationally intensive process in terms of both time and resources [27].

Active learning has emerged as a collaborative strategy that involves human participation alongside machine learning algorithms to iteratively construct predictive models and training sets. This iterative approach mitigates the costs of acquiring labeled data while concurrently enhancing the prediction accuracy of machine learning models [28]. Moreover, active learning facilitates the incorporation of new information into datasets utilized by automatic violence detection models. Active learning has been studied for several years; however, there have been few studies, although it is a very important topic in machine learning, especially now that many data exist but there are few high-quality datasets [29]. Today, it remains a focus of study in the academic research community (for example, see Refs. [30–32]) because of its utility in building robust machine learning models and obtaining high-quality labeled datasets.

In this paradigm of human–machine interaction, where human involvement spans various stages such as training, optimization, or evaluation of machine learning models, it is referred to as "the human in the loop" or "in the circuit". The aim is to harness human cognitive abilities and expertise to augment the performance of machine learning models. Consequently, this hybrid approach makes human participation in data labeling feasible, wherein an algorithm selects the class-unlabeled data deemed most pertinent for human expert labeling [33]. Subsequently, these labeled data are incorporated into the training dataset, enabling the training of models with diverse data to foster robustness across different environments.

This paper introduces a threshold active learning approach to construct robust models for physical violence detection on images obtained from video across diverse environments. Our objective is to effectively identify instances of physical violence within video images depicting diverse environments sourced from heterogeneous datasets.

The proposed method offers two key advantages: firstly, it facilitates the expansion of the training dataset by incorporating labeled data deemed relevant by human experts, thereby enhancing model performance. Secondly, it makes the process of building new datasets from varied environments more efficient, thereby reducing the associated effort. Notably, our approach is characterized by its simplicity and versatility, making it adaptable to a wide range of sophisticated deep learning models.

The main contribution of the proposed approach is the process used to obtain a threshold μ (process based on the neural network outputs) that allows human experts to contribute to the classification process to obtain more robust neural networks and high-quality datasets.

2. Related Works

The challenge inherent in any image identification system is attempting for 100% efficiency, a goal often elusive due to intrinsic complexities. This is particularly true in endeavors such as identifying physical violence, where false negatives are regrettably common. To address this challenge, our research adopts an active learning approach, mixing the collaborative efforts of human experts and machine learning algorithms. Active learning serves as a strategic framework aimed at refining classifiers to generalize well on instances not encountered within the problem domain [33].

Active learning finds its application in various domains, including image classification and object detection. Li and Guo [34] propose a method that employs a measure of uncertainty, coupled with information density and an adaptive information framework, to select instances from unlabeled sets based on their informativeness. By combining measures of uncertainty with adaptive information density, this approach effectively guides the selection of instances for labeling, thereby enriching the labeled dataset.

Beluch et al. [35] investigate and compare different active learning methods in the context of image classification using high-dimensional data and CNNs. The paper evaluates ensemble-based methods against Monte Carlo dropout and geometric approaches, finding that ensembles perform better and provide more calibrated predictive uncertainties, which are crucial for active learning algorithms. The study includes experiments on various datasets, including MNIST, CIFAR-10, and ImageNet, as well as a large, class-imbalanced diabetic retinopathy dataset. The results demonstrate that ensemble-based active learning is particularly effective in managing class imbalance during the acquisition phase.

In [36], Sener and Savarese introduce a core-set selection approach for active learning in CNNs, providing a theoretical framework to evaluate subset performance and demonstrating significant improvements in image classification tasks over existing methods. This method involves choosing a subset of data points such that a model trained on this subset performs competitively on the remaining data.

Pan et al. [37] produce significantly different results for a given sample based on the principle that if two deep neural networks (DNNs) of the same structure are trained on the same dataset, that sample should be selected for additional training. The dual active sampling on batch-incremental active learning approach has proven useful. It simplifies implementation and reduces computational requirements compared to other state-of-the-art methods, leading to better results on the CIFAR-10 dataset with reduced computational time compared to the core-set method.

Carbonneau et al. [38] introduce new methods for bag-level aggregation in multiple instances of active learning (MIAL), a technique used in instance classification problems. The study focuses on reducing labeling costs by identifying the most informative instances and querying the expert for their labels. The proposed method outperforms existing methods in various application domains, including aggregated informativeness and cluster-based aggregative sampling. This research is significant as it addresses the limitations of single-instance active learning methods in multiple-instance learning problems.

Recent efforts, exemplified by Chen et al. [30], emphasize using large datasets without full labeling. Chen et al. propose a method for object detection that amalgamates active and supervised learning techniques. Initially, labeled data are utilized to train a detection model in a semi-supervised manner. Subsequently, the stability of the unlabeled data is assessed, with low-stability instances manually labeled, while those with high uncertainty are pseudo-labeled using predictions from the detector. Integrating manual and pseudo-labeling contributes to refining the detection model, yielding promising results with a reported mean average precision (mAP) of 79.2%.

Another notable approach, as presented by Li et al. [31], uses active learning based on competence to classify unlabeled images by their complexity. By utilizing a competence function in conjunction with prediction probabilities, this method effectively identifies images with varying degrees of difficulty, iteratively augmenting the dataset. Training the model with stochastic gradient descent (SGD) and a learning rate of 0.1, Li et al. evaluated three active learning methods, selection based on similarity, probability-based similarity selection, and competence-based active learning, achieving a final accuracy of 92%.

Additionally, Mohammadi et al. [32] introduce a framework utilizing pre-trained vision transformers trained on the RWF dataset as "experts" alongside a routing system and a reinforcement learning-based classifier. This approach dynamically activates experts concerning each video clip, determining its category (fight or no-fight) through various selections and categorizations. The router module orchestrates the classification process, deciding whether to select a clip's category immediately or to gather further information by activating vision transformers, thus concluding the classification process by selecting the appropriate category.

3. Theoretical Foundations

An artificial neural network (ANN) comprises numerous interconnected layers, each composed of multiple neurons. There are two primary types of ANNs: feed-forward and recurrent. The distinction lies in their connectivity patterns. In recurrent neural networks (RNNs), neurons may be connected to other neurons within the same layer or to neurons in layers that are neither previous nor subsequent to it. Conversely, in feed-forward neural networks (FNNs), neurons in a given layer are only connected to neurons in the subsequent layer via synaptic weights (w) [39].

In the feed-forward process, the input to a neuron i in the l-th layer is determined by the product of the results of the activation function $\varphi(\cdot)$ (Equation (1)), derived from the preceding layer $(l-1)$, and the weight vector $\mathbf{w}$ of layer l, which includes the bias weight b_l. The output of the neuron corresponds to a spatial transformation of s_i^l by the activation function (Equation (2)) [40].

$$s_i^l = \sum_h w_{hi}^l \varphi_h^{(l-1)}(s_h^{(l-1)}), \tag{1}$$

$$\varphi^{(l)}(s_i^l) = \varphi^{(l)}(\mathbf{w}^l, s_i^l). \tag{2}$$

The output j of the last layer in the ANN is $z_j = \hat{f}(\mathbf{x}, \mathbf{w})$ (Equation (3)); it depends on the parameters $\mathbf{w}$ of all hidden layers and it estimates the transformation $\{f : \mathbb{R}^N \rightarrow \mathbb{R}^C\}$, which partitions the input space $\mathbb{R}^N$ into C classification regions ($\mathbb{R}^C$) [41] by using a linear combination $\varphi_h(\mathbf{x})$.

$$\hat{f}(\mathbf{x}, \mathbf{w}) = \sum_{h=1} w_{jh}^l \varphi_h^{(l-1)}(\mathbf{x}), \tag{3}$$

with $\mathbf{x}$ being the input vector ($\mathbf{x} \in \mathbb{R}^N$) and, when $l = 1$, $s_i = \sum_h w_{hi} x_h$ (i.e., s_i is the *i-th* neuron of the first hidden layer). w_i^l is the weight of the *i-th* input that is connected from the h neuron of the *l-th* hidden layer to the $(l-1)$*-th* hidden layer. φ_h^{l-1} is the *h-th* output node on the $(l-1)$*-th* hidden layer, where L is the total number of layers in the ANN.

3.1. Convolutional Neural Network

A convolutional neural network (CNN) is a type of ANN where the hidden layers consist of a set of convolutions that typically are a combination of linear and nonlinear operations (i.e., convolution operation and activation function) [42]. In addition, the CNN usually includes pooling and fully connected layers.

The convolutional layer constructs a feature map to extract key features from input data. Here, convolutional kernels function as local filters, operating on sequential data to produce non-invariant local features. Conversely, the pooling layer plays a fundamental role in reducing the dimensionality of the feature vector, preserving only the most pertinent features. By extracting essential features within fixed-length windows, the pooling layer aids in making the subsequent processing steps more efficient [43,44]. Moreover, fully connected layers serve as linear classifiers, facilitating the assignment of predicted classes to specific inputs $\mathbf{x}_p$. CNNs are formidable tools for feature extraction, particularly adept in tasks involving image data modeling, summarization, and classification [45,46].

In recent years, the utilization of pre-trained models for specific tasks has gained prominence, a practice known as transfer learning. This approach is invaluable when resources for training a neural network are limited or inadequate, allowing for adapting a model to varying classification problems. Notable pre-trained CNN models such as DenseNet121, EfficientNetB0, InceptionV3, MobileNetV2, ResNet50, and VGG16 offer the advantage of requiring fewer filters and parameters while yielding favorable results. Consequently, these models have found application in diverse domains, from processing biomedical images to violence detection in videos [20,47,48]. Below, we provide a summary overview of the three pre-trained models we used in our experimentation.

- **DenseNet121**. The DenseNet model [49] is a convolutional neural network comprising 121 densely connected layers. DenseNet utilizes MaxPool2D and AveragePool3D reduction layers with (2,2) and (7,7) pool sizes, respectively. With its robustness and efficiency, DenseNet requires fewer filters and parameters to achieve favorable performance.
- **EfficientNetB0**. It is a convolutional neural network model that utilizes the ImageNet database for classification tasks across 1000 object categories. One of its key innovations is the adoption of "compound scaling", a technique designed to enhance model performance by increasing the resolution of input images. This increase in resolution necessitates adjustments in both the width (number of channels) and depth (number of layers) of the network to effectively capture the additional details provided by higher-resolution images. This approach involves applying a fixed set of coefficients to uniformly scale the network architecture's width, depth, and resolution. It was specifically developed to improve the scalability of CNNs to achieve better model performance and efficiency. EfficientNet-B0 serves as the baseline model, characterized by 5.3 million parameters, and accepts input images of size 224×224. Subsequent variants, denoted by increasing numbers (e.g., B1, B2, B3, etc.), further incorporate width, depth, and resolution adjustments to optimize performance across different computational constraints and task requirements [50].
- **MobileNetV2**. The MobileNetV2 architecture consists of two main types of building blocks: residual blocks with stride and blocks with a stride of 2 to reduce the size. Each block comprises three layers: the first layer is a 1×1 convolutional layer followed by a rectified linear unit 6 (ReLU6) activation function, the second layer is a depthwise convolutional layer, and the third layer is another 1×1 convolutional layer used to linearly combine the output channels of the depthwise convolution with a ReLU6 activation function [51].

3.2. Physical Violence

Physical violence occurs when a person violates the bodily space of another without their consent, for example, by hitting, pulling, or pushing them. It also may include causing physical injuries with objects (weapons or other objects) or with the aggressor's body (punching, pulling, kicking, pushing). Physical violence has emotional and health consequences. The General Law on Women's Access to a Life Free of Violence [52], in its Article 6, section II, defines physical violence as "A type of violence referring to any act that inflicts non-accidental harm, using physical force or some type of weapon or object that may or may not cause injuries whether internal, external or both".

This work is focused on improving the automatic identification of physical violence (PV) or non-physical violence (NPV) in images obtained from video in diverse environments using a threshold active learning approach.

4. Proposed Approach

In this paper, we introduce an active learning approach based on a threshold parameter (μ) to enhance classifier performance, construct robust models capable of operating effectively across diverse environments for detecting physical violence in videos by analyzing each photogram, and build high-quality datasets with the intervention of a human expert. This approach comprises two stages, presented in Sections 4.1 and 4.2, and outlined in Algorithms 1–4.

Algorithm 1 initiates the active learning process by first constructing the initial classifier, followed by the execution of the active learning approach. Algorithm 2 delineates the key steps of the active learning approach based on a threshold. This includes classification by a neural network model and human expert intervention, facilitating iterative improvement of the classifier through selective labeling of data. Algorithm 3 focuses on training and testing the models using both the original and updated datasets (DB), incorporating ambiguous samples that have been labeled and added to DB. Finally, Algorithm 4 details the process used to obtain the threshold μ.

