Image Segmentation for COVID-19 CT Scans

De Mónica Monserrat Martínez Vásquez -| A01710965 | Tomas Perez Vera - A01025008 | Bárbara Paola Alcántara Vega - A01799609 | María José Soto Castro - A01705840 | Ulises Orlando Carrizalez Lerin - A01027715

Inteligencia artificial avanzada para la ciencia de datos I (Gpo 101)

Fecha: 18/09/2025

1. INTRODUCTION

The COVID-19 pandemic has posed unprecedented challenges to healthcare systems worldwide. One of the key tools assessing disease severity in patients with COVID-19 is computed tomography (CT) imaging. CT scans allow radiologists to visualize the extent of lung damage, in the form of "ground-glass opacities" and "consilidations". However, this process is time consuming and tends to vary, as it requires manual segmentation of hundreds of axial slices by specialists.

This project presents a baseline deep learning pipeline for segmentation of COVID-19 CT images using Pytorch. The goal is to automatically segment pathological regions, particularly ground-glass and consolidation patterns, from 2D CT slices. The project is aligned with the objectives of the "COVID-19 CT Images Segmentation" challenge and aims to support the development of AI-based tools for automated radiological analysis.

The solution is evaluated using mean Intersection over Union metric, averaged over all pixels and classes (ground-glass and consolidation). This baseline serves as a starting point for future optimization and experimentation with more advanced segmentation models and data augmentation techniques.

2. DESCRIPTION AND EXTRACTION DATA

The dataset used in this project is divided of two main sources:

- MedSeg
- Radiopaedia

Both of which provide annotated axial CT slices of COVID-19 patients. All images and masks are standardized to a resolution of 512×512 pixels, ensuring consistency for training and inference.

2.1 MEDSEG DATASET

This subset contains 100 CT slices and corresponding segmentation masks from over 40 patients. The ground-truth masks are multi-channel and labeled as follows:

- Class 0: Ground-glass opacity
- Class 1: Consolidation
- Class 2: Lungs other (not relevant for this task)
- Class 3: Background (not relevant for this task)

A small test set of 10 images is also extracted from this dataset.

2.2 RADIOPAEDIA DATASET

This dataset includes 829 CT slices extracted from full volumetric scans. It shares the same mask labeling structure as MedSeg. However, since it originates from full 3D volumes, it offers richer contextual variation and increased sample size for training.

2.3 DATA LOADING

The images and masks are provided in NumPy format, and are loaded into memory as float32 and int8 arrays, respectively.

Once loaded, the images are visualized in grayscale using a custom function. The segmentation masks for the two relevant classes (ground-glass and consolidation) are displayed below each corresponding image for inspection. These masks are one-hot encoded, and a helper function is defined to convert them into RGB format for visualization.

The masks of interest (classes 0 and 1) represent only a small portion of each image.

3. IMAGE PREPROCESSING

Medical images obtained from CT scans contain outliers and varying distributions. So to ensure consistency and enhance model performance, we apply a normalization procedure to all images arrays.

The raw pixel intensity values in CT images typically range from −1500 to +500 Hounsfield Units (HU). To reduce the influence of outliers and standardize the range of inputs, all pixel values are clipped within this interval

After clipping, the distribution of intensity values is restricted to the 5th and 95th percentiles. The mean and standard deviation are computed only on this valid range, and the image is normalized using Z-score standardization (Figure 1).

$$x_{norm} = \frac{x-\mu}{\sigma}$$

Figure 1. Formula of Z-score standardization

This preprocessing step ensures that the pixel intensities follow a normal distribution with mean of 0 and standard deviation of 1, which is optimal for training deep neural networks.

