Sharing is Caring: Concurrent Interactive Segmentation and Model Training using a Joint Model

Ivan Mikhailov[1,2], Benoit Chauveau[2,3], Nicolas Bourdel[1,2,3], and Adrien Bartoli[1,2,3]

[1]Université Clermont Auvergne [2]SURGAR, Clermont-Ferrand, France [3]CHU de Clermont-Ferrand

Abstract

The performance of neural predictors depends on the size and composition of the training dataset. However, annotating data is expensive. Efficient annotation systems usually feature a neural annotation predictor whose result can be edited by the expert using classical tools. Existing systems train the annotation predictor from an initial small subset of data annotated by classical tools and then freeze it for the rest of the annotation process. This is suboptimal as the annotation predictor does not benefit from the new annotations as the annotation process progresses. We propose a framework called Single Active Interactive Model (SAIM), which integrates the three steps of data selection, annotation and training into a single architecture. This is made possible by three key properties of SAIM in contrast with existing work: 1) SAIM uses a deep interactive predictor; hence the classical tools are not required and the annotation predictor can be pre-trained with limited data to produce quality annotations; 2) SAIM uses a single model shared between the three steps, hence the model is deployable and the annotation predictor improves as annotation progresses; 3) SAIM uses active learning to maximise the impact of each annotation on the predictor performance, making the model rapidly improve. We evaluated SAIM by emulating annotation scenarios on fully-labelled segmentation datasets. For a complex female pelvis MRI dataset, pre-training SAIM on 15% of data and annotating the whole dataset achieves 73.4% IoU with 6.3 hours of annotation time, against 75.8% IoU for complete manual annotation, requiring 40.0 hours. We also applied SAIM to a real-world case of very large MRI dataset (AMOS) segmentation, which cannot be feasibly annotated otherwise.

1. Introduction

Machine learning has gained widespread applicability in recent years, achieving good performance in various fields [1]. Still, the performance of supervised machine learning inherently depends on the size and composition of the annotated training dataset. However, data annotation is expensive, which is especially true in the domain of medical imaging [7]. The research community has made considerable efforts to alleviate the annotation problem and proposed a wide range of approaches. They include but are not limited to semi-, weakly-, self- and unsupervised learning, zero- and one-shot learning, transfer and multi-task learning, and self-training [35, 30, 28, 2], which limit or remove the need for human annotation. However, their applicability is limited: they often require more complex algorithms and fine-tuning, and may result in less accurate and interpretable models than with supervised learning, which generally offers a simpler way of achieving higher performance when sufficient annotated data is available [1]. Thus, finding a solution to expedit the annotation process remains pertinent.

The base supervised machine learning paradigm has three main steps: A - data annotation, B - target predictor training, C - evaluation. The system then may loop back to step A. In the simplest systems, step A may use classical tools such as intelligent scissors [25], which require a lot of effort and time. In more advanced systems, this may be improved using sample selection by active learning. However, the state of the art shows that replacing the classical tools with a suitable neural annotation predictor boosts annotation performance [33]. The annotation predictor suggests an annotation that the expert can validate or correct. This raises the question of training this annotation predictor, which existing systems do once a sufficient amount of data has been annotated by classical tools. This is suboptimal, as neither the annotation predictor nor the classical tools improve as more data is annotated. The main challenge is thus to exploit the data as they are annotated towards training the target predictor to improve the annotation mechanism itself, including the annotation predictor, which is yet an unresolved problem [6].

We propose a general framework called Single Active Interactive Model (SAIM), which addresses this problem

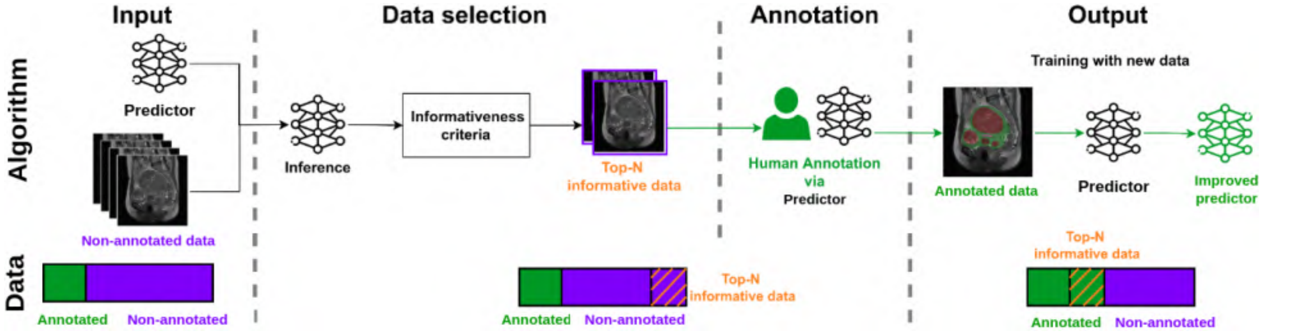

Figure 1. Detailed SAIM schematic: a single iteration is shown, only the input changes between iterations.

by integrating the three steps of data selection, annotation and training into a single architecture. This is made possible by three key properties of SAIM, which contrast with existing work. 1) deep interactive predictor - the annotation mechanism is based on an interactive neural predictor; hence the predictor can be pre-trained with limited data and still produce quality annotations thanks to the user input. 2) model-sharing - SAIM uses a single model shared between the three steps (A,B and C); the roles of the target predictor and the annotation predictor are thus performed by a single predictor. 3) active data selection - active learning is used to maximise the impact of each annotation on the predictor performance, exploiting the current predictor to optimally select data, making the model rapidly improve. To realise SAIM, two key components are required: an interactive neural predictor, which suggests annotations and enables interactive corrections, and a limited quantity of annotated data used for pre-training and testing. SAIM works in three steps. First, the predictor pre-trained on very limited initial annotations is used for data selection from the non-annotated data pool via active learning. Second, the expert uses the predictor to annotate the selected data. Third, the annotations are added to the training data for predictor re-training. The system then loops back to the first step and continues until stopped or all available data are annotated. As a result, SAIM allows one to efficiently annotate massive datasets from very limited initial annotations, while keeping the predictor up-to-date and deployable.

