DOI: 10.21917/ijivp.2023.0414

XAI USING FORMAL CONCEPT LATTICE FOR IMAGE DATA

Bhaskaran Venkatsubramaniam and Pallav Kumar Baruah
Department of Math and Computer Science, Sri Sathya Sai Institute of Higher Learning, India

Abstract

A Formal concept lattice can be used to generate explanations from a black box model. This novel approach has been applied and proven for tabular data. It has also been compared to popular techniques in XAI. In this work, we apply the approach to image data. Image data, in general, comprises large dimensions and hence poses a challenge to build a formal concept lattice from such large data. We break the image into parts and build multiple sub-lattices. Using the combination of sub-lattice explanations, we generate the complete explanation for the entire image. We present our work beginning with a simple synthetic dataset for providing an intuitive idea of explanations and its credibility. This is followed by explanations of a model built on the popular MNIST dataset proving consistency of explanations on a real dataset. Text explanations from the lattice are converted to images for ease of visual understanding. We compare our work with DeepLIFT by viewing image masks obtained through contrastive explanation for specific digits from the MNIST dataset. This work proves the feasibility of using formal concept lattices for image data.

Keywords:

Explainable AI, XAI, Formal Concept Analysis, Lattice for XAI, XAI for Images

1. INTRODUCTION

Machine Learning or Deep Learning models are deployed in production after training them on a dataset as the required evaluation parameters like accuracy are achieved. It has become common knowledge to best fit the curve to the dataset but not many of the models can explain their decisions. The model learns patterns from training data and predicts an outcome for an instance or maps an instance to a class. Most of the deep learning modes are extremely accurate with many datasets but it is not easy to see through a deep learning network to understand the reason behind an outcome. While Explainable AI (XAI) tries to open up new techniques to present the reasons behind an outcome, there are also approaches that suggest not to use such black box models at all [1]. But it is not easy to keep Deep Learning models away. Yet it is not easy to adopt them in production without explainability making XAI a necessity [2].

There are multiple ways to generate explanations. Some of them are model agnostic and some are model specific. Local Interpretable Model Agnostic Explanation (LIME) is a model agnostic technique that builds a locally interpretable model around an instance and provides the weights of the newly learnt model as its explanation. As many more XAI techniques evolve, it becomes important to evaluate these techniques themselves. Robustness of these methods is in question [4] when similar instances need not have similar explanations. LIME too faces the stability challenge especially in producing repeatable consistent explanations under the same conditions [5]. Some of these models also provide a global explanation of the model using weight aggregation of representative instances. But these need not be the true view of the model [6]. Unified approaches [7] are better but they can make assumptions like feature independence that may be false in a dataset. Specific to images, there are many models specific XAI techniques using class activation maps [8]-[10]. Saliency maps provide intuition but over relying on plain intuition without quantitation lack rigorous validation [11]. There is no standard yet on evaluating XAI techniques but invariance tests [12] and randomization tests are a good start [13].

In our earlier work [14], we propose a novel technique to extract explanation by building a formal concept lattice from the dataset. We also prove that our lattice based XAI technique passes all sanity tests, and its explanations are accurate. In similar research work that use a lattice, they use it to guide samples chosen by LIME [15] or produce individual features responsible for a decision [16], not the same as lattice based XAI [14].

In this work, we apply our XAI technique to image data by breaking the image into parts and building multiple sub-lattices. Using the combination of sub-lattice explanations, we generate the complete explanation for the entire image. The primary contribution of this work is to show feasibility of lattices in XAI even for images. Originality of this work lies in adapting our earlier lattice based XAI technique [14] for tabular data to images.

Section 2 introduces the formal concept lattice. Section 3 is a literature review of XAI techniques for deep learning with image data. Section 4 summarizes our novel technique to extract explanations from the lattice for tabular data [14] and how it is modified to adapt to images. Section 4 provides an intuitive idea of the method on a toy dataset, while Section 6 provides elaborate explanations on a real dataset (MNIST). Section 7 compares our method to DeepLIFT. Section 8 contains conclusions and future work.

2. FORMAL CONCEPT LATTICE

A context is a triple (G,M,I), where G is a set of objects, M is a set of attributes and I the relation between them. The notation gIm means that the object g has the attribute m.

For a set $A \subseteq G$, define $A' = \{m \in M \mid gIm \forall g \in A\}$ [A' is the set of attributes common to all the objects in A]

For a set $B \subseteq M$, define $B' = \{g \in G \mid gIm \forall m \in B\}$ [B' is the set of objects which have all attributes in B]

A concept of the context (G,M,I) is a pair (A,B) such that, $A \subseteq G$, $B \subseteq M$, $A'=B$ and $B'=A$. A is called the extent and B the intent of the concept (A,B).

If (A_1,B_1) and (A_2,B_2) are concepts of a context (G,M,I), then (A_1,B_1) is a subconcept of (A_2,B_2) (or (A_1,B_1) is a superconcept of (A_2,B_2)), denoted by $(A_1,B_1) \leq (A_2,B_2)$ (or $(A_2,B_2) \leq (A_1,B_1)$) if $A_1 \subseteq A_2$, equivalently $B_2 \subseteq B_1$ (or $A_2 \subseteq A_1$, equivalently $B_1 \subseteq B_2$). The relation $\leq$ is called the hierarchical order of the concepts. The ordered set of concepts is called the concept lattice of the context (G,M,I). Concept lattices are represented using a hasse line diagram [18]. The Fig.1 contains a sample formal concept lattice.

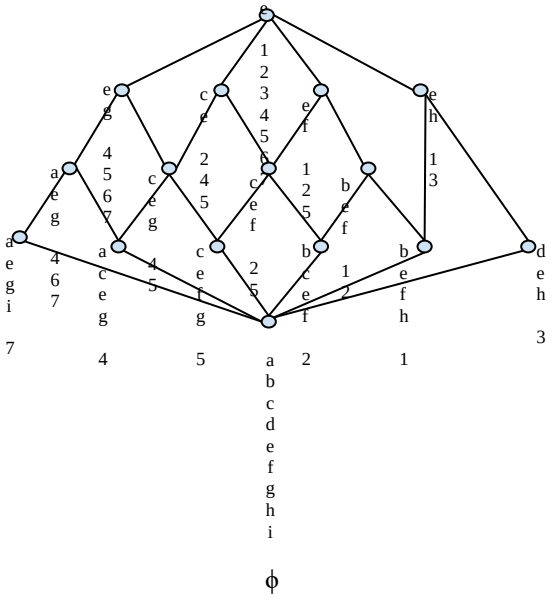

Fig.1. A sample Formal Concept Lattice

In [14], we use the formal concept lattice to extract global, local, similar, and contrastive explanations of a black box model around an instance of interest.

3. LITERATURE REVIEW

In this section, we review popular XAI techniques specific to deep learning with image data. One of the most popular techniques to explain a network is in observing the image pattern that the network recognizes by visualizing the early layer filters of the network as images by Fan et al. [21]. Another technique in the same direction by Van der Maaten et al. was to reduce the dimensionality of the last fully connected layer and visualize them [22]. The semantic nature of the network is brought out when these representations form clusters. Activating patches of the image were found by projecting the receptive field back till the input layer by Girshick et al. [23]. Part of the image could be occluded by a gray patch and output probabilities are obtained. That patch which when grayed out changes the class decision compared to the ground truth is the part most influential on the decision as found by Zeilur et al. [24]. Another class of techniques by Simonyan et al. use backpropagation. They start with a zero image, backpropagate to the input layer and update the image using gradient with respect to the objective function [25]. Viewing such images while not being realistic can yet provide an idea of what the network is viewing. Guided backpropagation passes only positive gradients through the previous layers to update the image, effectively retaining only pixels that have a positive influence on the decision. Gradients can be negative or zero leading to backpropagation not revealing anything. Since this can lead to the saturation problem, many other techniques evolved that avoid the saturation problem. Shrikumar et al. implement DeepLIFT [20] that uses relative differences with respect to a reference input and reference output to avoid the saturation problem. Similarly Integrated Gradients [12] by Sundarrajan et al. accumulated gradients at different pixel intensities to avoid the saturation problem. Also known as Saliency Maps, Class Attribution Map technique by Zhou et al. introduces a global average pooling layer and learns the weights of a linear model [26]. It uses these weights with the filters of the last convolutional layer to produce the heat map. GradCAM [27] by Selvaraju et al. proved that these weights are the same as the sum of gradients of each of the class scores with respect to every pixel in the feature map. Grad-CAM++ [28] by Chattopadhay et al. is an improvisation over GradCAM for images that have multiple instances of artifacts. Smilkov et al. [29] created SmoothGrad which can be used to visually sharpen saliency maps. Concept Activation Vectors [30] by Kim et al. provides a human friendly interpretation of neural network internal state.