Algorithm 1 Proposed active learning approach

Input: DB, M_j, $T[]$; */* DB is the training dataset, M_0 is the initial model, and T is a set of unlabeled datasets. */*

STAGE 1:
$j \leftarrow 0$;
$\mu = 0$; */* Default value for μ is zero.*/*
M_j =**OPTIMIZE**(M_j, DB, μ); */* Algorithm 3 is performed to built the initial model M_0 using DB, which is a labeled dataset. */*
μ = **THRESHOLD**(M_j, DB); */*Algorithm 4 is employed to obtain the best value to μ threshold.*
STAGE 2:
$j \leftarrow 1$;
/ T_j is an unlabeled and unknown dataset */*
while new unlabeled datasets T_j exist **do**
    M_j =**ACTIVE-LEARNING** ($\mathrm{M}_{j-1}, DB, T_j, \mu$); */* Algorithm 2 is performed returning the optimized model M_j*/*
    μ = **THRESHOLD**(M_j, DB); */* Re-calculate the threshold μ using Algorithm 4. */*
    $j \leftarrow j + 1$;
end while

Algorithm 2 Active learning based on a threshold μ

Input: $\mathrm{M}_j, DB, T_j, \mu$;
Output: M_j; /* *The optimized model using active learning* */
ACTIVE-LEARNING ( ):

for $q = 0$ **to** $\|T_j\|$ **do**
  $\mathbf{z} \leftarrow \mathrm{M}_j(x_q)$; /* $\mathbf{z}$ *is the neural network output to sample* x_q */
  if $(\mathbf{z}(0) - \mathbf{z}(1))^2 \leq \mu$ **then**
    $AM_k \leftarrow AM_k \cup x_q$; /* x_q *is considered ambiguous, so it is included in* AM_k. */
  else
    $x_q \leftarrow$ **MODEL_LABEL**(x_q); /* *Set the label generated by the classifier to* x_q. */
  end if
end for
if $\|AM_k\| \neq 0$ **then**
  $AM_k \leftarrow$ **HUMAN_LABEL**(AM_k); /* AM_k *is sent to human expert to be labeled;* */
  $DB \leftarrow DB \cup AM_k$; /* *Update DB with labeled ambiguous samples.* */
  $\mathrm{M}_j =$ **OPTIMIZE** (M_j, DB, μ); /* *Re-optimize the model with the updated DB.* */
end if
return M_j; /* *The optimized model after active learning is returned.* */

Algorithm 3 Procedure to build the optimized artificial neural network model M_j

Input: M_j, DB, μ; /* M_j *is the neural network model, DB is the training dataset.* */
Output: M_j; /* *Return the optimized model* */
OPTIMIZE ( ):

$k \leftarrow 0$; /* $k = 0, 1, 2, ..., K$, *where K is the maximum number of epochs* */
  /* *Obtain the training (DBTrain) and test (DBTest) datasets, where DB = DBTrain* $\cup$ *DBTest; and DBTrain* $\cap$ *DBTest* $=\varnothing$; */
$DBTrain =$ **getRamdomly**$(DB, 70\%)$;
$DBTest =$ **getRamdomly**$(DB, 30\%)$;
while $(AUC_{k-2} \leq AUC_{k-1}$ **and** $k > 1)$ **or** $(k < K)$ **do**
  LEARNING$(\mathrm{M}_j, DBTrain)$; /* *Train the model* M_j *with DB* */
  $AUC_k \leftarrow$ **TEST**$(\mathrm{M}_j, DBTest)$; /* *Test the model* M_j *with DBTest* */
  for $q = 0$ **to** $\|DB\|$ **do**
    $\mathbf{z} \leftarrow \mathrm{M}_j(x_q)$; /* $\mathbf{z}$ *is the neural network output for sample* x_q */
    if $(\mathbf{z}(0) - \mathbf{z}(1))^2 \leq \mu$ **then**
      $AM_k \leftarrow AM_k \cup x_q$; /* x_q *is considered ambiguous, so it is included in subset* AM_k */
    end if
  end for
  if $\|AM_k\| \neq 0$ **then**
    $DBTrain \leftarrow DBTrain \cup AM_k$; /* *Dataset is updated with ambiguous samples*/
  end if
  $k \leftarrow k + 1$;
end while
return M_j; /* *The optimized model* */

Algorithm 4 Obtaining the threshold μ

/ M_j is the recent model, and DB the training dataset */*
Input: M_j,DB;
Output: μ;
THRESHOLD( ):

1: $\mu_{ini} = 0$;
2: **for** $q = 0$ **to** $\|DB\|$ **do**
3: $\mathbf{z}_q \leftarrow M_j(x_q)$; */* $\mathbf{z}$ is the neural network output to sample x_q */*
4: $\mathbf{d}_q \leftarrow$ **LABEL**(x_q, DB); */* $\mathbf{d}$ is the correct label to sample x_q */*
5: **if** $(\mathbf{z}_q <> \mathbf{d}_q)$ **then**
6: $\mu_{ini} += (\mathbf{z}_q(0) - \mathbf{z}_q(1))^2$;
7: **end if**
8: **end for**
9: $\mu = \frac{\mu_{ini}}{\|DB\|} + \triangle$;
10: **return** μ;

4.1. Stage 1

In the first stage, the initial model is built using a labeled dataset DB (Algorithm 1, line 4). Immediately, the threshold parameter μ (Algorithm 4) is calculated (Algorithm 1, line 5). Observe that in the first stage the initial value of μ is zero (Algorithm 1, line 3); i.e., in the first stage, the training of the model is not biased by the value of μ. This stage comprises lines 2–5 of Algorithm 1.

4.2. Stage 2

The second stage is enclosed in lines 7–12 of Algorithm 1, in which the model is tested using unseen and unknown datasets (T_j). T_j differs from the training dataset ($DB \neq T_j$), ensuring an evaluation of the model's performance in novel environments. The threshold parameter μ (obtained in the first stage) is utilized to identify images (x_q) that the classifier deems ambiguous or challenging to classify (Algorithm 2, line 3). These ambiguous images represent scenarios where discerning physical violence from non-physical violence scenes may be particularly difficult, such as instances involving strong hugs or warm greetings, among others.

The selected P ambiguous images (x_p) are stored in a subset AM_k, where $AM_k = AM_k \cup_{p=1}^{P} (x_p)$ (Algorithm 2, line 4).

When all samples in T_j are processed by the model (i.e., when $q \geq \|T_j\|$), AM_k is then forwarded to a human expert to be labeled with its correct class: physical violence or non-physical violence scene (Algorithm 2, line 10). Subsequently, the dataset DB is updated with the labeled samples, i.e., $DB = DB \cup AM_k$ (Algorithm 2, line 11), and the model is retrained using DB (Algorithm 2, line 12). This process continues until no unseen datasets are available (Algorithm 1, lines 5–9).

The **OPTIMIZE** procedure (Algorithm 3) is used to build the optimized artificial neural network model M_j. It employs the μ parameter to improve the classifier performance, duplicating the samples the model finds difficult to learn. In this procedure, the model is retrained $k = 0, 1, 2, 3, ..., K$ times; i.e., the process is repeated for a total of K epochs or while AUC increases. The human expert does not intervene in this process because the datasets employed are labeled.

The two above stages (Sections 4.1 and 4.2) simulate an online system where the classifier is initially constructed using a labeled dataset and subsequently deployed in a real-world environment. When the model encounters ambiguity in classifying an image, it is referred to a human expert to reduce classification errors. This approach takes advantage of ambiguous images to enhance the model's performance in unseen scenarios.

4.3. Threshold μ

An artificial neural network serves as a mathematical model mapping input $\mathbf{x}$ to output $\mathbf{z}$, typically represented by a function $f(W)$, where $f : \mathbb{R}^N \rightarrow \mathbb{R}^C$. Here, $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{z} \in \mathbb{R}^C$ denote the input and output vectors, respectively [39].

Vector $\mathbf{z}$ contains the neural network outputs, and its dimensionality corresponds to the number of classes C. In this work, the dimensionality of $\mathbf{z}$ is two because we classify only two classes: physical violence (PV) or non-physical violence (NPV). Each position c in $\mathbf{z}_q$ is assigned to a particular class; in our case, $\mathbf{z}_q[0]$ corresponds to class PV and $\mathbf{z}_q[1]$ to the NPV class. Thus, $\mathbf{z}_q[c]$ is the likelihood of a sample $\mathbf{x}_q \in c$, where $c = 0, 1, 2, 3, ..., C$. The classifier assigns the class c to sample $\mathbf{x}_q$ if $\mathbf{z}_q[c]$ contains the maximum value of $\mathbf{z}_q$ [40].

In binary classification tasks ($C = 2$), if ($\mathbf{z}_q[0] << \mathbf{z}_q[1]$) or ($\mathbf{z}_q[0] >> \mathbf{z}_q[1]$), it is clear the class that the classifier should assign to $\mathbf{x}_q$, but when $\mathbf{z}_q[0] \approx \mathbf{z}_q[1]$, the classifier decision could be wrong. When the outputs $\mathbf{z}_q[0]$ and $\mathbf{z}_q[1]$ are significantly skewed towards one class, the classifier's decision is straightforward. However, when these outputs are close, the classifier may struggle to make a definitive decision. This uncertainty motivates the use of a threshold to identify ambiguous samples, where the difference between the outputs is negligible, potentially leading to misclassifications.

Prior studies have explored the effectiveness of employing thresholds to improve classification accuracy in scenarios where ambiguous samples are prevalent [53,54]. Determining an appropriate threshold value μ involves an iterative process, typically conducted using the initial dataset DB. Through trial and error, a threshold value that effectively distinguishes between unambiguous and ambiguous samples can be set, aiding in the refinement of classification models. This iterative process is described as follows:

1. The model M_j is built using DB.
2. M_j is tested with DB.
3. We choose the outputs from the samples of DB classified incorrectly by M_j, and we use Equation (4) to find the initial threshold μ_{ini}.

$$\mu_{ini} = \frac{\sum_{q=0}^{Q} (\mathbf{z}_q(0) - \mathbf{z}_q(1))^2}{Q}, \tag{4}$$

   where $Q = \|DB\|$.
4. Equation (5) obtains the final threshold μ, in which the initial μ_{ini} is increased by a constant $\triangle$ (obtained in a trial-and-error process performed by a human expert), to give a certain level of tolerance to threshold μ. The process for setting the $\triangle$ value is performed only one time, where $\triangle$ is a small value ($0 <\triangle$ and $\triangle << 1$). Algorithm 4 summarizes the process employed to obtain the best value for the μ threshold.

$$\mu = \mu_{ini} + \triangle \, . \tag{5}$$

Using a threshold μ takes advantage of the neural network output meaning to identify ambiguous images, i.e., those images where the classifier could misclassify. Each neural network output can be seen as the likelihood that a specific image or sample belongs to a determined class [39,40]. Thus, if the likelihoods of each output are very different between them, then the likelihood of error in the classifier may be reduced, as has been studied in Refs. [53,54].

4.4. Computational Complexity Analysis of the Threshold Active Learning Approach

The time-complexity (big-$\mathcal{O}$ notation) of Algorithm 1 is expressed in Equation (6), where $\mathcal{O}(OPT(DB, T_j)$ is the complexity of the **OPTIMIZE** procedure (Algorithm 3), J is the number of unlabeled datasets, and $\mathcal{O}(AL(M_{j-1}, DB, T_j))$ is the complexity of the **ACTIVE-LEARNING** procedure (Algorithm 2).

$$\mathcal{O}(\mathrm{OPT}(DB, T_j)) + J \cdot \mathcal{O}(\mathrm{AL}(M_{j-1}, DB, T_j)) \tag{6}$$

The complexity $\mathcal{O}(OPT(DB, T_j)$ of the **OPTIMIZE** sub-algorithm (Algorithm 3) can be approximated by considering the main operations within the while loop and the nested for loop, as follows:

- The complexity of training the model **LEARNING**(M_j, DB) depends on the learning algorithm used based on the pre-trained model. Suppose it is $\mathcal{O}(L(DB))$.
- The complexity of testing the model **TEST**(M_j, T_j) depends on the size of the test set. Suppose it is $\mathcal{O}(T(T_j))$.
- The for loop has a complexity of $\mathcal{O}(\|DB\|)$ to iterate over the dataset and compute the neural network output.

Therefore, the complexity of the **OPTIMIZE** sub-algorithm for k iterations, where K is the maximum number of epochs, is approximately

$$\mathcal{O}(OPT(DB, T_j) = \mathcal{O}(K) \times (\mathcal{O}(L(DB)) + \mathcal{O}(T(T_j)) + \mathcal{O}(\|DB\|)) \tag{7}$$

Similarly, to calculate the complexity $\mathcal{O}(AL(M_{j-1}, DB, T_j))$ of the **ACTIVE-LEARNING** sub-algorithm, we can consider the main operations within the for loop and the invoked functions:

- **For loop:**
  - The complexity of iterating over the test dataset T_j is $\mathcal{O}(\|T_j\|)$.
  - Within the loop, the operation $M_j(x_q)$ has a complexity dependent on the selected pre-trained model, which can be considered $\mathcal{O}(M)$.
  - The operations of comparison and updating the ambiguous set have a constant complexity $\mathcal{O}(1)$.
- **Human labeling and updating *DB*:**
  - The function **HUMAN_LABEL**(AM_k) will depend on the time taken by human experts but can be considered to be $\mathcal{O}(\|AM_k\|)$ in the worst case.
  - The operation of updating the dataset $DB \leftarrow DB \cup AM_k$ has a complexity of $\mathcal{O}(\|AM_k\|)$.
- **Re-optimization of the model:**
  - The complexity of the function **OPTIMIZE**(M_j, DB, DB) is the same as that of the **OPTIMIZE** sub-algorithm, denoted as $\mathcal{O}(\text{OPT}(DB))$, shown previously.

