Research Article

Advances in Machine Learning & Artificial Intelligence

A Novel Approach to Adopt Explainable Artificial Intelligence in X-ray Image Classification

Shaw-Hwa Lo* **and Yiqiao Yin**

Department of Statistics, Columbia University, New York, NY 10027, USA

*Corresponding author
Shaw-Hwa Lo, Department of Statistics, Columbia University, New York, NY 10027, USA

Submitted: 16 Dec 2021; **Accepted**: 27 Dec 2021; **Published**: 25 Jan 2022

Citation: *Shaw-Hwa Lo and Yiqiao Yin (2022) A Novel Approach to Adopt Explainable Artificial Intelligence in X-ray Image Classification. Adv Mach Lear Art Inte, 3(1): 01-11.*

Abstract
Robust "Blackbox" algorithms such as Convolutional Neural Networks (CNNs) are known for making high prediction performance. However, the ability to explain and interpret these algorithms still require innovation in the understanding of influential and, more importantly, explainable features that directly or indirectly impact the performance of predictivity. In view of the above needs, this study proposes an interaction- based methodology – Influence Score (I-score) – to screen out the noisy and non-informative variables in the images hence it nourishes an environment with explainable and interpretable features that are directly associated to feature predictivity. We apply the proposed method on a real-world application in Pneumonia Chest X-ray Image data set and produced state- of-the-art results. We demonstrate how to apply the proposed approach for more general big data problems by improving the explain ability and interpretability without sacrificing the prediction performance. The contribution of this paper opens a novel angle that moves the community closer to the future pipelines of XAI problems.

Introduction
Many successful achievements in machine learning and deep learning have accelerated real-world implementation of Artificial Intelligence (AI). This issue has been greatly acknowledged by the Department of Defense (DoD) [1]. DARPA 2016 initiated the explainable Artificial Intelligence (XAI) challenge and brought this new interest to the surface [1]. In addressing the concepts of interpretability and explainability, these scholars and researchers have made at- tempts towards discussing a trade-off between learning performance (usually measured by prediction performance) and effectiveness of explanations (also known as explainability) is presented in Figure 1 [2, 3]. This trade-off often occurs in any supervised machine learning problems that aim to use explanatory variable to predict response variable (or outcome variable) which happens between learning performance (also known as prediction performance) and effectiveness of explanations (also known as explainability) [4]. The proposed I-score, if implemented, can raise prediction performance as well as establishing explainability in the any neural network architecture.

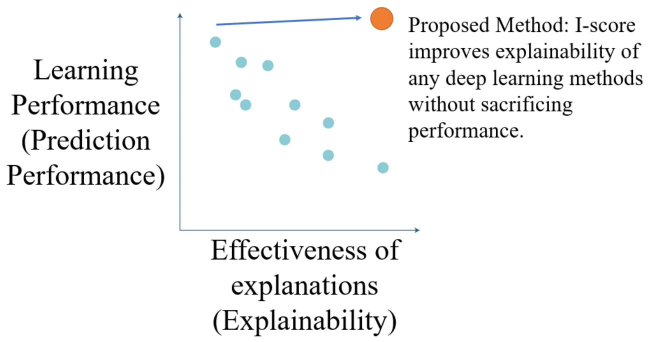


Figure 1: This diagram is a recreation DARPA document (DARPA-BAA-16-53) [1, 5]. The diagram presents the relationship between learning performance (usually measured by prediction performance) and effectiveness of explanations (also known as explain ability).

Definition of Feature Explainability
A popular description of interpretability is by which defined XAI as the ability to explain or to present in understandable terms to a human [6]. Another popular version states interpretability as the degree to which a human can understand the cause of a

decision [7]. Though intuitive, these definitions lack mathematical formality and rigorousness [4]. Moreover, it is yet unclear why variables provide us the good prediction performance and, more importantly, how to yield a relatively unbiased estimate of a parameter that is not sensitive to noisy variables and is related to the parameter of interest.

Three necessary conditions (C1, C2 and C3) for any feature to shed light to these questions, we define the following selection methodology to be explainable and interpretable.

- **C1.** The first condition states that the feature selection methodology does not require the knowledge of the underlying model of how explanatory variables affects out- come variable.
- **C2.** An explainable and interpretable feature selection method must clearly state to what degree a combination of explanatory variables influences the response variable. Moreover, it is beneficial if a statistician can directly compute a score for a set of variables in order to make reasonable comparisons.
- **C3.** In order for a feature assessment and selection technique to be interpretable and explainable, it must directly associate with the predictivity of the explanatory variables (for definition of predictivity [8, 9].

There is extensive evidence for the importance of explanation towards understanding and building trust in cognitive psychology, philosophy, and machine learning research [10-14]. Existing compute importance for a given base model (the one being explained) and an output category [15-21]. However, they require access to the internals of the base model, such as the gradients of the output. LIME offers such a black-box approach by drawing random samples around the instance to be explained and fitting an approximate linear decision model [22]. However, its saliency is based on super pixels, which may not capture correct regions [23]. Other techniques such as Random Input Sampling for Explanation or RISE produce importance region by using random masking and computation of Area-Under-Curve (AUC) value [23]. However, Area-Under-Curve, which this proposal shows later, is not very sensitive at detecting highly predictive variables and features. Under incorrect model specification, AUC can subject to severe negative impact. We summarize whether the conventional explainable methods satisfy the three explainability conditions in Table 1.

Table 1: Explainability Satisfaction Table. The table summarizes whether famous XAI methods and proposed I-score satisfy the definition of Explainability of a set of variables according to definition in Innovation 2.

Definition of Explainability	CAM	LIME	RISE	I-score
$C1$ Non-parametric	No	No	No	Yes
$C2$ Quantifiable Measure	No	Yes	Yes	Yes
$C3$ Predictivity	No	No	No	Yes

Proposed Method

The proposed methodology comes with three stages. First, we investigate variables to identify those with high potential to form influential modules. Secondly, we generate highly influential variable modules from variables selected in the first stage, where variables of the same module interact with each other to produce a strong effect on Y. Last, we combine the variable modules to carry out prediction process.

