DOI: 10.15439/2021F86
ISSN 2300-5963 ACSIS, Vol. 25

Enabling Autonomous Medical Image Data Annotation: A human-in-the-loop Reinforcement Learning Approach

Leonardo C. da Cruz, César A. Sierra-Franco, Greis Francy M. Silva-Calpa, Alberto Barbosa Raposo
Department of Informatics
Tecgraf Institute
Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Gávea, 22451-900, Rio de Janeiro, Brazil
Email: {lccruz,casfranco,greis,abraposo}@tecgraf.puc-rio.br

Abstract—**Deep learning techniques have shown significant contributions to several fields, including medical image analysis. For supervised learning tasks, the performance of these techniques depends on a large amount of training data as well as labeled data. However, labeling is an expensive and time-consuming process. With this limitation, we introduce a new approach based on Deep Reinforcement Learning (DRL) to cost-effective annotation in a set of medical data. Our approach consists of a virtual agent to automatically label training data, and a human-in-the-loop to assist in the training of the agent. We implemented the Deep Q-Network algorithm to create the virtual agent and adopted the method mentioned above, which employs human advice to the virtual agent. Our approach was evaluated on a set of medical X-ray data in different use cases, where the agent was required to create new annotations in the form of bounding boxes from unlabeled data. Results show that an agent training with advice positively impacts obtaining new annotations from a data set with scarce labels. This result opens up new possibilities for advancing the study and implementing autonomous approaches with human advice to create a cost-effective annotation in data sets for computer-aided medical image analysis.**

I. Introduction

ARTIFICIAL Intelligence (AI) techniques, mainly those based on supervised learning, require a large amount of annotated data for training a model. In intelligent systems for the health field, the use of these techniques has contributed to the processing and analysis of medical images [1] [2]; however, the absence of labeled data has been a limitation for the implementation of those solutions.

Annotated data is necessary to enable the network to learn the relationship between a desired input and output during a machine learning model training. With sufficient data and annotation, the accuracy of a model often corresponds to or exceeds the level of expert physicians in classifying and detecting diseases [3]. However, obtaining new annotations is an expensive and time-consuming task. That labeling process is often performed manually by human experts. To reduce efforts at annotations, researchers have explored approaches of cost-effective data annotation [4]. An example of this approach are Active Learning algorithms. These algorithms aim to reduce the cost of labeling, selecting only the images to be labeled by the human, which are informative to improve the accuracy of a model [5].

However, the active learning algorithm still needs the human to make annotations of data. This aspect motivates the development of this study, contributing to creating an approach to automatically label data.

We present an approach that aims to contribute to scarce annotations based on a cost-effective data annotation approach. In particular, we focus on creating new annotations automatically on medical examinations, reducing the time and cost of the annotations. To meet the proposal, we use two objectives: 1) use of the Reinforcement Learning (RL) algorithms [6]: for creating an autonomous virtual agent. 2) insertion of the human in the training process: to teach the autonomous agent to perform its task correctly even with scarce annotations.

Reinforcement Learning (RL) is a machine learning paradigm that consists of how a virtual agent (we will adopt the term RL agent) finds a solution to a given problem, exploring interactions in the environment. Mnih et al [7] proposed Deep Reinforcement Learning (DRL) that combines RL and Convolutional Neural Network (CNN). This model is a CNN trained with a variant of the RL algorithm called Q-Learning. This method aims to enable the connection between an RL algorithm and deep neural network algorithms, operating on images with raw pixels.

In recent years, DRL models have achieved advances that surpass human performance in games such as Atari [8], has also demonstrated promise in enabling physical robots to learn complex skills in the real world [9] and in real world deployment of autonomous driving [10]. Traditionally, DRL has employed one type of algorithm that is Deep Q-Network (DQN) [7] [11].

Some authors, such as Son and Gong [12], and Liu, et al.[13] have proposed resolving the problem of scarce annotations using DQN algorithm to automate the selection process of unlabeled data. With this, an RL agent learns a data selection criterion; however, they still require the participation of a human for the labeling process. Our study shows an RL agent for automatic labeling, where we include the human in the training loop of the RL algorithm. This inclusion is due to the human's ability to teach tasks, evaluate performance, intervene at certain times to avoid unwanted actions, and increase the RL agent's learning efficiency.

In summary, this study presents the following contributions:

1) A new approach to reduce efforts to acquire new annotated data.
2) Integration of the human to speed up the learning of RL agent contributing to efficiency in creating new annotations.

The rest of this paper is organized as follows. In Section 2, we present the related work. In Section 3, we detail the proposed approach, which includes a description of reinforcement learning, the steps for understanding the Deep Q-Network (DQN), followed by the implementation of algorithms and the methods of advice. The evaluation and experimental results are described in Section 4 and 5. Finally, Section 6 shows some concluding remarks and future research perspectives.

II. Related Work

In this section, we describe some related studies that use RL algorithms to solve the scarce annotations problem through a cost-effective annotation approach. Also, we present some studies that integrate the human in the training process of these algorithms.