Our main contribution is SAIM, which is the first general machine learning framework to integrate data selection, annotation and training into a single architecture by model-sharing. Two key proposed ideas required to realise such an integrated framework are the use of deep interactive annotation with the shared model and the use of active learning for efficient data selection using the shared model.

We evaluate SAIM and compare it to existing systems in emulated annotation scenarios in an automated manner with fully-annotated segmentation datasets, with here three datasets: a female pelvis MRI dataset and public liver and pancreas CT datasets [4]. We demonstrate SAIM in a real annotation scenario of kidney MRI segmentation from the AMOS dataset [16] with a human user and a 1 to 30 annotated to non-annotated data ratio. We estimate the time gain as compared to 3D Slicer, where SAIM allowed to double the total number of AMOS kidney MRI annotations in 2.3 hours against 10.0 hours for 3D Slicer using classical tools. SAIM jumpstarts efficient interactive annotation from limited annotated data and minimises the amount of data to annotate, while iteratively improving performance.

2. Related Work

There is no established classification of data annotation approaches in the literature [15, 18, 31, 5]. We propose to describe these approaches based on how they modify step A of the base machine learning paradigm. This leads to two broad groups of approaches we call static and dynamic, depending on whether the data annotation mechanism is kept fixed or is improved as annotation progresses. Step B is typically achieved by involving an expert or by continual learning [13], but the method of training is not in the scope of our contributions.

The data annotation mechanism may use classical tools, neural predictors, or a combination of both. The classical tools are non-neural and non-trainable, hence not specific to a single task or domain, allowing for a wide applicability. They may strongly vary in functionality, complexity and target domain. In image annotation, we find the intelligent scissors [25], GrabCut [27] and Random Walker [12]. The neural predictors are pre-trained and are generally specific to a task and a domain. They can be fixed or retrained through the data annotation process. This implies two general statements: 1) an approach which uses exclusively classical tools is necessarily a static approach, and 2) a dynamic approach necessarily uses a neural predictor with re-training. Fixed neural predictors require huge initial training datasets such as MS COCO [21], as shown in a survey of over 100 segmentation predictors [24]. Dynamic predictors are normally used in self-training [2] in a

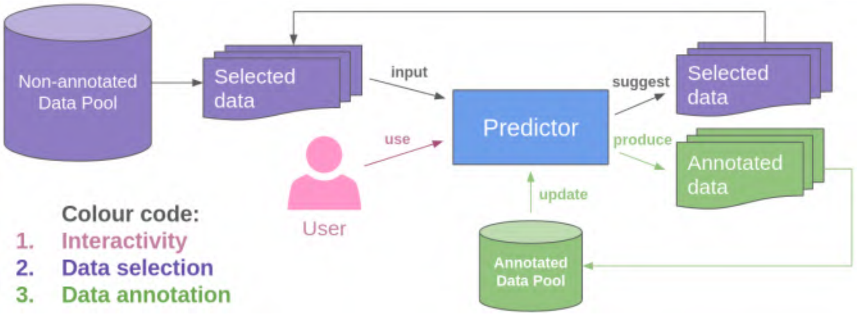

Figure 2. A functional schematic of SAIM with respect to the shared predictor.

semi-supervised manner, where predicted pseudo-labels are added to the dataset when re-training. They use classical tools to be trained in a supervised manner.

Most of the existing systems such as Synapse 3D [10], 3D Slicer [17], MITK [11] and Supervise.ly [29] implement static approaches. They provide access freely or commercially to a large variety of classical tools and fixed neural predictors. Any approach may be complemented by data selection via active learning [6, 26]. The idea is to select the most informative data to annotate. Existing work performs active learning independently of the annotation mechanism. For example, [22, 34] are static approaches based on classical tools. They use active learning as a data selection policy in reinforcement learning [22] or train a neural classifier to perform data selection [34]. The two closest works to ours are the non-medical object detection annotation system [32] and the MONAI Label toolbox [9]. System [32] is a dynamic system. Images selected based on the Euclidean distance are segmented by a pre-trained neural predictor. Annotation corrections are then done by classical tools. The neural predictor is periodically re-trained from the corrected annotations. The MONAI Label toolbox [9] is a static system. It combines classical tools provided by 3D Slicer and two fixed neural predictors, an automatic one and an interactive one. In contrast, SAIM is a dynamic system, which [9] is not, uses a dynamic interactive neural predictor, which [32] does not, and uses the predictor to perform data selection, which neither of [32, 9] do. This unique combination is the key to enable model-sharing, where the single predictor is used for data selection, interactive data annotation and undergoes dynamic retraining, which is not featured in any existing approach and system.