Therefore, the complexity of the **ACTIVE-LEARNING** sub-algorithm is approximately

$$\mathcal{O}(AL(M_{j-1}, DB, T_j)) = \mathcal{O}(\|T_j\|) \times \mathcal{O}(M) + O(\|AM_k\|) + \mathcal{O}(\text{OPT}(DB)) \tag{8}$$

Then, to determine the general big-$\mathcal{O}$ order of the complexity of the **MAIN** algorithm (Algorithm 1), we need to observe the dominant terms in the total complexity expression, $\mathcal{O}(MAIN())$, and simplify them:

1. Complexity of training and testing (**OPTIMIZE**):
   - $\mathcal{O}(K) \times \mathcal{O}(L(DB))$;
   - $\mathcal{O}(K) \times \mathcal{O}(T(T_j))$;
   - $\mathcal{O}(K) \times \mathcal{O}(\|DB\|)$.
2. Complexity of **ACTIVE-LEARNING**:
   - $\mathcal{O}(\|T_j\| \cdot M)$;
   - $\mathcal{O}(\|AM_k\|)$;
   - $J \times \mathcal{O}(K) \times (\mathcal{O}(L(DB)) + \mathcal{O}(T(T_j)) + \mathcal{O}(\|DB\|))$.

We assume that:

- $\mathcal{O}(L(DB))$ is the complexity of training (the **LEARNING** procedure in Algorithm 3), which can be very high if the model is complex.
- $\mathcal{O}(T(T_j))$ is the complexity of testing, generally lower than that of training.

- $\mathcal{O}(M)$ is the complexity of obtaining the model output, which may depend on the number of layers and parameters of the neural network model.
- $\mathcal{O}(\|DB\|)$ and $\mathcal{O}(\|T_j\|)$ are proportional to the sizes of the training and test datasets.

Generally, training (and thus the **OPTIMIZE** procedure) will be the most expensive operation. Assuming $L(DB)$, $T(T_j)$, and operations on the data ($\|DB\|$ and $\|T_j\|$) are of the same order of magnitude, we can consider the dominant terms:

1. Complexity of training (in the **OPTIMIZE** procedure): $\mathcal{O}(K) \times \mathcal{O}(L(DB))$;
2. Complexity of the while loop (in the **ACTIVE-LEARNING** procedure): $J \times (\mathcal{O}(\|T_j\| \cdot M) + \mathcal{O}(\|AM_k\|) + \mathcal{O}(K) \times \mathcal{O}(L(DB)))$.

Considering that J is the number of iterations of the while loop and K is the number of epochs, the overall complexity can be summarized and simplified in terms of the most dominant components:

$$\mathcal{O}(MAIN()) = \mathcal{O}(K \cdot L(DB)) + J \times (\mathcal{O}(\|T_j\| \cdot M) + \mathcal{O}(\|AM_k\|) + \mathcal{O}(K \cdot L(DB))) \quad (9)$$

If we assume that operations on the data ($\|DB\|$ and $\|T_j\|$) are of lower order compared to the model iterations:

$$\mathcal{O}(MAIN()) = \mathcal{O}(J \cdot K \cdot L(DB)) + \mathcal{O}(J \cdot \|T_j\| \cdot M) + \mathcal{O}(J \cdot \|AM_k\|) \quad (10)$$

The most dominant term, considering that K and J can be large, is

$$\mathcal{O}(MAIN()) = \mathcal{O}(J \cdot K \cdot L(DB)) \quad (11)$$

This implies that the complexity order is dominated by the number of iterations of the while loop J, the number of epochs K, and the complexity of the training process $L(DB)$, i.e., the **LEARNING** procedure in Algorithm 3.

As can be seen, the total complexity of the proposed approach is essentially dependent on the selected pre-trained model. That is, in the worst case, each of the evaluated models will dominate the total complexity. Although they all have unique characteristics, it can essentially be said that their complexity $\mathcal{O}(L(DB) = \mathcal{O}(L(DB)_1) + \mathcal{O}(L(DB)_2)$ depends on the convolutional (Equation (12)) and fully connected (Equation (13)) layers [55,56].

$$\mathcal{O}(L(DB)_1) = (\sum_{n=1}^{d} c_{n-1} \cdot S_n^2 \cdot f_n \cdot l_n^2) \cdot r \cdot b \quad (12)$$

where d is the depth of the convolutional layer, l_n is the length of the output feature map, f_n is the number of filters in the n-th later, S_n is the length of the filter, c_{n-1} is the number of input channels in the l-th layer, r is the learning rate, and b is the batch size.

$$\mathcal{O}(L(DB)_2) = \sum_{l=1}^{f} D \cdot W \cdot H \cdot N \quad (13)$$

where l is the depth of the fully connected layer, D is the dimension of the input/output channel, W is the width of the input, H is the height of the input, and N is the number of outputs.

4.5. Summary of the Proposed Approach

Finally, in order to better understand the proposed method, we present a visual summary in Figure 1, where its main components are presented. The first neural network model is built using a labeled dataset DB after the model works in an unknown environment. It is observed that unlabeled datasets (T_j) arrive at the classifier; thus, images or data samples are forwarded in the model, which must assign a label, but if it has uncertainty about the correct label, the model considers this sample as ambiguous and it is added to AM_k.

Otherwise, the label obtained by the model is assigned to the image. Thus, when enough ambiguous images exist in AM_k, it is sent to a human expert for labeling. Labeled AM_k is added to DB, and the model is retrained with updated DB, which now contains all ambiguous images correctly classified by human experts.

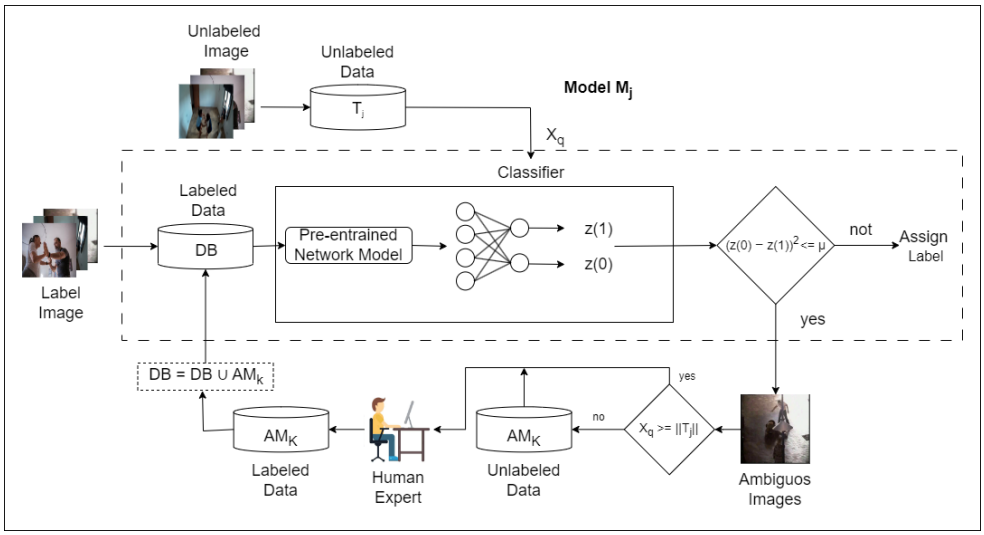


Figure 1. Summary of the main components of the proposed active learning approach.

5. Materials and Methods

5.1. Data Sources

Utilizing three distinct publicly available datasets enriches the diversity and realism of the training and evaluation process in our work. By incorporating diverse datasets, our study aims to train and evaluate the physical violence detection model across various scenarios, enhancing its adaptability and robustness in real-world applications. The datasets are:

1. AIRTLab dataset: This dataset presents scenes depicting everyday situations that may be misinterpreted by the algorithms as violent actions, such as hugs, clapping, and greetings, performed by nonprofessional actors. Including such scenarios challenges the classifier to discern subtle differences between benign and aggressive behaviors [57].
2. Real-Life Violence Situations (RVLS) dataset: This dataset comprises images extracted from videos depicting real-life situations and provides a comprehensive collection of diverse scenarios encountered in various environments [58]. By focusing specifically on instances of violence, this dataset facilitates the classifier's training in identifying and classifying violent behaviors accurately. The images in this dataset have been carefully selected and captured from multiple perspectives, incorporating a diversity of angles and different contexts or backgrounds to ensure wide variability and improve the effectiveness of recognition and analysis algorithms.
3. Pexels [59] is an online platform renowned for its vast collection of free stock photos and videos. The Pexels dataset was obtained by carefully selecting images from this platform in the context of this study. These images were specifically chosen to represent real-life scenes that do not contain physical violence, thus complementing the RVLS dataset, which focuses on violence images.
   Including Pexels images in the study is essential to evaluate the deep learning model's ability to distinguish between violent and non-violent scenes. By training and testing the model with images from both categories, researchers can assess its accuracy and robustness in classifying physical violence in various contexts.

The diversity of images available on Pexels allows researchers to select a representative sample of non-violent scenes that reflect the complexity of the real world. This ensures that the model learns to identify physical violence and differentiate it from everyday situations that could be misinterpreted as violent.

4. Smart-City CCTV Violence Detection (SCVD) dataset: The SCVD dataset offers a unique compilation of video footage captured through closed-circuit television (CCTV) systems deployed in urban settings [60]. This dataset accounts for variations in viewing angles, image quality, and recording contexts inherent to CCTV footage, which can significantly impact the effectiveness of violence detection algorithms. Notably, the SCVD dataset introduces a specific category dedicated to the identification of weapons, broadening the scope of violence detection to include any objects that could be employed harmfully. This expanded perspective enhances the classifier's ability to detect threats in urban environments and provides a different perspective on potential risks. Additionally, licensing the SCVD dataset under the "CC BY-NC-SA 4.0" license promotes collaboration and knowledge sharing within the research community while ensuring proper attribution to the original authors.

The heterogeneity in the collection of images is intended to simulate the variability of situations that artificial intelligence systems could face in real-world applications, from security systems to developments in artificial intelligence aimed at understanding complex social contexts.

It is important to note that all images included in the datasets used in this work are in the public domain. This means that they have been released from any existing copyright, allowing their use, redistribution, modification, and exploitation without restrictions or the need to request permission from the original authors. This feature facilitates their incorporation into research projects, software development, and other academic or commercial initiatives, providing a solid foundation free of legal restrictions for the advancement of studies in violence detection and behavioral analysis through video technologies of computer vision.

Table 1 shows details regarding the original public datasets used in this work, including "size" or number of the images or frames in the dataset, "resolution", and the number of samples in each class of interest for this work: physical violence and non-physical violence.

In order to give more information about the datasets used, Figure 2 shows some examples of the frames or data contained in AIRTLab, RVLS/Pexels, and SCVD datasets, where the left column of Figure 2 indicates examples of images of physical violence and the right column presents examples of non-physical violence.

Table 1. Video datasets used to build the models presented in this work.

Dataset	Size (Frames)	Resolution (Pixels)	Physical Violence (Frames)	Non-Physical Violence (Frames)
AIRTLab	4800	1920 × 1080	2400	2400
RVLS/Pexels	348	194 × 298/1280 × 850	220	128
SCVD	4000	194 × 298	2000	2000

From the original video dataset AIRTLab, we select a subset of Q images to form the initial dataset. This dataset is then partitioned into two distinct subsets: the training set, denoted as DB; and the test set, denoted as T. The training set is utilized to construct and optimize the model, while the test set is reserved for evaluating the performance of the proposed approach. Both subsets contain 70% (training) and 30% (testing) of the total data, ensuring a balanced distribution of samples for training and evaluation purposes. It is important to note that the intersection between the training set (DB) and the test set (T) is null, ensuring that no data overlap occurs between these two subsets ($DB \cap \mathrm{T} = \emptyset$).

(**a**) Physical violence scene AIRTLab (**b**) Non-physical violence scene AIRTLab

(**c**) Physical violence scene RVLS (**d**) Non-physical violence scene Pexels

(**e**) Physical violence scene SCVD (**f**) Non-physical violence scene SCVD

Figure 2. Some examples of images extracted from the datasets used, showing physical violence (**left column**) and non-physical violence (**right column**) scenes.

During the initial stage (refer to Section 4.1), the training set (*DB*) derived from the AIRTLab dataset is exclusively employed. Subsequently, in stage 2 (detailed in Section 4.2), datasets originating from RVLS, Pexels, and SCVD are utilized as the test set (*T*). Unlike the training set, these datasets are not split further and are used in their entirety to evaluate the performance of the developed model. This approach ensures that the model is assessed rigorously across diverse environments, facilitating a robust assessment of its effectiveness and generalization capabilities.