Influence Score (I-score)

The Influence Score (I-score) is a statistic derived from the partition retention method [24]. Consider a set of n observations of an outcome variable (or response variable) Y and a large number S of explanatory variables, X_1, X_2, ... X_S. Randomly select a small group, m, of the explanatory variables X's. We can denote this subset of variables to be X $\{X_k$, k = 1, ...,m$\}$. We suppose X_k takes values of only 1 and 0 (though the variables are binary in this discussion, it can be generalized into continuous variables [25, 26]. Hence, there are 2m possible partitions for X's. The n observations are partitioned into 2m cells according to the values of the m explanatory variables. We refer to this partition as Π_X. The proposed I-score (denoted by $I_{\Pi X}$) is defined in the following

$$I_{\Pi_X} = \frac{1}{ns_n^2} \sum_{j=1}^{2^m} n_j^2 (\bar{Y}_j - \bar{Y})^2 \qquad (1)$$

While insert eq. We notice that the I-score is designed to capture the discrepancy between the conditional means of Y on X_1, X_2, ..., X_m and the mean of Y.

The statistics I is the summation of squared deviations of frequency of Y from what is expected under the null hypothesis. There are two properties associated with the statistics I. First, the measure I is non-parametric which requires no need to specify a model for the joint effect of $\{X_{b1}$, ..., $X_{bk}\}$ on Y . This measure I is created to describe the discrepancy between the conditional means of Y on $\{X_{b1}$, ..., $X_{bk}\}$ disregard the form of conditional distribution. With each variable to be dichotomous, the variable set $\{X_{b1}$, ..., $X_{bk}\}$ form a well-defined partition, P (Chernoff, Lo, and Zheng 2009). Secondly, under the null hypothesis that the subset has no influence on Y, the expectation of I remains non-increasing when dropping variables from the subset. The second property makes I fundamentally different from the Pearson's χ^2 statistic whose expectation is dependent on the degree of freedom and hence on the number of variables selected to define the partition. The authors demonstrated in simulation that the value of I-score is a function of sensitivity and specificity, which can translate to the value of Area-Under-Curve (a famous metric researchers refer to when examining the classifiers) [25, 26].

Backward Dropping Algorithm (BDA)

The Backward Dropping Algorithm is a greedy algorithm to search for the optimal subsets of variables that maximizes the I-score through step-wise elimination of variables from an initial subset sampled in some way from the variable space. The steps of the algorithm are presented in Algorithm 1.

The proposed BDA (Algorithm 1) presents a systematic way of searching for important and explainable features. The following small example illustrates the usage of I-score and BDA. Suppose there are $X_i \sim$ Bernoulli (1/2)

Algorithm 1: Procedure of the Backward Dropping Algorithm (BDA)

Data: Training set $\{(y_1, x_1), ..., (y_n, x_n)\}$ of n observations, where $x_i = (x_{1i}, ..., x_{pi})$ is a p-dimensional vector of explanatory variables. Typically p is very large. All explanatory variables are discrete.
Initialize: Select an initial subset of k explanatory variables $X_b = \{X_{b_1}, ..., X_{b_k}\}$, $b = 1, ..., B$. The notation b indicates which rounds of BDA it is executing. In practice, we recommend B to be a large number. Then compute I-score: $I(X_b) = \sum_{j \in \Pi_{X_b}} n_j^2(\bar{Y}_j - \bar{Y})^2$. Set X_c to be X_b as the current set.

Run:

```
1: while ||Xc|| > 1 do
2: tentatively drop each variable in Xc.
3: recalculate the I-score with one variable less.
4: then drop the one that gives the highest I-score and
call this new subset $X'_b$, which has one variable less than Xc.
5: store this subset of variable and its corresponding Iscore
in a list for future reference.
6: reset $X_c$ to be $X'_b$.
7: end while
8: return The subset that yields the highest I-score in the
whole dropping process.
```

Table 2: Demonstration of BDA. The subset of variables left with the highest I-score value is $\{X_1, X_2\}$.

I-score	**13.10**	**25.90**	**1.05**
Variables left	X_1, X_2, X_3	X_1, X_2	X_2
Investigate	Drop X_1, I = 0.94 Drop X_2, I = 0.55 Drop X_3, I = 25.90	Drop X_1, I = 1.31 Drop X_2, I = 1.05	
Drop	X_3	X_1	

while $i = \{1, 2, 3\}$. Assume there is a toy model $Y = X_1 + X_2$ (mod 2). In other words, the variable X_3 would be a noisy variable because it does not contribute to the definition of Y. In addition, suppose we generate 100 samples for X_i and Y. BDA can be used to screen out X_3. We present the steps of BDA in Table 2.

Feature Engineering: Dagger Technique

The concept of interaction-based Feature is initially pro-posed in Lo and Yin (2021) [25, 26]. In their work, the authors defined an interaction-based feature that is used to replace the construction of using filters in designing Convolutional Neural Networks (CNNs). The conventional practice relies on pre-defined filters and these filters are small 2-by-2 or 3-by-3 window that are designed to capture certain information based on prior knowledge. The art of using interaction-based feature to create novel features within a 2-by-2 or 3-by-3 window in an image is to allow the data rather than meaningless filters to indicate the predictive information in the image. These new features are denoted as X†'s, and hence the name "dagger technique". The rest

Table 3: Interaction-based Engineer: "Dagger Technique". This table summarizes the construction procedure of X† (the "dagger technique"). Suppose we have a variable set $\{X_1, X_2\}$ and each of them can take values in {0, 1}. We can construct X† and the values of this new feature is defined using the local average of the target variable Y based on the partition retained from the variable set $\{X_1, X_2\}$. Here the variable set $\{X_1, X_2\}$ produces 4 partitions. Hence, we can define X† according to the following table. In test set, we do not observe target variable (or response variable) Y, so we use the training set values. Hence, the reminder is that in generating test set X† we use $\overline{yj}$'s from training set where j takes values in {1, 2, 3, 4}. of this subsection we formally define this method of using partitions to define novel features.

Training set : $X^\dagger$	X_1	X_2
$\bar{y}_1 = \mathbb{E}(Y \mid X_1 = 1, X_2 = 1)$	1	1
$\bar{y}_2 = \mathbb{E}(Y \mid X_1 = 1, X_2 = 0)$	1	0
$\bar{y}_3 = \mathbb{E}(Y \mid X_1 = 0, X_2 = 1)$	0	1
$\bar{y}_4 = \mathbb{E}(Y \mid X_1 = 0, X_2 = 0)$	0	0

$\rightarrow$

Test set : $X^\dagger$	X_1	X_2
$\bar{y}_1$ (generated from training set)	1	1
$\bar{y}_2$ (generated from training set)	1	0
$\bar{y}_3$ (generated from training set)	0	1
$\bar{y}_4$ (generated from training set)	0	0

A major benefit for using the proposed I-score is the partition retention technique. This is a feature engineering technique that helps us to preserve the information of a variable set and convert it into one feature. Since ROC AUC can- not be directly computed between a response variable and the potential variable set, common procedure tends to fit a model first before AUC is computed. This is a very costly method for the following two reasons. First, the fitting of a regression or a classification model can be very costly to train. Second, the model fitting procedure cannot guarantee the prediction results of the final predictor. If the AUC value is low, there is no solution to distinguish whether the poor AUC result comes from model fitting or variable selection.

