Detección de Instalaciones con Machine Learning

Jesús Santos Capote, Kenny Villalobos Morales, Jorge Soler González, Abraham González Rivero, Rainel Fernández Abreu, Eduardo García Maleta

Facultad de Matemática y Computación, Universidad de La Habana, La Habana, Cuba

Keywords: Detección de Objetos · Clasificación de Imágenes · Machine Learning · Pytorch · Keras · Tensorflow · Faster-RCNN

1. Intoducción

La detección de instalaciones es un tema de gran relevancia en diversas áreas, como la ingeniería, la construcción, la planificación urbana y la gestión de recursos naturales, entre otras. La detección precisa y eficiente de estas instalaciones es esencial para la toma de decisiones informadas y la optimización de los recursos.

Anteriormente, la detección de instalaciones se ha realizado principalmente de forma manual o mediante técnicas de procesamiento de imágenes tradicionales, lo que puede resultar costoso y poco efectivo. Con el auge de la inteligencia artificial y el aprendizaje automático, se ha abierto la posibilidad de aplicar estas técnicas para mejorar la detección de instalaciones. Particularmente, dentro del aprendizaje automático está el aprendizaje profundo(Deep learning), que no son más que modelos que repesentan los datos en diferentes niveles de abstracción por medio de múltiples capas de procesamiento, estos han logrado un éxito importante en la detección y clasifficación de instalaciones mediante la combinación de grandes modelos de redes neuronales, llamados redes neuronales convolucionales (CNN), con unidades potentes de procesamiento(GPUs). Este éxito ha provocado una revolución en la comprensión de imágenes, y las principales empresas de tecnología, incluidas Google, Microsoft y Facebook, ya han implementado productos y servicios basados en CNN.

Por lo general estos modelos utilizan imágenes satelitales multiespectrales para entrenarse, pero estas son difíciles de obtener, y tienen precios elevados. Por lo tanto el objetivo principal de este trabajo es comprobar la factibilidad de abordar el problema de detección de instalaciones utilizando imágenes satelitales en RGB.

Para la solución de este trabajo seguimos los siguientes pasos descritos a lo largo del mismo:

- Realizar una revisión exhaustiva de la literatura existente sobre detección de instalaciones y técnicas de machine learning aplicadas a este problema.
- Desarrollar una o varias propuestas de modelos de aprendizaje automático que permitan detectar instalaciones en las imágenes de prueba.

- Evaluar el desempeño del modelo a través de métricas de evaluación adecuadas y compararlo con otros enfoques existentes.

Los modelos implementados tratarán de identificar un subconjunto pequeño de tipos de instalaciones. Se proponen cuatro modelos, dos de clasificación de imágenes, uno de detección de objetos y uno segmentación con clustering.

El modelo de detección objetos propuesto es el Faster-RCNN [9]. Se entrenó con imágenes de múltiples tipos de estadios y se probó su rendimiento con imágenes de estadios de beibol cubanos. Dentro de su conjunto de entrenamiento no hay imágenes del territorio nacional. Los modelos de segmentación con clustering también son probados con el mismo conjunto de prueba de estadios cubanos.

Ambos modelos de clasificación son un ensamblado de distintas redes convolucionales. Una de ellas está compuesta por las redes InceptionV3[18], Xception[7], ResNet[17] y DenseNet[19]. Se entrenó para reconocer zoológicos, aeropuertos, cementerios, estadios y parques. El otro modelo de clasificación está compuesto por las redes InceptionV3 y DensNet y fue entrenado para reconocer aeropuertos y zoológicos.

2. Estado del Arte

Actualmente para resolver la problemática se utilizan técnicas de aprendizaje profundo, específicamente redes neuronales convolucionales, para la detección y clasificación de edificios e instalaciones en imágenes de teledetección y satélite. Estas redes neuronales convolucionales permiten extraer características relevantes de las imágenes y clasificarlas en diferentes categorías.

Algunos artículos utilizan técnicas de reducción de dimensionalidad, como Análisis de Componentes Principales (PCA), para extraer características de las imágenes y reducir la complejidad de las mismas. PCA selecciona las características más importantes de la imagen y las utiliza para entrenar el modelo de aprendizaje profundo, lo que acelera el proceso de entrenamiento y mejora la precisión.

Otra técnica utilizada es la transferencia de aprendizaje para mejorar el rendimiento de las redes neuronales. Esta técnica implica reutilizar una red previamente entrenada en una tarea relacionada para una nueva tarea. Al utilizar el conocimiento previo adquirido en la tarea anterior, el modelo puede aprender más rápido y mejorar su precisión.

