Universidad Internacional de la Rioja (UNIR)

Escuela Superior de Ingeniería y Tecnología

Máster en en Ingeniería Matemática y Computación

Optimización de un pipeline basado en BI-LSTM para la detección de violencia en video

Trabajo Fin de Estudios

presentado por: Cesar Antonio Madera Garcés

Dirigido por: Pablo Negre Rodriguez

Ciudad:Lima, Perú

Fecha: 23 de Marzo de 2025

Índice de Contenidos

Índice de Ilustraciones

Índice de Tablas

Resumen

Nota: En este apartado se introducirá un breve resumen en español del trabajo realizado (extensión máxima: 150 palabras). Este resumen debe incluir el objetivo o propósito de la investigación, la metodología, los resultados y las conclusiones.

Palabras Clave: Se deben incluir de 3 a 5 palabras claves en español

Abstract

Nota: En este apartado se introducirá un breve resumen en español del trabajo realizado (extensión máxima: 150 palabras). Este resumen debe incluir el objetivo o propósito de la investigación, la metodología, los resultados y las conclusiones.

Palabras Clave: Se deben incluir de 3 a 5 palabras claves en inglés

1. Introducción

La violencia sigue siendo un problema global crítico, con millones de incidentes reportados anualmente. Según la Organización Mundial de la Salud (OMS), la violencia interpersonal ha permanecido como una de las 10 principales causas de muerte anual en las regiones de las Américas, con innumerables casos de agresión física y delitos violentos que no se reportan (Organization, 2024). América Latina, en particular, tiene algunas de las tasas de violencia más altas, con países como Perú, Chile, Brasil, Colombia y México enfrentando importantes desafíos en la prevención del crimen y la seguridad pública (Bisca y cols., 2024). En 2024, el Instituto Nacional de Estadística y Geografía (INEGI) reportó 21.9 millones de víctimas mayores de edad solo en México, y 31.3 millones de delitos (INEGI, 2024). La creciente disponibilidad de videovigilancia y medios digitales presenta una oportunidad para desarrollar sistemas automatizados capaces de detectar y mitigar incidentes violentos en tiempo real.

La inteligencia artificial (IA) ha emergido como una herramienta potente en el campo del análisis de video, ofreciendo soluciones prometedoras para la detección automatizada de violencia. Los modelos de aprendizaje profundo, particularmente las redes neuronales convolucionales (CNNs) y las redes de memoria a largo y corto plazo (LSTM), han demostrado capacidades excepcionales en el procesamiento de características espaciotemporales de los datos de video (Orozco, Buemi, y Berlles, 2021). Aprovechando estas tecnologías, los sistemas basados en IA pueden analizar flujos de video, reconocer acciones violentas y generar alertas con alta precisión. Sin embargo, desafíos como el desequilibrio de clases, la escasez de datos y los falsos positivos siguen siendo obstáculos críticos en las aplicaciones del mundo real (Kulkarni, Batarseh, y Chong, 2021). Esta investigación tiene como objetivo mejorar la robustez e interpretabilidad de los sistemas de detección de violencia impulsados por IA, contribuyendo a entornos más seguros en México y más allá.

1.1. Justificación

Las tasas crecientes de violencia en América Latina, particularmente en México como se expuso en la sección anterior, subrayan la urgente necesidad de sistemas avanzados de videovigilancia capaces de detectar incidentes en tiempo real. Los enfoques tradicionales de monitoreo, que dependen de la supervisión humana, a menudo son ineficientes debido a

la fatiga cognitiva y las limitaciones en la escalabilidad (Marois, Hodgetts, Chamberland, Williot, y Tremblay, 2021). La inteligencia artificial (IA), particularmente el aprendizaje profundo, ha demostrado un gran potencial para automatizar la detección de violencia mediante la integración de redes neuronales convolucionales (CNNs) y redes de memoria a largo y corto plazo (LSTM) (Negre, Alonso, Prieto, Dang, y Corchado, 2024; Negre, Alonso, Prieto, Garcia, y Corchado, 2024; Abdali y Al-Tuma, 2019; Sharma, Sudharsan, Naraharisetti, Trehan, y Jayavel, 2021). Mientras que las CNNs extraen características espaciales de los fotogramas de video, las LSTMs capturan dependencias temporales, lo que las convierte en una combinación poderosa para analizar escenas dinámicas. Sin embargo, el diseño óptimo de estos modelos sigue siendo un desafío abierto, ya que las variaciones en las arquitecturas de CNN y las configuraciones de LSTM afectan directamente la precisión de la detección, la eficiencia computacional y la aplicabilidad en el mundo real.

Este estudio busca investigar sistemáticamente el balance entre diferentes extractores de características de CNN y el número de celdas LSTM para determinar la pipeline más efectiva para la detección de violencia. La elección de la CNN influye en la calidad de la extracción de características, mientras que el número de celdas LSTM impacta en la capacidad del modelo para capturar patrones temporales sin incurrir en costos computacionales excesivos. Al optimizar este balance, la investigación busca mejorar tanto el rendimiento como la eficiencia de los sistemas de detección de violencia impulsados por IA. Los resultados contribuirán no solo al avance académico del análisis espaciotemporal de video, sino también al despliegue práctico de soluciones de videovigilancia robustas y escalables, mejorando finalmente la seguridad pública en México y más allá.

1.2. Enfoque de Trabajo

Habiendo establecido la justificación para esta investigación, es evidente que la selección de técnicas de extracción de características y el número de celdas LSTM juegan un papel crucial en la optimización de los modelos de detección de violencia. Los enfoques existentes a menudo pasan por alto el balance entre estos dos factores, lo que puede limitar el rendimiento en aplicaciones del mundo real. Por lo tanto, este estudio tiene como objetivo evaluar y refinar el balance entre los extractores de características de CNN y las configuraciones de celdas LSTM para desarrollar una pipeline más eficiente y precisa para la detección automatizada de violencia.

1.3. Estructura del Documento

por definir

2. Contexto y Estado del Arte

La Sección 2 proporciona un análisis detallado sobre el problema de la violencia y las metodologías actuales para su detección en video, ambos fundamentales para el desarrollo de esta investigación. Se aborda la clasificación de los diferentes tipos de violencia y su impacto global, así como una revisión exhaustiva de los modelos más utilizados en la literatura para identificar eventos violentos en entornos audiovisuales.

La Sección 2.1, establece una categorización de diversas formas de violencia, la cual destaca su ocurrencia en diferentes contextos como espacios públicos, entornos domésticos y situaciones de conflicto. Además, se presentan estadísticas relevantes para ilustrar la magnitud del problema a nivel global, con un enfoque particular en América Latina y sus tendencias recientes.

En la Sección 2.2, se revisa sistemáticamente los enfoques más representativos en la literatura para la detección de violencia en video. Cada metodología há sdo clasificada en función de su fundamento teórico y técnico, distinguiendo entre modelos basados en extracción manual de características, redes neuronales convolucionales (CNNs) y enfoques híbridos. Finalmente, se discute la necesidad de evaluar diferentes configuraciones de estos modelos para optimizar su rendimiento en la identificación de incidentes violentos.

2.1. Contexto

La proliferación de la violencia en todas partes del mundo constituye un problema social creciente que afecta la convivencia y el sentido de seguridad entre las personas. Dependiendo de las características de quienes cometen el acto, la violencia puede clasificarse en las siguientes categorías (OMS, 2014):

- Autoinfligida (conducta suicida y autolesiones),
- Interpersonal (violencia doméstica, incluyendo a niños, parejas y personas mayores; así como violencia entre personas no relacionadas),
- Colectiva (social, política y económica).

La OMS clasifica los actos violencia según la naturaleza como: física, sexual, psicológica, privación y negligencia. Con base en datos de 2014, indicó que “los actos repetidos de violencia que van desde la intimidación, el acoso sexual y las amenazas hasta la humillación

y el menosprecio de los trabajadores pueden convertirse en casos muy graves debido al efecto acumulativo. En Suecia, se estima que dicho comportamiento ha sido un factor en el 10 % al 15 % de los suicidios". En el mismo documento, se menciona que en el año 2000, hubo alrededor de 199,000 homicidios de jóvenes en todo el mundo (9.2 por cada 100,000 habitantes). Es decir, en promedio, mueren diariamente 565 niños, adolescentes y adultos jóvenes de entre 10 y 29 años como resultado de la violencia interpersonal. Las tasas de homicidio varían considerablemente según la región, desde 0.9 por cada 100,000 habitantes en países de altos ingresos en Europa y algunas partes de Asia y el Pacífico hasta 17.6 en África y 36.4 por cada 100,000 en América Latina.