To tackle this problem, a proposed technique is to use partition retention. These new features are denoted as X†'s and hence we call this method the "dagger technique". Now we introduce this technique as follows. Suppose we have a supervised learning problem and we are given explanatory variables X and response variable Y. Suppose X has partitions size k. We can create a novel non-parametric feature using the following formula

$$X^\dagger := \bar{Y}_j, \text{ while } j \in \{1, 2, ..., k\} \tag{2}$$

where k is the size of the total partitions formed by X. For example, suppose we have a $X_1 \in \{1, 0\}$ and $X_2 \in \{1, 0\}$. Then the variable set $\{X_1, X_2\}$ has 4 partitions, i.e. computed using 22 = 4. In this case, the running index for notating the partition j can take values {1, 2, 3, 4}. Then, based on this variable set $\{X_1, X_2\}$, we can create a new feature called X†$\{X_1, X_2\}$ that is a combination of X_1 and X_2 using partition retention. Hence, this new feature can be defined as $X†_{\{X1, X2\}} := Y_j$ while j ∈ {1, 2, 3, 4} as discussed above. We can summarize this example in tabular form (see Table 3).

$$\begin{array}{|c|c|c|}\hline x_1 & x_2 & x_3 \\ \hline x_4 & x_5 & x_6 \\ \hline x_7 & x_8 & x_9 \\ \hline\end{array} \rightarrow \begin{array}{|c|c|}\hline \mathcal{B}(x_1,x_2,x_4,x_5) & \mathcal{B}(x_2,x_3,x_5,x_6) \\ \hline \mathcal{B}(x_4,x_5,x_7,x_8) & \mathcal{B}(x_5,x_6,x_8,x_9) \\ \hline\end{array} \xrightarrow{\dagger} \begin{array}{|c|c|}\hline x_1^\dagger & x_2^\dagger \\ \hline x_3^\dagger & x_4^\dagger \\ \hline\end{array}$$

Figure 2: Interaction-based Convolutional Neural Net- work. This is the executive diagram for the basic design of the proposed Interaction-based Convolutional Neural Net- Network (ICNN). The symbol "B" means the proposed Backward Dropping Algorithm (BDA). Suppose all variables are dichotomous. For example, the first group $B_{(x1, x2, x4, x5)}$, after running the BDA, may be reduced to $\{x_1, x_5\}$. The symbol "†" means the construction of interaction-based feature engineer using local averages of target variables based on the partition generated using high I-score variables or features in training set. For the set $\{x_1, x_5\}$, dagger technique can be used to construct x†1, which is based on the partition retained from the set $\{x_1, x_5\}$. Though this figure presents a simple situation, most complex ICNN can be designed using the same technique automatically. The proposed technique here can be generalized to any large-scale data set.

Interaction-based Convolutional Neural Network (ICNN)

I-score, Backward Dropping Algorithm, and Dagger Technique are proposed to replace pre-trained kernel in Convolutional Neural Network (CNN). While the original CNN uses a pre-trained filter, the proposed techniques can automatically extract features from image data set. How do we understand the medical images and why does certain medical? images carry diseased status? These questions can be answered by Interaction-based Convolutional Neural Network (ICNN), but they remain challenging for the original deep CNNs due to the fact that pre-defined kernels or filters are used without any feature selection method. This is because the kernels and filters used suffer low predictivity and they do not go through Backward Dropping Algorithm.

Suppose there is a 3-by-3 matrix (this can be considered as one image) as shown in the Figure 2. First, a kernel of size 2-by-2 is defined and this 2-by-2 kernel coins 4 variables. These 4 variables form a small group and Backward Dropping Algorithm (BDA, and it is denoted as B) is used. Each group, after BDA, finely selects a subset, of which the dagger technique is used. This procedure is illustrated in Figure 2.

We can design an I-score implemented neural networks. Since the pre-defined filters are replaced with proposed BDA and dagger technique, we call the new design an Interaction based Convolutional Neural Network (ICNN).

Forward Propagation (Forward Pass)

To illustrate the procedure of model training. Let us consider a set of input variables to be $\{X^\dagger_1, X^\dagger_2, X^\dagger_3\}$. In the proposed work, this is referring to the variable modules, also notated as $X^\dagger$'s, that we created using Interaction-based Feature Engineer (Equation 2). For this discussion, we define a set of weights $\{w_1, w_2, w_3\}$ to construct a linear transformation. The symbol Σ below in the following diagram represents this linear transformation that takes the form $X^\dagger_1 w_1 + X^\dagger_2 w_2 + X^\dagger_3 w_3$.

For simplicity of notation, we write $\Sigma = \sum_{j=1}^{3} w_j X_j^\dagger$ in short Then we denote a(•) as an activation function. We choose sigmoid to be this activation function a(•). This means we have output $\hat{y}$ to be defined as a(Σ). In other words, let us write the following

$$\hat{y} := a(\Sigma) = a(\sum_{j=1}^{3} w_j X_j^\dagger) = 1/(1 + \exp(-(\sum_{j=1}^{3} w_j X_j^\dagger))) \tag{3}$$

The general form (assuming there are p variable modules) is expressed below

$$\hat{y} := a(\Sigma) = a(\sum_{j=1}^{p} w_j X_j^\dagger) = 1/(1 + \exp(-(\sum_{j=1}^{p} w_j X_j^\dagger))) \tag{4}$$

Architecture

This architecture of neural network is presented below. For simplicity of drawing this picture, we assume there are 3 input variable modules, $\{X^\dagger_1, X^\dagger_2, X^\dagger_3\}$. In practice, the number of variable modules (the total number of $X^\dagger$'s) depends on image data dimensions, window size, stride level, and starting point [25].

$$\begin{tikzcd}
\boxed{X_1^\dagger} \arrow[dr] & \\
\boxed{X_2^\dagger} \arrow[r] & a\left(\sum_{j=1}^{p} w_j X_j^\dagger\right) \arrow[r] & \hat{Y} \\
\vdots & \\
\boxed{X_p^\dagger} \arrow[uur] &
\end{tikzcd}$$

Figure 3: The above architecture presents a feed forward neural network with three input variables. The input variables are $\{X_1^\dagger, X_2^\dagger, X_3^\dagger\}$ which are variable modules created using equation 2.