Otros utilizan técnicas de filtrado guiado para mejorar la calidad de la solución. Esta técnica utiliza una imagen guía de alta calidad para filtrar una imagen de baja calidad y mejorar la resolución y la claridad de la imagen.

La optimización estructurada también se emplea en algunos artículos para mejorar la precisión de la segmentación de edificios. Esta técnica utiliza restricciones estructurales para mejorar la coherencia y la continuidad de los objetos segmentados, lo que puede mejorar la precisión y la calidad de la solución.

Otro enfoque utilizado es la combinación de varias de estas técnicas. Algunos autores utilizan algoritmos de clustering para el procesamiento de las imágenes

y encontrar patrones en estas, los algoritmos más usados son el K-Means y Mean Shift. Otras de las más combinadas son la reducción de dimensiones con PCA y redes neuronales convolucionales.

Ross Girshick en su artículo "Fast R-CNN"hace una propuesta de una nueva técnica [9]. Fast R-CNN utiliza una red neuronal convolucional para extraer características de la imagen y una capa de regresión para localizar objetos en la imagen. A diferencia de R-CNN", "Fast R-CNNütiliza una sola red convolucional para extraer características de la imagen y detectar objetos, lo que mejora la eficiencia computacional y la velocidad de procesamiento. Su eficiencia computacional y su alta precisión en la detección de objetos lo hacen una técnica atractiva, por lo que nuestro modelo principal se basa en esta.

En el artículo "Satellite Image Classification with Deep Learning"[4], se ve un enfoque donde primeramente se preprocesan las imágenes para posteriormente brindarselas a una CNN, alargando los bordes de las cajas donde se encuentran los objetos en proporción con el tamaño de la imagen, este paso es necesario porque provee de un contexto de píxeles alrededor del objeto que son de interés para la red neuronal. Además el modelo CNN que utilizan trae consigo 4 arquitecturas (DenseNet, ResNet, Inception, Xception). Dado que el enfoque de este artículo tiene una alta precisión en la problemática que se trata y está explicado con bastante claridad se decidió usarlo en uno de los modelos implementados.

Recientemente se está trabajando en un modelo llamado Segment Anything Model (SAM) [10], que es particularmente útil en el campo de la detección y la clasificación de instalaciones en imágenes satelitales. Dada su fiabilidad usamos este modelo para comparar sus resultados con los implementados por nosotros.

3. Dataset

3.1. Functional Map of the World Dataset

El dataset usado para el entrenamiento de los modelos fue el llamado Functional Map of the World Dataset. El conjunto de datos "Functional Map of the World"(FMOW) es un conjunto de datos público de imágenes satelitales que se utiliza para tareas de clasificación de objetos y detección de objetos. El conjunto de datos contiene alrededor de 1 millón de imágenes de alta resolución de todo el mundo, que se han etiquetado manualmente con información sobre las clases de objetos presentes en la imagen.

El FMOW se divide en dos partes principales: una parte de entrenamiento y una parte de prueba. La parte de entrenamiento consta de alrededor de 900,000 imágenes etiquetadas, mientras que la parte de prueba contiene alrededor de 100,000 imágenes no etiquetadas. Las imágenes en el conjunto de datos muestran una variedad de paisajes y entornos, incluidas áreas urbanas y rurales, y se capturaron en diferentes momentos del día y en diferentes condiciones climáticas.

Las etiquetas de clase en el FMOW se basan en una taxonomía de objetos llamada "Functional Map of the World"(FMoW). La taxonomía FMoW se centra en las funciones que cumplen los objetos en lugar de en su apariencia física,

lo que permite una clasificación más precisa y consistente de los objetos en diferentes contextos y entornos. Las clases de objetos incluyen cosas como edificios, vehículos, cuerpos de agua, cultivos y áreas verdes.

Actualmente se encuentra libre para su descarga en Amazon S3.

3.2. Cuba Stadia

Este es un dataset creado por el equipo para el entrenamiento y pruebas de los modelos implementdos. Consta de 38 imágenes de estadios de beisbol de nuestro territorio nacional con sus correspondientes etiquetas. Cada etiqueta contiene el Bounding Box del estadio así como las dimensiones de la imágen. Se construyó tomando screenshots de Google Maps Satelital, sin etiquetas, luego he recortado las imagenes en forma cuadrada con 720 pixeles de ancho. Las anotaciones fueron realizadas con una aplicación de nuestra autoría.

