ESTOA

Artículo de investigación
2027 Enero - Junio
English version

TECNOLOGÍAS y CONSTRUCCIÓN

Detección visual de grietas en fachadas de viviendas autoconstruidas de Huancayo mediante *deep learning*

Visual detection of cracks in facades of self-built dwellings in Huancayo using deep learning

PAOLO MARTIN LOPEZ-GOMEZ
Universidad Nacional del Centro del Perú, Perú
E_2020101606C@uncp.edu.pe

RAFAEL JHON FLORES-CORILLOCLLA
Universidad Nacional del Centro del Perú, Perú
E_2020101614C@uncp.edu.pe

ADOLFO GUSTAVO CONCHA-FLORES
Universidad Nacional del Centro del Perú, Perú
aconcha@uncp.edu.pe

RESUMEN En Huancayo, las viviendas autoconstruidas sin supervisión técnica representan una problemática en el contexto sísmico regional, especialmente al presentar daño visible en sus fachadas. Este estudio desarrolló una herramienta basada en *deep learning* para la identificación preliminar de elementos de fachada y grietas visibles en imágenes. Se emplearon dos modelos especializados de segmentación de instancias con Mask R-CNN y Detectron2, uno para muros, columnas y vanos, y otro para grietas incorporando PointRend. La evaluación se realizó sobre subconjuntos de prueba aislados, reportando métricas COCO y operativas. En bounding box, los modelos de fachadas y grietas alcanzaron AP50 de 59,87% y 65,16%; en *instance segmentation*, los AP50 fueron 58,30% y 61,13%, con AP de 20,03% para *instance segmentation* en grietas. Los valores F1-score fueron 65,82% y 72,15% en *bounding box*, y 63,91% y 74,49% en segmentación. Los resultados evidencian un desempeño moderado, útil para apoyar el registro visual preliminar de fachadas.

PALABRAS CLAVE deep learning, vivienda autoconstruida, detección de grietas, segmentación de instancias, análisis de imágenes

ABSTRACT In Huancayo, self-constructed dwellings without technical supervision pose a concern within the regional seismic context, particularly when exhibiting visible damage on their facades. This study developed a deep learning–based tool for the preliminary identification of facade elements and visible cracks in images. Two specialized Mask R-CNN instance segmentation models were implemented in Detectron2: one for walls, columns, and openings, and another for cracks incorporating PointRend. Evaluation was conducted on independent test subsets using COCO and operational metrics. For bounding boxes, the facade and crack models achieved AP50 values of 59.87% and 65.16%; for instance segmentation, the AP50 values were 58.30% and 61.13%, with an AP of 20.03% for crack instance segmentation. F1-scores were 65.82% and 72.15% for bounding boxes, and 63.91% and 74.49% for segmentation. The results indicate moderate performance, supporting the tool's use as an aid for preliminary visual documentation of facades.

KEYWORDS deep learning, self-constructed housing, crack detection, instance segmentation, image analysis

Recibido: 02/03/2026
Revisado: 24/07/2026
Aceptado: 31/08/2026
Publicado: 08/10/2026

CC BY NC SA

Cómo citar este artículo/How to cite this article: Lopez-Gomez, P. M., Flores-Corilloclla, R. J. y Concha-Flores, A. G. (2027). Detección visual de grietas en fachadas de viviendas autoconstruidas de Huancayo mediante deep learning. *Estoa. Revista de la Facultad de Arquitectura y Urbanismo, 16*(31), e6651. https://doi.org/10.18537/est.v016.n031.a10

ESTOA **31** / Vol 16 / 2027
e - ISSN: 1390 - 9274
ISSN: 1390 - 7263

1. Introducción

En diversas ciudades del Perú, como Huancayo, la construcción informal constituye una forma predominante de producción habitacional que puede desarrollarse sin planos ni participación profesional. En una muestra local de 20 viviendas, García (2022) reportó ambas carencias en el 65% de los casos y clasificó el 60% con riesgo sísmico medio y el 40% con riesgo alto. La falta de documentación y supervisión técnica puede generar variaciones en materiales y procesos constructivos, además de dificultar la inspección preliminar. Esta situación adquiere relevancia por la dinámica sísmica de la cuenca del Mantaro, asociada al sistema de fallas del Huaytapallana (Tavera et al., 2025).

Esta situación se enmarca en la necesidad de automatizar el reconocimiento visual de edificaciones y el registro de sus condiciones observables. Wang et al. (2021) mostraron que la segmentación de imágenes puede utilizarse para identificar edificaciones de albañilería no reforzada. Aunque su estudio aborda una clasificación tipológica y no la detección de daño visible, evidencia la utilidad de automatizar el reconocimiento mediante imágenes frente a las limitaciones de la inspección manual. Ante ello, la automatización puede aumentar la velocidad y reducir los costos y posibles errores humanos durante el monitoreo visual (Mohtasham et al., 2020).

Diversos estudios de caso han señalado que la detección temprana de daño visible puede apoyar la priorización de inspecciones técnicas y la prevención de daños severos en edificaciones (Roy et al., 2025). En otros contextos, como las superficies metálicas, las manchas de óxido pueden actuar como pseudodefectos al asemejarse visualmente a daños reales y dificultar su identificación automática (Wang et al., 2023). De forma similar, en infraestructura, la calidad de imagen, la iluminación y la variabilidad superficial representan retos para la detección automática de defectos (Koch et al., 2015). Pese a ello, la visión computacional constituye una alternativa no contactante, rápida y de bajo costo que complementa los métodos convencionales de evaluación (Dong y Catbas, 2020).

Para la detección visual de grietas, esta investigación se fundamenta en el *deep learning*. La arquitectura central de este enfoque son las redes neuronales convolucionales (CNN), las cuales son cruciales para las aplicaciones de visión por computadora. La eficiencia de las CNN radica en su capacidad para aprender y reconocer patrones espaciales de forma jerárquica. Sus capas iniciales aprenden a identificar características simples como bordes y texturas, mientras que las capas más profundas combinan estas características para reconocer patrones progresivamente más complejos (Figura 1), como la morfología específica de una grieta visible (Goodfellow et al., 2016).

Las CNN permiten clasificar imágenes y localizar objetos mediante segmentación, generando máscaras con información geométrica útil para el registro y análisis cuantitativo del daño (Cha et al., 2017). En particular, la segmentación de instancias detecta y delimita individualmente cada objeto de interés a nivel de píxel, proporcionando una máscara que representa su forma con mayor detalle y permite analizarlo por separado (Bai et al., 2023).

