Continually Adaptive Machine Learning Applied to a Healthcare Platform for the Detection of Tuberculosis

Simon E. Sanchez Viloria

Master in Computer Science and Technology

Departamento de Informática

Universidad Carlos III de Madrid

Advisors: Jesus Carretero PhD, Lara Visuña

September 27, 2023

Summary

Motivations:

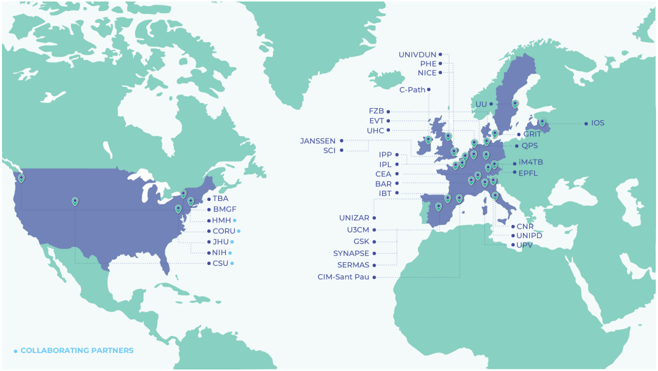
- Tuberculosis (TB) is an infectious disease that affects millions of people worldwide.
  - Is the leading cause of death for a single infectious disease.
  - Developing countries are the most affected by TB, where drug-resistant bacteria is a serious public health issue.
- Machine-learning and data-driven techniques can be used to support the development of computational methods to assist with TB research and detection.

Motivations:

- Tuberculosis (TB) is an infectious disease that affects millions of people worldwide.
  - Is the leading cause of death for a single infectious disease.
  - Developing countries are the most affected by TB, where drug-resistant bacteria is a serious public health issue.
- Machine-learning and data-driven techniques can be used to support the development of computational methods to assist with TB research and detection.

Context:

- The European Regimen Accelerator for Tuberculosis (ERA4TB).
- Over 31 Partners and Collaborators from Europe and the US.
- UC3M is a coordinating entity.



Background

- Deep Learning for TB Detection:
  - **Visuña et al. 2023** [1]: Developed a **computer-vision** model to localize TB bacilli from Sputum-smear microscopy.
- Continual Learning (CL) & Self-Adaptive Systems
  - Can we use continual learning, drawing from concepts of SAS, to improve these kinds of models?
  - Prioritize the acquisition of the most informative samples to reduce annotation costs.

[1] (First and last author were advisors)

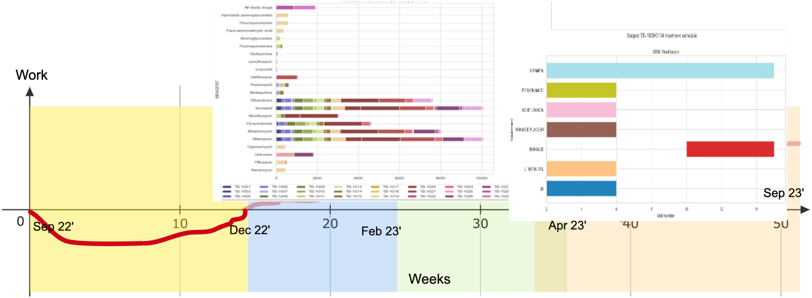
Background

- Deep Learning for TB Detection:
  - **Visuña et al. 2023** [1]: Developed a **computer-vision** model to localize TB bacilli from Sputum-smear microscopy.
- Continual Learning (CL) & Self-Adaptive Systems
  - Can we use continual learning, drawing from concepts of SAS, to improve these kinds of models?
  - Prioritize the acquisition of the most informative samples to reduce annotation costs.

[1] (First and last author were advisors)

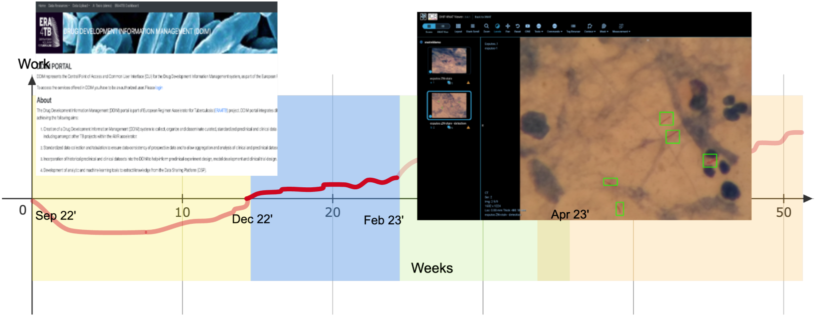
Timeline:

1. (Sep-Dec 22') Ups and downs, attempt to do a pharmacokinetics analysis



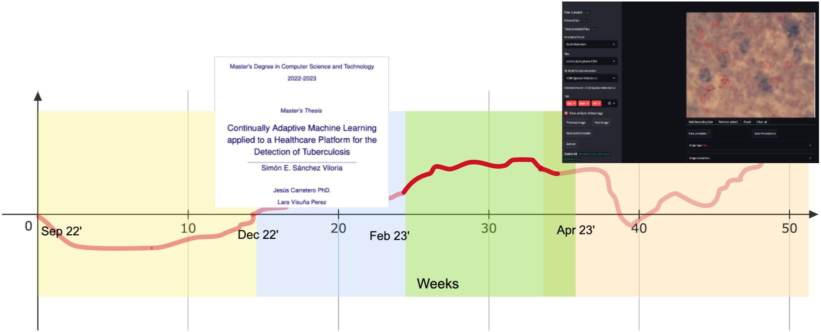
Timeline:

2. (Dec-Feb 23') Integrate TB-detection model to the ERA4TB platform (DDIM, XNAT, ...)



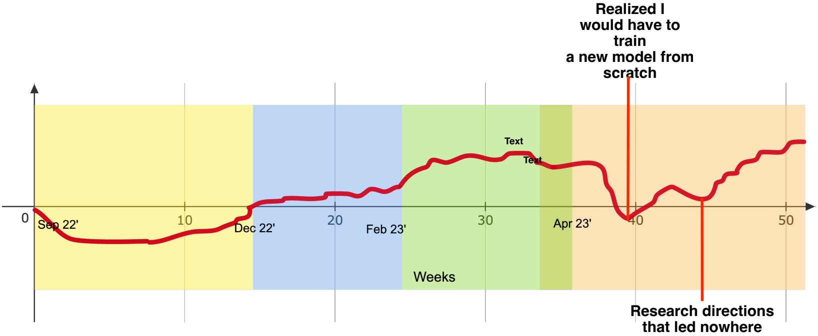
Timeline:

3. (Feb-Mar 23’) “We can use this as an opportunity to research CL techniques applied to CV and TB detection”



Timeline:

4 Most of the code was developed Mar 23'- Aug 23', thesis report was written from Apr 23' up to the deadline.



Summary:

1. Development of an interactive platform to visualize and annotate clinical images.
2. Trained and fine-tuned a transformer-based CV model to localize TB-bacilli from sputum-smear microscopy.
3. Implemented a system to use continual adaptation techniques to improve ML models.
   - Inspired by concepts of the Self-Adaptive Systems Literature.
   - Formalized each component in the design of the system.
4. Performed experiments using Active Learning to train incrementally using the most informative samples.

Summary:

1. Development of an interactive platform to visualize and annotate clinical images.
2. Trained and fine-tuned a transformer-based CV model to localize TB-bacilli from sputum-smear microscopy.
3. Implemented a system to use continual adaptation techniques to improve ML models.
   - Inspired by concepts of the Self-Adaptive Systems Literature.
   - Formalized each component in the design of the system.
4. Performed experiments using Active Learning to train incrementally using the most informative samples.

Summary:

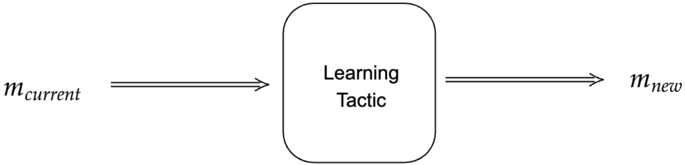
1. Development of an interactive platform to visualize and annotate clinical images.
2. Trained and fine-tuned a transformer-based CV model to localize TB-bacilli from sputum-smear microscopy.
3. Implemented a system to use continual adaptation techniques to improve ML models.
    - Inspired by concepts of the Self-Adaptive Systems Literature.
    - Formalized each component in the design of the system.
4. Performed experiments using Active Learning to train incrementally using the most informative samples.

