**Reconocimiento de cráteres en fotos de la luna tomadas por Observatorio ITM mediante Redes Neuronales Convolucionales**

**CAMILO ANDRES HERNANDEZ CADAVID**

**LUIS EVELIO JIMENEZ DURANGO**

**YULIETH VERONICA RAMIREZ MENDES**

**Docente:**

**JOSE F. PAMPLONA**

**RUBEN D. FONNEGRA**

**CARLOS MADRIGAL**

**INSTITUTO TECNOLÓGICO METROPOLITANO**

**FACULTAD DE INGENIERÍAS**

**PROYECTO INTEGRADOR DE SOFTWARE LIBRE**

**MEDELLÍN**

RESUMEN

El conteo de cráteres en la Luna y otros cuerpos es crucial para restringir la historia dinámica del Sistema Solar. Esto se ha hecho tradicionalmente mediante la inspección visual de imágenes, lo que limita el alcance, la eficiencia y / o la precisión de la de la información. Para este proyecto se realiza la implementación de redes neuronales convolucionales para determinar la identificación de los cráteres, con base a información recolectada por personal humano a lo largo de la historia donde se ha obtenido un 92 % del conjunto de datos. Nuestros resultados sugieren que el aprendizaje profundo será una herramienta útil para extraer cráteres rápida y automáticamente en varios cuerpos del Sistema Solar iniciando con la luna y a futuro con otros cuerpos celeste.

Dado que no se obtuvo un conjunto de datos con imágenes de cráteres lunares para la iniciación del proyecto se realiza una recolección, clasificación y etiquetación manual de cráteres basados en imágenes recolectadas a lo largo de la historia y de fotos tomadas por el observatorio del ITM. Teniendo el principal recurso (Data set) el cual contaba con 2200 imágenes entre cráteres y no cráteres y más de 1000 imágenes con etiquetas se procede mediante al uso de técnicas de Deep learning con la creación e implementación de un modelo de clasificación y un modelo de detección que posterior a varios intentos de entrenamiento y técnicas de regularización se obtienen los resultados esperados, detectar cráteres en imágenes de la luna y clasificar cuando es un cráter y cuando no lo es.

RECONOCIMIENTOS

Por parte del ITM queremos agradecer al equipo de docencia que imparten el curso de deep learning que con su apoyo y seguimiento hicieron posible la ejecución completa del proyecto, así mismo cada una de las clases brindadas fueron vitales para el desarrollo de este. Queremos agradecer al personal del Observatorio ITM quien con su disposición nos proporcionó información y recursos que contribuyeron a cumplir nuestros objetivos.

Por otra parte, un trabajo de grado es fruto del reconocimiento y del vital apoyo

que nos ofrecen las personas que nos estiman, familiares y compañeros, sin el cual no tendríamos la fuerza y energía que nos anima a crecer como personas y como profesionales.

ACRÓNIMOS

CNN Convolucional Neuronal Networks

RNA Red neuronal artificial

ResNet Red Neuronal Residual

YOLO You Only Look Once

SSD  Single Shot Multibox Detector

TABLA DE CONTENIDO

|  |  |
| --- | --- |
| [1. INTRODUCCIÓN 6](#_Toc396987387)  [2. MARCO TEÓRICO 7](#_Toc396987388)  [3. METODOLOGÍA 8](#_Toc396987389)  [4. RESULTADOS Y DISCUSIÓN 9](#_Toc396987390)  [5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO 10](#_Toc396987391)  [REFERENCIAS 11](#_Toc396987392)  [APÉNDICE 13](#_Toc396987393) |  |
|  |  |

1. INTRODUCCIÓN

Los cráteres formados por pequeños impactados constituyen una propiedad superficial importante para muchos cuerpos del Sistema Solar en este caso para la luna tradicionalmente, la detección de cráteres se ha realizado manualmente mediante la inspección visual de imágenes. Sin embargo, este enfoque no es práctico para la gran cantidad de cráteres de tamaño kilométrico y sub kilométrico en la Luna (y otros

Cuerpos del Sistema Solar), lo que da como resultado bases de datos generadas por humanos que son espacialmente completas pero restringidas a los cráteres más grandes, o tamaño.

En este trabajo entrenamos una arquitectura de aprendizaje profundo conocida como ... red neuronal convolucional (CNN) para realizar la identificación de cráteres en la Luna, las CNN han demostrado un desempeño impresionante en una variedad de problemas de visión por computadora y otros conjuntos de datos donde las características están correlacionadas lo que demuestra su versatilidad. Las CNN diseñan sus propias funciones de representación, aliviando la necesidad de un ser humano para desarrollar sofisticados algoritmos de preprocesamiento y personalizadas características de entrada. Finalmente, las CNN han podido clasificar objetos con éxito que aparecen a múltiples escalas en una sola imagen.

Nuestro principal objeto es identificar cráteres que no son visibles al ojo humano en imágenes captadas de la luna del observatorio del ITM y de imágenes de cráteres.

Se busca implementar modelos de detección de objetos y ajustar adecuadamente los parámetros para obtener resultados óptimos en la identificación y clasificación de un cráter, dando como resultado una imagen con los cráteres más relevantes resaltados en un cuadro.