Para lograr esto, la investigación emplea la arquitectura Mask R-CNN (Figura 2). El modelo opera en un proceso de dos fases principales. En la primera fase, una CNN base analiza la imagen de entrada para generar un mapa de características visuales. Sobre dicho mapa, una red de propuesta de regiones (RPN) identifica áreas de interés (RoI) con alta probabilidad

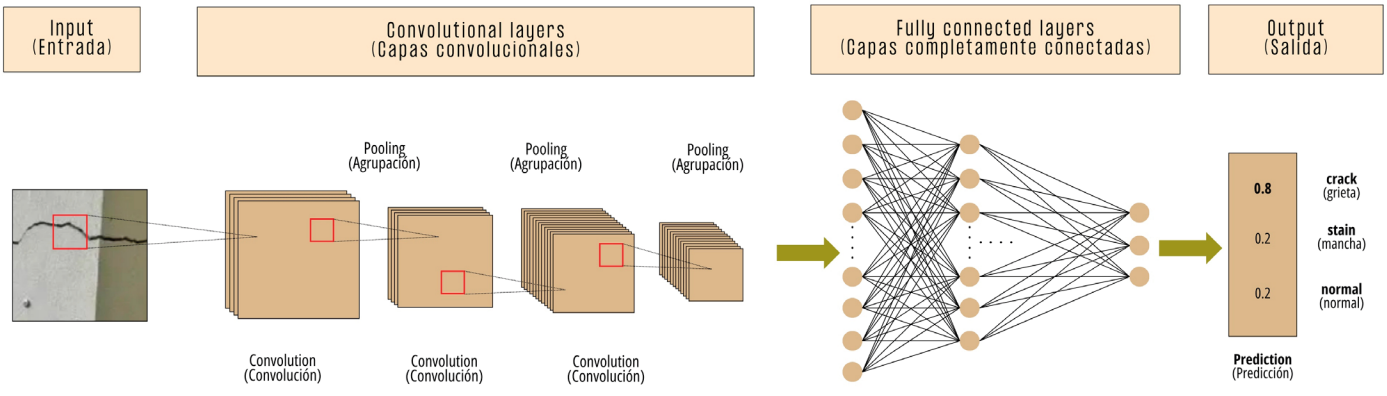


Figura 1: Esquema de funcionamiento de una CNN. (2025)

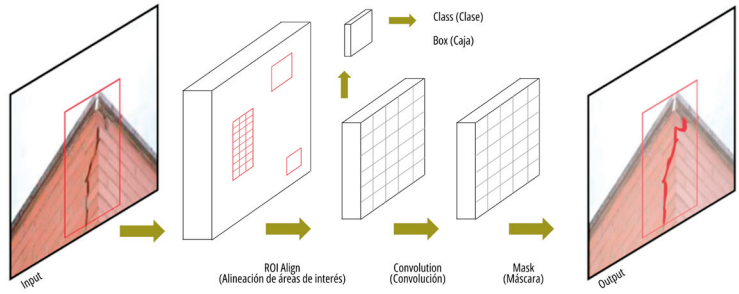


Figura 2: Esquema de funcionamiento de Mask R-CNN. (2025)

de contener objetos. Posteriormente, para cada región propuesta, Mask R-CNN clasifica el objeto, refina su ubicación mediante un cuadro delimitador y genera una máscara que representa su contorno visible a nivel de píxel (Wang et al., 2021).

Sin embargo, en tareas donde los objetos presentan bordes delgados, discontinuos o irregulares, la calidad de la máscara puede depender del refinamiento local de los contornos. En este sentido, PointRend plantea la segmentación como un proceso de renderizado adaptativo, calculando predicciones en puntos seleccionados de la imagen, especialmente en regiones inciertas o de alta frecuencia, como los límites de los objetos (Kirillov et al., 2020). Este enfoque resulta pertinente para la detección de grietas visibles, debido a su morfología fina, irregular y variable.

Asimismo, la automatización visual también puede apoyar la detección de daños superficiales y la evaluación visual de edificaciones a mayor escala. Los modelos basados en redes neuronales pueden mejorar la consistencia del registro frente a inspecciones manuales afectadas por la subjetividad, la fatiga visual y los costos operativos (Cha et al., 2024). En tareas donde la delimitación resulta prioritaria, Choi et al. (2024) compararon Mask R-CNN con YOLOv8 para identificar grietas en concreto. Mask R-CNN alcanzó un 96,5% de intersección sobre unión (IoU), frente al 90,6% de YOLOv8, y presentó una delimitación morfológica más constante, aunque la velocidad de procesamiento no constituyó su principal ventaja.

A pesar de los avances, aplicar estos algoritmos en contextos de autoconstrucción representa un desafío metodológico adicional. Kong y Petchsasithon (2024) evidencian que las estructuras de albañilería en zonas rurales o informales pueden presentar una alta complejidad visual, ya que carecen de recubrimientos y presentan juntas de mortero irregulares. Su investigación señala que la integración de algoritmos de aprendizaje profundo aplicados a imágenes contribuye a capturar eficazmente estas irregularidades y a generar información visual útil para el análisis del daño sísmico en edificaciones que no siguen normativas formales.

Al analizar el contexto nacional, Palomino Ojeda et al. (2024) señalan que las viviendas de albañilería confinada, una tipología común en diversas regiones del Perú, se caracterizan por ser autoconstruidas sin supervisión técnica, lo que puede incrementar sus condiciones de vulnerabilidad frente a eventos sísmicos. En este contexto, los modelos computacionales constituyen una vía relevante para analizar indicadores de daño de las viviendas, siempre que sus resultados se interpreten dentro de sus límites metodológicos.

A partir de las limitaciones de la inspección manual, la variabilidad visual de las viviendas autoconstruidas y la necesidad de registrar grietas visibles de forma sistemática, el presente estudio propone una herramienta de detección automatizada basada en aprendizaje profundo aplicado al análisis de imágenes. La herramienta no intenta determinar integralmente la vulnerabilidad estructural, ya que esta depende de múltiples factores de diseño, materiales, sistema resistente y demanda sísmica.



Figura 3: Dataset anotado en Roboflow. (2025)

En cambio, se centra en identificar y segmentar elementos de fachada y grietas visibles, consideradas indicadores preliminares de daño. Asimismo, estima preliminarmente el grosor de estas últimas mediante una referencia visual calibrada, como apoyo al registro visual de las mismas y a la priorización de inspecciones técnicas posteriores.

2. Método

2.1. Conjunto de datos

Para desarrollar la herramienta se organizaron dos conjuntos de datos destinados a modelos especializados de segmentación de instancias. El primero se orientó a identificar muros, columnas y vanos; el segundo, a detectar y segmentar grietas visibles. Esta separación permitió entrenar cada modelo con clases visualmente específicas, sin mezclar objetivos de aprendizaje diferentes.

