CEA Comité Español de Automática

XLV JORNADAS AUTOMÁTICA MÁLAGA 2024

UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Arquitecturas para Detección de Anomalías: Fusión de GAN, U-Net y Transformers

Pérez, B.[a,*], Resino, M.[a], García, F.[a], Al-Kaff, A.[a]

[a]*Departamento de Ingeniería Eléctrica, Electrónica y Automática, Universidad Carlos III de Madrid, c/ Madrid, 126-128. 28903 Getafe (Madrid), España*

To cite this article: Pérez, B., Resino, M., García, F., Al-Kaff, A. 2024. Anomaly Detection Architectures: Fusion of GAN, U-Net and Transformers. Jornadas de Automática, 45. https://doi.org/10.17979/ja-cea.2024.45.10917

Resumen

La detección y prevención de situaciones anómalas en entornos urbanos es crucial para la seguridad de todos los usuarios, siendo un área de estudio muy relevante actualmente. La abundancia de cámaras de seguridad en ciudades permite usar tecnologías de Inteligencia Artificial para monitorear y analizar comportamientos en tiempo real. Este estudio propone un sistema basado en la estructura GAN (*Generative Adversarial Networks*) para identificar situaciones anómalas en secuencias de imágenes. Se desarrollaron y compararon dos sistemas utilizando la arquitectura *PatchGAN*. El primero emplea la red U-Net para el generador, mientras que el segundo usa U-NetR, una variación de U-Net que mejora la contextualización de la imagen completa. Los resultados de diversos experimentos muestran la eficacia de ambos enfoques, proporcionando una comparación detallada de las ventajas y limitaciones de cada uno. Este trabajo contribuye al avance de las tecnologías de vigilancia.

Palabras clave: Visión y Programación, Machine Learning, Inteligencia Artificial

Anomaly Detection Architectures: Fusion of GAN, U-Net and Transformers

Abstract

The detection and prevention of anomalous situations in urban environments is crucial for the safety of all users, and is currently a very relevant area of study. The abundance of security cameras in cities allows the use of artificial intelligence technologies to monitor and analyze behaviors in real time. This study proposes a system based on the GAN (Generative Adversarial Networks) framework to identify anomalous situations in image sequences. Two systems were developed and compared using the *PatchGAN* architecture. The first employs the U-Net network for the generator, while the second uses U-NetR, a variation of U-Net that improves the contextualization of the entire image. Results from several experiments show the effectiveness of both approaches, providing a detailed comparison of the advantages and limitations of each. This work contributes to the advancement of surveillance technologies.

Keywords: Programming and Vision, Machine Learning, Artificial Intelligence

1. Introducción

Los grandes avances actuales de la Inteligencia Artificial han revolucionado numerosos ámbitos de la vida cotidiana, como la generación de imágenes a partir de descripciones del usuario, la creación de ritmos de música o la redacción de relatos de texto, entre otras.

Estas capacidades son aplicables a un gran rango de situaciones, entre las que vamos a destacar la seguridad urbana. Estos avances han facilitado significativamente el desarrollo de métodos automáticos para detectar anomalías en los entorno urbanos mediante sistemas inteligentes avanzados, acelerando la respuesta de los operarios correspondientes.

Estos sistemas destacan en la gestión urbana debido a que mantienen una vigilancia contínua sobre el entorno y están entrenados para detectar y alertar a las autoridades de forma

*Autor para correspondencia: boperezl@pa.uc3m.es

rápida. Además, pueden preveer situaciones anómalas antes incluso de que sucedan simplemente observando y analizando los elementos que puedan ocasionarlas.

