Scientific Research Publishing

Journal of Computer and Communications, 2023, 11, 84-120
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Toward Artificial General Intelligence: Deep Reinforcement Learning Method to AI in Medicine

Daniel Schilling Weiss Nguyen*, Richard Odigie

Aspen University, Phoenix, AZ, USA
Email: *dan.nguyen@aspen.edu, odigierichie@gmail.com

How to cite this paper: Nguyen, D.S.W. and Odigie, R. (2023) Toward Artificial General Intelligence: Deep Reinforcement Learning Method to AI in Medicine. *Journal of Computer and Communications*, **11**, 84-120.
https://doi.org/10.4236/jcc.2023.119006

Received: August 26, 2023
Accepted: September 25, 2023
Published: September 28, 2023

Abstract

Artificial general intelligence (AGI) is the ability of an artificial intelligence (AI) agent to solve somewhat-arbitrary tasks in somewhat-arbitrary environments. Despite being a long-standing goal in the field of AI, achieving AGI remains elusive. In this study, we empirically assessed the generalizability of AI agents by applying a deep reinforcement learning (DRL) approach to the medical domain. Our investigation involved examining how modifying the agent's structure, task, and environment impacts its generality. **Sample:** An NIH chest X-ray dataset with 112,120 images and 15 medical conditions. We evaluated the agent's performance on binary and multiclass classification tasks through a baseline model, a convolutional neural network model, a deep Q network model, and a proximal policy optimization model. **Results:** Our results suggest that DRL agents with the algorithmic flexibility to autonomously vary their macro/microstructures can generalize better across given tasks and environments.

Keywords

Artificial Intelligence, Deep Learning, General-Purpose Learning Agent, Generalizability, Algorithmic Flexibility, Internal Autonomy

1. Introduction

Artificial general intelligence (AGI) has been posited as one of the keys to addressing complex real-world challenges in many domains such as personalized healthcare, business decision making, education, among others. However, AGI remains out of the reach of today's tools and understanding [1].

With the generality of human intelligence as the ground truth for AGI, Kadam and Vaidya [2] attributed the generality of human intelligence to the ability of the human brain to adapt to new tasks and environmental challenges and to transfer its knowledge across multiple domains. Similarly, while the key performance metrics for assessing true AGI remain the subject of much debate, the broad standard for describing AGI was often encapsulated in the words of Muehlhauser of the Machine Intelligence Research Institute as an AI agent with "the capacity to solve somewhat-arbitrary problems in somewhat-arbitrary environments" [3]. Expressed differently, the concept of AI generality, known as AGI, is the ability of an AI agent to demonstrate human-level reasoning and proficiency at performing different tasks in different environments and to transfer its learning across multiple domains [2] [3]. Based on this premise that most real-world applications occur in environments that necessitate AI agents to engage in exploration, competition, and coordination activities with other intelligent agents, it follows that deep reinforcement learning (DRL)-based approaches should provide a pathway toward AGI [4]. Hence, we empirically investigated AI generalizability by applying the DRL-based general-purpose learning agent approach to the real-world problem domain of medicine using a quantitative method with an experimental design.

1.1. The Two Main Approaches to Building AGI Systems: Rule-Based vs. Learning Systems

Pei *et al.* [5] identified two broad conceptual approaches to AGI development: the neuroscience-based (emergentist) approach and the computer science-based approach. Historically, these approaches have manifested in either rule-based systems, such as expert systems, or learning-based systems, such as neural networks, ML, and RL systems [1] [6]. Rule-based systems are inspired by logic and symbolic reasoning, they rely on human-encoded knowledge and are inherently limited in their ability to generalize to novel situations [1] [6]. In contrast, learning-based systems are grounded in cognitive psychology and neuroscience, they exhibit greater adaptability and generalization capabilities [6].

1.1.1. Research Questions and Hypotheses

From the overarching research question, the scientific research is tasked with providing theories that answer one of the three research subquestions of the same basic form below, and then go beyond descriptions to explanations for the research problem by seeking evaluative answers for why the DRL-based general-purpose learning agent approach may be more generalizable across real-world problem domains and tasks.

***RQ*1.** How can the general-purpose learning agent approach lead to more generalizable artificial agents?

H1$_0$. The general-purpose learning agent approach cannot lead to a more generalizable AI agent as measured by the agent's performance on how well it predicts an unknown entry.

H1$_a$. The general-purpose learning agent approach can lead to a more generalizable AI agent as measured by the agent's performance on how well it predicts an unknown entry.

The research subquestions:

***RQ*2.** How will varying the agent's macro/microstructure affect its generality behavior while holding its task and environment constant?

***H*2_0.** Varying the agent's macro/microstructure will have no significant effect on its generality behavior as measured by the agent's performance on how well it predicts an unknown entry.

***H*2_a.** Varying the agent's macro/microstructure will have a significant effect on its generality behavior as measured by the agent's performance on how well it predicts an unknown entry.

***RQ*3.** How will varying the agent's task affect its generality behavior while holding its environment and structure constant?

***H*3_0.** Varying the agent's task will have no significant effect on its generality behavior as measured by the agent's performance on how well it predicts an unknown entry.

***H*3_a.** Varying the agent's task will have a significant effect on its generality behavior as measured by the agent's performance on how well predicts an unknown entry.

***RQ*4.** How will varying the agent's environment affect its generality behavior while holding its task and structure constant?

***H*4_0.** Varying the agent's environment will have no significant effect on its generality behavior as measured by the agent's performance on how well it predicts an unknown entry.

***H*4_a.** Varying the agent's environment will have a significant effect on its generality behavior as measured by the agent's performance on how well it predicts an unknown entry.

1.1.2. Relationships between Variables

- *Variable construct*: general-purpose learning agent framework
- ○ Underlying variable concepts: *structure*, *task*, and *environment*
- ■ *Independent variables*: parameters, features, and environment
- *Dependent construct*: generalizability
- ○ Underlying dependent concept: *performance*
- ■ *Dependent variable*: prediction accuracy.

The empirical part of the study involved exploratory techniques for visualizing, summarizing, exploring, and making modeling decisions; the experimental part involved tests and confirmatory procedures for testing the hypotheses and answering the research questions based on inferences drawn from the predictions using the testing batch [7]-[12]. The experimentation tests involved running the AI models several times on the test set, varying the independent variables, and observing the predicted outcomes for performance. These performances of the framework were then compared against a baseline to determine the generalizability (dependent variable) of the framework on the different tasks.

1.1.3. Theoretical Framework

As Goertzel and Pennachin [13] noted, the theory on AGI is at best a patchwork of frameworks that overlap; concepts and hypotheses that are somewhat synergistic, mutually contradictory, and oftentimes problematic. Hence, since this study investigated AGI from the context of learning systems and the application of the DRL-based general-purpose learning agent approach to medical use cases, it follows that DRL should form the basis of the conceptual framework [14]. DRL combines DL with RL, which results in a very powerful technique that harnesses the immense approximation power and the ability of DNNs to represent and comprehend the real world with the ability of RL to act upon that representation [15] [16]. This relies on the representation work done by neural networks, and by learning through a combination of estimating the quality of the environment states and probability to balance exploration with exploitation, and ultimately to find the optimal policy. As such, the DRL-based general-purpose learning agent architecture and the MDP form the foundation of the study [16]. Hence, these theoretical underpinnings include the following hierarchical structure:

- Learning systems
  - DL (DNNs)
  - RL
    - L
- General-purpose learning agent architecture
  - MDP
  - Convolutional neural network (CNN).

1) Deep Neural Networks

Figure 1 illustrates a multilayer perceptron (MLP) in the context of a DNN.

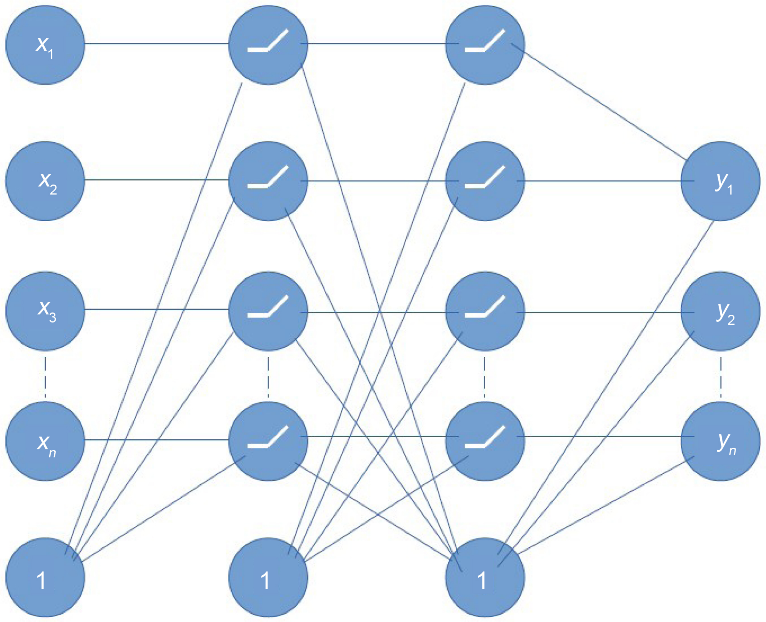


Figure 1. Multilayer perceptron (deep neural network).

The DNN uses back propagation to satisfy the constraint imposed by our data. Back propagation = *small circuit search* (in MLP in **Figure 1**).

Constraint: $y_i = f(x_i; \theta)$.

The constraint tells us that the output y_i must be the same whenever the input x_i is the same, and that if the function f is a continuous function, then similar values of the input x_i must lead to similar values of the output y_i. Assuming f is known, then the likelihood is a function of θ, which can be estimated by maximizing the likelihood.

a) SGD: $w \leftarrow w - \eta\left(\alpha\frac{\partial R(w)}{\partial w} + \frac{\partial Loss}{\partial w}\right)$

b) MLP: $a = \phi\left(\sum_j w_j x_j + b\right)$

where, the x_j are the inputs to the unit, w_j is the weight, b is the bias, φ is the nonlinear activation function, and a is the unit's activation.

Thus, constraints are imposed on the neural network circuits with our data, the neural network uses stochastic gradient descent to push information from our data with these equations into the network parameters, and uses back propagation to make small changes to its weights iteratively until its predictions satisfy the ground truth data established by our constraint. Hence, the neural network training may be viewed as a constraint satisfaction problem in which the neural network performs non-linear function approximations through a form of powerful parallel computation in the neurons of the deep neural network as shown in **Figure 1** with the MLP equation above.

2) Reinforcement Learning Theory

Apart from the agent-environment pair, the four main subelements of an RL framework system are the reward signal, value function, policy, and the model of the environment [16]. The reward signal is the single scaler number that the environment sends as feedback to an RL agent on each time step based on the agent's action. The goal of an RL problem and the immediate intrinsic desirability of environmental states (observations) in terms of the defining features of the problem the agent is facing are defined by a reward signal. Hence, the sole objective of the agent is to maximize the total cumulative reward. The reward signal is the main ground for changing the policy if an action selected by the policy results in a low reward. In contrast to a reward signal, the long-term desirability of an environmental state is specified by a value function. The value of a state can be viewed as the total amount of future expected rewards that an agent can amass, beginning from that state. The policy is a set of associations or stimulus-response rules that is core to an RL learning agent, and it is a mapping from perceived states that defines the agent's behavioral actions at any time in those environment states as shown in Equations (1) to (4) [16] [17].

The policy (π) maps states (s) to actions (a): $\pi(s) = a$ (1)

The action-value function Q gives the expected total reward from a state-action from some policy

$$Q^{\pi}(s,a) = E\left[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots \mid s_t = s, a_t = a\right] \tag{2}$$

The optimal action-value function Q^* gives the best value possible from any policy

$$Q^*(s,a) = \max_{\pi} E\left[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots \mid s_t = s, a_t = a, \pi\right] \tag{3}$$

$$= E_{s'}\left[r + \gamma \max_{a'} Q^*(s',a') \mid s,a\right] \tag{4}$$

The policy involves extensive computational search processes that use deep CNNs for their function approximation [16] (see Figure 2).

