Transformer-Based Detection of B-Lines in Lung Ultrasound

Pedro Serrano
54853
fc54853@alunos.ciencias.ulisboa.pt

Guilherme Cepeda
62931
fc62931@alunos.ciencias.ulisboa.pt

Abstract

Lung Ultra-sounds (LUS) have been a widely used tool for patient assessment for the last 50 years. However, subjectivity still applies when counting and evaluating the presence of B-lines, which are artifacts that appear to be heavily correlated with pulmonary congestion. In this work, we evaluate the performance of state-of-the-art transformer models against a deep learning approach benchmark to attempt to classify the presence of B-Lines in patient's Lung Ultra-sounds. Transformer models do achieve better results, having reached 0.864 f1-score, and 0.934 Area Under the Curve.

1. Introduction

Lung ultrasonography (LUS) has emerged as an important bedside imaging modality to support diagnostic assessment and therapeutic management in acute care settings. B-line artifacts in LUS are defined as hyperechoic lines that originate from the pleura and extend radially to the bottom of the screen, while moving synchronously with respiration. Despite the artifactual nature, B-lines play a key role in detecting and assessing the severity of pulmonary congestion in patients with disease states such as decompensated heart failure, chronic kidney disease on dialysis, viral and bacterial pneumonia, or interstitial lung disease. In clinical practice, physicians generally assign scores for disease severity based on a visual estimate of the quantity of B-lines. However, several studies have found both intra- and inter-observer variability in B-line quantifications [1].

During the COVID-19 pandemic, and before polymerase chain reaction (PCR) tests were normalized, lung ultrasound exams were a big factor in assessing the situation, and one of the aspects monitored closely were the presence and severity of B-lines in the LUS. Good tools for automated assessment of LUS could potentially reduce diagnostic errors, and help novice medical practitioners to correctly treat and care for their patients.

1.1. Related work

The authors in [3] curated and published BED-LUS, a lung ultrasound dataset, comprising 1.419 LUS videos, of 113 total different patients, with 15.775 B-lines annotaded by experts. Based on this dataset, the authors also present present deep learning benchmark values for B-line detection and quantification. They did this, by analysing the LUS in three different levels - multi-frame clips, single frames, and individual pixels. When dealing with clip, and frame level, the authors were attempting to solely classify the element as 'in the presence of B-lines' or not. However, in pixel level analysis, the focus was on segmentation, and they developed a novel way to identify B-lines, by identifying the origin pixel. After detecting such pixels, counting the total number is trivial.

Focusing on the frame level analysis, the authors ran a series of deep learning models, and published both the dataset and results to serve as a benchmark for future work.

2. Contributions

The main contributions of our work is to build on the work of [3], and use state-of-the-art transformer models in the same BED-LUS dataset. We achieved results of 0.864 f1-score, and 0.934 Area Under the Curve, by using frame-level analysis.

3. Dataset

The dataset used was BED-LUS again the same as [3]. The Boston Emergency Department Lung UltraSound dataset consists on data collected from patients admitted to the emergency department of Brigham's and Women's Hospital between November 2020, and March 2021, with symptoms suggestive of a flu-like illness. A total of 113 patients were examined, having each one typically taking 12 LUS over different positions, resulting in 1,419 videos, comprising 188,670 frames. The average age of the patient is 60 years old, 55% of them female. Also, 55% got discharge diagnosis that are typically correlated with the development of B-lines, while the remaining 45% had discharge diagnostics that have no correlation with this condition, making

this dataset relatively balanced. Of these 1,419 videos, 719 (50.7%) had been classified as 'in the presence of B-lines', while the remaining haven't.

The annotations were consensual by two experts, N.D. and A.G., where they indicate the presence of at least one B-line throughout the video. In the annotations in the dataset, we have the pixel coordinates for the origin of each B-line.

3.1. Data Processing

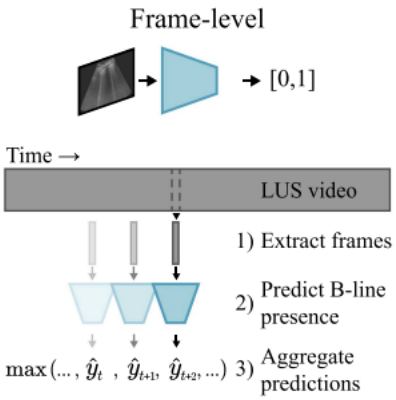

Figure 1. Frame Level Classification Architecture

Figure 2. Frame Level Classification Architecture

For the task of B-line detection at the frame level, there is a certain level of data-processing we must complete. Since what we have are videos with .txt labels attached, we must separate each raw video into multiple separated frames, and still have all annotations correctly representing each frame.

3.1.1 Sizing and treating

Since we are using transformers to classify our images, we must define a specific size, as transformers don't generally allow for different size inputs. Below, we display one of the frames.

In order to isolate the applicable information, we must deal with Lung UltraSounds labels and captions. Also, we must also pad the frame to a standard aspect ratio, and resize to the desired shape.