Availability of many techniques also leads to the birth of methods that can evaluate XAI techniques. From [12] and [13], we gather a set of axiomatic approaches and sanity tests that can evaluate these techniques. Sundarrajan et al. in [12], discuss four axioms to be satisfied by any XAI technique - namely, Sensitivity, Implementation Invariance, Completeness and Symmetry preserving. Abedayo et al. in [13] bring out basic sanity tests that should be passed by any XAI technique. Surprisingly, some of the most popular XAI techniques either do not satisfy the axioms or do not pass the sanity tests.

Formal Concept Lattices, which are mathematically well-defined structures, cannot fail to satisfy these axioms and pass all the sanity tests [14]. Considering this difference, it is meaningful to use the lattice approach to XAI.

4. METHODOLOGY

We summarize the steps for generating explanations from tabular data [14] as follows:

1. A formal concept lattice is constructed from the dataset and the implications with their support are derived from the lattice.
2. With the above generated implications and a user provided implication cutoff, synthetic data is generated for each feature from its range of values respecting the implications with support greater than the cutoff value.
3. A lattice is then constructed with this synthetic dataset and implications are applied to reduce the feature set at each node of the lattice.
4. Model outcome is obtained for the instances in this dataset and is communicated to all its child nodes. Union of these outcomes are calculated and distributed below similarly till the root of the lattice.
5. Global explanation is derived from the minimum set of features that imply or deny an outcome.
6. Local explanation is derived by traversing the nodes of the lattice in the sorted order of their number of reduced features.

We split each image in the dataset and use the above methodology to derive the lattice for each part of the image. These lattices that are built for part of the images are termed as sub-lattices. Explanations are extracted for a data instance by taking the union of the relevant explanations from each sub-lattice. While some parts of the image may be useful to distinguish certain outcomes, some other parts may not play an important role to

distinguish between certain outcomes. With similar and contrastive explanations, we can determine the set of pixels that when changed induce a change in the decision. We begin by extracting explanations on a toy dataset.

5. LATTICE EXPLANATIONS FROM A SIMPLE DATASET

We apply the above explained methodology to images of a simple dataset to provide an intuitive idea of the methodology. Shapes of numbers are created using the * symbol and classified to ten different classes indicating digits. Such shapes for ten digits are shown in Fig.2 and 3.

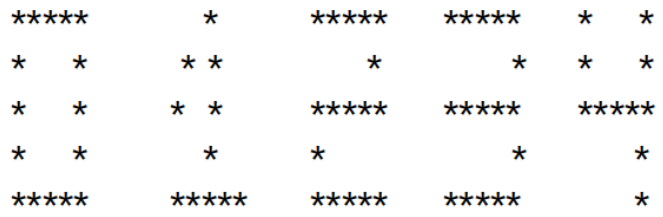

Fig.2. Shape of digits 0 to 4 using the * symbol

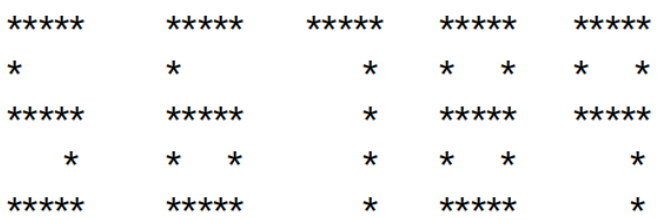

Fig.3. Shape of digits 5 to 9 using the * symbol

Each position is considered as a pixel as part of a 5×5 matrix. The 5×5 matrix for each digit is translated to 25 features indicating the presence of a symbol in that position. These translations follow in Table.1.

Table.1. Binary features of the digit shapes

Features (Binary) [00, 01, 02, .., 44]	Class
1,1,1,1,1,1,0,0,0,1,1,0,0,0,1,1,0,0,0,1,1,1,1,1,1	0
0,0,1,0,0,0,1,1,0,0,1,0,1,0,0,0,0,1,0,0,1,1,1,1,1	1
1,1,1,1,1,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,1,1,1,1,1	2
1,1,1,1,1,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,1,1,1,1,1	3
1,0,0,0,1,1,0,0,0,1,1,1,1,1,1,1,0,0,0,0,1,0,0,0,0,1	4
1,1,1,1,1,1,0,0,0,0,1,1,1,1,1,1,0,0,0,0,1,1,1,1,1,1	5
1,1,1,1,1,1,0,0,0,0,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1	6
1,1,1,1,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1	7
1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1	8
1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,0,0,0,0,1,0,0,0,0,1	9

With features titled 00, 01, …, 44, a lattice was constructed on the dataset in Table 2. Part of the Global explanation from this lattice is as follows:

1. (00,1) ===>>> 0 or 2 or 3 or 4 or 5 or 6 or 7 or 8 or 9
2. (02,1) ===>>> 0 or 1 or 2 or 3 or 5 or 6 or 7 or 8 or 9
3. (20,1) ===>>> 0 or 1 or 2 or 3 or 4 or 5 or 6 or 8 or 9
4. (30,0) ===>>> 1 or 3 or 4 or 5 or 7 or 9
5. (01,1) ===>>> 0 or 2 or 3 or 5 or 6 or 7 or 8 or 9
6. (14,1) ===>>> 0 or 2 or 3 or 4 or 7 or 8 or 9
7. (34,1) ===>>> 0 or 3 or 4 or 5 or 6 or 7 or 8 or 9
8. (10,0) ===>>> 1 or 2 or 3 or 7
9. (21,0) ===>>> 0 or 1 or 7
10. (22,1) ===>>> 1 or 2 or 3 or 4 or 5 or 6 or 8 or 9
11. (21,1) ===>>> 2 or 3 or 4 or 5 or 6 or 8 or 9
12. (40,1) ===>>> 0 or 1 or 2 or 3 or 5 or 6 or 8
13. (22,0) ===>>> 0 or 7
14. (01,0) ===>>> 1 or 4
15. (10,1) ===>>> 0 or 4 or 5 or 6 or 8 or 9
16. (34,0) ===>>> 1 or 2
17. (20,0) ===>>> 7
18. (00,0) ===>>> 1

In the very first pixel position (0,0), all digits have the * symbol placed there except for the digit 1. This is brought out in explanation 18, stating that the absence of the symbol * at position (0,0) leads to classifying it as 0. Similarly, absence of the * symbol in position (2,0) indicates the class 7 as no other digit misses the * symbol in that position.

Apart from global explanation, local explanation for an instance along with similar and contrastive explanations around an instance is obtained from the lattice [14]. Features for a Zero numeral were provided, and the following explanation was generated:

Enter an instance for an explanation (-1 *to quit*): 1,1,1,1,1,1,0,0,0,1,1,0,0,0,1,1,0,0,0,1,1,1,1,1,1

Features: (00,1) *deny class*(*es*): 1

Remaining class(*es*) *are*: 0 2 3 4 5 6 7 8 9

Features: (02,1) *deny class*(*es*): 4

Remaining class(*es*) *are*: 0 2 3 5 6 7 8 9

Features: (20,1) *deny class*(*es*): 7

Remaining class(*es*) *are*: 0 2 3 5 6 8 9

Features: (14,1) *deny class*(*es*): 5 6

Remaining class(*es*) *are*: 0 2 3 8 9

Features: (34,1) *deny class*(*es*): 2

Remaining class(*es*) *are*: 0 3 8 9

Features: (21,0) *deny class*(*es*): 3 8 9

Remaining class(*es*) *are*: 0

Lattice traversal has denied all class(*es*) *except* 0

Features: (22,0) *lead to class*(*es*) 0 *with a confidence of* 0.500000

Features: (10,1) (21,0) *lead to class*(*es*) 0 *with a confidence of* 1.000000

Generating similar and contrastive explanations:

Changing features: 14 (1 *to* 0) 21 (0 *to* 1) 22 (0 *to* 1) 23 (0 *to* 1) *changes the class to* 6.

