An Intelligent Collaborative System for Disease Detection

Youcef Djenouri, Asma Belhadi, Anis Yazidi, Gautam Srivastava, and Jerry Chun-Wei Lin*

***Abstract*—In this paper we introduce a novel framework for disease detection. The framework is based on intelligent agents where each agent studies the interaction among the different medical data observations using reinforcement learning and targets to detect the diseases. The agents then collaborate to reach a joint reliable conclusion on the detected diseases. Intensive experimentation has been conducted on medical data. The obtained results revealed the importance of using intelligent agents for identifying diseases in the healthcare decision making process. In addition, collaboration increases the detection rate where the numerical results reveal the superiority of the proposed framework compared to the baseline solutions for disease detection.**

***Index Terms*—Communicable Disease, Multi-Agents System, Correlation.**

I. Introduction

In the last two years, particularly since the start of the COVID-19 pandemic, technologies for controlling, managing, and detecting diseases have piqued attention [1], [2]. The pandemic has made the humanity aware of the necessity of the development of new intelligent systems for early disease detection. Artificial intelligence-based technologies hold a lot of promise in this regard and in medical applications in general [3], [4] with techniques such as multi-agent systems, deep learning networks, and evolutionary computation.

A. Motivations

Deep learning is a branch of artificial intelligence that entails creating complicated but complete models with intensive number of layers and high number of hyper-parameters. These models are capable of extracting useful characteristics directly from vast volumes of data, not just for learning. The analysis of medical data, particularly disease detection, is an intriguing area in deep learning [5]–[8]. For instance, COVID-19 samples were used to construct a smart model for calculating infection rates [5]. The latter work uses both supervised and unsupervised learning methodologies which led to a boost of detection speed by 40 percent. Using transfer learning, pathogen frames were evaluated and COVID-19 instances were validated with typical virus-based pneumonia [6]. The outcome highlights the value of employing intelligent approaches for COVID-19 diagnosis.

We can also observe examples that are substantially researched by studying different types of deep learning models in well-established in medical and disease detection in the newer fresh area of distributed deep learning [9]–[13]. The main purpose of these technologies, especially distributed ones, is to identify diseases in order to assist medical personnel in making fair and acceptable medical decisions. The detection of diseases is subject to a number of constraints, the most significant among them being data complexity. Indeed, diseases can be in different forms and shapes which will be hard to detect. To overcome these disadvantages, we are investigating a complete framework that is based on the incorporation of the deep learning (DL) and the multi-agent systems (MAS). The large number of hyper-parameters supplied by deep learning models is another significant barrier for disease detection process. Choosing these values at random results in a significant drop in the overall performance throughout the learning period. Furthermore, the parameter setting procedure for such frameworks takes a long time and there is no guarantee to reach a satisfactory convergence. The effectiveness of evolutionary computation (EC) in tackling complicated problems [14], [15] drove this research to tune the parameters of the proposed framework.

B. Contributions

To the best of our knowledge, this is the first study to look into a detailed combination of multi-agent systems, evolutionary computation, and deep learning for disease detection. The following is a list of the most important contributions:

1) ALMOST (An coLlaborative systeM fOr diSease deTection), a fresh new paradigm is provided that uses DL, MAS, and EC to identify diseases. Each agent uses various deep learning architectures to learn from medical training data and various viral diseases. Each iteration of the architecture establishes communication among the various agents for knowledge exchange and error learning rate reduction.
2) We show how several convolution neural networks can be collaborated to handle complex medical data. Different optimizations, such as batch normalization and dropout techniques, guarantee that the convolution neural network reach maturity in handling medical data.

Y. Djenouri is with Department of Mathematics and Cybernetics, SINTEF Digital, Oslo, Norway, youcef.djenouri@sintef.no

A. Belhadi is with Department of Technology, Kristiania University College, Oslo, Norway, asma.belhadi@kristiania.no

A. Yazidi is with the department of Computer Science, Oslo Metropolitan University, Oslo, Norway. He holds also a professor II position at NTNU, Trondheim, Norway and senior researcher at Oslo University Hospital (OuS), Oslo, Norway, anisy@oslomet.no

G. Srivastava is with the Department of Mathematics Computer Science, Brandon University, Brandon, Canada, SRIVASTAVAG@brandonu.ca

J. Chun-Wei Lin is with Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway, jerrylin@ieee.org (*Corresponding author)

Manuscript received February 15, 2022.

3) For intelligently exploring the configuration space of different hyper-parameter values, we suggest new evolutionary computation technique based on a genetic behaviour. This hyper-parameters optimization approach improves ALMOST's convergence for disease prediction from medical data.
4) Extensive testing was conducted to demonstrate the applicability of the ALMOST. The results revealed that ALMOST surpassed other well-known disease detection algorithms in terms of quality of returned outputs and also in terms of computational time when training large scale medical data.

C. Paper Outline

From here on out, the paper will be organized as follows. Section II provides an in-depth examination of related studies in disease detection. Section III gives a comprehensive understanding of the ALMOST methodology. A performance evaluation of ALMOST is shown in Section IV. Section V discusses the key consequences of using ALMOST on medical data, as well as the research's prospective future prospects. To conclude, Section VI ends the paper.