4. Modelos Utilizados

4.1. Faster R-CNN

Faster R-CNN es un algoritmo popular de detección de objetos que fue introducido por Shaoqing Ren, Kaiming He, Ross Girshick y Jian Sun en 2015. Es una extensión del modelo R-CNN original (Convolutional Neural Network basado en regiones), que fue introducido por Ross Girshick et al. en 2014.

La detección de objetos con Faster-RCNN se logra primero generando un conjunto de propuestas de región (es decir, ubicaciones de objetos candidatos) utilizando una Red de Proposición de Regiones (RPN), y luego clasificando estas propuestas utilizando una red de clasificación.

La RPN es una red neuronal completamente convolucional que toma una imagen como entrada y produce un conjunto de propuestas de objetos rectangulares, cada una con una puntuación de objetividad asociada. Estas propuestas se generan deslizando una pequeña red sobre el mapa de características convolucionales producido por una red de base pre-entrenada (típicamente una red VGG o ResNet). La RPN se entrena de extremo a extremo con la red de clasificación, utilizando una función de pérdida de múltiples tareas que combina una pérdida de clasificación binaria para la objetividad y una pérdida de regresión para las coordenadas del cuadro delimitador.

La red de clasificación toma cada propuesta generada por la RPN y realiza la clasificación de objetos y la regresión del cuadro delimitador. La red consta de una serie de capas completamente conectadas que toman las características de cada propuesta como entrada y producen una etiqueta de clase y coordenadas del cuadro delimitador.

La implementación de este modelo se realizó con la utilización de la biblioteca Pytorch de python y se realizó el entranmiento en Google Colab con la clase Stadium del dataset. Es decir el modelo se entrenó para detectar estadios en imágenes satelitales. De igual forma se puede entrenar para detectar otro tipo de edificación.

4.2. InceptionV3 + DNN

El modelo de machine learning que se presenta utiliza la arquitectura InceptionV3 de redes neuronales convolucionales (CNN) para la extracción de características, seguida de una red neuronal densa (DNN) para la clasificación.

El modelo InceptionV3 es una red neuronal convolucional profunda que ha demostrado ser altamente eficaz en la clasificación de imágenes. Fue desarrollado por Google y se encuentra disponible en la librería de aprendizaje profundo Keras, lo que lo hace accesible para su uso en proyectos de Machine Learning.

El modelo InceptionV3 se caracteriza por su arquitectura en forma de ïnception module", que se basa en la idea de combinar diferentes tamaños de filtros dentro de la misma capa convolucional. Esta técnica permite reducir la cantidad de parámetros que debe aprender el modelo, lo que a su vez reduce el riesgo de sobreajuste y mejora su capacidad de generalización.

Además, InceptionV3 utiliza técnicas como la regularización L2 y el dropout para prevenir el sobreajuste, y utiliza la función de activación ReLU para acelerar el entrenamiento de la red. El modelo también utiliza capas de agrupamiento máximo (max pooling) para reducir el tamaño de las características de la imagen y facilitar su procesamiento.

La red neuronal densa que se agrega a la arquitectura InceptionV3 se utiliza para la clasificación de las características extraídas por la red convolucional. La red consta de dos capas densas completamente conectadas con 128 y 2 neuronas respectivamente. La capa final de la red densa utiliza la función de activación softmax, que normaliza las salidas de la capa anterior para que representen probabilidades de clase para la clasificación multiclase. La función de activación ReLU se utiliza en la capa anterior para introducir no linealidad en la red y mejorar su capacidad de aprendizaje.

El modelo se entrenó para la clasificación de imágenes con aeropuertos y zoológicos. Los datos de validación y prueba son subconjuntos de los datos de entrenamiento.

4.3. DenseNet + ResNet + InceptionV3 + Xception

Este es un modelo de aprendizaje profundo que utiliza cuatro arquitecturas de redes neuronales pre-entrenadas (DenseNet121, ResNet152, InceptionV3 y Xception) para extraer características de una imagen de entrada de tamaño 224x224x3 y luego combinar estas características en una capa de fusión. La salida de la capa de fusión se alimenta a través de una capa densa completamente conectada con 1024 unidades y una función de activación ReLU, seguida de una capa de dropout para reducir el sobreajuste. Finalmente, hay una capa densa con una función de activación softmax que predice la probabilidad de pertenecer a cada una de las clases de destino.