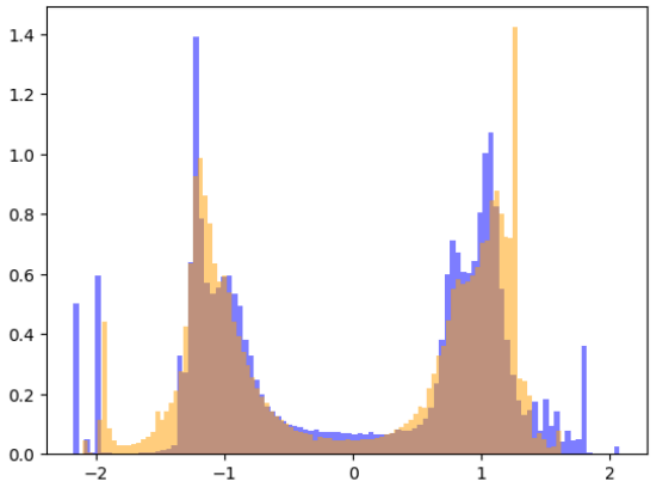

Figure 2. Histogram of pixel intensities [9]

As seen in Figure 2, we can validate the effect of preprocessing with a histogram of pixel intensities before and after normalization. We ensure that the transformation does not introduce unexpected distortions and that the network receives consistent input across different samples and datasets.

4. EVALUATION METRICS

To assess the performance of the semantic segmentation model, we implemented two key evaluation metrics that are pixel-wise accuracy and mean Intersection over Union (mIoU). These metrics are widely used in segmentation tasks due to their effectiveness in capturing both overall classification accuracy and class-specific overlap.

Pixel accuracy measures the proportion of correctly predicted pixels over the total number of pixels in the image. It provides a global estimate of performance but does not differentiate between classes, which may be misleading in the case of class imbalance.

The mean Intersection over Union is a more informative metric that evaluates the overlap between the predicted and ground-truth segmentation masks for each class.

While pixel accuracy gives a general overview, mIoU captures the spatial agreement for each class, which is particularly useful in medical segmentation tasks where some classes may be small but clinically significant.

5. MODEL TRAINING AND VALIDATION

The training loop is implemented using PyTorch and follows a standard supervised learning pipeline. It includes forward propagation, loss calculation, backpropagation, and optimization steps. Additionally, evaluation metrics such as mean IoU and pixel accuracy are computed at each epoch, and learning curves are recorded.

The validation images used were 24, or 24% of the available images. The images are extracted from the medseg dataset.

During the training the model used is U-Net with an encoder of efficientnet-b2. This model was originally created to be used for medical imaging. And the encoder was developed by google and uses MBConv, instead of maxpooling [3].

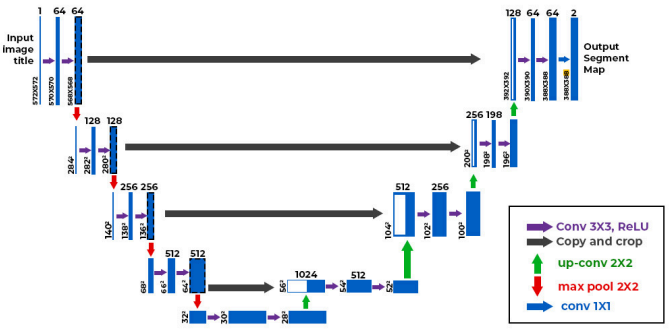

Figure 3. U-Net architecture [2]

For each epoch we include model training, validation phase, learning rate adjustment and early stopping and checkpointing. First, for each batch in the training set, the mode performs a forward pass, computes the loss using a predefined criteria and updates the weights via backpropagation.

The model is now evaluated on a separate validation set without gradient updates. All the metrics that mean IoU and accuracy are computed and stored. Then the LR updates using a scheduler after each batch. Finally, if validation loss does not improve for 7 epochs, training is stopped. The model is saved every 5th time an improvement is detected.

In the training process, an optimizer (AdamW) is used to prevent overfitting. The AdamW optimizer uses the momentum and L2 regularization, to optimize the learning rate. It is used on complex models and large scale datasets. [4]

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\frac{\partial L}{\partial w_t}$$

Figure 4. Formula Adam Optimizer [4]

$m_t = moving\ average\ of\ the\ gradients\ at\ time\ t$

$\alpha = learning\ rate$

$\beta_1 = momentum\ parameter$

$w_t = weight\ at\ time\ t$

To optimize the learning rate from the optimizer, OneCycleLearningRate scheduler was used. Which consists of gradually increasing the learning rate, to a determined maximum, and later gradually decreasing the learning rate to a smaller value [5].

