Preprocesamiento guiado por luminosidad para la detección automática de armas blancas en video vigilancia con Deep Learning

Alberto Castillo, Siham Tabik , Francisco Pérez, Roberto Olmos, and Francisco Herrera
*Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain.
Email: albertocl@decsai.ugr.es, siham@ugr.es, fperezhernandez@ugr.es, herrera@decsai.ugr.es

***Resumen*—La detección automática de armas blancas empuñadas por una o varias personas presentes en los vídeos de vigilancia pueden ayudar a reducir los delitos. Sin embargo, la detección de este tipo de objetos en vídeos se enfrenta a varios problemas, tales como la producida por la variabilidad de la luz ambiental junto con la la reflectancia de la superficie de las armas blancas. El objetivo de este trabajo es doble: i) Elaborar un modelo de detección automático de armas blancas para la videovigilancia mediante redes neuronales convolucionales (CNN) y ii) reforzar su robustez frente a diferentes condiciones lumínicas proponiendo una metodología de preprocesamiento guiado por luminosidad llamada DaCoLT (*Darkening and Contrast at Learning and Test stages*) para abordar condiciones de luminosidad perjudiciales.**

I. Introducción

Según la Organización Mundial de la Salud [1], cada año mueren más de 15,000 personas en crímenes violentos. Alrededor del 40 % por ciento de estos homicidios se cometen con navajas y armas blancas punzantes. En la videovigilancia, los agentes de seguridad tienen que detectar visualmente la presencia de armas en escenas monitorizadas y tomar decisiones en muy poco tiempo. Una de las soluciones más efectivas ante este problema es equipar las cámaras de vigilancia con un sistema automático y preciso de detección de armas blancas.

La mayoría de los estudios previos abordaron la detección de armas en rayos X, imágenes milimétricas o RGB utilizando métodos clásicos de aprendizaje automático [5], [6], [15], [16], [17]. Actualmente, los modelos de detección de objetos más precisos son basado en técnicas de Deep Learning, particularmente modelos basados en CNNs. El primer trabajo para abordar la detección de armas en vídeos utilizando CNNs fue [11]. Este trabajo se centró en las pistolas y fue evaluado en vídeos de películas de los noventa.

Por lo que sabemos, el presente estudio es el primero en desarrollar un sistema de detección de armas blancas usando Deep Learning y abordando el problema de la luminosidad aplicado en vídeos de vigilancia grabados en escenarios interiores. La detección de armas blancas en los vídeos de vigilancia en escenas de interior afronta varios desafíos:

- Las armas blancas pueden manejarse de diferentes formas y una gran parte del arma puede ser ocluida. Además, las armas blancas comunes, como los cuchillos, son pequeñas y la distancia entre el cuchillo y la cámara puede ser grande, lo que hace que la detección sea más difícil.
- El proceso de diseño de un nuevo conjunto de datos para entrenar con éxito el modelo de detección es manual y lleva mucho tiempo.
- La detección es sensible a la luz ambiental ya que, en general, las armas blancas, como los cuchillos, tienen superficies reflectantes.

Nos enfocamos en detectar con precisión los tipos más usados de armas blancas en delitos. Construimos un nuevo conjunto de datos que permite que el modelo aprenda con éxito las características distintivas de las armas blancas. A continuación, desarrollamos un modelo de detección de armas blancas apropiado para escenarios interiores y bajo diferentes condiciones de luz. Estudiamos las condiciones de luminosidad que afectan al rendimiento de la detección y proponemos una nueva metodología de preprocesamiento para solucionar los problemas de luminosidad.

Las principales contribuciones de este trabajo pueden resumirse como sigue:

- Construir una nueva base de datos etiquetada para detección de armas blancas, guiado por el proceso de clasificación .
- Analizar la mejor combinación de clasificadores basados en CNN y técnicas de selección de región para la detección automática de armas blancas en vídeos de vigilancia en escenarios interiores.
- Proponer una nueva metodología de preprocesamiento guiada por luminosidad, llamada Darkening and Contrast at Learning and Test time (DaCoLT), para superar las condiciones de luminosidad perjudiciales.

Nuestro estudio experimental muestra que el modelo de detección más preciso entrenado en nuestra nueva base de datos es R-FCN basado en ResNet-101, proporciona una medida F1 de 93 %. El F1 obtenido por el modelo en diferentes condiciones de luminosidad empeoraron hasta en un 15 % y usando nuestra metodología DaCoLT lo redujimos al 3 %.