DenseNet121: DenseNet es una arquitectura de red neuronal convolucional profunda que se caracteriza por tener conexiones densas entre las capas. DenseNet121 es una versión específica de esta arquitectura que tiene 121 capas y se ha

entrenado en un conjunto de datos masivo llamado ImageNet. Esta arquitectura es conocida por su eficiencia y precisión en la clasificación de imágenes.

ResNet152: ResNet es otra arquitectura de red neuronal convolucional profunda que se caracteriza por tener conexiones residuales entre las capas. Estas conexiones permiten que la información fluya directamente a través de la red, lo que facilita el entrenamiento de redes más profundas. ResNet152 es una versión específica de esta arquitectura que tiene 152 capas y también se ha entrenado en el conjunto de datos ImageNet. Esta arquitectura ha demostrado ser muy eficaz para tareas de clasificación de imágenes.

InceptionV3: Inception es una arquitectura de red neuronal convolucional que se caracteriza por tener múltiples caminos de procesamiento de información dentro de la red. InceptionV3 es una versión específica de esta arquitectura que se ha entrenado en el conjunto de datos ImageNet. Esta arquitectura es conocida por su eficiencia y precisión en la clasificación de imágenes.

Xception: Xception es una arquitectura de red neuronal convolucional profunda que se caracteriza por tener convoluciones separables en profundidad. Las convoluciones separables en profundidad son una variante de las convoluciones regulares que separan el procesamiento espacial y de características en dos etapas diferentes. Xception se ha entrenado en el conjunto de datos ImageNet y ha demostrado ser muy eficaz para tareas de clasificación de imágenes.

EL modelo fue entrenado para identificar las clases: airport terminal, burial site, park, stadium, zoo. Los datos de validación y prueba son subconjuntos de los datos de entrenamiento.

4.4. Segmentación con K-Means

Se experimentó con cuatro propuestas de algoritmos que utilizan K-Means para segmentar las imágenes satelitales:

K-Means sobre píxeles: Se aplica el algoritmo de K-Means sobre los valores RGB de los píxeles de la imágen satelital. Las edificaciones, en su mayoría tienen un mismo color, por tanto un agrupamiento de los píxeles por colores similares puede segmentar las edificaciones presentes en la imágen.

K-Means sobre RGB y coordenads XY: Mejorando la propuesta anterior, por cada píxel hay un vector de 5 componentes que contiene los valores RGB y las coordenadas XY de cada píxel. A estos vectores se les aplica el algoritmo de clustering. Incluir las coordenadas XY del píxel permite que la agrupación tenga en cuenta no solo el color sino también la cercanía entre los píxeles.

K-Means y Algoritmo de Canny: El algoritmo de Canny es un algoritmo de detección de bordes ampliamente utilizado en el procesamiento de imágenes. El resultado de aplicarlo es una imagen binaria que contiene los bordes detectados en la imagen original. Se aplica K-Means a la imágen satelital y se superponen las imagenes resultantes de aplicar Canny y K-Means para obtener la segmentacion.

La selección del parámetro K del algortimo K-Means se efectúa mediante el método del codo.

5. Experimentación

5.1. DenseNet + ResNet + InceptionV3 + Xception

ValCA_mean : media de Categorical_Accuracy en los datos de validacion
ValP_mean : media de Precision en los datos de validacion
ValCA_var : varianza de Categorical_Accuracy en los datos de validacion
ValP_var : varianza de Precision en los datos de validacion

Experimento	Arquitecturas	Épocas	Pesos	ValCA_mean	ValP_mean	ValCA_var	ValP_var
1	DenseNet,ResNet	4	None	0.25500	0.17135	0.0033	0.02216
2	DensNet,ResNet	4	ImageNet	0.55918	0.58653	0.00050	0.00090
3	DenseNet,ResNet,InceptionV3,Xception	4	ImageNet	0.52842	0.61173	0.00054	0.00908
4	DenseNet,ResNet,InceptionV3,Xception	4	None	0.48506	0.46642	0.00020	0.05466
5	DenseNet,ResNet,InceptionV3,Xception	12	ImageNet	0.64419	0.79838	0.0020	0.01904

Cuadro 1. Resultados de experimentos

La experimentación con esta red consistió en variar la arquitectura variando las capas internas y usando redes preentrenadas o usando pesos random. Los experimentos son descritos con más detalle en la 1

Se puede observar que el experimento numero 5 fue el de mejores resultados, se puede llegar a la conclusión que el uso de redes preentrenadas y las arquitecturas utilizadas en el modelo influyeron en los resultados de manera positiva. Al observar las gráficas de estos se presentó la incógnita de si aumentar el número de epocas podía influir en los resultados, y pues en el último experimentos se probó esta alternativa y demuestra resultados positivos. Las figuras 1 y 2 muestran las curvas entrenamiento y validacion del experimento 5.