Figure 2 shows that when assigned a task, the agent interacts with an environment through a sequence of observations, actions, and rewards. The agent's goal is to select actions in a fashion that maximizes its cumulative future reward. With the DQN model, we used a deep convolutional neural network to approximate the optimal action-value function. With the PPO model, we used the deep convolutional neural network to approximate the optimal policy directly. We used transfer learning to speed up training and effectively to relax the IID hypothesis [18] [19].

1.1.4. Hypothesized Research Model

The RL framework learns action sequences through an optimal policy that results in the maximum expected reward. In DRL mode (Figure 3), however, the huge approximation capability of DNN augments and enhances the RL framework [10] [14] [20]. The DNN learns the model as the set of actions that the

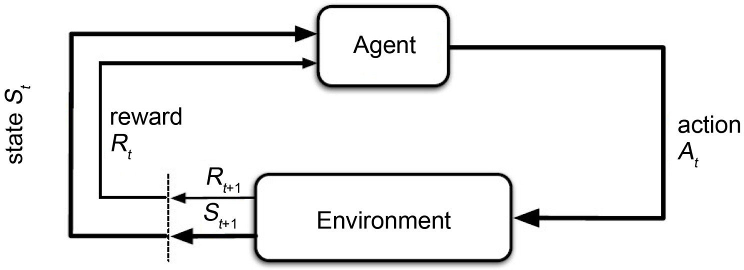


Figure 2. Agent-environment interaction in a Markov decision process [16].

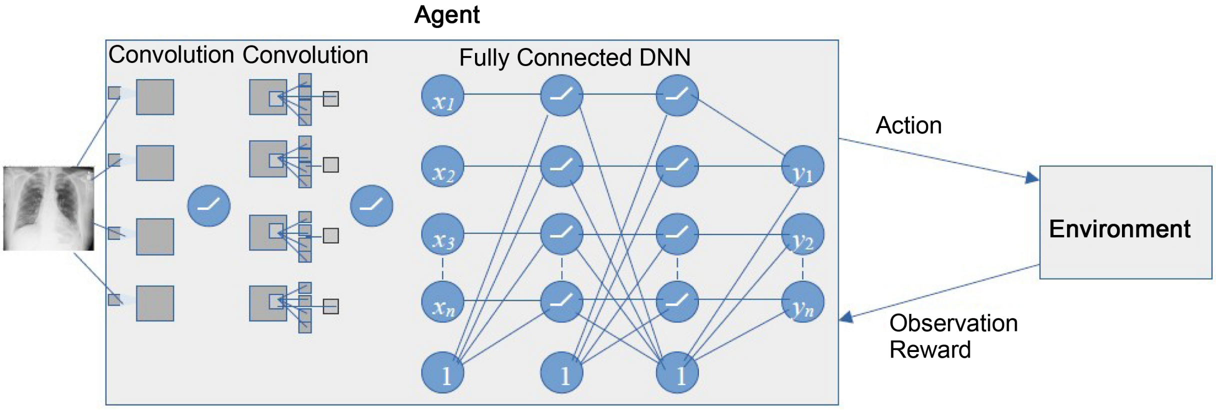


Figure 3. DRL-based general-purpose learning agent.

agent could take in the environment, it determines the choice of what decision is good to explore or what action to take, and the search degree increases or decreases at every step depending on the structure of the task [10] [15]. The model learns the quality of the environment and the most likely actions that will result in success, and therefore the most fruitful to explore to arrive at a successful environment state [10].

2. Materials and Methods

2.1. Methodology

A quantitative empirical research methodology is the optimal approach for this study since the study focuses on the scientific development aspect of AI, requires exploration and experimentation to answer the research questions, requires statistical analysis and numerical quantification of the data collection, and seeks facts and causal relationships [21] [22]. Adopting a quantitative empirical methodology for this study is in line with the tradition of empirical AI studies involved in the technological development aspects of AI where agents perform tasks in environments, with experimental research design being the de facto research design [10] [11] [21]. The choice of methodology in AI research is influenced by the objectives of the study—as studies focused on the scientific developmental aspect of AI traditionally employ the quantitative methodology, and conversely, studies focused on the social/societal aspects of AI such as the ethical issues, public perceptions, and impacts of AI, employ the qualitative methodology [21]. As such, an empirical quantitative research methodology is most suited to this study since it focuses on the technological development aspect of AI, and answering the research questions requires exploration and experimentation, quantifying of the data collection and analysis, and seeking facts and relationships [22]. This is in line with the findings of Kamiri and Mariga [21] whose analysis of 100 AI/ML articles published in IEEE journals since 2019 revealed that the quantitative research method with experimental research design was the de facto research approach for AI/ML research.

Neither a qualitative nor a mixed methodological inquiry is practical for this study since they both require people's lived experiences from the people's perspectives, whereas this study does not involve or interact with human subjects, but rather, uses existing datasets to generate results directly from the experimental process [22]. Also, a qualitative methodological inquiry is not suitable for this study because the research focus of this study is on the scientific developmental aspect of AI, and not on the social or societal aspects that require human interaction to generate rich descriptive data through structured interviews, cultural immersion, case studies, or observations. Further, since AGI is an aspirational goal that the general society does not yet access or understand, only AGI researchers within the broader AI field have some requisite lived experience to be useful human subjects in a qualitative study. In terms of furthering AGI development, however, it is doubtful that gathering data from AGI researchers through such human participant interactions would be of any benefit to AGI development or any real-world problem domain. However, experimenting with different solutions can provide us with the data to study the problem scientifically, along with providing some insight. Also, surveyed scientific developmental studies on AI/AGI use the quantitative empirical approach with experimental designs and generate results directly from the experimental process [21]. Only studies dealing with the social or societal aspects of AI utilize the qualitative method to gather data from human subjects [22].

2.2. Methodological Framework

This quantitative empirical study follows the tradition of empirical research methods for studying AI programs that perform tasks in environments. We employed quantitative data analysis and experimentation by running the DRL agent several times for convergence and observing the prediction outcome [7]-[12]. Each experimental test was a different run of the model on the test set [10] [11]. The test results are the predictions made by the AI model based on its prediction/classification of medical conditions from the medical image test set. These test results were then compared for the performance of the AI predictions against a baseline [23]. The generality of the model was then evaluated based on the test results to determine how general the model was at the different tasks. In this design, we built a DRL-based AI model with the RL approach at the top level for decision making through its policy-value network and reward function, and the lower level ran a CNN for computer vision to process medical images [7] [8] [11] [24]. Medical image data were collected from the NIH open access dataset libraries comprising thousands of already cataloged medical images [25] [26] [27] [28]. The images were randomly partitioned into a training set, validation set, and test set [20]. Hence, the study sample was the medical image test set, and the sample size was the number of images accessed in the test set.

2.3. Design of the Study

This study used an experimental design. The empirical AI research design clustered on the one hand into exploratory data analysis for visualizing, summarizing, exploring, and modeling; and on the other hand, into experimental confirmatory procedures for testing the research hypotheses, where *empirical* = *exploratory* + *experiments* [3] [17] [19] [24] [29] [30]. The agent performed a task according to the experimental protocol of medical image classification. In the exploratory part, the macro/microstructure of the agent's behavior was observed and analyzed. In the experiment part, its parameters were tuned or varied.

The six important components of the empirical AI research design were the *protocol*, *agent*, *environment*, *task*, *data collection*, and *analysis* [31]. The *agent*, *task*, and *environment* components belong to the theories of the agent's behavior domain, while the *protocol*, *data collection*, and *analysis* components belong to the domain of empirical method [31]. The behavior domain component (the interaction of these influences) was then observed and measured as the agent performed a task in an environment [31]. Medical image data were collected from the NIH open-access medical image dataset libraries available free online for ML research (see **Appendix A** for samples of medical images from the datasets and the download links). The open-access datasets library is comprised of thousands of already cataloged medical images [14] [31] [32] [33]. The images were randomly partitioned into a training set, validation set, and test set [29]. Hence, the study sample was the medical image test set, and the sample size was the number of images accessed in the test set. The design of the study is illustrated in **Figure 4**.

2.3.1. Population

The population was the anonymized dataset from the NIH open-access medical image dataset libraries available free online for ML research (see **Appendix A** for sample of medical images from the datasets and the download links). Medical imaging accounts for 90% of healthcare data and consists of different classes of medical conditions [9] [34]. These medical image data were randomly accessed from the open-access dataset library comprising thousands of deidentified and cataloged medical images [25] [26] [27] [28].

1) ***Datasets***

Annotated high-quality datasets are necessary to enhance the ability of DL-based models to draw useful hierarchical relationships [35]. The dataset is comprised of completely anonymized high-quality medical images that have been stripped of all identifying information before being made publicly available for free on the NIH open-access medical image dataset libraries. The images are generated from X-ray medical imaging technology for diagnoses of various medical conditions. In this study, these different image categories were run individually as medical tasks.

2) ***Data Preprocessing***

Real-world data are messy [36]. This requires that the data be first preprocessed and cleaned up to enable ML algorithms to process them correctly. Hence, the data were preprocessed to clean up missing values, quality, noise, and so on. Similarly, some data features were re-engineered for heterogeneity, one-hot encoding of annotations was carried out to ensure proper processing, and all images were normalized by 1/255 to ensure pixel values ranged between 0 and 1.

2.3.2. Sample

The sample comprised of 112,000 already anonymized and deidentified medical images from the NIH open-access medical image dataset libraries, and it was randomly partitioned into a training set, validation set, and test set [29]. The data were partitioned with 70% in the training set, 15% in the validation set, and the remaining 15% in the test set. Hence the medical image test set was 1500 images.

