

Tipo de artículo: Artículo original

# Metodología para el análisis y evaluación de daños en escenas aéreas

## *Methodology for the analysis and evaluation of damage in aerial scenes*

Luis Alberto Díaz Borge<sup>1\*</sup> , <https://orcid.org/0000-0002-9081-7625>  
Yudivián Almeida Cruz<sup>2</sup> , <https://orcid.org/0000-0002-2345-1387>

<sup>1</sup> Centro de estudios de Matemática Computacional. Facultad de Ciencias y Tecnologías Computacionales, Universidad de las Ciencias Informáticas.

<sup>2</sup> Centro de estudios de Inteligencia Artificial. Facultad de Matemática y Computación. Universidad de La Habana. E-Mail: [yudy@matcom.uh.cu](mailto:yudy@matcom.uh.cu)

\* Autor para correspondencia: [l.diazborg@gmail.com](mailto:l.diazborg@gmail.com)

### Resumen

Los desastres causados por fenómenos naturales han existido siempre a través de la historia de la humanidad, siendo cada vez más frecuentes y catastróficos. Debido a esta tendencia creciente, se hace de suma importancia su análisis y prevención para evitar males mayores. Gracias al avance científico-tecnológico, hoy en día se cuenta con grandes volúmenes de información referentes a desastres naturales. Una vez ocurrida la catástrofe se necesitan datos precisos y oportunos respecto a los daños causados con el objetivo de brindar una respuesta eficaz. En este trabajo se presenta una propuesta basada en redes neuronales para analizar y marcar daños en zonas afectadas al paso de catástrofes, haciendo uso de imágenes satelitales pre- y post- desastre. También se realizan análisis de los resultados alcanzados, así como algunas comparaciones con otros trabajos existentes sobre el tema.

**Palabras clave:** aprendizaje de máquina; aprendizaje profundo; red neuronal convolucional; imagen satelital; análisis de daños.

### Abstract

*Disasters caused by natural phenomena have always existed throughout the history of mankind, becoming more and more frequent and catastrophic. Due to this growing trend, its analysis and prevention is extremely important to avoid greater evils. Thanks to scientific and technological advances, today there are large volumes of information regarding natural disasters. Once the catastrophe has occurred, accurate and timely data regarding the damage caused is needed in order to provide an effective response. This paper presents a proposal based on neural networks to analyze and mark damage in areas affected by catastrophes, making use of pre- and post-disaster satellite images. Analysis of the results achieved is also carried out, as well as some comparisons with other existing works on the subject.*

**Keywords:** machine learning; deep learning; convolutional neural network; satellite imagery; damage analysis.

**Recibido:** 08/01/2021

**Aceptado:** 29/10/2021



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

## Introducción

Desde la invención de los satélites artificiales a mediados del siglo XX, el estudio y observación de la Tierra ha alcanzado niveles superiores. En zonas remotas como las del Ártico, puede proporcionar datos no solo de la extensión de hielo, sino también del grosor, que es un elemento de evaluación y previsión de la evolución del sistema de hielo en la zona ártica y antártica. El satélite nos da la oportunidad también de ver la evolución de los bosques, las actividades como la deforestación, el aumento del nivel del mar, la erosión costera y la contaminación marina. Todas estas cosas se ven fácilmente con los satélites, explica Simonetta Cheli, jefa de la oficina de coordinación en la Dirección de los Programas de Observación de la Tierra de la ESA, ESRIN (Aguirre Gómez, 2018).

Gracias a la era de los satélites, se está en una mejor posición para comprender las complejidades del planeta tierra. Los satélites de hoy se utilizan para pronosticar el clima, responder preguntas importantes de ciencias de la Tierra, proporcionar información esencial para mejorar las prácticas agrícolas, para la seguridad marítima y todo tipo de aplicaciones cotidianas. Así, hoy en día, en el caso de desastres, se proporciona acceso rápido alrededor de todo el mundo a datos satelitales para ayudar a las autoridades en la gestión de estos fenómenos como pueden ser una inundación, un terremoto o una erupción volcánica. Todo esto hace que la red de satélites de observación de la Tierra y las imágenes satelitales y aéreas desempeñe un papel importante en la adquisición de información acerca del planeta o sobre objetos en la superficie terrestre (Bohórquez, 2013).

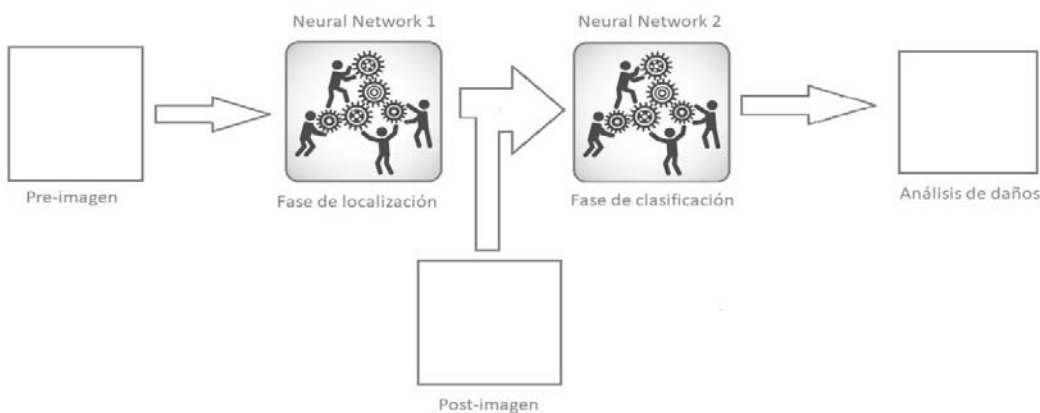
Atendiendo a los elementos antes descritos, se define como objetivo de la presente investigación desarrollar una propuesta basada en redes neuronales para analizar y marcar daños en zonas afectadas al paso de catástrofes, haciendo uso de imágenes satelitales pre- y post- desastre. También se realizan análisis de los resultados alcanzados, así como algunas comparaciones con otros trabajos existentes sobre el tema.

## Materiales y métodos

La propuesta para analizar y evaluar daños en escenas aéreas está basada en modelos DNNs, específicamente en redes neuronales basadas en la arquitectura U-Net (Ronneberger et al., 2015). Dicha propuesta se divide en dos grandes fases.