5.2. InceptionV3 + DNN

Se realizaron experimentos sobre tres variaciones del mismo modelo. Los tres clasificadores utilizan la técnica de fine tuning sobre el modelo inceptionV3, que nos facilita obtener mejores resultados con menos datos. A continuación se describe la arquitectura de las 3 variaciones utilizadas: En la arquitectura inicial los datos de entrada se pasaban al modelo inceptionv3 cuya salida era enviada a una capa densa de 256 neuronas, esta enviaba a una capa densa de 128 neurona la cual finalmente enviaba a la capa que daba la clasificación final. La evaluación del modelo en los datos de prueba tuvo resultados prometedores como se puede apreciar en la tabla. Aun así, resultaba interesante intentar mantener los mismo resultados utilizando una arquitectura menos profunda y menos compleja.

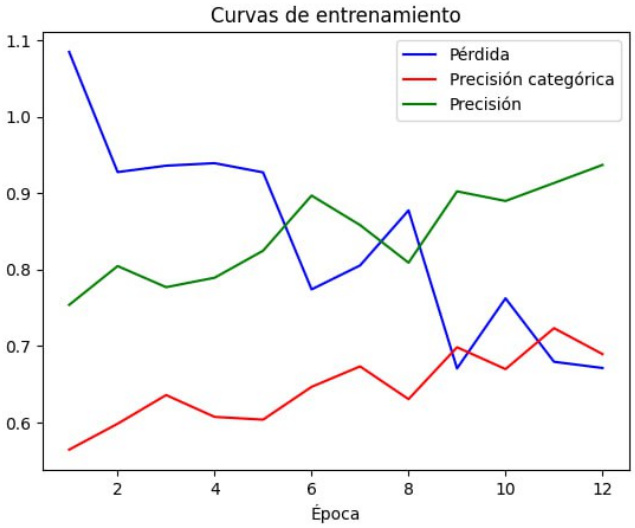

Figura 1. Curvas de Entrenamiento:Experimento 5-Densenet Resnet Inception Xception

Se ideó una nueva arquitectura donde las dos capas densas intermedias(la de 256 neuronas y 128 neuronas respectivamente) serían sustituidas por una sola capa de 256 neuronas. Esta variación proporciona los peores resultados de los experimentos realizados. Una última variación, utilizando 128 neuronas en lugar de 256 neuronas en la capa densa intermedia proporciona los mejores resultados hasta el momento, debido quizás al tamaño limitado del dataset de entrenamiento utilizado(2000 imágenes). Aunque se probaron muchas más variaciones a partir de la arquitectura inicial durante la experimentación, las tres mostradas anteriormente nos parecen las más prometedoras y las que describen mejor el proceso utilizado. A continuación, se muestran los datos obtenidos al evaluar los modelos con los datos de prueba:

mCA: media de la métrica categorical_accuracy
varCA: varianza de la métrica categorical_accuracy
mL: media de la función de pérdida
varL: varianza de la pérdida

Experimento	Arquitectura	Épocas	mCA	varCA	mL	varL
1	InceptionV3 densa(256) densa(128) salida(densa(2))	4	0.87541	0.000378	0.544901	0.031261
2	InceptionV3 densa(256) salida(densa(2))	4	0.64416	0.001130	25.20845	112.8327
3	InceptionV3 densa(128) salida(densa(2))	4	0.94125	0.001466	0.14125	0.002228

La Figura 3 muestra la curva de entrenamiento del experimento 3, el cual es el de mejor Presición Categórica.