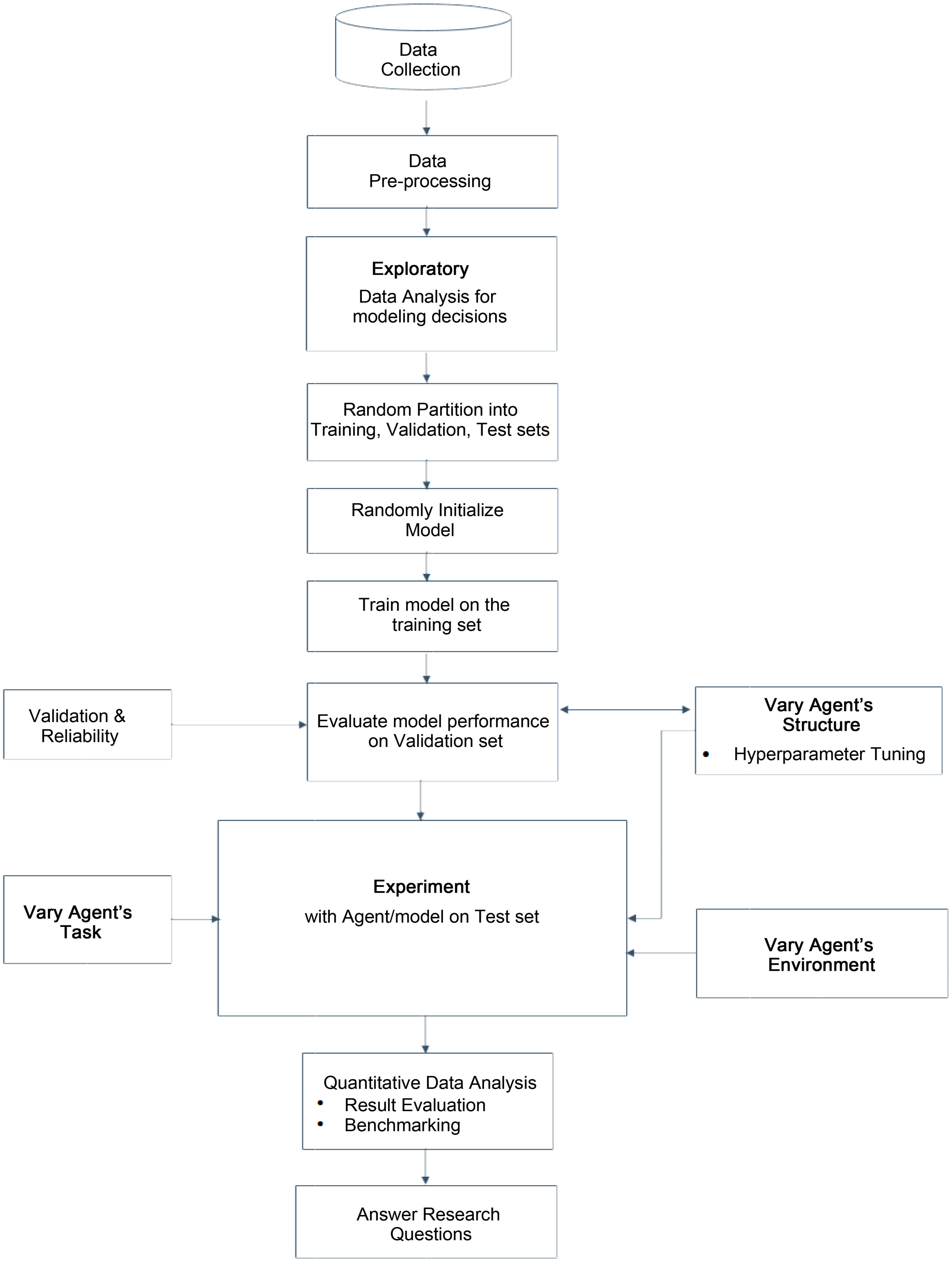

Data Collection
Data Pre-processing
Exploratory
Data Analysis for modeling decisions
Random Partition into Training, Validation, Test sets
Randomly Initialize Model
Train model on the training set
Validation & Reliability
Evaluate model performance on Validation set
Vary Agent's Structure
• Hyperparameter Tuning
Vary Agent's Task
Experiment
with Agent/model on Test set
Vary Agent's Environment
Quantitative Data Analysis
• Result Evaluation
• Benchmarking
Answer Research Questions

Figure 4. Study design.

2.3.3. Instruments

This was computer-based research conducted on a computer using software agents (model/program) to perform classification tasks on already anonymized and deidentified publicly available image data downloaded from the NIH. The experimental tasks were conducted in a software environment on a computer. The development packages were:

- Python 3.8,
- TensorFlow and Keras for DNN development,
- Sequential model API from Keras library, and,
- PyTorch for the RL framework.

2.3.4. Exploratory

The exploratory data analysis included data preprocessing or cleaning and feature engineering. These included dealing with missing values, noise, normality of data, and outliers. Similarly, some data features were re-engineered for heterogeneity, and one-hot encoding of annotations were performed. Exploratory data analysis included frequencies, skewness, standard deviation, and Pearson's correlation analysis.

2.3.5. Experimental

All experiments were run on a 64-bit Windows-10 Dell Latitude with an Intel® Core™ i7-4810MQ CPU @ 2.80 GHz processor and 16 GB of RAM. The general-purpose agent was trained and evaluated with the open-access medical data, and summaries of the experiments are presented in tables and charts. The DNN training part was carried out on the Google Colab™ platform with 25 GB of RAM, 12 GB of GPU, and 64 GB of HBM. The development packages were Python 3.8, TensorFlow and Keras for DNN development, sequential model API from Keras library, and PyTorch for the RL framework.

The collected data were partitioned with 70% in the training set, 15% in the validation set, and the remaining 15% in the test set [29]. The data were explored to enable modeling decisions. This included the following tasks.

- We randomly initialized the model,
- We trained the model on the training set,
- We evaluated the model's performance on the validation set,
- We evaluated the model on the test set [37].

The test set was left untouched until the experimental testing stage of the project [37]. In the meantime, only the training set and the validation set were used for hyperparameter tuning and optimization. After training the DNN model on the training set, the validation set was used to guide all decisions regarding model architecture and hyperparameters. During the experimental procedure, we varied the following to assess the agent's generality behavior based on the research questions:

1) We varied the structure of the agent while holding its task and environment constant.

2) We varied the agent's task while holding its environment and structure

constant.

3) We varied the agent's task environment while holding its task and structure constant.

2.3.6. Varying the Structure of the Agent

The micro/macrostructure of the agent was varied by tuning its hyperparameters. In the AI/ML field, the optimization activity that enables the learning process of an ML model to be influenced by the value of a parameter is referred to as hyperparameter tuning [38] [39]. These hyperparameters are distinct from model parameters, such as weights, which are learned by the model during training. The hyperparameter tuning involved configuring parameters such as the learning rate, number of hidden layers, number of nodes, number of epochs, regularization constant, kernels, momentum, and so on. The learning process of DNN-based models is extremely sensitive to the influence of hyperparameters [38]. For this study, we varied the following optimization settings for the different micro/macrostructure of the agents:

- We decided on the number of DNN layers (depth).
- We decided on the number of neurons in each layer (width).
- We decided on the shape/number of kernels at each layer of the CNN.
- We decided on a pretrained feature extractor and fixed the weights.
- We decided to use the ReLU and SoftMax activation function*s*.
- We decided to use dropout rather than early stopping, decided not to combine dropout with early-stopping, decided at which point in the model to use dropout, and decided on the dropout probability.
- We decided on batch normalization.
- We decided on which to use Stochastic Gradient Descent (SGD) loss optimizer.
- We decided on a learning rate.
- We decided on a loss function to optimize classification models.
- We decided on a batch size of 32 [37].

2.3.7. Varying the Task Assigned to the Agent

The task assigned to the agent was varied while its structure and environment were held constant. The problem tasks assigned to the agent were image classification and prediction problems. Its generality behavior was then observed, and its performance was compared against the baseline.

2.3.8. Varying the Agent's Environment

The agent's task environment was varied while its structure and task were held constant. During inference, images were varied between different dimensions in both binary and multiclass environments. Its generality behavior was then observed, and its performance was compared against a baseline.

2.4. Data Analysis Procedure

The data were collected, analyzed, and evaluated with exploratory data analysis and standard ML analysis procedures, and then interpreted. This exploratory first step in the ML approach helped us to develop the modeling process, understand the data, and resolve preliminary questions. The exploratory data analysis included preprocessing or data cleaning and feature engineering. These included dealing with missing values and dealing with normality of data, and outliers [35]. Similarly, some data features were re-engineered to one-hot encoding of annotations. Exploratory data analysis of the textual components of the dataset included frequencies, skewness and kurtosis, standard deviation, and Pearson's correlation analysis.

2.4.1. Evaluation Metrics

Classification evaluation metrics including the F1 weighted score, precision, and recall were the standard classification evaluation metrics. Accuracy, which is the percentage of classes that the model correctly predicts, was used to measure performance. Also, the Softmax function enabled probability outputs. Other metrics included the following.

- *Softmax Cross-Entropy Loss*: The loss function is the empirical loss (or the mean loss) across all our examples in the DNN. The Softmax cross-entropy loss for binary output classification was defined by the cross entropy between two probability distributions, and it measured how far apart the ground truth probability distribution was from the predicted probability distribution [15].
- *Mean Squared Error* (MSE): This is useful to predict the result as a real number rather than as a probability or percentage. This different output type is a continuous variable that requires a different loss called the mean squared error. This measures just the squared error, that is, the squared difference between our ground truth and our predictions averaged over the entire data set [15].
- *Confusion Matrix*: This is useful to evaluate the performance of classifiers in terms of Type 1 & Type 2 errors.
- *F1 Score*: This is the harmonic mean between precision and recall (sensitivity analysis), and it is used to evaluate the quality and performance of our classifier agents.
- *Hypothesis Testing*: This statistical testing based on the test data distribution.
- *Inference*: Inferences were drawn on the predictions using the testing batch.

2.4.2. Study Setup

This study had both an exploratory component and an experimental component. The exploratory part of this study involved an exploratory data analysis and evaluation of the dataset. The experimental part involved the training, testing, and performance evaluation of four different network model architectures, during which the performance on assigned tasks were explored and observed [7]-[12]. The setup used a basic DNN model as the baseline architecture. We improved upon the baseline model with a deep CNN architecture incorporating transfer learning [18] [19] [30]. This was followed by a DRL-based DQN agent architecture [17] [40], and a PPO agent architecture [41]. The results are as follows.

3. Results and Discussion

3.1. Results of the Exploratory Data Analysis

The preprocessed dataset contained 112,000 chest X-ray images with disease labels from 30,000 patients. Among the 15 labels observed in the dataset, one indicated no finding, while the remaining 14 represented various medical disease conditions. To facilitate binary classification, we created a new column called pneumonia_class, distinguishing cases with and without pneumonia. We removed columns with null values (no records) and dropped skewed data. Most disease conditions presented comorbidities. The no finding label was the most frequent in the dataset, accounting for 60,361 out of 112,000 images, signifying class imbalance. To address this imbalance, we down-sampled the majority class in the training set to achieve a 50-50 balance with the minority class. For the validation and test sets, we randomly sampled non-pneumonia cases at four times the size of the number of positive pneumonia cases to maintain the original real-world dataset proportions. We divided the 112,000 medical image datasets into a 70/15/15 partition, allocating 70% to the training set, 15% to the validation set, and the remaining 15% to the test set. Data splits and sizes we used as shown in **Table 1**.

3.2. Results of Experimental Tests

3.2.1. Baseline DNN Model

A supervised learning DNN model served as a performance baseline. This model consisted of five hidden dense layers with ReLU activations, dropout layers, and an output layer with a Softmax activation function [42]. Employing a batch size of 32, the model yielded a mean accuracy of 50.3%, as shown in **Table 2**. Table in **Appendix C** presents a summary of the mean prediction accuracy performance scores observed across all four models in twelve experimental trials between binary and multi-class classification tasks.

Table 1. Data splits.

Data Split	Size
Training set size	78,448
Validation set size	16,810
Test set size	16,811

Table 2. Baseline DNN model results.

Metric	Score
Test accuracy	0.503

3.2.2. Deep CNN Model with Transfer Learning

A deep CNN model was constructed and trained on the training set to classify the medical disease conditions, and then validated on the validation set. The model incorporated transfer learning by integrating a pretrained VGG-16 layer. VGG-16 is a deep CNN with 16 layers and approximately 138.3 million parameters [43]. Additionally, 12 hidden layers were added, comprising 28.7 million trainable parameters, ReLU activation, and dropout layers. A batch size of 32 was utilized, resulting in a mean accuracy of 65%, as indicated in **Table 3**.

3.2.3. DRL-Based DeepQ Network

A DRL-based learning agent employing a DQN from OpenAI Baselines was implemented [13]. As with the previous models, a batch size of 32 was used. The dueling component of the DQN introduced an architectural variation, enabling the final layer to be divided into two distinct 32-unit layers that separately converged into a single final output for each action [40]. A mean accuracy of 68% was achieved, as shown in **Table 4**.

3.2.4. DRL-Based Proximal Policy Optimization

A DRL-based learning agent utilizing a PPO model was implemented. As with the prior models, a batch size of 32 was employed. The PPO model introduced an architectural variation wherein a value head was added after the final DNN layer, with a final output [41] [44]. A mean accuracy of 69% was obtained (see **Table 5**).

3.3. Evaluation of Findings Pertaining the Research Questions

1) ***How Can the General-Purpose Learning Agent Approach Lead to More Generalizable Artificial Agents***?

