Visual Explanations for DNNs with Contextual Importance

Sule Anjomshoae(✉), Lili Jiang, and Kary Främling

Department of Computing Science, Umeå University, Umeå, Sweden
{sulea,lili.jiang,kary.framling}@cs.umu.se

Abstract. Autonomous agents and robots with vision capabilities powered by machine learning algorithms such as Deep Neural Networks (DNNs) are taking place in many industrial environments. While DNNs have improved the accuracy in many prediction tasks, it is shown that even modest disturbances in their input produce erroneous results. Such errors have to be detected and dealt with for making the deployment of DNNs secure in real-world applications. Several explanation methods have been proposed to understand the inner workings of these models. In this paper, we present how Contextual Importance (CI) can make DNN results more explainable in an image classification task without peeking inside the network. We produce explanations for individual classifications by perturbing an input image through over-segmentation and evaluating the effect on a prediction score. Then the output highlights the most contributing segments for a prediction. Results are compared with two explanation methods, namely mask perturbation and LIME. The results for the MNIST hand-written digit dataset produced by the three methods show that CI provides better visual explainability.

Keywords: Deep learning · Explainable artificial intelligence · Image classification · Contextual importance

1 Introduction

Deep Neural Networks (DNNs) have improved the accuracy of prediction tasks in many applications including object recognition and natural language processing. However, DNNs inability to show their reasoning is hindering the use of these models in safety-critical systems such as in autonomous driving and medical domains. Moreover, their susceptibility to adversarial inputs (i.e., image and audio data is modified in a subtle way that is undetectable to humans) easily leads to incorrect predictions. The existing work on adversarial attacks on DNNs shows graffiti and art stickers cause to misclassify a turn right sign as a stop sign, and a stop sign as a 45-speed limit sign [10]. Explanations for such cases can help to evaluate the model and identify the patterns which the model has learned during training. This is a reason for the recent increase in research about explainable black-box algorithms as one means of working toward more robust and interpretable DNNs [2,6,24].

D. Calvaresi et al. (Eds.): EXTRAAMAS 2021, LNAI 12688, pp. 83–96, 2021.
https://doi.org/10.1007/978-3-030-82017-6_6

In this work, we present the concept of contextual importance to provide visual justifications for image classification on a standard DNN. The DNN's convolutional layer applies sliding filters to capture shift-invariant patterns and learn robust features to make predictions. Given the predicted class index and the probability, we investigate the most important features contributing to the prediction using a perturbation method. In this way, we present the visual representation of contextual importance value to justify the image classification results. Our main contributions are summarized as follows:

- We introduce contextual importance explanations for image classification tasks that can be applied to any CNN-based network without requiring alteration to the model.
- We present contrastive explanations that highlight class-discriminating features for multiple class predictions.
- We show examples of visual explanations on visibly distorted and noisy images.
- We present explanations in high confidence cases for incorrect predictions to help diagnose features contributing to misclassication.

2 Related Work

Researchers have been focusing on integrating explanation facilities into computer vision algorithms [20,21,23]. Generally, these explanation methods can be categorized as interpretable models and post-hoc explanations. Interpretable models focus on the internal functioning of the models; they analyze the interaction between neurons and what each neuron has learned. Post-hoc methods explain instance-specific predictions on the basis of how each feature influences the final outcome. In general, both approaches can have some limitations and strengths. The results of interpretable models are directly explainable without requiring another model to generate explanations. However, they restraint the model to increase comprehensibility, as a result, these may oversimplify the problem at hand. In contrast, the post-hoc explanations are not restricting the model but they may be limited in their approximate nature [8].

Several interpretable models have been proposed to make the complex black-box models more understandable. Some of the techniques for understanding and diagnosing DNNs include gradients, visual analytics, and decomposition methods. Gradient methods highlight the unit changes and emphasize the important features or regions in an image. In this way, it is possible to find the prototypes that have the highest probability to be predicted as a certain class of a trained DNN. Considering this, Li *et al.,* proposed an interpretable neural network architecture whose predictions are based on the similarity of an input to a small set of prototypes learned during training [14]. This approach is able to produce artificial images that maximize a neuron activation or class confidence value. Nguyen *et al.,* presented the activation maximization method which synthesizes an image based on high activation on a neuron, then reveals the features learned by each neuron in an interpretable way [17]. Some works proposed visual analysis by clustering important neurons based on the features and the interactions between them [12,15]. Furthermore, decomposition methods such as layer-wise relevance methods are presented to analyze which pixels are contributing to what extent in a classification result [5]. Yoo *et al.,* proposed a regularization technique using forward and backward interacted-activation by defining a sum of layer-wise differences between neuron activations. This computation between forward and backward directions provide some kind of interpretability for DNNs [22].