Por otro lado, el informe del Fondo Monetario Internacional revela un aumento exponencial en la sensación de inseguridad y una mayor aceptación de que los crímenes violentos son, unánimemente, el problema más importante desde 2020, como se muestra en la Figura 2.1 (Bisca y cols., 2024):

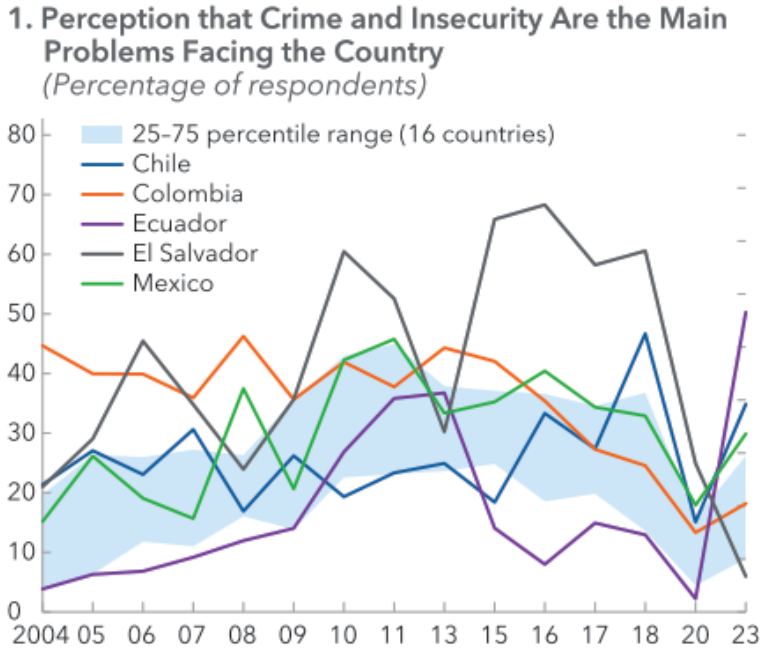

Figura 2.1: Crecimiento de la percepción de que la inseguridad es una prioridad (Bisca y cols., 2024)

En este sentido, este problema representa la prioridad más importante que cualquier gobierno debería abordar. Por esta razón, en este proyecto se abordará la detección de violencia interpersonal directa.

2.2. Marco Teórico

En la presente sección, se revisarán algunos conocimientos previos que serán útiles para comprender mejor tanto el problema de investigación como la solución propuesta. Entre estos conocimientos previos, se explicarán en detalle algunos modelos de vanguardia en clasificación de imágenes utilizando ML, así como el procedimiento de *Transfer Learning* (TL), el preprocesamiento de entradas y el *dataset* previsto para utilizar en este trabajo.

2.2.1. Modelos de ML para Clasificación

El ML ha evolucionado con el tiempo de tal manera que ahora diferentes modelos pueden realizar tareas más complejas, incluso superando a los humanos en eficiencia y precisión en algunos casos. Entre estas tareas, la clasificación es una de las áreas más investigadas y también cuenta con diversas aplicaciones en el mundo real. Dentro de este campo, el *Multi Layer Perceptron* (MLP), las *Convolutional Neural Networks* (CNN) y las *Long-Short Term memory*(LSTM) son las arquitecturas más comúnmente utilizadas.

Multi Layer Perceptron (MLP)

Esta arquitectura consta de múltiples capas de neuronas artificiales, que reciben los datos a procesar y los pasan a través de una función de activación para convertirse en la entrada de la siguiente capa. Este proceso imita el comportamiento de las neuronas humanas, como se muestra en la Figura 2.2.

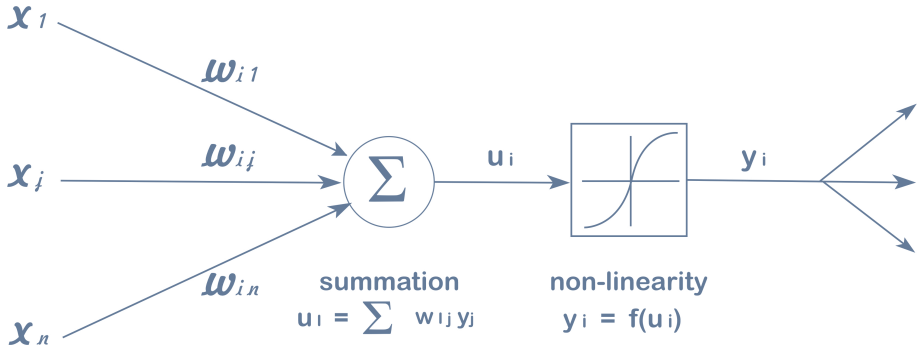

Figura 2.2: Comparación entre una neurona real y una artificial (Magiquo, 2019).

Un MLP consiste en varias capas "densas" de neuronas interconectadas. Una vez que la entrada es procesada a través de la red neuronal, comienza el proceso de modificación de los pesos w de cada capa. Este proceso se llama "retropropagación" y consiste en

propagar las derivadas de la función de cada neurona desde la última capa hasta la primera.

Este modelo espera como entrada un vector de datos lineal. Es por eso que este tipo de modelo no es particularmente bueno para extraer información de imágenes, ya que hacerlo sería computacionalmente complejo. Como alternativa, podría utilizarse un extractor de características para mejorar la eficiencia. Desafortunadamente, esto resultaría en pérdida de información. Además, los datos de imagen suelen cumplir con el principio de localidad (la información relevante tiende a concentrarse en zonas cercanas), y vectorizarlos omitiría características importantes para el análisis.

Convolutional Neural Networks (CNN)

Las CNN son modelos basados en una arquitectura diseñada específicamente para el análisis de imágenes, en particular la clasificación. Estas realizan la tarea de extracción de características que los MLP no pueden, a través de convoluciones. Esto evita la pérdida de información y mejora tanto la eficiencia como la precisión.

Las convoluciones se basan en un procedimiento que extrae características mediante la aplicación de una pequeña matriz de transformación cuadrada sobre la imagen original, la cual retorna una imagen modificada. Estas matrices de transformación se denominan kernels. La Figura 2.3 ilustra una iteración de convolución.

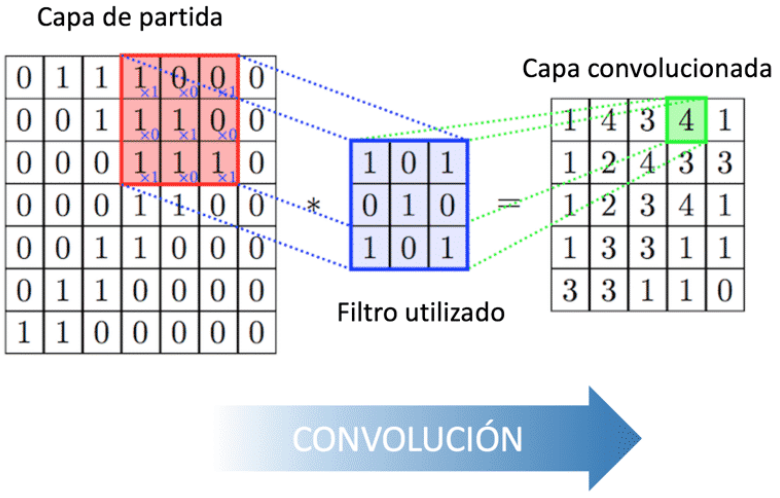

Figura 2.3: Proceso simplificado de una convolución(Diego Calvo, 2019a).