For the loss function, we used the binary cross-entropy loss function. This loss function is designed to minimize the distance between a target probability distribution P an estimated target distribution Q when the task is a two-class classification problem. The cross-entropŷ loss function is defined as the following

$$\mathcal{L}(y_i, \hat{y}_i) = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i) \tag{5}$$

where y_i is the ground truth of response variable for the it observation and y_i is the predicted value of response variable for the ith observation. The linear transformation, nonlinear transformation, and the computation of the loss function completes the forward propagation.

Backward Propagation (Backward Pass).

To search for the optimal weights, we use an optimizer algorithm called RMSprop (short for Root Mean Square Propagation, a named suggested by Geoffrey Hinton). With the loss function computed above, we derive the gradient of the loss function to be $\nabla L := \partial L(y, \hat{y})/\partial w$. At each iteration t, we compute $v_{t,\nabla L} := \beta_{v_{t-1},\nabla L} + (1 - \beta)\nabla L^2$ while β is a tuning parameter. Note that the square term on ∇L is elementwise multiplication. Then we can update the weights using $w_t := w_{t-1} - \eta \cdot \nabla\mathcal{L}/\sqrt{v_{t,\nabla\mathcal{L}}}$ while η is learning rate The value of learning rate η is a tuning parameter and it is usually a very small number. This process starts with the weights $w = \{w_1, w_2, w_3\}$. Hence, it earns the name back- loss function and goes back to the beginning to update the ward propagation.

Computation of Dimensions

We first discuss three tuning parameters and then we formally write out the dimension of a new matrix after convolutional operation on an image matrix.

- **Window Size.** Window size is the size of the local area that we narrow down to run the Backward Dropping Algorithm.
- **Stride Level.** The level of stride is how many rows or columns that get skipped over. This tuning parameter al- lows the algorithm to move faster but it makes sacrifice by skipping some variables.
- **Starting Point.** Another tuning parameter that we recommend to adjust is the starting point. The starting point represents the location of the first pixel that we start the proposed operation. The vanilla starting point is to start the rolling window from the pixel located at the first row and the first column.

The above discussion introduced the tuning parameters of window size, stride level, and starting point. These parameters update our input matrix and generate a new matrix with different sizes. Let us denote window size to be *w*, stride level to be *l*, and starting point to be *p*. Given an input matrix with size sin by sin, the output matrix has new dimensions computed as the following

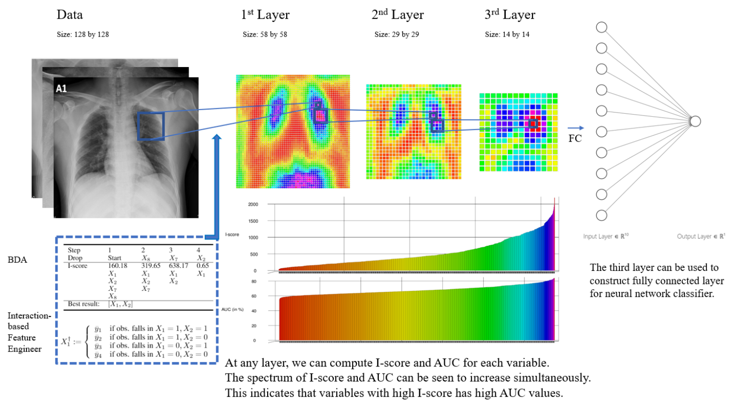
$$\lfloor \frac{s_{\text{in}} - p - w + 1}{l} + 1 \rfloor \times \lfloor \frac{s_{\text{in}} - p - w + 1}{l} + 1 \rfloor \tag{6}$$

For simplicity of this investigation, we assume input matrices to be a square. In other words, in the application of this paper, we process the input images to have the same width and height, i.e. 128 by 128 pixels. For future work, this can be generalized into different shapes.

Interaction-based Recurrent Neural Network (IRNN)

I-score and Dagger Technique are proposed to replace com- plicated designs such as Gated Recurrent Unit and Long Short-Term Memory in the text sentiment analysis. Many text sentiment classification tasks are conducted using com- plicated Recurrent Neural Networks (RNNs) and the reason is to detect long-term dependencies in sequential or time- series data sets. The proposed I-score and Dagger Tech- nique can address this explainability problem. Not only can I-score and Dagger Technique raise prediction performance, the proposed methods can also address the semantics in lan- guage problems, which provides explainability in Natural Language Processing (NLP). This innovation leads to a fam- ily of novel network architectures called Interaction-based Recurrent Neural Network (IRNN).

Suppose the data set is a text document with length 8. This means an instance or an observation in this text document data set may have up to 8 variables or features (this can be considered as a sentence with 8 words). The pro- posed technique is summarized in Figure 5. A kernel of size.


Data
Size: 128 by 128
1st Layer
Size: 58 by 58
2nd Layer
Size: 29 by 29
3rd Layer
Size: 14 by 14
A1
FC
Input Layer ∈ $\mathbb{R}^{10}$
Output Layer ∈ $\mathbb{R}^1$
BDA
Interaction-based Feature Engineer
I-score
AUC (in %)
The third layer can be used to construct fully connected layer for neural network classifier.
At any layer, we can compute I-score and AUC for each variable.
The spectrum of I-score and AUC can be seen to increase simultaneously.
This indicates that variables with high I-score has high AUC values.

Step	1	2	3	4
Drop	Start	X_8	X_7	X_2
I-score	160.18	319.65	638.17	0.65
	X_1 X_2 X_7 X_8	X_1 X_2 X_7	X_1 X_2	X_1
Best result:	$[X_1, X_2]$			

$$X_1^\dagger := \begin{cases} \bar{y}_1 & \text{if obs. falls in } X_1 = 1, X_2 = 1 \\ \bar{y}_2 & \text{if obs. falls in } X_1 = 1, X_2 = 0 \\ \bar{y}_3 & \text{if obs. falls in } X_1 = 0, X_2 = 1 \\ \bar{y}_4 & \text{if obs. falls in } X_1 = 0, X_2 = 0 \end{cases}$$

Figure 4: This executive diagram summarizes the key components of the method: Interaction Convolutional Neural Network, proposed in this paper. This design heavily relies on the I-score and has an architecture that is interpretable at each location of the image at each convolutional layer. More importantly, the proposed design satisfies all three dimensions (C1, C2, and C3 in the Introduction) of the definition of interpretability and explain ability.

