

Universidad del Valle de Guatemala
Data Science 1 - Sección 10
Christopher Kevin Sandoval García 13660
María Fernanda Estrada Cornejo 14198
Rodrigo Samayoa Morales 17332
David Uriel Soto Alvarez 17551
Ana Villela 18903
Guatemala 12 de noviembre de 2020



Presentación de resultados del proyecto 2

Investigación de algoritmos

- Procesamiento de datos médicos mediante el análisis de de imágenes y la geometría fractal

Los avances topológicos han permitido que los objetos y sus propiedades sean analizados de maneras más complejas. Un ejemplo, es el cambio de perspectiva de análisis de superficies, pasando de Euclidiano a topológico. El análisis topológico permite descomponer el espacio a analizar en partes más pequeñas, para poder representarlo e incluso medirlo con más precisión.

De igual forma, se introdujo el análisis fractal de cualquier sistema complejo o irregular, permitiendo encontrar patrones y similitudes en donde aparentemente solo hay caos. Con estos nuevos conceptos introducidos en el área de la medicina, comenzaron a surgir investigaciones que relacionaban comportamientos bioquímicos luego de traumas, con multifractales. Fue así como el instituto de oncología de Yugoslavia introdujo el análisis multifractal (MFA) para clasificación de imágenes de 256x256 píxeles en escala de grises, para el análisis de traumas epidermicos por quemaduras.

Estas interpretaciones fueron útiles para el Instituto de Medicina Patológica y Forense de la Academia Médica Militar de Belgrade, por lo que profundizaron el estudio y desarrollaron métodos y programas como FALFA Y HISTMF que seguirían mejorando y evolucionando hasta el día de hoy. (Reljin & Reljin, 2012)

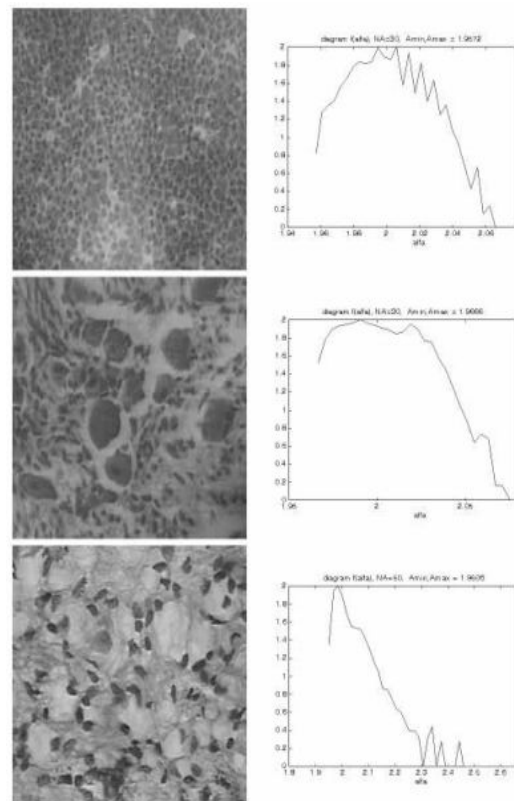


Imagen 1. Análisis de imagen mediante MFA por el Instituto de Oncología de Yugoslavia

- Análisis de imágenes utilizando modelos de redes neuronales convolucionales

Las redes neuronales convolucionales (CNN) es un algoritmo de Deep Learning utilizado principalmente para clasificar imágenes. Este algoritmo recibe como input una imagen, detecta las características principales de la misma y devuelve la categoría de la imagen. La forma en que detecta las características principales es que se le asigna un peso (importancia) a ciertos elementos de la imagen para poder diferenciar unos de otros. Algo importante a recalcar es que la salida se expresa en probabilidades, por lo que se debe tomar el que de mayor probabilidad.

El kernel en las CNN se considera como el filtro que se aplica a una imagen para extraer ciertas características o patrones importantes. Sirve para detectar bordes, enfoque, desenfocado, etc. Esto se logra al realizar una convolución entre el kernel y la imagen. La convolución en las redes neuronales consiste en tomar un grupo de píxeles de la imagen de entrada e ir realizando un producto escalar con un kernel. El kernel recorrerá todas las neuronas de entrada y obtendremos una nueva matriz, la cual se convierte en una de las hidden layers.