Summary:

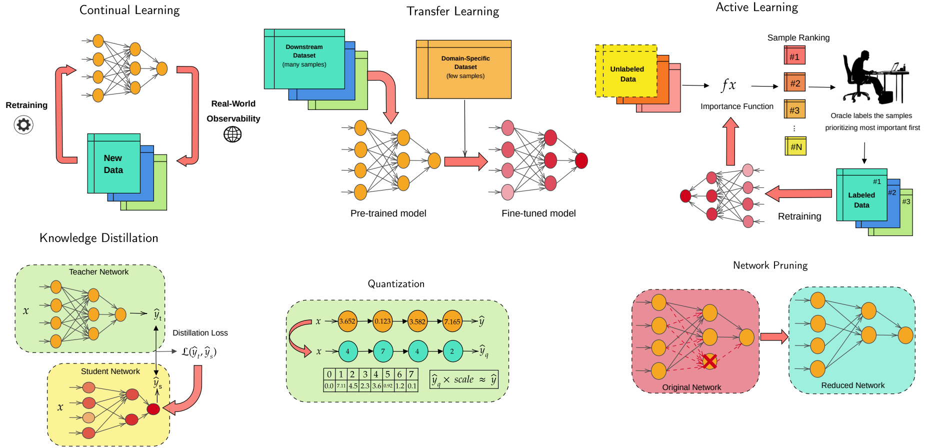
1. Development of an interactive platform to visualize and annotate clinical images.
2. Trained and fine-tuned a transformer-based CV model to localize TB-bacilli from sputum-smear microscopy.
3. Implemented a system to use continual adaptation techniques to improve ML models.
    - Inspired by concepts of the Self-Adaptive Systems Literature.
    - Formalized each component in the design of the system.
4. Performed experiments using Active Learning to train incrementally using the most informative samples.

"Continually Adaptive Machine Learning"

- Take an ML model/system and use learning techniques to optimize it for its respective purposes - observing the data it is fed to.
- Drawing from concepts of Continual Learning and Self-Adaptive Systems

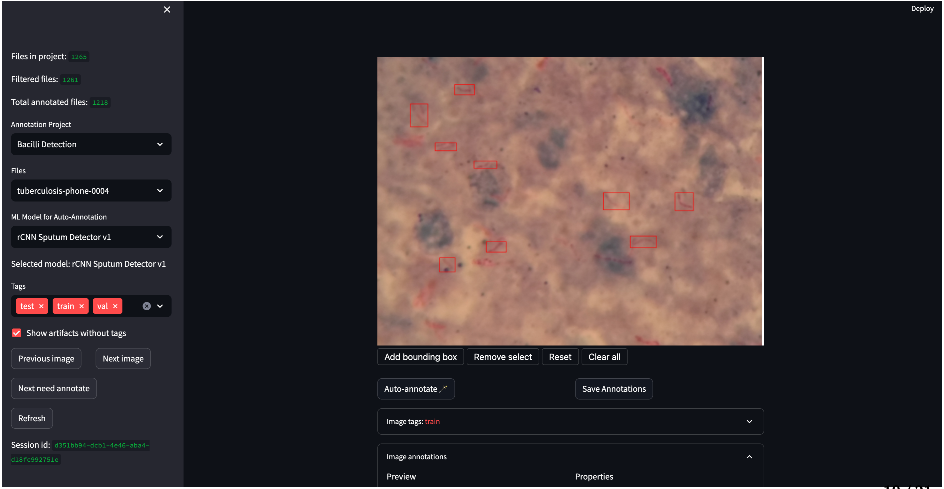


Examples of Continual Adaptation “Tactics”



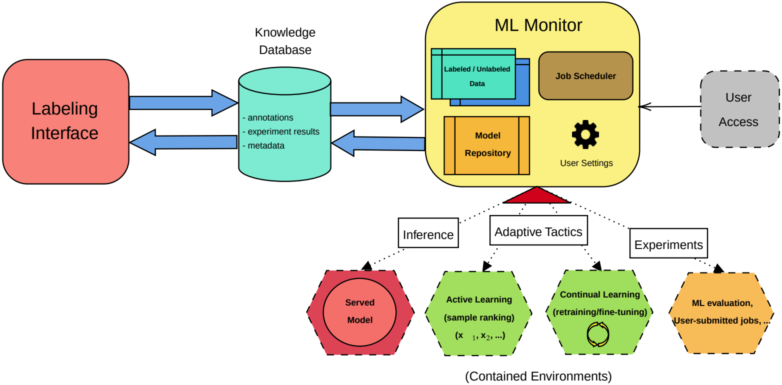
System Design

- Motivation: Implement a platform to annotate clinical images, reduce annotation efforts with AL, and improve CV models through CL



System Design

- Based on the idea of **MAPE-K** feedback loops (Monitor, Adapt, Plan, Execute, with a Knowledge system).