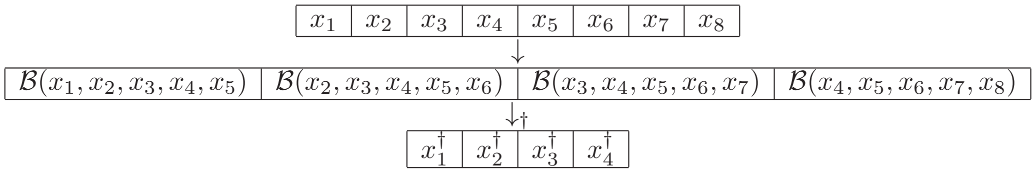

x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8
$\mathcal{B}(x_1,x_2,x_3,x_4,x_5)$ $\mathcal{B}(x_2,x_3,x_4,x_5,x_6)$ $\mathcal{B}(x_3,x_4,x_5,x_6,x_7)$ $\mathcal{B}(x_4,x_5,x_6,x_7,x_8)$
$\downarrow\dagger$
$x_1^\dagger$ $x_2^\dagger$ $x_3^\dagger$ $x_4^\dagger$

Figure 5: Interaction-based Recurrent Neural Network. This is the executive diagram for the basic design of the proposed Interaction-based Recurrent Neural Network (IRNN). The symbol "B" means the proposed Backward Dropping Algorithm. The symbol "†" means the construction of interaction-based feature engineer using local averages of target variables based on the partition generated using high I-score variables or features in training set. The methodology in this figure shares the same philosophy and approach in Figure 2, which implies that the proposed methodology can be generalized into almost all applications with different forms of data sets.

Forward Propagation (Forward Pass).

We briefly dis- cuss the basic RNN that we will use in the application and the diagram for the basic RNN. Suppose we have input features X_1, X_2,…. These features are directly processed for the text document which can be processed word index or they can be embedded word vectors. The features are fed into the hidden layer where the neurons (or units) are de- noted as h_1, h_2,…... There is a weight connecting the previous neuron with the current neuron. We denote this weigh as W. Each current neuron has contribution from current feature which is connected with a weight parameter U. For any t in $\{1, 2, ..., T\}$, we can compute each hidden neuron by using the following formula

$$h_t = g(W \cdot h_{t-1} + U \cdot X_t + b) \tag{7}$$

where W and U are trainable parameters, b is the bias term, and g(.) can be an activation function. This choice of the where W and U are trainable parameters, b is the bias term, activation function is completely determined by the dataset and the end-user. A list of famous activation functions can be found in [25, 26]. In the end of the architecture, we can finally compute the predicted probability of Y given the hidden neurons by using the formula

$$\hat{Y} = g(V \cdot h_T + b) \tag{8}$$

where the weights W, U, and V are shareable in the entire architecture. Forward propagation of RNN is referring to the procedure when information flow from the input features to output predictor using equation 7 and equation 8.

Backward Propagation (Backward Pass)

The forward propagation allows us to pass information from input layer which are features extracted from text document to the out-put layer which is the predictor. To search for the optimal weights, we need to compute the loss function and optimize the loss function by updating the weights. Since the task is text classification, we only have one output in the output layer. In addition, the task is a two-class classification problem because Y can only take value of $\{0, 1\}$. This means we can use the loss function called cross-entropy function, which is defined in equation 5. We write it again as follows

$$\mathcal{L}(Y, \hat{Y}) = -\frac{1}{n}\sum_{i=1}^{n} y_i \log \hat{y}_i - (1-y_i)\log(1-\hat{y}_i)$$

where y_i is the true label for instance i and $\hat{y}_i$ is the predicted value for instance i. Notice that in the proposed architecture we use I-score and the "dagger technique" to construct $X^\dagger$ for the input features where T' is smaller than T and it is a length dependent on tuning process. Next, we can use gradient descent (GD) to update the parameters. At each step s, we can update the weights by

$$\begin{aligned} V_s &= V_{s-1} - \eta \cdot \nabla_V \mathcal{L}(Y, \hat{Y}) \\ U_s &= U_{s-1} - \eta \cdot \nabla_U \mathcal{L}(Y, \hat{Y}) \\ W_s &= W_{s-1} - \eta \cdot \nabla_W \mathcal{L}(Y, \hat{Y}) \end{aligned} \tag{9}$$

where η is learning rate (usually a very small number), the symbol ∇ means gradient, and $\nabla L(\nabla)$ is the gradient (partial derivative) of the loss function $L(\cdot)$. We can formally write the gradients in the following'

$$\begin{aligned} \nabla_V \mathcal{L}(Y, \hat{Y}) &= \frac{\partial}{\partial V}\mathcal{L}(Y, \hat{Y}) \\ \nabla_U \mathcal{L}(Y, \hat{Y}) &= \frac{\partial}{\partial U}\mathcal{L}(Y, \hat{Y}) \\ \nabla_W \mathcal{L}(Y, \hat{Y}) &= \frac{\partial}{\partial W}\mathcal{L}(Y, \hat{Y}) \end{aligned} \tag{10}$$

where $\nabla_{\text{parameter}}$ means gradient or partial derivative with respect to that parameter.

Simulation: Why I-score captures more than AUC?