A. Cost-Effective Annotation

Currently, a considerable amount of medical data is available, however, the use of those data without sufficient labels or annotations is a problem when applications use supervised learning methods. Cost-effective annotation approaches are an important strategy to obtain additional annotations in a quick way, and avoiding high costs.

Saripalli et al.[14] present an approach to contribute with the labeling process where data from health monitoring devices need to be interpreted. The authors used RL algorithms to create an RL agent capable of annotating alarm data based on the annotations made by a specialist. As a result, the approach presented by the authors has created mock medical domain experts with high sensitivity, while still catching a notable number of false alarms.

Wang et al. [15] present the Deep Reinforcement Learning Active (DRLA), a new method for medical image classification. This method uses the DQN algorithm applied with the actor-critic paradigm to create an agent capable of learning a more informative image selection policy to be annotated by a human. The method presented a practical approach to relieve human efforts in making annotations.

Zimo et al.[13] proposed another approach using active learning called Deep Reinforcement Active Learning (DRAL). The objective of the study is to minimize human efforts to obtain annotation. Applied in the case of re-identification, the RL agent learns to select the best pair of images for the human annotator, which will give binary feedback to label the image as right or wrong. With each input from the human, a reward is given to the agent.

Sun and Gong [12] also present a new framework that uses active learning to annotate images. They proposed a structure that uses DRL as a data selection strategy. Instead of choosing which image to annotate using heuristic algorithms, the RL algorithm learns a selection policy. The authors evaluated the method with other studies of state of the art, which obtained superior results in a set of popular data.

Other studies address the making of automatic annotations, as a method based only on active learning [16] where the proposed method improved the classification performance compared to the baselines, in a tangent vector of the contour of the image [17]. In the present paper, the proposed method can greatly reduce the annotation time while obtaining the same or a higher annotation quality and through interaction [17].

B. Human-in-the-loop Reinforcement Learning

The inclusion of human-in-the-loop for the training of an RL agent is influenced by the human's ability to teach tasks, evaluate performance, and intervene at certain times to avoid destructive actions. This inclusion can increase the speed of the RL agent, making it confident to make quick and accurate decisions, as highlighted by Liang et al [18].

Torrey and Taylor [19] proposed an advice approach called action advice, where a human teacher suggests the student agent's actions to achieve its goal. With a fixed number of times that the human can advise, the authors present algorithms for different moments of counseling, which they call early advising, importance advising, mistake correcting, and predictive advising.

Lin et al. [20] present a method to analyze the performance of action advice in a DRL algorithm. They use human feedback to improve the performance of the RL agent through advice. This method uses an arbiter, which decides when to use actions generated by the policy of the DRL algorithm or actions advised by the human subject.

Krening [21] presents a study investigating whether human insertion as a teacher brings benefits to the student agent. As a contribution of that study, two algorithms for human interaction that promote positive experiences are presented, the Newtonian Action Advice and Object-Focused advice.

Another alternative presented in the literature to human-in-the-loop is modeling the reward that the RL agent will receive after performing actions. Denominated reward shaping, this method uses human feedback as a reward function. We find studies by Knox [22] and Arakawa [23], which show methods to train RL agents with humans as a reward function.

In the literature, there are other proposals to integrate the human in the training process of a DRL algorithm, such as by demonstration [24], imitation [25], and heuristic methods to select a state where the human subject should send actions to the RL system, as shown in the study by [26]. However, these methods need further investigation for agent training. Our approach aims to create an RL agent capable of creating new annotations from few human interactions, thus reducing the cost of generating new annotations.

Table I shows a comparison between our study and studies in the literature.

III. Proposed approach

We integrate the human in the training loop to contribute to the learning process of the RL agent. With this, the agent can generate a more significant number of annotations from a few annotated data samples. Hence, supervised convolutional neural networks could take advantage of an increased machine learning ready dataset for training purposes.

As shown in Figure 1, a problem that extends from dataset limitations are scarce annotations. Some strategies are adopted in the literature with some solutions to this problem, such as Data Augmentation, Leveraging External Labeled Datasets, Cost-Effective Annotation, Leveraging Unlabeled and Regularized Training. Based on the strategies of cost-effective annotation, we present an approach to reduce efforts to acquire new annotated data, creating an RL agent that does this task automatically.

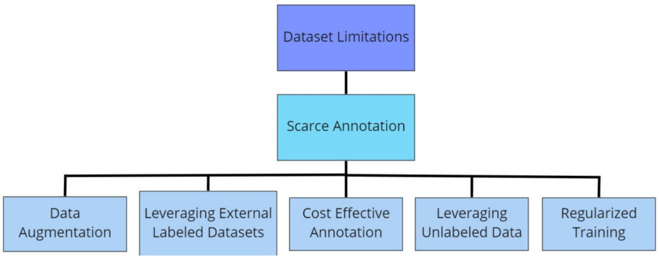
Dataset Limitations
Scarce Annotation
Data Augmentation
Leveraging External Labeled Datasets
Cost Effective Annotation
Leveraging Unlabeled Data
Regularized Training

Fig. 1. Organization of strategies that can be used based on the problem of scarce annotations (image adapted from [27]).