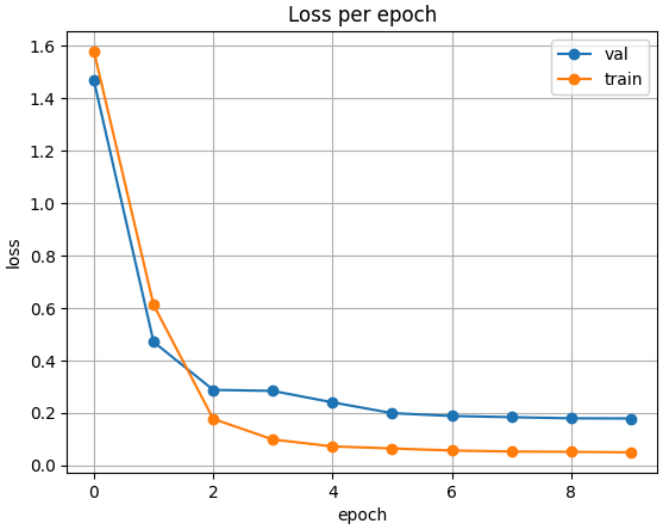

Figure 5. Loss per epoch history [9]

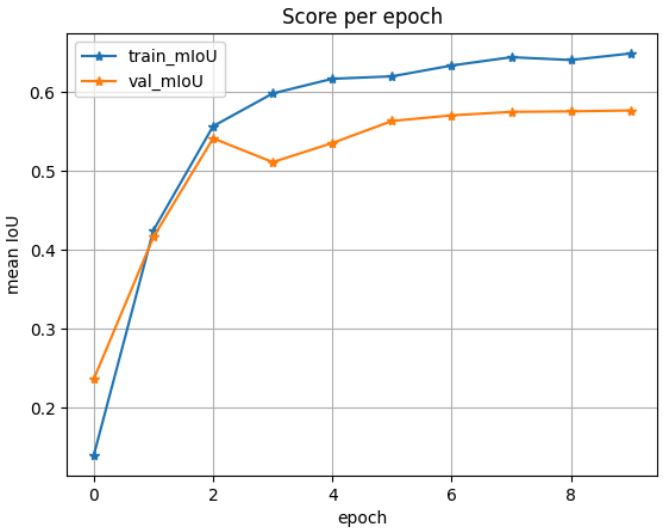

Figure 6. Score per epoch history [9]

Based on the previous graphics, it can be seen that along the training loop, the loss between the training set and the validation set tends to decrease in a more likely form, concluding that the model enters on the fit scenario.

On the other hand, the score per epoch graph, shows the mIoU evolution over epochs. Based on the factor that the mIoU grows on both, the validation and training set, this shows the factor that the model is capable on segment the classes in a proper way, nevertheless, the mIoU score, neither on the training set

nor on the validation set, reaches a value capable of having a solid segmentation.

6. VISUALIZATION AND RESULTS

On the first image, we can see the predictions of the model segmented by category and the original images. The yellow area represents the area predicted. The rows showing in order from top to bottom:

- Ground-Glass: Fluids
- Inflamed Pulmonary Tissue
- Healthy Pulmonary Tissue
- Background

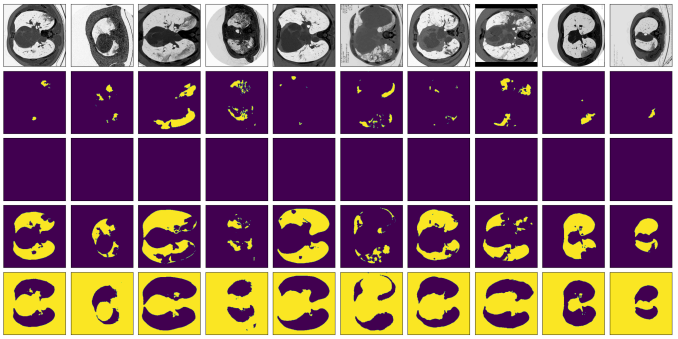

Figure 7. Prediction divided by category [9]

On the second image, the rows are segmented as such:

1. Predictions of the model, ground-glass [green], inflamed tissue [red], mixed label [yellow]
2. Images labeled by professionals on the subject ground-glass [green], inflamed tissue [red.
3. Overlapping of predictions and professional labeling. Correct labeling [red and green], missed labeling [yellow]
4. Prediction of the model, background [yellow], pulmonary tissue [black].