II. Literature Review

Hawaz *et al.* [16] investigated the use of pattern mining in medical diseases analytics. The set of COVID patients data is converted to a set of transactions, each patient is represented by a transaction, and each COVID-based information related to the patient is represented by an item. Afterwards, a pattern mining algorithm is performed on the set of transactions to extract relevant patterns. These latter were used to identify diseases based on the correlation among medical data features. Lai *et al.* [5] automated the image assessment process by exploring the segmentation and the classification deep learning based architectures. This allows to reach a reasonable estimate of the always illusory COVID-19 infection rate. Jain *et al.* [17] showed the performances of three deep learning architectures (Inception V3, Xception, and ResNeXt) to identify Covid-19 disease while using data augmentation for data enrichment. Chae *et al.* [18] predicted infectious diseases by successfully exploring the long-short term memory with the auto-regressive moving average. The proposed model is improved using the ensemble learning mechanism. Therefore, other sources of information have been collected and extracted from social networks. Wang *et al.* [6] find viral pneumonia from more than thousand of pathogen images. The experiments showed clear benefit of using intelligent methods for disease diagnosis. Ahuja *et al.* [19] implemented four deep learning architectures (ResNet18, ResNet50, ResNet101, and SqueezeNet) to capture COVID-19 from lungs CT-scan medical data. The models are pre-trained using large collection of images of different domains. The transfer learning mechanism is used to learn the COVID-19 cases from medical data. Wong *et al.* [20] analyzed the effect of the data-driven based solutions for infectious disease. They studied the combination of various data management and artificial intelligence techniques in helping the medical staffs to mitigate the risk of disease exploration, and allow better diagnosis in a smart healthcare environment. Hirano *et al.* [21] classified the different diseases using the deep learning model. The developed classification models are based on three kinds of medical images: photographic images, X-ray chest images, and retinopathy images. Three applications are then studied including skin cancer, referable diabetic, and pneumonia. Transfer learning with the adversarial neural network were implemented. The transfer learning mechanism allows to train the model developed from different medical sources, where the adversarial network allows to handle both non-targeted and targeted attacks, and to identify fake medical images. Jamshidi *et al.* [22] handle different sources of medical data, with the exploration of generative adversarial networks, extreme learning, and long-short term memory. This combination not only allows to handle heterogeneous medical data but also increases the disease detection rate. Singh *et al.* [23] worked on developing hybrid model based on both decomposition and deep learning to disease detection. The set of segments are created by deploying the k-means algorithm on medical data. These segments are then injected in the convolution neural network to predict diseases from the original medical images. Sedik *et al.* [24] showed the efficiency of using both the convolution neural network with the long-short term memory in COVID-19 identification. The study also revealed the importance of multi-modal data where the authors gathered the medical data from different sources including tomography and the X-ray images. Shalbaf *et al.* [25] implemented 15 pre-trained deep learning models to automatically identify the COVID-19. These models are based on three well-known classification based architecture including Inception, ResNet, and DenseNet. Ensemble learning is then investigated to merge the results obtained by these models using the majority voting strategy.

As can be seen from the above brief literature review, a lot of research studies explored deep learning for identifying diseases from medical data. These models used the transfer learning and data augmentation to deal with the lack of the medical data. They also used the adversarial neural network to secure the training process and deal with sensitive information of medical data. This is largely explored for distributed platforms. These techniques have a long way to go to gain acceptance in the medical field as they strive to improve the detection rate performance. To achieve mature solution for disease detection, this research work explores an intelligent collaboration mechanism involving intelligent agents, and deep learning.

III. ALMOST: An coLlaborative systeM fOr diSease deTection

A. Principle

We will start by explaining the most important aspects of the ALMOST (An coLlaborative systeM fOr diSease deTection). ALMOST is a combination of many smart strategies for solving disease detection problem, as depicted in Figure 1. For disease detection, the Convolution Neural Network (CNN) is used. The multi-agent system is researched to accurately execute the ALMOST in a distributed environment, where

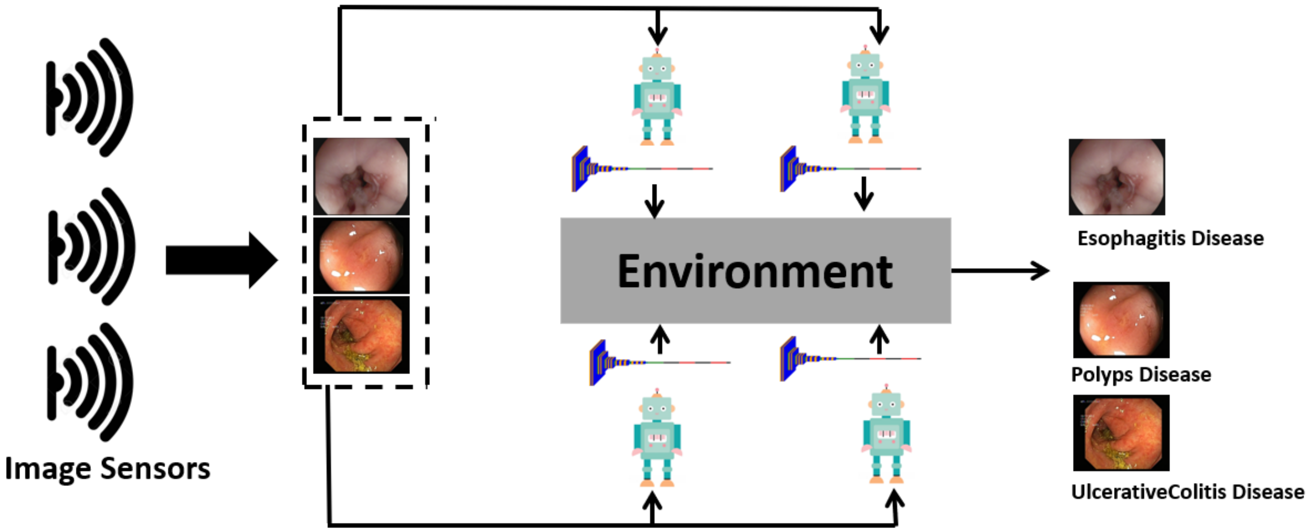

Fig. 1. ALMOST Framework.