3. Methodology

3.1. System Overview

On a general level, we build SAIM as shown in the row "Algorithm" of figure 1. As inputs, SAIM requires a minimally pre-trained interactive predictor, a non-annotated data pool, a set of criteria to perform data selection and the number N of samples selected for annotation at each iteration, which is determined as a function of the dataset size and the annotation capacity. We originally developed SAIM for 3D image segmentation but the framework is generic and not restricted to a specific task, domain, modality or predictor architecture. Indeed, the interactive neural predictor can be obtained from any trainable machine learning architecture, as long as at inference time the network architecture takes both the non-annotated data such as an image and the user corrections as inputs and outputs an annotation, such as in [3, 19, 23]. Such architectures are generally reusable for different tasks. Specifically, we reuse the neural interactive system [23], which inputs user clicks and outputs a segmentation mask for each class. Neural interactivity is a key feature of SAIM, which allows it to quickly produce better annotations through interactive refinement as opposed to static approaches. SAIM proceeds iteratively with three inner steps. At each iteration, SAIM inputs a predictor and outputs an updated predictor and new annotated data. Model-sharing is implemented by having a single predictor shared between the steps as shown in figure 2. First, we perform data selection from the non-annotated pool based on the se-

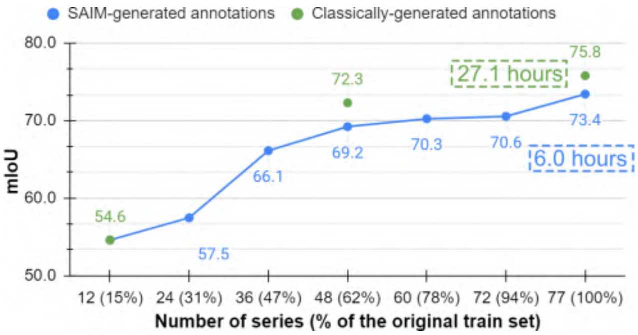

Figure 3. SAIM experimental evaluation results given as mIoU at each iteration on our female pelvis MRI dataset, where: green - performance using annotations created in 3D Slicer and MITK, blue - performance using SAIM-created annotations. Annotation time is reported excluding pre-training data annotation with classical tools.

lection criteria and the predictor. Second, we perform interactive annotation of the selected data using the predictor within the interactive neural annotation mechanism. Third, we update the predictor by re-training with the complete annotated data pool. Iteration $n+1$ therefore improves on iteration n by two factors: the quantity of annotated data increases owing to data selection and data annotation done in iteration n; the predictor at iteration $n+1$ is thus improved by benefiting from re-training from the increased quantity of data compared to iteration n. The iterations continue until SAIM is stopped or all available data are annotated. Thus, at each iteration SAIM uses a shared predictor to select data, annotate the data interactively and use the data in addition to the current annotated training set for re-training the predictor.

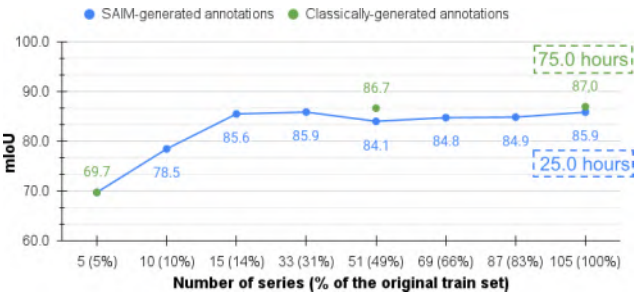

Figure 4. SAIM experimental evaluation results given as mIoU at each iteration on Liver CT dataset, where: green - performance using annotation produced using classical tools, blue - performance using SAIM-created annotations. Annotation time is estimated excluding pre-training data annotation with classical tools.

3.2. Data Selection

SAIM starts with an interactive predictor pre-trained on a generally small quantity of annotated data and is immediately ready to produce new annotations. However, during the first iterations, extra user interactions may be required to correct the output of such a predictor, resulting in prolonged annotation time. It is thus crucial to speed up the annotation process and to ensure the performance of the predictor is improved rapidly going forward. We address this point by maximising the impact of each individual annotation. Specifically, we perform data selection as the first step in the SAIM framework. This is done via active learning, which involves selecting the most informative data from a pool of non-annotated data and then requesting annotations for this data from a human expert. By doing so, the predictor can achieve high accuracy with fewer annotated examples required for training.

Data selection in active learning is often performed based on informativeness. Informativeness refers to the degree of usefulness of the selected data in improving the performance and generalisation of a predictor if added to the training set. Data informativeness is evaluated by informativeness criteria, which can all be exploited by SAIM. The informativeness criteria may be external or internal. External criteria involve additional information such as image meta-data reflexive of clinical characteristics. Internal criteria are based on the image data itself and may or may not involve the predictor. Internal criteria involving the predictor are generally based on uncertainty or representativeness [6].

We implemented SAIM with a classical entropy informativeness criterion, which is internal, being dependent on the predictor output, allowing us to directly select the images benefiting predictor improvement for subsequent annotation. For the image segmentation case, the entropy is calculated for each pixel from the predicted class probability distribution and then averaged over all pixels. A higher entropy is thus obtained for images with target classes having closer probabilities pixel-wise. Simply put, images with ambiguous predictions overall are considered more informative and are selected for subsequent annotation [6, 26]. We perform this selection in three steps: (1) the predictor outputs probability maps for all samples in the non-annotated data pool via inference; (2) the entropy of each sample is evaluated, resulting in a score; (3) the top N scoring samples are selected for annotation, while the rest of the data remains in the non-annotated pool. The schematic of the data selection step is shown in figure 1.

3.3. Data Annotation

Once data selection is performed, the selected samples are passed to the human user for annotation using the shared interactive predictor. Indeed, SAIM requires an interactive predictor, which suggests an annotation to the user, and is capable of accepting the subsequent user corrections. Along with data selection, interactivity is a key property of the SAIM architecture, which ensures that annotations of sufficient quality are produced even if the interactive predictor is initially pre-trained on a limited amount of data, and that the annotation mechanism fully benefits from the on-going annotation process. The interaction allows the predictor to achieve a far better annotation quality than that of an equivalent automatic system.