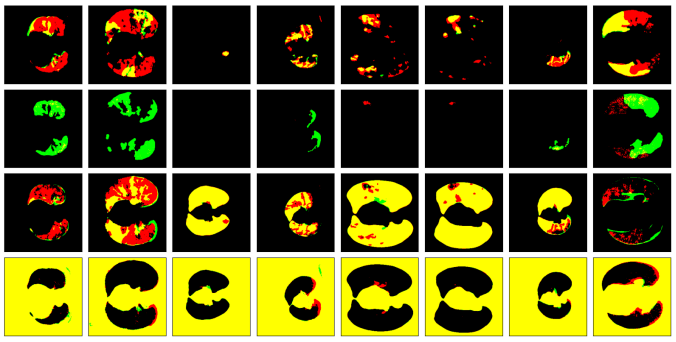

Figure 8. Mask prediction comparison [9]

7. FUTURE IMPROVEMENTS

While the current model provides a solid baseline, several improvements can be explored in future work, to enhance performance and real life applications.

- Increase the quality and quantity of data (more 3D volumes or using external datasets like MosMed).
- Improve the consistency of annotations to avoid overlaps between classes in the masks.
- Explore more advanced architectures (e.g., U-Net++ or attention-based models) to enhance detection in challenging areas.
- Implement additional metrics (e.g., Dice coefficient) to complement evaluation and provide a more clinically relevant perspective.

8. CONCLUSION

The model demonstrates an acceptable ability to identify the regions of interest from the diagnosis for COVID-19. From the processing steps up to the training process of the model, we achieve a robust model, in order to properly categorize CT Scans.

Some limitations for the model were the size of the dataset of roughly 900 images and the even smaller number of patients used on this model. Reducing the range of images used in a real life scenario.

There is also the experimentation and time ratio, that we could have experimented more on the training process, like removing the early stopping, or increasing the number of epochs to compare the results of each other.

Overall, the model resulted to be a robust baseline to increase the research on this industry and the diagnosis of COVID-19.

Tecnológico de Monterrey

Referencias

[1] Cantoral Ph.D. (2025). *Understanding Deep Learning Models*. YouTube. https://www.youtube.com/watch?v=x_cY9l1cwj0

[2] GeeksforGeeks. (2023, June 8). UNet Architecture Explained. GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/u-net-architecture-explained/

[3] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ArXiv.org. https://arxiv.org/abs/1905.11946

[4] GeeksforGeeks. (2020, October 22). *What is Adam Optimizer?* GeeksforGeeks. https://www.geeksforgeeks.org/deep-learning/adam-optimizer/

[5] G M. (2024, December 11). *One Cycle LR Scheduler: A Simple Guide*. Medium. https://medium.com/@g.martino8/one-cycle-lr-scheduler-a-simple-guide-c3aa9c4cbd9f

[6] Cantoral Ph.D. (2025). Semantic Segmentation with PyTorch. YouTube. https://www.youtube.com/watch?v=dfMEK4bKjRE

[7] Igor Slinko. *COVID-19 CT Images Segmentation*. Kaggle, 2020. Available: https://kaggle.com/competitions/covid-segmentation

[8] M. Maftouni. (2021, May 8). PyTorch Baseline for Semantic Segmentation. Kaggle. https://www.kaggle.com/code/maedemaftouni/pytorch-baseline-for-semantic-segmentation

[9] M. Tan and Q. V. Le. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ArXiv. https://arxiv.org/abs/1905.11946

[8] O. Ronneberger, P. Fischer, and T. Brox. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. ArXiv. https://arxiv.org/abs/1505.04597

[9] Alcantara, B. Carrizales, U. Martinez, M. Perez, T. Soto, M,*Monica Martinez V/Mod 3.Desarrollo De Software Aplicado A Ciencia Datos*. (2025). GitHub. https://github.com/MonicaMMartinezV/Mod3.DesarrolloDeSoftwareAplicadoACienciaDatos/tree/main