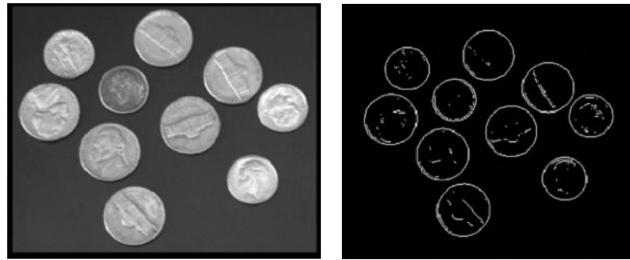


Imagen 2. Ejemplo de convolución entre el kernel y la imagen

Luego, en la capa de pooling, se combinan todas las salidas de los grupos de neuronas en una sola neurona para la siguiente capa. Esta capa es muy importante por el hecho que las imágenes no siempre se encuentran en la misma posición; pueden estar rotadas, posicionadas en la esquina, etc. Pooling reduce el tamaño de la representación de entrada y permite detectar objetos en una imagen sin importar dónde se encuentren. También ayuda a controlar el overfitting. Ya con estas capas creadas, se procede a la parte de flattening que consiste en aplanar estas matrices en un solo vector y ya el perceptrón hace la parte final de clasificación. (Bonner, 2019)

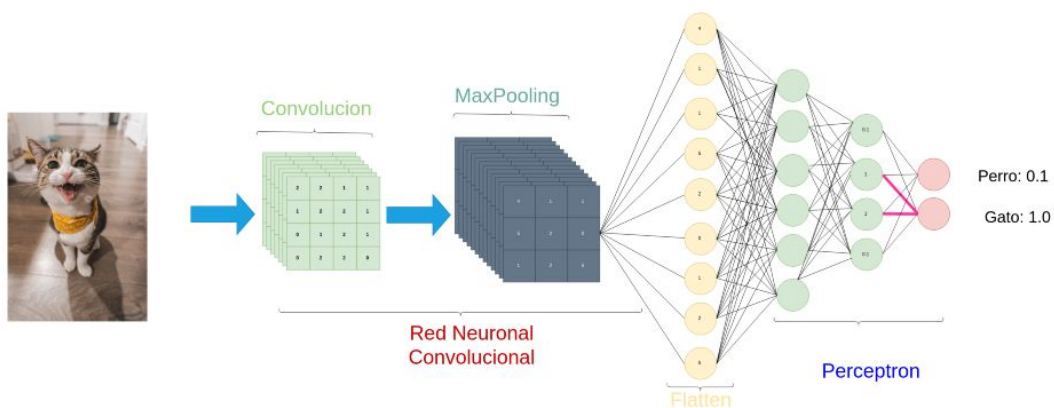


Imagen 3. Pasos de aplicación de una CNN

- Refinamiento de imágenes con el algoritmo Phase Stretch Transform

Existe una extensa cantidad de algoritmos que permiten el procesamiento de imágenes, los cuales permiten que se realicen cambios en ellas con el fin de poder mejorar su calidad, o en otros casos, permitir que la computadora o máquina pueda tener mayor facilidad al momento de reconocer o resaltar detalles. Estos algoritmos realizan acciones sobre las imágenes como por ejemplo cambiar la tonalidad, el contraste, mejorar definición, cambios de colores, rotación de contenido, reconstruir datos, entre otros. Estos algoritmos no proveen un tipo de modelo como tal para clasificación o predicción, pero son útiles al momento de apoyar a un algoritmo que sí haga este tipo de tareas facilitando el procesamiento o mejorando la calidad del modelo.

En este caso, el algoritmo Phase Stretch Transform es un algoritmo de procesamiento y reconocimiento de imágenes que es más o menos reciente, con 3 años de haber sido desarrollado con ingenieros de UCLA con acceso a código abierto. Este tiene como principal característica que mejora las imágenes, reconoce texturas y puede distinguir los bordes y otros detalles que aparecen en las imágenes. Capaz de mostrar detalles no visibles a simple vista, como estructuras de diversos materiales o las estrellas en una imagen del espacio. Algunos de los ejemplos que se han mostrado sobre cómo quedan las imágenes al usar el algoritmo son las siguientes:

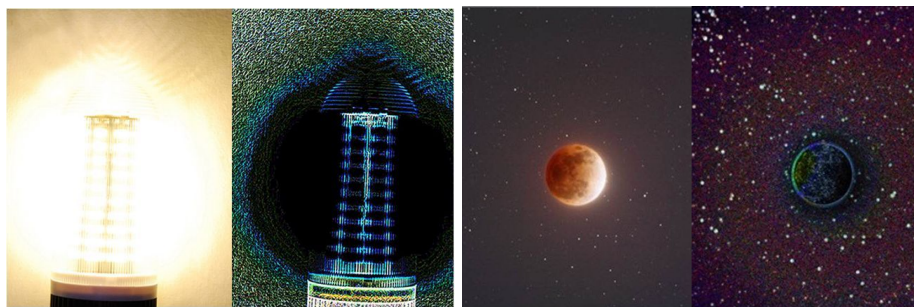


Imagen 4. Aplicaciones del algoritmo Phase Stretch Transform de los ingenieros de la UCLA sobre imágenes

Otro dato interesante del algoritmo es que puede procesar imágenes a una alta velocidad, lo que puede llegar a permitir el análisis e identificación de imágenes de video, o bien el procesamiento de una gran cantidad de imágenes en poco tiempo. Otras aplicaciones del algoritmo que mencionan los investigadores de la UCLA son la detección de células cancerígenas, análisis de huellas dactilares, caras y ojos.

Una pequeña descripción sobre cómo funciona el algoritmo es que la imagen es pasada a través de un filtro de suavizado seguido de la aplicación de una fase dependiente de la frecuencia no lineal descrita por el kernel de la fase PST. La salida de la transformación es la fase en el dominio espacial. El paso principal es la función de fase 2-D (kernel de fase PST) que normalmente se aplica en el dominio de la frecuencia. La cantidad de fase aplicada a la imagen depende de la frecuencia con una mayor cantidad de fase aplicada a las características de frecuencia más alta de la imagen. Dado que las transiciones nítidas, como los bordes y las esquinas, contienen frecuencias más altas, PST enfatiza la información de los bordes (Chin, 2016).

- Generar descripciones de imágenes con redes neuronales recurrentes

Se le conoce redes neuronales recurrentes a los algoritmos que implementan neuronas retroalimentadas, es decir neuronas que tienen capacidades de memoria en el tiempo. Para los humanos, es fácil describir una inmensa cantidad de detalles sobre una imagen. Sin embargo, esto es muy difícil realizarlo con modelos clásicos de reconocimiento visual. Normalmente el reconocimiento visual se ha centrado en etiquetar imágenes con un conjunto fijo de elementos visuales categorías. Sin embargo, mientras que los vocabularios cerrados de conceptos visuales nos otorgan modelos más sencillos de implementar, son muy restrictivos en comparación con los enormes cantidad de descripciones que un humano puede dar.

Para dar una descripción completa de una imagen utilizando redes neuronales recurrentes y convolucionales se analiza la imagen por regiones, identificando poco a poco el contexto de los posibles objetos que forman toda la imagen.

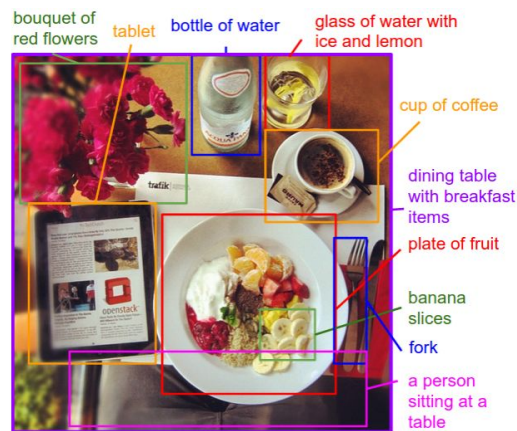


Imagen 5. Identificación de contexto por sectores

Luego basado en estos resultados se pueden unir las diferentes inferencias de la imagen en una descripción coherente y más completa de todo lo que se encuentra en la imagen original. El proceso de inferir a qué sectores corresponden diferentes descriptores se puede observar en la siguiente imagen. (Karpathy, *et. al* 2015)