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)



**Figura 1.** Etapas o fases que se llevan a cabo en la metodología propuesta.

## Fases de la Propuesta

La metodología o proceso de análisis de daños se divide en dos grandes fases. En la primera, se localizan y delimitan los objetos dentro de la pre-imagen (imagen satelital antes de la catástrofe). Con esta información, y junto a la post-imagen (imagen satelital después de la catástrofe), en la segunda se realiza la clasificación de daños (ver Figura 1).

Una vez que se tienen las Fases 1 y 2 listas, solo faltaría ponerlo todo junto. Para realizar el análisis de daños a alguna zona en específico, la pre-imagen es analizada durante la Fase 1. Luego, la concatenación de la post-imagen y la salida de la Fase 1 son procesadas durante la Fase 2. De esta manera se tienen los resultados en la salida de la Fase 2 en forma de matriz. Los edificios sin daños son marcados con el valor 1, daños menores con el valor 2, daños mayores con el valor 3, mientras que destruidos son marcados con el valor 4.

### Fase 1: Localización

En la primera fase, se localizan y delimitan los objetos dentro de la pre-imagen que tiene como entrada. La salida de esta fase es una matriz del mismo tamaño de la imagen de entrada, con la posición de los edificios que se encuentran en la pre-imagen. Para resolver esta tarea, se presentan dos posibles variantes utilizando un enfoque basado en redes neuronales.

#### Variante 1

La primera variante es una red neuronal convolucional basada en la arquitectura UNet (Ronneberger et al., 2015) (ver Figura 2), con 7 salidas diferentes (ver Figura 3). Cada valor de cada salida representa la probabilidad de que existan edificios en esa “zona” de la imagen de entrada. Por ejemplo, si la salida 1, que tiene tamaño 8x8, tiene valor 0 en la primera posición (esquina superior izquierda de la salida), entonces significa que la subimagen de tamaño 64x64 ubicada en la esquina superior izquierda de la imagen de entrada, no contiene ningún edificio o parte de alguno.



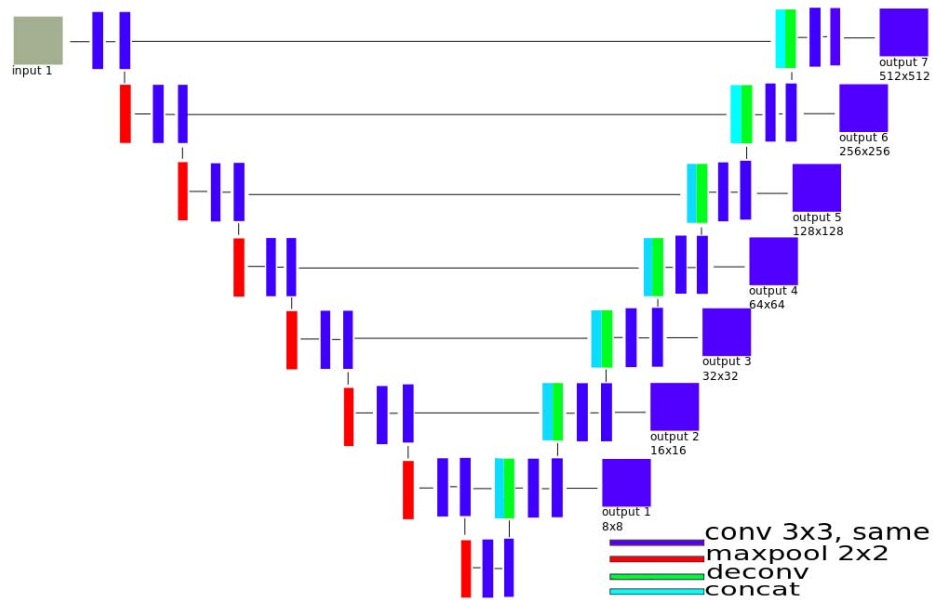


Figura 2. Red neuronal basada en la arquitectura U-Net.

Fuente: (Ronneberger et al., 2015)

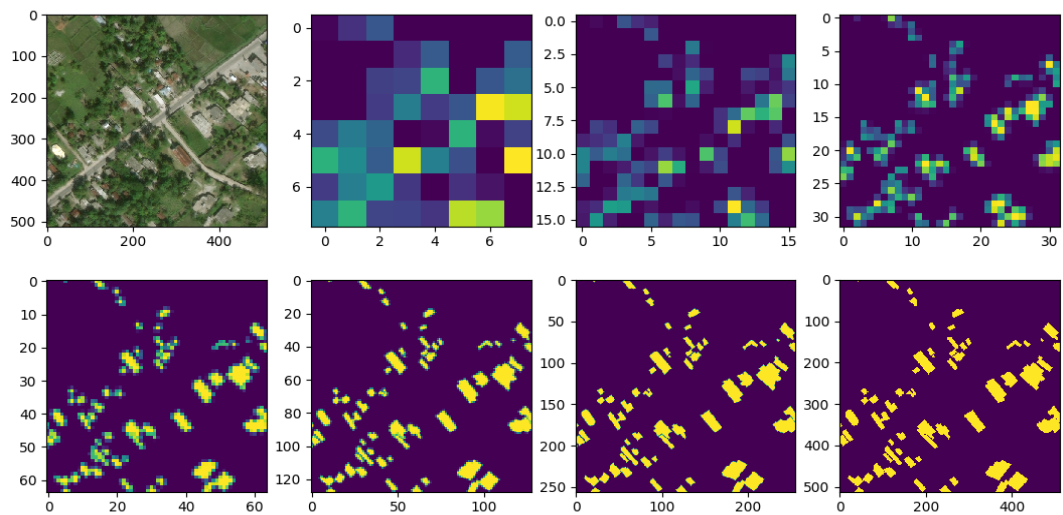


Figura 3. Representación de la entrada y salida de la red neuronal referida en la sección 2.2.1. De izquierda - derecha, arriba - abajo: Imagen antes del paso de la catástrofe (pre-imagen), salida 1, salida 2, salida 3, salida 4, salida 5, salida 6, salida 7.