Changing features: 14 (1 *to* 0) 21 (0 *to* 1) 22 (0 *to* 1) 23 (0 *to* 1) 30 (1 *to* 0) *changes the class to* 5.

Changing features: 21 (0 *to* 1) 22 (0 *to* 1) 23 (0 *to* 1) *changes the class to* 8.

Go deeper?(*y/n*): *y*

Changing features: 10 (1 *to* 0) 21 (0 *to* 1) 22 (0 *to* 1) 23 (0 *to* 1) 34 (1 *to* 0) *changes the class to* 2.

Changing features: 10 (1 *to* 0) 21 (0 *to* 1) 22 (0 *to* 1) 23 (0 *to* 1) 30 (1 *to* 0) *changes the class to* 3.

Go deeper?(*y/n*): *y*

Changing features: 10 (1 *to* 0) 20 (1 *to* 0) 30 (1 *to* 0) 40 (1 *to* 0) 41 (1 *to* 0) 42 (1 *to* 0) 43 (1 *to* 0) *changes the class to* 7.

Go deeper?(*y/n*): *y*

Changing features: 21 (0 *to* 1) 22 (0 *to* 1) 23 (0 *to* 1) 30 (1 *to* 0) 40 (1 *to* 0) 41 (1 *to* 0) 42 (1 *to* 0) 43 (1 *to* 0) *changes the class to* 9.

Go deeper?(*y/n*): *y*

Changing features: 00 (1 to 0) 01 (1 to 0) 03 (1 to 0) 04 (1 to 0) 10 (1 to 0) 11 (0 to 1) 12 (0 to 1) 14 (1 to 0) 22 (0 to 1) 24 (1 to 0) 30 (1 to 0) 32 (0 to 1) 34 (1 to 0) changes the class to 1.

From the contrastive explanation, we pick some of the statements and produce the pattern according to that statement.

Changing features: 14 (1 *to* 0) 21 (0 *to* 1) 22 (0 *to* 1) 23 (0 *to* 1) *changes the class to* 6.

From the contrastive explanation, we pick some of the statements and produce the pattern according to that statement.

Changing features: 14 (1 *to* 0) 21 (0 *to* 1) 22 (0 *to* 1) 23 (0 *to* 1) *changes the class to* 6.

The Fig.4 shows ZERO numeral followed by the position where the * symbol is removed followed by the positions where the * symbol is added in order to produce the numeral SIX.

Fig.4. Transition from 0 to 6

Changing features: 14 (1 *to* 0) 21 (0 *to* 1) 22 (0 *to* 1) 23 (0 *to* 1) 30 (1 *to* 0) *changes the class to* 5.

The Fig.5 shows ZERO numeral followed by the position where the * symbol is removed followed by the positions where the * symbol is added in order to produce the numeral FIVE.

Fig.5. Transition from 0 to 5

Changing features: 21 (0 *to* 1) 22 (0 *to* 1) 23 (0 *to* 1) *changes the class to* 8.

The Fig.6 shows ZERO numeral followed by the positions where the * symbol is added to produce the numeral EIGHT.

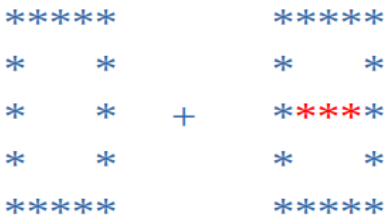

Fig.6. Transition from 0 to 8

6. LATTICE EXPLANATION FROM THE MNIST DATASET

The MNIST (Modified National Institute of Standards and Technology database) dataset comprises handwritten digit images commonly used for training various image processing systems. It consists of 60,000 images in the training set and 10,000 images in the test set. These are 28×28 sized images in grayscale. A sample of such images are shown in Fig.7. From this dataset, we pick around 500 images to train a model. For this section, let us assume a black box model trained with the 500 images and 100% accuracy.

Fig.7. Sample images from the MNIST dataset

In this work, we categorize pixel intensities into ten different ranges so that close pixel intensities do not influence change of class. We divide the images into 4×4 blocks and build sub-lattices for each block, leading to a total of 49 sub-lattices. The feature names are {row-col} values starting with index 0, from 0-0 to 27-27, while the sub-lattices are indexed similarly from 0 to 6, that is 00 to 66. The model outcome of an image is used as the label for all blocks of that image to build the sub-lattice. After constructing the sub-lattices, we generate the explanation for specific instances. Fig.8 shows the numeral instances picked for generating the explanations.

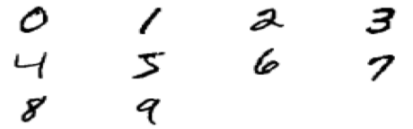

Fig.8. Instances picked for generating explanation from lattice

Here we produce the contrastive explanation to convert the numeral 6 to all the other digits from each of the sub-lattices (indicated by the number in each line). Following each contrastive

explanation is an image produced by applying the changes to the image containing the numeral 6. The newly created image provides an intuitive proof for the explanations from the lattice.

The contrastive explanation for converting 6 to 0 from each of the sub-lattices is as follows:

03: *Changing features*: 3-15 (5 *to* 0) *changes the class to* 0.

04: *Changing features*: 2-16 (1 *to* 0) 2-17 (8 *to* 0) 2-18 (9 *to* 0) 3-16 (9 *to* 0) 3-17 (10 *to* 0) 3-18 (7 *to* 0) *changes the class to* 0.

13: *Changing features*: 4-14 (7 *to* 0) 4-15 (10 *to* 2) 5-12 (1 *to* 0) 5-13 (7 *to* 0) 5-14 (10 *to* 1) 5-15 (10 *to* 9) 6-12 (2 *to* 0) 6-13 (10 *to* 2) 6-14 (10 *to* 9) 7-12 (8 *to* 2) 7-13 (10 *to* 8) 7-15 (3 *to* 10) *changes the class to* 0.

15: *Changing features*: 6-20 (0 *to* 2) 7-20 (0 *to* 10) 7-21 (0 *to* 4) *changes the class to* 0.

22: *Changing features*: 8-11 (5 *to* 6) 9-10 (7 *to* 2) 10-9 (3 *to* 1) 10-10 (10 *to* 9) 11-8 (0 *to* 1) 11-9 (5 *to* 6) 11-11 (10 *to* 9) *changes the class to* 0.

23: *Changing features*: 8-13 (7 *to* 10) 8-14 (1 *to* 10) 8-15 (0 *to* 10) 9-12 (8 *to* 10) 9-13 (1 *to* 10) 9-14 (0 *to* 7) 9-15 (0 *to* 4) 10-12 (5 *to* 10) 10-13 (1 *to* 7) 10-15 (0 *to* 3) 11-12 (4 *to* 8) 11-13 (0 *to* 3) *changes the class to* 0.

24: *Changing features*: 8-16 (0 *to* 10) 8-17 (0 *to* 10) 8-18 (0 *to* 3) 8-19 (0 *to* 7) 9-16 (0 *to* 10) 9-17 (0 *to* 9) 9-18 (0 *to* 1) 9-19 (0 *to* 3) 10-16 (0 *to* 4) 11-17 (4 *to* 0) 11-18 (5 *to* 0) 11-19 (5 *to* 0) *changes the class to* 0.

32: *Changing features*: 12-8 (0 *to* 7) 12-9 (7 *to* 10) 12-10 (10 *to* 9) 12-11 (10 *to* 2) 13-8 (3 *to* 10) 13-10 (10 *to* 2) 13-11 (8 *to* 0) 14-8 (4 *to* 10) 14-9 (10 *to* 7) 14-10 (9 *to* 0) 14-11 (5 *to* 0) 15-9 (10 *to* 4) 15-10 (5 *to* 0) *changes the class to* 0.