1. MARCO TEÓRICO

**Cráteres lunares**

Los cráteres lunares están compuestos en su mayoría por cráteres de impacto. Los cráteres en su mayoría cuentan con muchas de las siguientes características listadas a continuación:

* Un borde elevado, compuesto por materiales esparcidos durante el momento del impacto.
* Pared del cráter, siendo la parte más baja del borde del cráter.
* Suelo del cráter, es un área plana más o menos lisa que mientras más envejece acumula nuevos cráteres.
* Pico central, solo se denota en cráteres con un diámetro superior a 26 kilómetros, es un efecto creado por la energía cinética del objeto impactante. Es también conocido como efecto salpicadura.

**Deep learning**

Aprendizaje profundo, el aprendizaje automático y la inteligencia artificial son conceptos muy centrados en la ciencia de datos. El aprendizaje profundo ha tenido éxito en el campo de la visión por computadora, el procesamiento del habla y audio y el procesamiento del lenguaje natural. Tiene una gran capacidad de aprendizaje que puede mejorar la utilización de conjuntos de datos para la extracción de características en comparación con el algoritmo tradicional de aprendizaje automático lo que ha contribuido con éxito en la tecnología actual. La representación del conocimiento se hace desde el estrato más básico de la inteligencia: el cerebro, especialmente en las neuronas y las múltiples conexiones entre ellas es por eso que para un aprendizaje profundo las redes neuronales artificiales son una inspiración de un elemento fundamental del cerebro que es la neurona.

Las redes neuronales artificiales (RNA) son modelos que intentan reproducir el comportamiento del cerebro. Una RNA está constituida por nodos, o unidades, que están unidas mediante conexiones. Algunas de las unidades están conectadas al ambiente externo y se designan como unidades de entrada o de salida según reciban estímulos externos o den la respuesta del sistema, respectivamente. Hay otras unidades (neuronas) que simplemente están conectadas a otras neuronas del sistema y que al no tener relación directa con la información de entrada ni con la de salida del sistema, se denominan unidades ocultas. Se conoce como capa o nivel a un conjunto de neuronas cuyas entradas provienen de la misma fuente (que puede ser otra capa de neuronas) y cuyas salidas se dirigen al mismo destino (que puede ser otra capa de neuronas).

**Aprendizaje supervisado**

Es un aprendizaje bajo etiquetas, recopilamos un gran conjunto de datos similar a un rompecabezas en el que un objeto grande se divide en un número de imágenes pequeñas y se etiqueta cada una con su categoría. Una vez que se produce la salida, se expone a la máquina para obtener la salida deseada en forma de puntuación vectorial. El error entre la salida y el patrón deseado se calcula usando su función objetivo y si el resultado obtenido no es exacto la máquina modifica el parámetro interno ajustable para reducir el error, estos parámetros se denominan pesos. El algoritmo de aprendizaje supervisado analiza el conjunto de datos de entrenamiento y usa el resultado previsto para mapear las nuevas instancias.

**Redes neuronales convolucionales (CNN):** En detección de cráteres y clasificación de estos se hacia necesario utilizar CNN siendo tipo de Red Neuronal Artificial con aprendizaje supervisado que procesa sus capas imitando al córtex visual del ojo humano para identificar distintas características en las entradas , la CNN contiene varias capas ocultas especializadas y con una jerarquía: esto quiere decir que las primeras capas pueden detectar líneas, curvas y se van especializando hasta llegar a capas más profundas que reconocen formas complejas en imágenes de cráteres que no tienen figuras ni colores tan marcados. Las CNN constan de diversas capas convolucionales y de submuestreo alternadas, y al final tiene una serie de capas full-connected como una red perceptrón multicapa, por estas características se concluye que es la forma mas acetada de abordar el inconveniente presentado de detección de cráteres en la luna basado en imágenes.

**Red Neuronal Residual (ResNet)**

Una red neuronal residual (ResNet) es una red neuronal artificial (ANN) de un tipo que se basa en construcciones conocidas de las células piramidales de la corteza cerebral. Las redes neuronales residuales hacen esto mediante skip connections o atajos para saltar sobre algunas capas. Los modelos típicos de ResNet se implementan con saltos de doble o triple capa que contienen no lineales (ReLU) y normalización del batch. Se puede utilizar una matriz de peso adicional para aprender los pesos de salto; estos modelos se conocen como HighwayNets. Los modelos con varios saltos en paralelo se denominan DenseNets. En el contexto de redes neuronales residuales, una red no residual puede describirse como una red simple.

**RetinaNet**

Se ha formado mediante la realización de dos mejoras sobre los modelos existentes de detección de objetos de una sola etapa, esta arquitectura consta de cuatro componentes principales en su arquitectura:

1. Vía ascendente: Backbone (por ejemplo, ResNet) que calcula los mapas de características a diferentes escalas, independientemente del tamaño de la imagen de entrada o del Backbone.
2. Vía de arriba hacia abajo y conexiones laterales: la vía de arriba hacia abajo muestra los mapas de características espacialmente más gruesos de los niveles de pirámide más altos, y las conexiones laterales fusionan las capas de arriba hacia abajo y las capas de abajo hacia arriba con el mismo tamaño espacial.
3. Subred de clasificación: predice la probabilidad de que un objeto esté presente en cada ubicación espacial para cada caja de anclaje y clase de objeto.
4. Subred de regresión: regresa el desplazamiento de los cuadros delimitadores de los cuadros de anclaje para cada objeto de verdad del terreno.