On the other hand, post-hoc explanations are mostly proposed based on saliency maps and gradient visualizations methods in computer vision tasks. Saliency maps are being created in various ways. One way is to visualize by going backward through the inverted network from an output of interest. It highlights the discriminative features of the image with respect to the given class [21]. Another method uses class activation mapping with the gradients of a target input in the final convolutional layer to produce a rough localization map highlighting the important regions [19]. This method is further developed for explaining occurrences of multiple object instances in a single image [7]. In another work, class activation mapping is combined with the global average pooling layer to visualize class-specific image regions for revealing the attention map of DNNs on an image [25].

Moreover, some methods suggested generating explanations by approximating a black-box model by a simple model locally with the perturbations of the original instance. Then, they present the super-pixels with the highest positive weights as an explanation [18]. In our approach, contextual importance measures the influence of an individual feature on a prediction result by perturbing inputs without transforming the model into an interpretable one. The effects are then visualized to explain the outcome based on the main contributing group of pixels (i.e., interpretable regions). The visualization shows the relative importance of each region considering the whole image for the current prediction. Contextual importance explains the prediction results by analysing the effect on the output of the model, therefore remains faithful to the original model.

3 Method

The DNN (Deep Neural Network) architecture studied here is composed of three convolutional layers each followed by batch normalization, ReLu, max pooling, and finally fully connected layer as seen in Fig. 1. A convolution layer applies a sliding convolutional filter to an input image. The weighted sum of the input pixels within the window produces an output pixel at each point allowing the convolutional layer to learn visual patterns and features. Batch normalization reduces the sensitivity to network initialization and speeds up the training of the convolutional layer. ReLU activation function performs a threshold operation for each element by setting negative values to zero. Following this, the max-pooling layer performs down-sampling by dividing the input into sets of non-overlapping rectangles, and outputs the maximum for each region. Finally, a high-level reasoning is made through the fully connected layer in the network. This hierarchical architecture allows DNNs to extract increasingly abstract features from the first layer to the final layer.

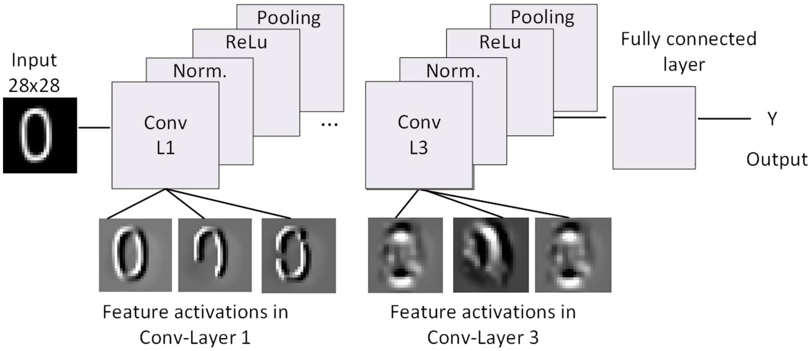

Fig. 1. DNN architecture for the recognition of the hand-written digit dataset

3.1 Contextual Importance

Contextual importance (CI) method is one of the earlier studies to address the post-hoc explainability for predictions made by black-box algorithms, specifically for a tabular data type [11]. However, the CI's potential for image explanations has yet to be investigated. In this study, we propose generating visual explanations for DNNs through the contextual importance. This concept originates from the idea that the set of input features forms the context, and given this, the importance of feature is dependent on other feature values. Therefore, contextual importance indicates the degree of significance of a feature value (or set of feature values) when changes are made to that particular value(s) while the rest of the inputs remain constant. This is a model-agnostic approach which makes it possible to explain the outcomes of both linear and non-linear learning models as presented in [3,4].

Feature importance usually signifies a measure for how much one feature affects the outcome when taking into account the whole dataset. Building an explanation method is simple in the case of a purely linear model, where every feature's importance is constant and irrelevant to the other feature values. For linear models such as the weighted sum, the weight directly expresses the importance of each feature and the combined importance of several features corresponds to the sum of the weights. Neural networks are mainly useful for tasks where linear models are not sufficiently expressive. When speaking about post-hoc explanations of non-linear methods, the feature importance might be specific for the current set of input values.