Continual Adaptation Tactics

- We can formalize all the continual adaptation tactics considered using the following algorithm:

Algorithm 1: AdaptationTactic(m, $\mathcal{K}'_m$)

Input: model $m \in \mathcal{M}$, data $\mathcal{K}'_m \subset \mathcal{K}_m$
Result: `success` if $m_{new} \in \mathcal{M}$, `failure` otherwise
Let $m_{new} = \texttt{Train}(m, \mathcal{K}_m)$
The platform should be able to complete the training process of m_{new}
for $B_i \in \{m \Rightarrow B_m\}$ **do**
  B_m *is the set of benchmark functions associated with* m
  if $B_i(m_{new}, \mathcal{K}'_m)$ *better than* $B_i(m, \mathcal{K}'_m)$ **then**
    The new model outperforms the old one
    $m \leftarrow \mathcal{M}$
    $m_{new} \rightarrow \mathcal{M}$
    return `success`
  end
end
return `failure`

Algorithm 2: ActiveLearning (R, m, $\mathcal{K}$)

Input: ranking function R, model $m \in \mathcal{M}$, knowledge database $\mathcal{K}$

Let $\mathcal{K}_{ranked} = \{x \in \mathcal{K} \mid (\text{'ranking'} \in (x \Rightarrow annotations \Rightarrow properties \Rightarrow property.name)) \wedge ((x \Rightarrow annotations \Rightarrow annotator) = m)\}$

Note $\mathcal{K}_{ranked}$ is the set of samples in $\mathcal{K}$ that have been ranked before for m

if $K_{ranked} = \emptyset$ **then**

  We rank the samples associated with m

  $\boldsymbol{\mathcal{K}_s} = \{x \in \mathcal{K}_m \mid \{a \in \{x \Rightarrow annotations\} \mid (a \Rightarrow annotator) \notin \mathcal{M}\} = \emptyset\}$

  Note $\mathcal{K}_s$ is the set of samples that are relevant to m and are not yet annotated by a human

  if $\mathcal{K}_s = \emptyset$ **then**

    There's no data to annotate

    exit

  end

  Rank the samples in $\mathcal{K}_s$ and start over

  `ImportanceSampling`(R, m, $\mathcal{K}_s$)

  return `ActiveLearning`(R, m, $\mathcal{K}$)

end

$\mathcal{K}^+_{ranked} = \emptyset$

for $(x_i, \ldots x_n) \in \mathcal{K}_{ranked}$ **do**

  Note we assume the samples in $\mathcal{K}'_{ranked}$ are sorted by their 'ranking' value

  annotate x_i

  $\mathcal{K}^+_{ranked} = \mathcal{K}^+_{ranked} \cup \{x_i\}$

  if *ContinualLearning*(m, $\mathcal{K}^+_{ranked}$) $=$ *success* **then**

    start over

    return `ActiveLearning`(R, m, $\mathcal{K}$)

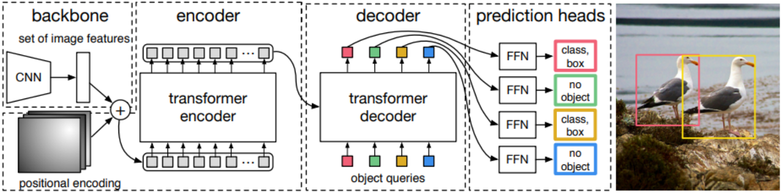
  end

end

Testing the algorithms with a model to detect TB

DETR: Detection Transformer (Carion et al. 2020)

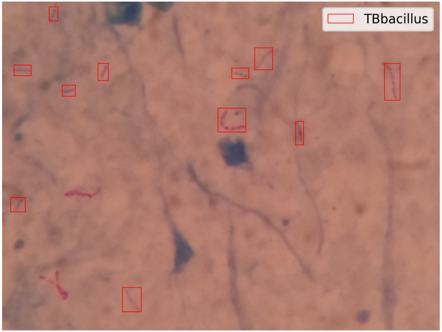
- Object detection model based on transformers (Vaswani et al. 2017)
- End-to-end detection using a CNN backbone and encoder-decoder attention layers
- Outputs "queries" (predictions) of the bounding boxes with the object detected and their corresponding class probabilities.
- Fine-tuned it with sputum-smear microscopy images to detect TB bacilli.
- $\approx$ 45 minutes training time w/ a T4 GPU (Colab).