SAIM does not depend on a specific architecture of the interactive predictor, which can be any trainable machine learning architecture, as long as it accepts user corrections and outputs the annotation. To demonstrate SAIM, we use the interactive system [23], which focuses solely on interactive image segmentation. This system consists of an embedded network, a user interaction loop and an interaction memory. It inputs an image, user interaction masks and optionally a segmentation mask, if available from previous steps, for each class. It outputs the segmentation probability maps. User interaction masks contain user clicks indicating foreground and background for each class. This system comes with a specific training approach, where the user corrections at training time are automatically generated on-the-fly from the annotated dataset by means of a virtual user simulating interactions. At test time, this system is used by the human user via a general-purpose GUI as in classical tools. In our experiments, we use the system with a human user to evaluate its impact in a real annotation scenario. We also reintroduce the virtual user at test time to conduct an extensive statistical evaluation in emulated annotation scenarios where the complete dataset annotation is already available.

3.4. Predictor Update

Once the data is annotated, it is used to extend the current training set, after which the predictor is updated, as shown in figure 1. The update can be a full re-training or simple fine-tuning, which can be done by means of continual learning or by an expert engineer. While re-training may be slower in comparison to other predictor update methods, such as fine-tuning, it allows for a clearer comparison between iterations, characterised by using different quantities of data. Specifically, fine-tuning involves modifying the weights of an existing model to fit a new dataset or task. This modification can result in some already learned information being lost or changed, but it is often difficult to pinpoint exactly what information has been affected. This becomes especially important when considering what is learned by the model after a large number of consecutive fine-tunings, as it could be in case of SAIM. We thus use re-training, which considers the complete annotated dataset at each iteration. This allows us to directly estimate the effect of each new annotation set on the predictor performance.

4. Experimental Results

4.1. Instantiation of SAIM and General Setup

We instantiate our system with an interactive predictor using ResNet34 [14] encoder pre-trained on ImageNet [8] as in [23]. To counter dataset imbalance, we use the focal loss [20] with per-class weights, which are re-calculated prior to re-training at each iteration. We preprocess all data via normalisation, standardisation, and perform random data augmentation: vertical and horizontal flipping, intensity shifting, gamma correction, blurring and unsharp masking.

4.2. Emulated Annotation Scenarios

General considerations. We perform a systematic evaluation of SAIM's performance in three emulated annotation scenarios: in female pelvis MRI segmentation on our dataset, in Liver and in Pancreas CT segmentation on decathlon datasets [4]. Two factors make these scenarios emulated: first, these datasets were fully annotated with high reliability using classical tools; second, human user involvement is not feasible due to the high number of series to annotate, and we thus use a virtual user to operate the interactive predictor by simulating user interactions from existing annotations, as during training, as in [23]. The simulated interactions are clicks, fixed to 3 per class and per image. For each task we proceeded as follows: first, we took a small subset of the annotated data and split it into three parts - the initial training set to pre-train the predictor, the validation and test sets, which remain fixed; second, we alternated between virtual data annotation and predictor update until all data was annotated, while reporting mIoU on the test set at each iteration. We also compared these results to those of a predictor trained with the same quantity of data annotated classically at the first, middle and last SAIM iterations. A key factor in these experiments is the initial training set size, which is task and data dependent. Since the interactive predictor should produce at least partial annotations, it makes sense to establish the initial training set size as a function of the predictor performance, which is measurable on the test set. For each dataset we pre-trained multiple predictors and selected the one which satisfies two criteria: (1) the IoU score is above 50% for all classes or, if impossible, the **m**IoU is above 50%, and (2) the quantity of data used for pre-training is as low as possible. In a real annotation scenario this additionally depends on the availability and performance of the expert. Therefore, we made the data selection size N as reasonably close to the size of the initial training set as possible, while keeping the overall number of iterations such that a meaningful performance change could be observed in-between. For all datasets we reported mIoU on the test set at each iteration and compared these results to those of classical system. We also estimated the annotation time using SAIM against 3D Slicer for all datasets.

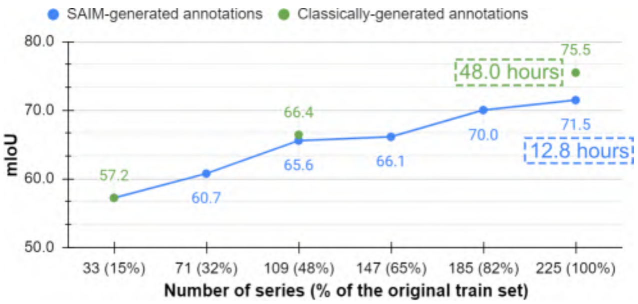

Figure 5. SAIM experimental evaluation results given as mIoU at each iteration on Pancreas CT dataset, where: green - performance using annotation produced using classical tools, blue - performance using SAIM-created annotations. Annotation time is estimated excluding pre-training data annotation with classical tools.

Female pelvis MRI. We use a female pelvis MRI segmentation dataset collected in our hospital. It consists of 97 MRI series with 3066 slices in total, manually annotated in 3D Slicer and in MITK by expert radiologists. It involves five classes: `uterus`, `bladder`, uterine `cavity`, `tumours` and `background`. The segmentation of anatomical structures of the female pelvis is particularly challenging due to a large variance in their representation. The dataset is strongly imbalanced due to the anatomical differences between the classes. The original dataset split between the training, validation and test sets is respectively: 77 series (2449 slices), 10 series (308 slices) and 10 series (309 slices). We pre-train the interactive predictor on 15% (12 out of 77 series) of the training set, which achieves 54.6% IoU on a fixed test set. We use this predictor to annotate the remaining 85% (65 series) in 6 iterations, adding 12 series at each of the first 5 iterations and 5 at the last iteration.