Con este objetivo, existían proyectos más simples como (Yun et al., 2014) donde se emplea un Motion Interaction Field (MIF) para crear un mapa de gaussianas de las interacciones entre los diferentes objetos a partir de su flujo óptico. De este modo, se analiza este mapa para detectar patrones de accidentes o anomalías en la escena, tal y como se puede ver en la Figura 1

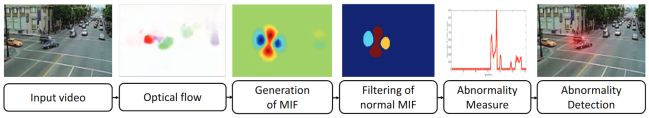

Figura 1: Arquitectura del Motion Interaction Field

Otro ejemplo es el de (Pathak et al., 2015) en el que no se analiza una sola imagen de la secuencia, sino que se divide el video en clips más cortos y se calcula la representación vectorial de las características de los objetos. A partir de estas características se obtienen los temas y se agrupan mediante *k-means*. Seguidamente, se identifican las regiones de la escena que no corresponden a estos temas clasificados y se categorizan como zonas de posibles anomalías. Finalmente estas regiones son clasificadas a partir de un modelo de árbol de decisiones entrenado con anomalías.

Siguiendo el planteamiento de emplear modelos de redes para la detección de anomalías, surgió el trabajo (Aboah, 2021). En primer lugar, analiza una secuencia del video para hacer una estimación del fondo y posteriormente se analiza este fondo mediante el modelo YOLOv5 para encontrar elementos estáticos inusuales. Finalmente, se aplica un threshold adaptativo para introducir todas las detecciones anteriores a un árbol de decisiones que identifique si es o no una anomalía.

Posteriormente, con los últimos avances en Inteligencia Artificial (IA), también surgieron nuevos planteamientos para el ámbito de detección de anomalías, como es el trabajo (Leporowski et al., 2023) en el que se presenta el método AVACA mostrado en la Figura 2, que trata de combinar información de audio y vídeo de una secuencia y emplea Transformers para la detección de anomalías. De este modo, lograron demostrar que la incorporación de audio a la información de entrada puede ser de gran utilidad para la detección de casos poco comunes.

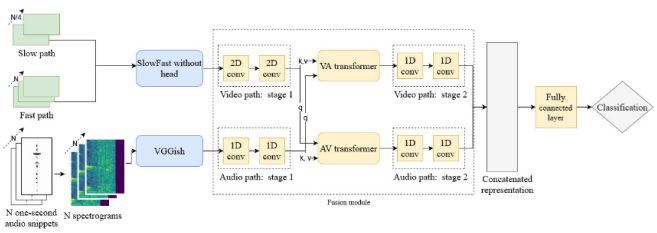

Figura 2: Arquitectura AVACA

Continuando con el uso de los sistemas Transformers, en el trabajo (Roka and Diwakar, 2023) implementaron una variante CViT (Convolutional Visual Transformer) para la detección y localización de anomalías en video. Se propone una variante de U-Net sustituyendo parte del codificador con bloques CViT apilados como se observa en la Figura 3 en la que se muestra esta fusión entre U-Net (Ronneberger et al., 2015) y Visual Transformers.

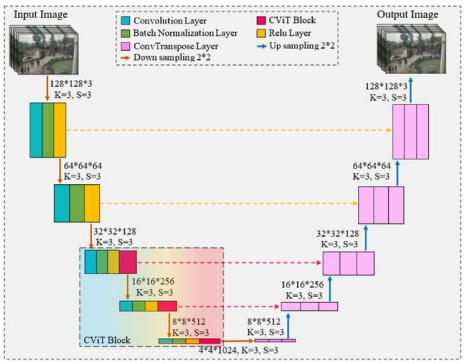

Figura 3: Arquitectura interna de la metodologías propuesta en (Ronneberger et al., 2015)

De este modo, el módulo de detección de anomalías obtiene una mayor cantidad de características locales y globales de la imagen para la detección. Seguidamente, se implementó un módulo de localización de anomalías basado en YOLO que recibe los frames en los que se ha detectado una anomalía, e identifica los objetos anómalos en la escena. Estos módulos se comunican tal y como podemos ver en la Figura 4 de la implementación del sistema completo.