Testing the algorithms with a model to detect TB

Tuberculosis Image Dataset

- Dataset consisted of 308 manually selected Gram-stained sputum-smear microscopy images of TB bacilli.
- Trained with 202 images and 40 images as validation.
- A subset of 60 images was assumed to be unseen to test performance



Active Learning Approach:

- Use the margin of the class probabilities as a metric of how "uncertain" the model is about a (image) sample.

Margin of Confidence (MoC)

$$\text{MoC}(p_{N\times 2}) = \frac{1}{N}\sum_{i=1}^{N} |p_{i,1} - p_{i,2}|$$

Where we let $p_{N\times 2}$ be a vector with the class scores of each of the **N** bounding boxes predicted by DETR, with $p_{i,1}$ being the probability of the background class, and $p_{i,2}$ the probability of bacilli.

- A base DETR model is trained with a random 50% of the training images while the other 50% is used as a "holdout set" (yet to be annotated).
- We set an incremental training schedule with steps of 25%, 50%, and 80% of the holdout data.
- A new model is trained from the base model using the data at each step of the schedule.
- Test with the AL approach and randomly selected samples.

Active Learning Approach:

- Use the margin of the class probabilities as a metric of how "uncertain" the model is about a (image) sample.

Margin of Confidence (MoC)

$$\text{MoC}(p_{N\times 2}) = \frac{1}{N}\sum_{i=1}^{N} |p_{i,1} - p_{i,2}|$$

Where we let $p_{N\times 2}$ be a vector with the class scores of each of the **N** bounding boxes predicted by DETR, with $p_{i,1}$ being the probability of the background class, and $p_{i,2}$ the probability of bacilli.

- A base DETR model is trained with a random 50% of the training images while the other 50% is used as a "holdout set" (yet to be annotated).
- We set an incremental training schedule with steps of 25%, 50%, and 80% of the holdout data.
- A new model is trained from the base model using the data at each step of the schedule.
- Test with the AL approach and randomly selected samples.

Active Learning Approach:

- Use the margin of the class probabilities as a metric of how "uncertain" the model is about a (image) sample.

Margin of Confidence (MoC)

$$\text{MoC}(p_{N\times 2}) = \frac{1}{N}\sum_{i=1}^{N} |p_{i,1} - p_{i,2}|$$

Where we let $p_{N\times 2}$ be a vector with the class scores of each of the **N** bounding boxes predicted by DETR, with $p_{i,1}$ being the probability of the background class, and $p_{i,2}$ the probability of bacilli.

- A base DETR model is trained with a random 50% of the training images while the other 50% is used as a "holdout set" (yet to be annotated).
- We set an incremental training schedule with steps of 25%, 50%, and 80% of the holdout data.
- A new model is trained from the base model using the data at each step of the schedule.
- Test with the AL approach and randomly selected samples.

Active Learning Approach:

- Use the margin of the class probabilities as a metric of how "uncertain" the model is about a (image) sample.

Margin of Confidence (MoC)

$$\mathrm{MoC}(p_{N\times 2}) = \frac{1}{N}\sum_{i=1}^{N} |p_{i,1} - p_{i,2}|$$

Where we let $p_{N\times 2}$ be a vector with the class scores of each of the **N** bounding boxes predicted by DETR, with $p_{i,1}$ being the probability of the background class, and $p_{i,2}$ the probability of bacilli.

- A base DETR model is trained with a random 50% of the training images while the other 50% is used as a "holdout set" (yet to be annotated).
- We set an incremental training schedule with steps of 25%, 50%, and 80% of the holdout data.
- A new model is trained from the base model using the data at each step of the schedule.
- Test with the AL approach and randomly selected samples.

Active Learning Approach:

- Use the margin of the class probabilities as a metric of how "uncertain" the model is about a (image) sample.