Esta obra está bajo una licencia *Creative Commons* de tipo *Atribución 4.0 Internacional* (CC BY 4.0)

De igual manera, si la salida 7, que tiene tamaño 512 x 512, tiene valor 1 en la primera posición, entonces significa que el píxel de la esquina superior izquierda de la imagen de entrada representa algún edificio o parte de él. Esta última salida contiene la información que representa la localización de las edificaciones y, por ende, representa la salida final de la fase 1. Un elemento a resaltar es que las salidas van de menor tamaño y descripción, a mayor tamaño y nivel de detalle. Por ello, el modelo enfrenta simultáneamente desde pequeños problemas y fáciles de resolver hasta problemas de mayor tamaño y complejidad. Otro punto a destacar, es cuando se efectúa el proceso *back-propagation* o propagación hacia atrás, durante el cual se rectifica el modelo en dirección del gradiente que apunta hacia una mejor predicción. Cuando se utiliza la frase “entrenamiento de una red neuronal” se habla de dicho proceso. Al rectificar una salida de mayor tamaño, se están rectificando las de menor tamaño, debido a la naturaleza del proceso y la arquitectura del modelo. Rectificar una salida más simple, contribuye a la solución de una más compleja indirectamente, debido al comparezco entre salidas.

## Variante 2

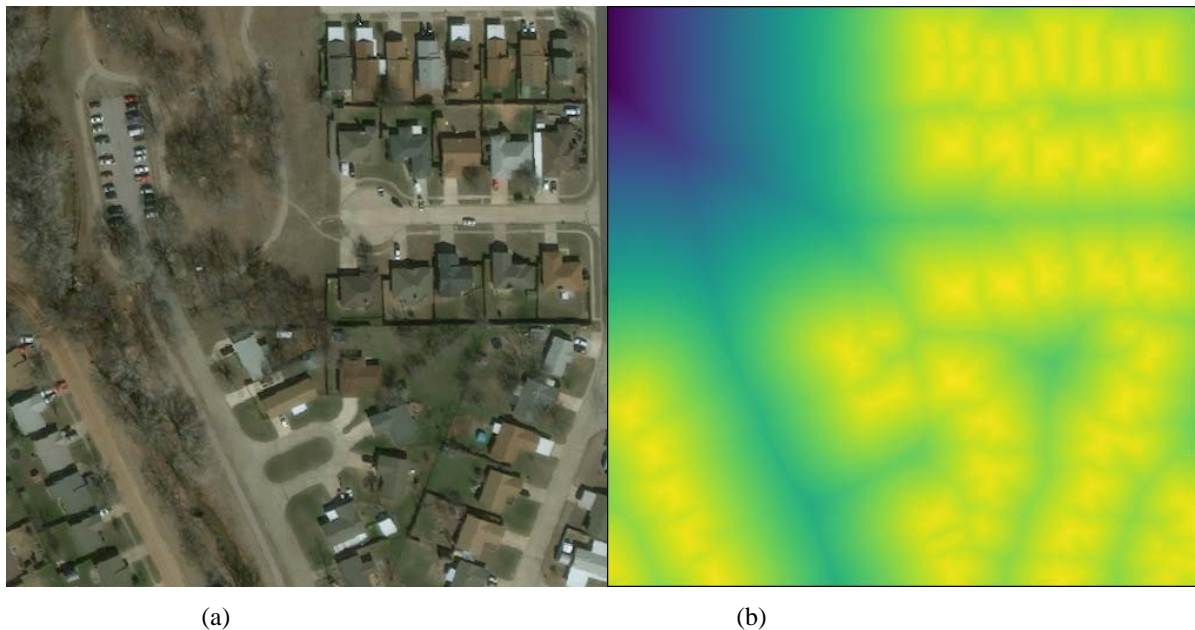
La segunda variante, también usa una red neuronal convolucional basada en la arquitectura U-Net (Ronneberger et al., 2015). La salida (ver Figura 4) es un matriz de tamaño 512x512, donde cada valor representa la distancia hacia el borde del edificio más cercano. Posiciones interiores toman valores negativos, mientras que posiciones exteriores valores positivos. Posiciones en el borde del edificio toman valor 0. Esta representación obliga a la red a que aprenda no solo a delimitar, sino también a saber que tan dentro del polígono o fuera del mismo se encuentra una determinada posición. Además, la misma está normalizada usando la función *softsign*.

De esta manera, para posiciones exteriores y muy alejadas, por ejemplo, lo importante sea saber si están lejos y no centrarse en “exactamente que tanto lejos” están del borde de algún polígono. Para calcular numéricamente qué tan errónea es la predicción de esta variante se usa la siguiente función de pérdida:

$$Loss = \sum_{i=1}^n |softsign(y_{true}) - softsign(y_{pred})| \quad (1)$$

Errores en posiciones tanto interiores como exteriores, cercanas a bordes de polígonos, son mayormente penalizados. De esta manera, el modelo es más exhaustivo en dicha zona. Teniendo en cuenta las características de la representación de la salida de la red neuronal, valores cercanos a cero significan posiciones cercanas a bordes. Si se tienen los valores 1 (valor real) y 2 (valor predicho), la pérdida aumenta en 0.17 aproximadamente. Por otra parte, si se tienen los valores 10 (valor real) y 15 (valor predicho), la pérdida aumentaría en 0.03 aproximadamente.





**Figura 4.** Representación de la entrada y salida de la red neuronal referida en la sección 2.2.2. (a) Imagen antes del paso de la catástrofe. (b) Transformación de distancia de señal. Píxeles exteriores toman color oscuro intenso (valores negativos). Píxeles interiores toman color amarillo intenso (valores positivos).