**DEM (digital elevation model)**

Es una matriz de celdas que contienen un valor numérico, que representa una unidad gráfica de un territorio analizado

1. METODOLOGÍA

Estudio del problema:

Inicialmente se habló con personal del observatorio del ITM para conocer la información (en este caso imágenes de la luna) para realizar un alcance y aproximación del desarrollo de Deep learning (inteligencia artificial) que se quería realizar. las imágenes enviadas por el observatorio fueron las siguientes:

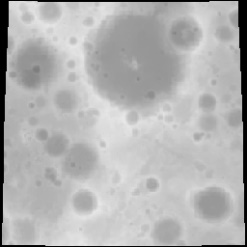
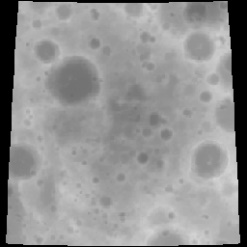
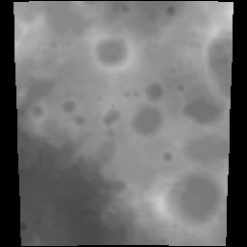
   

Las imágenes proporcionadas fueron de gran ayuda para empezar a planficiar lo que se queria desarrollar, por lo cual mirando la estructura de la imagen decidimos poder detectar cráteres de la imagen de la luna sin importar en que fase estuviera o con una acercamiento de la luna.

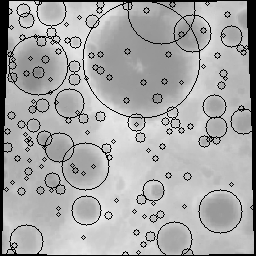
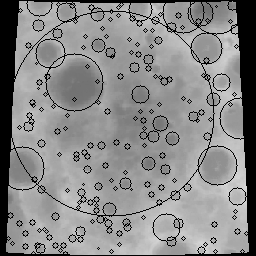
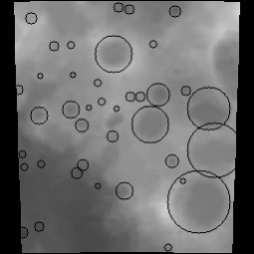
Búsqueda de información:

Nos valimos de buscar información en google ( [www.google.com](http://www.google.com)) para hacer un mejor estudio del problema planteado, inicialmente encontramos un proyecto similar llamado Deep Moon ( [www.github.com/silburt/DeepMoon](http://www.github.com/silburt/DeepMoon) ) en el cual recopilaban imágenes de la siguiente manera:

Imágenes de entrada y entrenamiento para su red :

Imágenes de resultado:

Es un excelente trabajo hecho por los desarrolladores, pero nosotros quisimos ir más allá de detección de crateres mediante DEM, pensamos en que cualquier persona con una imagen de la luna pudiera detectar en alguna medida los crateres de la luna.

Recopilación de bases de datos de imagenes propia:

Una vez analizamos la información encontrada, hicimos uso del software BITZI desarrollado por un compañero del ITM para la tarea de buscar imágenes de cráteres individuales y en grupo, aproximadamente dos mil imágenes, una vez con la base de datos procedimos a buscar información sobre los modelos para hacer detección de objetos.

Buscar información de modelos de deep learning para detección de objetos:

Hicimos la busqueda de modelos enseñados en el diplomado para detección de objetos como lo son:

* Fast-RCNN
* R-CNN
* Faster-RCNN
* YOLO
* SSD
* ExtremeNet

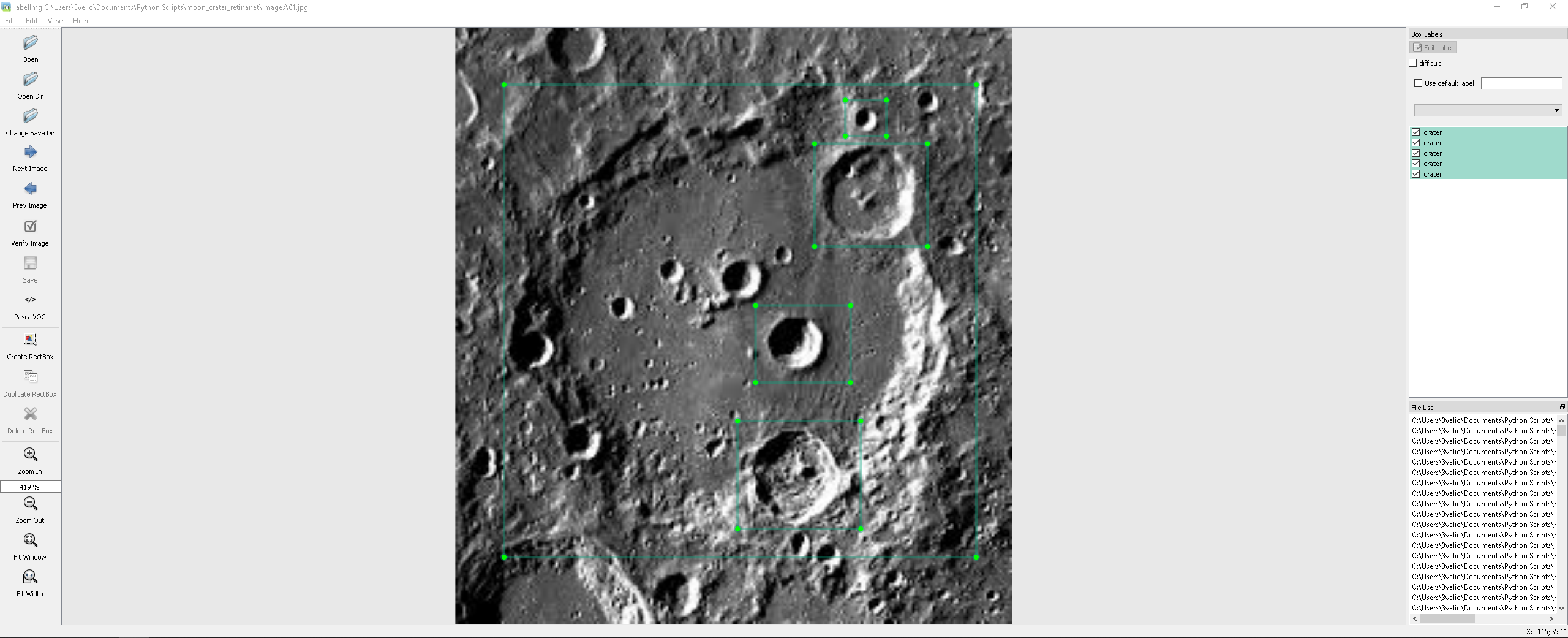
Pero en su busqueda hallamos un modelo llamado RetinaNet el cual está basado en Resnet50 para hacer la detección de objetos en pequeña y gran escala en una imagen.

Preprocesamiento de imágenes:

Debido a la arquitectura de Retinanet se deben pasar los datos de entrenamiento de diferentes maneras como lo son:

* Pascal VOC ( [www.host.robots.ox.ac.uk/pascal/VOC/](http://www.host.robots.ox.ac.uk/pascal/VOC/) )
* MS COCO ( www.cocodataset.org/#home )
* OID ( www.storage.googleapis.com/openimages/web/index.html )
* KITTI ( http://www.cvlibs.net/datasets/kitti/eval\_object.php )

Por facilidad de manejo y comprensión del problema hemos decidido usar la metodología de MS COCO, para esta metodología se deben etiquetar manualmente en las imágenes las zonas de interes en nuestro caso los cráteres, para ello usamos el sofware ([www.github.com/tzutalin/labelImg](http://www.github.com/tzutalin/labelImg)) y lo haciamos de la siguiente manera:



Las zonas de interes son los recuadros que están en los cuadros verdes. Una vez etiquetadas las imágenes se guardan en un formato XML, pero para el modelo de retinanet que lee estos datos de entrenamiento como archivos .csv, se hizo un algoritmo que lo llamamos xml2csv.ipynb en python que leyera los archivos xml en una carpeta y los almacene en un solo archivo csv y otro csv con cada una de sus respectivas clases (en este caso una, “cráteres”). Una vez hecho este proceso separamos las imágenes recolectadas en dos carpetas, una de entrenamiento y otra de pruebas. A ambas se les apicó el proceso de etiquetado y procesamiento para sacar tres archivos:

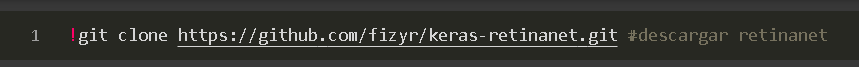
* Annotations.csv (datos de entrenamiento)
* Anotations\_test.csv (datos de prueba)
* Classes.csv (etiquetas de clases)

Los cuales necesita el RetinaNet para hacer todo su proceso de entrenamiento y prueba y poder sacar el modelo para probar su funcionalidad.

Entrenamiento del modelo RetinaNet con las imágenes de cráteres:

Una vez generados los archivos csv necesitados para el modelo de RetinaNet se aplica el siguiente proceso:

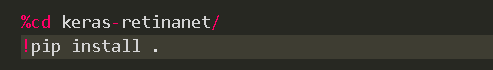
Se descarga el repositorio de keras retinanet para ser usado en una carpeta de nuetro google colab