3.1.2 Dataset generation

For the data generation, we gathered the preprocessed frames from which datasets with labels or annotations can be created for training deep neural networks at the single frames level.

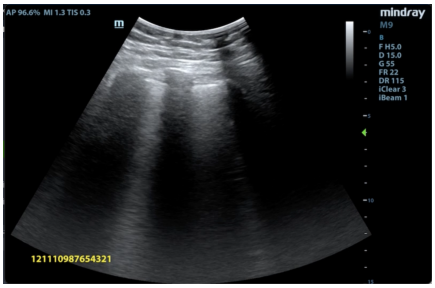

Figure 3. Unprocessed Frame

The dataset generation is divided into 3 steps:

- **Dataset Split** - Here we created the trained and test set splits. For the training set, we subdivided it into different folds for cross-validation.
- **Dataset Creation** - In this step we used the frame images that were prepared during the data preparation/processing stage. These images are then separated based on the dataset split that they were prepared in and whether they were labeled positive or negative.
- **Labelset Creation** - In this final step we created the label for the dataset that was prepared in the previous step. For this classification task, only a file with the corresponding label information is created.
  The labelmaps are separated based on the dataset split that was prepared. Only a single empty labelmap is created for all negative cases of a specific fold.

4. Model

4.1. Problem Formulation

Here, we are presented with a classification problem, meaning we are learning a mapping from images to a binary number.

4.1.1 Models used

We evaluated the following hugging face models:

- Vision transformer - variant tiny
- SWin transformer - base variant
- DeiT transformer - base variant
- CaiT transformer - base variant

All transformers were pre-trained with ImageNet-1k. And have an image size of 384x384.

4.1.2 Loss Function

Our transformers are trained by minimizing the Binary Cross-Entropy loss function (BCE), as usual with binary

classification models:

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p_i) + (1-y_i) \cdot log(1-p_i))$$

Here, for all N frames, y_i represents the real value, whereas p_i represents the model prediction for the same frame.

4.2. Evaluation

As per usual with the hugging face transformers, we get a series of outputs. However, we will focus on the F1-scores, and the Receiving Operating Characteristic (ROC), abbreviated as the AUC, to enable a direct comparison to the [3].

The F1-score is a simple metric that attempts to find a middle ground between precision and recall, and is represented by a number between 0 and 1, with 0 having all wrong predicitions, and 1 being a perfect classifier (at a specific test set). The Receiving Operating Characteristic is a curve that maps the True Positive Rate (TPR) into the False Positive Rate. The Area Under the Curve (AUC) represents the Area under this curve, and is a number between 0.5 and 1, such that 0.5 is a random classifier, and 1 is a perfect classifier (at a specific test set).

5. Experiments

5.1. Data processing

Before starting any network training, we need to guarantee a few steps in the data processing. One of those steps is related to our vision transformer models and variants, where normally the image size needs to be resized/zero-padded to the size of 384x384 pixels as required for the pre-trained implementation. We corrected the difference in the image size expected by the models using padding and cropping mechanisms. Task setting is another necessary step, it was to **frame classification** to act only on the single frame level as it is our objective.

5.2. Model Tuning

The benchmark created by [3] trained its own Vision Transformer model, with a learning rate of 0.2×10^{-5} and batch size 32. There were 5 networks trained for each model, up to a maximum of 100 epochs. However, they would be stopped prematurely when they started converging to a minimum loss. For each network, the epoch with the highest F1-score was chosen, and for each model, the average and standard deviation of the F1 and Area Under the Curve were computed.

In an attempt to compare new models to this baseline, we must run them with the same hyperparameters, to reach a fair conclusion. However, due to the computationally heavy nature of these models, we were unable to handle such a large batch size, and reduced it to 8.

To combat this, we chose to run a Vision Transformer Model of our own, with batch size = 8, in an attempt to understand how these newer models fare against the reference in [3]. No other fine-tuning was made, as we prefer to keep most hyperparameters constant.

Our goal now was simple - run all models, 5 times, to a max 100 epochs, with 0.2×10^{-5} learning rate, batch size = 8, and compare the results.

5.3. Results

All computations were ran on a laptop, equipped with a 13th Gen Intel Core i7-13700H CPU, 16GB of RAM, and a Nvidia GeForce RTX 4060 Laptop GPU. In the table below, we should be presenting the mean and standard deviaton of the F1-score and AUC of 5 different network instances.

However, some complications were found during training. First, models as such take a lot of time to run. Because of this, and due to the limited time we had to present results, not all transformers were possible to run, and not all 5 instances were performed for some. We did perform 5 network instances for the Vision Transformer, however, for both the Swin and DeiT, only 3 networks were trained, meaning the results will be less credible as a benchmark.

We also intended to run a fourth model, the Class-attention Image Transformer (CaiT), but were unable to run even one model, due to the massive run-time associated with the model. As a comparison, even with batch size = 2, each epoch would be estimated to train for 22 hours. This model was running with 270 million parameters, compared the the 5 to 7 million on the Vision Transformer.