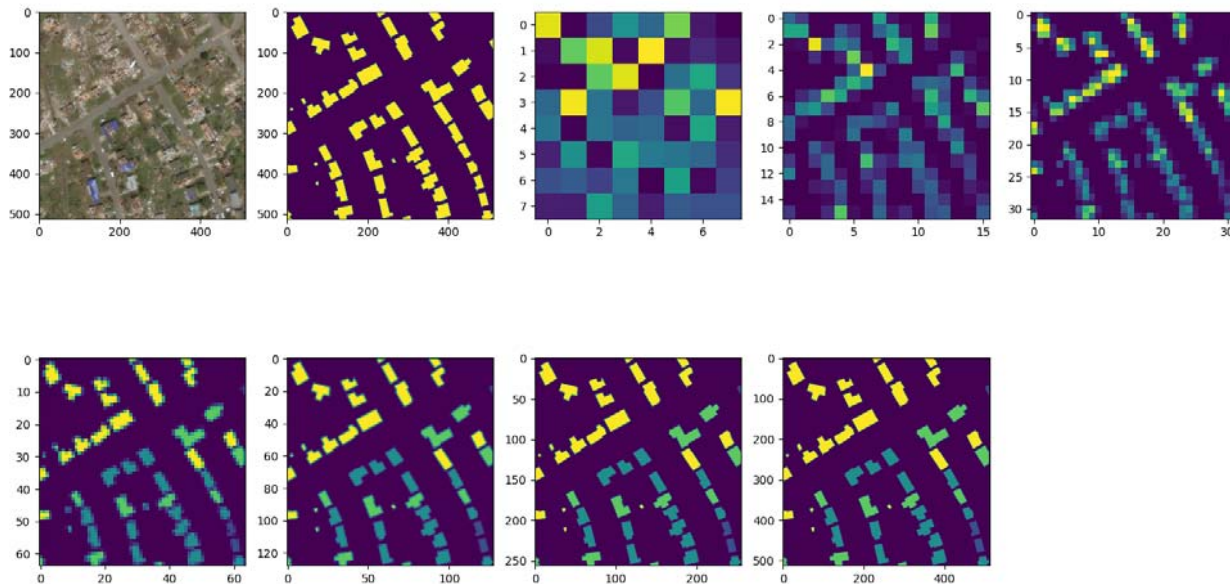
## Fase 2: Clasificación

En la segunda fase se analizan y clasifican los objetos detectados durante la Fase 1. Para ello se presenta un modelo también basado en la arquitectura U-Net (Ronneberger et al., 2015). La entrada tiene tamaño  $512 \times 512 \times 4$ , compuesta por la concatenación de la post-imagen y la matriz con la localización de los edificios. En la salida, al igual que en la sección 2.2.1, se hace uso de la misma técnica (ver Figura 5), teniendo 7 salidas diferentes, donde la última salida representa la salida final de la fase. En esta fase se presenta una característica muy común: ordinalidad de los datos a predecir (*no-building*, *undamage-building*, *minor-damage*, *mayor-damage*, *destroyed*) (Agregti, 2003).

Comprender la naturaleza de la salida para cualquier modelo predictivo es esencial para construir un buen estimador: regresión, clasificación, regresión ordinal, regresión de *poisson*, etc. La regresión ordinal es similar a la clasificación en que las etiquetas  $r_1 \dots r_k$  son categóricas, y similares a la regresión en que las etiquetas se ordenan  $r_1 \ll \dots \ll r_k$ , aún cuando la distancia entre rangos no esté definida (Gutiérrez et al., 2015). Para resolver estos aspectos, cada una de las etiquetas antes dichas son representadas mediante valores pertenecientes al intervalo continuo  $[0, 4]$ , donde 0 significa ‘*no-building*’, mientras que 4 ‘*destroyed*’ o daños máximos (ver Figura 5). Como función de pérdida se usa ‘error



cuadrático medio'. De esta manera, se está convirtiendo el problema inicial en un problema de regresión. Para saber a cuál clase pertenece una predicción en la salida 7, se utiliza el entero más cercano.



**Figura 5.** Representación de la entrada y salida de la red neuronal referida en la sección 2.3. De izquierda - derecha, arriba - abajo: Imagen después del paso de la catástrofe (post-imagen), localización de las edificaciones, salida 1, salida 2, salida 3, salida 4, salida 5, salida 6, salida 7

## Resultados y discusión

En esta sección se detalla el conjunto de datos utilizados en los diferentes modelos DNNs. Se describen las técnicas de pre procesamiento que se llevaron a cabo. Se analiza el desempeño de los modelos durante el entrenamiento, siguiendo las métricas de pérdida y precisión.

### Dataset

Las escenas aéreas o imágenes satelitales usadas en este trabajo fueron extraídas de xBD (Gupta et al., 2019). xBD provee imágenes anotadas pre- y post-, de diferentes tipos de desastres. En este trabajo se filtraron las imágenes relacionadas a desastres por fuertes vientos e incendios. Por cuestiones de tiempo y recursos cómputos, no se utilizaron el resto de fotos relacionadas a desastres como inundaciones o terremotos. Como resultado del pre procesamiento aplicado, se cuenta con un total de 3644 pares de fotos pre- y post- desastre, conteniendo un total de 113366 edificaciones a analizar. De estos, 76796 pertenecen a la categoría ‘undamage-building’, 17030 a ‘minor-



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

damage’, 5881 a ‘mayor-damage’, y 17718 a la categoría ‘destroyed’. La Figura 6 muestra la invarianza presente en los datos.

## Pre procesamiento

Inconvenientemente, los polígonos anotados en xBD (Gupta et al., 2019) potencialmente se superponen o son adyacentes unos con otros. Para resolver dichas deficiencias, primero todo solapamiento se elimina quitando la zona en común y después se reduce el contorno de cada uno en un píxel (ver Figura 8). De esta manera se obtienen polígonos totalmente independientes. Por otra parte, las fotos presentan alto nivel de detalle (escala RGB de 1024x1024 píxeles). Por ello, cada una es dividida formando 4 imágenes de 512 x 512 píxeles como se muestra en la Figura 7. Los polígonos fragmentados por la división de las imágenes no fueron eliminados y, por tanto, pueden existir por partes en bordes de las mismas (en el caso de la post-imagen, los fragmentos conservan su categoría de daño original).

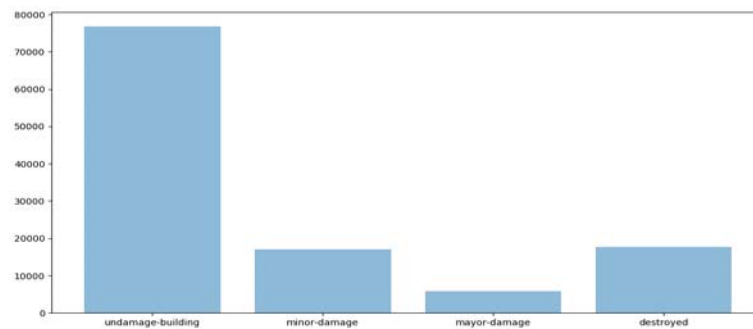


Figura 6. Cantidad de edificios en cada categoría.