each agent can benefit from the environment by applying the reinforcement learning paradigm. Because deep learning has a large number of parameters to tune, up to a million for some architectures, evolutionary computation is used to determine the best settings in real-time processing. The components of ALMOST will be discussed in the next parts.

B. *Learning Phase*

The learning phase is done using the CNN (Convolutional Neural Network) [26]. CNNs are a common sort of deep architecture in computer vision applications such as object detection and identification. In recent years, the adaptability of this method has helped both time series and text data. CNNs are built on the notion of extracting features from matrix data using convolutional filters. Convolutional filters create a new image by applying a set of weights to the matrix data of each pixel. In addition, well-known operators for deep learning models, batch normalization and dropout, are utilized in the training to improve the accuracy of the proposed framework. The batch normalization aids in the network's faster convergence, while the Dropout is a regulator that aids in the avoidance of overfitting. Both these methods are necessary for the network to achieve high accuracy. The following is a full description of these components:

1) **Batch Normalization**: For efficiently training a large number of layers, we adopted the batch normalization technique in all steps of the training phase. With only a few epochs, the learning process can be better converged. After each convolution layer in CNN, batch normalization is conducted.
2) **Dropout**: It's a technique for avoiding over-fitting throughout the workout. At each phase, it skips the outputs of the neurons in the hidden layers at random. It is simple to technique to make the predictions converge in the inference stage by propagating a deep network with a limited number of weights.

C. *Multi-Agents Systems*

The multi-agent system is used to learn the different diseases in the training phase. The agents collaborate with each other using the reinforcement learning process. Consider the tuple $<\mathcal{A},\mathcal{S},\mathcal{U},\mathcal{R}>$. $\mathcal{A}$ defines a multi-agent system. There are $\mathcal{A}$ agents in total, and each of them is considered a separate Markov decision process in this context. There is a finite set of environment states represented by $\mathcal{S}$, a set of actions represented by $\mathcal{U}$, and a reward function represented by $\mathcal{R}$. The strategies in $\mathcal{A}$ specify how each agent should behave given the current state and how it should make decisions about those actions. For example, in disease detection, the goal of each agent is to find an optimal strategy that maximizes the specified objective function, e.g., the number of correctly diseases detected. The following sections detail the various components of our multi-agent system:

1) Environment: The environment is a collection of databases containing a massive amount of data from smart sensor devices. This enables the environment to generate specific states for the agent's training and to estimate the optimal actions to take.
2) **State**: Each agent's next action is determined by the decisions made in earlier phases. As a result, each agent's state is composed of two components: a collection of previous actions and the current data to be processed. The number of observations in the database is used to determine the size of the state space $\mathcal{S}$.
3) **Action**: It is the assignment of each observation in the database's decision-making behavior. For instance, in a detection task, it is the assignment of each disease's category.
4) **Reward**: Determining an appropriate reward function is critical. It enables each agent in $\mathcal{A}$ to learn more effectively. We used data that contained ground truth to create a reward for the agent's actions.

So each agent $\mathcal{A}_i$ starts by scanning the observations of the

i^{th} smart sensor. It then computes the first and subsequent observations for the i^{th} intelligent sensor. A reward function for this choice is constructed using the ground truth for the first observation. This procedure is performed for each observation of the i^{th} intelligent sensor. This results in a collection of local choices, denoted LD_i, for each agent $\mathcal{A}_i$. The agents then learn from the local choices $\{LD_i\}$ to optimally find the global decision. This learning is realized by the reinforcement learning process, where a reward is given to the best agents that have high score for their local choices.

D. Hyper-parameters Optimization

To achieve optimal performance, we apply an evolutionary-based technique for hyper-parameters optimization. The adaptation of the genetic algorithm is proposed because of its well-known balance of intensification and diversification. For solving our hyper-parameters optimization problem, a full description of the proposed algorithm is given.

Let $\mathcal{HP} = \{\mathcal{HP}_1, \mathcal{HP}_2, \ldots, \mathcal{HP}_r\}$ be the set of the hyper-parameters where r represents the number of hyper-parameters in the developed ALMOST. Each $\mathcal{HP}_i$ represents a set of the potential values of the hyper-parameter in question. The configuration space $\mathcal{C}$ is then defined according to the set of all potential configurations where each configuration is a vector. The possible values of all the hyper-parameters belong to $\mathcal{HP}$. When it comes to hyper-parameters optimization, our framework focuses on deriving the optimal configuration that can provide the best accuracy result. The configuration space's size is determined by the number of all possible values for the hyper-parameters, as specified in Equation 1.

$$|\mathcal{C}| = \prod_{i=1}^{r} |\mathcal{HP}_i| \tag{1}$$

The size of the configuration space is very huge, thus it takes high computational cost to find the optimal solutions. For example, imagine that $1,000$ possible values are considered for epoch parameter, 100 possible values are considered for the error rate and 100 possible values is the number of the agents in the designed model, then the search space will include 10 million configurations, thus it is unfeasible to apply the exhaustive search methods in this case. In order to solve this challenge, evolutionary computation methods are used. The following are the primary components of our solution.