A. Background

Q-Learning is a classic algorithm for reinforcement learning implementation. This algorithm is an off-policy Temporal Difference that focuses on state-action value. The action value in each state is obtained using a table that is updated in each interaction with the environment, denoted Q-values, as shown in the equation 1.

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma.maxQ(s',a') - Q(s,a)] \quad (1)$$

where s is the current state and a is an action taken in this state. When each action a is taken, a new s‘ state is selected, and a reward issued for that pair of *(s, a)*. For the new selected state, a new action *a’* is taken, chosen randomly using a predefined probability (a method called Epsilon-Greedy Policy). α is the learning rate, *r* is the reward for an action taken in a given state and γ and the factor of discount.

With the success of this classic algorithm, Mnih et al [7] proposed combining Q-Learnig and Convolutions Neural Network (CNN) and presented a algorithm called Deep Q-Network.

B. Understanding Deep Q-Network

We used the DQN algorithm for the agent learning process. It uses a neural network with convolutional neural networks (CNN) to approximate the Q value of all possible actions in each state. Two techniques are the pivot for the success of this algorithm: experience replay and target network.

1) Experience replay: It serves to store the experiences acquired by the RL agent at each step. A memory buffer was used to store a predetermined amount of past experiences (batch size). At each step t, a transition is saved in this memory buffer and then used to train the neural network via stochastic gradient descent.

A transition is a tuple formed by the Markov Decision Process (MDP), where it is composed of an MPD tuple (S, A, R, S ’), being:

- *S (State)*: The current state.
- *A (action)*: Action performed in the current state.
- *R (reward)*: Reward for an action taken in a given state.
- *S’ (Next State)*: Next state.

Figure 2 illustrates the storage of transitions in a memory buffer;

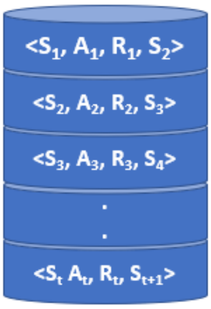

Fig. 2. Experience replay storage illustration in DQN algorithms.

2) Target Network: The Loss equation calculates the difference between the target and the prediction value, as shown in Equation 2. DQN uses a second neural network called target network to optimize the loss equation and calculate the target value.

$$Loss = (r + \gamma max_{a'} Q(s', a'; \Theta) - Q(s, a; \Theta))^2 \quad (2)$$

The Target network is a clone of the policy network and its used to calculate the target value. Initially, their weights are frozen with the weights of the original policy net and are updated with the new weights of the policy net for a certain period. The loss function given by,

$$Loss = (r + \gamma max_{a'} Q(s', a'; \Theta') - Q(s, a; \Theta))^2 \quad (3)$$

where:

- r = reward
- γ = discount factor
- Θ’ = Is updated weights once every target steps.
- Θ = Learns the correct weights by using gradient descent

TABLE I
COMPARISON TABLE BETWEEN OUR STUDY AND RELATED WORKS

Reference	algorithm	Medical Aplication	HRL Method
V. R. Saripall et al.[14]	DQN, A2C	Annotate medical signal data	N/A
J. Wang et al.[15]	DQN, AL	Image classification	N/A
Z.Liu, et al.[13]	RL, CNN, AL	N/A	Policy Shaping
Sun and Gong [12]	DQN, AL	N/A	N/A
Torrey e Taylor [19]	SARSA, Q-LEARNING	N/A	Advice
Lin, et al.[20]	DQN	N/A	advice
Krening [21]	BQL	N/A	advice
Knox [22]	Supervised Learning and RL	N/A	Reward shaping
Our study	**DQN**	**Automatic annotation in x-ray images**	**advice**

C. *Implementation of the Deep Q-Network algorithm*

Based on study of Caicedo et al. [28], [29], we started by implementing the DQN algorithm to locate objects in two-dimensional (2D) images.

At each step, the RL agent observes the current state (region of an image) and estimates the potential rewards based on the cost of taking different actions. After this calculation, it selects the action that will lead it to receive the maximum reward and moves on to the next state. This process is repeated until it reaches the terminal state. This cycle within the RL is called an episode. The following is a mapping of the MDP to the context of our work.

1) States: A medical image represents a state within our context of locating a desired region. The RL agent's area visualization is of the image size and will serve as input data for the network. At each step of the algorithm, the agent analyzes pixels of the image within its viewing area and thus calculates the best action to be taken. With each action performed by the RL agent, its viewing area will be adjusted until the object of interest is located. The next state is the current image, and the agent's viewing area is adjusted by the last action taken. The terminal state is when the agent stops performing actions because it has already completed its search. In this case, is create a new bounding box if was found an object.

2) Actions: We adopted a set of nine actions that agent RL can perform in the current state, were applied eight of which to the deformation of the agent's viewing area and one to indicate the terminal state, as shown in Figure 3. As the agent takes his actions, the agent's bounding box is deformed until it fits in the space of the object of interest.

Figure 5 illustrates the actions that the RL agent takes to detect a region of interest.

3) Rewards: The reward function used for this work is the same as presented by Caicedo et al. [28].