El conjunto de datos de fachadas estuvo conformado por imágenes de viviendas autoconstruidas de Huancayo. Las fotografías fueron capturadas en condiciones de iluminación natural, principalmente durante la mañana y la tarde, excluyendo escenas nocturnas o de baja visibilidad que pudieran afectar la identificación. Las imágenes originales fueron registradas con una cámara SONY DSC-W520, con una resolución de 4320 × 3240 píxeles.

La selección de imágenes consideró la presencia visual clara de elementos arquitectónico-estructurales relevantes para el estudio. Las clases consideradas fueron muro, columna y vanos, debido a que permiten ubicar espacialmente la presencia de grietas en superficies de fachada y reducir confusiones con aberturas, sombras o elementos no estructurales. El conjunto incluyó 355 fachadas diferentes, correspondiendo a una única imagen por vivienda, tanto con muros de ladrillo expuesto como con acabados pintados, con el fin de incorporar variabilidad visual y evitar que el modelo asociara los elementos de fachada únicamente con una textura constructiva específica.

El conjunto de datos de grietas estuvo conformado por imágenes orientadas exclusivamente a su identificación. Este conjunto proviene del dataset abierto Cracks Dataset, publicado en Roboflow Universe por Albaraa Hussein Osman bajo licencia CC BY 4.0 (Osman, 2025).

Cantidad de imágenes y anotaciones en Roboflow					
Modelo	**Clase**	**Anotaciones por clase**	**Imágenes train**	**Imágenes valid**	**Imágenes test**
Modelo de grietas	grietas	6241	3222	403	403
Modelo de fachadas	muro	2231			
Modelo de fachadas	columna	1013	859	36	36
Modelo de fachadas	vanos	1675			

Tabla 1: Cantidad de imágenes y anotaciones en Roboflow. (2025)

Su incorporación permitió fortalecer el entrenamiento del modelo especializado en grietas visibles mediante un volumen mayor de ejemplos anotados, concentrados en la morfología de esta clase. De esta manera, el sistema propuesto combina un conjunto propio de imágenes de fachadas de Huancayo con un conjunto abierto específico para esta tarea, manteniendo tareas de aprendizaje separadas para cada modelo.

2.2. Preparación del conjunto de datos

La preparación de los datos se llevó a cabo mediante etiquetado manual de instancias y organización de los conjuntos en la plataforma Roboflow (Figura 3). Este procedimiento constituye una etapa estándar en el desarrollo de modelos de visión por computadora, debido a que permite generar etiquetas verificadoras (*ground-truth labels*) que el modelo emplea durante el aprendizaje supervisado (Zhu et al., 2022).

Para el conjunto de fachadas, los objetos de interés fueron demarcados manualmente mediante líneas que conformaron polígonos, diferenciando las clases muro, columna y vanos. Esta anotación por instancia permitió representar la forma aproximada de cada elemento visible y no únicamente su localización rectangular.

En el caso del conjunto de grietas, se mantuvo la clase grietas como categoría única junto a las anotaciones disponibles originalmente. En ambos conjuntos, las anotaciones fueron preparadas en formato COCO JSON, lo que permitió su uso en Detectron2 y Mask R-CNN para el entrenamiento y evaluación de modelos de segmentación de instancias.

Durante la elaboración, las imágenes fueron redimensionadas a 1024 × 1024 píxeles en Roboflow, encajando la imagen con bordes negros para no deformarla. Esta resolución se adoptó con el fin de conservar detalle visual en texturas, bordes y grietas, manteniendo al mismo tiempo una dimensión compatible con el entrenamiento en una GPU de 8 GB de VRAM. La elección de esta resolución buscó equilibrar la preservación de información visual relevante con las limitaciones computacionales disponibles.

Debido al tamaño limitado del conjunto de fachadas, sus 355 imágenes originales se distribuyeron aproximadamente en 80/10/10: 283 para entrenamiento (*train*), 36 para validación (*valid*) y 36 para prueba (*test*). Esta partición se realizó antes del aumento de datos, el cual se aplicó únicamente al subconjunto de entrenamiento mediante Roboflow, ampliándolo a 859 imágenes; los subconjuntos de validación y prueba permanecieron sin aumento e independientes para el monitoreo y la evaluación final.

El aumento de datos incluyó volteo horizontal, variaciones de brillo entre valores de -20% y +20%, y modificaciones de saturación entre -15% y +15%. Esta estrategia se fundamenta en el uso del aumento de datos como una práctica habitual para incrementar la variabilidad del conjunto de entrenamiento y mejorar el aprendizaje de modelos de visión por computadora (Gouveia et al., 2024), evitando al mismo tiempo la incorporación de imágenes derivadas en los conjuntos de validación y prueba.

El conjunto de grietas estuvo conformado por 4028 imágenes y 6241 anotaciones correspondientes a la clase grietas. A diferencia del conjunto de fachadas, este dataset se empleó sin aplicar aumento de datos adicional. La partición utilizada fue de 3222 imágenes para entrenamiento (*train*), 403 para validación (*valid*) y 403 para prueba (*test*), correspondiente a una distribución aproximada de 80/10/10. La Tabla 1 resume la distribución final de imágenes y anotaciones empleadas para cada modelo especializado.

Para la evaluación cuantitativa posterior de ambos modelos, se reservaron sus respectivos subconjuntos de prueba, definidos durante la partición de cada conjunto de datos y mantenidos aislados del entrenamiento y la validación. Para el modelo de fachadas se consideraron 36 imágenes originales no aumentadas, mientras que para el modelo de grietas se reservaron 403 imágenes del subconjunto de prueba.

2.3. Marco de trabajo para el modelo

Se implementaron dos modelos especializados de segmentación de instancias: uno para identificar muros, columnas y vanos, y otro para detectar y segmentar grietas visibles. Ambos utilizaron Mask R-CNN y fueron entrenados mediante Detectron2, biblioteca de código abierto que proporciona herramientas para el entrenamiento, evaluación e inferencia de modelos de visión por computadora (Padsumbiya et al., 2022).

Mask R-CNN fue seleccionada por su capacidad para identificar, clasificar y delimitar objetos mediante cuadros delimitadores y máscaras de segmentación, permitiendo representar su ubicación y forma aproximada en la imagen. En ambos modelos, la arquitectura base se configuró con ResNet-50 y FPN para extraer características visuales en diferentes escalas. Asimismo, se aplicó aprendizaje por transferencia, esta técnica permite reutilizar pesos previamente aprendidos en grandes conjuntos de datos, de modo que las capas iniciales del modelo, asociadas al reconocimiento de patrones generales como bordes, texturas y formas simples, sirvan como punto de partida para el aprendizaje de clases específicas del estudio (Zhu et al., 2022).