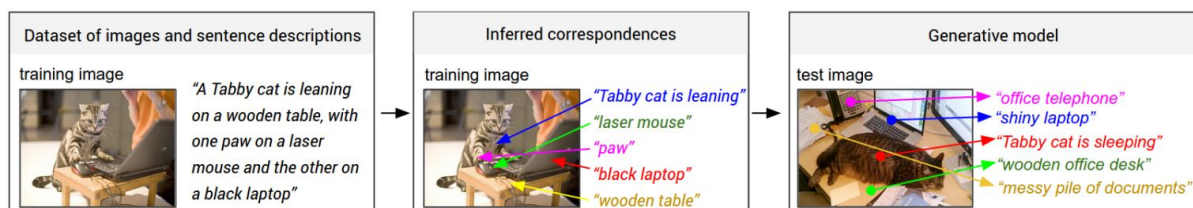


Imagen 6. Inferencia de descriptores

- Data Augmentation

Un problema en Deep Learning es que para entrenar correctamente un modelo se necesita una gran cantidad de datos. En esto nos ayuda Data Augmentation, nos permite aumentar el dataset de imágenes de 2 formas: Introduciendo perturbaciones en las imágenes, así como replicando la imagen descentrada o invirtiendo los ejes y también utilizando distintas distribuciones, por ejemplo, si se quiere entrenar un modelo de clasificación de imágenes de alta resolución, se le añadirá imágenes con ruido, baja resolución, etc, pero siempre manteniendo la mayor proporción en las imágenes de alta resolución.

Selección de algoritmos a probar

• Modelo de Convolución 2D - Keras

Para este modelo se utilizó la librería Tensor Flow y Keras, donde Keras nos provee la facilidad de utilizar el algoritmo CNN para crear un red neuronal convolucional 2D. Se toma como entrada un arreglo tridimensional el cual es perfecto para el procesamiento de imágenes con 3 canales (RGB). Lo que hace es ir haciendo reducciones de matrices generando capas con las cuales se procede a generar un modelo que pueda ser ajustado durante el entrenamiento a modo de que al hacer una clasificación de una nueva imagen, ésta pueda ser evaluada en las distintas capas hasta obtener un resultado. En este caso sería la predicción de la edad ósea. En este caso sería de mucha utilidad para este tipo de clasificación/predicción de edad ósea y por eso lo utilizaremos.

• Modelo de Convolución - MXNet

MXNet es un framework de deep learning en R, diseñado para clasificar imágenes de manera eficiente y flexible. Sin embargo, debe recibir ya preprocesadas las imágenes; en este caso, se utiliza el paquete EBImage para preprocesar las imágenes. Este framework provee las funciones necesarias para armar una red neuronal convolucional. En el primer paso, se definen cuántas capas queremos tener. Luego, se aplica la convolución, la activación y el pooling en cada capa para aplicar los filtros de características. Después de estas capas, se conectan todas por medio de flattening. Al finalizar este proceso, se tiene una capa totalmente conectada y se obtiene el output del modelo. Ya creado, se entrena con las imágenes seleccionadas para obtener una predicción de la edad ósea, por lo cual estas serían las razones por las que se utilizará este modelo.

• Modelo de Convolución - MobileNet

Al combinar las librerías de Matplotlib, Keras y Tensor Flow se cuenta con una gran variedad de funciones que permiten modificar el nivel de asertividad, según la capacidad de procesamiento y la magnitud de la data. Es por ello que se creó un modelo “entrenable” en el que se puede ajustar la cantidad de nodos y filtros a utilizar sobre la data. De esta forma no solo se toman en cuenta las dimensiones de cada elemento, sino también la cantidad de conexiones que se harán entre ellos. De la misma manera, al trabajar en Google Colab, se tienen ciertas facilidades de código. La mayor ventaja fue trabajar con un entorno GPU que permite una mayor velocidad de procesamiento que en el servidor local.

Modelos

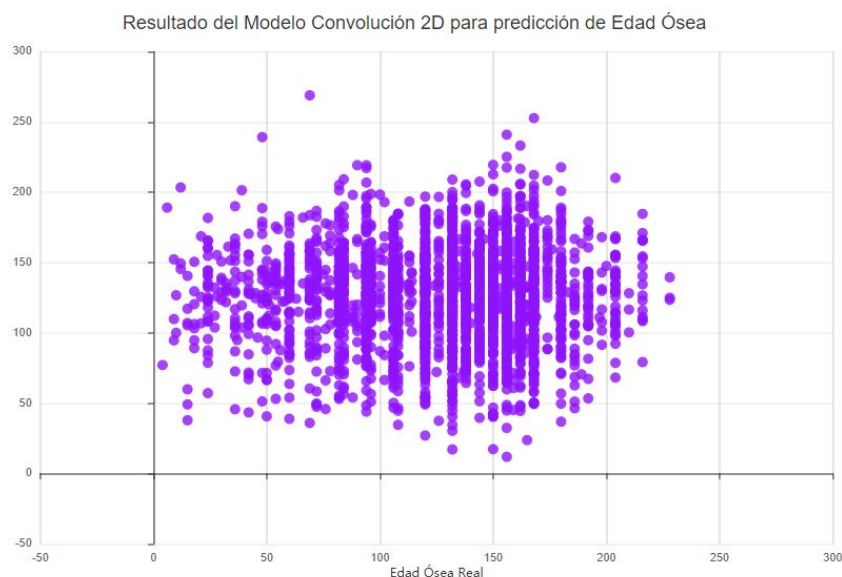
- *****Revisar archivos adjuntos con la elaboración de modelos*****