The findings suggest that the general-purpose learning agent approach can lead to more generalizable artificial agents by varying their macro/microstructures to solve their given task and environment. The findings further indicate that the

Table 3. Results for the deep CNN Model with transfer learning.

Metric	Score
Test accuracy	0.650

Table 4. Results for the DeepQ network.

Metric	Score
Test accuracy	0.680

Table 5. Results for proximal policy optimization.

Metric	Score
Test accuracy	0.690

macro/microstructure is most effective at making artificial agents more generalizable. Previous studies have similarly discovered that hyperparameter optimization plays a significant role in achieving higher performance [45] [46] [47]. Although the findings also demonstrate that tasks and environments can be modified to enhance agent generalizability, this effect can be attributed to the underlying network [48] and the net effect of aligning an appropriate task and environment with a suitable macro/microstructure, as the correct architecture should ultimately resolve any given task and environment.

Further, the experimental test observations and findings indicate that varying the agent's micro/macrostructure had the strongest influence on performance. This was followed by varying the type of assigned tasks in which higher accuracy was recorded for binary classification tasks. Varying the agent's environment to higher dimensions had the least positive influence on performance. This is accounted for by the high variance in the image pixel intensity distribution for the different class samples along with their respective mean and standard deviations. The higher dimensions present a significant challenge as the model tries to adjust its internal parameters to get each feature tensor in the diverse range of pixel values in the target as close as possible to the feature tensors in the test set [49].

2) *Statistical Inference and Hypotheses Tests*

To statistically test our hypotheses, we made inferences from statistics as functions on our samples to parameters as functions on the population to infer that the general-purpose learning agent approach will generalize well on a population of unknown entries. We defined the terms as follows:

- Level of significance α of 0.05 = 5%.
- μ_{Baseline}: The mean population performance of the baseline model obtained by the combined means for the baseline Keras and CNN models from twelve trials each = 58%.
- $\mu_{\text{General-purpose agent}}$: The mean population performance of the general-purpose learning agent approach obtained by the combined means for the PPO and DQN models from twelve trials each = 68.4%.

Thus, for the hypotheses **H1** of the main research question, we assumed that the mean population performance of the baseline and that of the general-purpose learning agent approach were equal. Then we assessed the probability of the sample result to see whether this probability was small enough to reject the assumption that the entities are equal. We formulated the expressions for the null and alternate hypotheses as follows:

$$\mathbf{H1}_{0}.\ \mu_{\text{Baseline}} = \mu_{\text{General-purpose agent}}$$

$$\mathbf{H1}_{a}.\ \mu_{\text{Baseline}} \neq \mu_{\text{General-purpose agent}}$$

To decide whether to reject or not reject the null hypothesis, we looked at three specific output values, namely the t Stat or test statistic, the t Critical two-tail, and the P (T ≤ t) two-tail or the two-tail p-value (see **Table 6** and **Figure 5**). First, we compared our test statistic to the critical value by putting half (2.5%) of our level of significance, α, of 5% in each tail of the two-tail test, resulting

Table 6. Unpaired t-test for H1.

Unpaired t-test		
Alpha	0.05	
Hypothesized Mean Difference	0	
	Baseline Models	**General-purpose agent approach**
Mean	57.58	68.39
Variance	73.72605042	29.6
Observations	36	36
Observed Mean Difference	−10.81	
standard error of difference	1.677	
df	70	
t Stat	−6.507	
P (T ≤ t) one-tail	0.0000000334	
t Critical one-tail	1.67202883	
P (T ≤ t) two-tail	0.0001	
t Critical two-tail	2.002465403	

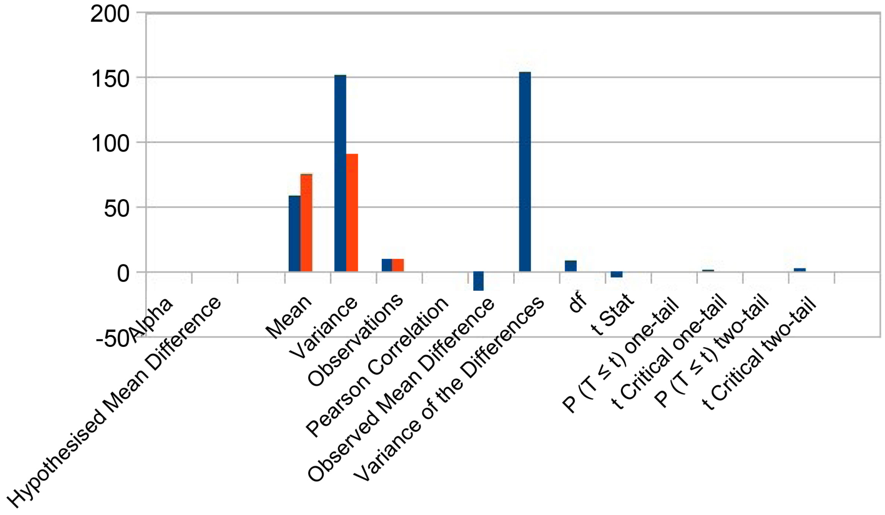


Figure 5. Unpaired t-test plot.

in two critical values of −3.25 and +3.25 on the left and right side respectively. The result showed that our test statistic of −6.507 fell in the rejection area on the left as it was less than −3.25. Thus, indicating that we must reject the Null. Next, we compared the P (T ≤ t) two-tail or the two-tail p-value against our level of significance *α*, of 5%. The result showed that our p-value of 0.0001% or 0.01% is smaller than our level of significance, *α*, of 5%. Thus, we conclude that there is a very statistically significant difference between the mean baseline model's generalizability and the mean generalizability of the general-purpose learning agent approach. Hence, we reject the Null hypothesis in favor of the alternative.

3) ***How Will Varying the Macro/Microstructure of the Agent Affect its Generality Behavior While Holding its Task and Environment Constant?***

We varied the macrostructure of the artificial agent through different model architectures. Similarly, we varied the microstructure of the artificial agent through hyperparameter tuning. Varying the agent's micro/macrostructure had the strongest influence on performance for the different model architectures ranging from 54% for the base DNN model, 69.3% for the deep CNN model, and 72.3% for the DQN model to 75.2% for the PPO model. This is in line with previous studies, which found that hyperparameter optimization was significantly responsible for higher performance [45] [46] [47].

For the hypotheses **H2** of research question ***RQ2***, we assumed that the mean population for the performance observed when the agent's macro/microstructure was varied and that of the general-purpose learning agent approach were equal. We formulated the expressions for the null and alternate hypotheses as follows:

$$\mathbf{H2}_{\mathbf{0}}.\mu_{\text{Macro/microstructure}} = \mu_{\text{Agent's Performance}}$$

$$\mathbf{H2}_{\mathbf{a}}.\mu_{\text{Macro/microstructure}} < \mu_{\text{Agent's Performance}}$$

We conducted a one-sample t-test (see **Table 7**) and compared the mean score of 73.6% obtained from varying the agent's macro/microstructure with the

Table 7. One-sample t-test for H2.

One sample t test results	
P value and statistical significance:	
	The two-tailed P value equals 0.1365 By conventional criteria, this difference is considered to be not statistically significant.
Confidence interval:	
	The hypothetical mean is 75.00000 The actual mean is 73.600000 The difference between these two values is −1.400000
The 95% confidence interval of this difference:	
	From −3.3181050474 to 0.5181050474
Intermediate values used in calculations:	

	t = −1.6065
	df = 11
	standard error of difference = 0.871

hypothetical mean value of = 75% obtained from the overall performance of the general-purpose learning agent approach to test whether the mean score from varying the agent's macro/microstructure differs significantly from 75%. The results showed that our p-value of 0.1365 or 13% is larger than our level of significance α, of 5%. Thus, we conclude that there is no statistically significant difference in the mean obtained from varying the agent's macro/microstructure with the mean value of 75% obtained from the overall performance of the general-purpose learning agent approach. Hence, we fail to reject the Null hypothesis.

4) ***How Will Varying the Agent's Task Affect Its Generality Behavior While Holding its Environment and Structure Constant?***

We varied the artificial agent's tasks between binary classification of a single condition (pneumonia), and multiclass classification of the 15 disease conditions shown in Figure C1 in **Appendix C**. The assigned task had a high influence on the performance of the different model architectures. However, since aligning a suitable task with a suitable macro/microstructure with the right architecture should ultimately solve any given task, the net positive effect on the model's performance can be attributed to the underlying network rather than the assigned task [48].

For the hypotheses **H3** of research question ***RQ*3**, we assumed that the mean population for the performance observed when the agent's task was varied and that of the general-purpose learning agent approach were equal. We formulated the expressions for the null and alternate hypotheses as follows:

$$\mathbf{H3_{0}}.\mu_{\text{Task}} = \mu_{\text{Agent's Performance}}$$

$$\mathbf{H3_{a}}.\mu_{\text{Task}} < \mu_{\text{Agent's Performance}}$$

We conducted a one-sample t-test (see Table 8) and compared the mean score of 68.4% obtained from varying the agent's task with the hypothetical mean value of 75% obtained from the overall performance of the general-purpose learning agent approach to test whether the mean score from varying the agent's task differs significantly from 75%. The results showed that our p-value of 0.00000118 is smaller than our level of significance α, of 5%. Thus, we conclude that there is a very statistically significant difference in the mean obtained from varying the agent's task with the mean value of 75% obtained from the overall performance of the general-purpose learning agent approach. Hence, we reject the Null hypothesis in favor of the alternative.

5) ***How Will Varying the Agent's Task Environment Affect Its Generality Behavior While Holding Its Task and Structure Constant?***

We varied the artificial agent's task environment between image environments with different input dimensions. On sizes greater than the standard 224 × 224 input medical image with three channels, the DRL-based models trained poorly showing little influence on performance improvement. This is a result of the significant number of computations involved in training the parameters [50]. Higher dimensions present a significant challenge for the model as it attempts to

Table 8. One-sample t-test for H3.

One sample t test results	
P value and statistical significance:	
	The two-tailed P value equals 0.00000118
	By conventional criteria, this difference is considered to be very statistically significant.
Confidence interval:	
	The hypothetical mean is 75.000000000
	The actual mean is 68.416666700
	The difference between these two values is −6.583333300
The 95% confidence interval of this difference:	
	From −8.102876501 to −5.063790099
Intermediate values used in calculations:	
	t = −9.5356
	df = 11
	standard error of difference = 0.690

adjust its internal parameters to get each feature tensor in the diverse range of pixel values in the target as close as possible to the feature tensors in the test set [49]. Hence, training improved with smaller input dimensions of one-dimensional feature vectors.

For the hypotheses **H4** of research question ***RQ*4**, we assumed that the mean population for the performance observed when the agent's task was varied and that of the general-purpose learning agent approach were equal. We formulated the expressions for the null and alternate hypotheses as follows:

$$\mathbf{H4}_{\mathbf{0}}.\mu_{\text{Environment}} = \mu_{\text{Agent's Performance}}$$

$$\mathbf{H4}_{\mathbf{a}}.\mu_{\text{Environment}} < \mu_{\text{Agent's Performance}}$$

We conducted a one-sample t-test (see Table 9) and compared the mean score of 63% obtained from varying the agent's environment with the hypothetical mean value of 75% obtained from the overall performance of the general-purpose learning agent approach to test whether the mean score from varying the agent's environment differs significantly from 75%. The results showed that our p-value of 0.00000022 is smaller than our level of significance α, of 5%. Thus, we conclude that there is an extreme statistically significant difference in the mean obtained from varying the agent's environment with the mean value of 75% obtained from the overall performance of the general-purpose learning agent approach. Hence, we reject the Null hypothesis in favor of the alternative.

Table 9. One-Sample t-test for H4.