Para el modelo especializado en grietas se incorporó PointRend como módulo de refinamiento de máscaras, debido a que estas presentan bordes delgados, discontinuos e irregulares. PointRend plantea la segmentación como un proceso de renderizado adaptativo, en el cual se calculan

predicciones en puntos seleccionados de la imagen, especialmente en regiones de mayor incertidumbre o alta frecuencia, como los límites de los objetos (Kirillov et al., 2020). Esta característica resulta adecuada para mejorar la representación de contornos finos en la segmentación de grietas visibles.

El sistema fue desarrollado en Python 3.11.9, utilizando Detectron2 0.6 y PyTorch 2.7.1, como bibliotecas principales para la implementación y el entrenamiento de los modelos. También se emplearon OpenCV 4.13.0 y NumPy 2.4.4 para el procesamiento de imágenes y las operaciones auxiliares. Jupyter Notebook se utilizó como entorno de apoyo visual para organizar el código e inspeccionar salidas intermedias. El entrenamiento se ejecutó con aceleración mediante GPU, utilizando CUDA 11.8 y una tarjeta gráfica de 8 GB de VRAM. Esta configuración permitió procesar imágenes de 1024 × 1024 píxeles, manteniendo un equilibrio entre el detalle visual y la estabilidad del procesamiento (Figura 4).

2.4. Asignación de parámetros

Una vez preparados los conjuntos de datos y configuradas las arquitecturas de segmentación, se procedió con la fase de entrenamiento supervisado de los dos modelos especializados. La asignación de parámetros se realizó considerando la arquitectura empleada, el tamaño de cada conjunto de datos, la complejidad visual de las clases y las restricciones computacionales disponibles. Esta decisión se fundamenta en que los modelos de aprendizaje automático dependen de hiperparámetros que pueden modificar su comportamiento, complejidad, velocidad y rendimiento, por lo que su selección debe realizarse con cuidado según el problema abordado (Bischl et al., 2023).

El modelo especializado en elementos de fachada se entrenó con un límite máximo de 25000 iteraciones, mientras que el modelo especializado en grietas visibles se configuró con un límite de 80000 iteraciones. Ambos modelos utilizaron pesos preentrenados en COCO Mask R-CNN R50-FPN 3x como punto de partida, aplicando aprendizaje por transferencia para adaptar la arquitectura a las clases del estudio.

La selección de los pesos finales se controló mediante evaluaciones periódicas sobre el conjunto de validación, lo que permitió monitorear el comportamiento de los modelos con datos no utilizados para el ajuste directo, reduciendo el riesgo de depender únicamente del último estado del proceso de optimización (Vilares Ferro et al., 2023). Asimismo, se conservaron puntos de control (*checkpoints*), producto de las evaluaciones periódicas asociadas al mejor desempeño en validación, considerando que la selección del estado del modelo durante el entrenamiento es relevante para evitar sobreajuste y conservar recursos computacionales (Miseta et al., 2024). Se configuraron periodos de evaluación y puntos de control diferenciados para cada modelo: cada 500 iteraciones para el modelo de fachadas y cada 2500 para el modelo de grietas.

La tasa de aprendizaje se administró mediante una estrategia de reducción escalonada para permitir un ajuste fino en las etapas finales del aprendizaje (Aravena et al., 2021) y mantener la estabilidad de los pasos de actualización durante la optimización profunda (Nakamura et al., 2021). Para el modelo de fachadas se estableció una tasa base de 0,001, reducida en las iteraciones 17 500 y 22 500, mientras que para el modelo de grietas se utilizó una tasa base de 0,0005, reducida en las iteraciones 56 000 y 72 000.

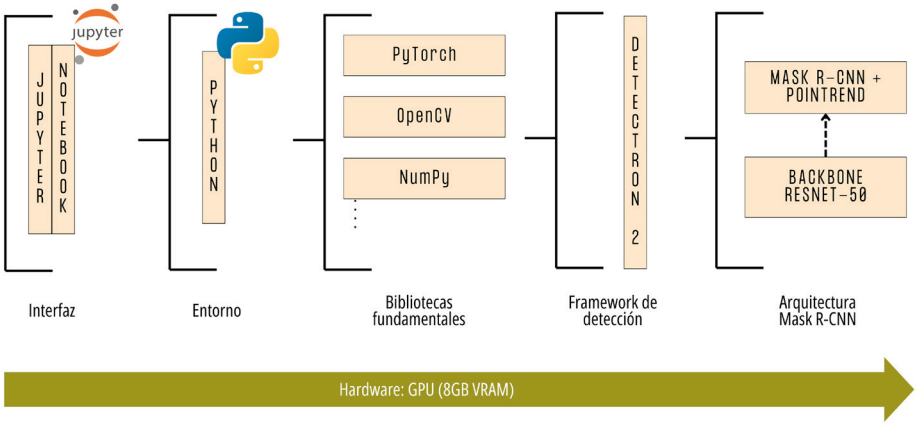


Figura 4: Marco de trabajo para el entrenamiento de los modelos especializados. (2025)

Configuración de parámetros para el entrenamiento		
Parámetro	**Modelo de fachadas**	**Modelo de grietas**
Marco de implementación *(Implementation framework)*	Detectron2	Detectron2
Arquitectura *(Architecture)*	Mask R-CNN R50-FPN	Mask R-CNN R50-FPN + PointRend
Clases entrenadas *(Training classes)*	muro, columna, vanos	Grietas
Pesos iniciales *(Initial weights)*	COCO Mask R-CNN R50-FPN 3x	COCO Mask R-CNN R50-FPN 3x
Resolución de entrada preprocesada *(Preprocessed input resolution)*	1024 x 1024 px	1024 x 1024 px
Máximo de iteraciones *(Maximum iterations)*	25 000	80 000
Tasa de aprendizaje *(Initial learning rate)*	0,001	0,0005
Pasos de reducción de tasa de aprendizaje *(Learning rate steps)*	(17 500, 22 500)	(56 000, 72 000)
Factor de decaimiento de la tasa de aprendizaje *(Learning rate decay factor)*	0,1	0,1
Tamaño del lote *(Batch size)*	4	2
Trabajadores del cargador de datos *(Data loader workers)*	2	2
Tamaño del lote por imagen RoI (*RoI batch size per image)*	128	512
Precisión mixta automática *(Automatic Mixed Precision, AMP)*	Activada	Activada
Periodo de evaluación *(Evaluation period)*	500	2500
Criterio de selección del mejor modelo *(Best model selection criterion)*	AP en segmentación	AP en segmentación
Periodo de guardado del punto de control (*Checkpoint period)*	500	2500

Tabla 2: Configuración de parámetros para el entrenamiento. (2026)

El tamaño de lote se ajustó según la memoria GPU y la complejidad del modelo: cuatro imágenes por iteración para fachadas y dos para grietas, debido al alto costo computacional del refinamiento de máscaras con PointRend. Esto concuerda con Kandel y Castelli (2020), quienes indican que el tamaño de lote es un hiperparámetro clave en redes convolucionales cuya relación con la tasa de aprendizaje afecta el rendimiento. Adicionalmente, para optimizar el uso de memoria VRAM y prevenir errores, ambos entrenamientos utilizaron precisión mixta automática (AMP) (Tabla 2).