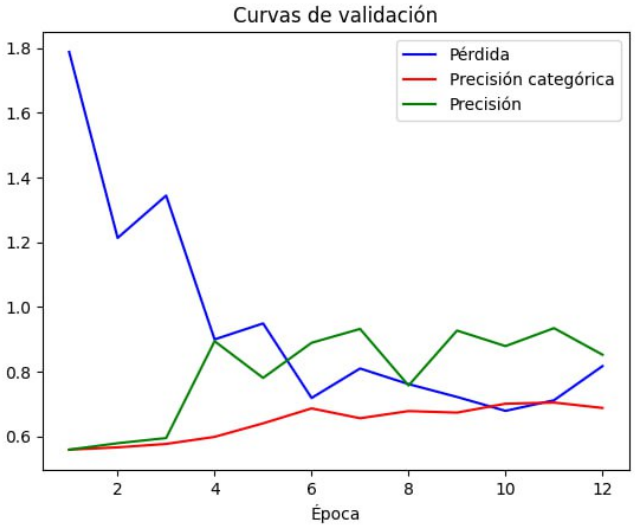

Figura 2. Curvas de Validacion:Experimento 5-Densenet Resnet Inception Xception

5.3. Faster-RCNN

Se realizaron siete experimentos sobre tres variaciones del modelo. Específicamente se variaron los backbones del modelo Faster-RCNN, se probaron distintas cantidades de datos de validación y prueba y se variaron la cantidad de épocas de entrenamiento. Para evaluar el rendimiento de los modelos de experimentación se calculó la métrica Mean Average Presition sobre el conjunto de predicciones del modelo sobre el dataset Cuba Stadia descrito en la sección Dataset. En el Cuadro 2 se muestran los resultados de los experimentos.

Datos Train: Cantidad de datos de entrenamiento Datos Val: Cantidad de datos de validacion mAP Test_Dataset: Mean Average Presition sobre Cuba Stadia

Experimento	Backbone	Épocas	Datos Train	Datos Val	mAP Test_Dataset
1	resnet50_fpn	5	3527	300	0.0264
2	mobilnet_v3_large_320_fpn	5	3527	300	0.0016
3	mobilnet_v3_large_320_fpn	10	3527	300	7.0721e-05
4	mobilnet_v3_large_320_fpn	20	6894	700	0.0125
5	mobilnet_v3_large_fpn	5	3527	300	0.0005
6	mobilnet_v3_large_fpn	15	6894	700	0.0051
7	resnet50_fpn	20	6894	700	0

Cuadro 2. Resultados de diferentes experimentos con diferentes backbones y configuraciones.

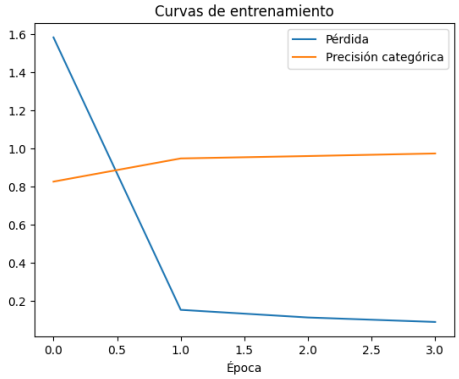

Figura 3. Predicción de Faster-RCNN

Como se puede apreciar el aumento de épocas en el experimento 3 con respecto al experimento 2 no influyó positivamente en el rendimiento del modelo, de esto se dedujo que, posiblemente faltaban datos. Esta hipótesis se probó la veracidad de esta hipótesis en el experimento 4, en el que se mejoró el rendimiento de los experimentos 3 y 2, esta variación del modelo resultó ser bastante rápida de entrenar. Dado su rapidez y la mejora de rendimiento vista se cree que con más datos de entrenamiento se podría obtener resultados prometedores con esta variante del modelo. La figura 4 muestra la curva de entrenamiento de para el experimento 4. Nótese como la tendencia de mAP es a crecer.

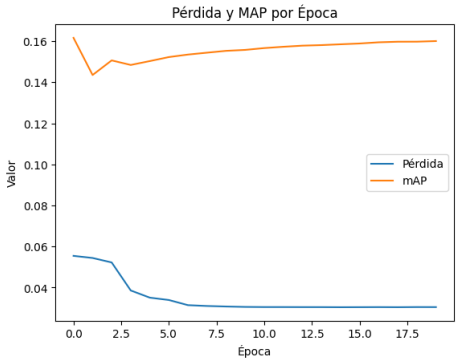

Figura 4. Curva de Entrenamiento Faster-RCNN Experimento 4

La viariante con mejor resultados fue la del experimento 1. Ante tal rendimiento con pocas épocas y pocos datos, se decidió realizar un último experimento, el 7, donde se aumentaba el número de épocas y la cantidad de datos.

Prueba con SAM: Se sometió a SAM a la misma prueba de rendimiento que se usó en los experimentos con Faster-RCNN, dando como resultado un valor de 0.1819 de Mean Average Presition sobre Cuba Stadia como dataset de prueba. Este resultado supera el de todas las variantes de Faster-RCNN con las que se experimentó, por lo que queda probada su eficacia en la tarea de Detección de Instalaciones en imágenes satelitales.