The following simulation is created to further investigate the advantage the proposed I-score has over the conventional measure AUC. In this experiment, suppose there are variables $X_1,...,X_{10}$ ~iid Binomial(2, 0.5), which means X_i has support $\{0, 1, 2\}$. In other words, the sequence of all 10 independent variables defined above can only take values from $\{0, 1, 2\}$. The experiment creates a sample of 2,000 observations for this experiment. Suppose the model takes the following form

$Y = X_1 + X_2 \pmod 2$

while "mod" refers to modulo of 2, i.e. $1 + 1 = 2 \equiv 0$, $1 + 2 = 3 \equiv 1$, and so on. We can compute the AUC values and the I-score values for all 10 variables. In addition, we also compute both measures for the following models (this is to assume when we do not know the true form of the real model): (i) $X_1 + X_2$, (ii) $X_1 - X_2$, (ii) $X_1 \cdot X_2$, (iv) $X_1/(X_2 + \epsilon)$. In model (iv), we add $\epsilon = 10^{-5}$ to ensure the division is legal. Any higher form of assumptions of the real model would just be a combination of model (i)-(iv), so we assume to use model (i)-(iv) first. To further illustrate the power of I-score statistics, we introduce a new variable specifically constructed by taking the advantage of partition retention, i.e. $X^\dagger := y_j$ while $j \in \Pi_{X1,X2}$ (the novel dagger technique that is defined using variable partition is widely used in application, see equation 2). The simulation has 2,000 observations. We make a 50-50 split. The first 1,000 observations are used to create partitions and the local average of the target variable Y that is required in creating $X^\dagger$ feature is only taken from the first 1,000 observations.

Application in Healthcare: Deploying XAI Method in COVID Early Detection

In the sixteen months since the WHO Emergency Commit- tee declared a global health emergency on January 30th, 2020 based on the outbreak of novel coronavirus SARS- Cov-2 (previously 2019-nCov, also known as COVID-19), the disease has spread to nearly every country in the world [27, 28]. This disease has extreme impact on the health and life form of many of us on a global scale. The fight we took to overcome COVID-19 is essential especially when it comes to detect the infected candidates at an early stage. Investigation through radiography images is the most basic procedure at diagnosing abnormalities of infected patients. Several deep learning algorithms for detection of COVID-19 diseases on CT scans have been proposed. Bai et al. provided the model output to radiologists, and demonstrated that AI-assistance significantly improved radiologist diagnostic accuracy from 85% to 90% in distinguishing COVID classes from non-COVID classes [29]. Minaee et al. (2020) have demonstrated using deep CNNs including ResNet18, ResNet50, DenseNet-121 to classify COVID-19 disease using X-ray images [30]. They have achieved sensitivity rate of 98% and specificity of 90%. on 5,000 Chest X-ray images.

A brief summary of comparison of conventional methods literature and the proposed method is presented in Table 5. A detailed report of the proposed methodology is presented in related studies [25, 26].

Table 4: Simulation Results. This table presents the simulation results for the model Y = $X_1 + X_2$ (mod 2) when X_1, $X_2 \sim$ Bin(2, 0.5). The measure of AUC values has a major drawback: it cannot successfully detect the useful information. Even with the correct variables selected (all guessed models only use the important variables $\{X_1, X_2\}$), AUC measure subjects to serious attack from incorrect model assumption. This flaw renders applications of using AUC measure to select models less ideal and sub-optimal. However, the proposed I-score is capable of indicating the most important variables, X_1 and X_2, disregard the forms of the underlying model. Moreover, the dagger technique of building $X^\dagger$ using partitions generated by the variable set X_1 and X_2 completely recovers full information of the true model even before any machine learning or model selection procedure, which is a novel invention that the literature has not yet seen.

		Average AUC	SD. of AUC	Average I-score	SD. of I-score
Important	X_1	0.50	0.01	0.78	0.71
	X_2	0.50	0.01	0.78	0.67
Noisy	X_3	0.50	0.01	0.43	0.38
	X_4	0.50	0.01	0.44	0.42
	X_5	0.50	0.01	0.56	0.55
	X_6	0.50	0.01	0.93	0.98
	X_7	0.50	0.01	0.56	0.52
	X_8	0.50	0.01	0.42	0.40
	X_9	0.50	0.01	0.36	0.40
	X_{10}	0.50	0.01	0.45	0.44
Guessed models (using $\{X_1, X_2\}$)	model (i): $X_1 + X_2$	0.51	0.01	544.81	9.61
	model (ii): $X_1 - X_2$	0.51	0.01	548.18	9.48
	model (iii): $X_1 + X_2$	0.51	0.03	264.81	11.60
	model (iv): $X1/(X2 + \epsilon)$	0.51	0.03	233.50	9.07
	$X^\dagger$ (see eq. 2)	1	0	999.11	0.60
	$\{X_1, X_2\}$	NA	NA	281.36	6.82
True model	$X_1 + X_2$ (mod 2)	1	0	999.11	0.60

Table 5: The table presents experimental results of COVID-19 data set from literature. A number of different ultradeep CNNs are used to classify COVID patients from non-COVID people. The performance is summarized below. The average number of parameters of the ultra-deep CNNs can exceed 25 million parameters with top AUC to be at 99.2%. The proposed methods have average number of parameters to be less than 100k with top AUC of 99.8%. This is a 99% reduction on number of parameters without sacrificing the prediction performance.

Previous Work	Number of Param.	AUC
DenseNet161 (Minaee et al. 2020)	0.8 - 40 mil. param.	97.6%
ResNet18 (Minaee et al. 2020)	11 mil. param.	98.9%
ResNet50 (Minaee et al. 2020)	25 mil. param.	99.0%
SqueezeNet (Minaee et al. 2020)	~ 1.2 mil. param.*	99.2%
Average	> 25 mil.	97% - 99.2%
Proposed	average 100k param.(a 99% reduction on num. of param.)	98.3% - 99.8%

Visualization and Interpretation

The proposed methods generate highlights that affect the target outcome in the image frame. The study of COVID-19 images can be processed using the proposed methods of Iscore, BDA, and the dagger technique. The visualization can be seen in Figure 6.

For classification tasks that involve COVID-19 and its many lung cancer variations, the proposed methods can also

Table 6: Multi-class Lung Cancer Variants Diagnosis. This table presents experiment results for multi-class lung cancer variants classification. In total, there are 4 classes (0: Healthy, 1: COVID-19, 2: Pneumonia, 3: Tuberculosis). The average prediction performance for 4-class diagnosis is 89% with 26 million parameters in a variety of different neural networks designs. The average prediction performance for 4-class diagnosis is 98% with a shy of 15,000 parameters in proposed network architectures. Proposed I-score enhanced deep learning methods deliver 90% error re- duction while reduce the number of training parameters in neural networks by 99% in diagnosing lung cancer variants under multi-class prediction tasks.