Equation 4 is calculated to assign rewards to the RL agent for each action taken. This equation is formed by the current visualization area of the agent RL b, together with the ground truth of the target object to be located g, and b' is the visualization area in the next step. In general, this function will attribute a positive reward to the agent if the action taken improves the IoU between the current and the next state, otherwise, the reward will be negative, as Equation 5 represents.

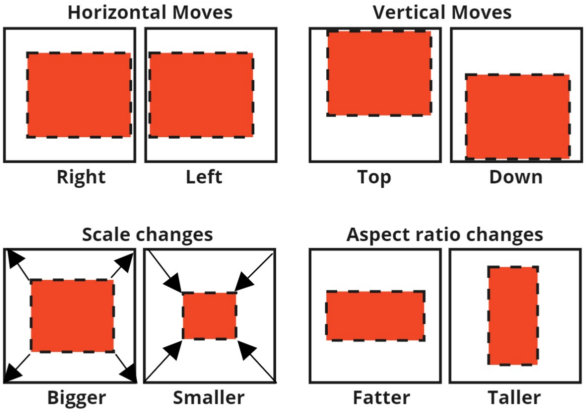

Fig. 3. Illustration of the actions that the RL agent perform in the States.

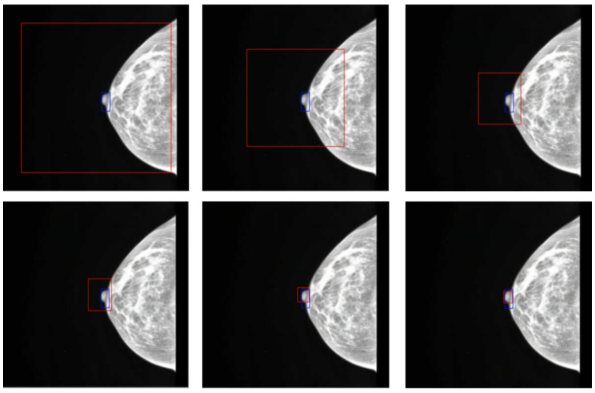

Fig. 4. Image illustrating agent RL creating an annotation in the form of the bounding box of the papilla in a mammography exam.

$$RewSign_a(s, s') = sign(IoU(b', g) - IoU(b, g)) \quad (4)$$

$$\begin{cases} +1, & \text{if } RewSign_a(s, s') > 0 \\ -1, & \text{Otherwise} \end{cases} \quad (5)$$

TABLE II
LEARNING HYPERPARAMETERS

Parameter	Value
Target network update	10000
Replay memory size	50000
Number of episodes	5
Discount factor	0.99
Learning steps	700
Leaning rate	0.00025
Epsilon start	1.0
Epsilon end	0.2
Batch size	32
Optimizer	RMSProp

Equation 6 rewards the agent when it reaches the terminal state according to the final result. In this case, we check if the IoU is greater than or equal to the threshold t (we adopt $t = 0.3$ and 0.5, depending on the use case). With that, the agent receives a positive or negative reward.

$$\begin{cases} +3, & \text{if } IoU \geq t \\ -3, & \text{Otherwise} \end{cases} \tag{6}$$

4) Hyperparameters: Table II sumarize the hyperparameters used for training the RL agent.

D. DQN architecture

DQN architecture uses a sequence of layers of a convolutional network to extract features of the image. The input to the network will be the raw frame of an image. It's common to downsample the pixel and convert the RGB values to grayscale values to reduce computation and consume less memory. Fully connected layers are used with an activation function to estimate Q values directly from the image. The last layer defines the number of units of the output layer according to the possible actions in the environment.

The following diagram shows the DQN architecture used:

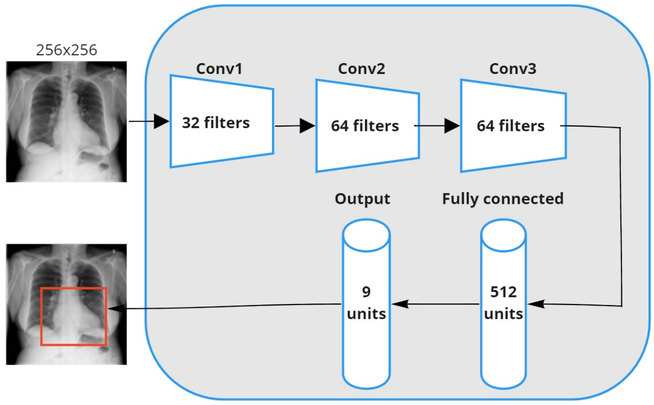

Fig. 5. Architecture used for the DQN algorithm. The input is an image with 256 x 256 pixels and processed by convolutional layers. The output layer predicts the value for the nine possible actions to be taken by the agent.