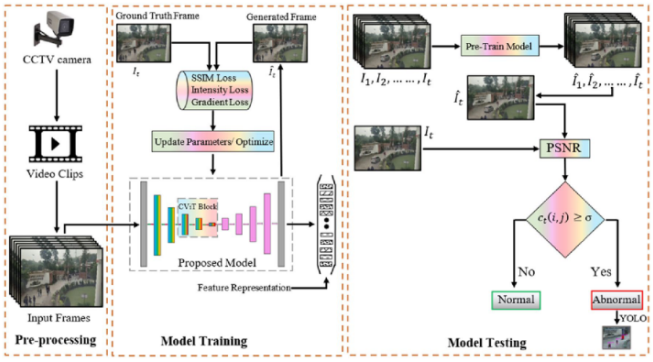

Figura 4: Pipeline implementado para la detección de anomalías mediante CViT

Por último, en el proyecto (Yuan et al., 2021) hace uso de las *Generative Adversarial Networks (GAN)* (Goodfellow et al., 2014) para el entrenamiento de una red generativa para que sea capaz de predecir los siguientes frames a partir del frame de entrada de una secuencia. Este entrenamiento enfrenta a la red generadora con otra red discriminadora, que debe aprender a discernir entre una imagen real y una generada, retroalimentandose mutuamente en el proceso de aprendizaje. De este modo, se logra mejorar la extracción de características y la generación de imágenes realistas, además de poder hacer una comparación más eficaz entre el frame generado y el real para detectar las anomalías que tienen lugar.

2. Metodología

Como se ha presentado anteriormente en la primera etapa de este trabajo, se realizó un estudio de las diferentes implementaciones existentes de Inteligencia Artificial para la detección de anomalías, aplicado al ámbito de la movilidad, debido a su velocidad de prepocesamiento y la flexibilidad de funcionamiento en cualquier situación o escenario. De este modo, se acordó el uso de las herramientas que se presentarán a continuación.

2.1. *U-Net*

U-Net es una red neuronal convolucional aplicada a la segmentación semántica que se caracteriza por su estructura como se muestra en la Figura 5. Como se puede observar, esta red se compone de una parte encoder y otra decoder con sus respectivas operaciones para la extracción de características de la imagen.

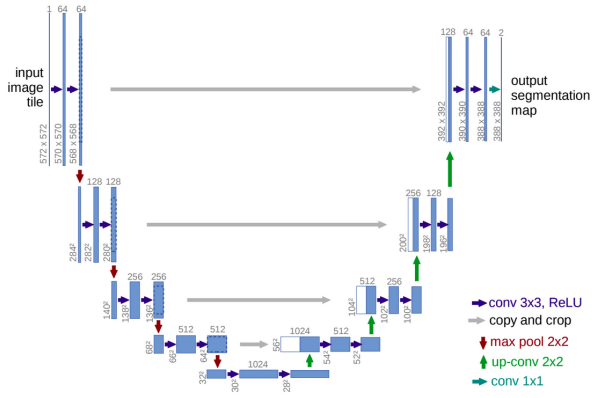

Figura 5: Arquitectura U-Net

Debido a esta estructura y las operaciones aplicadas a la imagen, es una herramienta muy útil para la extracción de características tanto locales como globales, proporcionando información más precisa para la detección de anomalías. Además, requiere de un menor tamaño de datos de entrenamiento para un funcionamiento eficaz, característica que resulta de gran interés en este trabajo puesto que existen pocos datasets etiquetados de anomalías disponibles para el entrenamiento.

2.2. *Generative Adversarial Networks (GAN)*

Las GANs son conocidas por su capacidad de generar datos sintéticos realistas, que se componen de dos redes neuronales enfrentadas: una generadora y otra discriminadora como se muestra en la Figura 6.