One sample t test results	
P value and statistical significance:	
	The two-tailed P value equals 0.00000022
	By conventional criteria, this difference is considered to be extremely statistically significant.
Confidence interval:	
	The hypothetical mean is 75.0000
	The actual mean is 63.0000
	The difference between these two values is −12.0000
The 95% confidence interval of this difference:	
	From −14.3463 to −9.6537
Intermediate values used in calculations:	
	t = −11.2570
	df = 11
	standard error of difference = 1.066

3.4. Exploratory Data Analysis

Exploratory data analysis and experiment results are presented in tables, graphs, and figures in this section and in the appendices. They include visualizations from the exploratory data analysis. The classes of disease conditions and their counts in the datasets, along with the top 15 disease co-morbidity and their counts are in Table 10. The percentage distribution of each disease occurrence in the dataset are presented in Figure C1 in **Appendix C**. Also, the comorbidity distribution of the top 30 disease conditions is presented in Figure 3, and the heatmap for the correlation of comorbidity of disease conditions is presented in Figure C2 in **Appendix C**. Similarly, the pixel intensity distributions of different class samples are analyzed and presented in Figures C3-C9 in **Appendix C** with their means and standard deviations.

3.5. Deep Neural Network Model Building and Training, Testing, and Performance Evaluation

The presented data from the experiments and analysis included visualizations and reported data. The receiver operating characteristic curve for the baseline CNN model, which is helpful for predicting the probability of binary outcomes for different potential threshold values is presented in Figure C6 in **Appendix C**, with optimal threshold values between 0.451 to 0.667. Figure C7 and Figure C8

Table 10. Classes of disease conditions in the dataset.

	Disease Condition Class	Count for Each Disease Class	Top-15 Comorbidity Occurrences	Comorbidity Count
1.	No Finding	60,361	Infiltration|Pneumonia	199
2.	Infiltration	19,894	Edema|Infiltration|Pneumonia	137
3.	Effusion	13,317	Atelectasis|Pneumonia	108
4.	Atelectasis	11,559	Edema|Pneumonia	83
5.	Nodule	6331	Effusion|Pneumonia	54
6.	Mass	5782	Effusion|Infiltration|Pneumonia	42
7.	Pneumothorax	5302	Consolidation|Pneumonia	36
8.	Consolidation	4667	Atelectasis|Infiltration|Pneumonia	34
9.	Pleural_Thickening	3385	Atelectasis|Effusion|Pneumonia	23
10.	Cardiomegaly	2776	Edema|Effusion|Infiltration|Pneumonia	21
11.	Emphysema	2516	Edema|Effusion|Pneumonia	19
12.	Edema	2303	Nodule|Pneumonia	19
13.	Fibrosis	1686	Atelectasis|Effusion|Infiltration|Pneumonia	18
14.	Pneumonia	1431	Atelectasis|Consolidation|Pneumonia	15
15.	Hernia	227	Consolidation|Infiltration|Pneumonia	13

in **Appendix C** show that the maximum F1 score is 0.571, the threshold is 0.451, the precision is 0.400, and the recall is 1.000. Figure C9 in **Appendix C** presents the model's training and validation loss vs. accuracy.

4. Discussion of Findings and Theoretical Foundations

DRL has been posited as a promising pathway toward general AI based on the premise that most real-world applications of AI occur in complex environments where artificial agents must engage in exploration, competition, and coordination activities with other intelligent agents [4] [10] [11]. Also, since DRL is the only subfield of AI that unbinds AI from fixed datasets to learn from the experience of interacting with its environment, DRL naturally lends itself toward being more generalizable than other current AI approaches [10] [35] [51]. A quantitative empirical research methodology was adopted for this study since the study required exploration and experimentation to answer the research questions, and it required statistical analysis and numerical quantification of the collected data to determine facts and relationships [21] [22]. Thus, we investigated AI generalizability through the application of the DRL-based general-purpose-learning agent approach to different medical tasks and tested the framework on NIH medical image datasets comprising 112,000 chest x-ray images with disease labels from 30,000 patients. There were 15 classes in the dataset indicating the different medical disease conditions shown in Table 10.

The study employed both an exploratory and an experimental component.

The exploratory component involved exploratory data analysis and evaluation of the dataset. The experimental part involved the training, testing, and performance evaluation of four different network model architectures, and their performance on assigned tasks was then observed and explored [7]-[12]. Specifically, we applied artificial agents to the classification of healthcare patients' medical image data and varied the agent's micro/macrostructure between different models and hyperparameters, we varied the agent's tasks between binary and multiclass classification tasks, and we varied the agent's task environment between image environments with different dimensions during inference [52]. The setup used a basic DNN model as the naive baseline architecture. We improved upon the baseline model with a deep CNN architecture incorporating transfer learning [19] [30] [51]. This was followed by a DRL-based DQN agent architecture [17] [40] and a PPO agent architecture [41]. The DQN agent used *Q*-learning which is a value-based off-policy method that enables learning from the data to compute the target without considering how the experience is generated, while the PPO is an on-policy actor-critic algorithm [16]. Both the DQN and PPO agents are model-free algorithms (see **Appendix B** for DQN and PPO algorithms). In defining the reward function for our RL models, greedy actions were preferred rather than taking actions that affect their long-term reward, the models focused on predicting each medical condition separately. Hence, we experimented with cumulative rewards per episode with discount factors between 0.1 and 0.9. Regularisation of the neural networks was implemented with dropouts in the final MLP layer. Since the NIH image dataset is a hard classification problem even for human experts, we implemented transfer learning through a pretrained CNN as the feature extractor.

On the main research question of *how the general-purpose learning agent approach can lead to more generalizable artificialagents*, our findings indicate that DRL-based AI agents can be more generalizable by varying their macro/microstructures to suit their task and environment. However, we encountered two primary obstacles to generalizability: task-independent learning due to catastrophic forgetting of previous knowledge as the Deep Neural Network (DNN) attempted to learn multiple new tasks sequentially, and the necessity of problem-specific design and tuning, which required us to hand-craft problem-specific representations for different types of tasks [53] [54] [55]. Although several solutions have been proposed to address these issues, none have been entirely satisfactory [16]. Consequently, this study's artificial agents were trained separately for each task, rather than achieving a single agent that could perform all tasks simultaneously. Regarding the challenge of problem-specific design and tuning, classification algorithms typically require classes or categories of items to be uniquely presented for clear distinction between them. However, the comorbidity disease conditions in the NIH dataset were presented as tuples, and this presented a special case implying that a condition could belong to multiple classes. While such logic is understandable to humans familiar with notions of shared characteristics, it confounded the AI models, necessitating some hand-craft hand-crafting. Consequently, this study did not produce a single agent that simultaneously performed all its assigned tasks. Instead, we manually modified and trained different agents separately for each task.

To overcome the limitations of problem-specific representations and the need for manual intervention, algorithmic flexibility could equip the agent with the versatility to explore different algorithmic options autonomously. This would require the agent to have internal autonomy to self-vary its macro/microstructure, thus eliminating the need for problem-specific design and tuning. This idea aligns with similar arguments presented in other studies [16] [56]. Ha [56] argued that instead of merely learning a policy to manipulate an agent with a fixed design, the agent's physical structure should be optimized by learning a version of its design along with a policy that best suits its task. Sutton and Barto [16] stressed the importance of having Reinforcement Learning (RL) agents select their tasks and predictions rather than relying on manual human intervention. They further argued that even if this requires a general language for predictions, it will promote exploration as the agent would have to systematically explore large spaces of possible predictions to identify the most useful ones.

On the subquestion of *how varying the agent's micro/macrostructure affects its generality behavior*, varying the agent's micro/macrostructure had the strongest influence on performance for the different model architectures with improvements ranging from 54% for the base DNN model, 69.3% for the deep CNN model, and 72.3% for the DQN model to 75.2% for the PPO model. Previous studies have similarly found that hyperparameter optimization was significantly responsible for better model performance [45] [46] [47]. On the subquestions of *how varying the agent's task and environment affects its generality behavior*, our findings indicate that while tasks and environments can be varied to make the agent more generalizable, this effect can be attributed to the underlying network [48] and the net effect of aligning a suitable task and environment with an appropriate macro/microstructure since building the right architecture should ultimately solve any given task and environment. This further indicates that the macro/microstructure is the most effective factor in making artificial agents more generalizable.

Intuitively, this suggests that internal autonomy to implement algorithmic flexibility is necessary for a truly versatile AGI agent. For any assigned task, we can manually vary the macro/microstructure of the artificial agent to solve the task. Similarly, for any environment in which an agent's task is assigned, the agent's structure can be altered to suit that environment. Hence, indicating that the agent's autonomy to self-vary its macro/microstructure (algorithmic flexibility) is essential for a true AGI agent. This idea is analogous to the generality of human intelligence, where, for example, humans never truly stop learning, and humans can autonomously choose how to adapt their knowledge to any task or situation [57] [58].

The RL theory has deep normative roots in the psychological and neuroscientific perspectives of animal behavior and how such biological agents can potentially optimize the control of their environment [17]. With the generality of human intelligence as the ground truth for the AGI theoretical concept [2] [3], studies show that through the cognitive process known as algorithmic flexibility, humans appear not to solve tasks with a single algorithm, but rather opportunistically switch their search procedures when faced with predicaments [57] [58]. This implies that humans accumulate various algorithms for solving a variety of simple and complex tasks that lead to some expected reward, and that they autonomously choose which algorithm to implement based on experience, by transferring prior knowledge from other domains, or through trial and error [58]. Hence, much like their human intelligence ground truth [34] [52], AI agents in the RL-based theory also attempt to maximize their expected reward by learning what actions to take and how to map their environment's states to their actions based on the environment's feedback. This is achieved through self-learning from experience, trial and error, and transfer learning when implemented [16].

However, in terms of applying its knowledge across multiple domains to address a diverse collection of challenging tasks, current model architectures equip the agent with a single algorithm rather than a suite of algorithms like its human ground truth [34] [52], and possibly with some prior knowledge through transfer learning. We, therefore, argue that AI generality can be enhanced with internal autonomy and ensemble methods that provide agents with a suite of algorithms and the internal autonomy to perform model selection among candidate models, akin to human intelligence. Hence, it stands to reason that the generalizability of AGI agents can be enhanced with access to a suite of algorithms and internal autonomy. This may be described as an AI agent with the capability to alter its structure optimally through algorithmic flexibility [57] [58]. While the subfield of Automated Machine Learning (AutoML) can provide tools that create high-level abstractions to help practitioners develop pipelines that streamline ML algorithms to different applications, such as setting hyperparameters and automating data preprocessing to expose the underlying structure of the task to the learning model, this current automatic ML effort is directed externally toward reducing human workflows and not internally toward the model's generality [31]. Additionally, the automatic ML effort primarily addresses classical ML and not the area of RL models.

Finally, our study highlights the importance of algorithmic flexibility and internal autonomy in achieving truly generalizable AGI agents. By incorporating a suite of algorithms and allowing agents to autonomously adapt their macro/micro-structures to suit their tasks and environments, we can move closer to the generality exhibited by humans. This would entail creating a more comprehensive and dynamic framework for AI agents that allows them to self-learn and adapt, similar to human intelligence. Such an approach would promote the development of more generalizable AI agents capable of addressing a wide array of complex tasks across various domains.

5. Conclusions

On the main research question of *how the general-purpose learning agent approach can lead to more generalizable artificialagents*, our findings indicate that DRL-based AI agents can be made more generalizable by varying their macro/microstructures to suit their task and environment. This suggests that an intelligent agent's *internalautonomy* to self-alter its macro/microstructure through *algorithmic flexibility* is an essential component for a true AGI agent. Hence, more studies are required to understand how the general-purpose learning agent can autonomously vary its macro/microstructure through internal autonomy and ensemble learning using a suite of algorithms for model selection, and how this approach can lead to more generalizable artificialagents irrespective of the assigned task or environment.