Algorithm 1 Algorithm for advice

```
Require: Medical image
Ensure: bouding box annotation
  for each episode do
    budget = 5;
    for each state do
      Calculates uncertainty;
      if uncertainty >= 1.2 then
        if budget > 0 then
          Aagent receives human advice
          budget = budget - 1
        else
          The agent takes action generated by your policy.
        end if
      end if
    end for
  end for
```

E. Implementation of an advice method

As an initial experiment, we adopted the method called early advising proposed by Torrey and Taylor [19]. The idea of this method is that the initial states are essential to the advise process, as they have a grater impact in the agent learning process. We adopted a limit of 5 pieces of advice that the human teacher can apply per episode. Algorithm 1 represents the pseudocode of the implemented method.

The human goes on to advise the RL agent when it has uncertainty about what action to take. For this experiment, a threshold of 1.2 was set, since, after a visual observation, we detected that the RL agent tends to take suitable actions below this value. As an experimental phase, the user informs the suggested action through the keyboard, inserting numbers that correspond to the agent's actions.

- Move right = 0
- Move down = 1
- Scale Bigger = 2
- Aspect ratio Fatter = 3
- Move left = 4
- Move up = 5
- Scale Smaller = 6
- Aspect ratio Taller = 7
- Trigger = 8

IV. Evaluation

As suggested by Poole and Mackworth [30], a way to measure an agent's performance is by analyzing the cumulative reward per episode. As the RL agent learns to perform the actions correctly, it receives increased rewards.

We also evaluate quantitatively the agent performance through the number of annotations that it was able to make with and without human help. In addition, we adopt metrics such as Intersection Over Union (IoU) and Average Precision (AP).

The IoU is an evaluation metric used to measure the accuracy of an object detector on a specific data set. It is a measure of the overlap between two areas, that of the bounding box generated by the algorithm and the ground-truth bounding box [31].

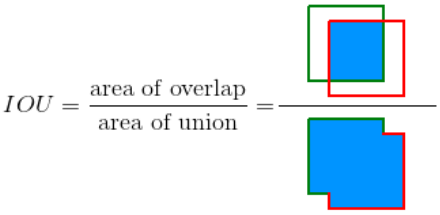

Fig. 6. The image illustrates a ground truth bounding box (in green) and a bounding box generated by a model (in red). Source [31]

Through a threshold t, the IOU allows one to classify whether the detection of the object is correct ($IOU >= t$) or incorrect ($IOU < t$). This implies that if the IOU is greater than or equal to the threshold, the *bouding box* created is within the expected (TP - True Positive). Otherwise, the created *bouding box* is lower than expected (FP - False positive).

Average Precision (AP) is the metric used to measure the model's ability to identify only the object of interest. The result ranges from 0 to 1. The closer to 1, the more accurate the model will be in creating new annotations.

After implementing our algorithm and the definitions of the evaluation metrics, we applied our approach in two different use cases.

A. Use case 1: Chest examination database

We started by analyzing the agent's performance with and without advice in a database with chest X-ray medical exams for cardiomegaly detection [32]. For this purpose, we use the chest X-ray database from NIH [33]. Cardiomegaly refers to an enlarged heart condition. It is one of the most common inherited diseases of cardiovascular diseases with a prevalence of at least 1 in 500 in the general population [34] [35].

Chest X-ray examinations are frequent and economical. However, the clinical diagnosis of a chest X-ray can be challenging and sometimes more complex than the diagnosis by chest computed tomography. The lack of large, publicly available data sets with meaningful annotations is challenging, delaying the detection and diagnosis of chest X-ray examinations.

B. Use case 2: Mammography exam database

A second use case, which we tested our approach, was in cases of mammography images. Breast cancer can be considered one of the most common global health problems and is considered the second leading cause of cancer mortality in women [36] [37].

Breast images are acquired through an x-ray examination. Two projections are made during the examination procedure: the Cranial Caudal (CC) and Medio Lateral Oblico (MLO) planes. In the CC view, the breast is seen from top-down, while in the MLO, the view is from the lateral region.

TABLE III
TRAINING DATA OF CARDIOMEGALY

Experiments	Advice	# Images	Pre-trained	# Annotations
exp1	No	31	No	11
exp2	Yes	31	No	17
exp3	No	31	Yes	17
exp4	**Yes**	**31**	**Yes**	**19**

TABLE IV
TEST DATA OF CARDIOMEGALY

Experiments	Advice	# Images	Pre-trained	# Annotations	AP
exp1	No	64	No	25	0.3
exp2	**Yes**	**64**	**No**	**38**	**0.5**
exp3	No	64	Yes	37	0.5
exp4	Yes	64	Yes	32	0.4

The nipple is a structure of interest to be observed in mammography exams. This structure helps the mammography technician verify the quality of the positioning of an exam, which can minimize the need for patients to return to repeat the exam caused by poor positioning [38]. However, detecting this structure is not trivial since, in addition to being a small structure, it does not always appear clearly in the images.

V. Experimental Results

A. Use case 1: Chest examination database

We conducted four training experiments with the RL agent to analyze its performance in taking notes automatically. The description of the data used for training is highlighted in Table III.

Table IV presents the results obtained on a set of unlabeled tests.

Figure 7 shows the evolution of the learning of the RL agent when creating annotations the structure of cardiomegaly. Throughout the episodes (indicated by the horizontal axis), is shown the accumulation of expenses (vertical axis) that the RL agent obtained. Negative rewards signify that the RL agent had a hard time learning how to take notes.