Se actualizan las librerias para usar el keras



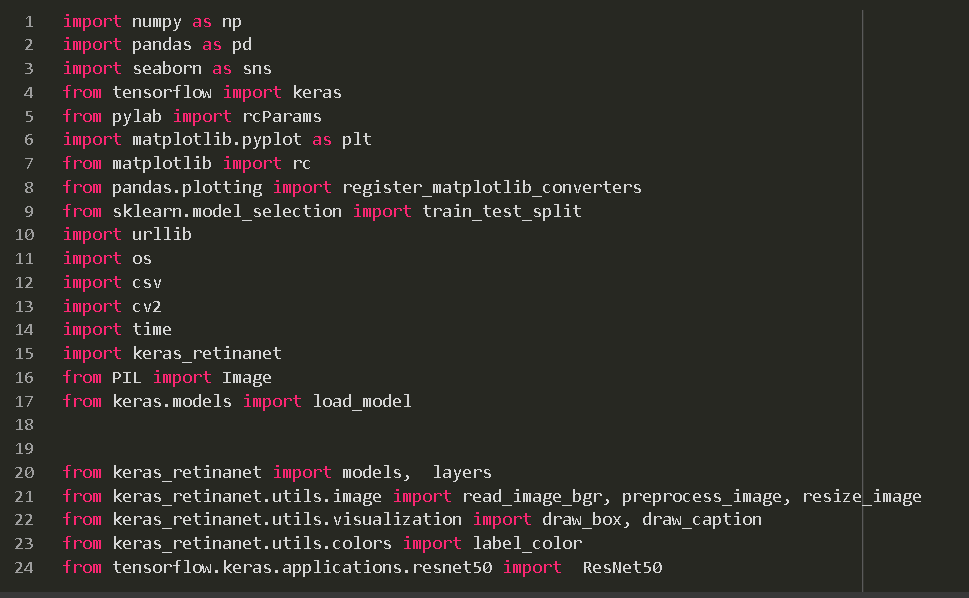
Se ingresa a la carpeta de keras-retinanet e instalamos las librerias necesarias



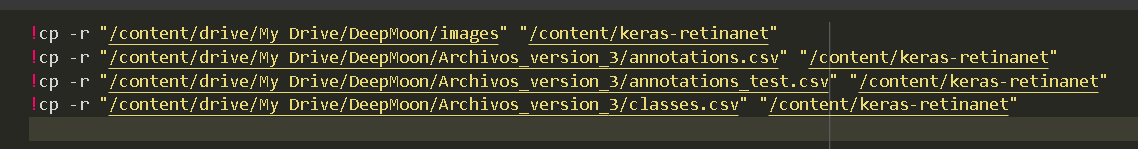
Construimos el modelo de retinanet para ser usado



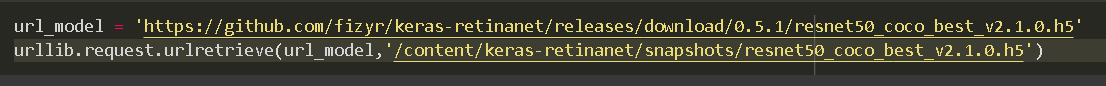
Instalamos las librerias necesarias para hacer el entrenamiento y validación de datos



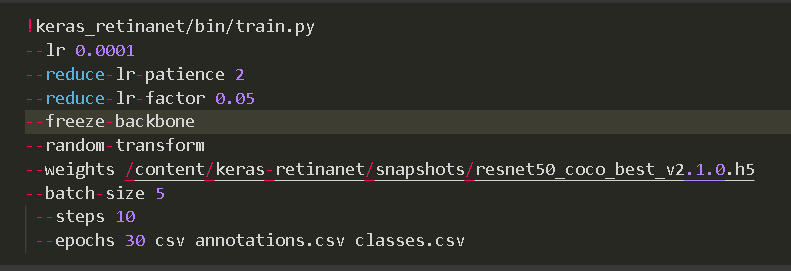
Copiamos los archivos cargados generados hacia la carpeta creada or retinanet para hacer el proceso



Se descarga el modelo ResNet coco50, con el cual se hace el transfer learning para hacer el entrenamiento con las imágenes propias en el RetinaNet y se guarda en la carpeta del RetinaNet



se procede hacer el entrenamiento con los siguientes parámetros



Lr = Tasa de aprendizaje 🡪 0.0001

Reduce-lr-patiente = Reducción de tolerancia tasa de aprendizaje 🡪 2

Reduce-lr-factor = Reducción de factor de tasa de aprendizaje 🡪 0.05

Freeze-backbone = congelar base de modelo entrenamiento (ResNet50)

Random-transform= transformar imagnees para data augmentation

Weights = pesos a usar del modelo (ResNet50)

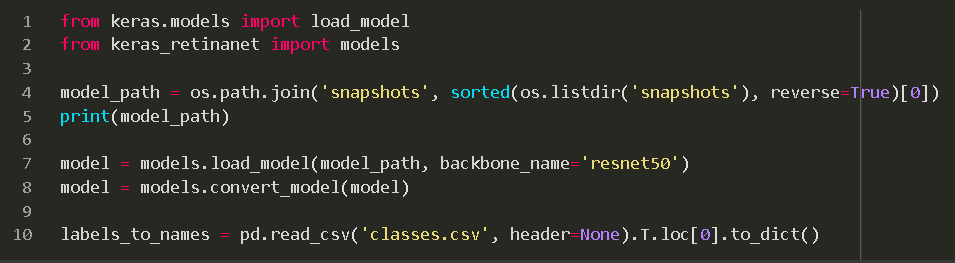
Batch-size = tamaño del lote de imágenes para usar en el entrenamiento