Eficiencia de modelos

- **Modelo de Convolución 2D - Keras**

En este caso el modelo no fue tan eficiente. Esto se determinó ya que al generar el modelo se utilizó el indicador del MSE (Error medio cuadrático) para poder obtener resultados sobre qué tan alejado del resultado estaba el valor real mientras el modelo se entrenaba. Y este comenzó con un error medio de aproximadamente 1300 meses, lo cual era demasiado, pero con 15 iteraciones que fueron las que se establecieron para generar el modelo para entrenar, este se redujo a alrededor de 250 meses. Esto aún es bastante, pero tomando en cuenta que eran muchos datos, quizás al sacar el error medio, el problema estaba más presente en algunas imágenes que en otras.

Esto debido a que cuando se evaluaron los resultados de algunas predicciones se podía observar que la predicción fallaba por 30 meses en algunas, en otras 70 meses, y en unas 100 meses o más. Pero en su mayoría el error era de unos 30 a 70 meses. Probablemente, algo que pudo mejorar pudieron ser los filtros a las imágenes para que el algoritmo pudiera identificar y clasificar de mejor manera en las 15 iteraciones. Lo positivo, fue observar que los resultados obtenidos iban acorde a edades óseas y no generación de predicciones con valores completamente erróneos. Dado que el resultado era una predicción de un número decimal que podía variar y no una etiqueta fija, es comprensible que el algoritmo no sea tan acertado. Haciendo una comparación por rangos de 30 meses se obtuvo un accuracy del 48.6%, por lo cual no es un modelo tan acertado como se esperaba.



Resultado de predicciones realizadas por el modelo de Convolución 2D

- **Modelo de Convolución - MXNet**

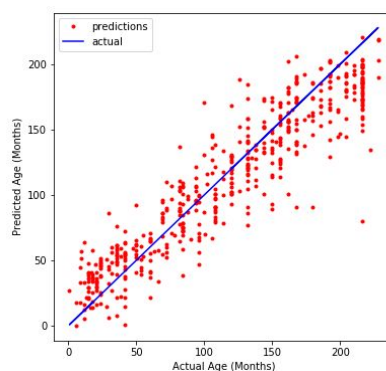
Este modelo logra clasificar categorías específicas y no números como lo es la edad ósea. Es por esto que antes de entrenar al modelo, se determinaron 10 grupos para las edades óseas (de 1-228 con saltos de 22) para que cada imagen se clasificará en alguno de estos grupos. El resultado inicial con dos capas, kernels de 5x5 y filtros de 20 y 50 respectivamente, dieron como resultado un accuracy del 48% después de 10 iteraciones. Inicialmente se determinó que el preprocesamiento de las imágenes no fue el adecuado, por lo que se aplicaron más modificaciones a las que no fueran tan claras o que se encontraran en una posición extraña. Sin embargo, después de hacer las nuevas modificaciones a las imágenes, el accuracy aumentó sólo un 5.2%.

Para mejorar el accuracy obtenido, se tomó la alternativa de modificar los valores de kernel, cantidad de filtros y número de iteraciones. Después de estos cambios, el accuracy llegó hasta 59.8% con 20 iteraciones. Un resultado similar se obtuvo al aplicar el modelo en el dataset de test. Al igual que en otros modelos, se pudieron mejorar los filtros de las imágenes, utilizando blanco y negro o aumentando más el contraste. Por otro lado, este modelo sólo clasificaba las imágenes en alguno de los 10 rangos definidos, por lo que no se puede dar un valor exacto de la edad ósea, solo un aproximado.

- **Modelo de Convolución - MobileNet**

Se inició importando todos los paquetes de Matplotlib, Keras y Tensor Flow necesarios para poder diseñar el modelo de predicción. Luego se definieron 20 epochs debido a que daría un nivel de asertividad aceptable sin recargar el peso de procesamiento. De igual forma, se estableció que cada elemento tendrá 3 dimensiones y se hizo un análisis de dimensiones para normalizar cada imagen. Al normalizarlas se lograría quitar cualquier tipo de rotación o zoom que se pudiese tener.