The performance steadily increases with each iteration. The largest change of 8.5pp IoU is observed between iterations 1 and 2 with 57.5% IoU and 66.1% IoU respectively. At 62% of the dataset annotated, SAIM achieves 69.2% IoU against 72.3% IoU for a classical system, which becomes 73.4% IoU against 75.8% IoU at 100% of data. In both cases SAIM slightly underperforms, which is expected since annotation with classical tools is expected to have a naturally higher precision. The metrics are reported in figure 3. While the performance of a classical annotation system is slightly higher in terms of accuracy, the above results show that with only 47% of data being annotated, SAIM achieves 87% of performance of this classical system trained with all 100% of the data. Crucially however, the annotation time for the whole dataset is decreased by 65%: with SAIM it takes 11.0 hours, including the time spent with 3D Slicer to annotate the initial training set, against 32.0 hours when 3D Slicer is used for the complete dataset. This shows the high impact of using SAIM in this context.

The segmentation results are shown in figure 7 and visually demonstrate the performance of the interactive predictor at the first and final SAIM iterations on samples from 6 different series. We use a fixed evaluation set to compare: (b) an interactive predictor trained only on annotations produced by classical tools (12 series - 15% of the complete training set); (c) the same interactive predictor using the complete training set (77 series) with all remaining annotations produced by the self-same predictor during SAIM iterations. Since in this dataset each class in a single image may be represented by multiple completely disconnected regions, user input is limited to 1 click per region to clearly demonstrate SAIM performance with minimal user input. We observe that in most of the cases additional data annotated using SAIM allowed to substantially improve accuracy, as supported by the metrics in figure 3. However, deterioration of accuracy can be observed in a limited number of cases, which we attribute to a shift in the training set between its partial and complete versions.

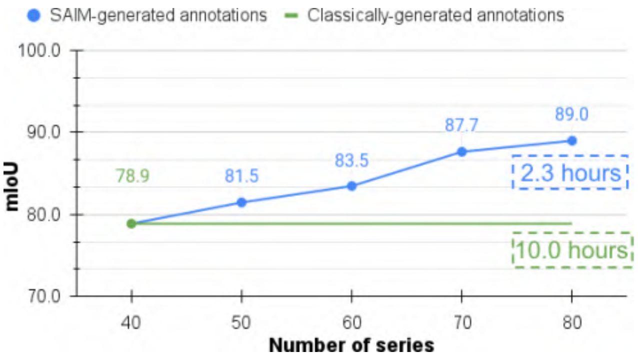

Figure 6. SAIM experimental evaluation results given as mIoU at each iteration on AMOS dataset, where annotations are done by human user via a specifically-developed GUI: green - performance using annotation produced using classical tools, blue - performance using SAIM-created annotations. Annotation time is reported excluding pre-training data annotation with classical tools.

Pancreas and liver CT. We further evaluate SAIM on the tasks of Pancreas and Liver CT segmentation. The test set ground truth is not available. We thus randomly split the training sets for both liver and pancreas, using 70%/15%/15% for training, validation and test respectively, resulting in 91/20/20 series for the liver and 198/42/42 series for the pancreas. Liver CT targets are `liver` and `cancer`, and pancreas CT targets are `pancreas` and `mass` (cyst or tumour). These datasets where annotated manually using classical tools, but the exact software and elapsed time are not specified [4]. We thus record the elapsed time from re-annotating randomly selected series in 3D Slicer, which is extrapolated to obtain the estimates.

Pancreas CT. We pre-train the interactive predictor on 15% of the training dataset (33 out of 225 series), which achieves 57.2% IoU. We use this predictor to annotate the remaining 85% (192 series) in 5 iterations, adding 38 series at each of the first 4 iterations and 40 at the last iteration. The performance increases with each iteration with the largest change of 4.9pp IoU between iterations 1 and 2 with 60.7% IoU and 65.6% IoU respectively. At 48% annotated data, SAIM marginally underperforms compared to a classical system with 65.6% IoU against 66.3% IoU, which is more notable at 100% of data with 71.5% IoU against 75.5% IoU. The metrics are reported in figure 5.

While SAIM shows a lesser performance growth between iterations on Pancreas CT compared to the female pelvis MRI dataset, the case still demonstrates that it is possible to annotate the whole Pancreas CT dataset using a predictor initially pre-trained on only 33 series. We attribute the uneven performance growth to the difficulty of distinction between the pancreas and the mass: at 48% annotated data they are at 71.9% and 59.2% against 71.1% and 61.7% IoU for SAIM and a classical system respectively. Still, it