1) Population Initialization: We attempt to distribute $|\mathcal{P}|$ which is the initial population, noted $\mathcal{P}$. This starting population should be uniformly distributed in the configuration space $\mathcal{C}$. The proper examination of each of the numerous alternative configurations that tend to cover most locations within $\mathcal{C}$ may then be accomplished using this even distribution technique. We must first create the basic population, and we must do it while respecting diversity.

This process itself is begun by randomly generating one individual that is represented by a single $\mathcal{C}$ configuration. Starting with this individual, we then can generate an additional $|\mathcal{P}|-1$, where each and every new individual should be different than the individuals already generated. We can make use of a distance measure between two back to back configurations to determine the dissimilarity using the individuals generated in those configurations. $\mathcal{P}$, shown as the initial population, should in turn be able to maximize the diversification function shown in Equation 2.

$$Diversify(\mathcal{P}) = \sum_{i=1}^{|\mathcal{P}|} \sum_{j=1}^{|\mathcal{P}|} Distance(\mathcal{C}_i, \mathcal{C}_j), \tag{2}$$

where $Distance(\mathcal{C}_i, \mathcal{C}_j)$ is the distance between the configurations of the i^{th}, and j^{th} individuals, respectively.

2) Crossover: To produce new offspring, each of the two individuals in the present population goes through the following steps:

- From 1 to r, we generate a random series of crossing points, each of which we divide into two halves, the *left* and *right*.
- The left side of the original is duplicated on the left side of the first descendant, and the right side of the original is duplicated on the right side of the second descendant.
- In the second generation, the left side of the second individual is inherited by the second generation, while the right side is inherited by the first generation.

3) Mutation: The process of mutation encourages the pursuit of diversity. We use a strategy where the value of a single parameter is randomly changed in each existing configuration. The mutation point is randomly generated and can have a value between 1 and r depending on the algorithm. At each iteration of the crossover operation, the crossover operator changes the value of the mutation point in the resulting offspring.

4) Fitness Function: ALMOST's objective is to maximize disease detection accuracy. Thus, we utilize the following function to assess individuals inside populations:

$$Fitness(\mathcal{C}_i) = Detection_{ALMOST}(\mathcal{C}_i) \tag{3}$$

Note that,

- The configuration of the population's i^th individual is represented by $\mathcal{C}_i$.
- $Detection_{ALMOST}(\mathcal{C}_i)$ shows the detection ratio of the ALMOST framework by using the $\mathcal{C}_i$.

On the basis of these operations, we proposed the following hyper-parameters optimization algorithm. To begin, the initial population size, defined as $|\mathcal{P}|$, is generated randomly. Following that, each individual is constructed using the population initialization. Following that, the mutation, and crossover with mutation and crossover rates (Mr and Cr) are used to generate configurations from $\mathcal{C}$. To ensure a stable population size, each individual is evaluated using the fitness function, with an emphasis on retaining the first high-quality $|\mathcal{P}|$ individuals. At this point, all others are removed. This process is then repeated indefinitely until the maximum number of iterations, noted IMAX, has been reached.

IV. Performance Evaluation

To validate the use of the proposed ALMOST framework, extensive tests were undertaken on well-known medical databases created for disease detection applications. The experiments were conducted on a desktop computer equipped with an Intel *i7* processor and 16 GB of main memory. PythonTorch was used to implement all algorithms. We used Kvasir medical database [27] for validating the applicability of ALMOST in disease detection namely for disease data for human digestive system. The aim is to automate the detection of the endoscopic findings in the esophagus, stomach, bowel and rectum. It is represented into two versions. The first version which is called Kvasir (v1), consists of 4,000 images grouped in 8 classes showing anatomical landmarks, phatological findings or endoscopic procedures. The second version which is called Kvasir (v2) extends the first version and consists of 8,000 images with the same number of classes. The ALMOST performance is calculated using the accuracy and the $F1$ formulas which are defined as follows:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

and,

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{5}$$

such as,

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

and,

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

where,

1) True positive (TP) is determined by counting the number of corrected positive observations. An observation is called correct and positive if it is endoscopic finding and the running model considers it as an endoscopic finding.
2) True negative (TN) is determined by counting the number of corrected negative observations. An observation is called correct and negative if it is not endoscopic finding and the running model considers it as non endoscopic finding.
3) False positive (FP) is determined by counting the number of wrongly positive observations. An observation is called wrong and positive if it is an endoscopic finding and the running model considers it as non endoscopic finding.
4) False negative (FN) is determined by counting the number of wrongly negative observations. An observation is called wrong and negative if it is not an endoscopic finding and the running model considers it as an endoscopic finding.