As shown in Figure 7, and Table IV, with the insertion of the human in the training loop, the agent was able to obtain better results compared to training without advice, where his learning oscillated more.

Figure 8 illustrates the result obtained by the RL agent when creating a new annotation in the form of a bounding box. The model used was the one that presented the best result, that is, the advice with a **AP = 0.5**.

B. Use case 2: Mammography exam database

Likewise, for this use case, we have carried out four training experiments. The description of the data used can be seen in Table V. The RL agent was trained to automatically create new notes of the nipple from exams projected on the CC plane (Cranio Caudal).

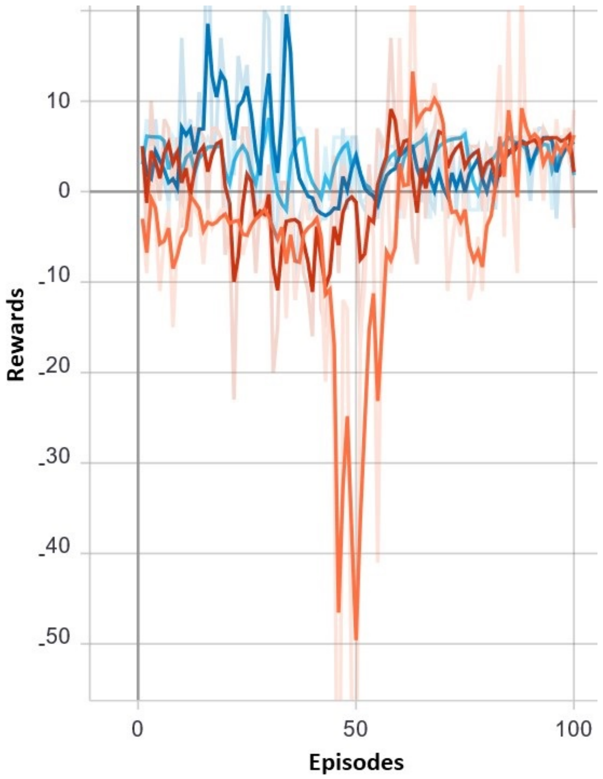

Fig. 7. Result of training of the RL agent to detect the structure of cardiomegaly. Different colors are highlighting the comparison between the experiments.

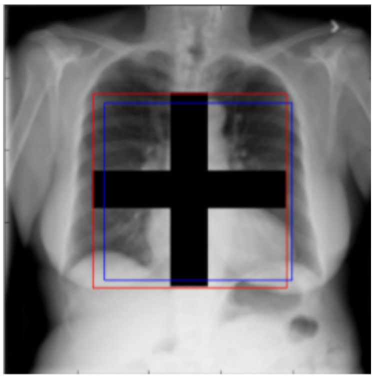

Fig. 8. Cardiomegaly image with being detected. In blue the ground truth, and red the bounding box generated by the agent.

TABLE V
TRAINING DATA OF NIPPLE

Experiments	Advice	# Images	Pre-trained	# Annotations
exp1	No	31	No	0
exp2	Yes	31	No	5
exp3	No	31	Yes	1
exp4	**Yes**	**31**	**Yes**	**15**

TABLE VI
TEST DATA OF NIPPLE

Experiments	Advice	# Images	Pre-trained	# Annotations	AP
exp1	No	192	No	0	0.003
exp2	Yes	192	No	34	0.16
exp3	No	192	Yes	6	0.03
exp4	**Yes**	**192**	**Yes**	**60**	**0.3**

We performed the RL agent testing experiments from a database without annotations. Table VI presents the results obtained.

Figure 9 shows the evolution of the learning of the RL agent when creating annotations of a region of interest to the breast. Throughout the episodes (indicated by the horizontal axis), it is shown the accumulation of expenses (vertical axis) that the RL agent obtained. Negative rewards signify that the RL agent had a hard time learning how to take notes. As the graph shows, the experiment that presented the best rewards, i.e., the agent obtained positive rewards, was through apprenticeship learning.

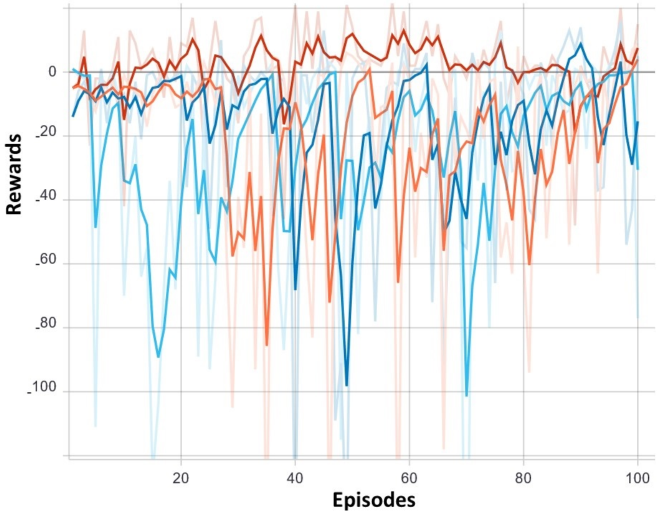

Fig. 9. Result of training of the RL agent to detect the structure of the papilla. Different colors are highlighting the comparison between the experiments.