2.5. Métricas de evaluación

La evaluación del rendimiento de los modelos se fundamentó en protocolos estandarizados basados en la métrica de intersección sobre unión (IoU) para definir detecciones correctas (Obi-Obuoha et al., 2024) y en la precisión media promedio (mAP) como indicador principal de rendimiento (Ferguson et al., 2018). A partir de la superposición espacial entre el área real y la detectada, la precisión (*precision*) cuantifica la proporción de detecciones verdaderas positivas respecto al total emitido, mientras que la sensibilidad (*recall*) mide la capacidad del modelo para localizar todos los objetos existentes del conjunto evaluado (Obi-Obuoha et al., 2024).

Asimismo, el AP (*Average Precision*) evalúa el rendimiento integral a través de distintos niveles de exigencia de IoU, reportando el AP principal promediado entre IoU = .50 e IoU = .95, el AP50 específico a IoU = .50 y el AR (Average Recall) como promedio del *recall* bajo diferentes umbrales (Obi-Obuoha et al., 2024). La extracción de estas métricas se automatizó mediante las herramientas de evaluación integradas en Detectron2, contrastando las predicciones frente a las etiquetas *ground-truth* del conjunto de prueba.

Para complementar las métricas COCO, se calcularon métricas operativas basadas en conteos de verdaderos positivos (TP), falsos positivos (FP) y falsos negativos (FN). En este análisis, una predicción se consideró correcta cuando alcanzó un umbral de superposición IoU ≥ .50 con una anotación de referencia y presentó un umbral de confianza *score* ≥ .50. A partir de estos conteos se calcularon *Precision, Recall/Sensitivity* y *F1-score* para ambos modelos. Las métricas COCO AP y AR se calcularon de forma independiente respecto a las operativas, debido a que responden a criterios de cálculo diferentes.

Para el modelo de grietas se aplicó un posprocesamiento para fusionar máscaras de la misma clase cuando una predicción se encontraba contenida dentro de otra o cuando correspondían a fragmentos superpuestos de una misma grieta continua, estableciendo como criterio que el área de intersección representara al menos el 35% de la máscara menor. Este procedimiento tuvo como objetivo reducir detecciones duplicadas sin alterar los pesos entrenados del modelo. Las predicciones resultantes se emplearon exclusivamente para las métricas operativas de segmentación de instancias, mientras que las métricas COCO AP, AP50 y AR se evaluaron con las predicciones originales.

2.6. Estimación métrica de las grietas

Se implementó un procedimiento para la estimación cuantitativa del grosor de las grietas detectadas. Este método se basó en el uso de un marcador ArUco como referencia visual calibrada en la imagen, específicamente un marcador DICT_5X5_50, ID 0, con lado conocido de 5 cm. El procedimiento consistió en detectar el marcador en la imagen mediante OpenCV y calcular la relación píxel-centímetro a partir de su dimensión física conocida. Posteriormente, para procesar la morfología de la grieta, el grosor promedio se determinó dividiendo el área de píxeles de la máscara segmentada por PointRend entre la longitud del eje mayor de su rectángulo orientado de área mínima obtenido mediante cv2.minAreaRect. Esta formulación modela el trazo como una franja continua equivalente, escalando su valor a centímetros mediante la referencia óptica calibrada. Por consiguiente, el resultado se define como una estimación preliminar que asume condiciones de coplanaridad en la captura y una trayectoria predominantemente lineal, siendo propensa a desviaciones geométricas ante grietas de alta sinuosidad o ramificadas.

De forma complementaria, el sistema determinó la orientación de cada grieta aplicando un ajuste de recta por mínimos cuadrados sobre el contorno de su máscara mediante la función cv2.fitLine de OpenCV. El ángulo de inclinación respecto a la horizontal se clasificó en tres intervalos operativos, horizontal de 0° a 25°, diagonal para valores mayores a 25° y menores a 65°, y vertical de 65° a 90°. Asimismo, la asociación al elemento estructural se efectuó evaluando la superposición espacial y el grado de inclusión de la máscara de la grieta dentro de las cajas delimitadoras de muros y columnas identificadas por el modelo de fachadas, descartando aquellas detecciones ubicadas en vanos para mitigar falsos positivos.

2.7. Despliegue en interfaz web

Para facilitar la demostración visual del sistema y la estimación métrica, se desplegó la herramienta en una interfaz web interactiva. Esta se desarrolló utilizando la biblioteca de software Gradio, una herramienta de código abierto diseñada para crear y compartir rápidamente interfaces web (Abid et al., 2019). La interfaz resultante fue diseñada para recibir una imagen, procesarla en tiempo real y generar dos salidas principales, una visualización de las detecciones generadas por los modelos especializados y un resumen textual automatizado. Cuando la imagen incluía un marcador ArUco visible, la herramienta incorporaba además una estimación métrica preliminar de las grietas; en ausencia de esta referencia, el sistema mantenía únicamente la identificación y segmentación visual de los elementos detectados.

3. Resultados

3.1. Rendimiento del modelo

Los resultados del procedimiento de evaluación cuantitativa se presentan a continuación. La Tabla 3 detalla el rendimiento obtenido en la tarea de detección mediante cuadros delimitadores (*bounding box*), mientras que la Tabla 4 presenta los resultados correspondientes a la segmentación de instancias (*instance segmentation*). En ambos casos se reportan métricas COCO, como AP, AP50 y AR, junto con métricas operativas basadas en TP, *FP, FN, Precision, Recall* y *F1-score*, calculadas bajo IoU ≥ .50 y score ≥ .50.