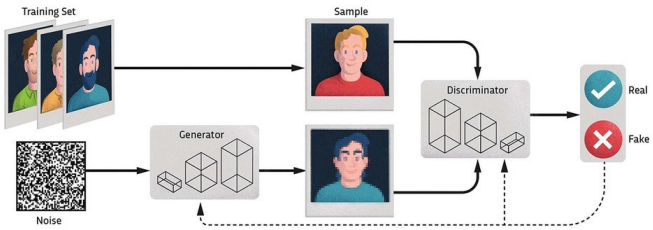

Figura 6: Estructura GAN

Debido a su capacidad de generación de imágenes, en este proyecto se adaptó el entrenamiento de estas redes enfrentadas para que aprendieran únicamente situaciones normales. De este modo, la red generadora será capaz de generar predicciones de situaciones normales. Por lo tanto, si existe una gran diferencia entre el suceso real y el generado, se tratará de una sitaución anómala, puesto que la situación real no se asemeja a una situación normal como la generada por la red a partir de la secuencia de entrada.

A partir de esta estructura GAN, han surgido diversas variantes como es el caso de la arquitectura *Patch GAN* (Isola et al., 2017). Esta variante destaca porque trabaja con las divisiones, conocidas como "*Patches*", de una imagen, en lugar de trabajar con la imagen como un elemento global. De este modo, se analizan regiones más pequeñas para permitir al discriminador centrarse en detalles más específicos, como la textura u otras características locales.
Partiendo de esta variante **Patch GAN**, para este trabajo se han implementado dos sistemas diferentes.

2.3. *U-Patch GAN*

En el primer sistema, se ha empleado una arquitectura de red U-Net Ronneberger et al. (2015) para la parte de red generadora, puesto que U-Net ha demostrado una gran eficacia en la segmentación de imágenes por su estructura codificador-decodificador simétrico. De esta forma, se aprovecha la estructura Patch-GAN para el entrenamiento de la red U-Net para la generación de imágenes, dando lugar a la variante U-Patch GAN (Fan et al., 2023). En este entrenamiento, se emplearon tan solo secuencias de situaciones comunes en un escenario teniendo como etiqueta de una instancia, la instancia siguiente. Así, la red aprendía cómo debía ser la imagen siguiente, en una situación normal, a partir de una imagen de entrada.
Finalmente, al aplicar esta red a otra secuencia de entrada, se compara la imagen generada con la imagen real para comprobar si se trata de un caso normal o anómalo. Es decir, si existe mucha diferencia entre estas imágenes, significa que la red no ha sido capaz de generar lo que ha sucedido en la escena y correspondería a un caso de anomalía.

2.4. *PatchGAN + U-NetR*

Por otro lado, se aplicó otra modificación a la estructura de la red U-Net para añadir un Visual Transformer (Dosovitskiy et al., 2021) a la parte del codificador para permitir a la red captar una mayor información global y contextual de la imagen en cada división de la misma, aprovechando el mecanismo de atención de los Transformers y su capacidad de modelado

de relaciones espaciales y contextuales. De este modo, se pretendía aplicar los mismos casos y pruebas para observar las ventajas que suponía la inclusión de esta tecnología para la generación de imágenes y detección de anomalías.

3. Experimentación

3.1. Configuración de la Experimentación

En primer lugar, los experimentos llevados a cabo durante el desarrollo de este proyecto se han ejecutado en un ordenador con una GPU Nvidia RTX 4090. Por otro lado, para la detección de anomalías, los modelos han sido entrenados únicamente con secuencias de situaciones normales, de manera que el modelo generador tan solo aprendiera a predecir situaciones normales. Seguidamente, para el test, se insertaron situaciones anómalas del mismo escenario para observar las dificultades que tiene el modelo para generarlas. En cuanto a la resolución de las imágenes, estas fueron escaladas a un tamaño de [224, 224] para la arquitectura que no empleaba transformers, y a [256,256] para la que sí. Además, los rangos de las imágenes están normalizados entre [0,1] para mejorar el aprendizaje de la red.

Por último, el entrenamiento de los modelos se ha realizado haciendo uso del optimizador Adam y una tasa de aprendizaje fijada en 0.0002 tanto para la red generadora como para la discriminadora.