Este documento está organizado de la siguiente manera. La sección II da un breve análisis de los estudios de investigación

[1] http://www.euro.who.int/__data/assets/pdf_file/0012/121314/E94277.pdf

más relacionados. La sección III describe el procedimiento para construir nuestra nueva base de datos de calidad para detección. La sección IV selecciona el modelo más adecuada para su uso como sistema de detección automática. Sección V analiza el rendimiento de detección en diferentes condiciones de luminosidad y propone la metodología DaCoLT. Finalmente, las conclusiones se resumen en la sección VI.

II. Trabajos relacionados

El problema de detectar un cuchillo empuñado por una persona en videovigilancia está estrechamente relacionado con (i) la detección de objetos pequeños en imágenes y ii) detección general de objetos mediante modelos de aprendizaje profundo.

El área tradicional de detección de armas en imágenes ha utilizado a menudo métodos clásicos supervisados de aprendizaje automático que requieren un alto nivel de supervisión humana, por ejemplo, FAST [2], SIFT [5], AAM [6], Harris [15]. Los medios utilizados son principalmente rayos X o imágenes milimétricas [16], [17] para armas ocultas y RGB para armas visibles [2], [6], [7]. En general, estos métodos proporcionan buenas precisiones pero sufren de varias limitaciones, son invasivas, necesitan costosas sistemas de detección de metales [5] como los sistemas utilizados en el acceso al aeropuerto, no puede detectar múltiples armas [9], [15] y son lentos para usar en sistemas de detección en tiempo real [2].

Los modelos de detección de objetos de última generación se basan en Convolutional Neural Networks y muestran resultados prometedores en los dos desafíos de detección más prestigiosos. El modelo de detección más preciso de ILSVRC 2017 (Large Scale Visual Recognition Challenge) [4] alcanzó una precisión media de alrededor del 73 % [2] en un benchmark de 527892 imágenes dispuestas en 200 clases de objetos, con un promedio de 2500 imágenes por clase. El modelo de detección más preciso en el benchmark de detección de 80 objetos comunes Common Objects in Context (COCO) [10] también alcanzó una precisión media de alrededor del 73 %. Los rendimientos más altos en COCO tiene una precisión del 60 % y un recall del 80 % pero fueron obtenidos en objetos grandes, y de menor rendimiento con una precisión del 30 % y un recall del 50 % en objetos pequeños [3].

Por lo que sabemos, el primer sistema automático de detección de armas de fuego basado en Deep Learning fue [11]. Este trabajo demostró ser preciso en las películas (descargadas de YouTube) con mejor calidad, es decir, mejor resolución, contraste y luminosidad que los vídeos comunes de vigilancia. Los mejores resultados reportados en este trabajo fueron obtenidos por el modelo de detección Faster R-CNN [12] basado en VGGNet [14] y con una velocidad de cinco fotogramas por segundo (fps), que es una tasa baja para un sistema de tiempo real.

[2]http://image-net.org/challenges/LSVRC/2017/
[3]http://cocodataset.org/#detections-leaderboard

III. Procedimiento de construcción de la base de datos para la detección de armas blancas

Nuestro objetivo es construir una base de datos que permita al modelo de detección distinguir con precisión entre cuchillos y todos los objetos que puedan confundirse con cuchillos. Con este fin, primero comenzamos con un conjunto de datos de clasificación inicial, Database-1, y lo ampliamos progresivamente con nuevas clases de objetos para mejorar el número de true positives (#TP), falses positives (#FP), true negatives (#TN) y false negatives (#FN) producidos por un modelo de clasificación simple (VGG-16). Este análisis nos permite entender qué objetos son críticos en el proceso de aprendizaje y considerarlos como objetos en el fondo de las imágenes de la base de datos a la hora de construir la base de datos final para detección.

Extendimos la base de datos en tres pasos:

- Database-1 incluye 2 clases, la clase de cuchillos contiene imágenes de cuchillos de diversos tamaños y la otra clase con diversos fondos.
- Database-2 contiene 28 clases e incluye nuevas clases de objetos que a menudo están presentes como fondo en la clase cuchillos de Database-1.
- Database-3 incluye clases de objetos que pueden manejarse de forma similar a un cuchillo, por ejemplo, bolígrafo, o teléfono móvil, ver cuatro ejemplos en la figura 1.