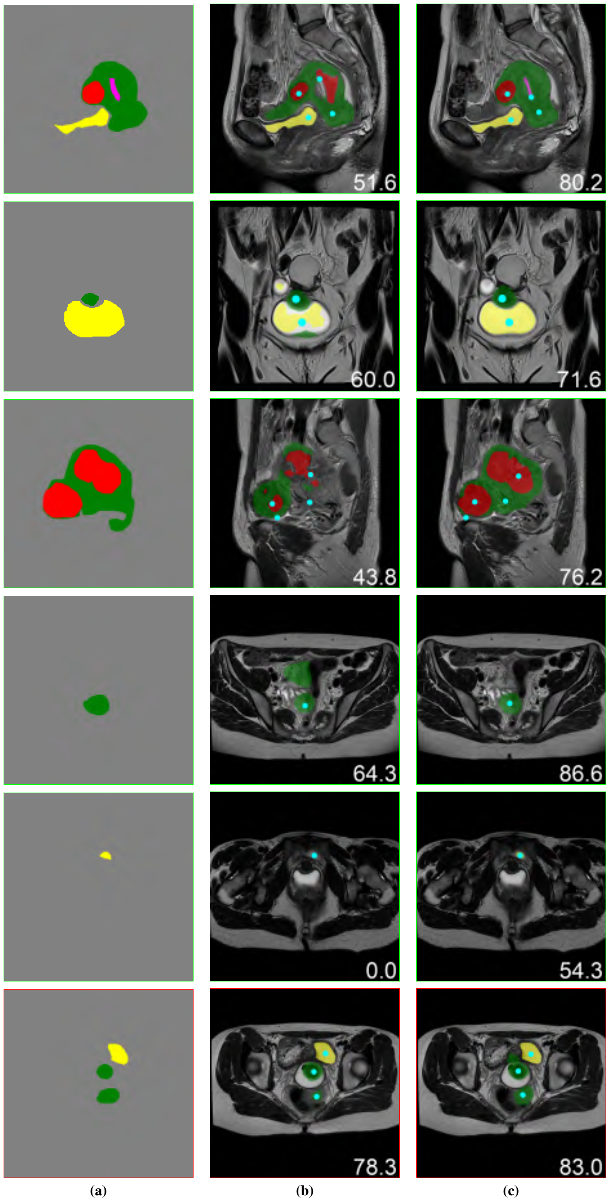

Figure 7. SAIM segmentation results in a emulated annotation scenario for female pelvis MRI segmentation dataset, where `uterus` - green, `bladder` - yellow, `tumours` - red, uterine `cavity` - pink, and user clicks are in cyan: **(a)** ground truth; **(b)** interactive predictor pre-trained on 15% (12 out of 77 series) of the complete training set **(c)** the same interactive predictor using the complete training set (77 series), but with the remaining 85% (65 out of 77 series) annotated as a part of SAIM. Performance-wise: rows in green (1-5) - improvement, row in red (6) - considered a degradation despite overall higher IoU due to the false positive for `uterus`. IoU is given in the bottom right corner.

is observed that data selection allows SAIM to achieve 87% of the performance of a classical system with 48% of data used, which demonstrated the feasibility of using SAIM for the case.

Liver CT. Applying SAIM to the Liver CT dataset allows us to start the annotation from pre-training on only 5% of the dataset (5 out of 105 series) with the initial performance at 69.7% IoU. We use this predictor to annotate the remaining 95% (100 series) in 7 iterations, adding 5 series at each of the first 2 iterations and 18 at each of the remaining iterations. The metrics are reported in figure 4. The performance grows significantly between that of the pre-trained model and iterations 1 and 2, which is an added 8.8pp and 7.1pp IoU respectively. Notably, it is enough to annotate 14% of data for SAIM to achieve 98% of the classical system's performance with all data. However, performance at iterations 3-7 fluctuates between 84.1% and 85.9%, stopping at the latter and slightly underperforming against 87.0% for a classical system. We attribute these fluctuations to SAIM being already very close to the best achievable performance.

4.3. Real Annotation Scenario

We demonstrate SAIM in a real annotation scenario for kidney MRI segmentation on the AMOS dataset, which involves three classes: `left kidney`, `right kidney` and `background`. It differs from the emulated scenarios: first, in AMOS only 100 series out of 1200 are annotated, owing to the unfeasible expert effort required; second, a human user operates the interactive predictor for the data annotation step via a developed GUI, for which the interaction number is not limited, but the elapsed time is reported. The AMOS dataset contains both annotated and non-annotated data, with 100 and 1200 MRI series respectively, collected from multi-center, multi-vendor, multi-modality, multi-phase, multi-disease patients. To the best of our knowledge the 1200 MRI series were never annotated previously, owing to the unfeasible expert effort it would require. The annotated data is originally split in 40/20/40 series for the training, validation and test sets respectively. Test annotations are unavailable. We thus leave the training set unchanged and split the validation set, with the new split being 40/10/10 series respectively.

To demonstrate SAIM we pre-trained the interactive predictor on the available 40 series annotated using classical tools and then proceeded using SAIM to double the dataset size. A total of four iterations was performed, each adding 10 series annotated by the human user. The performance steadily increases with each iteration, with the largest improvement being 4.2pp IoU between iterations 2 and 3. Overall, with 40 new annotated series the performance on the evaluation set increased by 10.1pp IoU, showing SAIM's efficiency when interactively annotating data from a large and varied non-annotated pool with the help of data selection. SAIM improved the performance of the interactive predictor by 10.1pp IoU from 78.9% IoU to 89.0% IoU. This is further reinforced by the time gain. Specifi-