5.2. Neural Network Parameters

The experiments were conducted on a workstation equipped with an Intel Xeon E-2186G (12) @ 4.700 GHz processor and an NVIDIA GeForce RTX 3060 graphics card featuring 6 GB of video memory. The system had 64 GB of RAM memory. The NVIDIA-SMI 535.86.05 driver version and CUDA Version 12.2 were utilized for GPU acceleration. For software, TensorFlow 2.14.0, scikit-learn 1.3.0, Python 3.10.11, pandas 2.1.0, numpy 1.23.5, jupyterlab 3.6.3, notebook 6.5.4, ipykernel 6.25.2, and ipython 8.15.0 were employed.

In this work, we choose DenseNet121, EfficientNetB0, and MobileNetV2 pre-trained models because they are mature, effective, and robust, but mainly because they are well-known models for image classification (see Section 3.1). Also, their computational efficiency has been tested in terms of floating point operations (FLOPS); for example, see Ref. [61]. Thus, this allows us to focus only on evaluating the performance of our active learning approach (see Section 4) to build robust models for physical violence detection in images

obtained from video in diverse environments. In addition, these models are still the focus of research about image classification tasks.

The setup details of the pre-trained models are summarized in Table 2. "Pooling" represents the pooling method applied, while "dense" indicates the configuration of the dense layer, with the first number representing the number of nodes and the second indicating the activation function used. "Dropout" refers to the probability value for dropping out nodes in the hidden layers. "BN" denotes the batch normalization method. "η" signifies the initial learning rate, "epoch" represents the number of iterations executed by the optimizer, and "freezing" denotes the number of layers frozen to preserve and reuse fundamental layers, focusing learning on more specific and new tasks such as the detection of physical violence in images from videos. "μ" represents the threshold computed, as presented in Section 4.3.

Table 2. Setup summary of the pre-training models studied in this work.

Parameters	DenseNet121	EfficientNetB0	MobileNetV2
Pooling	Average 2D	Average 2D	Average 2D
Dense	512/ReLU	512/ReLU	512/ReLU
Dropout	0.5	0.5	0.5
Dense	256/ReLU	256/ReLU	256/ReLU
BN	True	True	True
Dense	2/Softmax	2/Softmax	2/Softmax
η	0.0001	0.0001	0.001
Batch size	64	64	32
Epochs	29	27	11
Freezing	10	10	4
μ	0.3934	0.3934	0.3934

The parameters presented in Table 2 were determined through a trial-and-error process. Initially, 70% of the *DB* was used to train different configurations of parameters, while the remaining 30% was reserved for testing. Different values of the learning rate, batch size, dropout, epochs, and values of threshold μ were explored to identify the most appropriate settings. The performance of each configuration was evaluated based on the area under the receiver operating characteristic (ROC) curve, as described in Section 5.3.

Once the parameters yielding the best results were identified, the model was retrained using the full *DB* with these optimal settings. It is important to note that the test set **T** was not used during the trial-and-error process but was reserved solely for evaluation, as outlined in Section 4.1. In the output layer, a softmax function [40] with two neurons was employed for classification, and the Adam optimization method [62] was chosen for training the model.

5.3. Classifier Performance

The performance of the studied models is evaluated using the area under the curve ROC, commonly referred to as AUC. This metric is widely used in binary classification problems as it measures a model's ability to discriminate between positive and negative classes, regardless of the threshold chosen for classification. Unlike accuracy, AUC considers the trade-off between sensitivity (true positive rate) and specificity (true negative rate), making it less susceptible to imbalanced datasets.

Recall or sensitivity is defined as the proportion of physical violence data correctly identified (true positives) or how effectively a classifier identifies positive labels; and specificity is the proportion of non-physical violence data correctly labeled (true negatives) or the effectiveness of a classifier to identify negative labels. In the context of binary classification, recall (or sensitivity) and specificity are important performance metrics that provide insights into how well a classifier distinguishes between positive and negative classes. These metrics are typically derived from a confusion matrix. The formal definitions of AUC, sensitivity, and specificity can be found in Ref. [63].

6. Results and Discussion

The experimental results are presented in this section. Firstly, we present the evaluation of the first stage of the proposed method, followed by the results of the second stage, and finally, an overall analysis of the method's performance.

Table 3 presents the results after the first stage (see Section 4.1), while Tables 4 and 5 present the second stage of the proposed method. The rows in Tables 3–5 correspond to the pre-trained model employed: DenseNet121, EfficientNetB0, or MobileNetV2 (see Section 3.1). The columns are "epoch", the number of epochs performed in the proposed approach (see Algorithm 3), and the metrics used to evaluate the built models "recall", "specificity", and "AUC" (see Section 5.3). Additionally, the tables include the number of physical violence ("PV") and non-physical violence ("NPV") scenes considered by the classifier as ambiguous or hard to classify in a respective epoch (see Section 4.3). Finally, the "size" column indicates the number of images in the training dataset ($\|DB\|$), including those added to DB, because they were considered ambiguous.

Figures 3–5 show representative scenes where the proposed approach has uncertainty about the correct label in two contexts: uncertainty when the model labeled the image as an image with physical violence (Figures 3a, 4a and 5a), and uncertainty when the model labeled the image as an image with non-physical violence (Figures 3b, 4b and 5b). These figures are intended to understand what scenes are confusing or ambiguous for the model and could be misclassified.

Table 3 presents the results of the initial model built with the AIRTLab dataset (dataset base, DB) in epoch 0, followed by subsequent epochs (epochs 1, 2, ..., 5) where the model is retrained using ambiguous images to enhance its robustness. Retraining the model with ambiguous scenes generally improves classifier performance compared to the initial model or epoch 0 (Figure 3, shows some samples of images identified as ambiguous by the neural network model). Additionally, it is observed that the number of ambiguous samples decreases as the number of epochs increases. Moreover, it is noteworthy that, overall, specificity values are lower than the recall values. The latter indicates that the classifier correctly identifies more images as physical violence (PV) than non-physical violence (NPV). However, it is notable that a more significant number of ambiguous samples are selected by the model in the class PV than in the class NPV.

Also, note in Table 3 in the column "size" that increasing the number of epochs implies an increase in the size of the dataset base (DB). This is because the ambiguous samples are added to the training dataset in each iteration to robust the model performance, as explained in Section 4. In Figure 3a,b, examples of images (from the AIRTLab dataset) selected by the classifier as ambiguous or hard to classify are shown to clarify the ambiguous concept. The left and right columns display physical violence and non-physical violence scenes, respectively; it is very unclear whether physical violence exists in these images, which means the proposed threshold works properly.

The results of the second stage are presented in Tables 4 and 5. In this stage, the last model obtained in epoch 5 of the first stage is tested with an unknown dataset, simulating an online system akin to a camera surveillance system. It is worth noting that while the primary objective of this work is to demonstrate the effectiveness of the proposed approach (as outlined in Section 4), it does not aim to implement a camera surveillance system for real-time operation. Instead, the focus is on evaluating the performance of the proposed method under simulated real-world conditions.

Table 4 presents the experimental results obtained after testing the model with the RVLS/Pexels dataset, which contains images markedly different from those used in the first stage (see Figure 2) or those employed to build the model. The objective is to evaluate the model's performance in scenarios with diverse characteristics. The RVLS/Pexels dataset, as described in Section 5.1, comprises confusing images in class PV and clearly non-physical violence images in class NPV (see Figure 2c,d). Despite being a small dataset, the model trained in epoch 5 of the first stage demonstrates robustness in classifying this dataset. The initial performance of the model in epoch 0 varies across different architectures. For

instance, DenseNet121 achieves an *AUC* value of 0.892, while EfficientNetB0's performance is notably lower, with an *AUC* of 0.489. However, the classifier's performance is significantly improved upon retraining the model with ambiguous images (epochs 1 through 5). For instance, the performance of the EfficientNetB0 model in epoch 5 reaches an *AUC* of 0.982, suggesting the effectiveness of the proposed approach in enhancing classification accuracy. Consider that the model was retrained using approximately less than 21% of the RVLS/Pexels dataset, and the performance was notably increased. DenseNet121 and MobileNetV2 are more robust models in dealing with unknown environments, and EfficientNetB0's performance was also improved.

Table 3. Experimental result values on recall, specificity, and AUC metrics corresponding to the AIRTLab dataset using our proposed active learning approach. These results were obtained in the first stage (Section 4.1).

Model	Epoch	Recall	Specificity	AUC	PV	NPV	Size
DenseNet121	0	1.000	0.658	0.829	471	3	2400
	1	0.988	0.846	0.917	154	54	2874
	2	0.993	0.793	0.893	282	20	3082
	3	0.973	0.907	0.940	139	33	3384
	4	1.000	0.738	0.869	280	0	3556
	5	1.000	0.813	0.907	237	11	3836
EfficientNetB0	0	0.963	0.919	0.941	140	11	2400
	1	0.969	0.937	0.953	52	19	2551
	2	0.993	0.793	0.893	13	20	2622
	3	0.990	0.947	0.968	22	0	2655
	4	1.000	0.738	0.869	4	5	2677
	5	1.000	0.813	0.907	5	4	2686
MobileNetV2	0	1.000	0.926	0.963	111	2	2400
	1	1.000	0.952	0.976	39	7	2513
	2	1.000	0.939	0.969	66	13	2559
	3	1.000	0.930	0.965	50	0	2638
	4	1.000	0.939	0.969	35	0	2688
	5	0.994	0.884	0.939	78	0	2723

(**a**) (**b**)

Figure 3. Some examples of images selected by the classifier as ambiguous or hard to classify in AIRTLab dataset. (**a**) and (**b**) display physical violence and non-physical violence scenes, respectively.

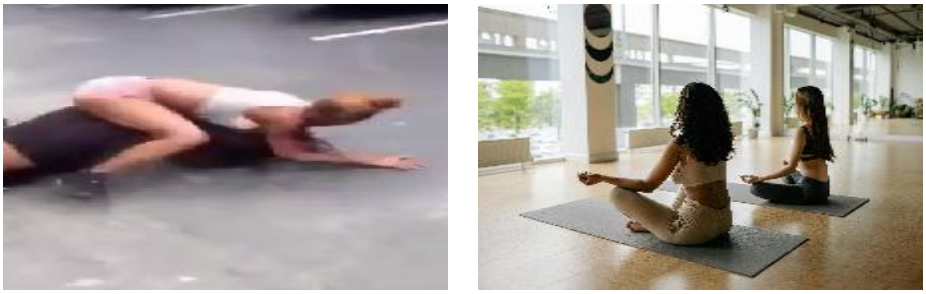
In terms of recall and specificity, Table 4 shows a similar trend to the first stage, where the models perform better in classifying scenes as physical violence than non-physical violence, but it is not clearly defined. Regarding ambiguous images (see Figure 4), it is possible to notice that the maximum number of images considered as ambiguous was 76 in this dataset (with MobileNetV2), i.e., about 21% of samples from the RVLS/Pexels dataset (see Table 1). In other words, the proposed approach uses fewer samples than traditional

methods that mix different datasets to obtain the best performance, for example, the studies in Refs. [2,16,25].

Table 4. Experimental result values on recall, specificity, and AUC metrics corresponding to the RVLS/Pexels dataset using our proposed active learning approach. These results were obtained in the second stage (Section 4.2).

Model	Epoch	Recall	Specificity	AUC	PV	NPV	Size
DenseNet121	0	0.862	0.922	0.892	16	12	3864
	1	0.950	0.921	0.936	19	8	3891
	2	0.972	0.931	0.952	10	5	3906
	3	0.986	0.933	0.959	11	3	3920
	4	0.986	0.977	0.981	5	3	3928
	5	0.968	0.953	0.961	15	3	3946
EfficientNetB0	0	0.624	0.354	0.489	47	19	2752
	1	0.896	0.881	0.889	6	15	2773
	2	0.981	0.939	0.960	8	1	2782
	3	0.965	0.984	0.974	2	4	2788
	4	0.978	0.984	0.981	5	3	2796
	5	0.995	0.969	0.982	2	1	2799
MobileNetV2	0	0.771	0.767	0.769	40	22	2785
	1	0.886	0.980	0.933	4	14	2803
	2	0.991	0.962	0.976	12	5	2820
	3	0.978	1.000	0.989	0	4	2824
	4	0.991	0.940	0.965	20	3	2847
	5	0.978	1.000	0.989	0	9	2856

In contrast, Table 4 does not exhibit a trend in terms of what class is more confusing for the classifier, as was observed in Table 3. Images selected by the classifier using the threshold μ could be clearly defined as physical violence or non-physical violence by a human expert (see Figures 2c and 4b), but for the model this is not true. However, adding these ambiguous images increases the classifier's performance; in other words, using the threshold μ helps the classifier to improve its effectiveness.



(**a**) Physical violence scene in RVLS (**b**) Non-physical violence scene in Pexels

Figure 4. Some examples of images selected by the classifier as ambiguous or hard to classify. (**a**) and (**b**) display physical violence and non-physical violence scenes, respectively.

