**Pneumonia detection using Deep Learning**

**Author:** Ayah Kouider

**Problem Statement:**

Pneumonia is a potentially life-threatening lung infection that inflames the air sacs. It is one of the leading causes of death for children under the age of 5 and is responsible for millions of hospitalizations globally each year. Despite being treatable, early and accurate diagnosis remains a challenge in low resource areas where access to trained radiologists is limited.

In this project we aim to develop a deep learning-based system for pneumonia detection in chest X-ray images. The goal is to support overwhelmed healthcare systems, speed up detection, reduce missed cases and provide scalable solutions in underserved areas. This research is significant due to its potential to improve health outcomes, particularly for vulnerable populations, and it is interesting due to the technical challenges involved in medical image analysis, including data imbalance, image variability, and the need for model interpretability.

**Dataset:RSNA pneumonia detection dataset**

The dataset used in this project is the RSNA Pneumonia detection data set on Kaggle. It contains over 30,000 chest X-ray images in DICOM format, each named after a unique patient id collected from pediatric patients. Each image is labeled by certified radiologists with annotations indicating the presence of pneumonia and bounding boxes marking the affected lung regions.

**The dataset includes 2 key CSV files:**

1. **Stage\_2\_train\_labels.csv:** This file includes both the classification labels and bounding box annotations. Each row includes:

* **Patient\_id:** Unique identifier corresponding to a DICOM image.
* **Bounding box coordinates (x,y,width,height):** provided only for pneumonia patients. NAN for non-pneumonia cases.
* **Target:** Binary indicator 1=pneumonia and 0=no pneumonia.

1. **stage\_2\_detailed\_class\_info.csv:** provides additional class information for each patient id categorized as:

* Normal
* Lung Opacity (pneumonia)
* No lung Opacity/Not Normal (not normal but not pneumonia)

**Challenges:**

* **Data imbalance:** Pneumonia cases make up 1/3 of the dataset, making them about half as common as non-pneumonia cases.
* **Sparse annotations:** Only pneumonia images include bounding boxes, which limit training data for detection.

Despite these challenges, the dataset offers a realistic clinical scenario and is suitable for developing and evaluating both classification and object detection models.

**Methodology:**

1. **Data preprocessing:**

The first step in the development process is data preprocessing. We began by loading the chest X-ray images using the pydicom library and converting them into pixel arrays for further processing. These arrays were then resized to a consistent resolution using OpenCV to standardize the model input size. The bounding box coordinates from the stage\_2\_train\_labels.csv file were converted from the format (x, y, width, height) to (x1, y1, x2, y2) for compatibility with common object detection models. Class labels from the stage\_2\_detailed\_class\_info.csv file were transformed into binary form: 0 for no pneumonia and 1 for pneumonia.

We structured the data into arrays and created a custom MedicalDataset class to convert data into tensors. Finally, the dataset was split into training and validation sets using an 80/20 ratio.

1. **Handling Data imbalance:**

The Dataset presents a serious data imbalance with pneumonia cases being half as common as no pneumonia cases. To address this, we tried different techniques including undersampling,oversampling,stratified K-fold cross-validation, and using a weighted sampler in the Data loader. We also tried training solely on pneumonia cases and fine tuning on combined data to increase the models focus on minority class. However, most of these approaches resulted in low F1 scores and poor Recall indicating they were not effective for this task.

The most successful approach combined a manual train/validation split, undersampling of the majority class, and the use of a weighted sampler. we separated the positive (pneumonia) and negative (non-pneumonia) cases, applied undersampling to reduce the size of the negative class, and then split each class independently into training and validation sets. These were then merged to form balanced train and validation sets. A weighted sampler was also applied during training to further emphasize the minority class and reduce prediction bias.

1. **Model architecture:**

For the main detection model, we used RetinaNet, a one-stage object detection architecture known for its effectiveness in handling class imbalance through the use of Focal Loss. RetinaNet offers a strong balance between speed and accuracy, and it performs well at detecting challenging or small objects, making it well-suited for medical image analysis. We also experimented with Faster R-CNN, a two-stage detector that often achieves high accuracy in medical tasks due to its region proposal mechanism. In addition, we tested an ensemble model that combined predictions from both RetinaNet and Faster R-CNN to evaluate whether it would yield improved performance. While all three models achieved similar F1 scores and AUC values, their strengths varied: some were better at detecting positive (pneumonia) cases, while others performed better on negative cases.

All three models were trained for **10 epochs** on a GPU using the **AdamW** optimizer, which couples Adam’s adaptive moments with decoupled weight decay to improve generalization. An **LR scheduler** was employed to automatically reduce the learning rate when validation loss plateaued, helping the models converge without over-fitting.