Para la detección mediante cuadros delimitadores (Tabla 3), el modelo de fachadas obtuvo un AP@[IoU = .50:.95] de 38,81%, un AP50 de 59,87% y un AR de 49,15%. El modelo de grietas registró un AP de 49,63%, un AP50 de 65,16% y un AR de 55,69%. En las métricas operativas, el modelo de fachadas alcanzó una *Precision* de 67,98%, un *Recall* de 63,79% y un F1-score de 65,82%, mientras que el modelo de grietas obtuvo una *Precision* de 81,64%, un *Recall* de 64,63% y un *F1-score* de 72,15%.

Rendimiento (*bounding box*)						
Métrica	**Configuración**	**Modelo de fachadas**	**Modelo de grietas**	**Muro**	**Columna**	**Vanos**
AP	IoU = .50 - .95	38,81%	49,63%	33,61%	29,33%	53,47%
AP50	IoU = .50	59,87%	65,16%	53,08%	51,95%	74,59%
AR	IoU = .50 - .95	49,15%	55,69%	44,14%	37,89%	65,41%
TP	IoU ≥ .50, score ≥ .50	518	338	205	114	199
FP	IoU ≥ .50, score ≥ .50	244	76	115	47	82
FN	IoU ≥ .50, score ≥ .50	294	185	150	99	45
Precision	IoU ≥ .50, score ≥ .50	67,98%	81,64%	64,06%	70,81%	70,82%
Recall	IoU ≥ .50, score ≥ .50	63,79%	64,63%	57,75%	53,52%	81,56%
F1-score	IoU ≥ .50, score ≥ .50	65,82%	72,15%	60,74%	60,96%	75,81%

Tabla 3: Métricas de rendimiento – *bounding box.* (2026)

Rendimiento (*instance segmentation*)						
Métrica	**Configuración**	**Modelo de fachadas**	**Modelo de grietas**	**Muro**	**Columna**	**Vanos**
AP	IoU = .50 - .95	37,04%	20,03%	30,59%	25,75%	54,78%
AP50	IoU = .50	58,30%	61,13%	50,42%	50,63%	73,86%
AR	IoU = .50 - .95	47,23%	25,69%	41,41%	34,08%	66,19%
TP	IoU ≥ .50, score ≥ .50	503	349	198	108	197
FP	IoU ≥ .50, score ≥ .50	259	65	122	53	84
FN	IoU ≥ .50, score ≥ .50	309	174	157	105	47
Precision	IoU ≥ .50, score ≥ .50	66,01%	84,30%	61,88%	67,08%	70,11%
Recall	IoU ≥ .50, score ≥ .50	61,95%	66,73%	55,77%	50,70%	80,74%
F1-score	IoU ≥ .50, score ≥ .50	63,91%	74,49%	58,67%	57,75%	75,05%

Tabla 4: Métricas de rendimiento – *instance segmentation*. (2026)

En el análisis por clase del modelo de fachadas para *bounding box*, la clase muro obtuvo un AP50 de 53,08% y un *F1-score* de 60,74%; la clase columna alcanzó un AP50 de 51,95% y un *F1-score* de 60,96%; y la clase vanos registró un AP50 de 74,59% y un *F1-score* de 75,81%.

Para la tarea de segmentación de instancias (Tabla 4), el modelo de fachadas obtuvo un AP@[IoU = .50: .95] de 37,04%, un AP50 de 58,30% y un AR de 47,23%. El modelo de grietas registró un AP de 20,03%, un AP50 de 61,13% y un AR de 25,69%. En las métricas operativas, el modelo de fachadas alcanzó una *Precision* de 66,01%, un *Recall* de 61,95% y un F1-score de 63,91%, mientras que el modelo de grietas obtuvo una *Precision* de 84,30%, un *Recall* de 66,73% y un F1-score de 74,49%.

En el análisis por clase del modelo de fachadas para segmentación de instancias, para la clase muro se obtuvo un AP50 de 50,42% y un *F1-score* de 58,67%; la clase columna registró un AP50 de 50,63% y un *F1-score* de 57,75%; y la clase vanos alcanzó un AP50 de 73,86% y un *F1-score* de 75,05%.

3.2. Demostración visual de la herramienta

Como complemento cualitativo e ilustrativo a la evaluación cuantitativa, se realizó una demostración visual de la herramienta desplegada en interfaz web. Este procedimiento se aplicó sobre imágenes independientes de viviendas de Huancayo y no fue utilizado para el cálculo de métricas de rendimiento. La Figura 5 presenta dos ejemplos de inferencia, mostrando la imagen de entrada, la salida visual con cuadros delimitadores y máscaras de segmentación, y el informe preliminar generado a partir de las detecciones obtenidas.

En la Figura 5a, correspondiente a una vivienda de fachada blanca, la inferencia se realizó desde una distancia que permitió capturar una vista general de la fachada. El sistema identificó muros, vanos y una columna. El informe preliminar registró una grieta con orientación vertical asociada a un muro, con una confianza de 61%. Debido a que la imagen no incorporó un marcador ArUco visible, la herramienta indicó que la medición métrica no estaba disponible.

En la Figura 5b, correspondiente a una vivienda de fachada celeste, la inferencia también se realizó sobre una vista general de fachada. El sistema identificó muros, vanos y columnas. En este caso, el informe preliminar registró una grieta con orientación horizontal asociada a un muro, con una confianza de 96%. Al no contar con un marcador ArUco visible, la herramienta indicó que no se disponía de medición métrica.

Adicionalmente, se realizó una inferencia a corta distancia para mostrar la estimación métrica preliminar mediante una referencia visual calibrada. La Figura 6 presenta una imagen de muro con una grieta visible y un marcador ArUco utilizado como escala. A diferencia de los casos anteriores, esta captura no buscó representar una vista general de fachada, sino mostrar el funcionamiento del módulo de medición cuando existe una referencia métrica en la imagen.



Figura 5: a) Demostración visual en vivienda de fachada blanca; b) demostración visual en vivienda de fachada celeste. (2026)

En la Figura 6, el sistema detectó una grieta con orientación vertical asociada a un muro, con una confianza de 96%. El informe preliminar registró la presencia del marcador ArUco DICT_5X5_50, ID 0, con lado conocido de 5 cm, y calculó una estimación del grosor de la grieta de aproximadamente 0,14 cm, valor comprendido dentro del intervalo de 0,10 a 0,15 cm verificado manualmente mediante regla milimetrada. Esta medición corresponde a una estimación preliminar dependiente de la correcta detección del marcador, la perspectiva de captura y la ubicación relativa entre la referencia y la superficie evaluada.