5.4. Clustering sobre los píxeles de la imagen

Por falta de tiempo solo se realizaron comparaciones visuales entre la segmentación producida por los algoritmos de clustering implementados y la segmentación producida por SAM. Para la gran mayoría de imágenes probadas, SAM produce una segmentación mucho mejor y mas clara de los estadios que las propuestas de clustering. Por lo que se concluye, empíricamente, que SAM es superior a las propuestas de segmentación por clustering implementadas. Las imágenes 9, 8, 7 son evidencia de la superioridad de SAM para la segmentación.

6. Conclusiones

Los resultados obtenidos son prometedores dado que solo se usan imágenes en RGB y todos los resultados importantes sobre el tema abordado utilizan imágenes multiespectrales. Los métricas obetenidas para la mejor variante del modelo Faster-RCNN no están muy distantes de las obtenidas por SAM en el mismo conjunto de prueba. Por tanto, se concluye que las imágenes satelitales en RGB pueden ser suficientes para lograr un entrenar un modelo de detección con resultados satisfactorios.

Por otro lado, recalcar el buen rendimiento de SAM. Este modelo no fue reentrenado con imágenes de el territorio nacional y aún así logra una segmentación correcta en la mayoría de las fotos de estadios cubanos. Aunque su rendimiento en imágenes con más de un estadio es mejorable.

Por último se concluye que las técnicas de aprendizaje no supervisado, por si solas, no son las más apropiadas para enfrentar los problemas de detección de objetos en imágenes satelitales. Los algoritmos de clustering implementados proveen resultados prometedores, aunque no tan descriptivos y prácticos como los algoritmos supervisados implementados o la propuesta de SAM.

7. Trabajos pendientes

Esta sección será eliminada en cuanto se terminen las labores pendientes:

- Faltan los resultados del experimento 7 de Faster-RCNN
- Luego de tener entrenado ese modelo, posiblemente el mejor, probarlo con fotos que contengan mas de un estadio
- Añadir al informe una prediccion de SAM en imágenes con más de un estadio.

8. Resultados

Ver figuras 5, 6, 7, 8, 9.

Figura 5. Imagen Original: Estadio Guillermón Moncada

Referencias

1. K. He et al., "Deep residual learning for image recognition," arXiv 1512.03385, Dec 2015.
2. G. Huang, "Dense connected convolutional neural networks," IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
3. Ishii, Tomohiro, Edgar Simo-Serra, Satoshi Iizuka, Yoshihiko Mochizuki, Akihiro Sugimoto, Hiroshi Ishikawa, y Ryosuke Nakamura. «Detection by classification of buildings in multispectral satellite imagery». En 2016 23rd International Conference on Pattern Recognition (ICPR), 3344-49, 2016. https://doi.org/10.1109/ICPR.2016.7900150.
4. Pritt, Mark, y Gary Chern. «Satellite Image Classification with Deep Learning». En 2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), 1-7, 2017. https://doi.org/10.1109/AIPR.2017.8457969.
5. Reda, Kinga, y Michal Kedzierski. «Detection, Classification and Boundary Regularization of Buildings in Satellite Imagery Using Faster Edge Region Convolutional Neural Networks». Remote Sensing 12, n.º 14 (enero de 2020): 2240. https://doi.org/10.3390/rs12142240.
6. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going Deeper with Convolutions."Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. doi: 10.1109/CVPR.2015.7298594