When dealing with non-linear models, it becomes non-trivial how to define feature importance. It seems like most current model-agnostic methods use the local gradient as an indicator of the contextual importance [9]. The definition of contextual importance is based on a different principle. Rather than observing how much an output value changes with fixed amount of small perturbations to the current input value of the studied feature(s), we study how much perturbations over the whole range of possible feature values affects the output range. *CI* is then the ratio of the observed output range and the greatest possible output range. If the observed output range is greater for one feature than another, then the former is more important. Contextual importance is formally defined as:

$$CI_j(\vec{C},\{i\}) = \frac{Cmax_j(\vec{C},\{i\}) - Cmin_j(\vec{C},\{i\})}{absmax_j - absmin_j} \tag{1}$$

where $\vec{C}$ is the vector of input values that defines the context, $Cmax_j(\vec{C},\{i\})$ is the maximal value of output j observed when modifying the values of inputs $\{i\}$ and keeping the values of the other inputs at those specified by $\vec{C}$. Correspondingly, $Cmin_j(\vec{C},\{i\})$ is the minimal value of output j observed. $absmin_j$ and $absmax_j$ are typically the minimal and maximal values of output j in the training set, which signifies 0 and 1 for classification tasks. The estimation of $Cmax_j(\vec{C},\{i\})$ and $Cmin_j(\vec{C},\{i\})$ can be performed with Monte-Carlo simulation or more efficient sampling methods regardless of the black-box model or the learned function. For image case, we create samples through perturbing the image. This results in getting only one probability value for each perturbed sample, then the range for minimal and maximal values corresponds to probability value of when a region is on the scene and when it is absent. Therefore *CI* is defined as;

$$CI_j(C,\{i\}) = \frac{out_j(C) - out_j(C,\{i\})}{absmax_j - absmin_j} \tag{2}$$

where $out_j(C)$ is the value of the output j for the context C. $out_j(C,\{i\})$ is the prediction score for the perturbed sample. In this way, $CI_j(C,\{i\})$ expresses where the value of $out_j(C) - out_j(C,\{i\})$ is located in the $[absmax_j, absmin_j]$ range.

3.2 Contextual Importance Explanations for DNNs

DNNs learn to detect the most defining features such as color and edges in their first convolutional layer since this layer receives weighted connections from the input layer. The activation of each node is a weighted sum of pixel intensity values that are passed over to an activation function. So, the set of incoming weights to a node is measuring what that node is about. This can be easily observed in the first layer (see Fig. 1). Then the prediction is made following the last convolutional feature-map where the parts with the highest gradient are the most important for the prediction. Generally, the saliency maps created based on the final layer or another intermediate layer feature-map loses details significantly. We propose identifying interpretable regions through an over-segmentation method which is the process of segmenting the object(s) from the background and fracturing an image into subcomponents [1]. As a result, this increases the chances that boundaries of importance are extracted.

Then, we measure the contextual importance by masking each subcomponent at a time and present the results as visual evidence for a prediction. We note that “interpretable regions” are referred to as “features” throughout the paper.

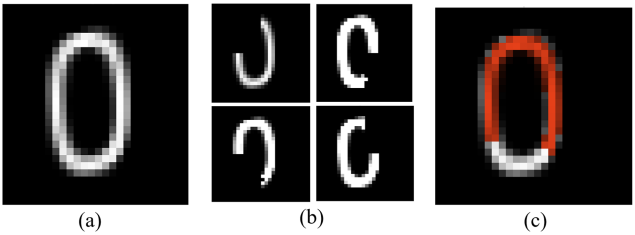

Fig. 2. Contextual importance explanations a)Input image b)Perturbed samples obtained through over-segmentation c)Visualization of the contextual importance

This process is consisting of four steps as outlined in Algorithm 1. Given an input C, we get the prediction class $out_j(C)$ (line 1). We then find samples $(C, \{i\}) \in C$ by filtering each region one after another from C where $(C, \{i\})$ is representative of the input, which is shown in Fig. 2(b). Model runs for each perturbed sample for prediction on output index j (line 2). We find the most important features by simply observing how the prediction score drops for the each interpretable samples $(C, \{i\})$ when they are not on the scene. Then, we concatenate prediction values from the interpretable samples for $out_j(C) - out_j(C, \{i\})$. We compute $CI_j(C, \{i\})$ and visualize the results (line 3 and line 4) as seen in Fig. 2(c). The CI values higher than the threshold (0.01) are represented in color. In this way, we identify which features were activated the most from the perturbed inputs. Our `MATLAB` implementation of this algorithm is available in the `GitHub` repository.[1]