Por otro lado, existen otras capas importantes llamadas *poolings*, como se muestra en la Figura 2.4. Estas capas aplican una lógica a un cuadrante de la matriz resultante de las convoluciones. Esta lógica varía dependiendo de las necesidades del usuario. La figura muestra una comparación entre *Max Pooling* y *Average Pooling*, que obtienen respectivamente los valores máximos y promedios de los cuadrantes seleccionados para generar un nuevo resultado con menor dimensionalidad.

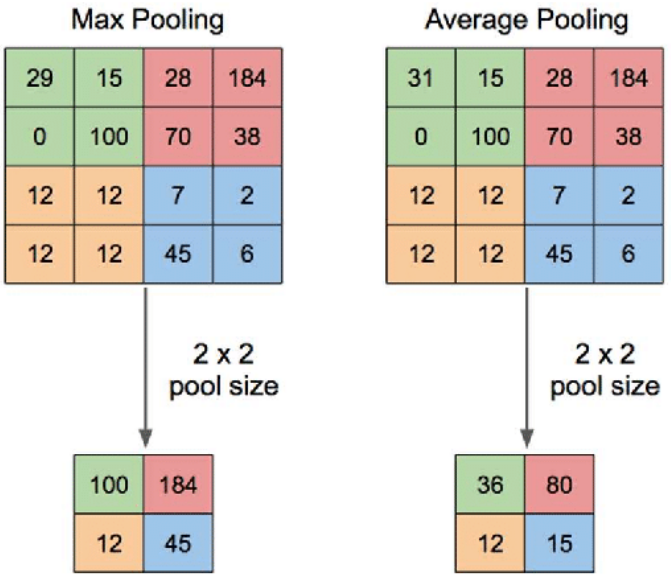

Figura 2.4: Proceso simplificado de un pooling (Muhamad Yani, Budhi Irawan, Casi Setianingsih, 2019).

Las CNN están compuestas por combinaciones repetidas de capas convolucionales y de *pooling*, finalizando en un MLP que realiza el procesamiento final de las características extraídas. La Figura 2.5 muestra la estructura de una CNN. Como se mencionó anteriormente, la extracción de características se realiza dentro de la propia red neuronal, evitando la pérdida de información que se observa en los MLP y dejando únicamente la tarea de clasificación a estos últimos.

2.2.2. State-of-the-Art Architectures

VGG-19

Esta CNN tiene una profundidad de 19 capas y fue creada por Karen Simonyan y Andrew Zisserman (Simonyan y Zisserman, 2015) en la Universidad de Oxford en 2014, y publicada posteriormente en 2015. Su versión detallada se ilustra en la Figura 2.6. Este

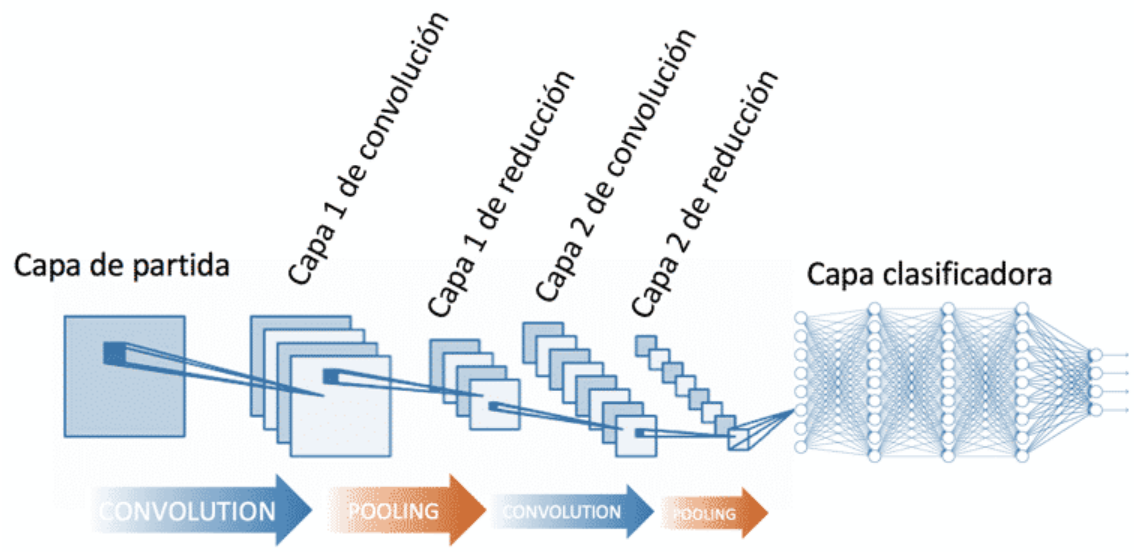

Figura 2.5: Arquitectura de una CNN convencional (Diego Calvo, 2019b).

modelo fue utilizado para clasificar imágenes en el *dataset* ImageNet, logrando clasificar hasta 1000 objetos diferentes. Espera una imagen de entrada de 224x224 píxeles para su procesamiento. Debido a su profundidad, este modelo es bastante pesado, llegando a consumir hasta 550 MB de memoria con alrededor de 143 millones de parámetros. Cabe destacar que el 70 % de estos parámetros se encuentran entre la última capa convolucional y la primera capa de clasificación. Con todas estas características, logró un 90 % de precisión en ImageNet.

InceptionV3

Aunque VGG19 logró una alta precisión, consumía demasiados recursos. Por esta razón, Google desarrolló InceptionV3, una red que prometía resultados similares pero a un menor costo. Presentado en "*Going deeper with convolutions*" (Szegedy y cols., 2014), este modelo alcanza un 93.7 % de precisión en el mismo *dataset*.

Aunque esta red tiene 50 capas de profundidad (más que VGG19), tiene menos parámetros entrenables (23.8 millones). Su diseño propuso que realizar convoluciones unidimensionales en serie (como se muestra en la Figura 2.7) es equivalente a una convolución bidimensional, evitando el uso de filtros matriciales. Esto redujo la complejidad de los modelos CNN convencionales, haciéndolo más liviano en memoria y mejorando su capacidad

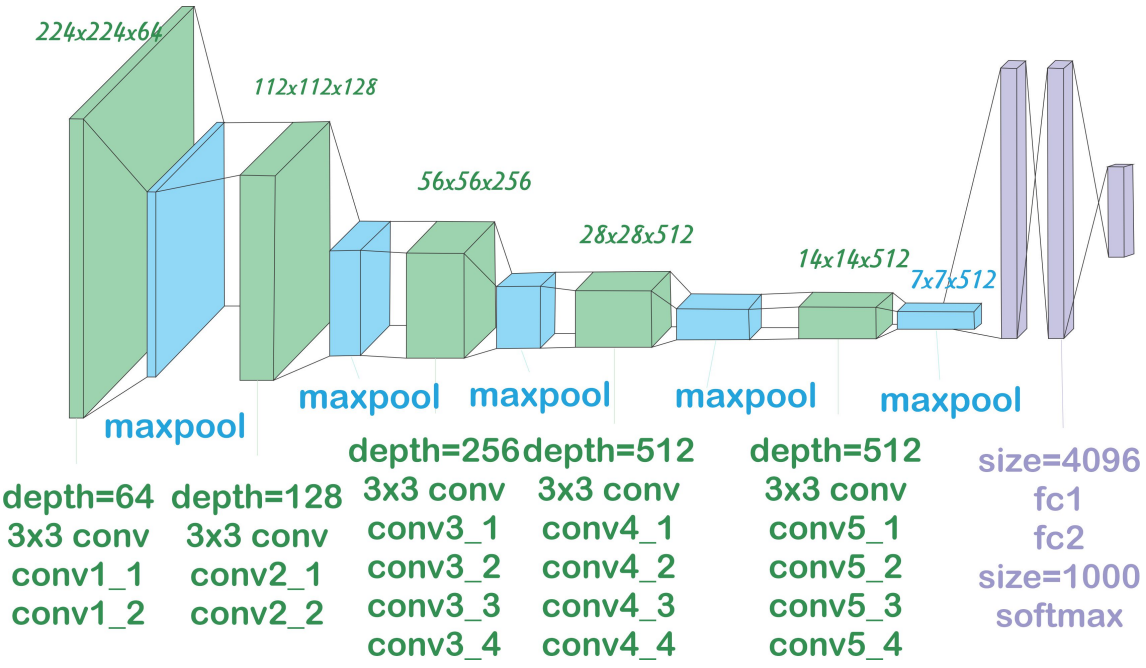

Figura 2.6: Version simplificada de una VGG19 (IchiPro, 2020).

de aprendizaje. En total, este modelo pesa 92 MB—aproximadamente seis veces menos que el anterior. Requiere imágenes de entrada de 299x299 píxeles, y su arquitectura se muestra en la Figura 2.8.