3.2. Datasets Utilizados

Para la evaluación del sistema desarrollado, se han empleado tres datasets distintos: UCSD, UBnormal y una recopilación de secuencias de accidentes reales en la carretera.

En primer lugar, el dataset UCSD se puede dividir en dos subconjuntos: Ped1 y Ped2. Ped1 consta de 34 clips para entrenamiento y 36 clips para test, mientras que Ped2 tiene 16 clips para entrenar y 12 clips para test. Los videos de entrenamiento solo contienen escenas normales, mientras que el test incluye anomalías como ciclistas, coches y patinadores. También cabe destacar que el subconjunto Ped1 cuenta con una perspectiva isométrica a diferencia de Ped2, que es una vista horizontal, por lo que en Ped1 el tamaño de los elementos que la componen varía más durante la secuencia que en Ped2.

También se cuenta con el dataset UBnormal que es un benchmark para la detección supervisada de anomalías en videos en un entorno simulado. Este dataset se distingue por incluir múltiples escenas virtuales con eventos anómalos anotados a nivel de píxel durante el entrenamiento.

Finalmente, se hace uso de la recopilación de escenas anómalas extraídas de internet tan solo para la validación en distintos escenarios de los modelos entrenados.

3.3. Función de Pérdida

Por otra parte, se estuvo analizando la opción de incluir diferentes funciones de pérdida para intentar mejorar los resultados obtenidos de los diferentes modelos.

En concreto, las dos funciones de pérdida que se han tenido en cuenta, además de la propia función de pérdida de la GAN, han sido el error cuadrático medio (MSE) y una función de pérdida que hace uso del flujo óptico, comparando el flujo óptico obtenido de las imágenes reales con el flujo óptico obtenido a partir de las imágenes generadas. Estas funciones de pérdida vienen multiplicadas por un factor gamma individual que permite regular el impacto de cada una de ellas durante el entrenamiento.

3.4. Peak Signal-to-Noise Ratio (PSNR)

En cuanto a la evaluación de los modelos entrenados, se empleó la métrica de PSNR para medir la calidad de la reconstrucción de una imagen de los modelos en comparación con la imagen original correspondiente. De este modo, un PSNR más alto indica que la imagen reconstruida tiene una similitud mayor a la imagen original y por lo tanto, la reconstrucción ha sido de mejor calidad.

3.5. Arquitectura

Como se puede observar en la Tabla 1, la red U-Net base, sin el entrenamiento GAN, consigue el menor tiempo de entrenamiento y unos resultados razonables. Sin embargo, tratando un tema como es la detección de anomalías, dedicar más tiempo de entrenamiento con el fin de obtener mejores resultados puede ser decisivo en una situación de peligro. Cabe destacar que, los modelos que no hacen uso del Transformer, son entrenados con un tamaño de paquete de 16, mientras que los que sí, son entrenados con un tamaño de paquete de 4 debido a que estos requieren una mayor capacidad computacional, aumentando los tiempos de entrenamiento.

	T(min)	T/iteración(min)	PSNR máximo
U-Net	**25**	**1.25**	37.73
GAN	67	3.35	37.33
PatchGAN	58	2.9	39.17
U-NetR	110	5.5	**40.38**

Tabla 1: Tiempo de entrenamiento y PSNR máximo obtenido por modelo

Como ya se ha mencionado anteriormente, se estuvieron realizando experimentos teniendo en cuenta diferentes funciones de pérdida a la hora de entrenar. En la Figura 7 se puede apreciar cómo, en este caso específico, añadir la función del MSE a la función de perdida, incrementa positivamente los resultados obtenidos por el modelo, mientras que añadir el flujo óptico (OF) no supone ninguna mejora, si no que además, reduce los valores de PSNR.

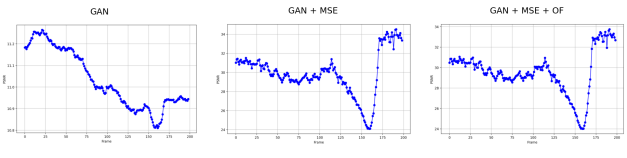

Figura 7: Modelos entrenados con diferentes funciones de pérdida

Aunque tan solo se han mostrado resultados específicos de estas pruebas en la arquitectura GAN, este comportamiento se repite también en los demás modelos.