On the subquestion of *how varying the agent's micro/macrostructure affects its generality behavior*, varying the agent's macrostructure had the strongest influence on performance for all the model architectures. Similarly, varying the microstructure of the models resulted in prediction accuracy improvement. Thus, varying the micro/macrostructure of the DRL-based models resulted in overall prediction accuracy improvement. This is in line with previous studies, which similarly found that hyperparameter optimization was significantly responsible for better model performance. On the sub questions of *how varying the agent's task and environment affects its generality behavior*, while the findings also show that the tasks and environment can be varied to make the agent more generalizable, this effect can be attributed to the underlying network [48] and the net effect of aligning a suitable task and environment with a suitable macro/microstructure since building the right architecture should ultimately solve any given task and environment. This further indicates that the macro/micro-structure is most effective at making artificial agents more generalizable.

In light of the theories, previous studies, and our findings, artificial agents should, in principle, be capable of addressing any given task and environment provided they possess an appropriate macro/microstructure. Although our agents solved their task and environments through manual adaptation or modification of their macro/microstructure, our observations suggest that for any arbitrary task and environment, an agent should be able to solve it if its macro/microstructure aligns with that task and environment. Consequently, instead of relying on human intervention to manually adjust the macro/microstructure through problem-specific design and tuning, internal autonomy emerges as a crucial component for truly versatile AGI agents to demonstrate algorithmic flexibility and autonomously modify their macro/microstructure.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Aamodt, A. and Plaza, E. (2017) Case-Based Reasoning and the Upswing of AI. https://www.iiia.csic.es/~enric/papers/Keynote-AamodtPlaza.pdf

[2] Kadam, S. and Vaidya, V. (2021) Cognitive Evaluation of Machine Learning Agents. *Cognitive Systems Research*, **66**, 100-121. https://doi.org/10.1016/j.cogsys.2020.11.003

[3] Gobble, M.A.M. (2019) The Road to Artificial General Intelligence. *Research Technology Management*, **62**, 55-59. https://doi.org/10.1080/08956308.2019.1587336

[4] Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J.P., Jaderberg, M., Silver, D., *et al.* (2019) Grandmaster Level in StarCraft II Using Multi-Agent Reinforcement Learning. *Nature*, **575**, 350-354. https://doi.org/10.1038/s41586-019-1724-z

[5] Pei, J., Deng, L., Song, S., Zhao, M., Zhang, Y., Wu, S., Wang, G., Zou, Z., Wu, Z., He, W., Chen, F., Deng, N., Wu, S., Wang, Y., Wu, Y., Yang, Z., Ma, C., Li, G., Han, W., Shi, L., *et al.* (2019) Towards Artificial General Intelligence with Hybrid Tianjic Chip Architecture. *Nature*, **572**, 106-111. https://doi.org/10.1038/s41586-019-1424-8

[6] Hassabis, D. (2018) DeepMind-Learning from First Principles-Artificial Intelligence NIPS. https://www.youtube.com/watch?v=DXNqYSNvnjA&list=RDCMUC5g-f-g4EVRkqL8Xs888BLA&index=6

[7] Dalgaard, M., Motzoi, F., Sørensen, J.J. and Sherson, J. (2020) Global Optimization of Quantum Dynamics with *α*Zero Deep Exploration. *NPJ Quantum Information*, **6**, Article No. 6. https://doi.org/10.1038/s41534-019-0241-0

[8] Hsueh, C.H., Wu, I.C., Chen, J.C. and Hsu, T.S. (2018) *α*Zero for a Non-Deterministic Game. 2018 *Conference on Technologies and Applications of Artificial Intelligence* (*TAAI*), Taichung, 30 November-2 December 2018, 116-121. https://doi.org/10.1109/TAAI.2018.00034

[9] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S.A.A., Ballard, A.J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Hassabis, D., *et al.* (2021) Highly Accurate Protein Structure Prediction with *α*Fold. *Nature*, **596**, 583-589. https://doi.org/10.1038/s41586-021-03819-2

[10] Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T. and Silver, D. (2020) Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model. *Nature*, **588**, 604-609. https://doi.org/10.1038/s41586-020-03051-4

[11] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K. and Hassabis, D. (2018) A General Reinforcement Learning Algorithm That Masters Chess, Shogi, and Go through Self-Play. *Science*, **362**, 1140-1144. https://doi.org/10.1126/science.aar6404

[12] Tunyasuvunakool, K., Adler, J., Wu, Z., Green, T., Zielinski, M., Žídek, A., *et al.* (2021) Highly Accurate Protein Structure Prediction for the Human Proteome. *Nature*, **596**, 590-596. https://doi.org/10.1038/s41586-021-03828-1

[13] Goertzel, B. and Pennachin, C. (2014) Artificial General Intelligence: Concept, State of the Art, and Future Prospects. *Journal of Artificial General Intelligence*, **5**, 1-48. https://doi.org/10.2478/jagi-2014-0001

[14] Jonsson, A. (2019) Deep Reinforcement Learning in Medicine. *Kidney Diseases*, **5**, 18-22. https://doi.org/10.1159/000492670

[15] Fridman, L. (2019) MIT 6.S091: Introduction to Deep Reinforcement Learning (Deep RL). https://www.youtube.com/watch?v=zR11FLZ-O9M&list=RDLV5tvmMX8r_OM&index=27

[16] Sutton, R. and Barto, A. (2018) Reinforcement Learning: An Introduction. 2nd Edition, MIT Press, Cambridge.

[17] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. and Hassabis, D. (2015) Human-Level Control through Deep Reinforcement Learning. *Nature*, **518**, 529-533. https://doi.org/10.1038/nature14236

[18] Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C. and Liu, C. (2018) A Survey on Deep Transfer Learning. In: Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L. and Maglogiannis, I., Eds., *ICANN* 2018: *Artificial Neural Networks and Machine Learning—ICANN* 2018, Springer, Cham, 270-279. https://doi.org/10.1007/978-3-030-01424-7_27

[19] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H. and He, Q. (2021) A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, **109**, 43-76. https://doi.org/10.1109/JPROC.2020.3004555

[20] Zhou, S.K., Le, H.N., Luu, K., Nguyen, H.V. and Ayache, N. (2021) Deep Reinforcement Learning in Medical Imaging: A Literature Review. *Medical Image Analysis*, **73**, Article ID: 102193. https://deepai.org/publication/deep-reinforcement-learning-in-medical-imaging-a-literature-review https://doi.org/10.1016/j.media.2021.102193

[21] Kamiri, J. and Mariga, G. (2021) Research Methods in Machine Learning: A Content Analysis. *International Journal of Computer and Information Technology*, **10**. https://doi.org/10.24203/ijcit.v10i2.79

[22] Roberts, C. and Hyatt, L. (2018) A Practical and Comprehensive Guide to Planning, Writing, and Defending Your Dissertation. 3rd Edition, Corwin Press, Thousand Oaks.

[23] Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S. and Dean, J. (2019) A Guide to Deep Learning in Healthcare. *Nature Medicine*, **25**, 24-29. https://doi.org/10.1038/s41591-018-0316-z

[24] Mousavi, S., Schukat, M. and Howley, E. (2018) Deep Reinforcement Learning: An Overview. In: Bi, Y., Kapoor, S. and Bhatia, R., Eds., *IntelliSys* 2016: *Proceedings of SAI Intelligent Systems Conference* (*IntelliSys*) 2016, Springer, Cham, 426-440. https://doi.org/10.1007/978-3-319-56991-8_32

[25] Dai, Y., Wang, G., Muhammad, K. and Liu, S. (2020) A Closed-Loop Healthcare Processing Approach Based on Deep Reinforcement Learning. *Multimedia Tools and Applications*, **81**, 3107-3129.

[26] Ker, J., Wang, L., Rao, J. and Lim, T. (2017) Deep Learning Applications in Medical Image Analysis. *IEEE Access*, **6**, 9375-9379. https://doi.org/10.1109/ACCESS.2017.2788044

[27] Kohli, M.D., Summers, R.M. and Geis, J.R. (2017) Medical Image Data and Datasets in the Era of Machine Learning—Whitepaper from the 2016 C-MIMI Meeting Dataset Session. *Journal of Digital Imaging*, **30**, 392-399.

https://doi.org/10.1007/s10278-017-9976-3

[28] Oakden-Rayner, L. (2020) Exploring Large-Scale Public Medical Image Datasets. *Academic Radiology*, **27**, 106-112. https://doi.org/10.1016/j.acra.2019.10.006

[29] Xu, Y. and Goodacre, R. (2018) On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *Journal of Analysis and Testing*, **2**, 249-262. https://doi.org/10.1007/s41664-018-0068-2

[30] Zhang, W., Panum, T.K., Jha, S., Chalasani, P. and Page, D. (2020) Transfer Learning via Learning to Transfer. *Proceedings of the* 37*th International Conference on Machine Learning*, 13-18 July 2020, 11171-11181.

[31] Xin, D., Wu, E.Y., Lee, D.J.L., Salehi, N. and Parameswaran, A. (2021) Whither AutoML? Understanding the Role of Automation in Machine Learning Workflows. *Proceedings of the* 2021 *CHI Conference on Human Factors in Computing System*s, Yokohama, 8-13 May 2021, 1-16. https://doi.org/10.1145/3411764.3445306

[32] GE Healthcare (2018) Beyond Imaging: The Paradox of AI and Medical Imaging Innovation. https://twitter.com/GEHealthCare/status/1063079423832842240

[33] Ramamoorthy, A. and Yampolskiy, R. (2018) Beyond MAD: The Race for Artificial General Intelligence. *ICT Discoveries*, 1-8. https://www.semanticscholar.org/paper/BEYOND-MAD-%3A-THE-RACE-FOR-ARTIFICIAL-GENERAL-Ramamoorthy-Yampolskiy/7371bb45f85d297fbad25dee15a6b7f089cd60df

[34] Lake, B.M., Ullman, T.D., Tenenbaum, J.B. and Gershman, S.J. (2017) Building Machines That Learn and Think Like People. *Behavioral and Brain Sciences*, **40**, E253. https://doi.org/10.1017/S0140525X16001837

[35] Saripalli, V.R. (2019) Scalable and Data Efficient Deep Reinforcement Learning Methods for Healthcare Applications. Master's Thesis, Colorado State University, Fort Collins. https://aspenuniversity.idm.oclc.org/login?url=https://www.proquest.com/dissertations-theses/scalable-data-efficient-deep-reinforcement/docview/2349665414/se-2?accountid=34574

[36] Brink, H., Richards, J. and Fetherolf, M. (2016) Real-World Machine Learning. Manning Publications, New York.

[37] Draelos, R. (2019) Best Use of Train/Val/Test Splits, with Tips for Medical Data. https://glassboxmedicine.com/2019/09/15/best-use-of-train-val-test-splits-with-tips-for-medical-data/

[38] Cho, H., Kim, Y., Lee, E., Choi, D., Lee, Y. and Rhee, W. (2020) Basic Enhancement Strategies When Using Bayesian Optimization for Hyperparameter Tuning of Deep Neural Networks. *IEEE Access*, **8**, 52588-52608. https://doi.org/10.1109/ACCESS.2020.2981072

[39] Claesen, M., Simm, J., Popovic, D. and De Moor, B.L.R. (2014) Hyperparameter Tuning in Python Using Optunity. *Proceedings of the International Workshop on Technical Computing for Machine Learning and Mathematical Engineering*, 6-7. https://homes.esat.kuleuven.be/~claesenm/optunity/varia/abstract-tcmm2014.pdf

[40] Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M. and de Freitas, N. (2015) Dueling Network Architectures for Deep Reinforcement Learning. arXiv: 1511.06581. http://arxiv.org/abs/1511.06581

[41] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. (2017) Proximal Policy Optimization Algorithms. arXiv: 1707.06347. http://arxiv.org/abs/1707.06347

[42] Keras (2021) Simple MNIST Convnet. https://keras.io/examples/vision/mnist_convnet/

[43] Simonyan, K. and Zisserman, A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv: 1409.1556.