1. **Related work and comparison:**

Top competitors in the RSNA challenge used several techniques for tackling data imbalance and training.They used model level techniques rather than dataset level techniques to address data imbalance. [3] JiYuanFeng (5th place) utilized Faster/Mask R-CNN for detection and DenseNet169 for classification, applying cross-validation training, Online Hard Example Mining (OHEM), and a classification branch to help reduce false positives.

Similarly, competitors like Ian Pan & Alexandre Cadrin(1st place)[1] and [Dmytro Poplavskiy](https://www.kaggle.com/dmytropoplavskiy) (2nd place)[3] employed stratified K-fold cross-validation and trained models primarily on positive samples to address the imbalance, often followed by fine-tuning on the complete dataset. These approaches emphasized model tuning and ensembling methods such as multi-scale inputs, intra-model NMS, and box voting strategies.

However, after testing methods like stratified K-fold and positive-sample-only training , we found that they didn’t improve performance in our case. Instead we adopted a more dataset level approach by manually under sampling the negative samples and using a weighted sampler during training to balance class contributions. We also tested RetinaNet, Faster R-CNN, and an ensemble of both to evaluate performance across architectures. Unlike many competitors who focused on model-level adjustments alone, our approach directly tackled the data imbalance at the dataset level in addition to architectural experimentation, allowing for more controlled learning and better generalization. This strategy proved more effective for our pipeline, especially in reducing false positives and improving overall precision.