Steps = pasos a usar en el entrenamiento

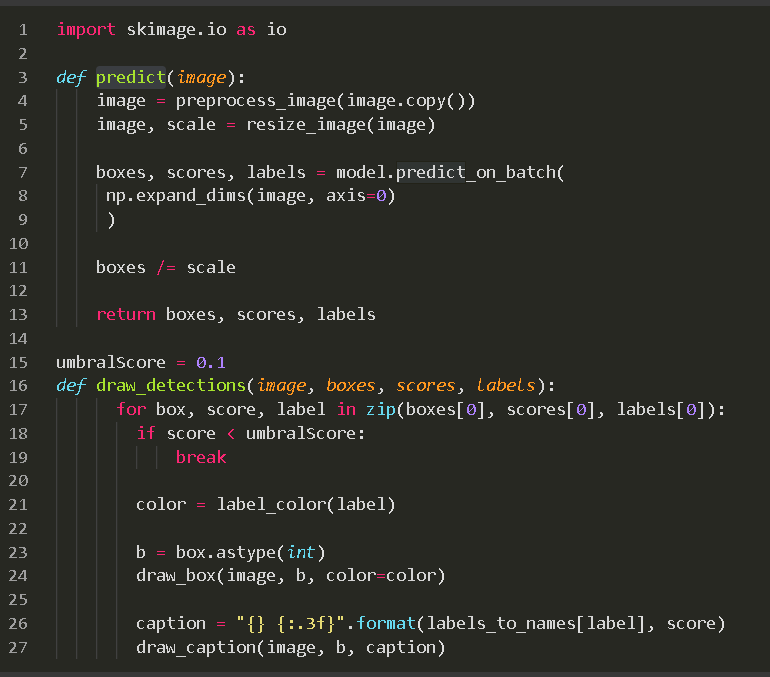
Epoch = épocas en las que se realiza el proceso

En este proceso se van calculando los pesos del nuevo entrenamiento y se van guardando backups en formato .h5 de cada época entrenada y se van guardando en la subcarpeta snapshots que está en keras-retinanet.

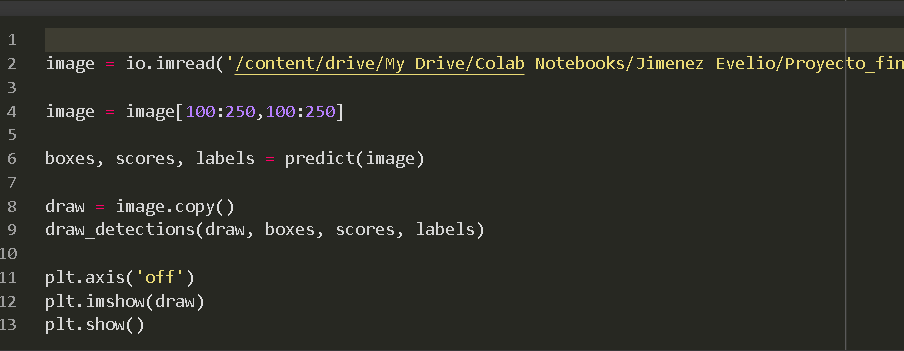
en este código cogemos el ultimo h5 para usarlo en el modelo para predecir un resultado



Se crean las funciones para predecir y pintar el recuadro en la imagen



Se lee una imagen para predecir y se pinta el lugar donde se encuentren las diferentes predicciones



En el proceso de entrenamiento

En esta sección se condensa la información relacionada con el cómo fue realizado su trabajo, los parámetros que utilizó, sobre qué datos fueron apoyados, que fue lo evaluado. Se debe citar las fuentes de información utilizadas, las personas o instituciones que proporcionaron los datos y en general los recursos humanos, tecnológicos o institucionales sobre los cuales desarrolló la tesis. Deberá mostrar en forma organizada, precisa y lógica como fueron alcanzados cada uno de los objetivos específicos planteados.

1. RESULTADOS Y DISCUSIÓN

Iniciando con un aproximado de 500 imágenes no obtuvimos éxito alguno en la detección de cráteres comprobando que la base de datos es crucial para realizar un buen entrenamiento. Pero posterior a la etiquetación e inclusión de más imágenes, con el entrenamiento de varios modelos se logra el objetivo de alcanzar un 80 % en identificación de cráteres en la luna a continuación los resultados esperados.