Margin of Confidence (MoC)

$$\text{MoC}(p_{N\times 2}) = \frac{1}{N}\sum_{i=1}^{N} |p_{i,1} - p_{i,2}|$$

Where we let $p_{N\times 2}$ be a vector with the class scores of each of the **N** bounding boxes predicted by DETR, with $p_{i,1}$ being the probability of the background class, and $p_{i,2}$ the probability of bacilli.

- A base DETR model is trained with a random 50% of the training images while the other 50% is used as a "holdout set" (yet to be annotated).
- We set an incremental training schedule with steps of 25%, 50%, and 80% of the holdout data.
- A new model is trained from the base model using the data at each step of the schedule.
- Test with the AL approach and randomly selected samples.

Active Learning Approach:

- Use the margin of the class probabilities as a metric of how "uncertain" the model is about a (image) sample.

Margin of Confidence (MoC)

$$\text{MoC}(p_{N\times 2}) = \frac{1}{N}\sum_{i=1}^{N} |p_{i,1} - p_{i,2}|$$

Where we let $p_{N\times 2}$ be a vector with the class scores of each of the **N** bounding boxes predicted by DETR, with $p_{i,1}$ being the probability of the background class, and $p_{i,2}$ the probability of bacilli.

- A base DETR model is trained with a random 50% of the training images while the other 50% is used as a "holdout set" (yet to be annotated).
- We set an incremental training schedule with steps of 25%, 50%, and 80% of the holdout data.
- A new model is trained from the base model using the data at each step of the schedule.
- Test with the AL approach and randomly selected samples.

Benchmark Metrics

- IoU = Intersection over Union
- Average Precision at IoU of 0.5 and 0.3 (AP@.50, AP@.30).
- Average Recall at IoU of 0.5 and 0.3 (AR@.50, AR@.30).
- Baseline Models.
- 0.3 thresholds to prioritize positive detections.

Results

- We include the performance of the DETR model trained with **half** of the training data, which we will refer to as the 'base' model - a reference to observe if a model can improve by applying the tactics proposed.
- We also include the performance of our model with the Mobile NasNet developed by Visuña et al. 2023.

Table 1: Baseline results

Model	AP@50	**AP@30**	AR@50	AR@30	Avg IoU
DETR (Full)	0.824	**0.874**	0.731	0.776	0.579
Mobile NasNet	0.451	0.801	0.450	0.781	0.470
DETR (50%)	0.731	0.798	0.687	0.735	0.543

- Improved average precision/recall at IoU of 0.50 by over 37%.
- Compared with MobileNasNet, our DETR model achieved 7x faster inference speed (0.2s/image compared to 1.2s/image)

Results

- We include the performance of the DETR model trained with **half** of the training data, which we will refer to as the 'base' model - a reference to observe if a model can improve by applying the tactics proposed.
- We also include the performance of our model with the Mobile NasNet developed by Visuña et al. 2023.

Table 1: Baseline results

Model	AP@50	**AP@30**	AR@50	AR@30	Avg IoU
DETR (Full)	0.824	**0.874**	0.731	0.776	0.579
Mobile NasNet	0.451	0.801	0.450	0.781	0.470
DETR (50%)	0.731	0.798	0.687	0.735	0.543

- Improved average precision/recall at IoU of 0.50 by over 37%.
- Compared with MobileNasNet, our DETR model achieved 7x faster inference speed (0.2s/image compared to 1.2s/image)