33: *Changing features*: 12-12 (2 *to* 0) 12-13 (0 *to* 1) 12-15 (4 *to* 0) 13-12 (1 *to* 0) 13-14 (1 *to* 0) 13-15 (6 *to* 0) 14-13 (1 *to* 0) 14-14 (6 *to* 0) 14-15 (10 *to* 0) 15-13 (7 *to* 0) 15-14 (10 *to* 0) 15-15 (9 *to* 0) *changes the class to* 0.

34: *Changing features*: 12-16 (4 *to* 0) 12-17 (8 *to* 0) 12-18 (10 *to* 0) 12-19 (10 *to* 0) 13-16 (10 *to* 0) 13-17 (10 *to* 0) 13-18 (9 *to* 0) 13-19 (8 *to* 0) 14-16 (9 *to* 0) 14-17 (6 *to* 0) 14-18 (1 *to* 0) 14-19 (2 *to* 0) 15-16 (5 *to* 0) 15-18 (1 *to* 0) 15-19 (8 *to* 0) *changes the class to* 0.

35: *Changing features*: 12-21 (1 *to* 10) 12-22 (0 *to* 7) 13-21 (7 *to* 10) 13-22 (1 *to* 7) 14-21 (8 *to* 10) 14-22 (1 *to* 7) 15-20 (9 *to* 10) 15-21 (5 *to* 10) 15-22 (0 *to* 5) *changes the class to* 0.

41: *Changing features*: 16-6 (0 *to* 3) 16-7 (4 *to* 10) 17-6 (0 *to* 3) 17-7 (5 *to* 10) 18-6 (0 *to* 3) 18-7 (5 *to* 10) 19-6 (0 *to* 3) 19-7 (2 *to* 10) *changes the class to* 0.

42: *Changing features*: 16-8 (10 *to* 9) 16-9 (10 *to* 1) 17-8 (10 *to* 8) 17-9 (6 *to* 0) 17-11 (1 *to* 0) 18-8 (10 *to* 5) 18-9 (6 *to* 0) 18-11 (1 *to* 0) 19-8 (10 *to* 9) 19-9 (10 *to* 0) 19-10 (2 *to* 0) 19-11 (4 *to* 0) *changes the class to* 0.

44: *Changing features*: 16-17 (2 *to* 0) 16-18 (10 *to* 0) 16-19 (10 *to* 5) 17-16 (1 *to* 0) 17-17 (8 *to* 0) 17-18 (10 *to* 5) 17-19 (4 *to* 10) 18-16 (9 *to* 1) 18-17 (8 *to* 6) 18-18 (1 *to* 10) 18-19 (0 *to* 6) 19-17 (2 *to* 10) 19-18 (0 *to* 6) *changes the class to* 0.

45: *Changing features*: 16-20 (5 *to* 10) 16-21 (0 *to* 7) 17-20 (0 *to* 9) 17-21 (0 *to* 2) *changes the class to* 0.

51: *Changing features*: 20-6 (0 *to* 3) 20-7 (2 *to* 10) 21-6 (0 *to* 3) 21-7 (0 *to* 10) 22-6 (0 *to* 1) 22-7 (0 *to* 7) 23-7 (0 *to* 1) *changes the class to* 0.

53: *Changing features*: 20-12 (10 *to* 3) 20-13 (10 *to* 7) 20-14 (10 *to* 9) 20-15 (9 *to* 10) 21-13 (6 *to* 10) 21-14 (4 *to* 10) 21-15 (2 *to* 7) 22-12 (0 *to* 10) 22-13 (0 *to* 9) 22-14 (0 *to* 5) 23-12 (0 *to* 5) 23-13 (0 *to* 1) *changes the class to* 0.

54: *Changing features*: 20-16 (5 *to* 8) 20-17 (0 *to* 6) 20-18 (0 *to* 2) 21-16 (0 *to* 5) *changes the class to* 0.

Applying changes from the contrastive explanation above and changing pixel values in the image containing numeral 6 we obtain the image in Fig.9.

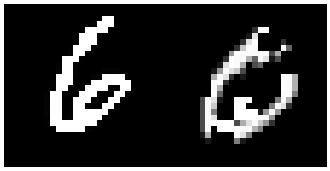

Fig.9. Transition from 6 to 0 in MNIST

The end loop with which the numeral 6 is identified is removed and the rest completed to form a complete loop to form the numeral 0.

The contrastive explanation for converting 6 to 1 from each of the sub-lattices is as follows:

13: *Changing features*: 4-12 (0 *to* 5) 4-13 (0 *to* 10) 4-14 (7 *to* 8) 4-15 (10 *to* 1) 5-12 (1 *to* 9) 5-13 (7 *to* 10) 5-15 (10 *to* 2) 6-12 (2 *to* 7) 6-15 (10 *to* 2) 7-12 (8 *to* 5) 7-15 (3 *to* 2) *changes the class to* 1.

14: *Changing features*: 4-16 (10 *to* 0) 4-17 (8 *to* 0) 5-16 (4 *to* 0) 5-17 (2 *to* 0) 6-16 (3 *to* 0) *changes the class to* 1.

22: *Changing features*: 8-11 (5 *to* 0) 9-10 (7 *to* 0) 9-11 (9 *to* 0) 10-9 (3 *to* 0) 10-10 (10 *to* 0) 10-11 (10 *to* 0) 11-9 (5 *to* 0) 11-10 (10 *to* 0) 11-11 (10 *to* 0) *changes the class to* 1.

24: *Changing features*: 8-16 (0 *to* 2) 8-17 (0 *to* 9) 8-18 (0 *to* 10) 8-19 (0 *to* 8) 9-16 (0 *to* 9) 9-17 (0 *to* 10) 9-18 (0 *to* 10) 9-19 (0 *to* 3) 10-16 (0 *to* 10) 10-17 (0 *to* 10) 10-18 (0 *to* 7) 11-16 (0 *to* 10) 11-17 (4 *to* 9) 11-18 (5 *to* 2) 11-19 (5 *to* 0) *changes the class to* 1.

31: *Changing features*: 15-7 (2 *to* 0) *changes the class to* 1.

32: *Changing features*: 12-9 (7 *to* 0) 12-10 (10 *to* 0) 12-11 (10 *to* 0) 13-8 (3 *to* 0) 13-9 (10 *to* 0) 13-10 (10 *to* 0) 13-11 (8 *to* 0) 14-8 (4 *to* 0) 14-9 (10 *to* 0) 14-10 (9 *to* 0) 14-11 (5 *to* 0) 15-8 (10 *to* 0) 15-9 (10 *to* 0) 15-10 (5 *to* 0) *changes the class to* 1.

33: *Changing features*: 12-12 (2 *to* 0) 12-13 (0 *to* 7) 12-14 (0 *to* 10) 12-15 (4 *to* 5) 13-12 (1 *to* 0) 13-13 (0 *to* 7) 13-14 (1 *to* 10) 13-15 (6 *to* 5) 14-13 (1 *to* 7) 14-14 (6 *to* 10) 14-15 (10 *to* 6) *changes the class to* 1.

34: *Changing features*: 12-17 (8 *to* 0) 12-18 (10 *to* 0) 12-19 (10 *to* 0) 13-16 (10 *to* 4) 13-17 (10 *to* 0) 13-18 (9 *to* 0) 13-19 (8 *to* 0) 14-16 (9 *to* 4) 14-17 (6 *to* 0) 14-18 (1 *to* 0) 14-19 (2 *to* 0) 15-16 (5 *to* 4) 15-18 (1 *to* 0) 15-19 (8 *to* 0) *changes the class to* 1.

35: *Changing features*: 12-20 (10 *to* 0) 12-21 (1 *to* 0) 13-20 (10 *to* 0) 13-21 (7 *to* 0) 13-22 (1 *to* 0) 14-20 (10 *to* 0) 14-21 (8 *to* 0) 14-22 (1 *to* 0) 15-20 (9 *to* 0) 15-21 (5 *to* 0) *changes the class to* 1.

41: *Changing features*: 16-7 (4 *to* 0) 17-7 (5 *to* 0) 18-7 (5 *to* 0) 19-7 (2 *to* 0) *changes the class to* 1.