ResNet50

Creado por Microsoft en 2015, este modelo también cuenta con 50 capas de profundidad. Emplea una técnica llamada "residual learning" (He, Zhang, Ren, y Sun, 2015), que consiste en guardar una copia de la salida actual y sumarla al resultado obtenido de un conjunto de convoluciones (típicamente cada tres). La Figura 2.9 ilustra esta modificación y la arquitectura general del modelo. Este modelo también fue probado en el mismo *dataset*, obteniendo una precisión del 92.1 %, y el aprendizaje residual evitó un aumento en la dimensionalidad del modelo. Contiene aproximadamente 25.6 millones de parámetros y ocupa 98 MB de memoria.

EfficientNet

"*EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*" (Tan y Le, 2020) en 2020, consiste en 8 implementaciones diferentes (B0 a B7). La versión más ligera

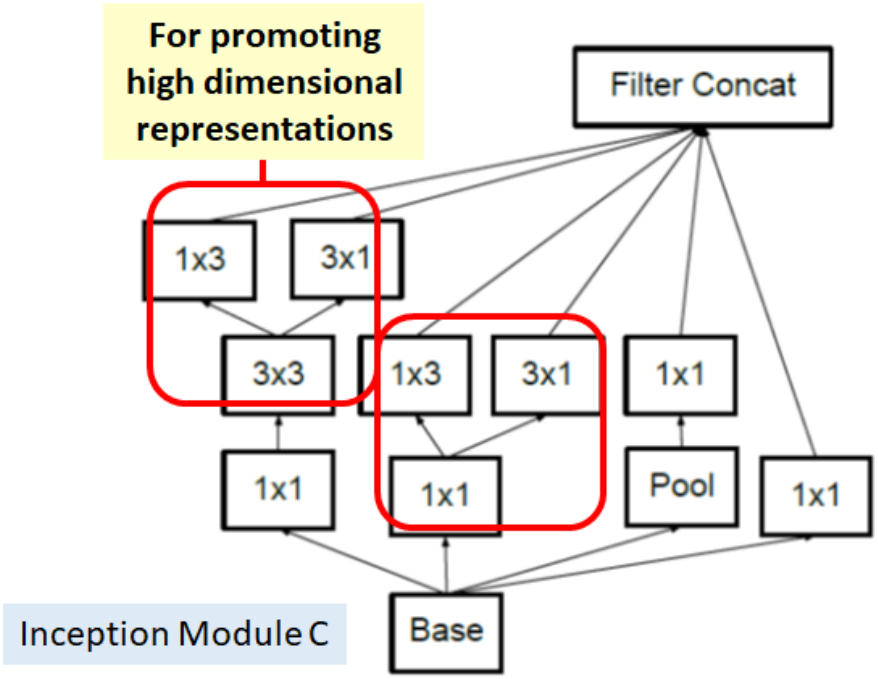

Figura 2.7: Reducción de dimensionalidad de InceptionV3 (IchiPro, 2020).

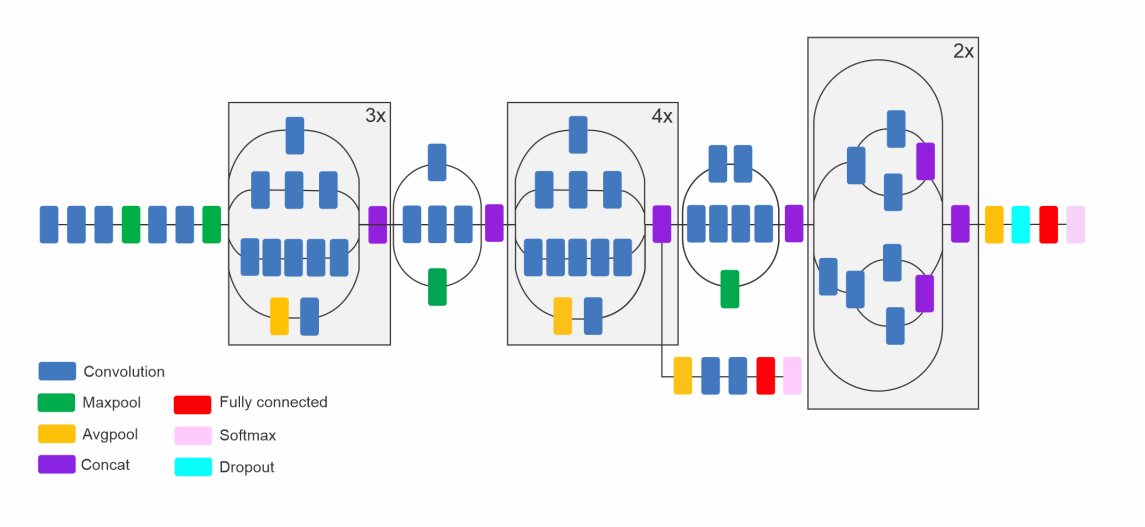

Figura 2.8: Versión simplificada de InceptionV3 (IchiPro, 2020).

(B0) contiene aproximadamente 5.5 millones de parámetros y logró un 93 % de precisión en el mismo *dataset*. Otras versiones continúan aumentando el número de parámetros y la precisión resultante. Comparado con los modelos anteriores, EfficientNetB4 con 19.5 millones de parámetros alcanza un 96.4 % de precisión, superándolos. Este modelo optimiza su aprendizaje mediante un algoritmo de aproximación que genera parámetros para crear

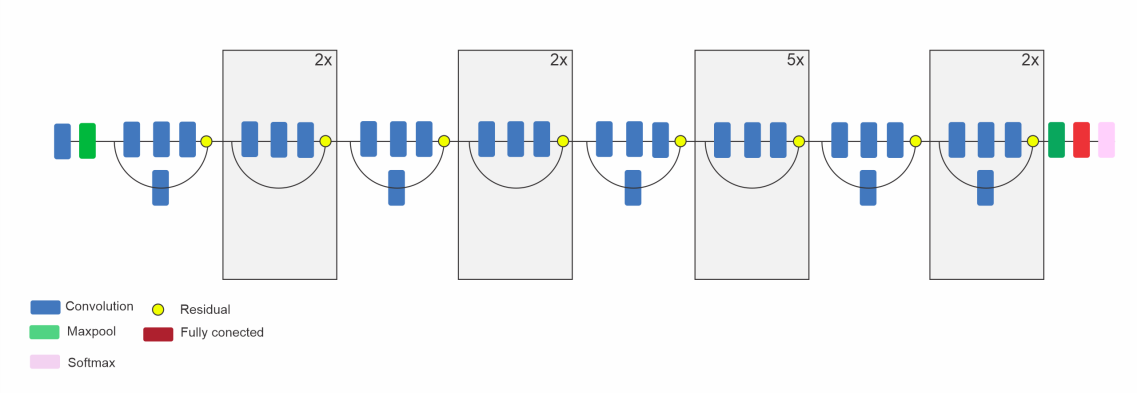

Figura 2.9: Versión simplificada de ResNet50 (IchiPro, 2020).

cada uno de los ocho modelos. Este algoritmo considera tres factores:

- Profundidad de las capas
- Ancho de las capas (capas múltiples)
- Resolución de la imagen

La Figura 2.10 muestra la estructura de EfficientNetB0.