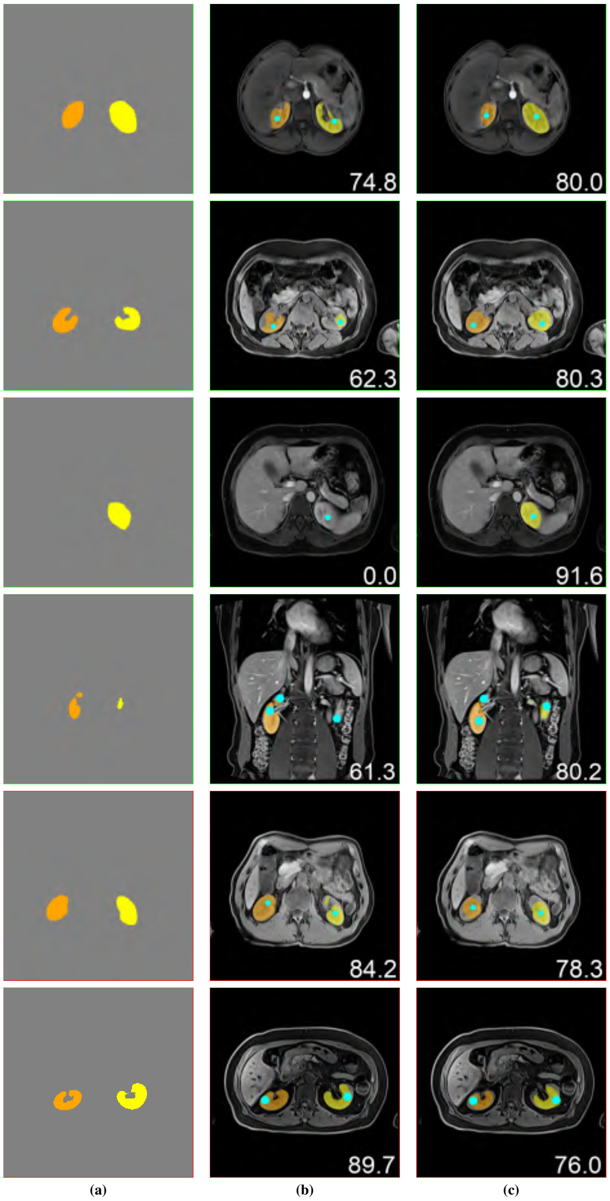

Figure 8. SAIM segmentation results in a real annotation scenario for kidney MRI segmentation on the AMOS dataset, where `right kidney` and `left kidney` are orange and yellow respectively with user clicks in cyan: **(a)** ground truth; **(b)** interactive predictor pre-trained on original AMOS training set (40 series) **(c)** the same interactive predictor after doubling the training set as a part of SAIM (80 series). Performance-wise: rows in green (1-4) - improvement, rows in red (5-6) - degradation. IoU is given in the bottom right corner.

cally, it takes 2.3 hours for SAIM and 10.0 hours estimated for classical annotation tools in 3D Slicer to double the size of the AMOS training set from 40 to 80 series. With SAIM, a single series took 3'43" against 15' on average, significantly decreasing the annotation time, all the while iteratively contributing to SAIM's predictor improvement. The metrics are reported in figure 6.

The segmentation results are shown in figure 8. We use a fixed evaluation set to compare: (b) an interactive predictor trained only on annotations produced by classical tools (40 series - original AMOS training set); (c) the same interactive predictor after doubling the training set with all new annotations produced by the self-same predictor during SAIM iterations (80 series: 40 series - original AMOS training set and 40 series - newly annotated data). The user interactions as clicks were limited to 3 per image. We observe that in most of the cases additional data annotated using SAIM allowed to improve accuracy, as supported by metrics in figure 6. As in section 4.2, we attribute the limited number of degradation cases to the training set shift.

5. Conclusion

We have proposed a general concurrent neural predictor training and data annotation framework called Single Active Interactive Model (SAIM). The strength of SAIM is its unique ability to exploit the newly annotated data as the annotation task progresses in order to improve the annotation mechanism. This is achieved by involving the predictor being trained in the steps of data selection and of interactive annotation. The neural model is thus always up-to-date and coherently shared by all the system components, contributing to optimal choices and quick improvements of the predictor performance as annotation proceeds. As a consequence, SAIM allows one to annotate massive datasets fast from very limited initial annotations.

We have evaluated SAIM by emulating annotation scenarios on fully-labelled segmentation datasets. Specifically, we have demonstrated the framework in female pelvis MRI segmentation, using a new dataset, and successfully applied it to the tasks of liver and pancreas CT segmentation from the medical segmentation decathlon challenge. We also applied SAIM to AMOS kidney MRI segmentation - a real-world case of very large dataset annotation, which cannot be feasibly annotated otherwise. This shows that SAIM jump-starts efficient interactive annotation from limited annotated data and minimises the amount of data to annotate, while improving predictor performance. This makes it a powerful two-in-one annotation and training solution to drag-and-drop in a large dataset annotation task without the need for an efficient neural predictor to be prepared first. We plan to further improve the SAIM framework towards its clinical usage. First, by considering a more efficient method of predictor update, such as fine-tuning, with re-training being a potential bottleneck of the system due to the training time required. Second, by considering a more advanced data selection approach targeting to minimise the bias introduced by the predictor. Third, through application to other tasks and expansion of the user study.

References

[1] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Q. Al-Dujaili, Ye Duan, Omran Al-Shamma, Jesus Santamaría, Mohammed Abdulraheem Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8, 2021.

[2] Massih-Reza Amini, Vasilii Feofanov, Loïc Pauletto, Emilie Devijver, and Yury Maximov. Self-training: A survey. *ArXiv*, abs/2202.12040, 2022.

[3] Mario Amrehn, Sven Gaube, M. Unberath, Frank Schebesch, Tim Horz, Maddalena Strumia, Stefan Steidl, Markus Kowarschik, and Andreas K. Maier. Ui-net: Interactive artificial neural networks for iterative image segmentation based on a user model. In *Eurographics Workshop on Visual Computing for Biomedicine*, 2017.

[4] Michela Antonelli et al. The medical segmentation decathlon. *Nature Communications*, 13, 2021.

[5] P. K. Bhagat and Prakash Choudhary. Image annotation: Then and now. *Image Vis. Comput.*, 80:1–23, 2018.