Las imágenes utilizadas para construir la Database-1, -2 y -3 fueron descargadas de diversos sitios web. Las características de las tres bases de datos auxiliares, Database-1, 2 y 3, se muestran en la Tabla I

Para evaluar el rendimiento de la clasificación y detección sobre las bases de datos propuestas, hemos construido dos conjuntos de pruebas, Test-clas y Test-det.

- Test-clas se utiliza para evaluar el modelo de clasificación, consta de 512 imágenes, 260 imágenes contienen la clase cuchillo y 252 imágenes contienen otras clases de objetos.
- El Test-det se utiliza para evaluar los modelos de detección, contiene 388 imágenes, 378 contienen al menos un cuchillo. Test-det incluye fotogramas tomados por una cámara IP de videovigilancia (Hikvision DS-2CD2420F-IW 1080p para vídeo, ratio de frames 30 fps, campo de visión 95^o y compresión MJPEG).

Figura 1: Imágenes de ejemplo de cuatro clases de objetos de la base de datos 3, (a) clase cuchillo, (b) clase bolígrafo, (c) clase teléfono móvil y (d) clase cigarrillos.

(a) (b) (c) (d)

Usamos Keras API 2.0.4 [3] para los experimentos. Medimos el rendimiento, precisión, recall, y F1, obtenidos por el

Tabla I: Características de las bases de datos.

Database-	clases	total img	img arma	otras img	enfoque
1	2	1654	598	1056	clasificación
2	28	5538	598	4940	clasificación
3	100	10039	618	9421	clasificación
4	1	1250	1250	-	detección
Test-clas	-	512	260	252	clasificación
Test-det	-	388	378	10	detección

Tabla II: Resultados del modelo de clasificación para la clase cuchillo.

Database-	#TP	#FN	#TN	#FP	Precisión	Recall	F1 score
1	181	79	174	78	69,88 %	69,62 %	69,75 %
2	209	51	228	24	89,70 %	80,38 %	84,78 %
3	213	47	228	24	**89,87 %**	**81,92 %**	**85,71 %**

modelo de clasificación cuando se entrena en Database-1, -2 y -3 se muestra en la Tabla II. El rendimiento de la clase de cuchillo ha aumentado al ampliar el conjunto de datos con más clases de objetos. El mejor rendimiento se obtiene cuando el modelo es entrenado en Database-3, pero no puede ser usada directamente para entrenar el modelo de detección ya que el detector requiere una estrategia de anotación diferente.

Como paso final, construimos el conjunto de entrenamiento, Database-4, teniendo en cuenta todas las clases de objetos, de Database-1,-2 y -3, que mejoran el aprendizaje porque se manejan de la misma manera que un cuchillo o tienen características similares a las de un cuchillo. A diferencia de la clasificación de imágenes, el proceso de anotación para la detección requiere indicar la clase de objeto utilizando un cuadro delimitador. Consideramos dos clases, el cuchillo como la clase verdadera y el resto de objetos como fondo. Incluimos imágenes de i) armas blancas de diversos tipos, formas, colores, tamaños y hechos de diferentes materiales ii) cuchillos ubicados cerca y lejos de la cámara, iii) cuchillos ocluidos parcialmente por la mano, iv) objetos que pueden ser empuñados de la misma manera que los cuchillos y v) imágenes capturadas en escenarios de interior y exterior, conjunto de datos de 1250 imágenes. Figura 2 muestra ejemplos de Database-4.

Las imágenes utilizadas para construir esta base de datos fueron descargadas de Internet, algunos fotogramas fueron extraídos de vídeos de Youtube y vídeos de vigilancia. En el resto del trabajo usaremos Database-4 para entrenar el modelo de detección.

Figura 2: Imágenes de ejemplo de Database-4. Estas imágenes muestran un contexto más rico.

IV. Análisis del enfoque de Deep Learning para la detección de armas blancas

En esta sección, analizamos el rendimiento de varias combinaciones de los modelos de clasificación más avanzados y algoritmos de selección de regiones con el objetivo de encontrar el mejor modelo de detección para la videovigilancia. En particular, analizamos estas combinaciones:

- SSD basado en Inception-v2
- R-FCN basado en ResNet101
- Faster R-CNN basado en: Inception-ResNetV2, ResNet50, ResNet101 y Inception-V2

Todos los modelos de clasificación y detección se construyeron utilizando TensorFlow [1]. Para evaluar la detección usamos Tensorflow Object Detection API [8]. Todos los experimentos fueron llevados a cabo en una GPU NVIDIA Titan Xp.