**Figura 7.** Splitter. Se obtienen 4 imágenes de 512 x 512, de una imagen de 1024 x 1024.

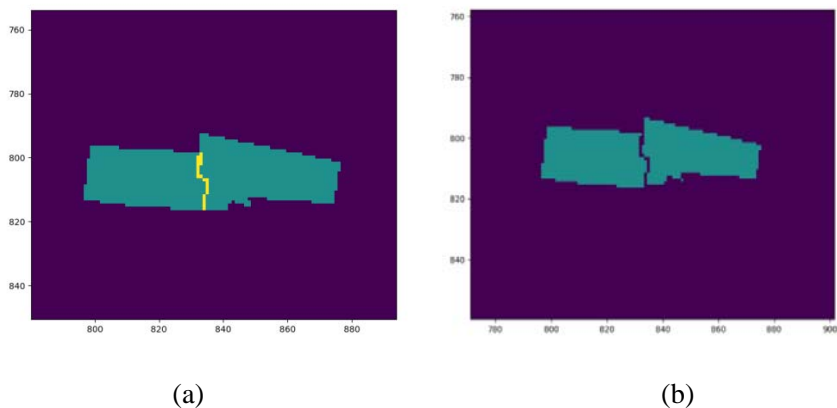


Figura 8: Pre procesamiento. (a) Polígonos sin pre procesar. Píxeles amarillos representan zona de solapamiento. (b) Polígonos después del pre procesamiento.

Después de eliminar solapamientos, encoger polígonos y dividir las imágenes, muchas que originalmente tenían muy poca densidad de edificaciones, resultaron no contener información sustancial para ser utilizadas durante el entrenamiento de los modelos. Debido a dicho motivo, se descartó toda imagen resultante del pre procesamiento que no tuviese al menos el 2.5% de densidad de polígonos.

Para leer las coordenadas de los polígonos anotados en xBD se usaron las herramientas que brinda el paquete *shapely*. Para realizar el pre procesamiento, fueron imprescindibles los paquetes *cv2*, *scipy* y *numpy*. Todos los paquetes mencionados fueron utilizados en el lenguaje de programación Python (versión 3.6).

## Análisis Experimental

Los resultados alcanzados provienen de los modelos *lunet* y *cunet*. Por cuestiones de tiempo y recursos computacionales, la variante explicada en la sección 2.2.2 no es analizada en este trabajo. Ambos modelos (*lunet* y *cunet*) fueron entrenados independientemente. Para formar el conjunto de entrenamiento se tomó el 80% de los 3644 pares de forma aleatoria, y el 20% restante para validación. No se realiza ninguna técnica de aumento de los datos.

Cuando se trabaja con modelos de redes neuronales que tienen pocos parámetros a entrenar, es muy frecuente que sucedan problemas de *underfitting* debido a la poca capacidad de la red. Las Figuras 9 y 10 describen el comportamiento de la pérdida de los modelos *lunet* y *cunet* respectivamente.