A. Parameter Setting

In ALMOST, several parameters need to be optimized including the number of agents, the number of generations, the crossover and the mutation rates, and the population size. The choice of the optimal values of these parameters is crucial for better performance of ALMOST framework. This experiment is conducted by analyzing the behaviour of ALMOST with varying the numbers of agents, and the number of generations, the crossover rate and the mutation rate values. We varied the number of agents from 2 to 20, the number of generations, and the population size from 10 to 100, the crossover and the mutation rate from 0.01 to 0.99, the behaviour of ALMOST is summarized as follows:

1) Number of agents: The experimentation showed when we varied the number of agents from 2 to 20, the accuracy of ALMOST increases until 5 agents for Kvasir (V1), and 8 agents for Kvasir (V2) where the stabilization of the accuracy is observed.
2) Number generations: The experimentation showed when we varied the number of generations from 10 to 100, the accuracy of ALMOST increases until 45 generations for Kvasir (V1), and 58 generations for Kvasir (V2) where the stabilization of the accuracy is observed.
3) Population size: The experimentation showed when we varied the population size from 10 to 100, the accuracy of ALMOST increases until 85 individuals for Kvasir (V1), and 93 individuals for Kvasir (V2) where the stabilization of the accuracy is observed.
4) Crossover rate: The experimentation showed when we varied the crossover from 0.01 to 0.99, the accuracy of ALMOST increases until 0.35 for Kvasir (V1), and 0.47 for Kvasir (V2) where the stabilization of the accuracy is observed.
5) Mutation rate: The experimentation showed when we varied the mutation from 0.01 to 0.99, the accuracy of ALMOST increases until 0.53 for Kvasir (V1), and 0.61 for Kvasir (V2) where the stabilization of the accuracy is observed.

TABLE I
SUMMARY OF PARAMETER SETTING OF ALMOST

Dataset	$\|A\|$	IMAX	$\|P\|$	Cr	Mr
Kvasir (v1)	5	45	85	0.35	0.53
Kvasir (v2)	8	58	93	0.47	0.61

Table I gives the optimal values of the parameters used in ALMOST for both Kvasir (v1), and Kvasir (v2). The next experiments target validating the usability of the suggested ALMOST framework for disease detection. To reach this conclusion, intensive analysis has been carried out by comparing ALMOST with the baseline solutions InceptionResNet [23], and DenseNet [25]). The detailed results with complete explanation will be shown in the following.

B. Quality of the Outputs

Table II presents the quality of the outputs of ALMOST and the baseline solutions: InceptionResNet, DenseNet on Kvasir (V1) and Kvasir (V2). We varied the percentage of images used in the training from 1000 to 4000 for Kvasir (V1), and from 1000 to 8000 images for Kvasir (V2). Then, we compute the quality of the outputs represented by F1 and accuracy formulas. The results reveal the superiority of ALMOST compared to the baseline solutions for all scenarios. For instance, ALMOST accuracy is 0.96 when handling the entire data of Kvasir (V2), whereas the accuracy for the two solutions is below 0.80 when training the same data. This great achievement is obtained thanks to efficient components

TABLE II
ALMOST VS. DISEASE DETECTION SOLUTIONS.

$Dataset_{\|}Images\|$	ALMOST		InceptionResNet		DenseNet	
	F1	Accuracy	F1	Accuracy	F1	Accuracy
Kvasir(V1)_1000	0.53	0.57	0.48	0.51	0.47	0.49
Kvasir(V1)_2000	0.56	0.59	0.50	0.53	0.50	0.51
Kvasir(V1)_3000	0.58	0.63	0.52	0.55	0.52	0.53
Kvasir(V1)_4000	0.63	0.66	0.55	0.58	0.54	0.54
Kvasir(V2)_1000	0.57	0.62	0.56	0.56	0.53	0.54
Kvasir(V2)_2000	0.64	0.66	0.59	0.60	0.54	0.57
Kvasir(V2)_3000	0.69	0.73	0.60	0.60	0.58	0.61
Kvasir(V2)_4000	0.75	0.77	0.65	0.69	0.63	0.64
Kvasir(V2)_5000	0.80	0.84	0.68	0.72	0.65	0.67
Kvasir(V2)_6000	0.83	0.86	0.72	0.74	0.66	0.69
Kvasir(V2)_7000	0.87	0.91	0.75	0.77	0.71	0.72
Kvasir(V2)_8000	0.92	0.96	0.77	0.79	0.72	0.75

TABLE III
ALMOST VS. ADVANCED DISEASE DETECTION SOLUTIONS WITH DIFFERENT NUMBER OF ERROR LOSS VALUES (0.10, 0.08, 0.05, 0.02, 0.01).

Dataset	ALMOST					Xception					SqueezeNet				
	0.10	0.08	0.05	0.02	0.01	0.10	0.08	0.05	0.02	0.01	0.10	0.08	0.05	0.02	0.01
Kvasir(V1) X 1K	2178	2357	2498	2603	2759	2542	2865	2980	3006	3284	2540	2640	2759	3112	3294
Kvasir(V1) X 10K	4578	4744	4857	5009	5131	4892	5123	5546	5980	6129	4754	5123	5545	5760	5982
Kvasir(V1) X 100K	6657	7135	8249	9983	10234	7129	8832	9123	11209	12398	7105	9125	10510	11234	12786
Kvasir(V2) X 1K	2543	2764	2986	3319	3349	2769	3104	3340	3876	4129	2831	3127	3349	3981	4068
Kvasir(V2) X 10K	5874	6592	7193	8675	9831	6907	7764	8125	10942	11237	8754	9211	9938	11204	12305
Kvasir(V2) X 100K	7123	9746	11204	17594	18594	9210	10395	11204	15473	21381	10954	13058	16759	21462	24568

of ALMOST represented by the deep learning solution, and the multi-agent systems, and also to the accurate way of the hyper-optimization process.