Model	AUC (Test Set)	No. of Parameters
Benchmarks:		
ResNet (Punn and Agarwal 2021)	0.82 - 0.90	11-25 million
Inception (Punn and Agarwal 2021)	0.89 - 0.91	23-56 million
DenseNet (Punn and Agarwal 2021)	0.93 - 0.94	0.8-40 million
Average Average	0.89	26 million
Proposed:		
ICNN (Θ_1: {starting point: 6, window size: 2 by 2, stride: 2})	0.97	12,000
ICNN (Θ2: {starting point: 4, window size: 3 by 3, stride: 3})	0.98	13,000
ICNN (use Θ_1, Θ_2 to gen. features, then concatenate them)	0.98	20,000
(For the above ICNN, see Panel A of Figure 4)		
IRNN (see Panel B of Figure 4)	0.99	15,000
Average	0.98	15,000

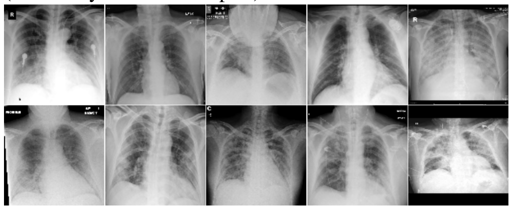
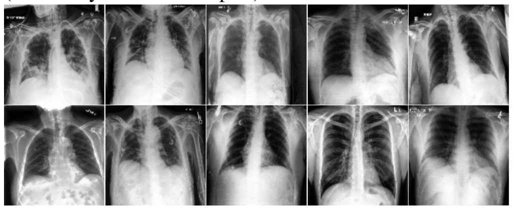
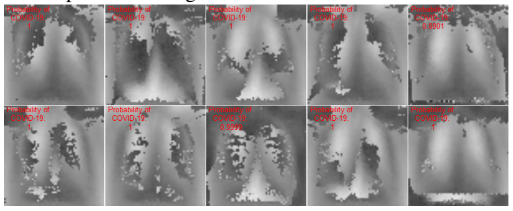
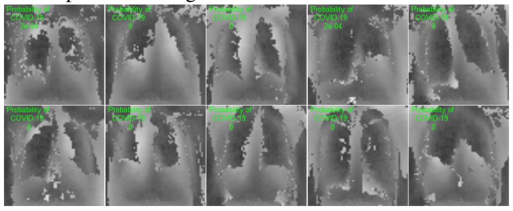
Panel A: True Label: COVID	Panel B True Label: Non-COVID
Input Images: 128 by 128 (Randomly select 10 samples) 	Input Images: 128 by 128 (Randomly select 10 samples) 
1st Conv. Layer: 61 by 61 (Starting Point = 6, Window 2 by 2, Stride = 2) Remark: $61 \times 61 = 3,721$ variables Same 10 images above with 3,721 variables Labels predicted using Model 1 	1st Conv. Layer: 61 by 61 (Starting Point = 6, Window 2 by 2, Stride = 2) Remark: $61 \times 61 = 3,721$ variables Same 10 images above with 3,721 variables Labels predicted using Model 1 

Figure 7: Visualization for Multi-class Classification Using I-score Enhanced Deep Learning. This figure presents 4 samples (each row is a sample with different label). There are 4 classes (0: Healthy, 1: COVID, 2: Other Pneumonia, 3: Tuberculosis). To present the highlighted region for dis- eased diagnosis, we use proposed "dagger technique" to create dagger features. Then we check the numerical values of these dagger features, which then indicate marginal dagger signals of diseased status per class. We observe that pro- posed class-specific dagger features can make correct diagnosis amongst different lung cancer variants.

be applied to create the following visualization (Figure7).

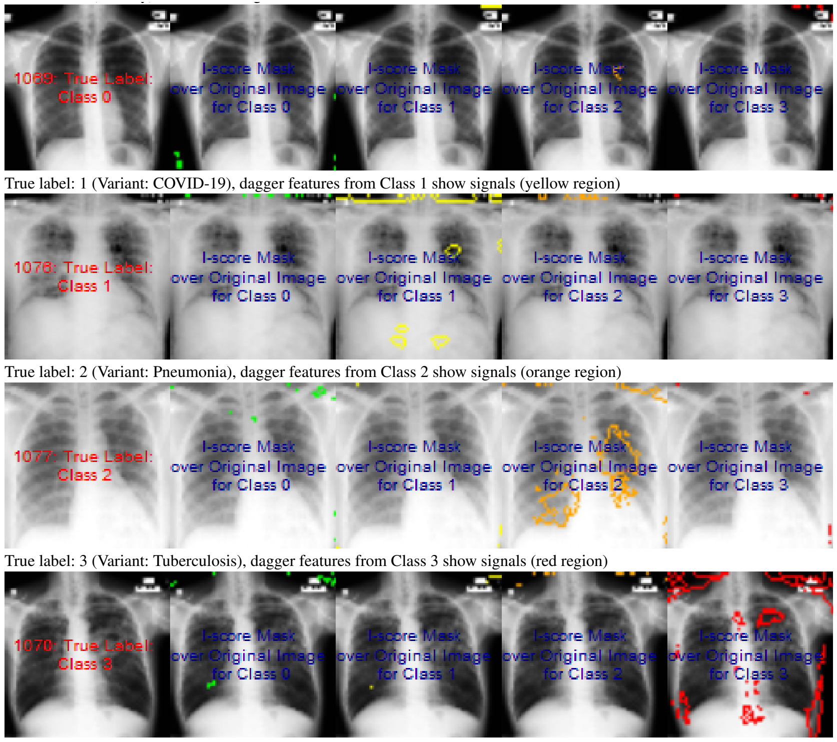

1069: True Label: Class 0
I-score Mask over Original Image for Class 0
I-score Mask over Original Image for Class 1
I-score Mask over Original Image for Class 2
I-score Mask over Original Image for Class 3
True label: 1 (Variant: COVID-19), dagger features from Class 1 show signals (yellow region)
1076: True Label: Class 1
True label: 2 (Variant: Pneumonia), dagger features from Class 2 show signals (orange region)
1077: True Label: Class 2
True label: 3 (Variant: Tuberculosis), dagger features from Class 3 show signals (red region)
1070: True Label: Class 3

Figure 8: Visualization for Multi-class Classification Using I-score Enhanced Deep Learning. This figure presents 4 samples (each row is a sample with different label). There are 4 classes (0: Healthy, 1: COVID, 2: Other Pneumonia, 3: Tuberculosis). To present the highlighted region for diseased diagnosis, we use proposed "dagger technique" to create dagger features. Then we check the numerical values of these dagger features, which then indicate marginal dagger signals of diseased status per class. We observe that proposed class-specific dagger features can make correct diagnosis amongst different lung cancer variants [].