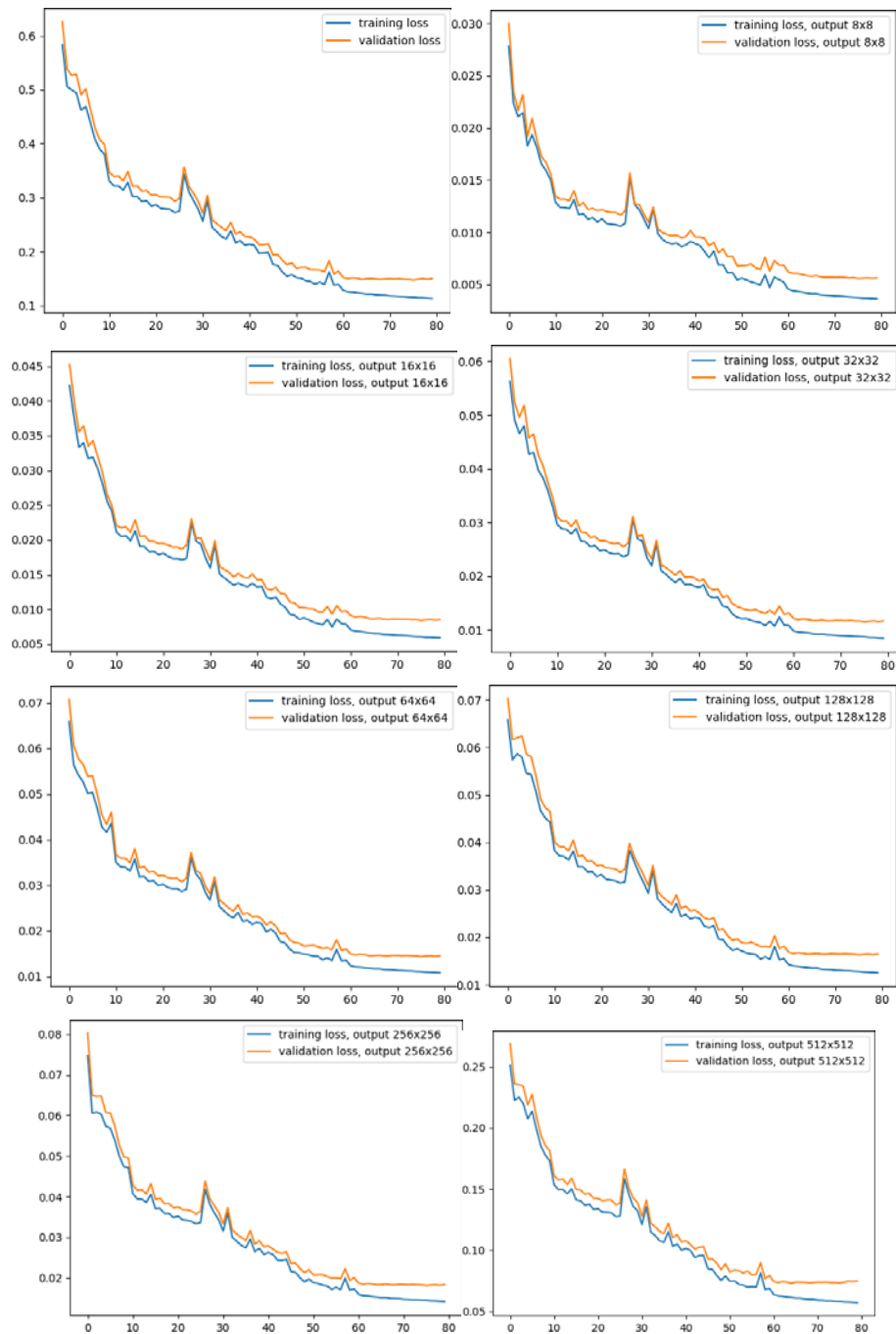


Figura 9. Comportamiento de la pérdida del modelo *lunet* durante el entrenamiento.



Esta obra está bajo una licencia *Creative Commons* de tipo *Atribución 4.0 Internacional* (CC BY 4.0)

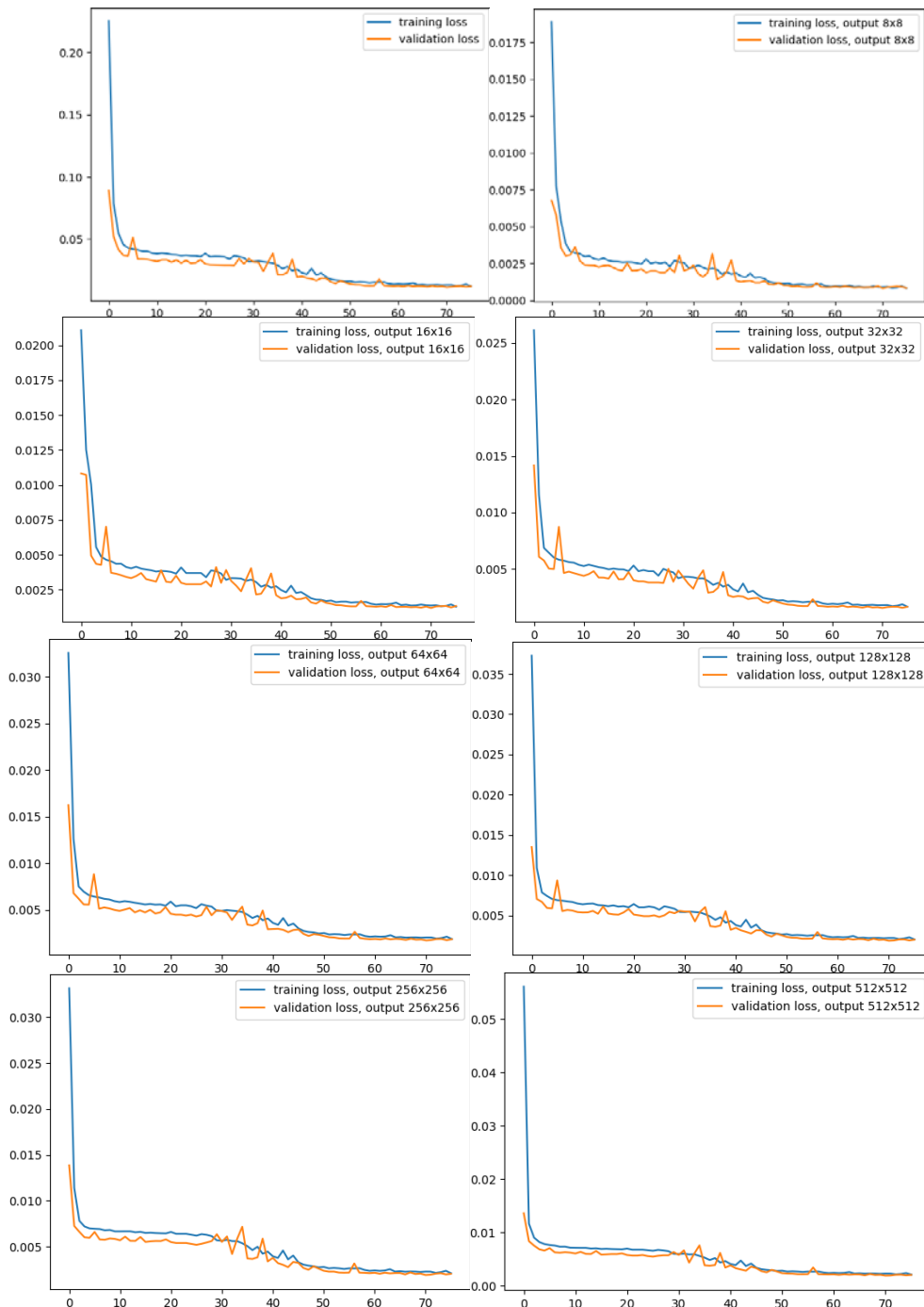


Figura 10. Comportamiento de la pérdida del modelo *cnnet* durante el entrenamiento.



Esta obra está bajo una licencia *Creative Commons* de tipo *Atribución 4.0 Internacional* (CC BY 4.0)

Estas son evidencia del buen desempeño de dichos modelos durante el entrenamiento, a pesar de contar con pocos parámetros entrenables (ver Tabla 1).

**Tabla 1.** Resumen de los resultados alcanzados.

Model	Params	Epochs	Training f1 score	Validation f1 score	Average f1 score
lunet	143347	80	0.839	0.839	binary
cunet	501892	76	0.641 / 0.978	0.627 / 0.979	macro / weighted
lunet + cunet	645239	80   76	0.570 / 0.951	0.561 / 0.953	macro / weighted

Para medir la precisión, se usa la medida F1. El modelo *lunet* tiene como entrada la pre-imagen, y en la salida devuelve las localizaciones de los edificios. El modelo *cunet* recibe como entrada la post-imagen junto con la posición exacta de los edificios, y devuelve la clasificación de los daños. El modelo *lunet + cunet* hace referencia a la metodología. Este, recibe como entrada la pre- y post- imagen, y en la salida devuelve la clasificación de los daños. Las tablas 1 y 2 contiene distintos valores de precisión evaluados en los diferentes modelos. Un elemento a resaltar es la similitud entre los valores de precisión de los modelos clasificadores. A pesar de que la metodología (modelo *lunet + cunet*) tiene desventaja (error que aporta el modelo *lunet* cuando localiza los edificios), su precisión a la hora de clasificar daños es muy cercana a la del modelo *cunet*, el cual recibe con exactitud la posición de los edificios. Esto se debe a la alta precisión que reporta el modelo *lunet* a la hora de localizar los edificios dentro de la pre-imagen.