42: *Changing features*: 16-8 (10 *to* 0) 16-9 (10 *to* 0) 17-8 (10 *to* 0) 17-9 (6 *to* 0) 17-11 (1 *to* 0) 18-8 (10 *to* 0) 18-9 (6 *to* 0) 18-11 (1 *to* 0) 19-8 (10 *to* 0) 19-9 (10 *to* 0) 19-10 (2 *to* 0) 19-11 (4 *to* 0) *changes the class to* 1.

43: *Changing features*: 16-12 (7 *to* 1) 17-12 (9 *to* 1) 17-14 (4 *to* 10) 17-15 (0 *to* 1) 18-12 (9 *to* 1) 18-14 (6 *to* 10) 18-15 (9 *to* 1) 19-12 (9 *to* 1) 19-15 (10 *to* 1) *changes the class to* 1.

44: *Changing features*: 16-17 (2 *to* 0) 16-18 (10 *to* 0) 16-19 (10 *to* 0) 17-17 (8 *to* 0) 17-18 (10 *to* 0) 17-19 (4 *to* 0) 18-16 (9 *to* 1) 18-17 (8 *to* 0) 18-18 (1 *to* 0) 19-16 (9 *to* 6) 19-17 (2 *to* 0) *changes the class to* 1.

45: *Changing features*: 16-20 (5 *to* 0) *changes the class to* 1.

51: *Changing features*: 20-7 (2 *to* 0) *changes the class to* 1.

52: *Changing features*: 20-8 (9 *to* 0) 20-9 (10 *to* 0) 20-10 (10 *to* 0) 20-11 (10 *to* 0) 21-8 (3 *to* 0) 21-9 (7 *to* 0) 21-10 (10 *to* 0) 21-11 (10 *to* 0) *changes the class to* 1.

53: *Changing features*: 20-13 (10 *to* 1) 20-14 (10 *to* 0) 20-15 (9 *to* 0) 21-12 (10 *to* 5) 21-13 (6 *to* 0) 21-14 (4 *to* 0) 21-15 (2 *to* 0) *changes the class to* 1.

54: *Changing features*: 20-16 (5 *to* 0) *changes the class to* 1.

Applying changes from the contrastive explanation above and changing pixel values in the image containing numeral 6 we obtain the image in Fig.10.

Fig.10. Transition from 6 to 1 in MNIST

The end loop with which the numeral 6 is identified has been removed and the rest straightened to form the numeral 1. The contrastive explanation for converting 6 to 2 from each of the sub-lattices is as follows:

13: *Changing features*: 4-14 (7 *to* 0) 4-15 (10 *to* 0) 5-12 (1 *to* 0) 5-13 (7 *to* 0) 5-14 (10 *to* 0) 5-15 (10 *to* 0) 6-12 (2 *to* 0) 6-13 (10 *to* 1) 6-14 (10 *to* 6) 6-15 (10 *to* 8) 7-12 (8 *to* 6) 7-13 (10 *to* 9) 7-15 (3 *to* 10) *changes the class to* 2.

14: *Changing features*: 4-16 (10 *to* 0) 4-17 (8 *to* 0) 5-16 (4 *to* 7) 5-17 (2 *to* 0) 6-16 (3 *to* 10) 6-17 (0 *to* 8) 7-16 (0 *to* 9) 7-17 (0 *to* 10) 7-18 (0 *to* 5) *changes the class to* 2.

22: *Changing features*: 8-10 (0 *to* 6) 8-11 (5 *to* 9) 9-9 (0 *to* 3) 9-10 (7 *to* 10) 9-11 (9 *to* 10) 10-9 (3 *to* 7) 11-9 (5 *to* 1) 11-10 (10 *to* 5) 11-11 (10 *to* 2) *changes the class to* 2.

23: *Changing features*: 8-12 (10 *to* 0) 8-13 (7 *to* 0) 8-14 (1 *to* 0) 9-12 (8 *to* 0) 9-13 (1 *to* 0) 10-12 (5 *to* 0) 10-13 (1 *to* 0) 11-12 (4 *to* 0) *changes the class to* 2.

24: *Changing features*: 8-16 (0 *to* 5) 8-17 (0 *to* 10) 8-18 (0 *to* 8) 9-16 (0 *to* 1) 9-17 (0 *to* 10) 9-18 (0 *to* 9) 10-16 (0 *to* 1) 10-17 (0 *to* 10) 10-18 (0 *to* 10) 10-19 (0 *to* 4) 11-17 (4 *to* 8) 11-18 (5 *to* 10) *changes the class to* 2.

32: *Changing features*: 12-9 (7 *to* 0) 12-10 (10 *to* 0) 12-11 (10 *to* 0) 13-8 (3 *to* 0) 13-9 (10 *to* 0) 13-10 (10 *to* 0) 13-11 (8 *to* 0) 14-8 (4 *to* 0) 14-9 (10 *to* 0) 14-10 (9 *to* 0) 14-11 (5 *to* 2) 15-8 (10 *to* 0) 15-9 (10 *to* 2) 15-10 (5 *to* 9) 15-11 (0 *to* 10) *changes the class to* 2.

33: *Changing features*: 12-12 (2 *to* 0) 12-15 (4 *to* 0) 13-12 (1 *to* 0) 13-14 (1 *to* 0) 13-15 (6 *to* 0) 14-13 (1 *to* 0) 14-14 (6 *to* 0) 14-15 (10 *to* 0) 15-13 (7 *to* 0) 15-14 (10 *to* 0) 15-15 (9 *to* 1) *changes the class to* 2.

34: *Changing features*: 12-16 (4 *to* 0) 12-19 (10 *to* 5) 13-16 (10 *to* 0) 13-17 (10 *to* 8) 13-18 (9 *to* 10) 13-19 (8 *to* 5) 14-16 (9 *to* 1) 14-17 (6 *to* 10) 14-18 (1 *to* 10) 15-16 (5 *to* 3) 15-17 (0 *to* 10) 15-18 (1 *to* 10) *changes the class to* 2.

35: *Changing features*: 12-20 (10 *to* 2) 12-21 (1 *to* 0) 13-20 (10 *to* 0) 13-21 (7 *to* 0) 13-22 (1 *to* 0) 14-20 (10 *to* 0) 14-21 (8 *to* 0) 14-22 (1 *to* 0) 15-20 (9 *to* 0) 15-21 (5 *to* 0) *changes the class to* 2.

41: *Changing features*: 16-7 (4 *to* 0) 18-6 (0 *to* 7) 18-7 (5 *to* 10) 19-6 (0 *to* 10) 19-7 (2 *to* 8) *changes the class to* 2.

42: *Changing features*: 16-8 (10 *to* 3) 16-10 (0 *to* 10) 16-11 (0 *to* 5) 17-9 (6 *to* 10) 17-10 (0 *to* 7) 17-11 (1 *to* 0) 18-8 (10 *to* 9) 18-9 (6 *to* 4) 18-11 (1 *to* 0) 19-8 (10 *to* 3) 19-9 (10 *to* 0) 19-11 (4 *to* 6) *changes the class to* 2.

43: *Changing features*: 16-12 (7 *to* 2) 16-13 (10 *to* 0) 16-14 (10 *to* 0) 16-15 (2 *to* 8) 17-12 (9 *to* 0) 17-13 (10 *to* 0) 17-14 (4 *to* 8) 17-15 (0 *to* 10) 18-12 (9 *to* 2) 18-13 (10 *to* 8) 18-14 (6 *to* 10) 18-15 (9 *to* 8) 19-14 (10 *to* 5) 19-15 (10 *to* 1) *changes the class to* 2.

44: *Changing features*: 16-16 (0 *to* 10) 16-17 (2 *to* 10) 17-16 (1 *to* 10) 17-17 (8 *to* 10) 17-18 (10 *to* 6) 17-19 (4 *to* 2) 18-16 (9 *to* 10) 18-18 (1 *to* 0) *changes the class to* 2.

45: *Changing features*: 16-20 (5 *to* 8) 16-21 (0 *to* 8) 16-22 (0 *to* 5) 16-23 (0 *to* 2) *changes the class to* 2.

46: *Changing features*: 16-24 (0 *to* 1) *changes the class to* 2.

51: *Changing features*: 20-6 (0 *to* 10) 20-7 (2 *to* 8) 21-6 (0 *to* 8) 21-7 (0 *to* 10) 22-7 (0 *to* 5) *changes the class to* 2.