Conclusions

Explainable AI System for Early COVID-19 and COVID-Variant Screening

As the most important contribution of this paper, an Explainable Artificial Intelligence (XAI) system is proposed to assist radiologists for the initial screening of COVID-19 and other related variants using chest X-ray images for treatment and disease control. This innovation can be widely adapted in the application of how AI systems are deployed in hospitals and healthcare systems. We anticipate that other related diseases with viral pneumonia signs can use the same detection methods proposed in our paper, which ensure the development of testing procedures with accountability, responsibility, and transparency to human users and patients.

A Heuristic and Theoretical Framework of XAI. This paper introduces a heuristic and theoretical framework for addressing the XAI problems in large-scale and high- dimensional data sets. **We provide three dimensions as necessary conditions and premises required for a measure to be regarded as explainable and interpretable.**

An I-score Enhanced Deep Learning Framework. To address the XAI problems heuristically described above, this paper introduced a novel design of an explainable and self-interpretable Interaction-based Convolutional Neural Net- work (ICNN) and Interaction-based Recurrent Neural Net- work (IRNN). Our work provides a flexible approach to con- tribute to the major issues about explain ability, interpretability, transparency, and trustworthiness in black-box algorithms. We introduce and implement a non-parametric and interaction-based feature selection methodology and use this as replacement of pre-defined filters that are widely used in ultra-deep CNNs.

Any CNN architecture that adapts the proposed technology with transformation from two-dimensional array to the proposed "dagger technique" features can be regarded as Interaction-based Convolutional Neural Network (ICNN) or from one-dimensional array to the same proposed "dagger technique" can be regarded as Interaction-based Recurrent Neural Network (IRNN) with sequential model assumption. We encourage both the statistics and computer science com- munities to further explore this area to deliver more transparency, trustworthiness, and accountability to deep learning algorithms and to build a world with truly Responsible A.I. [31, 32].

Acknowledgments

We would like to dedicate this to H. Chernoff, a well-known statistician and a mathematician worldwide, in honor of his 98th birthday and his contributions in Influence Score (I- score) and the Backward Dropping Algorithm (BDA). We are particularly fortunate in receiving many useful comments from him. Moreover, we are very grateful for his guidance on how I-score plays a fundamental role that measures the potential ability to use a small group of explanatory variables for classification which leads to much broader impact in fields of pattern recognition, computer vision, and representation learning.

References

1. Gunning, D. (2016). *Broad Agency Announcement Explainable Artificial Intelligence (XAI).* Technical report.
2. Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2021). Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1), 18.
3. Lipton, Z. C. (2018). The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3), 31-57.
4. Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE access*, 6, 52138-52160.
5. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), 206-215.
6. Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. a*rXiv preprint arXiv*:1702.08608.
7. Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267, 1-38.
8. Lo, A., Chernoff, H., Zheng, T., & Lo, S. H. (2015). Why significant variables aren't automatically good predictors. *Proceedings of the National Academy of Sciences*, 112(45), 13892-13897.
9. Lo, A., Chernoff, H., Zheng, T., & Lo, S. H. (2016). Framework for making better predictions by directly estimating variables' predictivity. Proceedings of the *National Academy of Sciences,* 113(50), 14277-14282.
10. Lombrozo, T. (2006). The structure and function of explanations. *Trends in cognitive sciences,* 10(10), 464-470.
11. Lombrozo, T. (2011). The instrumental value of explanations. *Philosophy Compass,* 6(8), 539-551.
12. Dzindolet, M. T., Peterson, S. A., Pomranky, R. A., Pierce, L. G., & Beck, H. P. (2003). The role of trust in automation reliance. *International journal of human-computer studies,* 58(6), 697-718.
13. Lipton, Z. C. (2018). The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue,* 16(3), 31-57.
14. Ramanishka, V., Das, A., Zhang, J., & Saenko, K. (2017). Top-down visual saliency guided by captions. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7206-7215).
15. Fong, R. C., & Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. *In Proceedings of the IEEE international conference on computer vision* (pp. 3429-3437).
16. Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., & Clune, J. (2016). Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems,* 29, 3387-3395.
17. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. *In Proceedings of the IEEE international conference on computer vision* (pp. 618-626).
18. Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep

inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv*:1312.6034.
19. Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., & Lipson, H. (2015). Understanding neural networks through deep visualization. *arXiv preprint arXiv*:1506.06579.
20. Zhang, J., Bargal, S. A., Lin, Z., Brandt, J., Shen, X., & Sclaroff, S. (2018). Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10), 1084-1102.
21. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2921-2929).
22. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). " Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD *international conference on knowledge discovery and data mining* (pp. 1135-1144).
23. Petsiuk, V., Das, A., & Saenko, K. (2018). Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv*:1806.07421.
24. Chernoff, H., Lo, S. H., & Zheng, T. (2009). Discovering influential variables: a method of partitions. *The Annals of applied statistics*, 3(4), 1335-1369.
25. Lo, S. H., & Yin, Y. (2021). An Interaction-based Convolutional Neural Network (ICNN) Towards Better Understanding of COVID-19 X-ray Images. *arXiv preprint arXiv*:2106.06911.
26. Lo, S. H., & Yin, Y. (2021). Language Semantics Interpretation with an Interaction-Based Recurrent Neural Network. *Machine Learning and Knowledge Extraction*, 3(4), 922-945.
27. Velavan, T. P., & Meyer, C. G. (2020). The COVID-19 epidemic. *Tropical medicine & international health*, 25(3), 278.
28. Wang, S., Kang, B., Ma, J., Zeng, X., Xiao, M., Guo, J., ... & Xu, B. (2021). A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). *European radiology*, 1-9.
29. Bai, H. X., Wang, R., Xiong, Z., Hsieh, B., Chang, K., Halsey, K., ... & Liao, W. H. (2020). Artificial intelligence augmentation of radiologist performance in distinguishing COVID-19 from pneumonia of other origin at chest *CT. Radiology*, 296(3), E156-E165.
30. Minaee, S., Kafieh, R., Sonka, M., Yazdani, S., & Soufi, G. J. (2020). Deep-covid: Predicting covid-19 from chest x-ray images using deep transfer learning. *Medical image analysis*, 65, 101794.
31. Punn, N. S., & Agarwal, S. (2021). Automated diagnosis of COVID-19 with limited posteroanterior chest X-ray images using fine-tuned deep neural networks. *Applied Intelligence*, 51(5), 2689-2702.
32. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2921-2929).