Visual Results

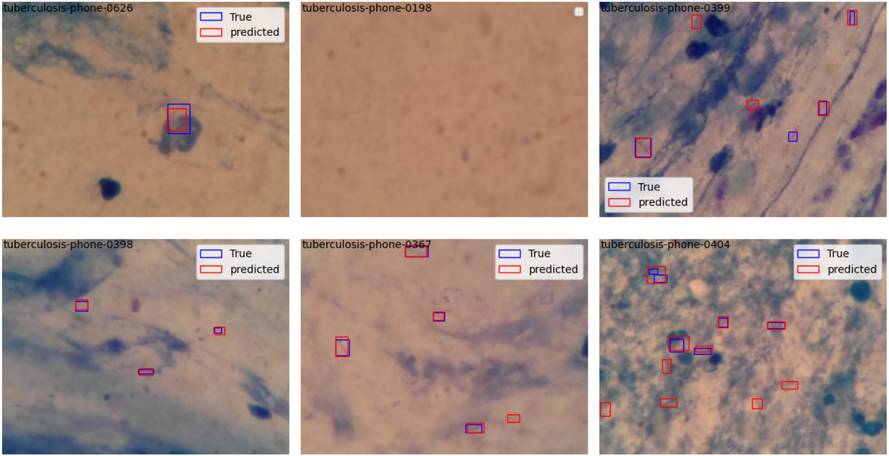


Table 2: Evaluation metrics for the trained models. Ranked by AP@30

Experiment	AP@50	AP@30	AR@50	AR@30	Avg IoU
DETR (Full)	**0.824**	**0.874**	**0.731**	**0.776**	**0.579**
RS - (Retr) step:1+2+3	0.766	0.873	0.690	0.774	0.549
AL - (Retr) step:1	0.794	0.869	0.574	0.617	0.438
AL - step:1	0.806	0.869	0.695	0.744	0.552
RS - step:3	0.803	0.865	0.541	0.573	0.458
AL - step:2	0.713	0.847	0.573	0.696	0.468
RS - step:4	0.761	0.842	0.704	0.773	0.540
AL - step:3	0.729	0.840	0.575	0.656	0.465
AL - (Retr) step:1+2+3	0.802	0.831	0.557	0.576	0.461
AL - step:4	0.718	0.824	0.691	0.788	0.540
RS - step:1	0.727	0.818	0.703	0.784	0.548
RS - (Retr) step:1	0.713	0.807	0.728	0.813	0.530
DETR (50%)	**0.731**	**0.798**	**0.687**	**0.735**	**0.543**
RS - step:2	0.595	0.641	0.239	0.260	0.222

We remark that:

- The Active-Learning model trained incrementally with 25 samples achieved a similar accuracy than a model trained with **7x** more data
- The models that use an incremental learning approach with AL exhibit signs of **catastrophic forgetting**

We conclude that:

Conclusion

- Through our work, we assess how adopting a data-driven continual adaptation framework can improve the performance of learning systems for TB detection.
- The experimental results demonstrate the effectiveness of these tactics and the value they can have in the healthcare domain by the potential of reducing annotation costs significantly.
- The proposed system was designed in a way to be easy to extend and adapt to other tasks and problems and become a valuable part of the ERA4TB platform

For future work, we propose:

Future Work

- Further research/implementation to overcome problems in the continual learning process (catastrophic forgetting, loss of plasticity, etc.)
- Implement other continual adaptation tactics: Unlearning, KD, pruning.
- Federated Learning, other research directions...

We conclude that:

Conclusion

- **Through our work, we assess how adopting a data-driven continual adaptation framework can improve the performance of learning systems for TB detection.**
- **The experimental results demonstrate the effectiveness of these tactics and the value they can have in the healthcare domain by the potential of reducing annotation costs significantly.**
- **The proposed system was designed in a way to be easy to extend and adapt to other tasks and problems and become a valuable part of the ERA4TB platform**

For future work, we propose:

Future Work

- Further research/implementation to overcome problems in the continual learning process (catastrophic forgetting, loss of plasticity, etc.)
- Implement other continual adaptation tactics: Unlearning, KD, pruning.
- Federated Learning, other research directions...

We conclude that:

Conclusion

- **Through our work, we assess how adopting a data-driven continual adaptation framework can improve the performance of learning systems for TB detection.**
- **The experimental results demonstrate the effectiveness of these tactics and the value they can have in the healthcare domain by the potential of reducing annotation costs significantly.**
- **The proposed system was designed in a way to be easy to extend and adapt to other tasks and problems and become a valuable part of the ERA4TB platform**

For future work, we propose:

Future Work

- Further research/implementation to overcome problems in the continual learning process (catastrophic forgetting, loss of plasticity, etc.)
- Implement other continual adaptation tactics: Unlearning, KD, pruning.
- Federated Learning, other research directions...

We conclude that:

Conclusion

- **Through our work, we assess how adopting a data-driven continual adaptation framework can improve the performance of learning systems for TB detection.**
- **The experimental results demonstrate the effectiveness of these tactics and the value they can have in the healthcare domain by the potential of reducing annotation costs significantly.**
- **The proposed system was designed in a way to be easy to extend and adapt to other tasks and problems and become a valuable part of the ERA4TB platform**

For future work, we propose:

Future Work

- Further research/implementation to overcome problems in the continual learning process (catastrophic forgetting, loss of plasticity, etc.)
- Implement other continual adaptation tactics: Unlearning, KD, pruning.
- Federated Learning, other research directions...