C. ALMOST for large scale data

The next experiment has as goal to study the scalability of ALMOST compared to the baseline solutions in handling large scale data. Xception [17] and SqueezeNet [19] are used for comparison. These algorithms proved their efficiency in training large scale data. Different training scenario are launched with different data sizes of Kvasir (v1), and Kvasir (v2). Data duplication is generated by multiplying Kvasir (v1), and Kvasir (v2) multiple times (1000, 10000, and 100000). For each redundant sample, changes are generated using a generative adversarial network. We varied the error loss to be optimized from 0.10 to 0.01, the results are given in Table III. From these results, we can say clear superiority of ALMOST against the two other solutions in terms of training time. This performance can be explained by the fact that ALMOST is an optimized deep learning where collaboration between the different agents speedup the training process.

D. Case Study for ALMOST

This last part of experiments is to show some real cases detected by ALMOST. Figure 2 shows some of the correct diseases detected by ALMOST. The first three images are considered as esophagitis disease. It is an inflammation that could harm the esophagus, i.e., the muscular tube that transports food from the mouth to the stomach. The second three images are considered as polyps which is a disease characterized by tissue growths that resemble little, flat bumps or miniature mushroom stems. The majority of polyps are tiny, measuring less than half an inch in diameter. Polyps in the uterus and colon are the most prevalent, but they can also form in other sites like the ear canal and cervix. The last three images are considered as ulcerative colitis disease. They are inflammation and ulcers in the digestive tract. The innermost linings of the large intestine and rectum are affected by ulcerative colitis. Symptoms usually appear gradually rather than quickly. These images show the complexity of the disease detection problem, where the disease appears in different shape and in different sizes. ALMOST is able to identify these diseases efficiently compared to the other algorithms. These promising results confirm the usability of ALMOST in real case applications. Even ALMOST gives good results however more mature solutions need to be developed. For instance, how can we explain these diseases, how these diseases interact, and communicate? All these issues need further investigation and open research direction.

V. DISCUSSIONS AND FUTURE DIRECTIONS

The primary benefits of applying the propounded ALMOST framework to disease detection data are presented in this section. We also make some recommendations for how to improve the ALMOST framework.

1) The effective combination of smart technologies represented by deep learning, multi-agent system, and meta-heuristics produces high level of precision. For managing medical data and identifying diseases in real time, runtime performance is still a challenge. Making hybrid systems between evolutionary and exact approaches [28] to ameliorate ALMOST performances could be an interesting way to go.