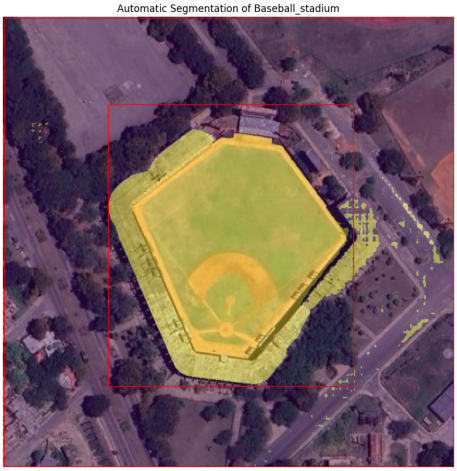

Figura 6. Predicción y segmentación de SAM para la imágen 5

7. François Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions."Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. doi: 10.1109/CVPR.2017.195
8. W. Shao, Ünsupervised feature learning for urban land use classification using deep convolutional neural networks", ISPRS Journal of Photogrammetry and Remote Sensing, 2016
9. Girshick Ross, "Fast R-CNN", En 2015 ICCV. https://arxiv.org/abs/1504.08083
10. https://encord.com/blog/segment-anything-model-explained/#h3
11. Wenzhi Zhao; Shihong Du, "Spectral-Spatial Feature Extraction for Hyperspectral Image Classification: A Dimension Reduction and Deep Learning Approach", En 2016 IEEE Transactions on Geoscience and Remote Sensing, https://ieeexplore.ieee.org/abstract/document/7450160
12. Adriana Romero; Carlo Gatta; Gustau Camps-Valls, Ünsupervised Deep Feature Extraction for Remote Sensing Image Classification", en 2016 IEEE Transactions on Geoscience and Remote Sensing, https://ieeexplore.ieee.org/document/7293195
13. Aaron Reite, Scott Kangas, Zackery Steck, Steven Goley, Jonathan Von Stroh, Steven Forsyth, Ünsupervised Feature Learning in Remote Sensing", en 2019 https://arxiv.org/abs/1908.02877
14. Tomohiro Ishii; Edgar Simo-Serra; Satoshi Iizuka; Yoshihiko Mochizuki; Akihiro Sugimoto; Hiroshi Ishikawa; Ryosuke Nakamura, "Detection by classification of buildings in multispectral satellite imagery", en 2016 23rd International Conference on Pattern Recognition (ICPR), https://ieeexplore.ieee.org/document/7900150
15. Dixit, M., Chaurasia, K., Mishra, V.K. (2021). Automatic Building Extraction from High-Resolution Satellite Images Using Deep Learning Techniques. In: Dave, M., Garg, R., Dua, M., Hussien, J. (eds) Proceedings of the International Conference on Paradigms of Computing, Communication and Data Sciences. Algorithms for Intelligent Systems. Springer, Singapore. $https://doi.org/10,1007/978-981-15-7533-4_61$
16. M M Zhu, Y L Xu, S P Ma, H Q Ma, .Airport Detection on Remote Sensing Images Using Fater Region-based Convolutional Neural Net-

Figura 7. Predicción de Faster-RCNN para la imágen 5

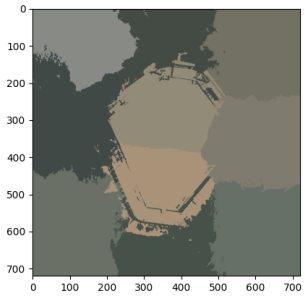

Figura 8. K-Means RGB + XY: segmentación para la imágen 5

work", Journal of Physics: Conference Series, vol.1060, pp.012037, 2018. https://iopscience.iop.org/article/10.1088/1742-6596/1060/1/012037/meta

17. Targ, Sasha, Diogo Almeida, y Kevin Lyman. «Resnet in Resnet: Generalizing Residual Architectures». arXiv, 25 de marzo de 2016. https://doi.org/10.48550/arXiv.1603.08029.
18. Xia, Xiaoling, Cui Xu, y Bing Nan. «Inception-v3 for flower classification». En 2017 2nd International Conference on Image, Vision and Computing (ICIVC), 783-87, 2017. https://doi.org/10.1109/ICIVC.2017.7984661.
19. Zhang, Ke, Yurong Guo, Xinsheng Wang, Jinsha Yuan, y Qiaolin Ding. «Multiple Feature Reweight DenseNet for Image Classification». IEEE Access 7 (2019): 9872-80. https://doi.org/10.1109/ACCESS.2018.2890127.

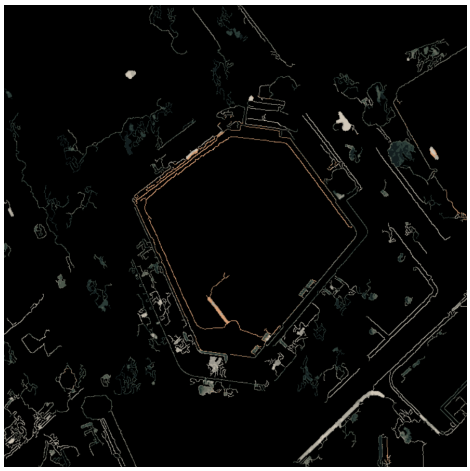

Figura 9. K-Means + Canny: segmentación para la imágen 5