We conclude that:

Conclusion

- **Through our work, we assess how adopting a data-driven continual adaptation framework can improve the performance of learning systems for TB detection.**
- **The experimental results demonstrate the effectiveness of these tactics and the value they can have in the healthcare domain by the potential of reducing annotation costs significantly.**
- **The proposed system was designed in a way to be easy to extend and adapt to other tasks and problems and become a valuable part of the ERA4TB platform**

For future work, we propose:

Future Work

- Further research/implementation to overcome problems in the continual learning process (catastrophic forgetting, loss of plasticity, etc.)
- Implement other continual adaptation tactics: Unlearning, KD, pruning.
- Federated Learning, other research directions...

We conclude that:

Conclusion

- **Through our work, we assess how adopting a data-driven continual adaptation framework can improve the performance of learning systems for TB detection.**
- **The experimental results demonstrate the effectiveness of these tactics and the value they can have in the healthcare domain by the potential of reducing annotation costs significantly.**
- **The proposed system was designed in a way to be easy to extend and adapt to other tasks and problems and become a valuable part of the ERA4TB platform**

For future work, we propose:

Future Work

- Further research/implementation to overcome problems in the continual learning process (catastrophic forgetting, loss of plasticity, etc.)
- Implement other continual adaptation tactics: Unlearning, KD, pruning.
- Federated Learning, other research directions...

We conclude that:

Conclusion

- **Through our work, we assess how adopting a data-driven continual adaptation framework can improve the performance of learning systems for TB detection.**
- **The experimental results demonstrate the effectiveness of these tactics and the value they can have in the healthcare domain by the potential of reducing annotation costs significantly.**
- **The proposed system was designed in a way to be easy to extend and adapt to other tasks and problems and become a valuable part of the ERA4TB platform**

For future work, we propose:

Future Work

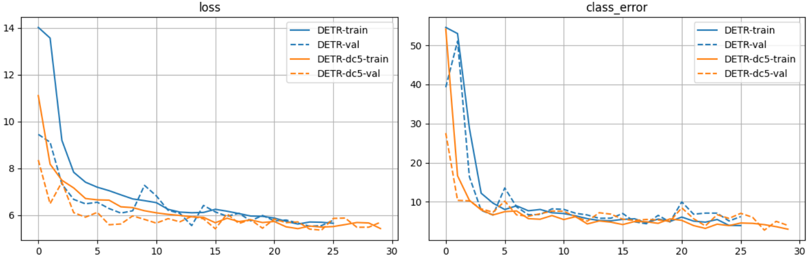
- Further research/implementation to overcome problems in the continual learning process (catastrophic forgetting, loss of plasticity, etc.)
- Implement other continual adaptation tactics: Unlearning, KD, pruning.
- Federated Learning, other research directions...

References

- Carion, Nicolas et al. (May 2020). *End-to-End Object Detection with Transformers*. DOI: 10.48550/arXiv.2005.12872. arXiv: 2005.12872 [cs].
- Vaswani, Ashish et al. (June 2017). *Attention Is All You Need*. Comment: 15 pages, 5 figures. DOI: 10.48550/arXiv.1706.03762. arXiv: 1706.03762 [cs].
- Visuña, Lara, Javier Garcia-Blas, and Jesus Carretero (2023). "Novel Deep Learning-Based Technique for Tuberculosis Bacilli Detection in Sputum Microscopy". In: *International Conference on Interactive Collaborative Robotics*. Springer, pp. 269–279.

Appendix

Plot of the loss and class error of DETR during training



- DETR-DC5 is the model that includes a dilated convolution layer to increase the receptive field of the model - which theoretically makes it easier to detect smaller objects but increases memory costs 2x.
- Even though DC5 reached a lower loss error faster, it eventually converged to the same value as the regular model.

Appendix

Why not just use the MobileNasNet model developed by Visuña et al. 2023?

- Fragmentation technique requires passing around 1200 image fragments of 80x80 to a fine-tuned CNN model → Very high inference/training costs
- After attempts to reduce computational costs, I decided to move forward with training DETR