Algorithm 1. Explanations for DNN with CI

```
Given: Context $C$ that specifies input image, output index $j$, $absmin_j$, $absmax_j$, model $f$.
Require: Sample set $(C, \{i\})$ contains perturbed samples.

1: Run $f$ with $C$ to get $out_j(C)$
2: Run $f$ on set $(C, \{i\})$ and get $out_j(C, \{i\})$
3: Calculate $CI_j(C, \{i\})$ using (2)
4: Return $CI_j(C, \{i\})$
```

[1] https://github.com/shulemsi/MNIST_CNN_CI_Explanations.

4 Experimental Results

We apply the proposed method to the MNIST hand-written digit dataset, which has 60,000 training and 10,000 test samples [13]. We provide region-wise explanations derived from the image context as supporting evidence for the class predictions.

The results of contextual importance are illustrated in Fig. 3. The proposed explanation method shows the contributing features to the predicted class based on the degree of contextual importance (i.e., those regions with higher CI value than the threshold). The second row in Fig. 3 shows the over-segmentation clusters, which are used as interpretable regions to perturb the image. The pixel location of high-significance features is rendered in color in the last row. We also report the prediction scores for the input image and the highlighted region, which shows the difference when the low-importance regions are left out. We observe that omitting the features with low importance slightly increased the confidence for the predicted class, which also justifies why the model predicts a certain class.

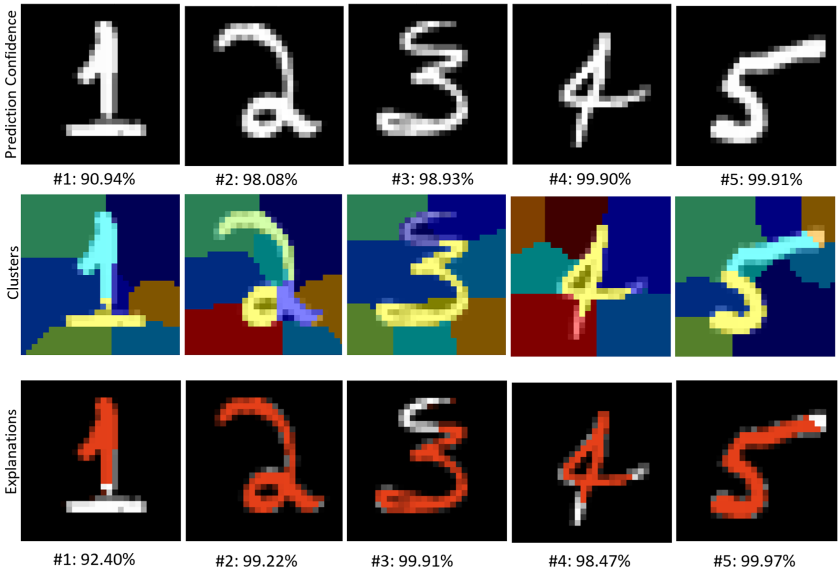

Fig. 3. Contextual importance explanations. The first line is the input images with the probability score for the predicted class. The second line is the over-segmentation clusters for generating perturbed samples. The last line shows features with the highest contextual importance and the prediction score for the explanations.

4.1 Visual Comparisons

To evaluate the proposed explanation method, we provide a visual comparison with two methods namely mask perturbation [23] and Local Interpretable Model-agnostic Explanations (LIME) [18]. Both methods produce explanations based on how the prediction score varies when the features are altered. With this exercise, we look for the immediate intelligibility and explicitness in visual explanations. Figure 4 shows the comparison results. The masking method analyses how sensitive the model's prediction on a class by occluding different parts of an image with a gray square, then provides a pixel-level explanation where the results are presented as heatmaps. Parts with high impacts on the output are highlighted with bright colors; conversely, low impacts are shown in dark colors.