Seguidamente, se realizaron pruebas de inferencia en diferentes secuencias de los datasets para comprobar el funcionamiento de los mejores modelos de cada arquitectura en distintas situaciones.

3.5.1. UCSD1 y UCSD2

En primer lugar, en las Figuras 8 y 9 se muestran los resultados obtenidos de inferencia en ambos datasets. Como se puede apreciar, todas las arquitecturas presentan una evolución similar del PSNR durante la secuencia, siendo fácilmente apreciable el tramo de vídeo correspondiente a la anomalía con la caída de los valores de PSNR. Por ello, dada esta similitud entre los resultados, se puede observar que el modelo de U-Net es el que obtiene los valores de PSNR más altos para ambas secuencias.

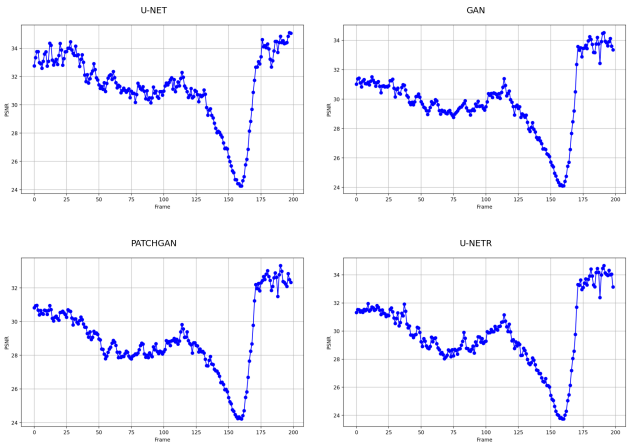

Figura 8: Inferencia de los modelos sobre el dataset UCSD 1

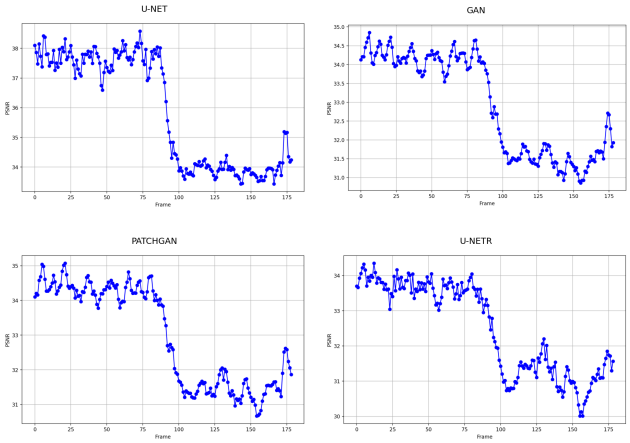

Figura 9: Inferencia de los modelos sobre el dataset UCSD 2

3.5.2. UBnormal

En este caso se comprobó que hacer uso del modelo de UCSD para realizar la inferencia sobre el dataset UBnormal no daba buenos resultados, por lo que se decidió entrenar un modelo con dicho dataset. Como se puede ver en la Figura 10, los nuevos resultados obtenidos (derecha) son mejores que los modelos entrenados fuera del dataset (izquierda), aunque también presentan cierto ruido, dejando en duda si realmente se trata de una detección de anomalía o no.