Table 5 exhibits the results after testing the last model retrained with the AIRTLab dataset and ambiguous samples from the RVLS/Pexels dataset on a new and unknown dataset (in this case, the SCVD dataset), which corresponds to the models developed in epoch 5 of the preceding phase. The classifier performance is relatively low, with the DenseNet121 model achieving the highest AUC value of 0.444. However, the SCVD dataset differs significantly from the AIRTLab and RVLS/Pexels datasets, as it contains more

diverse scenarios with more people in the frame and less defined scenes (see Figure 2). Despite these challenges, the process of retraining the classifier leads to an improvement in effectiveness, with AUC values ranging from 0.811 to 0.914.

The number of ambiguous scenes selected by the model is higher compared to the other datasets (see Tables 3 and 4), indicating that these images are more difficult for the classifier to learn compared to those used previously (see Figure 2). Furthermore, the images selected as ambiguous in this phase are confusing for both the classifier and the human expert. Some examples of such ambiguous scenarios are depicted in Figure 5a,b. There is a tendency in recall and specificity to classify non-physical violence images better than physical violence ones, even though the distinction is unclear.

(**a**) (**b**)

Figure 5. Two examples of images selected by the classifier as ambiguous or hard to classify from the SCVD dataset. (**a**) and (**b**) display physical violence and non-physical violence scenes, respectively.

Table 5. Experimental results values on recall, specificity, and AUC metrics corresponding to the SCVD dataset using our proposed active learning approach. These results were obtained in the second stage (Section 4.2).

Model	Epoch	Recall	Specificity	AUC	PV	NPV	Size
DenseNet121	0	0.417	0.470	0.444	276	171	4393
	1	0.781	0.671	0.726	333	261	4987
	2	0.805	0.767	0.786	484	342	5813
	3	0.717	0.917	0.817	148	440	6401
	4	0.914	0.701	0.807	568	89	7058
	5	0.689	0.933	0.811	86	363	7507
EfficientNetB0	0	0.585	0.565	0.575	298	347	3444
	1	0.839	0.876	0.857	157	117	3718
	2	0.907	0.917	0.912	162	105	3985
	3	0.876	0.962	0.919	74	216	4275
	4	0.951	0.871	0.911	293	45	4613
	5	0.846	0.983	0.914	34	251	4898
MobileNetV2	0	0.564	0.593	0.579	356	261	3473
	1	0.750	0.850	0.800	278	206	3957
	2	0.805	0.930	0.867	221	231	4409
	3	0.818	0.906	0.862	205	328	4942
	4	0.924	0.953	0.938	122	100	5164
	5	0.932	0.874	0.903	338	181	5683

The performance of the neural network models studied in this work could still be improved despite including a threshold active learning approach. Tables 3–5 show that there still exists a margin for improvement because all AUC values are lower than 1. In other words, the model misclassifies some images, in which it has no doubts about their

actual classes (it does not consider them as ambiguous), but it is wrong. Figure 6 presents some images misclassified by the neural network model, which notices the challenge of identifying physical violence in images obtained from video across several environments. In addition, there is not a clear trend in terms of accuracy or recall in Tables 3–5 because sometimes these values decrease or increase after a new training or in the next iteration. This behavior is related to the new samples included in each new iteration in the training dataset. Thus, the learning process has different conditions in each iteration, which is reflected in the experimental results of specificity and recall.

Finally, Table 6 compares the performance of the initial and final models. The initial model for the AIRTLab column was built solely using the AIRTLab dataset and ambiguous samples from AIRTLab; the initial model for RVLS/Pexels was constructed with the AIRTLab dataset and ambiguous samples from the RVLS/Pexels dataset; and the initial model for SCVD was built with the AIRTLab dataset and ambiguous samples from the RVLS/Pexels and SCVD datasets. The final model is trained with AIRTLab along with ambiguous images from RVLS/Pexels and SCVD, representing the last model of the second stage. Observe that the initial and final models are the same in the SCVD dataset (last column in Table 6).

Each model (initial and final) was tested with the original datasets: AIRTLab, RVLS/Pexels, and SCVD. Table 6 shows that the final model is highly robust for the AIRTLab test dataset; also, for most models, the AUC values increase or only a minimal decrease is observed. However, when the models are tested with the RVLS/Pexels dataset, the performance is affected, with AUC values reduced to 0.836 (for the MobileNetV2 model), while the other models are less affected. This reduction in performance indicates that the model's ability to classify new images from different environments is diminished, but it still retains acceptable performance on previously learned tasks.

(**a**) Non-physical violence scene in AIRTLab (**b**) Non-physical violence scene in Pexels

(**c**) Physical violence scene in RVLS (**d**) Physical violence scene in SCVD

Figure 6. Examples of images misclassified by the model; it does not identify them as ambiguous or hard to classify but assigns them a wrong label. The first row displays non-physical violence and the second row physical violence scenes.

Table 6. Comparison of the initial and final models' performance (AUC metric) with respect to the original dataset test of AIRTLab, RVLS/Pexels, and SCVD.

		AUC		
	Model	**AIRTLab**	**RVLS/Pexels**	**SCVD**
DenseNet121	Initial model	0.907	0.961	0.811
	Final model	0.895	0.881	0.811
EfficientNetB0	Initial model	0.907	0.982	0.914
	Final model	0.964	0.844	0.914
MobileNetV2	Initial model	0.939	0.989	0.903
	Final model	0.963	0.836	0.903

The experimental results on the RVLS/Pexels dataset significantly differ from the other datasets because it represents a sudden environmental change. The reasonable response of the model to that dataset indicates the proposed approach's ability to adapt to changing scenarios. Still, it is limited when the scenarios are very different (see Figure 2). However, the advantage of the proposed model is that it continuously learns from the environment, and the classifier will adapt in future iterations (see Algorithm 1) to improve its performance.

The performance on the SCVD dataset remains consistent with that obtained by the final model, as in this phase the initial and final models are equivalent. Another advantage of our proposed approach is that it is only retrained with samples that are hard to learn for the classifier, and those samples are confirmed by a human expert.

The experimental results presented in this section show that the proposed approach's performance is appropriate for working in changing environments. The key is the classifier's ability to retain the knowledge learned while it learns new knowledge verified by a human expert. In other words, the main idea is to maintain an acceptable classification performance on the previous images learned while new images are learned from unknown environments.

The additional effort regarding human expert hours associated with the proposed threshold active learning (see Section 4) is justified. The model cannot remain static because the nature of the environment is changeable: new videos in different scenarios are generated daily. Consequently, the classifier should be rebuilt to adapt to new scenarios (which is a challenge for static classifiers). For this, new high-quality datasets are needed; thus, the contribution of human experts in this process is of great importance because, unlike static classifiers, the proposed approach only uses those samples where the classifier has uncertainty about their correct label.

State of the Art: The Best-Performing Techniques

The state-of-the-art best-performing techniques for violence detection in surveillance videos include two main approaches: sequence and frame levels. Oriented Violent Flows (OViF), SVM classifiers, and spatiotemporal autocorrelation of gradients (STACOG) for feature extraction are some relevant representatives of the first approach. Deep learning methods such as CNN, LSTM, and 3D ConvNets have shown promising results in violence detection at a frame-level analysis [64].

In Table 7, we present the results of similar works that employed various models at the frame level to detect violence in videos. These works consistently report good performance. However, our research emphasizes the use of active learning, which, while not yielding substantially superior results compared to other authors, does enhance the performance compared to those without active learning. Specifically, our approach with active learning improves the pre-trained models' performance in nearly all cases (see Table 6). Furthermore, it is important to highlight that no modifications or adaptations to the model architecture were necessary. This indicates the potential of active learning to optimize model efficiency and performance, even within established frameworks, and distinguishes our approach since we have not identified references where active learning is part of the method.

Table 7. Comparison to related research; all Refs. worked at the frame level of the video.

Ref	Model	Metric	Value
[11]	3D CNN (C3D) + SVM (support vector machine) C3D + FC (fully connected network as classifier) ConvLSTM (convolutional LSTM)	Accuracy AUC	97.86–99.6% 0.996–1.0 96.67–99.05% 0.9927–0.9994 84.19–96.57% 0.9443–0.9931
[12]	ResNet50 + 7 layer FC VGG16 + 7 layer FC	Accuracy	94–100% 95.5–100%
[14]	4D Convolution	Accuracy F1-score	94.67–100% 0.94–1.00
[16]	DenseNet121 + LSTM (long short-term memory) VGG16 + LSTM Oriented FAST and Rotated BRIEF (ORB) + VGG16 + LSTM ORB + DenseNet121 + LSTM	Accuracy	92.05–96.40% 79.4–93.3% 92% 94.49%
[65]	C3D + SVM	Accuracy AUC	95.51–99.29% 0.9832–0.99
[66]	MobileNetV2 + LSTM	Accuracy	82.0–99.5%
This work (final model only)	DenseNet121 + Active Learning EfficientNetB0 + Active Learning MobileNetV2 + Active Learning	AUC	0.811–0.881 0.844–0.964 0.836–0.963

The purpose of conducting research using different techniques and models for violence detection in videos is to improve surveillance systems' accuracy, efficiency, and reliability in identifying and preventing physical violence incidents. By exploring various approaches, researchers aim to develop robust algorithms that can effectively detect physical violence behavior in different scenarios, such as crowd violence, school violence, and aggressive behaviors. Additionally, comparing and evaluating various methods helps identify the strengths and limitations of each approach, leading to the advancement of technology for enhancing security and safety through automated violence detection in surveillance videos. The goal is to create more effective and reliable systems for detecting and responding to physical violence activities, thereby contributing to the overall security and well-being of individuals and communities.

Some challenges in accurately detecting violence in surveillance videos include the complexity of identifying features that describe behavior accurately, the impact of scene changes and appearance variations on object behavior analysis, the lack of publicly available datasets for violence detection, data imbalance between positive and negative samples, the computational and time-consuming costs of feature representation during video violence detection, among others [64,66]. This work analyzes active learning as a tool that could be applied to different models by combining human expert input and an automatic threshold parameter, which also is able to manage diverse scene variations due to the re-training process with human-labeled images during the active learning stage; this was explored by using five well-known and well performing models. Additionally, to our knowledge, no other works have been identified in the field of violence detection in videos that include active learning as a strategy to improve the training process.

We discussed the main experimental results obtained by the proposed approach using pre-trained models. Still, clarifying that the presented method could be applied in other neural network models is necessary. We are interested in testing its performance in other advanced deep learning models like transformers in the future. The main goal of this work is only to highlight the effectiveness of the threshold active learning approach to build robust models for violence detection in images obtained from video in diverse

environments, regardless of which pre-trained model is used or which is the one that attains the best classification performance.

7. Conclusions

This work analyzes the effectiveness of using a threshold active learning approach to develop robust neural network models capable of classifying physical violence or non-physical violence scenes in video frames from unknown environments. The main contribution of this approach is the process used to obtain the threshold μ (process based on the neural network output) that allows human experts to contribute to the classification process to obtain more robust neural networks and high-quality datasets.

The proposed approach consists of two stages: In the first stage, the neural network model is trained with an initial dataset and tested with unseen images from the same dataset (training and testing subsets). The selected pre-trained models usually perform well, as presented in the experimental results section. In the second stage, the model is exposed to unknown environments (images from other datasets); in this stage, the model reduces its classification performance but still retains an acceptable performance on previously learned tasks and the new unlearned tasks. However, the classifier performance improves if the model is retrained with images selected as ambiguous (using the threshold μ). This implies that the proposed approach continuously learns from the environment, i.e., it constantly adapts to new scenarios and improves its classification performance in unknown environments.

Moving forward, there are several opportunity areas for further exploration and improvement. One area of interest is the identification of the onset of physical violence, which would enable the model to issue more precise alerts. The model could provide timely interventions or alerts by pinpointing the exact moment when physical violence begins within a video sequence. This would increase its practical utility in real-world scenarios.

Author Contributions: Conceptualization, I.M.A.; methodology, I.M.A. and R.A.; software, F.P.P. and J.A.A.V.; validation, E.E.G.-G. and R.A.; formal analysis, R.A. and E.E.G.-G.; investigation, I.M.A. and R.A.; data curation, F.P.P. and J.A.A.V.; writing—original draft preparation, I.M.A. and R.A.; writing—review and editing, O.P.-R. and E.E.G.-G.; supervision, O.P.-R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: We express our gratitude to Tecnológico Nacional de México/Instituto Tecnológico de Toluca for supporting Itzel María Abundez Barrera by releasing her from her teaching activities to develop this research.

Conflicts of Interest: The authors declare they have no conflicts of interest.