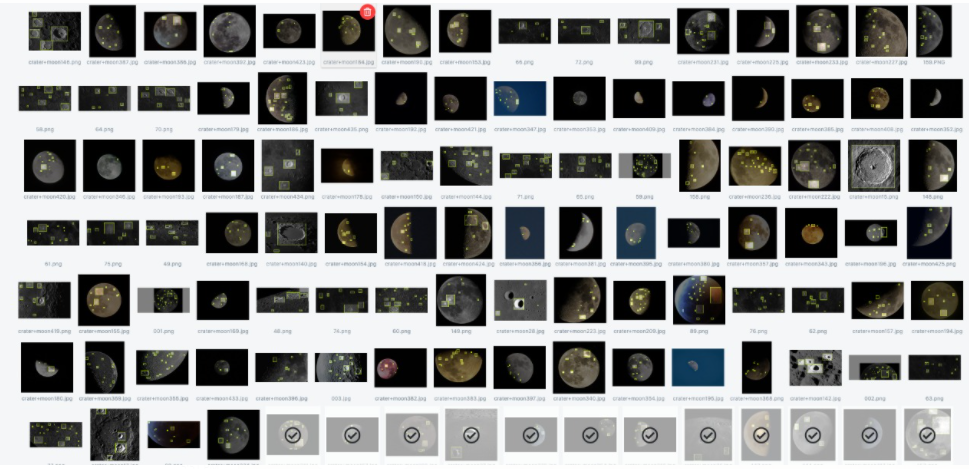


Ilustración Base de datos

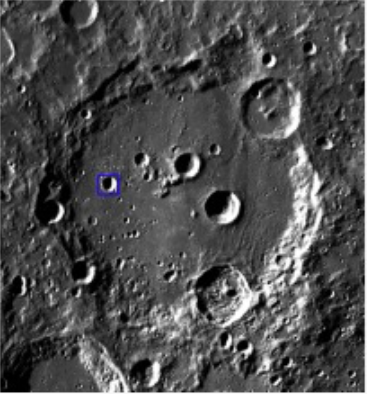
 

Ilustración Deteccion defectuosa de crater sobre un 1 % de probabilidad

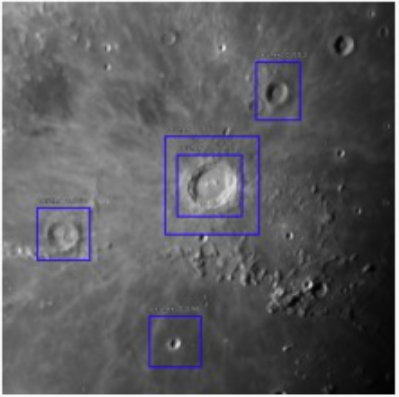
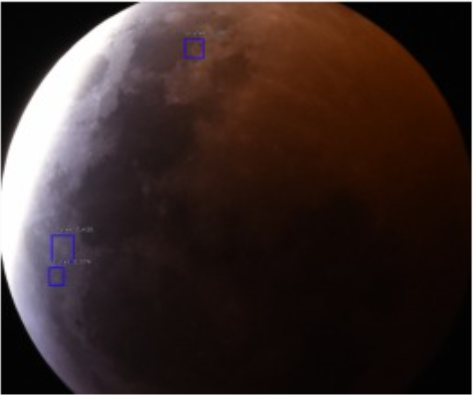
 

Ilustración Imagen observatorio ITM un 20% de probabilidad en los crateres identificados

creamos un data set más grande que aportara más información al modelo

Y logrando una detección de cráteres en un 40 % en fotos tomadas por el observatorio del ITM y en imágenes concretas con cráteres sin embargo esto no era suficiente para cubrir el objetivo por lo cual se registran más diversidad de imágenes y se realizan cambios en el modelo.