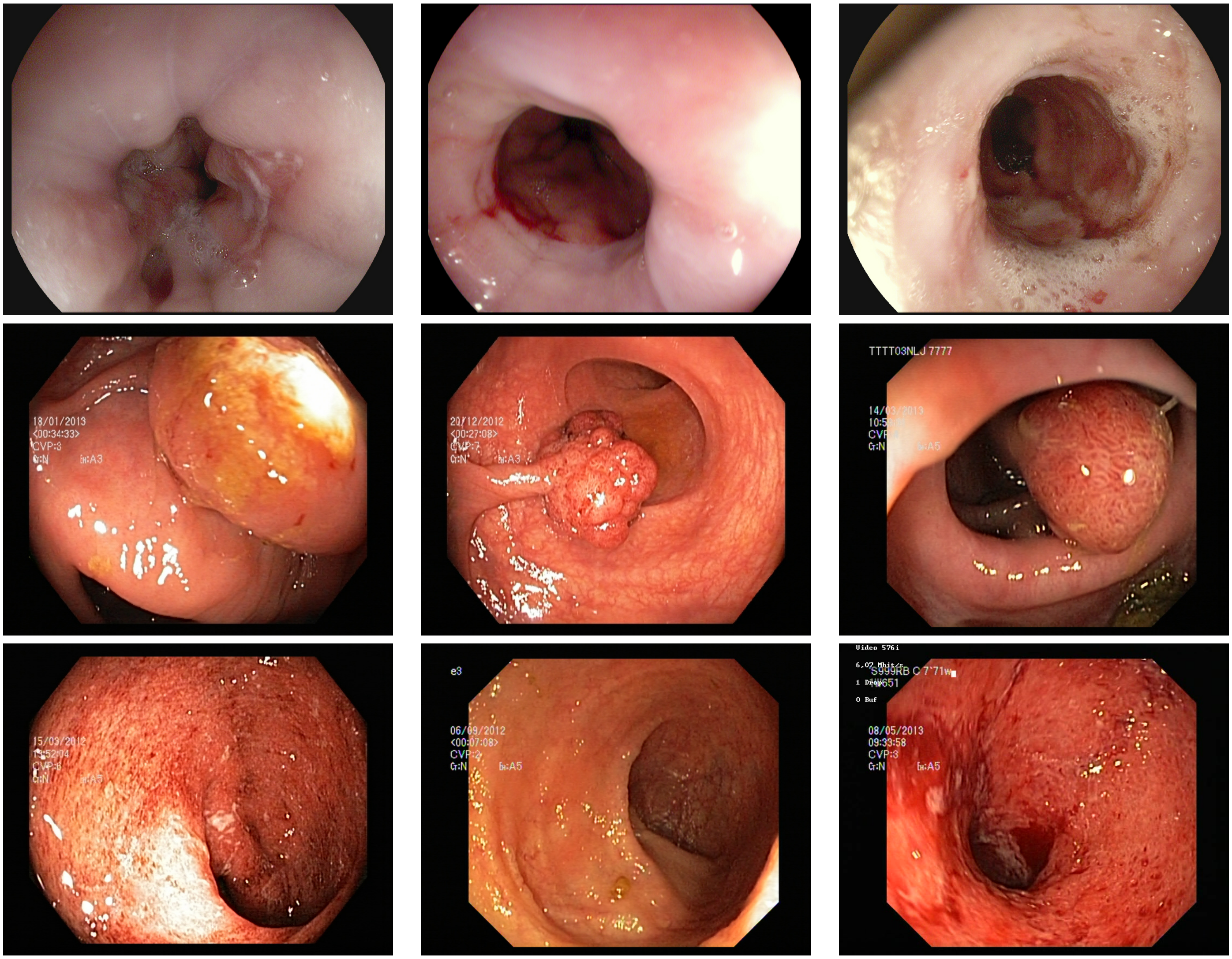

Fig. 2. Case study of ALMOST: The first three images are considered as esophagitis disease, the second three images are considered as polyps disease, where the last three images are considered as ulcerative colitis disease.

2) The proposed methodology has been used to successfully detect diseases. It yields better results than the previous approaches for detecting diseases.
   The results of the ALMOST on additional smart healthcare applications, such as brain tumor detection [29], surgery [30] and medical pattern recognition [31] would be very interesting to explore.
3) The output interpretation is a challenge in ALMOST. In fact, it is built on black-box models that do not explain the output inference process implicitly. Practitioners in healthcare settings must understand how the given outcome is achieved to trust it. This issue is being addressed by the developing discipline of XAI (eXplainable Artificial Intelligence), which provides numerous approaches for providing some level of explanation to deep learning AI solutions. We intend to incorporate XAI approaches into ALMOST. This gives more accurate interpretation of the outputs of ALMOST.

VI. Conclusion

This paper proposed an intelligent collaborative system to identify diseases. It studied the different interactions among the medical data using the intelligent agents with an efficient reinforcement learning mechanism. This allows to significantly determine the different diseases in the healthcare systems. The proposed framework has been tested on different medical data sets. The initial outcomes revealed the benefit of resorting to intelligent agents for diagnosis in the healthcare settings. Numerical results also reveal the superiority of the proposed framework compared to the baseline solutions in terms of disease detection rate.

References

[1] Y. Han and H. Yang, "The transmission and diagnosis of 2019 novel coronavirus infection disease (covid-19): a chinese perspective," *Journal of medical virology*, vol. 92, no. 6, pp. 639–644, 2020.

[2] J. He, Y. Guo, R. Mao, and J. Zhang, "Proportion of asymptomatic coronavirus disease 2019: A systematic review and meta-analysis," *Journal of medical virology*, vol. 93, no. 2, pp. 820–830, 2021.

[3] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. Iyengar, “A survey on deep learning: Algorithms, techniques, and applications,” *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–36, 2018.

[4] A. Sharma, S. Bahl, A. K. Bagha, M. Javaid, D. K. Shukla, A. Haleem *et al.*, “Multi-agent system applications to fight covid-19 pandemic,” *Apollo Medicine*, vol. 17, no. 5, p. 41, 2020.

[5] B. Wang, S. Jin, Q. Yan, H. Xu, C. Luo, L. Wei, W. Zhao, X. Hou, W. Ma, Z. Xu *et al.*, “Ai-assisted ct imaging analysis for covid-19 screening: Building and deploying a medical ai system,” *Applied Soft Computing*, vol. 98, p. 106897, 2021.

[6] S. Wang, B. Kang, J. Ma, X. Zeng, M. Xiao, J. Guo, M. Cai, J. Yang, Y. Li, X. Meng *et al.*, “A deep learning algorithm using ct images to screen for corona virus disease (covid-19),” *European radiology*, pp. 1–9, 2021.

[7] D. M. Khan, K. Masroor, M. F. M. Jailani, N. Yahya, M. Z. Yusoff, and S. M. Khan, “Development of wavelet coherence eeg as a biomarker for diagnosis of major depressive disorder,” *IEEE Sensors Journal*, 2022.

[8] R. Wang, X. Huang, X. Xu, J. Sun, S. Zheng, X. Ke, J. Yao, W. Han, M. Wei, J. Chen *et al.*, “A standalone and portable microfluidic imaging detection system with embedded computing for point-of-care diagnostics,” *IEEE Sensors Journal*, 2022.

[9] N. Balachandar, K. Chang, J. Kalpathy-Cramer, and D. L. Rubin, “Accounting for data variability in multi-institutional distributed deep learning for medical imaging,” *Journal of the American Medical Informatics Association*, vol. 27, no. 5, pp. 700–708, 2020.