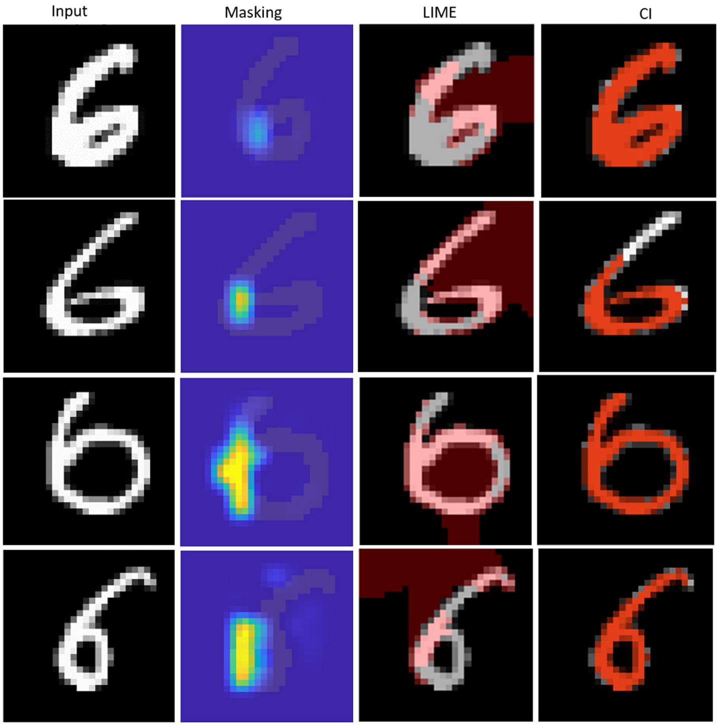

Fig. 4. Visual comparison of explanation methods. The masking method provides a saliency map based on the prediction scores. LIME highlights the relevant super-pixels for the prediction. Contextual importance provides a visual representation of the importance value as the most contributing features to the outcome.

LIME provides a region-based explanation by discovering the relevant segments. Segments with the highest positive weights are presented as an explanation (in red). Yet, the parts of the background are shown with positive relevance along with non-zero values. The masking method is implemented on CNN results and LIME uses the random forest algorithm for the hand-written digit classification task. While both methods give important intuitions, neither is precise in demonstrating the features leading to a prediction. Masking method gives significant insight into identifying the minimal region which has the most impact on the output, however their results are not immediately intelligible. Morever, the masking method could be computationally expensive when handling an instance with high dimensions since they need to sequentially perturb the image. LIME results in losing significant parts of the object. Given that humans intuitively pinpoint the most typical features of an class on an image, the contextual importance could be considered better for humans to understand and evaluate the predictions.

4.2 Explanations on Contrastive Cases

Humans generally explain the cause of an event relative to some other event that did not occur. Thus explanations are usually in the form of "Why this?- and not that", with a contrasting case that did not occur, even it is implicit in the question [16]. It is common for humans to state and request contrastive facts to distinguish between similar examples instead of giving complete explanations (i.e., listing all the causes of an event). Contrastive explanations could be more intuitive and valuable particularly for multivariate datasets since the cognitive load of complete explanations could be high in those cases.

As the contextual importance can be computed for every function that outputs a prediction value for all classes, this makes it possible to explain why a certain class $out_j(C)$ is more likely than another (i.e., here $out_c(C)$ indicates the contrastive case). For this, feature importance over the perturbation variable $(C, \{i\})$ is computed to explain the model's prediction results for the contrasting case $out_c(C)$ and compare it with the initial case. Thus, we are not only interested in visualizing the present features but also looking for missing features. Given an input C with prediction class $out_c(C)$, we find predictions for interpretable perturbations for the index of the contrastive case. We find the importance of the features that are contributing to prediction class $out_c(C)$ using the same equation (Eq. 2). The visualization demonstrates the most critical features contributing to different classes (see Fig. 5).

Thereby contrastive explanations are generated in the form of either;

- The C is predicted as class $out_j(C)$, because features $(C, \{i\})$ have high importance, which are not typical of class $out_c(C)$.
- The C is not predicted as class $out_j(C)$, because features $(C, \{i\})$ has no importance, which are typical of class $out_c(C)$.