Todos los modelos de detección fueron inicializados usando los pesos pre-entrenados en el conjunto de datos COCO integrado por más de 200.000 imágenes etiquetadas. Utilizamos fine-tuning mediante el entrenamiento de las dos últimas capas completamente conectadas de la red. El proceso de entrenamiento dura de tres a cuatro horas.

Tabla III: Analisis comparativo de los modelos de detección del estado del arte.

Detector	modelo base CNN	#TP	#FP	Precisión	Recall	F1	fps
Faster R-CNN	Inception-ResNetV2	345	0	100 %	91,27 %	**95,44 %**	1,3
Faster R-CNN	ResNet101	332	8	97,65 %	89,73 %	93,52 %	4,8
Faster R-CNN	Inception-V2	329	3	99,1 %	87,04 %	92,64 %	12,8
Faster R-CNN	ResNet50	326	2	99,39 %	86,24 %	92,35 %	4,4
R-FCN	ResNet101	335	0	100 %	88,62 %	93,97 %	10
SSD	InceptionV2	245	0	100 %	64,81 %	78,65 %	20,4

El rendimiento de los modelos de detección se mide en términos de true positives, false positives, precision, recall, F1 y tasa de tiempo de inferencia (frames per second). El entrenamiento y el test se llevaron a cabo en Database-4 y test-det respectivamente. En general, los modelos de detección logran un alto rendimiento como se puede ver en la Tabla III. Esto se explica por el hecho de que el aprendizaje transferido de COCO ha sido muy beneficioso para el proceso de aprendizaje, ya que COCO incluye la clase cuchillos compuesta por unas 8.500 imágenes. Al centrarnos en la videovigilancia, la detección debe ser precisa y rápida al mismo tiempo. Por lo tanto, seleccionamos R-FCN_ResNet101 para construir nuestro detector de armas blancas. Utilizando 100 regiones de interés R-FCN_ResNet101 logra una buena precisión $100\,\%$, recall $88,62\,\%$ y F1 $93,97\,\%$, lo que está cerca del mejor modelo y proporciona una tasa razonable de inferencia.

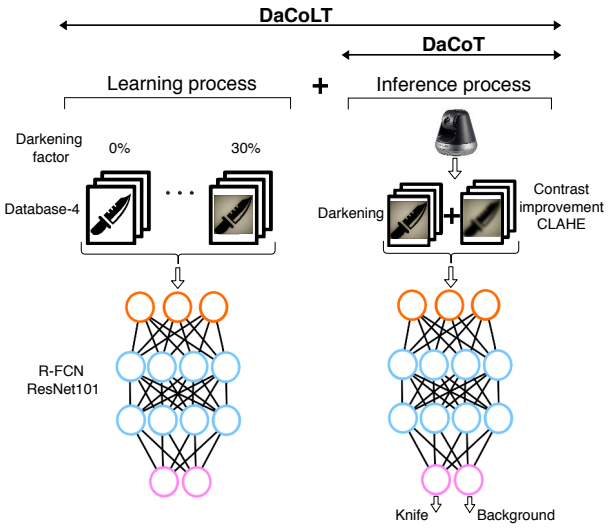
Todo el proceso de detección utiliza R-FCN_ResNet101 en una resolución por fotograma de Full HD, 1920×1080-píxeles, que es dos veces más rápido que el detector de pistola propuesto en [11]. Esto permite que el detector de armas blancas pueda ser usado en tiempo real para la videovigilancia.

V. Preprocesamiento guiado por luminosidad: metodología DaCoLT

Para resolver los problemas de luminosidad, proponemos dos alternativas:

- DaCoT: durante el proceso de prueba, los fotogramas con luminosidad alta son oscurecidos por un factor específico, luego su contraste es mejorado usando el algoritmo CLAHE.
- DaCoLT: durante el proceso de aprendizaje, el modelo de detección se entrena utilizando una técnica específica de *d*ata augmentation para oscurecimiento. Luego, DaCoT se aplica durante el tiempo de prueba. La diferencia entre estos dos procedimientos se ilustra en la figura 3.

Figura 3: Una ilustración de nuestro procedimiento, DaCoT se aplicó en el tiempo de test y DaCoLT se aplicó tanto en el tiempo de aprendizaje como en el tiempo de test.

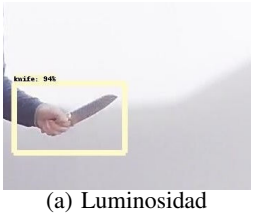


En particular, primero determinamos el rango de luminosidad que afecta la calidad de la detección (sección V-A), a continuación, se analiza y mejorar la detección simulando las condiciones ideales de luminosidad en test(sección V-B) y tanto en tiempo de aprendizaje como test(sección V-C).