Nevertheless, we present our results below. We should again note the 'Baseline' results were taken from [3], with a higher batch size, and all others were run as described.

	F1-Score	SD	AUC	SD
Baseline [1]	0.849	0.012	0.918	0.005
ViT	0.849	0.016	0.916	0.01
Swin	**0.864**	0.009	**0.934**	0.008
DeiT	0.853	0.01	0.929	0.007

Figure 4. Results Table - Baseline is the Vit trained in the [3], ViT stands for Vision Transformer, Swin stands for Swin Transformer and the DeiT stands for Data Efficient Image Transformer. The SD stands for standard deviation

As we can observe the results appear to outperform the baseline model [3], in which 2 models stand out, the **Data Efficient Image Transformer (DeiT) model** with a F1-Score of 0.853 and a Area Under the Curve (AUC) score of 0.929 and the **Swin Transformer (Swin) model with the best F1 and AUC scores presented of 0.864 and 0.934 respectively**.

5.4. Swin Model

The Swin model described in [2], is a model developed with the goal of serving as a general purpose backbone for computer vision. The work is based around building hierarchical feature maps by merging image patches in deeper layers, as such.

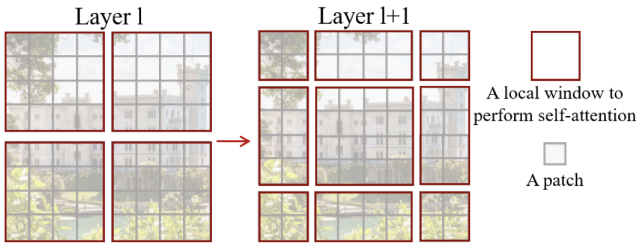

Figure 5. Shifted Windows - an example of two consecutive layers. Figure from [2]

We can also see the overall architecture of the Swin Model, presented below.

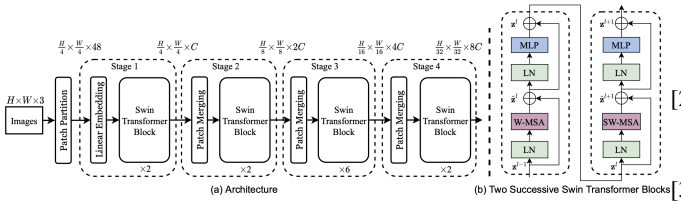

Figure 6. a) the overall architecture of the Swin Model; b) two consecutive Swin transformer blocks W-MSA and SW-MSA stand for regular and shifted windows Muti-Head Self Attention models. Figure from [2]

6. Conclusions

6.1. Conclusion

In this work, we measured state-of-the-art transformer architectures against previously achieved benchmarks that used other deep learning techniques, at the frame level.

The results in Figure 4 show that we transformer models seem, once again, overall succesful when compared to other deep learning models, despite our inability to run all we wished.

6.2. Limitations

There are still some limitations of previous work that weren't approached in our study, such as imperfections in the dataset. Given that all LUS were taken in the same hospital, by the same machines, there may be some biases. Also, only 2 experts annotated the clips, meaning our annotations might be biased.

Regarding the limitations of our own work, we were heavily halted by the lack of computing power (and time). As mentioned before, all tests were run on a Nvidia RTX 4060 Laptop GPU, and we were therefore unable to run as thorough of a test as the previous work, and we were also unable to fine-tune certain hyperparameters, such as the learning rate.

6.3. Further work

When comparing results to a certain benchmark, all results are positive. However, there is still some aspects to improve. Firstly, our models were run with batch size 8, while the benchmarks were run with 32, meaning our results aren't directly comparable. More networks could have also been run to improve the confidence in our results.

More work could have been done in attempting to benchmark at the clip, and pixel levels, for both classification and segmentation, using transformer models.

References

[1] John Gullett, John P. Donnelly, Richard Sinert, Bill Hosek, Drew Fuller, Hugh Hill, Isadore Feldman, Giorgio Galetto, Martin Auster, and Beatrice Hoffmann. Interobserver agreement in the evaluation of b-lines using bedside ultrasound. *Journal of Critical Care*, 30(6):1395–1399, 2015. 1

[2] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. 4

[3] Ruben T. Lucassen, Mohammad H. Jafari, Nicole M. Duggan, Nick Jowkar, Alireza Mehrtash, Chanel Fischetti, Denie Bernier, Kira Prentice, Erik P. Duhaime, Mike Jin, Purang Abolmaesumi, Friso G. Heslinga, Mitko Veta, Maria A. Duran-Mendicuti, Sarah Frisken, Paul B. Shyn, Alexandra J. Golby, Edward Boyer, William M. Wells, Andrew J. Goldsmith, and Tina Kapur. Deep learning for detection and localization of b-lines in lung ultrasound. *IEEE Journal of Biomedical and Health Informatics*, 27(9):4352–4361, 2023. 1, 3