

Proyecto 2 – Neural Covid19



En la lucha contra la pandemia, inicialmente surgieron formas de observar el daño pulmonar que causa el Covid19 en imágenes de rayos X. Este daño es distinto al causado por pulmonía, aunque similar. Para el ojo entrenado, puede que sea más sencillo saber las diferencias, pero en general puede ser útil tener un sistema automático que ayude a detectar esos patrones en las fotos, que sirva de apoyo a los médicos e incluso a la población en general.

Este proyecto se enfoca en un set de datos de rayos X tomados de personas con Covid19, con pulmonía, sanos, y una cuarta categoría. Es un problema supervisado, ya que tenemos los labels de cada una de las 4 categorías.

En este proyecto usaremos redes neuronales programadas en Pytorch, haciendo uso de GPUs. Vamos a comparar los resultados de usar redes tradicionales como el perceptron multicapa, contra redes convolucionales, y crearemos una visualización de aquellas zonas de la imagen que fueron de peso para clasificar la foto, para dar insights de qué tipo de patrones visuales pueden estar asociados con Covid19.

X Ray Covid-19 Dataset [1, 2]

Link: <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>

Este set de datos contiene imágenes de neumonía, COVID-19, pulmones sanos y una cuarta afectación. Contiene aproximadamente 20 mil fotos, entre las 4 clases, con resoluciones de 299x299 pixels, en formato PNG.

Arquitecturas de Redes Neuronales

Los modelos a usar no deben ser programados desde cero, puede usar código ya hecho en Pytorch para crear el entrenamiento y los modelos.

Multilayer Perceptron (MLP)

La red neuronal por excelencia, se trata de una serie de capas de tipo fully connected. Deben tener cuidado con que la red no se vuelva muy pesada para correrse en un tiempo prudencial. La red debe usarse en conjunto con algún feature extractor a escogencia del estudiante, en lugar de usar directamente los pixels de la foto.

Preprocesamiento: El feature extractor queda a escogencia del estudiante. Basicamente algoritmos estilo Local Binary Patterns, Histogramas de Color, SIFT, ORB, entre otros, permiten tomar la imagen y convertirla en un vector de menor dimensionalidad, capturando lo que cada algoritmo cree que son features más representativos. Estos features extractors no son entrenables, sino que son funciones claramente definidas, creadas a mano, pero permiten incluso procesar las imágenes antes de cualquier entrenamiento, y pueden ser usados en conjunto.

Convolutional Neural Network (CNN)

Las redes de convolución suelen ser más eficientes ya que contienen menos parámetros que aprender, pueden ser entrenadas en menos tiempo y necesitan menos espacio en memoria. Hay muchos modelos de redes de convolución, algunos **MUY** pesados. Deben buscar uno que puedan correr en su ambiente de trabajo, o bien, crear su propia red de convolución más pequeña.

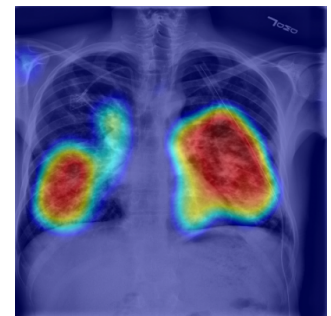
Preprocesamiento: se debe comparar correr la CNN con las imágenes crudas, y también con un filtro llamado Bilateral Filter. Este filtro toma en cuenta el vecindario de cada pixel para reemplazar su valor, con un efecto de suavidad y de preservación de bordes. Existen muchas implementaciones, incluyendo la de OpenCV. Se recomienda hacer el filtrado de las imágenes offline, no durante el entrenamiento, y guardarlas ya filtradas para evitar tener que filtrar una y otra vez.

Data augmentation

Se recomienda hacer data augmentation, usando transformaciones de distintos tipos, pero se deja a criterio del estudiante, dependiendo del tiempo de entrenamiento, ya que la cantidad de datos incrementa con cada transformación.

Visualización de Mapas de Calor

Se debe visualizar, una vez entrenada alguna de las redes, cuáles son las zonas de la imagen que más influencia tuvieron en la toma de decisiones respecto a la clasificación final. Se recomienda usar Saliency Maps, aunque es posible usar otras técnicas. Pueden usar código de internet para esto, y solo deben visualizar una de las arquitecturas.



Herramientas

Las redes neuronales deben implementarse en Pytorch, usando Python. Se recomienda el uso de GPUs en Google Colab, subiendo el set de datos en Google Drive. De tener GPU propio, todavía mejor. Otra opción para obtener GPUs es usar el cluster Kabré de CeNAT, o bien en caso de tener la posibilidad, Azure o AWS son opciones viables.

Métricas/Conclusiones

Como mínimo, deben reportar Accuracy por clase, así como otras métricas como la matriz de confusión y sus derivados. Recuerden que los resultados deben ser en términos de test sets (aunque pueden mencionar los de training set, especialmente para hablar sobre Overfitting o Underfitting). En general, el test set debería ser el mismo para las 2 implementaciones. Por cuestiones computacionales, no se recomienda hacer cross validation, salvo si cuentan con muy buen equipo computacional y el tiempo para hacerlo.

Pueden leer papers que han trabajado este mismo set de datos y darse ideas para aplicar en el proyecto, además de tomarlos como punto de comparación con sus propios experimentos.

Es importante responder de forma objetiva, qué tan bueno es usar feature extractors con un MLP versus los pixels crudos en la CNN. ¿Cuál es el tamaño de los modelos? Qué ventajas tiene usar feature extractor sobre los pixels, si es que hay alguna. Qué modelo se entrena más rápido con respecto a la cantidad de parámetros. La idea es que comparen ambos enfoques lo mejor que se pueda, con números. Finalmente, qué “ve” cada modelo en la imagen para decidir si es realmente Covid19 o no.

Paper en Latex / PDF

Deben crear en latex (y entregar los fuentes de latex) un paper donde describen los experimentos realizados, su metodología, así como su escogencia de feature extractor. Deben incluir los resultados y por supuesto sus propias conclusiones. El documento debe seguir el template de la IEEE para publicaciones en ingeniería el cual pueden encontrar aquí:

<https://www.ieee.org/conferences/publishing/templates.html>

Evaluación

Tarea	Puntaje Máximo
MLP con feature extractor- Experimentos+Análisis	20
CNN sin filtros - Experimentos+Análisis	20
CNN+Filtro bilateral - Experimentos+Análisis	20
Visualización de mapas de calor	10
Paper y análisis de resultados global y comparativos, conclusiones	30
TOTAL	100

[1] M.E.H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z.B. Mahbub, K.R. Islam, M.S. Khan, A. Iqbal, N. Al-Emadi, M.B.I. Reaz, M. T. Islam, “Can AI help in screening Viral and COVID-19 pneumonia?” IEEE Access, Vol. 8, 2020, pp. 132665 - 132676. [Paper link](#)

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Maestría en Ciencias de la Computación
Curso: Aprendizaje Automático
Profesor: Dr. José Carranza-Rojas
Valor: 20%
Proyecto en parejas o tríos

Semestre 1, 2021

[2] Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Kashem, S.B.A., Islam, M.T., Maadeed, S.A., Zughaier, S.M., Khan, M.S. and Chowdhury, M.E., 2020. Exploring the Effect of Image Enhancement Techniques on COVID-19 Detection using Chest X-ray Images. [Paper Link](#)