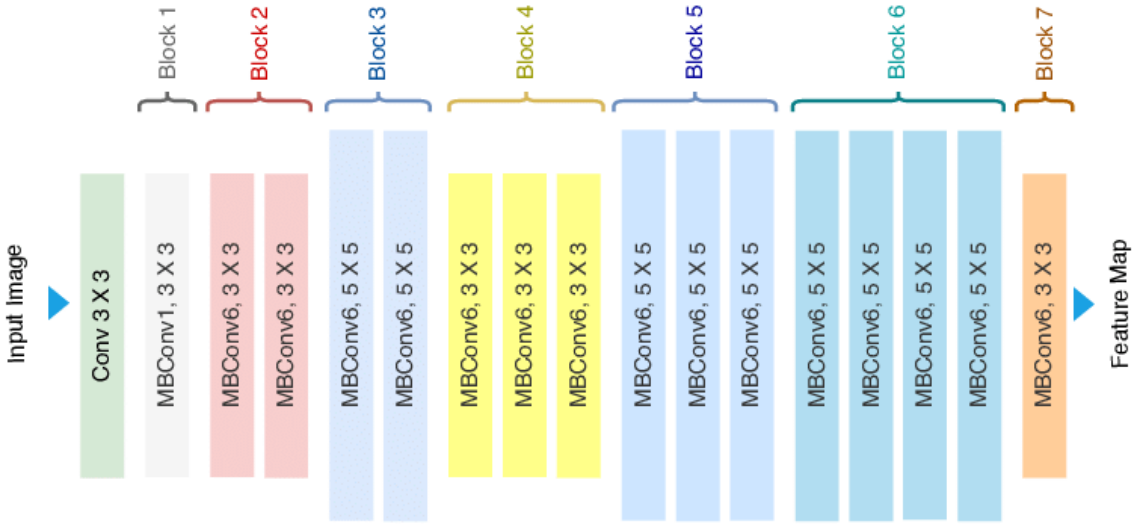

Figura 2.10: Versión simplificada de EfficientNetB0 (Tashin Ahmed, 2020).

YOLO

YOLO (*you only look once*) es una arquitectura que permite la "predicción simultánea de múltiples *bounding boxes* y probabilidades de clase para ellas" (Redmon, Divvala, Girshick, y Farhadi, 2015). Este modelo está basado en una CNN convencional, que, a diferencia de implementaciones anteriores (R-CNN y FR-CNN), realiza la predicción de la caja de enlace internamente, reduciendo la latencia y habilitando su uso en tiempo real.

Este cálculo se basa en generar cuadrículas o particiones de la imagen, dentro de las cuales se inicializa un número predeterminado de *bounding boxes* predefinidas (ambos son hiperparámetros de la arquitectura). Esto es replicable usando cualquier arquitectura CNN como base (por ejemplo, uno de los modelos mencionados previamente), solo ajustando la salida y los hiperparámetros en consecuencia. Las versiones más recientes logran una mejor detección, módulos de atención y otras variaciones que la hacen cada vez más robusta. La Figura 2.11 muestra un ejemplo de la arquitectura del modelo YOLOv5.

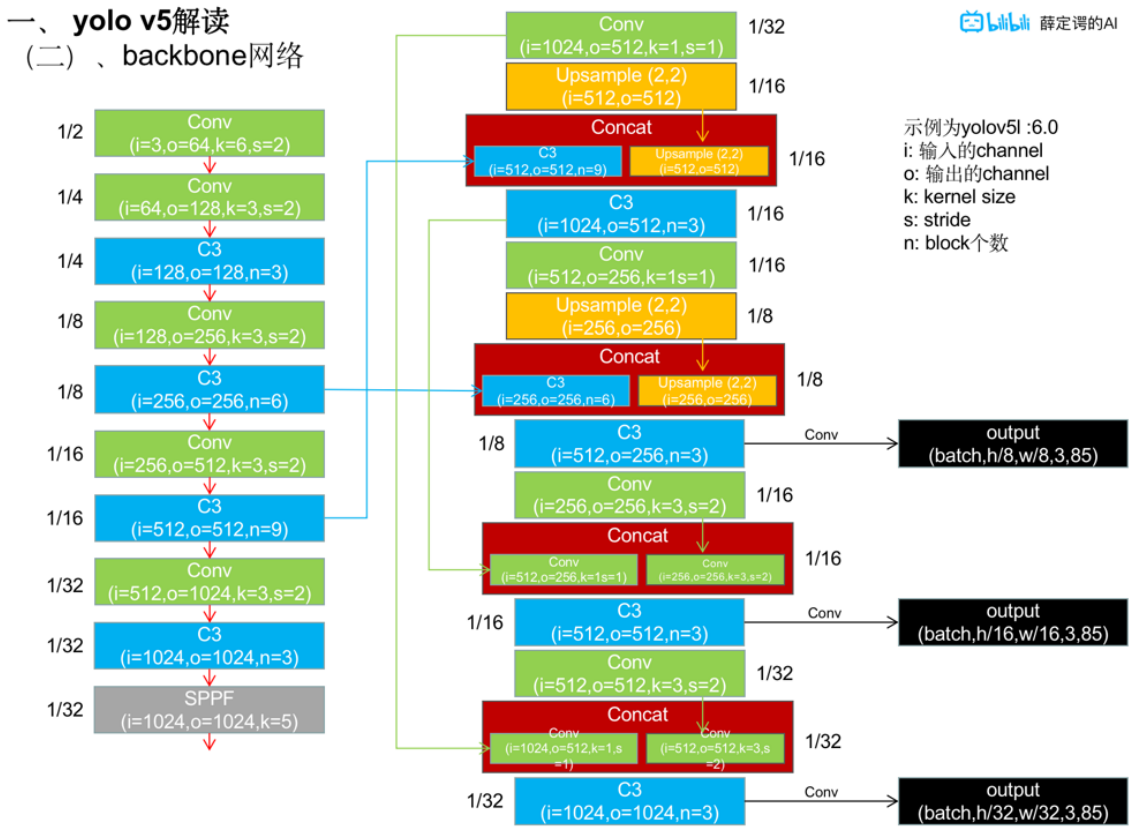

Figura 2.11: Representación de la arquitectura de YOLOv5(Jocher y cols., 2022).

Long Short-Term Memory (LSTM)

Las redes *Long Short-Term Memory* (LSTM) fueron desarrolladas como una extensión de las redes neuronales recurrentes (RNN) con el propósito de superar la dificultad de aprender dependencias a largo plazo en secuencias (Hochreiter y Schmidhuber, 1997). Las RNN tradicionales tienden a sufrir del problema del desvanecimiento o explosión del gradiente durante el proceso de entrenamiento, lo que limita su capacidad para capturar relaciones temporales distantes. Las LSTM abordan esta limitación mediante la incorporación de una memoria interna controlada por compuertas, lo que permite conservar información relevante a lo largo del tiempo.

La arquitectura de las LSTM se basa en una celda de memoria capaz de mantener su estado a lo largo de múltiples pasos temporales. Esta celda está regulada por tres compuertas fundamentales: la compuerta de entrada, que controla qué nueva información se almacena en la celda; la compuerta de olvido, que decide qué información debe eliminarse del estado anterior; y la compuerta de salida, que determina qué parte del contenido de la celda se utiliza como salida. Estas compuertas permiten que la red aprenda a retener o descartar información de manera adaptativa, mejorando significativamente el aprendizaje de secuencias largas.

Gracias a esta estructura, las LSTM han demostrado un rendimiento notable en una amplia gama de tareas secuenciales donde el contexto temporal es esencial. Aplicaciones como el modelado del lenguaje natural, la traducción automática, el reconocimiento de voz y el análisis de series temporales se han beneficiado enormemente de su capacidad para capturar dependencias a largo plazo. En consecuencia, las LSTM se han convertido en una arquitectura fundamental dentro del campo del aprendizaje profundo secuencial.La arquitectura de este modelo se puede ver en la imagen 2.12:

2.2.3. *Transfer Learning*

Según Muhamad Yani (Yani, Irawan, y Setiningsih, 2019), *Transfer Learning (TL)* se define como “el proceso de transferir el conocimiento de un entrenamiento previo para ser utilizado en un nuevo modelo con el fin de reducir el tiempo de aprendizaje”. Este proceso puede ser observado en la Figura 2.13.