[44] Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y. and Zhokhov, P. (2017) OpenAI Baselines. GitHub Repository. https://github.com/openai/baselines

[45] Elgeldawi, E., Sayed, A., Galal, A.R. and Zaki, A.M. (2021) Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis. *Informatics*, **8**, Article 79. https://doi.org/10.3390/informatics8040079

[46] Hoque, K.E. and Aljamaan, H. (2021) Impact of Hyperparameter Tuning on Machine Learning Models in Stock Price Forecasting. *IEEE Access*, **9**, 163815-163830. https://doi.org/10.1109/ACCESS.2021.3134138

[47] Wong, J., Manderson, T., Abrahamowicz, M., Buckeridge, D.L. and Tamblyn, R. (2019) Can Hyperparameter Tuning Improve the Performance of a Super Learner? A Case Study. *Epidemiology*, **30**, 521-531. https://doi.org/10.1097/EDE.0000000000001027

[48] Thompson, J., Bengio, Y. and Schoenwiesner, M. (2019) The Effect of Task and Training on Intermediate Representations in Convolutional Neural Networks Revealed with Modified RV Similarity Analysis. 2019 *Conference on Cognitive Computational Neuroscience*, Berlin, 13-16 September 2019. https://doi.org/10.32470/CCN.2019.1300-0

[49] Hashemi, M. (2019) Enlarging Smaller Images before Inputting into Convolutional Neural Network: Zero-Padding vs. Interpolation. *Journal of Big Data*, **6**, Article No. 98. https://doi.org/10.1186/s40537-019-0263-7

[50] Weidele, D.K.I., Weisz, J.D., Oduor, E., Muller, M., Andres, J., Gray, A. and Wang, D. (2019) AutoAIViz: Opening the Blackbox of Automated Artificial Intelligence with Conditional Parallel Coordinates. *Proceedings of the 25th International Conference on Intelligent User Interfaces*, Cagliari, 17-20 March 2020, 308-312. https://doi.org/10.1145/3377325.3377538

[51] Callaway, E. (2020) "It Will Change Everything": DeepMind's AI Makes Gigantic Leap in Solving Protein Structures. *Nature*, **588**, 203-204. https://doi.org/10.1038/d41586-020-03348-4

[52] Li, J., Zhu, G., Hua, C., Feng, M., Basheer-Bennamoun, Li, P., Lu, X., Song, J., Shen, P., Xu, X., Mei, L., Zhang, L., Shah, S.A.A. and Bennamoun, M. (2021) A Systematic Collection of Medical Image Datasets for Deep Learning. arXiv: 2106.12864. http://arxiv.org/abs/2106.12864

[53] Atkinson, C., McCane, B., Szymanski, L. and Robins, A. (2018) Pseudo-Rehearsal: Achieving Deep Reinforcement Learning without Catastrophic Forgetting. *Neurocomputing*, **428**, 291-307. https://doi.org/10.1016/j.neucom.2020.11.050

[54] Kaushik, P., Gain, A., Kortylewski, A. and Yuille, A. (2021) Understanding Catastrophic Forgetting and Remembering in Continual Learning with Optimal Relevance Mapping. arXiv: 2102.11343.

[55] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D. and Hadsell, R. (2017) Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the National Academy of Sciences of the United States of America*, **114**, 3521-3526. https://doi.org/10.1073/pnas.1611835114

[56] Ha, D. (2019) Reinforcement Learning for Improving Agent Design. *Artificial Life*,

25, 352-365. https://doi.org/10.1162/artl_a_00301

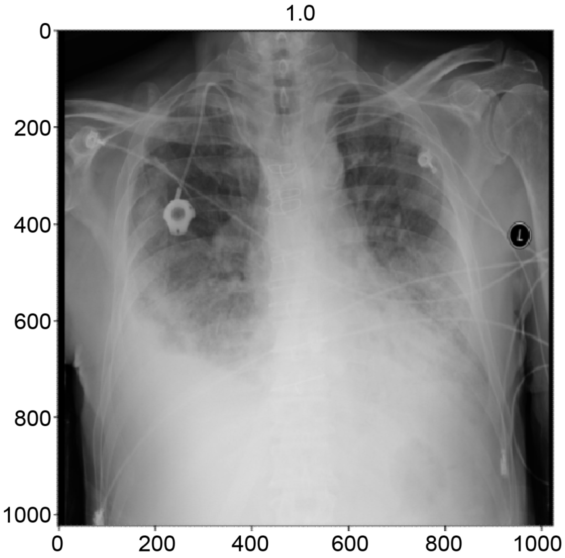
[57] Murawski, C. and Bossaerts, P. (2016) How Humans Solve Complex Problems: The Case of the Knapsack Problem. *Scientific Reports*, **6**, Article No. 34851. https://doi.org/10.1038/srep34851

[58] Wang, Y. and Chiew, V. (2010). On the Cognitive Process of Human Problem Solving. *Cognitive Systems Research*, **11**, 81-92. https://doi.org/10.1016/j.cogsys.2008.08.003

Appendix A: Data Analysis

Sample of Data

An image from National Institute of Health dataset showing a human chest X-ray.



Data Collection

The medical image datasets were downloaded from the following links below at the National Institute of Health (NIH) open-access medical image dataset libraries available free online for ML research at:

https://nihcc.app.box.com/v/ChestXray-NIHCC

Appendix B: Algorithms

Deep Q-Network and Proximal Policy Optimization Algorithms.

Algorithm 1: deep Q-learning with experience replay [36].

```
Initialize replay memory $D$ to capacity $N$
Initialize action-value function $Q$ with random weights $\theta$
Initialize target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$
**For** episode = 1, $M$ **do**
  Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequence $\phi_1 = \phi(s_1)$
  **For** $t = 1$,T **do**
     With probability $\varepsilon$ select a random action $a_t$
     otherwise select $a_t = \text{argmax}_a Q(\phi(s_t), a; \theta)$
     Execute action $a_t$ in emulator and observe reward $r_t$ and image $x_{t+1}$
     Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
     Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $D$
     Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $D$
     Set $y_j = \begin{cases} r_j & \text{if episode terminates at step } j+1 \\ r_j + \gamma \max_{a'} \hat{Q}(\phi_{j+1}, a'; \theta^{-}) & \text{otherwise} \end{cases}$
     Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ with respect to the
     network parameters $\theta$
     Every $C$ steps reset $\hat{Q} = Q$
  **End For**
**End For**
```

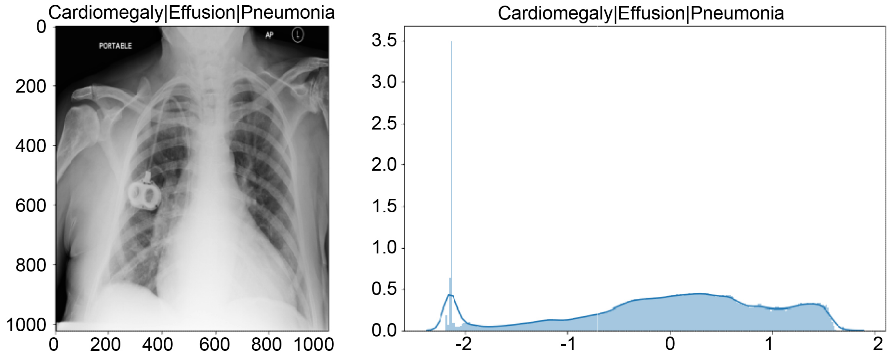
Algorithm 2: PPO, Actor-Critic Style [47].

```
for iteration=1, 2, ... do
    for actor=1, 2, ..., $N$ do
        Run policy $\pi_{\theta_{\text{old}}}$ in environment for $T$ timesteps
        Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
    end for
    Optimize surrogate $L$ wrt $\theta$, with $K$ epochs and minibatch size $M \leq NT$
    $\theta_{\text{old}} \leftarrow \theta$
end for
```

Summary of the PPO Algorithm

1. First, collect some trajectories based on some policy π_θ, and initialize theta prime $\theta' = \theta$
2. Next, compute the gradient of the clipped surrogate function using the trajectories
3. Update θ' using gradient ascent $\theta' \leftarrow \theta' + \alpha\nabla_{\theta'} L^{\text{clip}}_{\text{sur}}(\theta', \theta)$
4. Then we repeat step 2-3 without generating new trajectories. Typically, step 2-3 are only repeated a few times
5. Set $\theta = \theta'$, go back to step 1, repeat.

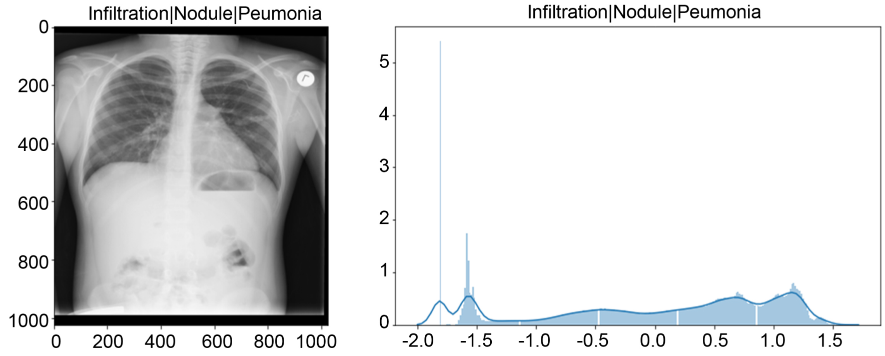
Appendix C: Additional Figures



Mean value: −1.0408340855860843e−16.

Standard deviation: 1.0000000000000004.

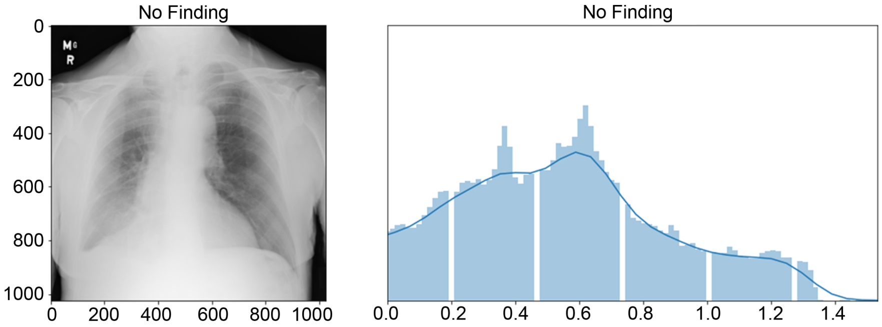
Figure C1. Sample pixel intensity distribution for Cardiomegaly, Effusion, and Pneumonia comorbidity.



Mean value: −1.1058862159352145e−16

Standard deviation: 1.0000000000000013.

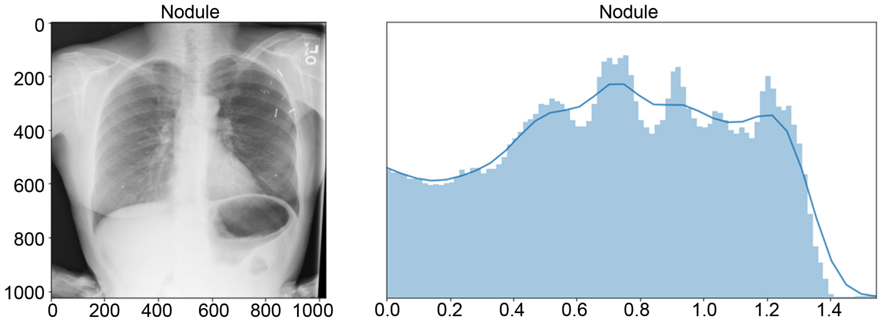
Figure C2. Sample pixel intensity distribution for infiltration, nodule, and pneumonia.