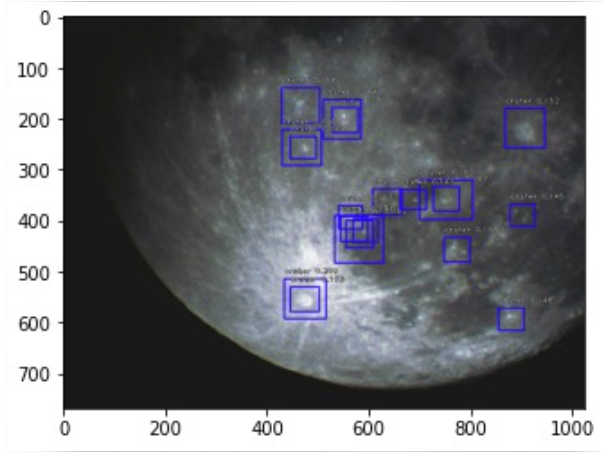
 

Ilustración 5 Imagen tomada con mas de un 50 % de identificacion de crater

Finalmente se obtiene un resultado con porcentajes acertivos de un 80 %

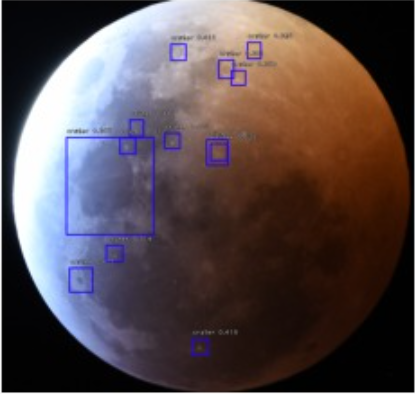
 

Ilustración 6 Imagen timada desde el observatorio ITM

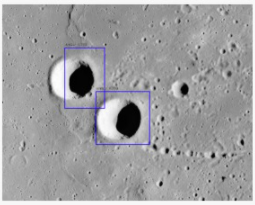
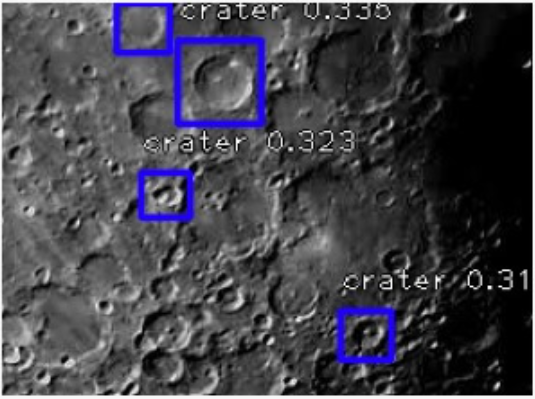
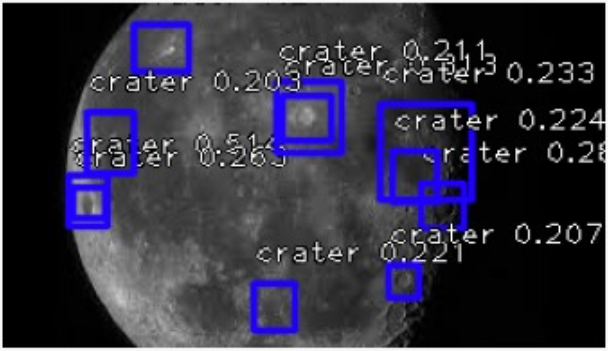
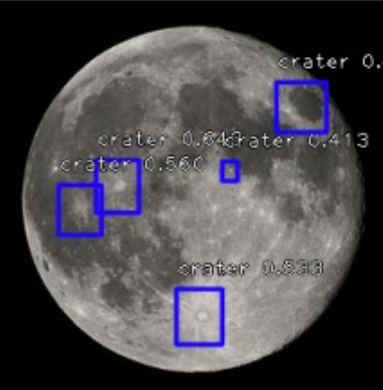


Ilustración 7 75 % en identificación de crater

1. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

En el análisis realizado en cada uno de los modelos montados se concluye que para un buen entrenamiento se requiere un mínimo de 1000 y que entre mas incremente el dataset se pueden lograr mejores resultados, en el caso nuestro al momento de realizar mas etiquetados de forma distribuida en los diferentes cráteres y en varios tonos los resultados fueron mejores donde se pasa de no detectar ningún cráter en la luna a detectar varios con un 70% en imágenes que ante el ojo humano estos no eran fáciles de detectar logrando nuestro objetivo.

Si bien nos fue proporcionada diferente información de modelos como lo son Fast-RCNN, R-CNN, Faster-RCNN, YOLO, SSD, ExtremeNet solo con RetinaNet basado en Resnet50 se logro tener una deteccion de crateres efectiva y esto nos da mejores resultados complementando con un buen etiquetado siendo muy precisos en la ubicación del crater dentro de un la imagen y teniendo encuenta que las etiquetas no debian ser precisas en talmaño si no por el contrario tener buen fondo para el modelo poder percibir una diferencia de color, bordes entre otros.

Durante el primer entrenamiento de modelo, se identificó en la red un sobre entrenamiento puesto que en la mayoría de las imágenes se etiquetaron en su mayoría cráteres con las mismas características, además de tener pocas imágenes, se recomienda gran cantidad de imágenes, y evaluar un Tasa de aprendizaje apropiado. A pesar que no es recomendable crear una base de datos desde cero por que requiere gran esfuerzo en su etiquetado, en su clasificación podríamos decir que, si es posible siendo muy detallados en lo que se busca, ser precisos e iniciar desde lo más mínimo para llegar al objetivo más grande.

Con la aplicación de RetinaNet para la identificación de cráteres lunares, con la base de datos de cráteres ya identificados da como resultado un modelo que nos permitirá asemejar y dar oportunidades de detectar cráteres en diferentes cuerpos celestes y tener un mejor entendimiento de cómo se han generado los impactos y se generaran nuevos cráteres en cualquier elemento solido de nuestro sistema solar.

REFERENCIAS

<http://lroc.sese.asu.edu/posts>

[Arshiya Begum](https://ieeexplore-ieee-org.itm.elogim.com:2443/author/37085572328).  (Abril 6 de 2019). Implementación del algoritmo de aprendizaje profundo con Perceptron usando la biblioteca TenzorFlow, Chennai, India, India. Recuperado de

<https://ieeexplore-ieee-org.itm.elogim.com:2443/document/8697910>

[Srihari Humbarwadi](https://twitter.com/srihari_rh) (Mayo 16 2020 ) Object Detection with RetinaNet . Recuperado de <https://keras.io/examples/vision/retinanet/>

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Sawabe, Y., Matsunaga, T., & Rokugawa, S. (2006). Automated detection and classification of lunar craters using multiple approaches. Advances in Space Research, 37(1), 21-27.

David Alberto Saji Santander. (2019) reconocimiento de montos manuscritos en cheques a través de modelos de detección de objetos basados en redes convolucionales, Santiago de Chile, recuperado de

http://repositorio.uchile.cl/bitstream/handle/2250/174542/cf-saji\_ds.pdf?sequence=1

APÉNDICE

Apéndice A,

Código en GitHub <https://github.com/3venjd/Moon-crater-detection-retinanet>