TL difiere del proceso de entrenamiento convencional de una red en que no es necesario

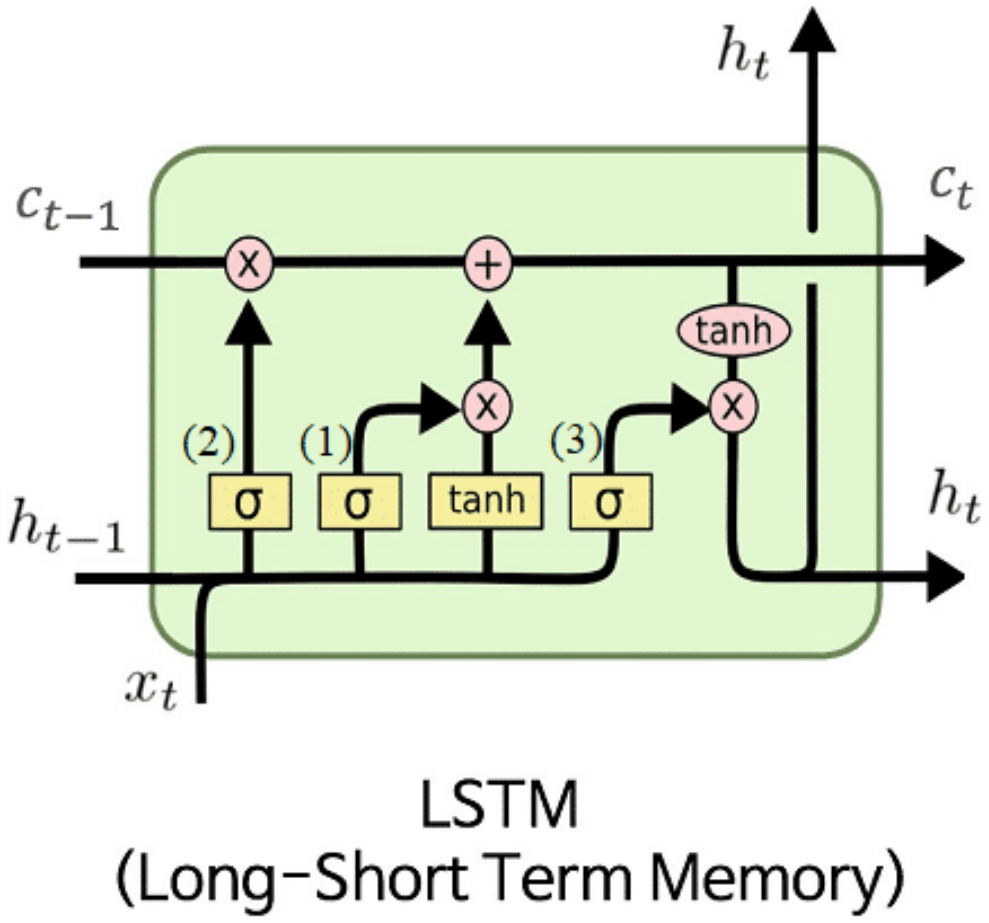

Figura 2.12: Representación de la arquitectura de LSTM(DataScientest, 2024).

entrenarla con un gran conjunto de datos. A diferencia del proceso tradicional, las capas iniciales del modelo (en nuestro caso, las capas convolucionales) están congeladas. Estas capas contienen todo el conocimiento pre-aprendido del conjunto de datos con el cual el modelo fue originalmente entrenado. Una vez obtenidas esas capas, se agregan nuevas capas densas (similares a las de un MLP) que sirven como clasificadores para nuestro propósito específico. Gracias a esto, solo las capas finales necesitan ser entrenadas, lo que requiere menos datos y generalmente resulta en predicciones precisas. Este tipo de entrenamiento se llama "*fine-tuning*", lo que permite que la red ajuste el aprendizaje previo para predecir en función de un conjunto de datos diferente al que fue entrenada originalmente, ahorrando tiempo y mejorando la precisión.

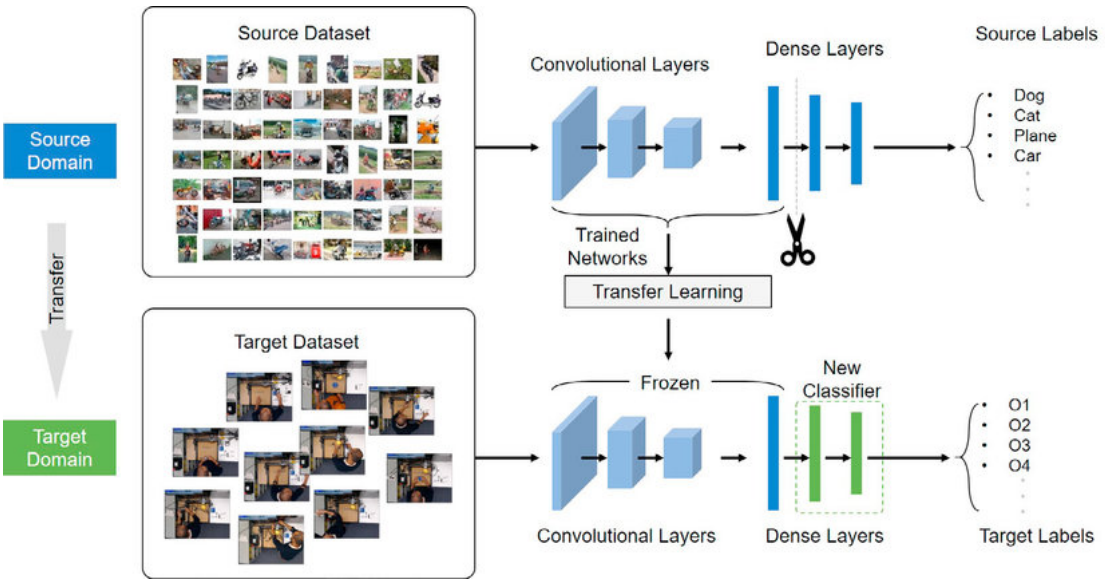

Figura 2.13: Comparativa entre un entrenamiento normal y por TL(Wenjin Taoa, Md Al-Aminb, Haodong Chena, Ming C. Leua, Zhaozheng Yinc, Ruwen Qinb, 2020).

2.2.4. Preprocesamiento de Entradas

Ahora que entendemos cómo funcionan estos modelos en general, necesitamos observar las características requeridas de las entradas para que el modelo aprenda de manera efectiva.

Tamaño de la Entrada

Considerando solo las diferentes CNN, cada modelo espera una matriz de entrada (imagen) de un tamaño específico. En las implementaciones actuales, se usan comúnmente tensores para referirse a un *batch* (conjunto) de imágenes. La representación es la siguiente:

$$(batch, channels, m, n), donde:$$

- batch: Número de imágenes que se ingresan a la vez.
- channels: Número de canales de la imagen (para imágenes a color, hay 3 canales que representan RGB, mientras que las imágenes en escala de grises tienen solo 1 canal).
- m & n: Dimensiones de la imagen. Estos valores dependen de la arquitectura del modelo, ya que cada capa realizará operaciones que reducen la dimensionalidad de

la imagen. Esto varía según la implementación debido a las diversas configuraciones posibles para cada matriz convolucional y operación de pooling.

One Hot Encoder

Este es un tipo de representación que consiste en crear una matriz identidad de tamaño $n \times n$, donde n es el número de *etiquetas*. La codificación de cada etiqueta puede tomarse como una de las filas de la matriz. Así, podemos representar la codificación de un objeto de la clase j entre n clases como:

$$OneHotEncoder(i, j, n) = [a_1, a_2, ..a_n], donde :$$

$$a_{i,j} = \begin{cases} 1 & \text{si j = i} \\ 0 & \text{otherwise} \end{cases}$$

Este tipo de codificación tiene la ventaja de evitar relaciones entre etiquetas, y permite una clasificación sencilla al comparar resultados. En el lado negativo, puede llevar a una alta dimensionalidad cuando hay muchas *etiquetas*.