[6] Samuel Budd, Emma Claire Robinson, and Bernhard Kainz. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical image analysis*, 71, 2019.

[7] Isabella Castiglioni, Leonardo Rundo, Marina Codari, Giovanni Di Leo, Christian Salvatore, Matteo Interlenghi, Francesca Gallivanone, Andrea Cozzi, Natascha Claudia D'Amico, and Francesco Sardanelli. Ai applications to medical images: From machine learning to deep learning. *Physica medica*, 83:9–24, 2021.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[9] Andrés Diaz-Pinto, Sachidanand Alle, Alvin Ihsani, Muhammad Hamza Asad, V. Nath, Fernando P'erez-Garc'ia, Pritesh Mehta, Wenqi Li, Holger R. Roth, Tom Kamiel Magda Vercauteren, Daguang Xu, Prerna Dogra, Sébastien Ourselin, Andrew Feng, and Manuel Jorge Cardoso. Monai label: A framework for ai-assisted interactive labeling of 3d medical images. *ArXiv*, abs/2203.12362, 2022.

[10] Fujifilm. Synapse 3D. `https://synapse-emea.fujifilm.com/synapse-3d.html`. [Accessed 06-Mar-2023].

[11] Caspar Jonas Goch, Jasmin Metzger, and Marco Nolden. Abstract: Medical research data management using mitk and xnat - connecting medical image software and data management systems in a research context. In *Bildverarbeitung für die Medizin*, 2017.

[12] Leo J. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1768–1783, 2006.

[13] Raia Hadsell, Dushyant Rao, Andrei A. Rusu, and Razvan Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in Cognitive Sciences*, 24:1028–1040, 2020.

[14] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[15] Eric Heim, Tobias Ross, Alexander Seitel, Keno März, Bram Stieltjes, Matthias Eisenmann, Johannes Lebert, Jasmin Metzger, Gregor Sommer, Alexander Walter Sauter, Fides Regina Schwartz, Andreas Termer, Felix Wagner, Hannes Kenngott, and Lena Maier-Hein. Large-scale medical image annotation with crowd-powered algorithms. *Journal of Medical Imaging*, 5, 2018.

[16] Yuanfeng Ji, Haotian Bai, Jie Yang, Chongjian Ge, Ye Zhu, Ruimao Zhang, Zhuguo Li, Lingyan Zhang, Wanling Ma, Xiang Wan, and Ping Luo. Amos: A large-scale abdominal multi-organ benchmark for versatile medical image segmentation. *ArXiv*, abs/2206.08023, 2022.

[17] Ron Kikinis, Steven D. Pieper, and Kirby G. Vosburgh. 3d slicer: A platform for subject-specific image analysis, visualization, and clinical support. 2014.

[18] C. Langlotz et al. A roadmap for foundational research on artificial intelligence in medical imaging: From the 2018 nih/rsna/acr/the academy workshop. *Radiology*, 291 3:781–791, 2019.

[19] Xuan Liao, Wenhao Li, Qisen Xu, Xiangfeng Wang, Bo Jin, Xiaoyun Zhang, Ya Zhang, and Yanfeng Wang. Iteratively-refined interactive 3d medical image segmentation with multi-agent reinforcement learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9391–9399, 2019.

[20] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:318–327, 2017.

[21] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

[22] Zimo Liu, Jingya Wang, Shaogang Gong, Dacheng Tao, and Huchuan Lu. Deep reinforcement active learning for human-in-the-loop person re-identification. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6121–6130, 2019.

[23] Ivan Mikhailov, Benoit Chauveau, Nicolas Bourdel, and Adrien Bartoli. A deep learning-based interactive medical image segmentation framework. In *Applications of Medical Artificial Intelligence*, pages 98–107. Springer Nature Switzerland, 2022.

[24] Shervin Minaee, Yuri Boykov, Fatih Murat Porikli, Antonio J. Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:3523–3542, 2020.

[25] Eric N. Mortensen and William A. Barrett. Intelligent scissors for image composition. *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995.

[26] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A survey of deep

active learning. *ACM Computing Surveys (CSUR)*, 54:1 – 40, 2020.

[27] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. ”grabcut”: interactive foreground extraction using iterated graph cuts. *ACM SIGGRAPH 2004 Papers*, 2004.

[28] Lars Schmarje, Monty Santarossa, Simon-Martin Schroder, and Reinhard Koch. A survey on semi-, self- and unsupervised learning for image classification. *IEEE Access*, 9:82146–82168, 2020.

[29] Supervisely OU. Supervise.ly. `https://supervise.ly/`. [Accessed 06-Mar-2023].

[30] Yaqing Wang, Quanming Yao, James Tin-Yau Kwok, and Lionel Ming shuan Ni. Generalizing from a few examples: A survey on few-shot learning. *arXiv: Learning*, 2019.

[31] Martin J. Willemink, Wojciech A Koszek, Cailin Hardell, Jie Wu, Dominik Fleischmann, Hugh Harvey, Les R. Folio, Ronald M. Summers, D. Rubin, and Matthew P. Lungren. Preparing medical imaging data for machine learning. *Radiology*, 2020.

[32] Vivian Wen Hui Wong, Max Ferguson, Kincho H. Law, and Yung-Tsun Tina Lee. An assistive learning workflow on annotating images for object detection. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1962–1970, 2019.

[33] Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liangbo He. A survey of human-in-the-loop for machine learning. *Future Gener. Comput. Syst.*, 135:364–381, 2021.

[34] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *ArXiv*, abs/1506.03365, 2015.

[35] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109:43–76, 2019.