References

1. Ye, L.; Yan, S.; Zhen, J.; Han, T.; Ferdinando, H.; Seppänen, T.; Alasaarela, E. Physical Violence Detection Based on Distributed Surveillance Cameras. *Mob. Netw. Appl.* **2022**, *27*, 1688–1699. [CrossRef]
2. Ramzan, M.; Abid, A.; Khan, H.U.; Awan, S.M.; Ismail, A.; Ahmed, M.; Ilyas, M.; Mahmood, A. A Review on State-of-the-Art Violence Detection Techniques. *IEEE Access* **2019**, *7*, 107560–107575. [CrossRef]
3. Ojala, T.; Pietikäinen, M.; Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit.* **1996**, *29*, 51–59. [CrossRef]
4. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893. [CrossRef]
5. Rosten, E.; Drummond, T. Machine Learning for High-Speed Corner Detection. In *Computer Vision—ECCV 2006*; Leonardis, A., Bischof, H., Pinz, A., Eds.; Series Title: Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3951, pp. 430–443. [CrossRef]

6. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
7. Zhou, H.; Yuan, Y.; Shi, C. Object tracking using SIFT features and mean shift. *Comput. Vis. Image Underst.* **2009**, *113*, 345–352. [CrossRef]
8. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary Robust invariant scalable keypoints. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2548–2555. [CrossRef]
9. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571. [CrossRef]
10. Wang, D.; Zhang, Z.; Wang, W.; Wang, L.; Tan, T. Baseline Results for Violence Detection in Still Images. In Proceedings of the 2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance, Beijing, China, 18–21 September 2012; pp. 54–57. [CrossRef]
11. Sernani, P.; Falcionelli, N.; Tomassini, S.; Contardo, P.; Dragoni, A.F. Deep learning for automatic violence detection: Tests on the AIRTLab dataset. *IEEE Access* **2021**, *9*, 160580–160595. [CrossRef]
12. Honarjoo, N.; Abdari, A.; Mansouri, A. Violence detection using pre-trained models. In Proceedings of the 2021 5th International Conference on Pattern Recognition and Image Analysis (IPRIA), Kashan, Iran, 28–29 April 2021; pp. 1–4. [CrossRef]
13. Ciampi, L.; Foszner, P.; Messina, N.; Staniszewski, M.; Gennaro, C.; Falchi, F.; Serao, G.; Cogiel, M.; Golba, D.; Szczęsna, A.; et al. Bus violence: An open benchmark for video violence detection on public transport. *Sensors* **2022**, *22*, 8345. [CrossRef] [PubMed]
14. Magdy, M.; Fakhr, M.W.; Maghraby, F.A. Violence 4D: Violence detection in surveillance using 4D convolutional neural networks. *IET Comput. Vis.* **2023**. [CrossRef]
15. Khan, S.U.; Haq, I.U.; Rho, S.; Baik, S.W.; Lee, M.Y. Cover the violence: A novel Deep-Learning-Based approach towards violence-detection in movies. *Appl. Sci.* **2019**, *9*, 4963. [CrossRef]
16. Elkhashab, Y.R.; El-Behaidy, W.H. Violence Detection Enhancement in Video Sequences Based on Pre-trained Deep Models. *FCI-H Inform. Bull.* **2023**, *5*, 23–28. [CrossRef]
17. Vrskova, R.; Hudec, R.; Kamencay, P.; Sykora, P. A New Approach for Abnormal Human Activities Recognition Based on ConvLSTM Architecture. *Sensors* **2022**, *22*, 2946. [CrossRef] [PubMed]
18. Abbass, M.A.B.; Kang, H.S. Violence Detection Enhancement by Involving Convolutional Block Attention Modules into Various Deep Learning Architectures: Comprehensive Case Study for UBI-Fights Dataset. *IEEE Access* **2023**, *1*, 37096–37107. [CrossRef]
19. Vieira, J.C.; Sartori, A.; Stefenon, S.F.; Perez, F.L.; de Jesus, G.S.; Leithardt, V.R.Q. Low-Cost CNN for Automatic Violence Recognition on Embedded System. *IEEE Access* **2022**, *10*, 25190–25202. [CrossRef]
20. Rendón-Segador, F.J.; Álvarez-García, J.A.; Enríquez, F.; Deniz, O. Violencenet: Dense multi-head self-attention with bidirectional convolutional lstm for detecting violence. *Electronics* **2021**, *10*, 1601. [CrossRef]
21. Wang, X.; Yang, J.; Kasabov, N.K. Integrating spatial and temporal information for violent activity detection from video using deep spiking neural networks. *Sensors* **2023**, *23*, 4532. [CrossRef] [PubMed]
22. Sultani, W.; Chen, C.; Shah, M. Real-world anomaly detection in surveillance videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6479–6488.
23. Vosta, S.; Yow, K.C. A cnn-rnn combined structure for real-world violence detection in surveillance cameras. *Appl. Sci.* **2022**, *12*, 1021. [CrossRef]
24. Yousaf, K.; Nawaz, T. A deep learning-based approach for inappropriate content detection and classification of youtube videos. *IEEE Access* **2022**, *10*, 16283–16298. [CrossRef]
25. Ghadekar, P.; Agrawal, K.; Bhosale, A.; Gadi, T.; Deore, D.; Qazi, R. A Hybrid CRNN Model for Multi-Class Violence Detection in Text and Video. In *ITM Web of Conferences*; EDP Sciences: Les Ulis, France, 2023; Volume 53.
26. Zhou, L.; Pan, S.; Wang, J.; Vasilakos, A.V. Machine learning on big data: Opportunities and challenges. *Neurocomputing* **2017**, *237*, 350–361. [CrossRef]
27. IBM. ¿Qué es el Etiquetado de Datos? Available online: https://www.ibm.com/es-es/topics/data-labeling (accessed on 17 July 2023).
28. Bengar, J.; van de Weijer, J.; Twardowski, B.; Raducanu, B. Reducing Label Effort: Self-Supervised meets Active Learning. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 1631–1639. [CrossRef]
29. Goswami, M.; Sanil, V.; Choudhry, A.; Srinivasan, A.; Udompanyawit, C.; Dubrawski, A. AQuA: A Benchmarking Tool for Label Quality Assessment. In *Advances in Neural Information Processing Systems*; Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S., Eds.; Curran Associates, Inc.: New York, NY, USA, 2023; Volume 36, pp. 79792–79807.
30. Chen, S.; Yang, Y.; Hua, Y. Semi-Supervised Active Learning for Object Detection. *Electronics* **2023**, *12*, 375. [CrossRef]
31. Li, X.; Wang, X.; Chen, X.; Lu, Y.; Fu, H.; Wu, Y.C. Unlabeled data selection for active learning in image classification. *Sci. Rep.* **2024**, *14*, 424. [CrossRef]
32. Mohammadi, H.; Nazerfard, E.; Firoozi, T. Reinforcement Learning-based Mixture of Vision Transformers for Video Violence Recognition. *arXiv* **2023**, arXiv:2310.03108.
33. Mosqueira-Rey, E.; Hernández-Pereira, E.; Alonso-Ríos, D.; Bobes-Bascarán, J.; Fernández-Leal, Á. Human-in-the-loop machine learning: A state of the art. *Artif. Intell. Rev.* **2023**, *56*, 3005–3054. [CrossRef]

34. Li, X.; Guo, Y. Adaptive active learning for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 859–866.
35. Beluch, W.H.; Genewein, T.; Nurnberger, A.; Kohler, J.M. The Power of Ensembles for Active Learning in Image Classification. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9368–9377. [CrossRef]
36. Sener, O.; Savarese, S. Active Learning for Convolutional Neural Networks: A Core-Set Approach. *arXiv* **2018**, arXiv:1708.00489.
37. Phan, J.; Ruocco, M.; Scibilia, F. Dual Active Sampling on Batch-Incremental Active Learning. In *Nordic Artificial Intelligence Research and Development*; Bach, K., Ruocco, M., Eds.; Springer: Cham, Switzerland, 2019; pp. 127–132.
38. Carbonneau, M.A.; Granger, E.; Gagnon, G. Bag-Level Aggregation for Multiple-Instance Active Learning in Instance Classification Problems. *IEEE Trans. Neural Networks Learn. Syst.* **2019**, *30*, 1441–1451. [CrossRef] [PubMed]
39. Haykin, S. *Neural Networks: A Comprehensive Foundation*; Prentice Hall PTR: Hoboken, NJ, USA, 1998.
40. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
41. Fawzi, A.; Moosavi-Dezfooli, S.; Frossard, P.; Soatto, S. Classification regions of deep neural networks. *arXiv* **2017**. arxiv:1705.09552.
42. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: An overview and application in radiology. *Insights Imaging* **2018**, *9*, 611–629. [CrossRef]
43. Bengfort, B.; Bilbro, R.; Ojeda, T. *Applied Text Analysis with Python*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2018.
44. Zhao, R.; Yan, R.; Wang, J.; Mao, K. Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks. *Sensors* **2017**, *17*, 273. [CrossRef]
45. Wu, J.L.; He, Y.; Yu, L.C.; Lai, K.R. Identifying Emotion Labels From Psychiatric Social Texts Using a Bi-Directional LSTM-CNN Model. *IEEE Access* **2020**, *8*, 66638–66646. [CrossRef]
46. Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 24–49. [CrossRef]
47. Yang, Y.; Zhang, L.; Du, M.; Bo, J.; Liu, H.; Ren, L.; Li, X.; Deen, M.J. A comparative analysis of eleven neural networks architectures for small datasets of lung images of COVID-19 patients toward improved clinical decisions. *Comput. Biol. Med.* **2021**, *139*, 104887. [CrossRef]
48. Ganesh, K.; Rao, B.P. Classification of Breast Cancer from Mammogram Images using DenseNET. *J. Biomed. Eng.* **2023**, *40*, 192–199.
49. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
50. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
51. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
52. CNDH, Ley general de acceso de las mujeres a una vida libre de violencia. Diario Oficial de la Federación, published in DOF 26-01-2024. Available online: https://www.diputados.gob.mx/LeyesBiblio/pdf/LGAMVLV.pdf (accessed on 1 July 2024)
53. Alejo, R.; Monroy-de Jesús, J.; Pacheco-Sánchez, J.H.; López-González, E.; Antonio-Velázquez, J.A. A selective dynamic sampling back-propagation approach for handling the two-class imbalance problem. *Appl. Sci.* **2016**, *6*, 200. [CrossRef]
54. Alejo, R.; Monroy-de Jesús, J.; Ambriz-Polo, J.C.; Pacheco-Sánchez, J.H. An improved dynamic sampling back-propagation algorithm based on mean square error to face the multi-class imbalance problem. *Neural Comput. Appl.* **2017**, *28*, 2843–2857. [CrossRef]
55. Shah, B.; Bhavsar, H. Time Complexity in Deep Learning Models. *Procedia Comput. Sci.* **2022**, *215*, 202–210. [CrossRef]
56. Khan, A.; Sohail, A.; Zahoora, U.; Qureshi, A.S. A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **2020**, *53*, 5455–5516. [CrossRef]
57. Bianculli, M.; Falcionelli, N.; Sernani, P.; Tomassini, S.; Contardo, P.; Lombardi, M.; Dragoni, A.F. A dataset for automatic violence detection in videos. *Data Brief* **2020**, *33*, 106587. [CrossRef]
58. Soliman, M.M.; Kamal, M.H.; Nashed, M.A.E.M.; Mostafa, Y.M.; Chawky, B.S.; Khattab, D. Violence recognition from videos using deep learning techniques. In Proceedings of the 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 8–10 December 2019; pp. 80–85.
59. Pexels. Free Stock Photos, Royalty Free Stock Images & Copyright Free Pictures · Pexels , 2023. Available online: https://www.pexels.com/ (accessed on 1 December 2023).
60. Aremu, T.; Li, Z.; Alameeri, R.; Khan, M.; Saddik, A.E. SSIVD-Net: A Novel Salient Super Image Classification & Detection Technique for Weaponized Violence. In *Intelligent Computing*; Springer Nature: Cham, Switzerland, 2024; pp. 16–35, ISBN 978-3-031-62269-4. [CrossRef]
61. Baumgartl, H.; Buettner, R. Developing efficient transfer learning strategies for robust scene recognition in mobile robotics using pre-trained convolutional neural networks. *arXiv* **2021**, arXiv:2107.11187.
62. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
63. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]

64. Omarov, B.; Narynov, S.; Zhumanov, Z.; Gumar, A.; Khassanova, M. State-of-the-art violence detection techniques in video surveillance security systems: A systematic review. *PeerJ Comput. Sci.* **2022**, *8*, e920. [CrossRef]
65. Accattoli, S.; Sernani, P.; Falcionelli, N.; Mekuria, D.N.; Dragoni, A.F. Violence Detection in Videos by Combining 3D Convolutional Neural Networks and Support Vector Machines. *Appl. Artif. Intell.* **2020**, *34*, 329–344. [CrossRef]
66. Vijeikis, R.; Raudonis, V.; Dervinis, G. Efficient Violence Detection in Surveillance. *Sensors* **2022**, *22*, 2216. [CrossRef] [PubMed]

Capítulo 5

Discusión

En esta investigación se incorporó un mecanismo de aprendizaje activo basado en un umbral dinámico (μ) en un modelo de detección de violencia física utilizando redes neuronales profundas preentrenadas. Este mecanismo selecciona de manera óptima las imágenes ambiguas, es decir, aquellas cuyas predicciones tienen baja confianza según el umbral (μ), para ser etiquetadas por un experto humano. Una vez etiquetadas, se retroalimenta al modelo con estos datos.