Como comprobación de la estimación métrica en condiciones reales, se llevó a cabo una evaluación sobre 30 fotografías de inspección capturadas en una edificación residencial multifamiliar de 15 pisos en Huancayo, la cual presentó daño tras los eventos sísmicos registrados entre julio y agosto de 2026. Se evaluó un total de 36 patologías de daño superficial identificadas en muros, contrastando las estimaciones del modelo frente a mediciones físicas de control obtenidas in situ mediante una regla milimetrada de precisión. La evaluación experimental registró un error absoluto medio (EAM) de 0,016 centímetros y un desfase relativo promedio de 17,9%, evidenciando una concordancia orientada al apoyo preliminar en el registro visual del daño, sin sustituir una inspección técnica formal.



Figura 6: Demostración de estimación métrica preliminar mediante marcador ArUco. (2026)

4. Discusión

Los resultados muestran que la herramienta detecta grietas visibles y elementos de fachada en imágenes de viviendas de Huancayo como apoyo preliminar al registro visual, no como evaluación estructural integral, pues una grieta no determina por sí sola la vulnerabilidad de una edificación, aunque puede orientar la priorización de inspecciones posteriores. El modelo de fachadas obtuvo un AP de 38,81% y un AP50 de 59,87% en cuadros delimitadores, y un AP de 37,04% y un AP50 de 58,30% en segmentación de instancias. Aunque identificó muros, columnas y vanos, presentó limitaciones para delimitar sus contornos, posiblemente asociadas a variaciones de textura, color, iluminación, perspectiva y materialidad.

En el modelo de grietas, la diferencia entre AP y AP50 fue más marcada. Para cuadros delimitadores se obtuvo un AP de 49,63% y un AP50 de 65,16%, mientras que en segmentación de instancias se alcanzó un AP de 20,03% y un AP50 de 61,13%. Esta diferencia, especialmente pronunciada en segmentación, indica que el modelo puede localizar grietas con un umbral de IoU = .50, pero pierde rendimiento al aplicarse criterios de superposición más estrictos, debido a que pequeñas variaciones en los contornos de grietas delgadas, discontinuas e irregulares reducen considerablemente la coincidencia con la anotación de referencia.

La revisión cualitativa identificó dos patrones recurrentes de error, en primer lugar, algunas máscaras predichas cubrieron solo parcialmente la grieta anotada o sobrepasaron sus contornos, lo que impidió alcanzar la superposición requerida y contribuyó a los conteos de falsos positivos y falsos negativos. En segundo lugar, algunas grietas continuas fueron fragmentadas en varias detecciones, generando segmentos adicionales contabilizados como falsos positivos. Con menor frecuencia, las texturas sobresalientes o rugosas que producían sombras fueron confundidas con grietas. Para reducir las duplicidades causadas por la fragmentación, el posprocesamiento fusionó máscaras superpuestas o contenidas correspondientes a un mismo trazo visible. Este procedimiento reorganizó las predicciones utilizadas en las métricas operativas, sin modificar los pesos del modelo ni las métricas COCO AP y AR, calculadas sobre las predicciones originales.

Frente a esta limitación, existen enfoques alternativos orientados específicamente a la segmentación *pixel-wise*. Pandey y Mishra (2025) señalan que los métodos de aprendizaje profundo para grietas comprenden tareas de clasificación, detección y segmentación, entre las que destaca el uso de arquitecturas como U-Net. Mientras Mask R-CNN detecta cada objeto y genera posteriormente una máscara asociada, los enfoques *pixel-wise* clasifican directamente cada píxel como grieta o fondo. Esta estrategia puede favorecer la segmentación de grietas finas o continuas al no depender de la detección previa de una instancia; sin embargo, no proporciona por sí sola la separación individual, los cuadros delimitadores, las máscaras por instancia y las puntuaciones de confianza ofrecidas por Mask R-CNN.

Respecto a la elección de la arquitectura, los modelos de una etapa como YOLOv8 ofrecen ventajas en velocidad de inferencia. No obstante, Choi et al. (2024) reportaron que, en detección de grietas de concreto, Mask R-CNN alcanzó mayor consistencia

en la delimitación morfológica respecto a YOLOv8, aunque con menor velocidad de procesamiento. Esta comparación respalda la pertinencia de emplear Mask R-CNN cuando la delimitación espacial de las grietas visibles es prioritaria frente al procesamiento en tiempo real. Asimismo, la estimación métrica mediante marcador ArUco y PointRend evaluada sobre 36 patologías en Huancayo registró un error absoluto medio de 0,016 centímetros frente a la medición manual, evidenciando concordancia como apoyo preliminar al registro visual.

En conjunto, los resultados muestran que la herramienta puede apoyar el registro visual preliminar, aunque mantiene limitaciones en la segmentación precisa de grietas finas, el control de falsos positivos y la generalización a condiciones visuales más diversas.

5. Conclusiones

Este estudio desarrolló una herramienta basada en *deep learning* para la detección visual de grietas y elementos de fachada en imágenes de viviendas de Huancayo. El sistema, implementado con modelos especializados de Mask R-CNN y Detectron2, permitió identificar muros, columnas, vanos y grietas visibles, además de generar una salida visual e informe preliminar mediante una interfaz web.

Los resultados muestran que el sistema puede apoyar el registro visual de fachadas y la priorización de inspecciones técnicas posteriores. No obstante, su alcance es preliminar: la presencia de una grieta visible no permite determinar por sí sola la vulnerabilidad estructural de una edificación, ya que esta requiere información constructiva, material, geométrica y contextual.

La principal limitación se relaciona con la segmentación precisa de grietas finas y la variabilidad visual de los elementos de fachada. Por ello, sería recomendable explorar con enfoques de segmentación *pixel-wise* para grietas, así como con conjuntos de datos más amplios y diversos para mejorar la detección de elementos de fachada en diferentes condiciones de materialidad, iluminación y perspectiva.

Conflicto de intereses. Los autores declaran no tener conflictos de intereses.

© Derechos de autor: Paolo Martin Lopez-Gomez, Rafael Jhon Flores-Corilloclla, Adolfo Gustavo Concha-Flores, 2027.
© Derechos de autor de la edición: *Estoa*, 2027.