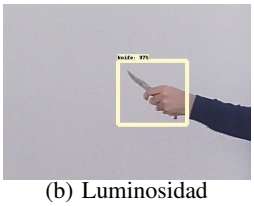
V-A. Análisis del impacto de la luminosidad en el redimiento de detección

A continuación, analizamos el impacto de las condiciones de luminosidad en el rendimiento del modelo de detección R-FCN_ResNet101. Usamos doce vídeos de prueba grabados con una cámara de seguridad IP, Samsung SNH-V6410PN de resolución 1080p, frame rate 30fps y amplitud de vista 96.1^o. Los vídeos de prueba se dividen en cuatro grupos de diferentes condiciones de luminosidad, luminosidad alta, luminosidad media, luminosidad baja y luminosidad artificial. Para una comparación justa, todos los vídeos muestran a la misma persona repitiendo las mismas acciones al mismo tiempo y distancia de la cámara. Todos los vídeos fueron grabados usando la misma cámara en la misma escena interior. Los vídeos de prueba incluyen tres cuchillos comunes con diferentes tamaños, pequeño, mediano y grande. Pequeño, mediano y grande se refieren a la proporción de la parte no ocluida del arma. Ver ejemplos en la figura 4. Los vídeos de test se pueden encontrar a través de este repositorio en github [4].

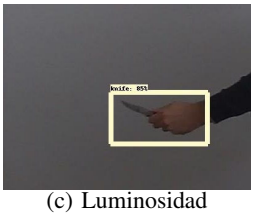
Figura 4: Resultados de detección en cuatro condiciones de luminosidad diferentes.



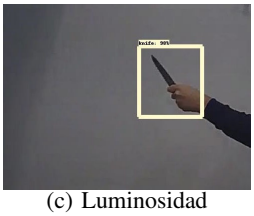
(a) Luminosidad alta



(b) Luminosidad media



(c) Luminosidad baja



(c) Luminosidad artificial

Consideramos que un cuchillo es un ground truth cuando es reconocible por el ojo humano. Los resultados en términos de número total de Ground Truth positivos #GT_P, #TP, #FP, precisión, recall, y F1 en cada video de prueba se muestran en la Tabla IV. Consideramos una detección como TP si el solapamiento entre el área del cuchillo manipulada en el fotograma y la caja delimitadora predicha es mayor que 70 %.

Tabla IV: Rendimiento de detección obtenido en vídeos grabados en diferentes condiciones de luminosidad.

Luminosidad	Tamaño arma	#frames	#GT_P	#TP	#FP	Precisión	Recall	F1
Alta	Grande	121	112	78	0	100 %	69,64 %	82,10 %
	Mediano	107	90	44	0	100 %	48,89 %	65,67 %
	Pequeño	137	103	53	0	100 %	51,46 %	67,95 %
				Promedio		100 %	56,66 %	71,91 %
Media	Grande	109	98	85	0	100 %	86,73 %	92,89 %
	Mediano	116	98	73	0	100 %	74,49 %	85,38 %
	Pequeño	138	110	64	0	100 %	58,18 %	7356 %
				Promedio		100 %	73,13 %	83,94 %
Baja	Grande	126	114	104	1	99,05 %	92,04 %	95,41 %
	Mediano	114	100	70	0	100 %	70 %	82,35 %
	Pequeño	138	101	74	0	100 %	73,27 %	84,57 %
				Promedio		99,68 %	78,44 %	87,44 %
Artificial	Grande	119	110	95	0	100 %	86,36 %	92,68 %
	Mediano	113	99	75	3	96,15 %	78,13 %	86,21 %
	Pequeño	96	90	65	4	94,20 %	75,58 %	83,87 %
				Promedio		96,78 %	**80,02 %**	**87,59 %**

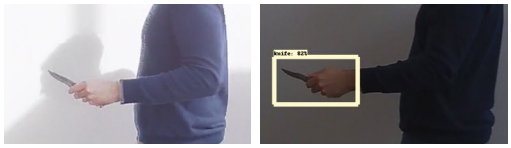
Como se puede observar en la Tabla IV, el rendimiento del modelo de detección es inestable en un escenario de cambio de luminosidad. El peor rendimiento se obtiene en condiciones de alta luminosidad, y el mejor rendimiento con luminosidad artificial. De las condiciones de luminosidad más bajas a las

[4]https://github.com/alcasla/Automatic-Cold-Steel-Detection-Alarm

más altas, el recall promedio disminuyó de 80,02 % a 56,66 %, y el F1 promedio de 87,59 % a 71,91 %.