Mean value: 6.938893903907228e−18

Standard deviation: 0.9999999999999993.

Figure C3. The pixel intensity distribution for a "No Finding" sample.



Mean value: 6.5052130349130266e−18

Standard deviation: 1.0000000000000004

Figure C4. The pixel intensity distribution for a Nodule sample.

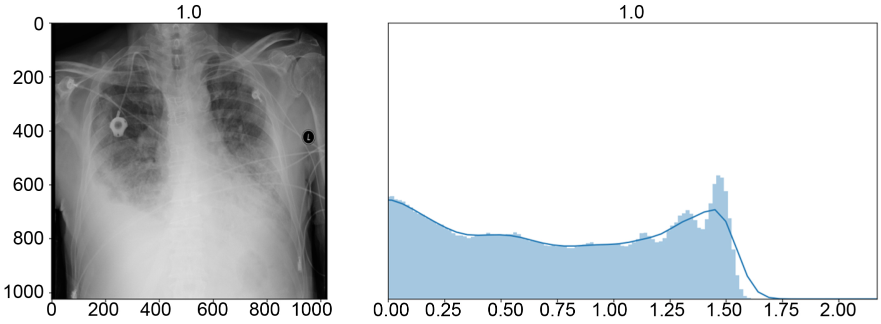


Figure C5. The pixel intensity distribution for a Pneumonia sample.

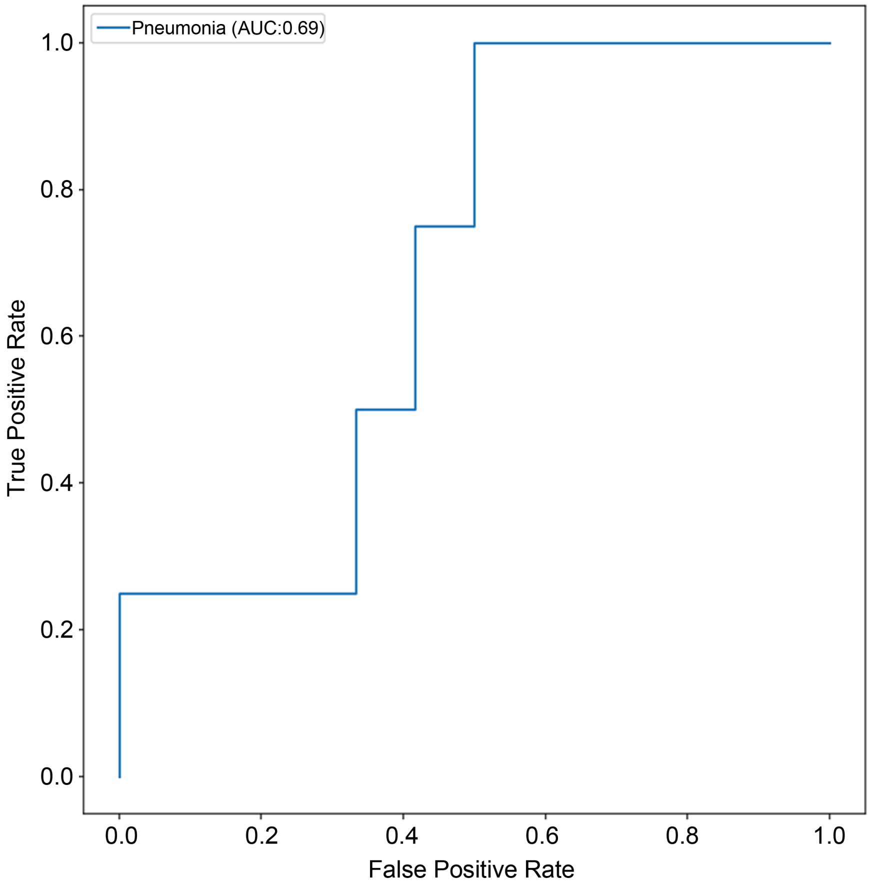


Figure C6. Sample receiver operating characteristic curve showing the Area under the Curve (AUC).

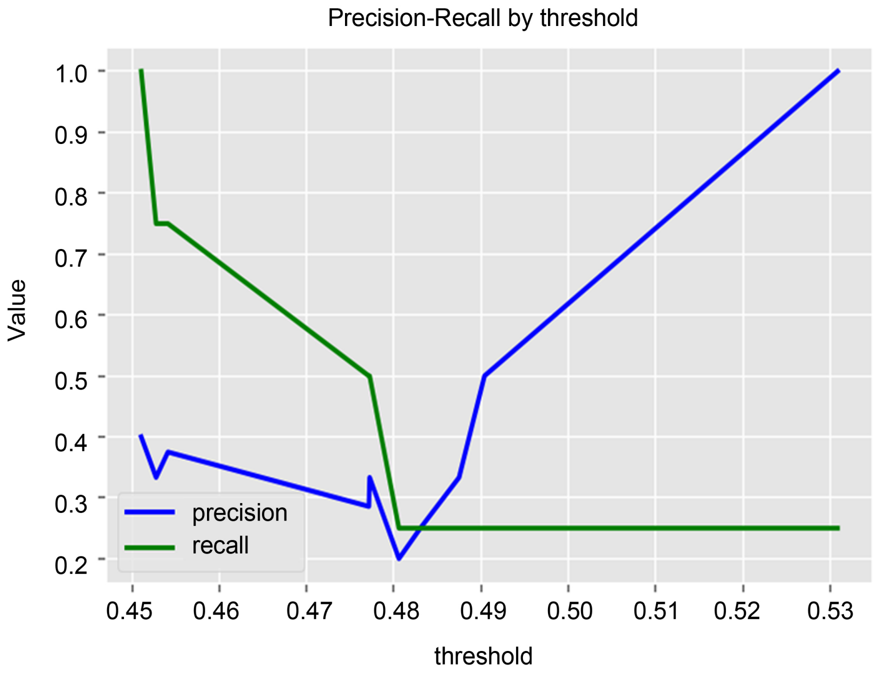


Figure C7. Sample precision and recall plotted against the threshold.

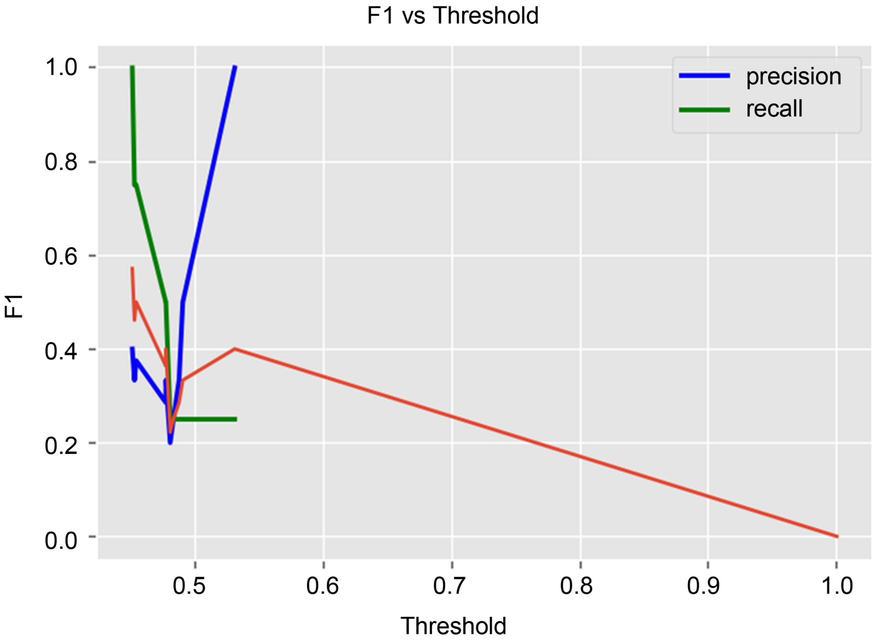


Figure C8. Sample F1 plotted against the threshold.

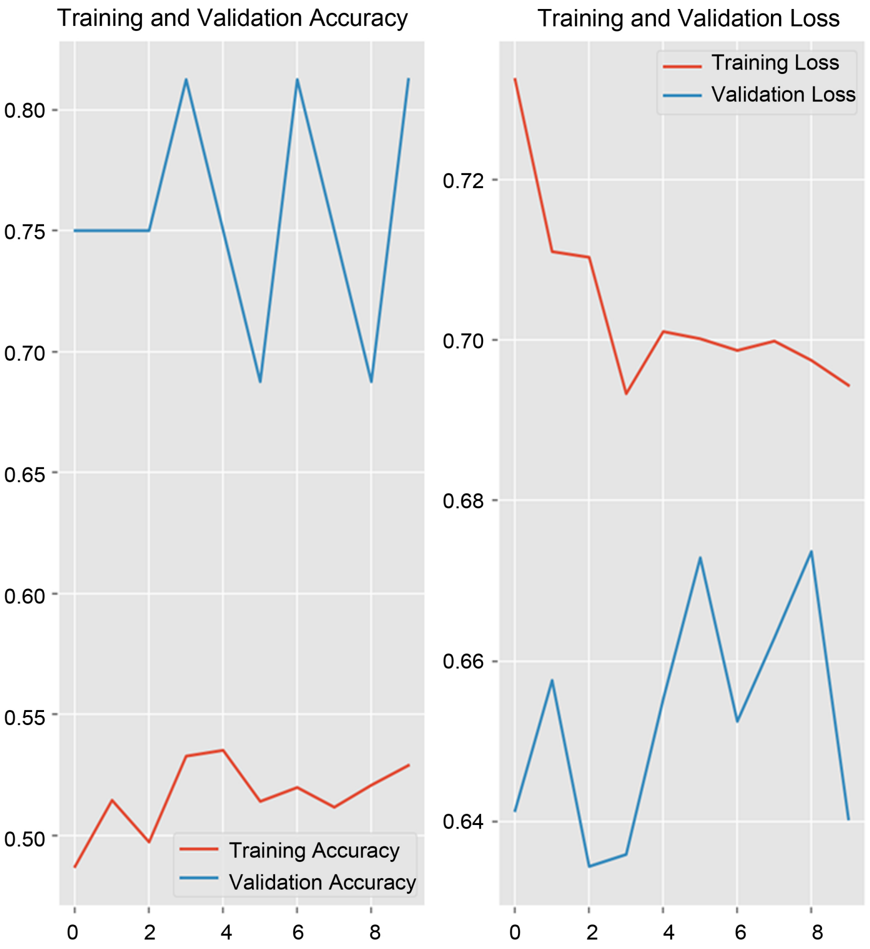


Figure C9. Sample of model's training and validation loss vs. accuracy.

Table C1. Summary of the means of prediction accuracy performance scores in twelve trials between binary and multi-class classification tasks.

Trial	Models under different settings	Binary Task	Binary Structure	Binary Environment	Multi-class Task	Multi-class Structure	Multi-class Environment
1	Keras Model	52	56	47	47	52	41
2	Keras Model	53	54	51	48	53	43
9	Keras Model	53	56	49	50	53	47
2	CNN Model	64	67	61	65	71	59
6	CNN Model	67	69	64	62	68	56
10	CNN Model	66	74	61	65	67	62
3	PPO Model	68	78	57	66	74	57
7	PPO Model	71	76	66	67	69	64
11	PPO Model	74	78	69	69	76	61
4	DQN Model	66	70	62	66	70	62
8	DQN Model	70	73	67	68	73	62
12	DQN Model	69	75	63	67	73	66

For each trial, we experimented with three different settings for each model and took the mean scores. The model's task performance was obtained by the combined mean scores for the structure and environment in that trial.