6. Referencias bibliográficas

Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., & Zou, J. (2019). Gradio: Hassle-free sharing and testing of ML models in the wild. *arXiv*. https://arxiv.org/abs/1906.02569

Aravena, P., Geiß, C., Aguirre, P., Santa María, H., Merino, Y., & Taubenböck, H. (2021). Automated building characterization for seismic risk assessment using street-level imagery and deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, *180*, 370–386. https://doi.org/10.1016/j.isprsjprs.2021.07.004

Bai, Y., Zha, B., Sezen, H., & Yilmaz, A. (2023). Engineering deep learning methods on automatic detection of damage in infrastructure due to extreme events. *Structural Health Monitoring*, *22*(1), 338-352. https://doi.org/10.1177/14759217221083649

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A.-L., Deng, D., & Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery*, *13*(2), e1484. https://doi.org/10.1002/widm.1484

Cha, Y.-J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, *32*(5), 361–378. https://doi.org/10.1111/mice.12263

Cha, Y.-J., Ali, R., Lewis, J., & Büyüköztürk, O. (2024). Deep learning-based structural health monitoring. *Automation in Construction*, *161*, 105328. https://doi.org/10.1016/j.autcon.2024.105328

Choi, Y., Bae, B., Han, T. H., & Ahn, J. (2024). Application of Mask R-CNN and YOLOv8 algorithms for concrete crack detection. *IEEE Access*, *12*, 165314-165321. https://doi.org/10.1109/ACCESS.2024.3469951

Dong, C., & Catbas, F. N. (2020). A review of computer vision–based structural health monitoring at local and global levels. *Structural Health Monitoring, 20*(2), 692–743. https://doi.org/10.1177/1475921720935585

Ferguson, M., Ak, R., Lee, Y. T., & Law, K. H. (2018). Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning. *Smart and Sustainable Manufacturing Systems*, *2*(1), 137-164. https://doi.org/10.1520/SSMS20180033

García, M. E. (2022). *Riesgo sísmico de las viviendas autoconstruidas en el barrio de Yanama del distrito de Huancayo* [Tesis de maestría, Universidad Nacional del Centro del Perú]. http://hdl.handle.net/20.500.12894/8854

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Gouveia, F., Silva, V., Lopes, J., Moreira, R. S., Torres, J. M., & Guerreiro, M. S. (2024). Automated identification of building features with deep learning for risk analysis. *Discover Applied Sciences*, *6*, 466. https://doi.org/10.1007/s42452-024-06070-2

Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*, *6*(4), 312–315. https://doi.org/10.1016/j.icte.2020.04.010

Kirillov, A., Wu, Y., He, K., & Girshick, R. (2020). PointRend: Image segmentation as rendering. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 9796–9805). https://doi.org/10.1109/CVPR42600.2020.00982

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., & Fieguth, P. (2015). A review on computer vision-based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*, *29*(2), 196–210. https://doi.org/10.1016/j.aei.2015.01.008

Kong, C., & Petchsasithon, A. (2024). Information acquisition and seismic damage prediction of masonry structures in rural areas based on UAV inclined photogrammetry. *Journal of Information Systems Engineering and Management*, *9*(1), 25183. https://doi.org/10.55267/iadt.07.14315

Miseta, T., Fodor, A., & Vathy-Fogarassy, Á. (2024). Surpassing early stopping: A novel correlation-based stopping criterion for neural networks. *Neurocomputing*, *567*, 127028. https://doi.org/10.1016/j.neucom.2023.127028

Mohtasham, M., Vahidnia, S., Ghasemzadeh, L., Ozturk, Y. E., Yuvalaklioglu, M., Akin, S., & Ure, N. K. (2020). Deep-learning-based crack detection with applications for the structural health monitoring of gas turbines. *Structural Health Monitoring*, *19*(5), 1440–1452. https://doi.org/10.1177/1475921719883202

Nakamura, K., Derbel, B., Won, K.-J., & Hong, B.-W. (2021). Learning-rate annealing methods for deep neural networks. *Electronics*, *10*(16), 2029. https://doi.org/10.3390/electronics10162029

Obi-Obuoha, A., Rizama, V. S., Okafor, I., Ovwenkekpere, H. E., Obe, K., & Ekundayo, J. (2024). Real-time traffic object detection using Detectron 2 with Faster R-CNN. *World Journal of Advanced Research and Reviews*, *24*(02), 2173–2189. https://doi.org/10.30574/wjarr.2024.24.2.3559

Osman, A. H. (2025). *Cracks Computer Vision Dataset* [Open source dataset]. Roboflow Universe. https://universe.roboflow.com/albaraa-hussein-osman/cracks-tfqny

Padsumbiya, M., Brahmbhatt, V., & Thakkar, S. P. (2022). Automatic crack detection using convolutional neural network. *Journal of Soft Computing in Civil Engineering*, *6*(3), 1–17. https://doi.org/10.22115/scce.2022.325596.1397

Palomino Ojeda, J., Quiñones Huatangari, L., Revilla Arce, J., Arce Fernández, N., Gonzales Santiesteban, M., & Martínez Serrano, M. (2024). Assembly algorithms for seismic vulnerability estimation in confined masonry dwellings. *International Journal of Safety and Security Engineering*, *14*(3), 967-984. https://doi.org/10.18280/ijsse.140327

Pandey, V., & Mishra, S. S. (2025). A review of image-based deep learning methods for crack detection. *Multimedia Tools and Applications*, *84*, 35469–35511. https://doi.org/10.1007/s11042-025-20729-x

Roy, S., Yogi, B., Majumdar, R., Ghosh, P., & Das, S. K. (2025). Deep learning-based crack detection and prediction for structural health monitoring. *Discover Applied Sciences*, *7*, 674. https://doi.org/10.1007/s42452-025-07272-y

Tavera, H., Nina, V., Bernal, I., & Mercado, A. (2025). Analysis and evaluation of recent deformation on the Huaytapallana Fault (Central Region of Peru) based on data obtained from a local seismic network. *Seismological Research Letters*, *96*(3), 1687–1702. https://doi.org/10.1785/0220240257

Vilares Ferro, M., Doval Mosquera, Y., Ribadas Pena, F. J., & Darriba Bilbao, V. M. (2023). Early stopping by correlating online indicators in neural networks. *Neural Networks*, *159*, 109–124. https://doi.org/10.1016/j.neunet.2022.11.035

Wang, C., Antos, S., & Triveno, L. (2021). Automatic detection of unreinforced masonry buildings from street view images using deep learning-based image segmentation. *Automation in Construction*, *132*, 103968. https://doi.org/10.1016/j.autcon.2021.103968

Wang, J., Xu, G., Yan, F., Wang, J., & Wang, Z. (2023). Defect transformer: An efficient hybrid transformer architecture for surface defect detection. *Measurement*, *211*, 112614. https://doi.org/10.1016/j.measurement.2023.112614

Zhu, J., Zhong, J., Ma, T., Huang, X., Zhang, W., & Zhou, Y. (2022). Pavement distress detection using convolutional neural networks with images captured via UAV. *Automation in Construction*, *133*, 103991. https://doi.org/10.1016/j.autcon.2021.103991