As shown in Figure 9 and Table VI, training the RL agent with advice impacts positively in creating new annotations automatically. On the other hand, the RL agent, without counseling, proved to be less effective, having difficulties in learning the task.

Figure 10 illustrates the result obtained by the RL agent when creating a new annotation in the form of the papilla's bounding box. The model used was the one that presented the best result, that is, the advice with a **AP = 0.3**.

VI. Conclusions

This paper presents a new approach for a cost-effective annotation in a set of medical data, where annotations are performed in an automated manner by a virtual agent through

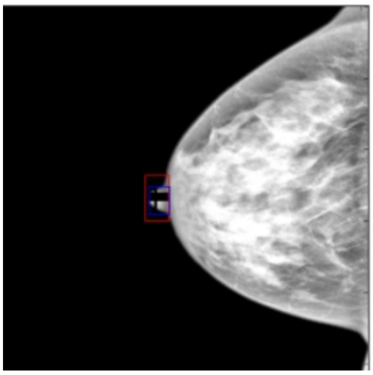

Fig. 10. Nipple image with being detected. In blue the ground truth, and red the bounding box generated by the agent.

human advice. We evaluated our approach in medical datasets for chest and mammography X-ray. The results showed that the human advice allowed the RL agent to perform learning even with a small sample of annotated data. The results also showed how early human assistance increased both precision and convergence speed to the annotation learning process.

For future work, we plan to perform experiments adjusting a more significant number of hyperparameters, analyze the amount of advice given by the human, and advise at different times during the agent training process. In addition, we intend to implement an active learning approach to increase the autonomous agent accuracy, increasing its capacity to create new annotations suitable for supervised machine learning algorithms.

References

[1] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017. [Online]. Available: https://doi.org/10.1016/j.media.2017.07.005

[2] A. Esteva, K. Chou, S. Yeung, N. Naik, A. Madani, A. Mottaghi, Y. Liu, E. Topol, J. Dean, and R. Socher, "Deep learning-enabled medical computer vision," *NPJ digital medicine*, vol. 4, no. 1, pp. 1–9, 2021. [Online]. Available: https://doi.org/10.1038/s41746-020-00376-2

[3] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *nature*, vol. 542, no. 7639, pp. 115–118, 2017. [Online]. Available: https://doi.org/10.1038/nature21056

[4] J. Yang, J. Fan, Z. Wei, G. Li, T. Liu, and X. Du, "Cost-effective data annotation using game-based crowdsourcing," *Proceedings of the VLDB Endowment*, vol. 12, no. 1, pp. 57–70, 2018. [Online]. Available: https://doi.org/10.14778/3275536.3275541

[5] S. Budd, E. C. Robinson, and B. Kainz, "A survey on active learning and human-in-the-loop deep learning for medical image analysis," *Medical Image Analysis*, p. 102062, 2021. [Online]. Available: https://doi.org/10.1016/j.media.2021.102062

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015. [Online]. Available: https://doi.org/10.1038/nature14236

[9] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, "How to train your robot with deep reinforcement learning: lessons we have learned," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 698–721, 2021. [Online]. Available: https://doi.org/10.1177/0278364920987859

[10] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, 2021. doi: 10.1109/TITS.2021.3054625

[11] T. Tajmajer, "Modular multi-objective deep reinforcement learning with decision values," in *2018 Federated conference on computer science and information systems (FedCSIS)*. IEEE, 2018, pp. 85–93. [Online]. Available: http://dx.doi.org/10.15439/2018F231

[12] L. Sun and Y. Gong, "Active learning for image classification: A deep reinforcement learning approach," in *2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*. IEEE, 2019. doi: 10.1109/CCHI.2019.8901911 pp. 71–76.

[13] Z. Liu, J. Wang, S. Gong, H. Lu, and D. Tao, "Deep reinforcement active learning for human-in-the-loop person re-identification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019. doi: 10.1109/ICCV.2019.00622 pp. 6122–6131.

[14] V. R. Saripalli, D. Pati, M. Potter, G. Avinash, and C. W. Anderson, "Ai-assisted annotator using reinforcement learning," *SN Computer Science*, vol. 1, no. 6, pp. 1–8, 2020. [Online]. Available: https://doi.org/10.1007/s42979-020-00356-z

[15] J. Wang, Y. Yan, Y. Zhang, G. Cao, M. Yang, and M. K. Ng, "Deep reinforcement active learning for medical image classification," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 33–42. [Online]. Available: https://doi.org/10.1007/978-3-030-59710-8_4

[16] J. Shim, S. Kang, and S. Cho, "Active learning of convolutional neural network for cost-effective wafer map pattern classification," vol. 33, no. 2. IEEE, 2020. doi: 10.1109/TSM.2020.2974867 pp. 258–266.