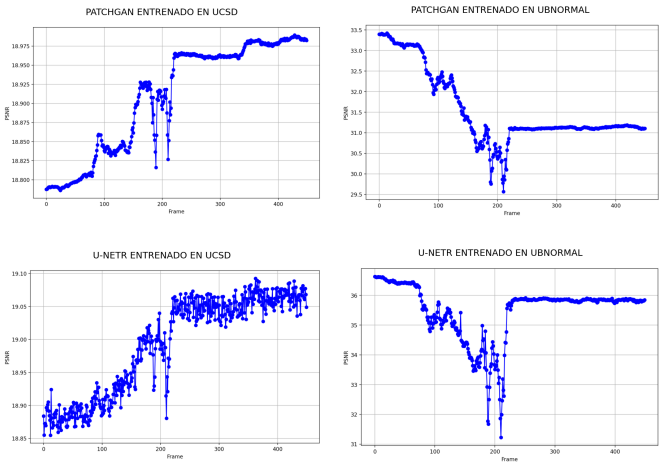

Figura 10: Inferencia de los modelos con distinto dataset de entrenamiento sobre UBnormal

3.5.3. Dataset Recopilado

En la Figura 11 se puede observar la inferencia obtenida por los cuatro mejores modelos de cada arquitectura sobre el mismo vídeo de prueba.

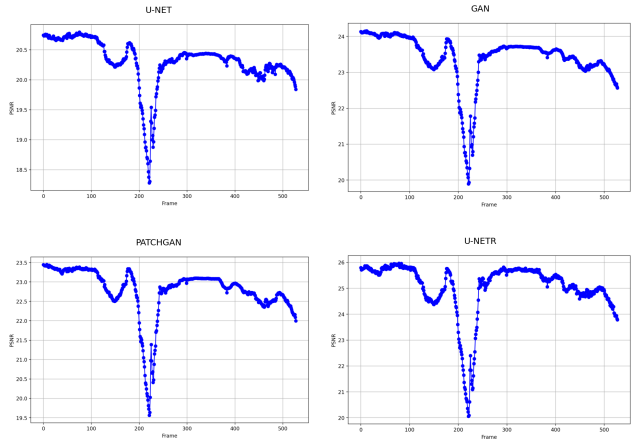

Figura 11: Inferencia de los modelos sobre el dataset recopilado

Adicionalmente, se realizó un estudio comparativo de los diferentes datasets para comprobar la diferencia o *delay* entre el instante donde se puede observar la anomalía a simple vista en el vídeo y el instante en que es detectado por el sistema. Así se obtuvieron los resultados mostardos en la Tabla 2 donde, en el caso de UCSD1, las cuatro arquitecturas detectan la anomalía en tiempos similares, a diferencia de UCSD2 en el que existe una mayor diferencia de tiempos, siendo la arquitectura *U-Net* la de mejores resultados en ambas. No obstante, para el último caso del dataset recopilado, se puede observar que la mejor arquitectura, con cierta diferencia respecto al resto, es la de *U-NetR*.

	U-Net	*GAN*	*PatchGAN*	*U-NetR*
Delay UCSD1	1	5	3	3
Delay UCSD2	0	5	5	10
Delay recopilado	9	7	6	4

Tabla 2: Inferencia y estudio de *delays* sobre los distintos datasets.

Por último, en la Figura 12 se puede ver un ejemplo de cómo funciona el sistema desarrollado, en este caso, aplicado a una secuencia del último dataset mencionado. Así, mediante la definición de diferentes límites, podemos definir las zonas de alerta por situación anómala.