**Tabla 2.** Valores de las medidas F1s alcanzados por los modelos en las diferentes clases

Model	No-building	No-damage	Minor-damage	Mayor-damage	Destroyed
Cunet	0.999	0.888	0.259	0.189	0.800
Lunet + cunet	0.982	0.754	0.212	0.166	0.692

Un aspecto negativo es que las clases ‘*minor-damage*’ y ‘*mayor-damage*’ son las que brindan menores valores de precisión. La explicación se puede encontrar en la invarianza de los datos. Al ser poca la cantidad de ejemplos para entrenar que contengan cierta clase, al modelo le resulta más difícil encontrar un separador no lineal, que prediga con exactitud dicha clase. En cada pasada por el conjunto de datos, el modelo estaría aprendiendo más de otras clases que tienen mayor frecuencia.

Para medir la medida F1 en los modelos clasificadores multi-clases se usa *macro-averaging* y *weighted-averaging*. En el primer caso todas las clases toman igual importancia, mientras que en el segundo las clases toman importancia según su número de ocurrencias.



La figura 11 muestra un ejemplo del resultado alcanzado en la Fase 1 (fase de localización) de la metodología. La figura 12 muestra un ejemplo del resultado alcanzado en la Fase 2 (fase de clasificación) de la metodología. En la figura 13 se aprecian los resultados alcanzados con la metodología, aplicada a dos imágenes satelitales pre- y post-desastre.

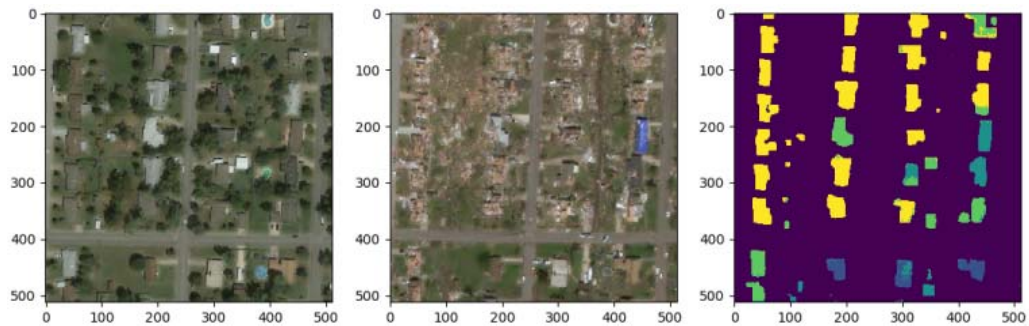


Figura 11. Resultados alcanzados durante la Fase 1

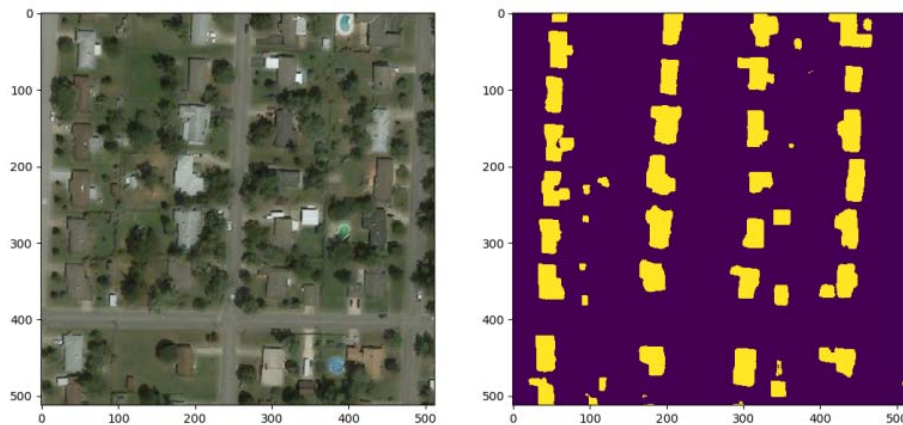


Figura 12. Resultados alcanzados durante la Fase 2



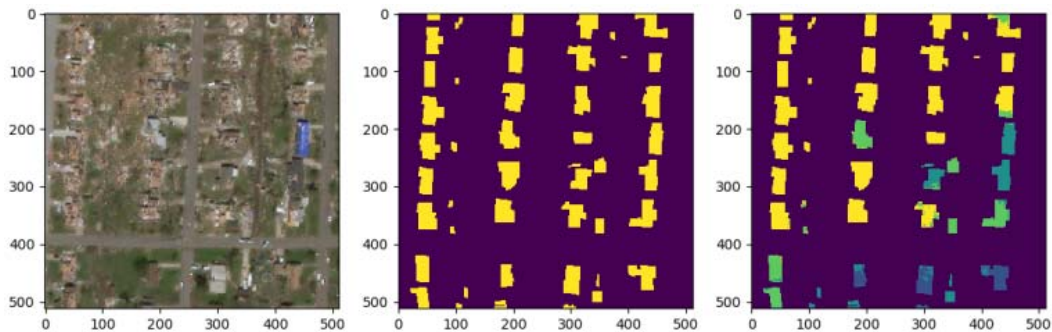


Figura 13. Clasificación de daños obtenidos al aplicar la metodología.

Todos los resultados alcanzados fueron realizados haciendo uso del lenguaje de programación Python (versión3.6). Se utilizó el módulo *sklearn* y *numpy* para el cálculo de los valores de precisión de los modelos. Para el manejo de las redes neuronales, se utilizaron las herramientas que brindan el *framework tensorflow.keras* (2.1.0 es la versión utilizada de *tensorflow*).