Al diseñar el modelo de predicción, se utilizó un formato secuencial y para maximizar el nivel de asertividad se utilizaron las funciones flatten y dense, mediante la función Max Pooling. Obteniendo una cantidad de parámetros entrenables de 22,431,617. A partir del modelo realizado, se tomó como indicador de efectividad el error absoluto medio. Esto dio un resultado de alrededor de 20 meses, el cual mejoró significativamente con respecto a los otros 2 modelos. Al hacer un análisis de rango de 20 meses se obtuvo un accuracy de 83.9%.



Discusión

A partir del uso de los modelos seleccionados para poder predecir la edad ósea de los pacientes con el procesamiento de las imágenes de radiografías de sus manos, se obtuvieron distintos resultados. Los resultados obtenidos fueron los porcentajes de efectividad al predecir en cada uno de los modelos. Con respecto al modelo de Convolución 2D se realizaron un máximo de 15 iteraciones debido a que el modelo era muy tardado para optimizar el resultado. Sin embargo, el mejor resultado que se pudo obtener fue tener un error medio cuadrático (MSE) de 250 meses. Para poder obtener un resultado de porcentaje de efectividad del modelo se permitió que el resultado tuviera una variabilidad de 30 meses, con lo que el modelo obtuvo una efectividad del 48.6% al utilizar el set de imágenes para validación.

Por otro lado, el modelo de Convolución MXNET obtuvo un porcentaje de efectividad del 59.8% al realizar 20 iteraciones para entrenar el modelo. Este modelo fue consistente al utilizar el set de imágenes para validación. Al compararlo con el primer modelo se puede observar claramente que este tuvo una mejoría considerable, aunque no fuera una mejora que nos permitiera confiar completamente en la predicción. Las ventajas de este modelo es que trabaja un poco más rápido para el proceso de entrenamiento y predicción y que tiene un mejor porcentaje de efectividad.

Al utilizar los paquetes de tensor flow y keras, se tiene la facilidad de modificar la cantidad de filtros y nodos que se le apliquen a los elementos de nuestro set. Así se puede manipular el grado de accuracy que se desea, tomando en cuenta que el tiempo de procesamiento será mayor o menor dependiendo de ello. De igual forma, se pudo seleccionar la cantidad de epochs necesarios para mejorar la asertividad y según el modelo que se desee utilizar. Cabe mencionar que se utilizó un entorno GPU para poder trabajar con la magnitud del set en un tiempo considerable. Este último modelo mejoró considerablemente, obteniendo un porcentaje de efectividad del 83.9%.

Con lo discutido anteriormente, se determinó que el mejor modelo de predicción para la edad ósea a través de imágenes es el modelo de convolución de MobileNet, tanto por su rendimiento al momento de entrenar y predecir, y porque es el modelo que obtuvo el mejor porcentaje de efectividad. Es por esto que para los resultados y visualizaciones interactivas se hará uso de este modelo para presentar los resultados. Por otro lado, es posible que los modelos eleven su eficiencia si las imágenes tuvieran otro preprocesamiento con el fin de que los modelos puedan trabajar de mejor manera con los datos, como por ejemplo otro tipo de filtrado sobre las imágenes previo a entrenar el modelo.

Visualizaciones estáticas

- Visualización infografía - La edad ósea en niños a través del análisis de radiografías:
 - <https://my.visme.co/view/x4ed4opw-3ezl33mkvpyql0q1>
- * Tener cuenta en Visme y solicitar acceso

Referencias y bibliografía

Chin, M. (10 de febrero de 2016). *UCLA researchers release open source code for powerful image detection algorithm*. Science and Technology. Extraído de: <https://newsroom.ucla.edu/releases/ucla-researchers-release-open-source-code-for-powerful-image-detection-algorithm>

Gandhi, A. (2018). *Data Augmentation | How to use Deep Learning when you have Limited Data — Part 2*. Extraído de: <https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/>

Reljin, I., & Reljin, B. (4 de octubre de 2012). *Fractal Geometry and Multifractal in Analyzing and Processing Medical Data and Images*. Belgrade, Yugoslavia: Facultad de Ingeniería.

Bonner, A. (2 de febrero del 2019). *The Complete Beginner's Guide to Deep Learning: Convolutional Neural Networks and Image Classification*. Extraído de: <https://towardsdatascience.com/wtf-is-image-classification-8e78a8235acb>

Karpathy, A. & Fei-Fei, L. (2015). *Deep Visual-Semantic Alignments for Generating Image Descriptions*. Extraído de: https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Karpathy_Deep_Visual-Semantic_Alignments_2015_CVPR_paper.pdf