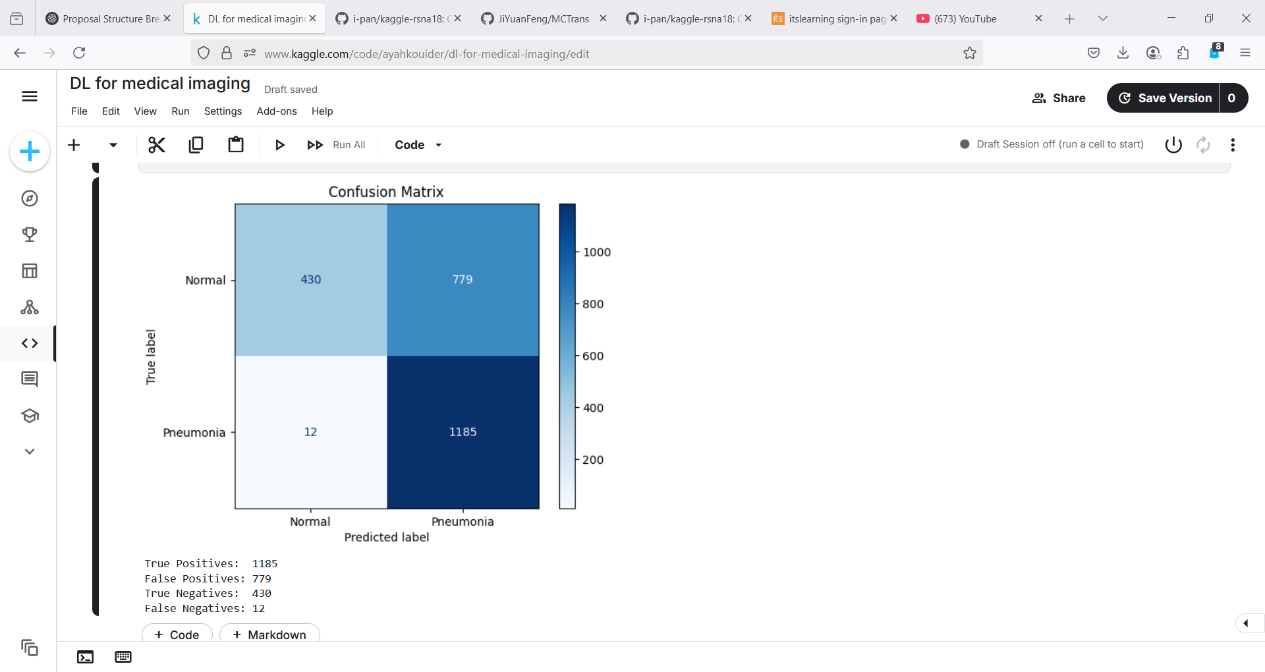
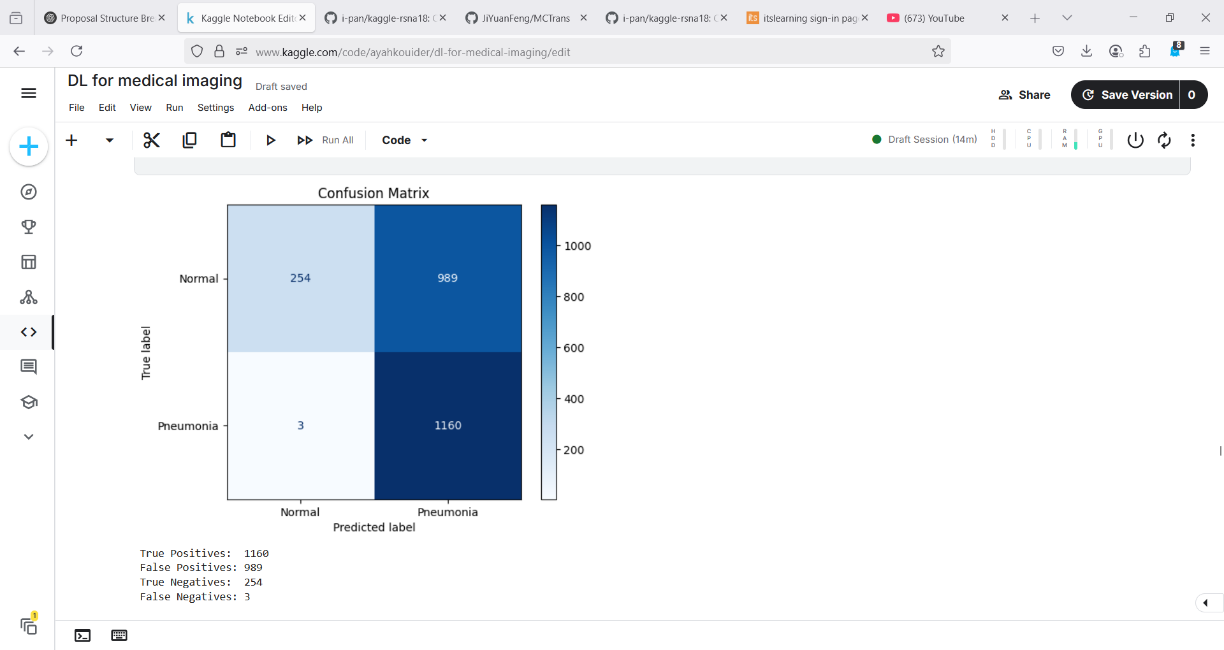
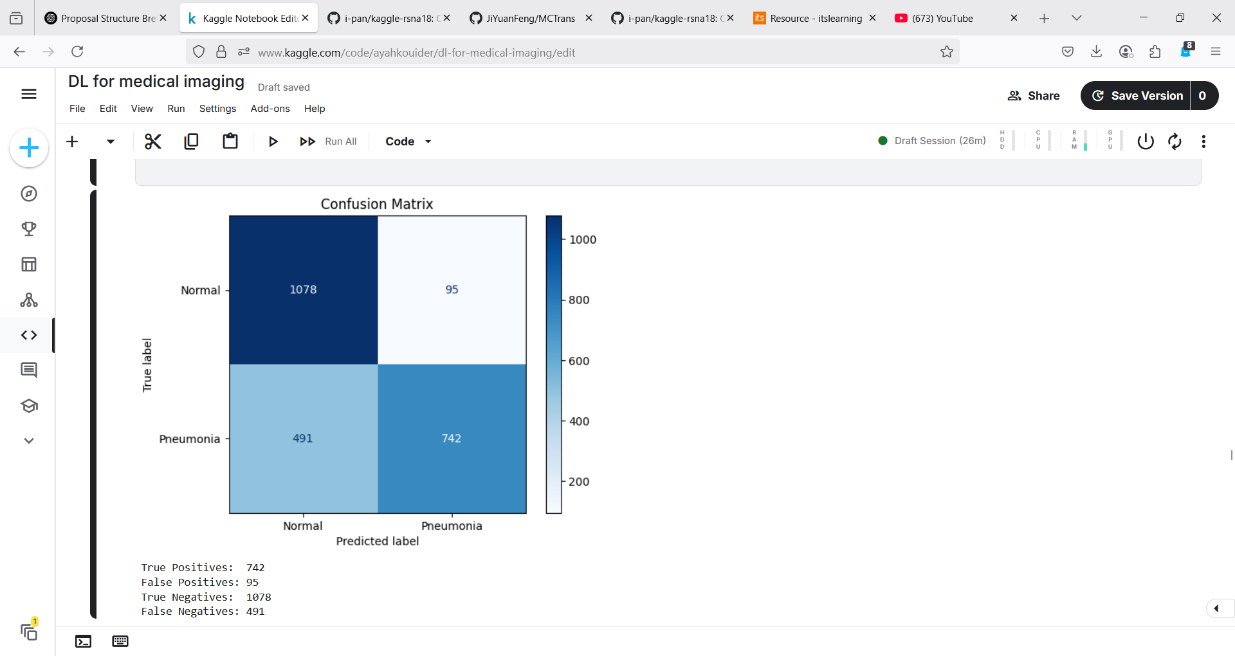
**Results and Evaluation:**