52: *Changing features*: 20-8 (9 *to* 8) 20-9 (10 *to* 8) 21-8 (3 *to* 10) 21-9 (7 *to* 10) 21-10 (10 *to* 7) 21-11 (10 *to* 4) 22-8 (0 *to* 4) *changes the class to* 2.

53: *Changing features*: 20-13 (10 *to* 4) 20-14 (10 *to* 0) 20-15 (9 *to* 0) 21-12 (10 *to* 1) 21-13 (6 *to* 0) 21-14 (4 *to* 0) 21-15 (2 *to* 0) *changes the class to* 2.

54: *Changing features*: 20-16 (5 *to* 2) *changes the class to* 2.

Applying changes from the contrastive explanation above and changing pixel values in the image containing numeral 6 we obtain the image in Fig.11.

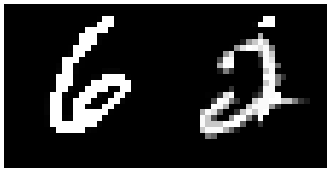

Fig.11. Transition from 6 to 2 in MNIST

The end loop with which the numeral 6 is identified has been revised with a tail and a curve added on top to form the numeral 2.

In a similar fashion, the contrastive explanations were obtained for converting 6 to other numerals from each of the sub-lattices and are brought out in Fig.12 to 17.

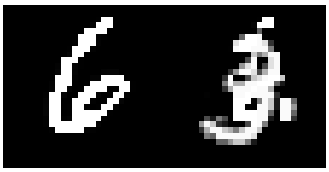

Fig.12. Transition from 6 to 3 in MNIST

The end loop with which the numeral 6 is identified has been removed and two curves in the opposite direction are added to form the numeral 3.

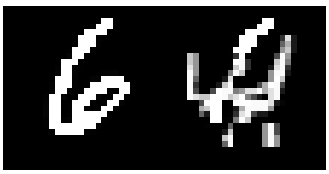

Fig.13. Transition from 6 to 4 in MNIST

The end loop with which the numeral 6 is identified has been removed and a bucket-like curve is added with a bottom tail to form the numeral 4.

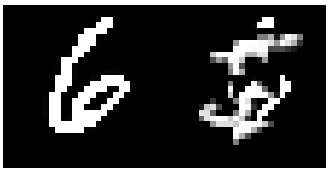

Fig.14. Transition from 6 to 5 in MNIST

The end loop with which the numeral 6 is identified is reversed, opening out in the opposite direction with a horizontal line on top to form the numeral 5.

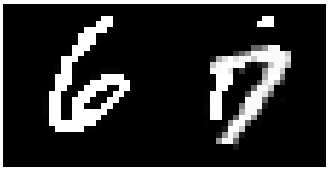

Fig.15. Transition from 6 to 7 in MNIST

The end loop with which the numeral 6 is identified is discarded to be replaced by the slant line and horizontal line on top to form the numeral 7.

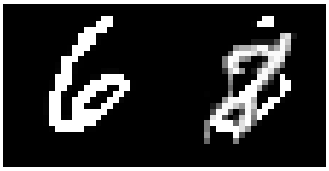

Fig.16. Transition from 6 to 8 in MNIST

The end loop with which the numeral 6 is identified has been removed and the rest is converted to another loop on top to form the numeral 8.

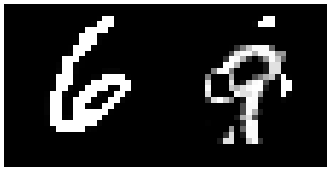

Fig.17. Transition from 6 to 9 in MNIST

The end loop with which the numeral 6 is identified has been removed and a new loop is formed on top to form the numeral 9.

7. COMPARISON BETWEEN LATTICE EXPLANATION AND DEEPLIFT ON THE MNIST DATASET

Deep Learning Important FeaTures (DeepLIFT) [20] is an improvement over its first form known as "Gradient * Input" method. It decomposes output prediction of a network on an input by backpropagating the contributions of all neurons to every feature of the input. It compares the activation of each neuron to the reference activation and assigns contribution scores according to the difference. It can generate contrastive explanation by optionally considering positive and negative contributions. This work was applied on models trained on the MNIST dataset and Fig.18 shows the masks generated by DeepLIFT for explaining the transition from 8 to 3 and 8 to 6.

Fig.18. Masks from 8 to 3 and 8 to 6 for MNIST in DeepLIFT

The lattice contrastive explanation for the transition from 8 to 3 is as follows:

13: *Changing features*: 5-12 (0 *to* 6) 5-13 (0 *to* 6) 5-14 (0 *to* 5) 5-15 (0 *to* 2) 6-12 (0 *to* 10) 6-13 (0 *to* 10) 6-14 (0 *to* 10) 6-15 (1 *to* 10) 7-14 (7 *to* 8) 7-15 (8 *to* 10) *changes the class to* 3.

14: *Changing features*: 6-16 (1 *to* 8) 6-17 (1 *to* 4) 6-19 (3 *to* 0) 7-18 (5 *to* 3) 7-19 (7 *to* 0) *changes the class to* 3.

15: *Changing features*: 5-20 (8 *to* 6) 5-21 (9 *to* 0) 5-22 (1 *to* 0) 6-21 (8 *to* 6) 7-21 (7 *to* 10) 7-22 (0 *to* 2) *changes the class to* 3.

20: *Changing features*: 8-3 (0 *to* 1) *changes the class to* 3.

21: *Changing features*: 8-4 (0 *to* 2) 8-5 (0 *to* 1) 8-6 (0 *to* 1) 8-7 (0 *to* 1) *changes the class to* 3.

22: *Changing features*: 8-11 (4 *to* 0) 9-10 (5 *to* 0) 9-11 (9 *to* 0) 10-10 (5 *to* 0) 10-11 (10 *to* 0) 11-10 (2 *to* 0) 11-11 (8 *to* 2) *changes the class to* 3.

23: *Changing features*: 8-12 (10 *to* 9) 8-13 (10 *to* 5) 8-14 (10 *to* 5) 8-15 (9 *to* 5) 9-12 (10 *to* 0) 9-13 (6 *to* 0) 9-14 (5 *to* 0) 10-12 (8 *to* 0) 10-13 (1 *to* 0) 11-12 (10 *to* 0) 11-13 (6 *to* 0) 11-15 (1 *to* 0) *changes the class to* 3.

24: *Changing features*: 8-16 (6 *to* 5) 8-17 (9 *to* 7) 9-17 (1 *to* 0) 10-16 (1 *to* 0) 10-17 (6 *to* 3) 10-18 (10 *to* 9) 10-19 (9 *to* 10) 11-

16 (7 *to* 3) 11-17 (10 *to* 7) 11-18 (9 *to* 10) 11-19 (4 *to* 10) *changes the class to* 3.

32: *Changing features*: 12-10 (0 *to* 2) 12-11 (6 *to* 10) 13-10 (0 *to* 5) 13-11 (5 *to* 10) 14-10 (0 *to* 2) 14-11 (4 *to* 7) 15-11 (6 *to* 0) *changes the class to* 3.

33: *Changing features*: 12-12 (10 *to* 0) 12-13 (9 *to* 0) 13-12 (10 *to* 7) 13-13 (10 *to* 7) 15-15 (7 *to* 10) *changes the class to* 3.

34: *Changing features*: 12-18 (4 *to* 1) 13-17 (6 *to* 10) 13-18 (1 *to* 5) 14-16 (7 *to* 10) 14-17 (1 *to* 10) 14-18 (0 *to* 6) 15-16 (0 *to* 9) 15-17 (0 *to* 10) 15-18 (0 *to* 6) *changes the class to* 3.

35: *Changing features*: 12-20 (0 *to* 8) 13-20 (0 *to* 1) *changes the class to* 3.

41: *Changing features*: 19-7 (2 *to* 3) *changes the class to* 3.

42: *Changing features*: 16-10 (6 *to* 0) 16-11 (10 *to* 0) 17-8 (1 *to* 0) 17-9 (4 *to* 0) 17-10 (10 *to* 0) 17-11 (10 *to* 0) 18-8 (3 *to* 0) 18-9 (10 *to* 0) 18-10 (10 *to* 0) 18-11 (4 *to* 0) 19-8 (9 *to* 1) 19-9 (10 *to* 5) *changes the class to* 3.