La Figura 5 muestra un ejemplo de los resultados de detección de escenas muy similares, es decir, la misma pose y contexto, diferentes niveles de luminosidad y contraste, pero diferentes resultados de detección.

Figura 5: Un ejemplo del resultado de la detección en dos situaciones similares con diferentes condiciones de luminosidad.



Luminosidad alta Luminosidad baja

V-B. Darkening and Contrast at Test stage (DaCoT)

Para resolver la inestabilidad del modelo de detección en condiciones de luminosidad variable, primero analizamos el procedimiento llamado DaCoT, Oscurecimiento y Contraste en tiempo de test, el cual simula la condición de luminosidad que produce el mejor desempeño, luminosidad baja y alto contraste. Procedemos de la siguiente manera:

- Primero comprobamos el nivel de luminosidad de cada fotograma. Si el nivel de luminosidad es de medio a alto, oscureceremos el fotograma multiplicando los valores de los píxeles por el factor de oscurecimiento correspondiente. Este factor se calcula en base a la diferencia entre el nivel de luminosidad ideal y el nivel de luminosidad actual de la imagen.
- A continuación, aumentamos el contraste del fotograma obtenido mediante el algoritmo *Contrast-Limited Adaptive Histogram Equalization*(CLAHE) [13].
- A continuación, el fotograma se introduce al modelo de detección para inferencia.

La evaluación del enfoque propuesto en condiciones de luminosidad alta al considerar diferentes factores de oscurecimiento se proporciona en la Tabla V. El rendimiento del modelo de detección ha mejorado cuando se utiliza un factor de oscurecimiento del 30 %. En promedio, con un factor de oscurecimiento de 30 % el recall y F1 han mejorado en 6,53 % y 5,07 % respectivamente en comparación con la condición original de luminosidad alta.

El preprocesamiento propuesto, oscurecimiento más CLAHE, tarda alrededor de 29 ± 3 ms por fotograma en la CPU, lo que no ralentiza el proceso general de detección, ya que esta tarea de preprocesamiento se realiza en paralelo con la tarea de detección en la GPU. Es decir, la hebra de preprocesamiento se ejecuta en la CPU y la de detección se ejecuta en la GPU.

V-C. Darkening and Contrast at Learning and Test stages (DaCoLT)

Del análisis anterior, encontramos que el DaCoT mejora el rendimiento del modelo de detección bajo condiciones de luminosidad alta. En esta sección, analizamos el uso de DaCoLT, que es una extensión del DaCoT, mediante la aplicación de diferentes niveles de oscurecimiento no sólo en la etapa de prueba sino también durante la etapa de aprendizaje del modelo de detección. El método de aumento de datos de oscurecimiento consiste en oscurecer imágenes de entrenamiento individuales seleccionando aleatoriamente un factor de oscurecimiento en el rango [0 % 30 %].

Tabla V: Los resultados de los fotogramas de vídeo grabados originalmente en condiciones de luminosidad alta(es decir, en el peor de los casos) al aplicar DaCoT.

Factor oscurecimiento	Tamaño arma	#frames	#GT_P	#TP	#FP	Precisión	Recall	F1
original luminosidad alta	Grande	121	112	78	0	100 %	69,64 %	82,11 %
	Mediano	107	90	44	0	100 %	48,89 %	65,67 %
	Pequeño	137	103	53	0	100 %	51,46 %	67,95 %
				Promedio		100 %	56,66 %	71,91 %
10 %	Grande	121	112	81	0	100 %	72,32 %	83,94 %
	Mediano	107	90	52	0	100 %	57,78 %	73,24 %
	Pequeño	137	103	57	1	98,28 %	55,34 %	71,25 %
				Promedio		99,43 %	61,81 %	76,14 %
20 %	Grande	121	112	83	0	100 %	74,11 %	85,13 %
	Mediano	107	90	55	0	100 %	61,11 %	75,86 %
	Pequeño	137	103	53	0	100 %	51,46 %	67,95 %
				Promedio		100 %	62,23 %	76,31 %
30 %	Grande	121	112	85	0	100 %	75,89 %	86,29 %
	Mediano	107	90	56	0	100 %	62,22 %	76,71 %
	Pequeño	137	103	53	0	100 %	51,46 %	67,95 %
				Promedio		100 %	**63,19 %**	**76,98 %**
40 %	Grande	121	112	80	0	100 %	71,43 %	83,33 %
	Mediano	107	90	52	0	100 %	57,78 %	62,60 %
	Pequeño	137	103	51	0	100 %	49,51 %	66,23 %
				Promedio		100 %	59,57 %	70,72 %
50 %	Grande	121	112	78	0	100 %	69,64 %	82,11 %
	Mediano	107	90	41	0	100 %	45,56 %	65,67 %
	Pequeño	137	103	50	0	100 %	48,54 %	65,36 %
				Promedio		100 %	54,58 %	71,04 %