Para evaluar el modelo se utilizaron tres conjunto de datos públicos diferentes, esto con el fin de enriquecer la diversidad y realismo del modelo, siendo los conjuntos de datos:

- Conjunto de datos AIRTLab [44]: Este conjunto de datos presenta escenas que representan situaciones cotidianas que pueden ser interpretadas por los algoritmos como acciones violentas, como abrazos, aplausos y saludos, realizados por actores no profesionales.

- Conjunto de datos de Situaciones de Violencia en la Vida Real (RVLS) [45]: Este conjunto de datos incluye imágenes extraídas de vídeos que muestran situaciones de la vida real y proporciona una colección completa de diversos escenarios encontrados en varios entornos.

- Conjunto de datos (Pexels) [46]: Plataforma en línea famosa por su amplia

colección de fotos y vídeos de archivo gratuitos. El conjunto de datos de Pexels se obtuvo seleccionando cuidadosamente imágenes de esta plataforma en el contexto de este estudio. Estas imágenes se eligieron específicamente para representar escenas de la vida real sin violencia física, complementando así el conjunto de datos RVLS, que se centra en imágenes de violencia. Incluir imágenes de (Pexels) en el estudio fue esencial para evaluar la capacidad del modelo de aprendizaje profundo para distinguir entre escenas violentas y no violentas. Al entrenar y probar el modelo con imágenes de ambas categorías, y con ello evaluar su precisión y al clasificar la violencia física en diversos contextos.

La diversidad de imágenes disponibles en (Pexels) permitió seleccionar una muestra representativa de escenas no violentas que reflejen la complejidad del mundo real y con ello garantizar que el modelo aprendiera a identificar la violencia física y a diferenciarla de situaciones cotidianas que podrían malinterpretarse como violentas.

- Conjunto de datos de detección de la violencia por circuito cerrado de televisión en ciudades inteligentes (SCVD): El conjunto de datos SCVD ofrece una compilación única de imágenes de vídeo captadas mediante sistemas de circuito cerrado de televisión (CCTV) desplegados en entornos urbanos [47]. Este conjunto de datos tiene en cuenta las variaciones en los ángulos de visión, la calidad de la imagen y los contextos de grabación inherentes a las grabaciones de CCTV, que pueden afectar significativamente a la eficacia de los algoritmos de detección de la violencia. En particular, el conjunto de datos SCVD introduce una categoría específica dedicada a la identificación de armas, ampliando el alcance de la detección de la violencia para incluir cualquier objeto que pueda emplearse con fines nocivos. Esta perspectiva ampliada mejora la capacidad del clasificador para detectar amenazas en

entornos urbanos y ofrece una perspectiva diferente de los riesgos potenciales. Además, la concesión de licencias del conjunto de datos SCVD bajo la licencia "CC BY-NC-SA 4.0" fomenta la colaboración y el intercambio de conocimientos dentro de la comunidad investigadora, al tiempo que garantiza la atribución adecuada a los autores originales.

La heterogeneidad en la colección de imágenes permitió simular la variabilidad de situaciones a las que podrían enfrentarse los sistemas de inteligencia artificial en aplicaciones reales, desde sistemas de seguridad hasta desarrollos de inteligencia artificial contextos sociales complejos.

Para incluir los conjuntos de datos en esta investigación se aseguro que fueran de dominio público liberadas de cualquier derecho de autor existente, permitiendo su uso, redistribución, modificación y explotación sin restricciones ni la necesidad de solicitar permiso a los autores originales. Esta característica facilita su uso en proyectos de investigación, desarrollo de software y otras iniciativas académicas o comerciales, proporcionando una base sólida libre de restricciones legales para el avance de estudios en detección de violencia y análisis de comportamiento mediante tecnologías de vídeo de visión por computadora.

A continuación se muestran los resultados al aplicar la metodología propuesta en la Figura 3.1.

El modelo inicial fue construido con el conjunto de datos AIRTLab aquellas imágenes que el clasificador considero ambiguas son etiquetadas por un experto e incorporadas a las base de datos para reentrenar el modelo. El reentrenamiento del modelo con escenas ambiguas suele mejorar el rendimiento del clasificador en comparación con el modelo inicial observando que el número de muestras ambiguas disminuye a medida que se incrementa el número de épocas. Los valores de especificidad son inferiores a los de recuerdo (Recall) lo que indica que el clasificador identifica correctamente más imágenes como violencia física (VF) que como

no violencia física (NVF). El tamaño del conjunto de datos (Tamaño) incrementa conforme pasan las épocas, esto se debe a que las muestras ambiguas se añaden al conjunto de entrenamiento en cada iteración para reforzar el rendimiento del modelo, ver tabla 5.1.

Tabla 5.1: Resultados experimentales sobre las métricas de Recall, Especificidad y Exactitud correspondientes al conjunto de datos AIRTLab utilizando el enfoque propuesto aprendizaje activo. Estos resultados se obtuvieron en la primera etapa.

Modelo	Epoca	Recall	Especificidad	Exactidud	VF	NVF	Tamaño
DenseNet121	0	1.000	0.658	0.829	471	3	2400
	1	0.988	0.846	0.917	154	54	2874
	2	0.993	0.793	0.893	282	20	3082
	3	0.973	0.907	0.940	139	33	3384
	4	1.000	0.738	0.869	280	0	3556
	5	1.000	0.813	0.907	237	11	3836
EfficientNetB0	0	0.963	0.919	0.941	140	11	2400
	1	0.969	0.937	0.953	52	19	2551
	2	0.993	0.793	0.893	13	20	2622
	3	0.990	0.947	0.968	22	0	2655
	4	1.000	0.738	0.869	4	5	2677
	5	1.000	0.813	0.907	5	4	2686
MobileNetV2	0	1.000	0.926	0.963	111	2	2400
	1	1.000	0.952	0.976	39	7	2513
	2	1.000	0.939	0.969	66	13	2559
	3	1.000	0.930	0.965	50	0	2638
	4	1.000	0.939	0.969	35	0	2688
	5	0.994	0.884	0.939	78	0	2723

A continuación se evalúa la segunda etapa de la metodología mostrando los resultados en las tablas 5.2 y 5.3. En esta etapa, se toma el último modelo obtenido en la época 5 de la primera etapa tabla 5.1 y se prueba con un conjunto de datos desconocido, simulando un sistema en línea, similar a un sistema de

vigilancia por cámara. Cabe señalar que, aunque el objetivo principal de este trabajo fue demostrar la eficacia del enfoque propuesto, no se pretende implementar un sistema de vigilancia por cámara para funcionamiento en tiempo real. En su lugar, se centro en evaluar el rendimiento del método propuesto en condiciones reales simuladas.

A continuación se procedió a probar el modelo con el conjunto de dato RVLS/-Pexeles, el cual contiene imágenes diferentes a las utilizadas en la etapa de entrenamiento permitiendo evaluar el rendimiento del modelo en escenarios con características diversas. Este conjunto de datos comprende imágenes confusas en la clase VF e imágenes no violencia de la clase NVF. El rendimiento inicial del modelo en la época 0 varía según las distintas arquitecturas. En la tabla 5.2 se aprecia como mejora significativamente el rendimiento del clasificador al volver a entrenar el modelo con imágenes ambiguas ya etiquetadas por un experto humano, lo que muestra una eficacia del enfoque propuesto para mejorar la precisión de la clasificación.

A continuación se agrega un conjunto de datos nuevo y desconocido (en este caso, el conjunto de datos SCVD). La tabla 5.3 muestra los resultados tras probar el último modelo reentrenado con el conjunto de datos AIRTLab y muestras ambiguas del conjunto de datos RVLS/Pexels. El rendimiento de los clasificadores es relativamente bajo, siendo el modelo DenseNet121 el que obtiene una menor exactitud, de 0.444. Sin embargo, el conjunto de datos SCVD difiere significativamente de los conjuntos de datos AIRTLab y RVLS/Pexels, ya que contiene una mayor cantidad de escenarios. A pesar de estos retos, el proceso de reentrenamiento del clasificador conduce a una mejora de la eficacia, con valores de exactitud que oscilan entre 0.811 y 0.914.

El número de escenas ambiguas seleccionadas por el modelo es mayor en comparación con los otros conjuntos de datos, lo que indica que estas imágenes son

más difíciles de aprender para el clasificador en comparación con las utilizadas anteriormente. Además, las imágenes seleccionadas como ambiguas en esta fase son confusas tanto para el clasificador como para el experto humano. Se observa una tendencia en el recuerdo y la especificidad a clasificar mejor las imágenes de violencia no física que las de violencia física, aunque la distinción no esté clara.

Tabla 5.2: Resultados experimentales sobre las métricas de Recall, Especificidad y Exactitud correspondientes al conjunto de datos RVLS/Pexels utilizando el enfoque propuesto aprendizaje activo. Estos resultados se obtuvieron en la segunda etapa .

Modelo	**Epoca**	**Recall**	**Especificidad**	**Exactitud**	**VF**	**NVF**	**Tamaño**
DenseNet121	0	0.862	0.922	0.892	16	12	3836
	1	0.950	0.921	0.936	19	8	3864
	2	0.972	0.931	0.952	10	5	3891
	3	0.986	0.933	0.959	11	3	3906
	4	0.986	0.977	0.981	5	3	3920
	5	0.968	0.953	0.961	15	3	3928
EfficientNetB0	0	0.624	0.354	0.489	47	19	2686
	1	0.896	0.881	0.889	6	15	2752
	2	0.981	0.939	0.960	8	1	2773
	3	0.965	0.984	0.974	2	4	2782
	4	0.978	0.984	0.981	5	3	2788
	5	0.995	0.969	0.982	2	1	2796
MobileNetV2	0	0.771	0.767	0.769	40	22	2723
	1	0.886	0.980	0.933	4	14	2785
	2	0.991	0.962	0.976	12	5	22803
	3	0.978	1.000	0.989	0	4	2820
	4	0.991	0.940	0.965	20	3	2824
	5	0.978	1.000	0.989	0	9	2847

Tabla 5.3: Resultados experimentales sobre las métricas de recuerdo, Especificidad y Exactitud correspondientes al conjunto de datos SCVD utilizando el enfoque propuesto aprendizaje activo. Estos resultados se obtuvieron en la segunda etapa.

Modelo	Epoca	Recall	Especificidad	Exactitud	VF	NVF	Tamaño
DenseNet121	0	0.417	0.470	0.444	276	171	3946
	1	0.781	0.671	0.726	333	261	4393
	2	0.805	0.767	0.786	484	342	4987
	3	0.717	0.917	0.817	148	440	5813
	4	0.914	0.701	0.807	568	89	6401
	5	0.689	0.933	0.811	86	363	7058
EfficientNetB0	0	0.585	0.565	0.575	298	347	2799
	1	0.839	0.876	0.857	157	117	3444
	2	0.907	0.917	0.912	162	105	3718
	3	0.876	0.962	0.919	74	216	3985
	4	0.951	0.871	0.911	293	45	4275
	5	0.846	0.983	0.914	34	251	4613
MobileNetV2	0	0.564	0.593	0.579	356	261	2856
	1	0.750	0.850	0.800	278	206	3473
	2	0.805	0.930	0.867	221	231	3957
	3	0.818	0.906	0.862	205	328	4409
	4	0.924	0.953	0.938	122	100	4942
	5	0.932	0.874	0.903	338	181	5164

Al compara el rendimiento de los modelos inicial y final. Donde el modelo inicial para el conjunto de datos AIRTLab se construyó únicamente con el conjunto de datos AIRTLab y muestras ambiguas de AIRTLab; el modelo inicial para RVLS/Pexels se construyó con el conjunto de datos AIRTLab y muestras ambiguas del conjunto de datos RVLS/Pexels; y el modelo inicial para SCVD se construyó con el conjunto de datos AIRTLab y muestras ambiguas de los conjuntos de datos RVLS/Pexels y SCVD. El modelo final fue entrenado con AIRTLab y con imágenes ambiguas de RVLS/Pexels y SCVD, representando el último mo-

delo de la segunda etapa. Observe que los modelos inicial y final son los mismos en el conjunto de datos SCVD (última columna de la tabla 5.4).

El modelo final es robusto para el conjunto de datos de prueba AIRTLab, lo cual se observa en la tabla 5.4; para la mayoría de los modelos, los valores de exactitud incrementan o sólo sufren una disminución mínima. Sin embargo, cuando los modelos se prueban con el conjunto de datos RVLS/Pexels, el rendimiento se ve afectado, con valores de exactitud reducidos a 0.836 (para el modelo MobileNetV2), mientras que los demás modelos se ven menos afectados. Disminuyendo la capacidad del modelo para clasificar imágenes nuevas procedentes de entornos diferentes, conservando un rendimiento mayor al 0.811 en tareas aprendidas.

Tabla 5.4: Comparación del rendimiento de los modelos inicial y final (métrica de Exactitud) con respecto a la prueba del conjunto de datos original de AIRTLab, RVLS/Pexels y SCVD.