43: *Changing features*: 16-12 (10 *to* 1) 16-13 (10 *to* 1) 16-14 (10 *to* 1) 16-15 (7 *to* 1) 17-12 (8 *to* 0) 17-13 (6 *to* 0) 17-14 (10 *to* 0) 17-15 (8 *to* 0) 18-13 (3 *to* 0) 18-14 (10 *to* 0) 18-15 (6 *to* 0) 19-12 (1 *to* 0) 19-13 (8 *to* 0) 19-14 (10 *to* 0) 19-15 (5 *to* 3) *changes the class to* 3.

44: *Changing features*: 16-16 (0 *to* 3) 16-17 (0 *to* 8) 16-18 (0 *to* 10) 16-19 (0 *to* 10) 17-18 (0 *to* 7) 17-19 (0 *to* 10) 18-18 (0 *to* 5) 18-19 (0 *to* 10) 19-18 (0 *to* 7) 19-19 (0 *to* 10) *changes the class to* 3.

45: *Changing features*: 16-20 (0 *to* 10) 16-21 (0 *to* 10) 16-22 (0 *to* 4) 17-20 (0 *to* 10) 17-21 (0 *to* 10) 17-22 (0 *to* 6) 18-20 (0 *to* 10) 18-21 (0 *to* 10) 18-22 (0 *to* 4) 19-20 (0 *to* 10) 19-21 (0 *to* 10) 19-22 (0 *to* 2) *changes the class to* 3.

51: *Changing features*: 20-7 (5 *to* 0) 21-7 (5 *to* 6) 22-7 (6 *to* 4) 23-7 (4 *to* 1) *changes the class to* 3.

52: *Changing features*: 20-8 (10 *to* 1) 20-9 (7 *to* 0) 21-8 (10 *to* 5) 21-9 (7 *to* 5) 21-10 (0 *to* 5) 21-11 (3 *to* 5) 22-9 (6 *to* 10) 22-10 (7 *to* 10) 23-11 (9 *to* 10) *changes the class to* 3.

53: *Changing features*: 20-12 (4 *to* 0) 20-13 (9 *to* 0) 20-14 (10 *to* 3) 20-15 (4 *to* 9) 21-12 (10 *to* 5) 21-13 (9 *to* 8) 21-14 (6 *to* 10) 21-15 (0 *to* 10) 22-13 (3 *to* 10) 22-14 (0 *to* 10) 22-15 (0 *to* 9) 23-12 (5 *to* 8) 23-13 (0 *to* 8) 23-14 (0 *to* 5) 23-15 (0 *to* 1) *changes the class to* 3.

54: *Changing features*: 20-16 (0 *to* 10) 20-17 (0 *to* 5) 21-16 (0 *to* 8) *changes the class to* 3.

61: *Changing features*: 24-7 (0 *to* 4) *changes the class to* 3.

62: *Changing features*: 24-8 (4 *to* 0) 24-9 (9 *to* 3) 24-10 (7 *to* 6) *changes the class to* 3.

64: *Changing features*: 24-16 (0 *to* 3) 24-17 (0 *to* 1) *changes the class to* 3.

Fig.19. Contrastive explanation from Lattice for 8 to 3 in MNIST

Applying changes from the contrastive explanation above and changing pixel values in the image containing numeral 6 we obtain the image in Fig.19.

The lattice contrastive explanation for the transition from 8 to 6 is as follows:

04: *Changing features*: 2-16 (0 *to* 1) 2-17 (0 *to* 8) 2-18 (0 *to* 9) 3-16 (0 *to* 9) 3-17 (0 *to* 10) 3-18 (0 *to* 7) *changes the class to* 6.

13: *Changing features*: 4-15 (0 *to* 2) 5-14 (0 *to* 3) 5-15 (0 *to* 9) 6-13 (0 *to* 1) 6-14 (0 *to* 9) 6-15 (1 *to* 7) 7-12 (6 *to* 0) 7-14 (7 *to* 8) 7-15 (8 *to* 0) *changes the class to* 6.

14: *Changing features*: 4-16 (0 *to* 10) 4-17 (0 *to* 8) 5-16 (0 *to* 4) 5-17 (0 *to* 2) 6-16 (1 *to* 3) 6-17 (1 *to* 0) 6-18 (1 *to* 0) 6-19 (3 *to* 0) 7-16 (10 *to* 0) 7-17 (10 *to* 0) 7-18 (5 *to* 0) 7-19 (7 *to* 0) *changes the class to* 6.

22: *Changing features*: 8-11 (4 *to* 0) 9-10 (5 *to* 0) 9-11 (9 *to* 0) 10-10 (5 *to* 0) 10-11 (10 *to* 5) 11-10 (2 *to* 0) 11-11 (8 *to* 7) *changes the class to* 6.

23: *Changing features*: 8-13 (10 *to* 7) 8-14 (10 *to* 1) 8-15 (9 *to* 0) 9-12 (10 *to* 8) 9-13 (6 *to* 1) 9-14 (5 *to* 0) 10-12 (8 *to* 5) 11-12 (10 *to* 4) 11-13 (6 *to* 0) 11-15 (1 *to* 0) *changes the class to* 6.

24: *Changing features*: 8-16 (6 *to* 0) 8-17 (9 *to* 0) 8-18 (10 *to* 0) 8-19 (10 *to* 0) 9-17 (1 *to* 0) 9-18 (9 *to* 0) 9-19 (10 *to* 0) 10-16 (1 *to* 0) 10-17 (6 *to* 0) 10-18 (10 *to* 0) 10-19 (9 *to* 0) 11-16 (7 *to* 0) 11-17 (10 *to* 4) 11-18 (9 *to* 5) 11-19 (4 *to* 5) *changes the class to* 6.

31: *Changing features*: 15-7 (0 *to* 2) *changes the class to* 6.

32: *Changing features*: 12-10 (0 *to* 2) 12-11 (6 *to* 10) 13-10 (0 *to* 6) 13-11 (5 *to* 8) 14-10 (0 *to* 10) 14-11 (4 *to* 3) 15-9 (0 *to* 1) 15-10 (0 *to* 10) 15-11 (6 *to* 3) *changes the class to* 6.

33: *Changing features*: 12-12 (10 *to* 2) 12-13 (9 *to* 0) 12-14 (3 *to* 0) 12-15 (8 *to* 4) 13-12 (10 *to* 1) 13-13 (10 *to* 0) 13-14 (10 *to* 1) 13-15 (10 *to* 6) 14-12 (10 *to* 0) 14-13 (10 *to* 1) 14-14 (10 *to* 6) 15-12 (10 *to* 0) 15-13 (10 *to* 7) 15-15 (7 *to* 9) *changes the class to* 6.

34: *Changing features*: 12-16 (10 *to* 2) 12-17 (10 *to* 1) 12-18 (4 *to* 0) 13-17 (6 *to* 9) 13-18 (1 *to* 4) 14-16 (7 *to* 0) 14-17 (1 *to* 3) 14-18 (0 *to* 9) 14-19 (0 *to* 2) 15-18 (0 *to* 5) 15-19 (0 *to* 7) *changes the class to* 6.

35: *Changing features*: 12-20 (0 *to* 10) 12-21 (0 *to* 1) 13-20 (0 *to* 10) 13-21 (0 *to* 7) 13-22 (0 *to* 1) 14-20 (0 *to* 10) 14-21 (0 *to* 8) 14-22 (0 *to* 1) 15-20 (0 *to* 9) 15-21 (0 *to* 5) *changes the class to* 6.

41: *Changing features*: 16-7 (0 *to* 4) 17-7 (0 *to* 5) 18-7 (0 *to* 5) *changes the class to* 6.

42: *Changing features*: 16-8 (0 *to* 10) 16-9 (0 *to* 10) 16-10 (6 *to* 0) 16-11 (10 *to* 0) 17-8 (1 *to* 10) 17-9 (4 *to* 6) 17-10 (10 *to* 0) 17-11 (10 *to* 1) 18-8 (3 *to* 10) 18-9 (10 *to* 6) 18-10 (10 *to* 0) 18-11 (4 *to* 1) 19-8 (9 *to* 10) 19-10 (3 *to* 2) 19-11 (0 *to* 4) *changes the class to* 6.