[17] F.-Q. Liu and Z.-Y. Wang, "Automatic "ground truth" annotation and industrial workpiece dataset generation for deep learning," *International Journal of Automation and Computing*, pp. 1–12, 2020.

[18] H. Liang, L. Yang, H. Cheng, W. Tu, and M. Xu, "Human-in-the-loop reinforcement learning," in *2017 Chinese Automation Congress (CAC)*, 2017. doi: 10.1109/CAC.2017.8243575 pp. 4511–4518.

[19] L. Torrey and M. Taylor, "Teaching on a budget: Agents advising agents in reinforcement learning," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 2013, pp. 1053–1060.

[20] Z. Lin, B. Harrison, A. Keech, and M. O. Riedl, "Explore, exploit or listen: Combining human feedback and policy model to speed up deep reinforcement learning in 3d worlds," *arXiv preprint arXiv:1709.03969*, 2017.

[21] S. Krening, "Humans teaching intelligent agents with verbal instruction," Ph.D. dissertation, Georgia Institute of Technology, 2019.

[22] W. B. Knox and P. Stone, "Tamer: Training an agent manually via evaluative reinforcement," in *2008 7th IEEE International Conference on Development and Learning*. IEEE, 2008, pp. 292–297.

[23] R. Arakawa, S. Kobayashi, Y. Unno, Y. Tsuboi, and S.-i. Maeda, "Dqn-tamer: Human-in-the-loop reinforcement learning with intractable feedback," *arXiv preprint arXiv:1810.11748*, 2018.

[24] G. Li, B. He, R. Gomez, and K. Nakamura, "Interactive reinforcement learning from demonstration and human evaluative feedback," in *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2018. doi: 10.1109/RO-MAN.2018.8525837 pp. 1156–1162.

[25] N. Navidi, "Human ai interaction loop training: New approach for interactive reinforcement learning," *arXiv preprint arXiv:2003.04203*, 2020.

[26] T. Mandel, Y.-E. Liu, E. Brunskill, and Z. Popovic, "Where to add actions in human-in-the-loop reinforcement learning." in *AAAI*, 2017, pp. 2322–2328.

[27] N. Tajbakhsh, L. Jeyaseelan, Q. Li, J. N. Chiang, Z. Wu, and X. Ding, "Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation," *Medical Image Analysis*, p. 101693, 2020. [Online]. Available: https://doi.org/10.1016/j.media.2020.101693

[28] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in *Proceedings of the IEEE international conference on computer vision*, 2015. doi: 10.1109/ICCV.2015.286 pp. 2488–2496.

[29] M. Otoofi, "Object localization using deep reinforcement learning Mohammad Otoofi," Master's thesis, University of Glasgow, Scotland, 2018.

[30] D. L. Poole and A. K. Mackworth, *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, 2010.

[31] R. Padilla, S. L. Netto, and E. A. da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE, 2020. doi: 10.1109/IWSSIP48289.2020.9145130 pp. 237–242.

[32] H. Amin and W. J. Siddiqui, "Cardiomegaly," *StatPearls [internet]*, 2020.

[33] K. Monowar, "National institutes of health chest x-ray dataset," May 2020. [Online]. Available: https://www.kaggle.com/khanfashee/nih-chest-x-ray-14-224x224-resized

[34] C. Semsarian, J. Ingles, M. S. Maron, and B. J. Maron, "New perspectives on the prevalence of hypertrophic cardiomyopathy," *Journal of the American College of Cardiology*, vol. 65, no. 12, pp. 1249–1254, 2015. doi: 10.1016/j.jacc.2015.01.019

[35] B. J. Maron, J. M. Gardin, J. M. Flack, S. S. Gidding, T. T. Kurosaki, and D. E. Bild, "Prevalence of hypertrophic cardiomyopathy in a general population of young adults: echocardiographic analysis of 4111 subjects in the cardia study," *Circulation*, vol. 92, no. 4, pp. 785–789, 1995. doi: 10.1161/01.cir.92.4.785

[36] M. L. Kwan, L. H. Kushi, E. Weltzien, B. Maring, S. E. Kutner, R. S. Fulton, M. M. Lee, C. B. Ambrosone, and B. J. Caan, "Epidemiology of breast cancer subtypes in two prospective cohort studies of breast cancer survivors," *Breast Cancer Research*, vol. 11, no. 3, p. R31, 2009. doi: 10.1186/bcr2261

[37] M. Moghbel, C. Y. Ooi, N. Ismail, Y. W. Hau, and N. Memari, "A review of breast boundary and pectoral muscle segmentation methods in computer-aided detection/diagnosis of breast mammography," *Artificial Intelligence Review*, pp. 1–46, 2019. [Online]. Available: https://doi.org/10.1007/s10462-019-09721-8

[38] V. Gupta, C. Taylor, S. Bonnet, L. M. Prevedello, J. Hawley, R. D. White, M. G. Flores, and B. S. Erdal, "Deep learning-based automatic detection of poorly positioned mammograms to minimize patient return visits for repeat imaging: A real-world application," *arXiv preprint arXiv:2009.13580*, 2020.