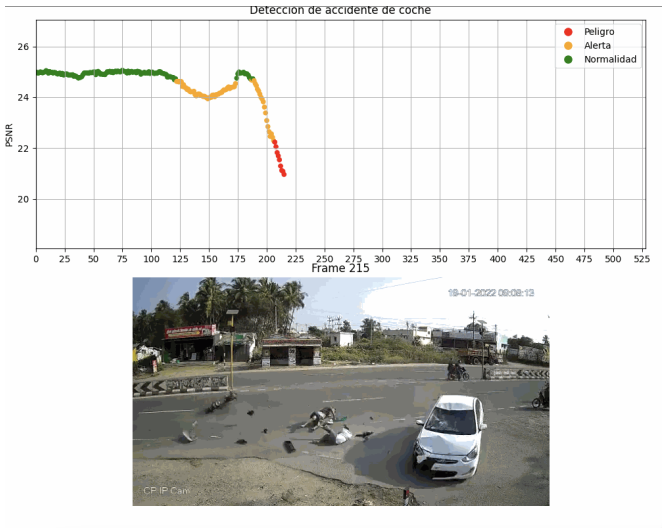

Figura 12: Funcionamiento del sistema desarrollado

4. Conclusiones

En este paper se han propuesto cuatro arquitecturas distintas para el desarrollo de un sistema que permita la detección de anomalías en entornos urbanos. Además, se han llevado a cabo diversos experimentos de cada una de ellas para llegar a la conclusión de que la mejor arquitectura y la más simple para este caso de uso es la **U-Net**, puesto que es con la que mejores resultados se han obtenido y con la que mejores métricas en el estudio comparativo se han conseguido. Como se ha podido observar en el documento, las variaciones entre los modelos obtenidos no son muy dispares, sin embargo, tratando un tema tan importante como es la detección de anomalías, es crucial emplear el modelo con la mayor precisión y eficacia, aunque esta diferencia entre modelos sea mínima.

Además, también se ha observado que existen ciertas limitaciones a la hora de utilizar un modelo entrenado con secuencias de un entorno real sobre un entorno simulado, empeorando los resultados cuanta mayor disparidad exista entre el aspecto de la simulación y el real. Sin embargo, se ha demostrado que es posible extrapolar los modelos entrenados en un entorno real a otros entornos reales diferentes que nunca han visto, dotando al sistema de una mayor adaptabilidad, permitiendo una correcta generación de imágenes y detección de anomalías.

Por último y como trabajo a futuro, se plantea la opción de implemenentar en el sistema un modelo de detección de objetos para detectar exactamente dónde se encuentra la anomalía. También resultaría interesante el uso de un modelo LLM (Largue Lenguage Model) capaz de añadir una descripción al fotograma detectado como anomalía. De esta forma, el proceso de vigilancia de un entorno urbano se verá reducido a atender los avisos de anomalías detectadas. Por último, se plantea la opción de probar otras funciones de pérdida que pudieran mejorar aun más las reconstrucciones de las imágenes u obteniendo mejores resultados en alguno de los modelos.

Agradecimientos

Este trabajo ha sido financiado por el Gobierno Español a través de los proyectos ID2021-128327OA-I00, PID2021-124335OB-C21, y TED2021-129374A-I00 financiado por MCIN/AEI/10.13039/501100011033, por la Unión Europea NextGenerationEU/PRTR.

Referencias

Aboah, A., 2021. A vision-based system for traffic anomaly detection using deep learning and decision trees. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4207–4212.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale.

Fan, C., Lin, H., Qiu, Y., 2023. U-patch gan: A medical image fusion method based on gan. Journal of Digital Imaging 36 (1), 339–355.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial networks.

Isola, P., Zhu, J.-Y., Zhou, T., Efros, A. A., 2017. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134.

Leporowski, B., Bakhtiarnia, A., Bonnici, N., Muscat, A., Zanella, L., Wang, Y., Iosifidis, A., 2023. Audio-visual dataset and method for anomaly detection in traffic videos. arXiv preprint arXiv:2305.15084.

Pathak, D., Sharang, A., Mukerjee, A., 2015. Anomaly localization in topic-based analysis of surveillance videos. In: 2015 IEEE winter conference on applications of computer vision. IEEE, pp. 389–395.

Roka, S., Diwakar, M., 2023. Cvit: a convolution vision transformer for video abnormal behavior detection and localization. SN Computer Science 4 (6), 829.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer, pp. 234–241.

Yuan, H., Cai, Z., Zhou, H., Wang, Y., Chen, X., 2021. Transanomaly: video anomaly detection using video vision transformer. IEEE Access 9, 123977–123986.

Yun, K., Jeong, H., Yi, K. M., Kim, S. W., Choi, J. Y., 2014. Motion interaction field for accident detection in traffic surveillance video. In: 2014 22nd International Conference on Pattern Recognition. IEEE, pp. 3062–3067.