44: *Changing features*: 16-17 (0 *to* 2) 16-18 (0 *to* 10) 16-19 (0 *to* 10) 17-16 (0 *to* 1) 17-17 (0 *to* 8) 17-18 (0 *to* 10) 17-19 (0 *to* 4) 18-16 (0 *to* 9) 18-17 (0 *to* 8) 18-18 (0 *to* 1) 19-16 (0 *to* 9) 19-17 (0 *to* 2) *changes the class to* 6.

45: *Changing features*: 16-20 (0 *to* 5) *changes the class to* 6.

52: *Changing features*: 20-8 (10 *to* 9) 20-9 (7 *to* 10) 20-10 (0 *to* 10) 20-11 (0 *to* 10) 21-8 (10 *to* 3) 21-10 (0 *to* 10) 21-11 (3 *to*

10) 22-8 (10 *to* 0) 22-9 (6 *to* 0) 22-10 (7 *to* 0) 22-11 (10 *to* 0) 23-8 (10 *to* 0) 23-9 (10 *to* 0) 23-10 (10 *to* 0) 23-11 (9 *to* 0) *changes the class to* 6.

53: *Changing features*: 20-12 (4 *to* 10) 20-13 (9 *to* 10) 20-15 (4 *to* 9) 21-13 (9 *to* 6) 21-14 (6 *to* 4) 21-15 (0 *to* 2) 22-12 (10 *to* 0) 22-13 (3 *to* 0) 23-12 (5 *to* 0) *changes the class to* 6.

54: *Changing features*: 20-16 (0 *to* 5) *changes the class to* 6.

Applying changes from the contrastive explanation above and changing pixel values in the image containing numeral 6 we obtain the image in Fig.20.

Fig.20. Contrastive explanation from Lattice for 8 to 6 in MNIST

This comparison proves that lattice-based explanations are in line with the ones produced by DeepLIFT.

8. CONCLUSION AND FUTURE WORK

This approach of breaking images into parts, building sub-lattices for each part and generating the complete explanation provides a good beginning to use lattice-based explanations for images. Generating satisfactory explanations for the simple dataset and the MNIST dataset clearly proves that the lattice-based approach to image explanations is feasible. Comparing image masks generated from contrastive explanation of the lattice with a well-known technique like DeepLIFT proves its credibility and correctness. Yet, to improve the accuracy of some of the image masks, it is worth building the lattice without splitting the image. It imposes a large memory intensive computation that can be studied on a distributed computing framework. Alternate strategies like dynamic construction of relevant parts of the lattice or pruning unnecessary nodes of the lattice may also prove useful.

REFERENCES

[1] C. Rudin, "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead", *Nature Machine Intelligence*, Vol. 1, No. 5, pp. 206-215, 2019.

[2] Alejandro Barredo Arrieta, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila and Francisco Herrera, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI", *Information Fusion*, Vol. 58, pp. 82-115, 2020.

[3] M.T. Ribeiro and C. Guestrin "Why Should I Trust You?: Explaining the Predictions of Any Classifier", *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135-1144, 2016.

[4] Alvarez-Melis and Tommi S. Jaakkola, "On the Robustness of Interpretability Methods", *arXiv preprint arXiv:*1806.08049, 2018.

[5] G. Visani, Alessandro Poluzzi and Davide Capuzzo, "Statistical Stability Indices for LIME: Obtaining Reliable Explanations for Machine Learning Models", *Journal of the Operational Research Society*, Vol. 73, No. 1, pp. 91-101, 2022.

[6] Marzyeh Ghassemi, Luke Oakden-Rayner and Andrew L Beam, "The False Hope of Current Approaches to Explainable Artificial Intelligence in Healthcare", *The Lancet Digital Health*, Vol. 3, No. 11, pp. 745-750, 2021.

[7] S.M. Lundberg and S.I. Lee, "A Unified Approach to Interpreting Model Predictions", *Advances in Neural Information Processing Systems*, Vol. 30, pp. 4765-4774, 2017.

[8] R.R. Selvaraju and D. Batra, "Grad-CAM: Why did you say that?", *CoRR abs*/1611.07450, pp.1-13, 2016.

[9] D. Smilkov and M. Wattenberg, "SmoothGrad: Removing Noise by Adding Noise", *CoRR abs*/1706.03825, pp. 1-12, 2017.

[10] J.T. Springenberg and M.A. Riedmiller, "Striving for Simplicity: The All-Convolutional Net", *Proceedings of International Conference on Machine Learning*, 2015.

[11] M.L. Leavitt and A. Morcos, "Towards falsifiable interpretability research", Proceedings of International Conference on Neural Information Processing Systems ML Retrospectives, Surveys and Meta-Analyses, pp. 1-13, 2020.

[12] M. Sundararajan and Q. Yan, "Axiomatic Attribution for Deep Networks", *Proceedings of International Conference on Machine Learning*, pp. 3319-3328, 2017.

[13] J. Adebayo and B. Kim, "Sanity Checks for Saliency Maps", *Proceedings of International Conference on Neural Computing*, pp. 9525-9536, 2018.

[14] Venkatsubramaniam Bhaskaran and Pallav Kumar Baruah, "A Novel Approach to Explainable AI Using Formal Concept Lattice", *International Journal of Innovative Technology and Exploring Engineering*, Vol. 11, No. 7, pp. 36-48, 2022.

[15] A. Sangroya and L. Vig, "Guided-LIME: Structured Sampling based Hybrid Approach towards Explaining Blackbox Machine Learning Models", *Proceedings of International Conference on Machine Learning*, pp. 1-16, 2020.

[16] A. Sangroya and M. Rastogi, "Using Formal Concept Analysis to Explain Black Box Deep Learning Classification Models", *Proceedings of International Conference on Artificial Intelligence*, pp. 19-26, 2019.

[17] UCI, "UC Irvine Machine Learning Repository", Available at: https://archive.ics.uci.edu/ml/index.php, Accessed at 2022.

[18] R. Wille, "*Concept Lattices and Conceptual Knowledge Systems*", Computers and Mathematics with Applications, 1992.

[19] UCI, "UCI Car Evaluation DataSet", Available at: https://archive.ics.uci.edu/ml/datasets/Car+Evaluation, Accessed at 2022.

[20] Avanti Shrikumar, Peyton Greenside and Anshul Kundaje, "Learning Important Features through propagating Activation Differences", *Proceedings of International Conference on Machine Learning*, pp. 3145-3153, 2017.

[21] Jianqing Fan, Cong Ma and Yiqiao Zhong, "A Selective Overview of Deep Learning", *Proceedings of International Conference on Machine Learning*, pp. 98-104, 2019.

[22] Laurens Van Der Maaten and Geoffrey Hinton, "Visualizing Data using t-SNE", *Journal of Machine Learning Research*, Vol.12, No. 2, pp. 1-15, 2008.

[23] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", *Proceedings of International Conference on Machine Learning*, pp. 1-5, 2014.

[24] Matthew D. Zeiler and Rob Fergus, "Visualizing and Understanding Convolutional Networks", *Proceedings of European Conference on Computer Vision*, pp. 1-8, 2014.

[25] Karen Simonyan, Andrea Vedaldi and Andrew Zisserman, "Deep Inside Convolutional Network: Visualizing image classification models and Saliency Maps", *Proceedings of International Conference on Machine Learning*, pp. 1-9, 2014.

[26] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, "Learning Deep Features for Discriminative Localization", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2921-2929, 2016.

[27] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization", *Proceedings of IEEE International Conference on Computer Vision*, pp. 618-626, 2017.

[28] A. Chattopadhay, A. Sarkar, P. Howlader and V.N. Balasubramanian, "Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks", *Proceedings of IEEE Winter Conference on Applications of Computer Vision*, pp. 839-847, 2018.

[29] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viegas and Martin Wattenberg, "SmoothGrad: Removing Noise by Adding Noise", CoRR abs/1706.03825, pp.1-9, 2017.

[30] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler and Fernanda Viegas. "Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)", *Proceedings of International Conference on Machine Learning*, pp. 2668-2677. 2018.