[10] S. S. Roy, K. Samanta, S. Modak, S. Chatterjee, and R. Bose, “Cross spectrum aided deep feature extraction based neuromuscular disease detection framework,” *IEEE Sensors Letters*, vol. 4, no. 6, pp. 1–4, 2020.

[11] H. Ku, W. Susilo, Y. Zhang, W. Liu, and M. Zhang, “Privacy-preserving federated learning in medical diagnosis with homomorphic re-encryption,” *Computer Standards & Interfaces*, vol. 80, p. 103583, 2022.

[12] X. Xu, H. Tian, X. Zhang, L. Qi, Q. He, and W. Dou, “Discov: Distributed covid-19 detection on x-ray images with edge-cloud collaboration,” *IEEE Transactions on Services Computing*, 2022.

[13] R. Dwivedi, S. Dey, C. Chakraborty, and S. Tiwari, “Grape disease detection network based on multi-task learning and attention features,” *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17 573–17 580, 2021.

[14] U. Ahmed, J. C.-W. Lin, G. Srivastava, R. Yasin, and Y. Djenouri, “An evolutionary model to mine high expected utility patterns from uncertain databases,” *IEEE transactions on emerging topics in computational intelligence*, 2020.

[15] P. Srinivas and R. Katarya, “hyoptxg: Optuna hyper-parameter optimization framework for predicting cardiovascular disease using xgboost,” *Biomedical Signal Processing and Control*, vol. 73, p. 103456, 2022.

[16] M. S. Nawaz, P. Fournier-Viger, A. Shojaee, and H. Fujita, “Using artificial intelligence techniques for covid-19 genome analysis,” *Applied Intelligence*, vol. 51, no. 5, pp. 3086–3103, 2021.

[17] R. Jain, M. Gupta, S. Taneja, and D. J. Hemanth, “Deep learning based detection and analysis of covid-19 on chest x-ray images,” *Applied Intelligence*, vol. 51, no. 3, pp. 1690–1700, 2021.

[18] S. Chae, S. Kwon, and D. Lee, “Predicting infectious disease using deep learning and big data,” *International journal of environmental research and public health*, vol. 15, no. 8, p. 1596, 2018.

[19] S. Ahuja, B. K. Panigrahi, N. Dey, V. Rajinikanth, and T. K. Gandhi, “Deep transfer learning-based automated detection of covid-19 from lung ct scan slices,” *Applied Intelligence*, vol. 51, no. 1, pp. 571–585, 2021.

[20] Z. S. Wong, J. Zhou, and Q. Zhang, “Artificial intelligence for infectious disease big data analytics,” *Infection, disease & health*, vol. 24, no. 1, pp. 44–48, 2019.

[21] H. Hirano, A. Minagi, and K. Takemoto, “Universal adversarial attacks on deep neural networks for medical image classification,” *BMC medical imaging*, vol. 21, no. 1, pp. 1–13, 2021.

[22] M. Jamshidi, A. Lalbakhsh, J. Talla, Z. Peroutka, F. Hadjilooei, P. Lalbakhsh, M. Jamshidi, L. La Spada, M. Mirmozafari, M. Dehghani *et al.*, “Artificial intelligence and covid-19: deep learning approaches for diagnosis and treatment,” *IEEE Access*, vol. 8, pp. 109 581–109 595, 2020.

[23] P. Singh, A. Verma, and J. S. R. Alex, “Disease and pest infection detection in coconut tree through deep learning techniques,” *Computers and Electronics in Agriculture*, vol. 182, p. 105986, 2021.

[24] A. Sedik, M. Hammad, F. E. Abd El-Samie, B. B. Gupta, and A. A. Abd El-Latif, “Efficient deep learning approach for augmented detection of coronavirus disease,” *Neural Computing and Applications*, pp. 1–18, 2021.

[25] A. Shalbaf, M. Vafaeezadeh *et al.*, “Automated detection of covid-19 using ensemble of transfer learning with deep convolutional neural network based on ct scans,” *International journal of computer assisted radiology and surgery*, vol. 16, no. 1, pp. 115–123, 2021.

[26] D. Moolchandani, A. Kumar, and S. R. Sarangi, “Accelerating cnn inference on asics: A survey,” *Journal of Systems Architecture*, vol. 113, p. 101887, 2021.

[27] K. Pogorelov, K. R. Randel, C. Griwodz, S. L. Eskeland, T. de Lange, D. Johansen, C. Spampinato, D.-T. Dang-Nguyen, M. Lux, P. T. Schmidt *et al.*, “Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection,” in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 164–169.

[28] Y. Djenouri and M. Comuzzi, “Combining apriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem,” *Information Sciences*, vol. 420, pp. 1–15, 2017.

[29] M. Woźniak, J. Siłka, and M. Wieczorek, “Deep neural network correlation learning mechanism for ct brain tumor detection,” *Neural Computing and Applications*, pp. 1–16, 2021.

[30] P. N. Ramkumar, K. N. Kunze, H. S. Haeberle, J. M. Karnuta, B. C. Luu, B. U. Nwachukwu, and R. J. Williams, “Clinical and research medical applications of artificial intelligence,” *Arthroscopy: The Journal of Arthroscopic & Related Surgery*, vol. 37, no. 5, pp. 1694–1697, 2021.

[31] H. Mao and Y. Deng, “Negation of bpa: A belief interval approach and its application in medical pattern recognition,” *Applied Intelligence*, pp. 1–18, 2021.