To evaluate the effectiveness of each model and strategy we compared performance using Precision, Recall, F1-Score, and AUC.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1-score | AUC | Total training time |
| RetinaNet (imbalanced) | 1 | 0.0016 | 0.003 | 0.87 | 10 hours |
| RetinaNet (balanced) | 0.90 | 0.61 | 0.73 | 0.90 | 2.5 hours |
| Fast RCNN | 0.63 | 0.94 | 0.76 | 0.89 | 2 hours |
| Ensemble (RetinaNet + Fast RCNN) | 0.60 | 0.99 | 0.74 | 0.88 | - |

Based on the evaluation metrics, all models trained on the balanced dataset performed comparably, each excelling in different aspects. RetinaNet achieved the highest AUC (0.90), indicating its strong overall ability to distinguish between classes. Fast R-CNN yielded the highest F1-score (0.76), demonstrating the best balance between precision and recall among the individual models. Meanwhile, the ensemble model attained the highest recall (0.99), making it the most effective at identifying positive cases, which is particularly valuable in imbalanced classification tasks where false negatives are critical.

**Confusion matrices for RetinaNet(balanced), Fast RCNN and Ensemble.**



RetinaNet achieved the highest True negatives and lowest false positives reflecting high precision and avoiding false alarms. Fast RCNN identified the most true positives the and lowest false negatives but very high false positives.Since RetinaNet is better at detecting negatives and Fast RCNN is better at detecting positive we tested an ensemble of both, the ensemble achieved the highest true positives overall while also improving the number of true negatives compared to Fast R-CNN. It maintained a reasonable balance by slightly increasing false negatives but reducing false positives, offering a more robust trade-off between sensitivity and precision.

**Evaluation Based on IoU Score**

The competition's primary evaluation metric was the Intersection over Union (IoU), which measures the overlap between the predicted bounding boxes and the ground truth annotations. Top-ranking solutions in the competition achieved IoU scores in the range of approximately 0.23 to 0.25.

In comparison, our RetinaNet model trained on balanced data achieved an IoU score of **0.015**, indicating very limited overlap with the ground truth boxes. Both the Fast R-CNN and the ensemble model performed significantly better, each achieving an IoU score of **0.089**. While still below the top competitors’ scores, these results show a meaningful improvement over RetinaNet alone and reflect the benefit of combining models with complementary strengths.

**Conclusion:**

In this project, we explored multiple object detection models to address the challenge of detecting specific targets in a highly imbalanced dataset. We implemented and compared RetinaNet, Fast R-CNN, and an ensemble of both. RetinaNet excelled in detecting negative and had low false positives while Fast rcnn excelled at detecting positives but achived high false positive. To leverage the strengths of both models, we developed an ensemble approach, which achieved the highest recall and true positive count among all tested methods. It also improved the precision compared to Fast R-CNN.

However, despite these improvements in classification metrics, all models showed relatively low performance in terms of Intersection over Union (IoU), with the ensemble achieving a maximum IoU score of only 0.089. This is considerably lower than the top competition results, which ranged from 0.23 to 0.25, leaving substantial room for improvement in terms of precise localization of pneumonia regions.

**References:**

[1] I. Pan, "1st Place Solution," \*Kaggle RSNA Pneumonia Detection Challenge\*, 2018. [Online]. Available: https://www.kaggle.com/competitions/rsna-pneumonia-detection-challenge/discussion/70421

[2] D. Poplavskiy, "2nd Place Solution," \*Kaggle RSNA Pneumonia Detection Challenge\*, 2018. [Online]. Available: https://www.kaggle.com/competitions/rsna-pneumonia-detection-challenge/discussion/70427

[3] J. Feng (JiYuanFeng), "5th Place Solution," \*Kaggle RSNA Pneumonia Detection Challenge\*, 2018. [Online]. Available: <https://www.kaggle.com/competitions/rsna-pneumonia-detection-challenge/discussion/79381>

[4] Radiological Society of North America (RSNA), “RSNA Pneumonia Detection Challenge,” Kaggle, 2018. [Online]. Available: https://www.kaggle.com/competitions/rsna-pneumonia-detection-challenge/data