## Comparaciones con otros resultados

Existen diferentes trabajos relacionados con el tema principal que se aborda en esta investigación, que igual hacen uso de redes neuronales y están entrenadas usando imágenes tomadas de xBD. Las tablas 3 y 4 muestran los resultados de la metodología así como dos propuestas presentes en la literatura: la que propone Jamyoung Koo, Junghoon Seo et al (Jamyoung Koo et al., 2020) y la que propone Colin Alstad (Alstad, 2020). Es importante mencionar, que estas dos propuestas hacen uso de todas las imágenes presentes en xBD, mientras que la metodología propuesta solo emplea las fotos relacionadas a incendios y fuertes vientos.

Tabla 3. Medidas F1s de modelos que localizan edificios en imágenes satelitales.

Model	Localization binary f1 score
Colin Alstad	0.810
5th place xView2 challenge	0.866
lunet	0.839



**Tabla 4.** Media armónica de las medidas F1s de cada una de las clases, de modelos que clasifican daños en edificios haciendo uso de imágenes satelitales.

Model	Classification harmonic mean F1 score
Colin Alstad	0.660
5th place xView2 challenge	0.726
lunet + cunet	0.296

Jamyoun Koo, Junghoon Seo et al (Jamyoun Koo et al., 2020) propone el modelo **Dual-HRNet** para localización y clasificación, compuesto por dos **HRNets**, y bloques de fusión. Cada par de features de cada **HRNet** es fusionado mediante un bloque de fusión. La propuesta de Colin Alstad está basada en la arquitectura U-Net con 9 capas down-sampling.

Como se observa, la metodología propuesta ofrece mayores resultados competitivos en la localización de edificios. No tanto es así en la clasificación de los daños, debido al reducido tamaño del conjunto de entrenamiento además de la invarianza de los datos. Igual se reportan altos valores de precisión en la predicción de daños en edificios totalmente destruidos. Esto la muestra como una variante promisoría para la evaluación de zonas luego de la ocurrencia de desastres.

La métrica usada en la Tabla 4 se calcula tomando la media armónica de las medidas F1s de las 4 clases ‘no-damage’, ‘minor-damage’, ‘major-damage’ y ‘destroyed’.

## Conclusiones

En este trabajo se propone una metodología para detectar y clasificar automáticamente edificios dañados en imágenes de satélite pre- y post- desastre. La metodología se divide en dos grandes Fases: Localización y Clasificación. En cada fase, se hace uso de redes neuronales basadas en la arquitectura U-Net . En la Fase de Localización se presentan dos posibles variantes, de las cuales se valida una.

Como resultado de la experimentación, el modelo *lunet* con solo 80 pasadas por el dataset y tener un total de no más de 144000 parámetros a entrenar, alcanza niveles de precisión por encima del 80%. Por su parte, la metodología en sus dos fases (*lunet + cunet*) obtuvo valores de precisión macro F1 score de 0.561 y *weighted* F1 score de 0.953. Finalmente la metodología fue comparada con otras dos propuestas mostrándose competitiva con respecto a estas.

## Conflictos de intereses



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)

Los autores de la presente investigación declaran que no poseen conflictos de intereses

### Contribución de los autores

1. Conceptualización: Luis Alberto Díaz Borge, Yudivián Almeida Cruz.
2. Curación de datos: Luis Alberto Díaz Borge.
3. Análisis formal: Luis Alberto Díaz Borge, Yudivián Almeida Cruz.
4. Adquisición de fondos: Luis Alberto Díaz Borge.
5. Investigación: Luis Alberto Díaz Borge, Yudivián Almeida Cruz.
6. Metodología: Luis Alberto Díaz Borge, Yudivián Almeida Cruz.
7. Administración del proyecto: Luis Alberto Díaz Borge.
8. Recursos: Luis Alberto Díaz Borge.
9. Software: Luis Alberto Díaz Borge.
10. Supervisión: Luis Alberto Díaz Borge.
11. Validación: Luis Alberto Díaz Borge.
12. Visualización: Luis Alberto Díaz Borge.
13. Redacción – borrador original: Luis Alberto Díaz Borge, Yudivián Almeida Cruz.
14. Redacción – revisión y edición: Luis Alberto Díaz Borge, Yudivián Almeida Cruz.

### Financiamiento

La presente investigación no requirió fuente de financiamiento externa.

### Referencias

- Agresti, A. (2003). *Categorical data analysis* (Vol. 482). John Wiley & Sons.  
<https://www.academia.edu/download/45095661/AGRESTI.PDF>
- Aguirre Gómez, R. (2018). Laboratorio Nacional de Observación de la Tierra (LANOT). *Investigaciones geográficas*(96), 0-0.
- Alstad, C. (2020). A look at ibm's approach to a challenge on using satellite imagery to identify buildings and rate the amount of damage they suffered after a natural disaster. <https://www.ibm.com/cloud/blog/the-xview2-ai-challenge>



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional** (CC BY 4.0)



- Bohórquez, J. E. T. (2013). Evaluación de la vulnerabilidad social ante amenazas naturales en Manzanillo (Colima). Un aporte de método. *Investigaciones Geográficas, Boletín del Instituto de Geografía*, 2013(81), 79-93. <https://www.sciencedirect.com/science/article/pii/S0188461113727705>
- Gupta, R., Goodman, B., Patel, N., Hosfelt, R., Sajeev, S., Heim, E., Doshi, J., Lucas, K., Choset, H., & Gaston, M. (2019). Creating xBD: A dataset for assessing building damage from satellite imagery. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops,
- Gutiérrez, P. A., Perez-Ortiz, M., Sanchez-Monedero, J., Fernandez-Navarro, F., & Hervas-Martinez, C. (2015). Ordinal regression methods: survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering*, 28(1), 127-146. <http://orca.cf.ac.uk/112802/1/2016-Ordinal%20regression%20methods:%20Survey%20and%20experimental%20study.pdf>
- Jamyoun Koo, Junghoon Seo, Kwangjin Yoon, & Jeon, T. (2020). Dual-hrnet for building localization and damage classification. [https://github.com/DIUx-xView/xView2\\_fifth\\_place/blob/master/figures/xView2\\_White\\_Paper\\_SI\\_Analytics.pdf](https://github.com/DIUx-xView/xView2_fifth_place/blob/master/figures/xView2_White_Paper_SI_Analytics.pdf).
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention,