		Exactitud		
	Modelo	**AIRTLab**	**RVLS/Pexels**	**SCVD**
DenseNet121	Modelo Inicial	0.907	0.961	0.811
	Modelo final	0.895	0.881	0.811
EfficientNetB0	Modelo inicial	0.907	0.982	0.914
	Final model	0.964	0.844	0.914
MobileNetV2	Modelo inicial	0.939	0.989	0.903
	Modelo final	0.963	0.836	0.903

En este trabajo, se eligieron los modelos de redes preentrenadas DenseNet121, EfficientNetB0 y MobileNetV2 porque son maduros, eficaces y robustos, pero sobre todo porque son modelos bien conocidos para la clasificación de imágenes. Además, su eficiencia computacional se ha probado en términos de operaciones en coma flotante (FLOPS); por ejemplo, véase la Ref. [48]. Por lo tanto, esto permitió centrarnos únicamente en evaluar el rendimiento de nuestro enfoque de

aprendizaje activo para construir modelos robustos para la detección de violencia física en imágenes. A pesar de que el modelo es robusto una limitante del modelo es que un solo experto etiquetó las imágenes ambiguas.

Capítulo 6

Conclusiones

Este trabajo analiza el aprendizaje activo como una herramienta que podría aplicarse a diferentes modelos mediante la combinación de la aportación de expertos humanos y un parámetro de umbral automático, que también es capaz de gestionar diversas variaciones de la escena debido al proceso de entrenamiento con imágenes etiquetadas por expertos humanos durante la etapa de aprendizaje activo; esto se exploró mediante el uso de tres modelos bien conocidos y de buen rendimiento. Además, hasta donde conocemos, no se han identificado otros trabajos en el campo de la detección de violencia en vídeos que incluyan el aprendizaje activo como estrategia para mejorar el proceso de entrenamiento. No esta de mas recordar que el método presentado podría aplicarse en otros modelos de redes neuronales.

Teniendo como la aportación de esta investigación la propuesta de obtener un umbral μ (proceso basado en la salida de la red neuronal) que permite a los expertos humanos contribuir en el proceso de clasificación para robustecer el modelo e incrementar el conjuntos de datos, quedando confirmada la hipótesis planteada ya que los modelos se adaptaron progresivamente a nuevos escenarios, incrementando la capacidad del modelo para generalizar en la detección de violencia en diferentes entornos.

Con la presente investigación se demuestra que es posible mejorar la precisión,

eficacia y fiabilidad de los sistemas de vigilancia para la identificación de violencia física y con ello prevenir incidentes de violencia física. El desarrollar algoritmos robustos que puedan detectar eficazmente comportamientos de violencia física en diferentes escenarios, como la violencia entre multitudes, la violencia escolar. La comparación y evaluación de diferentes redes preentrenadas ayuda a identificar los puntos fuertes y las limitaciones de cada enfoque, lo que conduce al avance de la tecnología para mejorar la seguridad y la protección mediante la detección automática de la violencia física en los sistemas de vídeo de vigilancia. En un futuro es posible crear sistemas más eficaces y fiables para detectar actividades de violencia física y responder a ellas, contribuyendo así a la seguridad y el bienestar general de las personas y las comunidades.

Existen áreas de oportunidad para seguir explorando y mejorando. Un área de interés es la identificación del inicio de la violencia física, que permitiría al modelo emitir alertas más precisas. El modelo podría proporcionar intervenciones o alertas oportunas señalando el momento exacto en que comienza la violencia física dentro de una secuencia de vídeo. Esto aumentaría su utilidad práctica en situaciones reales.

Referencias

[1] P. Sernani, N. Falcionelli, S. Tomassini, P. Contardo y A. F. Dragoni, «Deep learning for automatic violence detection: Tests on the AIRTLab dataset», *IEEE Access*, vol. 9, págs. 160 580-160 595, 2021.

[2] N. Honarjoo, A. Abdari y A. Mansouri, «Violence detection using pre-trained models», en *2021 5th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, IEEE, 2021, págs. 1-4.

[3] X. Wang, J. Yang y N. K. Kasabov, «Integrating spatial and temporal information for violent activity detection from video using deep spiking neural networks», *Sensors*, vol. 23, n.º 9, pág. 4532, 2023. DOI: 10.3390/s23094532.

[4] L. Ciampi, P. Foszner, N. Messina et al., «Bus violence: An open benchmark for video violence detection on public transport», *Sensors*, vol. 22, n.º 21, pág. 8345, 2022.

[5] M. Magdy, M. W. Fakhr y F. A. Maghraby, «Violence 4D: Violence detection in surveillance using 4D convolutional neural networks», *IET Computer Vision*, 2023.

[6] S. U. Khan, I. U. Haq, S. Rho, S. W. Baik y M. Y. Lee, «Cover the violence: A novel Deep-Learning-Based approach towards violence-detection in movies», *Applied Sciences*, vol. 9, n.º 22, pág. 4963, 2019.

[7] Y. R. Elkhashab y W. H. El-Behaidy, «Violence Detection Enhancement in Video Sequences Based on Pre-trained Deep Models», *FCI-H Informatics Bulletin*, vol. 5, n.º 1, págs. 23-28, 2023, ISSN: 2537–0901. DOI: 10.21608/fcihib.2023.153245.1075.

[8] R. Vrskova, R. Hudec, P. Kamencay y P. Sykora, «A New Approach for Abnormal Human Activities Recognition Based on ConvLSTM Architecture», *Sensors*, vol. 22, n.º 8, pág. 2946, 2022.

[9] M. A. B. Abbass y H.-S. Kang, «Violence Detection Enhancement by Involving Convolutional Block Attention Modules into Various Deep Learning Architectures: Comprehensive Case Study for UBI-Fights Dataset», *IEEE Access*, 2023.

[10] J. C. Vieira, A. Sartori, S. F. Stefenon, F. L. Perez, G. S. de Jesus y V. R. Q. Leithardt, «Low-Cost CNN for Automatic Violence Recognition on Embedded System», *IEEE Access*, vol. 10, págs. 25 190-25 202, 2022. DOI: 10.1109/ACCESS.2022.3155123.

[11] F. J. Rendón-Segador, J. A. Álvarez-García, F. Enríquez y O. Deniz, «Violencenet: Dense multi-head self-attention with bidirectional convolutional lstm for detecting violence», *Electronics*, vol. 10, n.º 13, pág. 1601, 2021.

[12] M. Cheng, K. Cai y M. Li, «RWF-2000: an open large scale video database for violence detection», en *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, págs. 4183-4190.

[13] W. Sultani, C. Chen y M. Shah, «Real-world anomaly detection in surveillance videos», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, págs. 6479-6488.

[14] S. Vosta y K.-C. Yow, «A cnn-rnn combined structure for real-world violence detection in surveillance cameras», *Applied Sciences*, vol. 12, n.º 3, pág. 1021, 2022.

[15] K. Yousaf y T. Nawaz, «A deep learning-based approach for inappropriate content detection and classification of youtube videos», *IEEE Access*, vol. 10, págs. 16 283-16 298, 2022.

[16] M. Ramzan, A. Abid, H. U. Khan et al., «A Review on State-of-the-Art Violence Detection Techniques», *IEEE Access*, vol. 7, págs. 107 560-107 575, 2019. DOI: 10.1109/ACCESS.2019.2932114.

[17] P. Ghadekar, K. Agrawal, A. Bhosale, T. Gadi, D. Deore y R. Qazi, «A Hybrid CRNN Model for Multi-Class Violence Detection in Text and Video», en *ITM Web of Conferences*, EDP Sciences, vol. 53, 2023.

[18] L. Zhou, S. Pan, J. Wang y A. V. Vasilakos, «Machine learning on big data: Opportunities and challenges», *Neurocomputing*, vol. 237, págs. 350-361, 2017, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2017.01.026.

[19] IBM, *¿Qué es el etiquetado de datos?*, https://www.ibm.com/es-es/topics/data-labeling, Accessed: 20-07-2023, 2023.

[20] E Etecé, «Enciclopedia de conceptos», *Obtenido de: https://concepto. de/-conocimiento*, 2021.

[21] G. F. Acevedo et al., «Ley general de acceso de las mujeres a una vida libre de violencia: Importancia, evolución y dificultades», *Entretextos*, vol. 9, n.º 25, págs. 1-11, 2017.

[22] B. Sierra et al., *Aprendizaje automático: conceptos básicos y avanzados. Aspectos prácticos utilizando el software Weka*, 2006.

[23] Y. Cui, P. Koppol, H. Admoni et al., «Understanding the Relationship between Interactions and Outcomes in Human-in-the-Loop Machine Learning», en *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2021, págs. 4382-4391. DOI: 10.24963/ijcai.2021/599.

[24] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán y Á. Fernández-Leal, «Human-in-the-loop machine learning: a state of the art», *Artificial Intelligence Review*, vol. 56, n.º 4, págs. 3005-3054, 2023. DOI: 10.1007/s10462-022-10246-w.

[25] B. Settles, «Active learning literature survey», 2009.

[26] V.-L. Nguyen, M. H. Shaker y E. Hüllermeier, «How to measure uncertainty in uncertainty sampling for active learning», *Machine Learning*, vol. 111, n.º 1, págs. 89-122, 2022. DOI: 10.1007/s10994-021-06003-9.

[27] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[28] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1998.

[29] A. Krizhevsky, I. Sutskever y G. E. Hinton, «Imagenet classification with deep convolutional neural networks», *Advances in neural information processing systems*, vol. 25, 2012.

[30] K. Simonyan y A. Zisserman, «Very deep convolutional networks for large-scale image recognition», *arXiv preprint arXiv:1409.1556*, 2014.

[31] K. He, X. Zhang, S. Ren y J. Sun, «Deep residual learning for image recognition», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, págs. 770-778.

[32] M. Tan y Q. Le, «Efficientnet: Rethinking model scaling for convolutional neural networks», en *International conference on machine learning*, PMLR, 2019, págs. 6105-6114.

[33] G. Huang, Z. Liu, L. Van Der Maaten y K. Q. Weinberger, «Densely connected convolutional networks», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, págs. 4700-4708.

[34] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov y L.-C. Chen, «Mobilenetv2: Inverted residuals and linear bottlenecks», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, págs. 4510-4520.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li y L. Fei-Fei, «Imagenet: A large-scale hierarchical image database», en *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, págs. 248-255.

[36] X. Li e Y. Guo, «Adaptive active learning for image classification», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, págs. 859-866.

[37] W. H. Beluch, T. Genewein, A. Nürnberger y J. M. Köhler, «The power of ensembles for active learning in image classification», en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, págs. 9368-9377.

[38] O. Sener y S. Savarese, «Active learning for convolutional neural networks: A core-set approach», *arXiv preprint arXiv:1708.00489*, 2017.

[39] J. Phan, M. Ruocco y F. Scibilia, «Dual Active Sampling on Batch-Incremental Active Learning», en *Nordic Artificial Intelligence Research and Development: Third Symposium of the Norwegian AI Society, NAIS 2019, Trondheim, Norway, May 27–28, 2019, Proceedings 3*, Springer, 2019, págs. 127-132.

[40] M.-A. Carbonneau, E. Granger y G. Gagnon, «Bag-level aggregation for multiple-instance active learning in instance classification problems», *IEEE transactions on neural networks and learning systems*, vol. 30, n.º 5, págs. 1441-1451, 2018.

[41] S. Chen, Y. Yang e Y. Hua, «Semi-Supervised Active Learning for Object Detection», *Electronics*, vol. 12, n.º 2, pág. 375, 2023.

[42] X. Li, X. Wang, X. Chen, Y. Lu, H. Fu e Y. C. Wu, «Unlabeled data selection for active learning in image classification», *Scientific Reports*, vol. 14, n.º 1, pág. 424, 2024.

[43] H. Mohammadi, E. Nazerfard y T. Firoozi, «Reinforcement Learning-based Mixture of Vision Transformers for Video Violence Recognition», *arXiv preprint arXiv:2310.03108*, 2023.

[44] M. Bianculli, N. Falcionelli, P. Sernani et al., «A dataset for automatic violence detection in videos», *Data in brief*, vol. 33, pág. 106 587, 2020.

[45] M. M. Soliman, M. H. Kamal, M. A. E.-M. Nashed, Y. M. Mostafa, B. S. Chawky y D. Khattab, «Violence recognition from videos using deep learning techniques», en *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, IEEE, 2019, págs. 80-85.

[46] A. Ali, A. Marisetty y F. Bremond, «P-Age: Pexels Dataset for Robust Spatio-Temporal Apparent Age Classification», en *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, págs. 8606-8615.

[47] T. Aremu, Z. Li, R. Alameeri, M. Khan y A. E. Saddik, «SSIVD-Net: A Novel Salient Super Image Classification & Detection Technique for Weaponized Violence», *Research Square*, 2023. DOI: 10.21203/rs.3.rs-3024402/v3. dirección: https://doi.org/10.21203/rs.3.rs-3024402/v3.

[48] H. Baumgartl y R. Buettner, «Developing efficient transfer learning strategies for robust scene recognition in mobile robotics using pre-trained convolutional neural networks», *arXiv preprint arXiv:2107.11187*, 2021.