Tabla VI: Los resultados al aplicar DaCoT y DaCoLT en vídeos grabados originalmente bajo condiciones de luminosidad alta usando diferentes tamaños de cuchillos, grande, mediano y pequeño.

	Tamaño arma	#frames	#GT_P	#TP	#FP	Precisión	Recall	F1
original luminosidad alta	Grande	121	112	78	0	100 %	69,64 %	82,11 %
	Mediano	107	90	44	0	100 %	48,89 %	65,67 %
	Pequeño	137	103	53	0	100 %	51,46 %	67,95 %
				Promedio		100 %	56,66 %	71,91 %
guiado lumino. DaCoT (Test)	Grande	121	112	85	0	100 %	75,89 %	86,29 %
	Mediano	107	90	56	0	100 %	62,22 %	76,71 %
	Pequeño	137	103	53	0	100 %	51,46 %	67,95 %
				Promedio		100 %	63,19 %	76,98 %
guiado lumino. DaCoLT (Train+Test)	Grande	121	112	84	0	100 %	75 %	85,71 %
	Mediano	107	90	64	0	100 %	71,11 %	83,12 %
	Pequeño	137	103	74	0	100 %	71,84 %	83,61 %
				Promedio		100 %	**72,65 %**	**84,15 %**

La Tabla VI muestra el impacto al aplicar DaCoLT en el rendimiento de detección para las peores condiciones de luminosidad. La primera parte muestra los resultados del modelo de

detección en vídeos filmados originalmente en condiciones de luminosidad alta y utilizando diferentes tamaños de cuchillos, grandes, medianas y pequeñas. La segunda parte muestra el efecto de aplicar el enfoque de preprocesamiento propuesto en la etapa de inferencia, DaCoT. La tercera parte muestra el efecto de la metodología DaCoLT propuesta durante las etapas de inferencia y aprendizaje. De esta tabla podemos ver que el paso de data augmentation incluido en DACOLT mejora el aprendizaje del modelo de detección bajo condiciones de luminosidad alta. El recall y F1 promedio han mejorado respectivamente en $9{,}46\,\%$ y $7{,}17\,\%$ en comparación con el rendimiento considerando sólo el preprocesamiento en el momento de la inferencia, DaCoT.

Aplicando el preprocesamiento de luminosidad durante los pasos de inferencia y aprendizaje en los vídeos filmados bajo condiciones de luminosidad alta, se mejora el recall en $15{,}99\,\%$ y el F1 en $12{,}24\,\%$ en comparación con las condiciones de luminosidad alta originales.

Como estudio final, mostramos en la Tabla VII los resultados al aplicar la metodología de preprocesamiento DaCoLT en vídeos filmados bajo diferentes condiciones de luminosidad.

Tabla VII: El efecto de aplicar la metodología DaCoLT en vídeos filmados originalmente bajo diferentes condiciones de luminosidad.

Luminosidad	Tam arma	#Frames	#GT_P	#TP	#FP	Precisión	Recall	F1	orig. F1
Alta	Grande	121	112	84	0	100 %	75,00 %	85,71 %	82,10 %
	Mediano	107	90	64	0	100 %	71,11 %	83,12 %	65,67 %
	Pequeño	137	103	74	0	100 %	71,84 %	83,61 %	67,95 %
			Promedio			100 %	72,65 %	84,15 %	71,91 %
Media	Grande	109	98	84	0	100 %	85,71 %	92,31 %	92,89 %
	Mediano	116	98	78	0	100 %	79,59 %	88,64 %	85,38 %
	Pequeño	138	110	75	0	100 %	68,18 %	81,08 %	73,56 %
			Promedio			100 %	77,83 %	87,34 %	83,94 %
Baja	Grande	126	114	103	0	100 %	90,35 %	94,93 %	95,41 %
	Mediano	114	100	74	0	100 %	74,00 %	85,06 %	82,35 %
	Pequeño	138	101	72	0	100 %	71,29 %	83,24 %	84,57 %
			Promedio			100 %	**78,55 %**	**87,74 %**	87,44 %
Artificial	Grande	119	110	95	0	100 %	86,36 %	92,68 %	92,68 %
	Mediano	113	99	73	1	98,65 %	74,49 %	84,88 %	86,21 %
	Pequeño	96	90	63	1	98,44 %	70,79 %	82,36 %	83,87 %
			Promedio			99,03 %	77,21 %	86,64 %	87,59 %