2.2.5. *Overfitting*

El proceso de aprendizaje de los modelos se basa en la retroalimentación obtenida a través de la retropropagación mencionada anteriormente. Una vez que se actualizan los pesos de cada neurona utilizando el gradiente resultante, se puede decir que el modelo ha aprendido a clasificar esa imagen específica. Sin embargo, este gradiente puede desvanecerse debido a la profundidad de la arquitectura, las funciones de activación u otras razones. Este gradiente desvanecido impide que el modelo siga aprendiendo del conjunto de datos, lo que lo hace inutilizable para aplicaciones del mundo real. Algunas estrategias utilizadas para abordar esto son:

- *Data Augmentation*: Expansión del conjunto de datos original utilizando rotaciones, escalados, recortes, volteos, etc. Esto hace que el modelo sea más resistente a los cambios en la posición o orientación del objeto en la imagen.

- *Batch Normalization*: Reescalado de los datos de entrada a diferentes rangos relativos a una escala común, generando una distribución de datos más manejable.

- *Dropout*: Desactivación aleatoria de un porcentaje de neuronas artificiales en una capa. Esto impide que cada neurona dependa de las desactivadas, mejorando generalmente el rendimiento.

Este capítulo presentó los conocimientos previos que ayudarán a los lectores a comprender mejor la metodología que se introducirá más adelante particularmente las arquitecturas de los modelos que se revisarán, utilizarán y compararán para lograr los mejores resultados posibles en la detección y clasificación de imágenes de peces dentro de la fauna marina peruana.

2.3. Estado del arte

En la presente sección se presentarán algunos trabajos realizados enfocados al uso de *pipelines* para la detección y clasificación de violencia en videos.

2.3.1. *Pipelines* actuales para la clasificacion de violencia

La literatura actual muestra un creciente interés en la detección automática de violencia en videos mediante la combinación de redes convolucionales (CNN) y redes de memoria a largo plazo (LSTM). Las CNN se utilizan para extraer características espaciales relevantes de los fotogramas, permitiendo identificar patrones visuales y contextos que podrían indicar situaciones violentas. Posteriormente, estas características son procesadas por las LSTM, las cuales se encargan de capturar la dinámica temporal y las dependencias de largo plazo entre los fotogramas, mejorando la capacidad de detectar eventos violentos que se desarrollan a lo largo del tiempo.

En diversos estudios se ha demostrado que la integración de ambas arquitecturas permite explotar de forma sinérgica las ventajas de cada una: las CNN fortalecen la comprensión del contenido visual a nivel de detalle y textura, mientras que las LSTM aportan una perspectiva temporal que es crucial para identificar patrones de comportamiento complejos y secuenciales propios de la violencia. Este tipo de *pipeline* ha sido evaluado en diferentes escenarios, incluyendo videos de vigilancia y grabaciones deportivas, logrando así mejores tasas de detección en comparación con métodos que utilizan únicamente análisis espacial o temporal de manera aislada.

Un claro ejemplo de lo mencionado anteriormente, es el artículo de Jeff Donahue, *et al.* (Donahue y cols., 2014). Su investigación se centra en el desafío de trabajar con datos visuales que contienen información espacial y temporal. La idea central es integrar ambas arquitecturas en un único sistema end-to-end: las CNNs se encargan de extraer representaciones ricas de contenido visual, y las LSTMs modelan la dinámica secuencial, permitiendo aplicaciones en reconocimiento de actividades en video y generación de descripciones de imágenes.

Entre sus principales contribuciones, este trabajo logró sentar las bases las bases en la integración de arquitecturas visuales y secuenciales. Su enfoque ha influido en numerosos trabajos posteriores en áreas como la descripción automática de videos, el análisis de secuencias complejas y el desarrollo de modelos más integrados para tareas multimodales.

Utilizando esta lógica, el trabajo de Orozco, *et al.*(Orozco y cols., 2021) el cual destaca la efectividad de estas estrategias al combinar etapas de preprocesamiento, extracción de características espaciales mediante CNN, y análisis secuencial temporal con LSTM. Asimismo, se discuten los desafíos asociados a la variabilidad de escenarios y la detección en tiempo real, aspectos que continúan motivando la investigación y optimización de estos sistemas. Como resultado final de su experimentación, los autores lograron un F1-score de 91 % lo cual indica que fue *pipeline* robusto el problema que intentó resolver basado en la clasificación de acciones humanas basado en videos. Los pipelines híbridos basados en CNN y LSTM representan una tendencia robusta en el campo de la detección de violencia, contribuyendo significativamente a la mejora en la precisión y eficiencia de los sistemas de análisis de video.

Otro artículo que también utiliza esta misma lógica es el de Swathikiran Sudhakaran and Oswald Lanz(Sudhakaran y Lanz, 2017). Su objetivo consistió en el mismo del presente trabajo, la detección de violencia. Para ello, se utilizó el *dataset* de Hockey Fights, el cuál contiene videos de eventos violentos en partidos de hockey. Este *dataset* es bastante usado como *benchmark* y consiste de un conjunto de imágenes etiquetadas de manera homogenea. La variación más significativa que realizaron fue el uso de celdas convLSTM en vez de LSTM regulares para poder obtener un *accuracy* más elevado. Con su *pipeline* lograron obtener una *accuracy* de 97 %, generando un nuevo record para este *benchmark*

y marcando un nuevo estado del arte.

De la misma manera y utilizando el mismo *dataset* de Hockey Fights, se encuentra el trabajo de Al-Maamoon R. Abdali y Rana F. Al-Tuma(Abdali y Al-Tuma, 2019). Ellos se basaron en el trabajo anteriormente mencionado para su implementación. Para ello, utilizaron la misma configuración del *pipeline* pero incluyeron capas conv3d y 40 celdas para el aprendizaje de la LSTM sin ser convolucionales. Utilizando aquella configuración logrando obtener un *accuracy* de 96.33 % pero al mismo tiempo obteniendo una mejora de 4 veces en la velocidad (representado en el número de fps), representando una mejora del estado del arte en su tiempo al mantener el mismo nivel de *accuracy* pero mejorando el *performance*.

El último artículo a revisar es el de Patel Mann (Mann, 2021). En su trabajo utilizó Resnet50, InceptionV3 y VGG19 como extractores de características. En cambio, utilizaron solo una celda LSTM para la clasificación y como resultados obtuvieron 90 % de precisión para el dataset de Hockey fights, representando una disminución a comparación de los anteriores trabajos, aunque permitió optimizar el uso de memoria sin perder drásticamente el *performance* de todo el *pipeline*.

Como se puede ver, el *pipeline* CNN-LSTM propuesto ha sido utilizado a lo largo de los últimos años para resolver problemas similares e iguales al nuestro. En ese sentido nos hace sentido tratar de extender la aplicación de este y optimizarlo para tratar de ver como es que diferentes configuraciones del mismo permiten mejorar ya sea el *performance* o el *accuracy* de la propuesta.

3. Objetivos

3.1. Objetivos

En este capítulo se presentarán los objetivos y contribuciones de esta tesis. Una comprensión clara de estos aspectos es esencial para contextualizar el alcance y la relevancia de esta investigación. Las siguientes secciones ofrecerán una discusión detallada de los objetivos clave perseguidos en este trabajo, así como de las contribuciones que se pretende aportar al campo.

- Objetivo principal:
  - optimizar el equilibrio entre la modificación del extractor de características CNN y el ajuste del número de celdas LSTM para construir el pipeline más efectivo para la detección de violencia
- Objetivos secundarios:
  - Evaluar el impacto de diversas arquitecturas CNN en la calidad de las características espaciotemporales extraídas para la detección de violencia.
  - Investigar cómo la variación en el número de celdas LSTM afecta la modelación temporal y el rendimiento en la clasificación.
  - Identificar el equilibrio óptimo entre la complejidad de la extracción de características por CNN y la capacidad de las LSTM para lograr el mejor rendimiento con un costo computacional mínimo.
  - Automatizar el proceso de etiquetado mediante la creación de una aplicación en tiempo real para el pipeline.

3.2. Contribuciones

La contribución principal de esta tesis es el desarrollo de un pipeline optimizado para la detección de violencia que equilibra la complejidad de la extracción de características basada en CNN y el número de celdas LSTM para lograr una mayor precisión y eficiencia. Al analizar sistemáticamente los compromisos entre estos dos componentes, este trabajo

proporciona un enfoque estructurado para diseñar arquitecturas de deep learning destinadas al reconocimiento espaciotemporal de violencia, mejorando tanto el rendimiento en la detección como la viabilidad computacional. Esta contribución tiene como objetivo avanzar en el análisis de video impulsado por IA para aplicaciones de vigilancia en tiempo real.