Como se observa, DaCoLT mejora la detección especialmente en las peores condiciones (luminosidad más alta). En otras palabras, DaCoLT permite alcanzar precisiones similares en los vídeos independientemente de su nivel de luminosidad.

VI. Conclusiones y trabajo futuro

Este trabajo presenta un modelo de detección automática de armas blancas para videovigilancia basado en una nueva métodología de preprocesamiento guiado por luminosidad, denominado DaCoLT, que mejora la calidad de la detección. El modelo de detección obtenido muestra un alto potencial incluso en vídeos de baja calidad.

Nuestro sistema de detección de armas blancas puede ser utilizado en varias aplicaciones, por ejemplo, i) detección en tiempo real de armas blancas en videovigilancia y ii) control parental de vídeos o imágenes con contenido violento.

Como trabajo futuro, abordaremos la detección de armas en escenarios al aire libre, donde pueden estar presentes objetos en movimiento y las condiciones climáticas adversas pueden aumentar la dificultad de la detección.

Agradecimientos

Este trabajo contó con el apoyo del Ministerio de Ciencia y Tecnología de España bajo el proyecto TIN2017-89517-P. Siham Tabik contó con el apoyo del Programa Ramón y Cajal (RYC-2015-18136). La GPU Titan X Pascal utilizada para esta investigación fue donado por NVIDIA Corporation.

Referencias

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. *Operating Systems Design and Implementation*, 16:265–283, 2016.

[2] Himanshu Buckchash and Balasubramanian Raman. A robust object detector: Application to detection of visual knives. *IEEE Multimedia and Expo Workshops*, pages 633–638, July 2017.

[3] François Chollet. Keras: Theano-based deep learning library. *Code: github. com/fchollet. Documentation: http://keras.io*, 2015.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[5] Greg Flitton, Toby P Breckon, and Najla Megherbi. A comparison of 3d interest point descriptors with application to airport baggage object detection in complex ct imagery. *Pattern Recognition*, 46(9):2420–2436, 2013.

[6] Andrzej Glowacz, Marcin Kmieć, and Andrzej Dziech. Visual detection of knives in security applications using active appearance models. *Multimedia Tools and Applications*, 74(12):4253–4267, Jun 2015.

[7] Michał Grega, Andrzej Matiolański, Piotr Guzik, and Mikołaj Leszczuk. Automated detection of firearms and knives in a cctv image. *Sensors, dx.doi.org/10.3390/s16010047*, 16(1), 2016.

[8] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Zbigniew Fischer, Ianand Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Tensorflow object detection api. *Code: github.com/tensorflow/models/tree/master/object_detection*, CVPR 2017 (developing).

[9] Marcin Kmieć and Andrzej Glowacz. Object detection in security applications using dominant edge directions. *Pattern Recognition Letters*, 52:72 – 79, 2015.

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[11] Roberto Olmos, Siham Tabik, and Francisco Herrera. Automatic handgun detection alarm in videos using deep learning. *Neurocomputing*, 275:66–72, 2018.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[13] Ali M. Reza. Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement. *Journal of VLSI signal processing systems for signal, image and video technology*, 38(1):35–44, Aug 2004.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[15] Rohit Kumar Tiwari and Gyanendra K. Verma. A computer vision based framework for visual gun detection using harris interest point detector. *Procedia Computer Science*, 54:703–712, 2015.

[16] Ivan Uroukov and Robert Speller. A preliminary approach to intelligent x-ray imaging for baggage inspection at airports. *Signal Processing Research*, 4:1–11, January 2015.

[17] Zelong Xiao, Xuan Lu, Jiangjiang Yan, Li Wu, and Luyao Ren. Automatic detection of concealed pistols using passive millimeter wave imaging. In *2015 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–4. IEEE, 2015.