4. Metodología

5. Desarrollo del trabajo

6. Conclusiones y Trabajo Futuro

En la Ecuación (6.1)

$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \tag{6.1}$$

En la siguiente Tabla 6.1

1	2
22	11

Tabla 6.1: Tabla 1

En la siguiente Figura 6.1

Figura 6.1: Logo Unir

Referencias

Abdali, A. M. R., y Al-Tuma, R. F. (2019, 3). Robust real-time violence detection in video using cnn and lstm. *SCCS 2019 - 2019 2nd Scientific Conference of Computer Sciences*, 104-108. doi: 10.1109/SCCS.2019.8852616

Bisca, P. M., Chau, V., Dudine, P., Espinoza, R. A., Fournier, J.-M., Guérin, P., ... Salas, J. (2024, 11). Violent crime and insecurity in latin america and the caribbean – a macroeconomic perspective. *Departmental Papers, 2024*. Descargado de `https://www.elibrary.imf.org/view/journals/087/2024/009/article-A001-en.xml` doi: 10.5089/9798400288470.087.A001

DataScientest. (2024). *Memoria a largo plazo a corto plazo (lstm): ¿qué es?* Descargado de `https://datascientest.com/es/memoria-a-largo-plazo-a-corto-plazo-lstm` (Accedido el 13 de abril de 2025)

Diego Calvo. (2019a). *red-neuronal-convolucional.* Descargado de https://www.diegocalvo.es/red-neuronal-convolucional/ ([Online; accessed December 05, 2021])

Diego Calvo. (2019b). *red-neuronal-convolucional-arquitectura.* Descargado de https://www.diegocalvo.es/red-neuronal-convolucional/red-neuronal-convolucional-arquitectura/ ([Online; accessed December 05, 2021])

Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., y Darrell, T. (2014). Long-term recurrent convolutional networks for visual recognition and description. *CoRR*, *abs/1411.4389*. Descargado de http://arxiv.org/abs/1411.4389

He, K., Zhang, X., Ren, S., y Sun, J. (2015). *Deep residual learning for image recognition.*

Hochreiter, S., y Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

IchiPro. (2020). *4 modelos de cnn previamente entrenados para usar en visión artificial con aprendizaje por transferencia.* Descargado de https://ichi.pro/es/4-modelos-de-cnn-previamente-entrenados-para-usar-en-vision-artificial-con-aprendizaje-por-transferencia-9370731228668 ([Online; accessed December 05, 2021])

INEGI. (2024). *Encuesta nacional de victimizaciÓn y percepciÓn sobre seguridad pÚblica (envipe) 2024* (Inf. Téc.). Autor. Descargado de https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2024/ENVIPE/ENVIPE_24.pdf

Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., ... Jain, M. (2022, noviembre). *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation.* Zenodo. Descargado de https://doi.org/10.5281/zenodo.7347926 doi: 10.5281/zenodo.7347926

Kulkarni, A., Batarseh, F. A., y Chong, D. (2021). Chapter 5: Foundations of data imbalance and solutions for a data democracy. *ArXiv*.

Magiquo. (2019). *Atomicredes neuronales o imitar al cerebro humano?* Descargado de https://magiquo.com/wp-content/uploads/2019/11/neurona.png ([Online; accessed December 05, 2021])

Mann, P. (2021, 7). Real-time violence detection using cnn-lstm. *charotar university of science and technology*. Descargado de https://www.researchgate.net/publication/353330450_Real-Time_Violence_Detection_Using_CNN-LSTM doi:

10.48550/arXiv.2107.07578

Marois, A., Hodgetts, H. M., Chamberland, C., Williot, A., y Tremblay, S. (2021, 7). Who can best find waldo? exploring individual differences that bolster performance in a security surveillance microworld. *Applied Cognitive Psychology*, *35*, 1044-1057. doi: 10.1002/ACP.3837

Muhamad Yani, Budhi Irawan, Casi Setianingsih. (2019). *Application of transfer learning using convolutional neural network method for early detection of terry's nail.* Descargado de `https://www.researchgate.net/figure/Illustration-of-Max-Pooling-and-Average-Pooling-Figure-2-above-shows-an-example-of-max_fig2_333593451` ([Online; accessed December 05, 2021])

Negre, P., Alonso, R. S., Prieto, J., Dang, C. N., y Corchado, J. M. (2024, 3). Systematic mapping study on violence detection in video by means of trustworthy artificial intelligence. *SSRN Electronic Journal*. Descargado de `https://papers.ssrn.com/abstract=4757631` doi: 10.2139/SSRN.4757631

Negre, P., Alonso, R. S., Prieto, J., Garcia, O., y Corchado, J. M. (2024, 5). Violence detection in video models implementation using pre-trained vgg19 combined with manual logic, lstm layers and bi-lstm layers. *SSRN Electronic Journal*. Descargado de `https://papers.ssrn.com/abstract=4832475` doi: 10.2139/SSRN.4832475

OMS. (2014). Violencia y salud mental. *Organismo Mundial de la Salud*. Descargado de `https://www.uv.mx/psicologia/files/2014/11/violencia-y-salud-mental-oms.pdf`

Organization, W. H. (2024). Monitoring health for the sdgs, sustainable development goals. *World Health Organization Journal*.

Orozco, C. I., Buemi, M. E., y Berlles, J. J. (2021, 6). Cnn–lstm con mecanismo de atención suave para el reconocimiento de acciones humanas en videos. *Elektron*, *5*, 37-44. doi: 10.37537/REV.ELEKTRON.5.1.130.2021

Redmon, J., Divvala, S., Girshick, R., y Farhadi, A. (2015, jun). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 779–788. Descargado de `https://arxiv.org/abs/1506.02640v5` doi: 10.48550/arxiv.1506.02640

Sharma, S., Sudharsan, B., Naraharisetti, S., Trehan, V., y Jayavel, K. (2021, 8). A fully integrated violence detection system using cnn and lstm. *International Journal of*

Electrical and Computer Engineering, *11*, 3374-3380. doi: 10.11591/IJECE.V11I4 .PP3374-3380

Simonyan, K., y Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition.* Descargado de `http://www.robots.ox.ac.uk/`

Sudhakaran, S., y Lanz, O. (2017, 10). Learning to detect violent videos using convolutional long short-term memory. *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017*. doi: 10.1109/AVSS.2017.8078468

Szegedy, C., Liu, W., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., ... Rabinovich, A. (2014). *Going deeper with convolutions.*

Tan, M., y Le, Q. V. (2020). *Efficientnet: Rethinking model scaling for convolutional neural networks.* Descargado de `https://arxiv.org/pdf/1905.11946.pdf`

Tashin Ahmed, N. H. S. (2020). *Classification and understanding of cloud structures via satellite images with efficientunet.* Descargado de `https://www.researchgate.net/figure/Architecture-of-EfficientNet-B0-with-MBConv-as-Basic-building-blocks_fig4_344410350` ([Online; accessed December 05, 2021])

Wenjin Taoa, Md Al-Aminb, Haodong Chena, Ming C. Leua, Zhaozheng Yinc, Ruwen Qinb. (2020). *Real-time assembly operation recognition with fog computing and transfer learning for human-centered intelligent manufacturing.* Descargado de `https://www.researchgate.net/figure/The-architecture-of-our-transfer-learning-model_fig4_342400905` ([Online; accessed December 05, 2021])

Yani, M., Irawan, B., y Setiningsih, C. (2019, 5). Application of transfer learning using convolutional neural network method for early detection of terry's nail. *Journal of Physics: Conference Series*, *1201*. Descargado de `https://www.researchgate.net/publication/333593451_Application_of_Transfer_Learning_Using_Convolutional_Neural_Network_Method_for_Early_Detection_of_Terry's_Nail` doi: 10.1088/1742-6596/1201/1/012052

A. Apendices