Here, an example is presented for number 7 and a set of features that distinguish it from number 2, which has the second high probability. Figure 5 illustrates the visualizations of contextual importance values as well as the absent features in number 2, which distinguishes it from 7. Our comparisons with LIME show

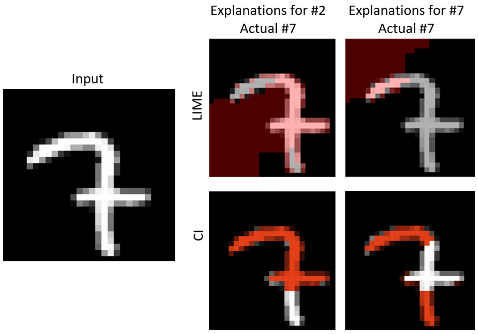

Fig. 5. CI and LIME comparisons for contrastive cases.

varying explanation results for both the actual class and the contrastive case. The important features are highlighted in order to identify and distinguish the two classes. LIME shows background as the positive relevance besides the nonzero pixels. The reason for this could be the result of the clustering algorithm. Moreover, it is very often that LIME produces the same explanations for different class predictions. Figure 6 demonstrates this for two different forms of the same digit. LIME resulted in identical explanations for number 5 and actual class number 6, which is not very explanatory to know why this image is labeled as either of the two classes.

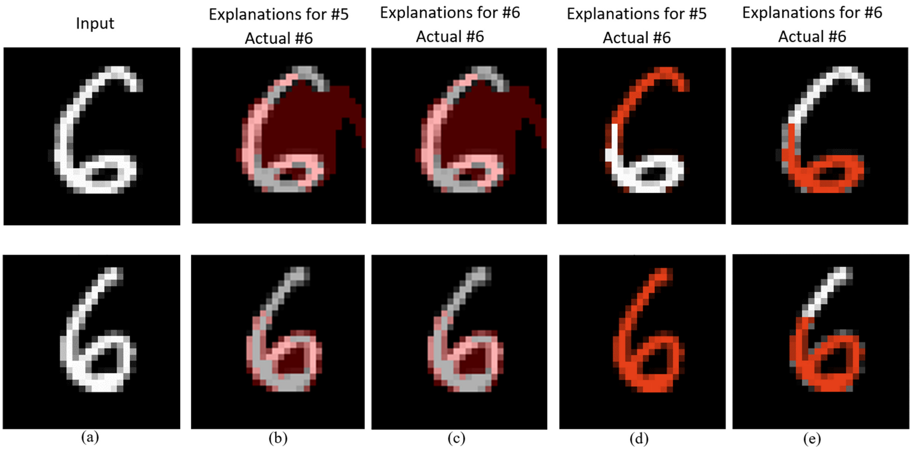

Fig. 6. Contrastive explanations (a) Input image number 6 (b) LIME explanations for 5 (c) LIME explanations for 6 (d) Contextual importance explanations for 5 (e) Contextual importance explanations for 6.

4.3 Explanations on Distorted Images

Even minor alterations in an image can produce completely different classification results. Handwritten text is likely to have varying noises such as underlines, neighboring characters, or stray marks. Here, we tested the model with different images of the numbers that were not in the training set to observe whether the model can extract enough common features from distorted pictures to identify a number. Finally, the explanations are generated to check whether they are having invariance to these visual differences.

We also tested the contextual importance for the consistency of the explanations by adding noise to the input image and evaluating how it affects the explanations. The degree of noise resistance and distortion invariance is experimented through contextual importance. The distorted images are tested to verify if the model predicts the correct label. The importance over the perturbation variable $(C, \{i\})$ is computed for the output index j, then the contextual importance values are visualized for each class. We are interested in visualizing whether such noises would be shown as relevant features and if the explanations are robust to such variations.

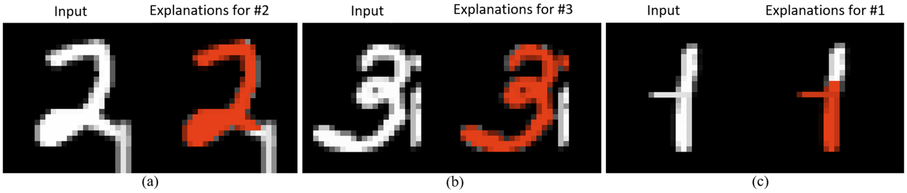

Fig. 7. Contextual importance explanations for partially distorted and noisy images (a) Explanations for 2 (98.80%) (b) Explanations for 3 (99.95%) (c) Explanations for 1 (98.21%).

The explanations for the distorted numbers are shown in Fig. 7. Despite the noise and distortion, these samples were correctly recognized by the model with high confidence. For the given examples, DNN is able to extract the salient features from the cluttered image and contextual importance provided robust explanations under the noisy conditions. Although the completely invariant classification of complex shapes in the physical world is still a challenging task, contextual importance could provide a solution to the problem of robustness concerning partial distortions. We also note that such explanations depend on the outcome of over-segmentation clusters.

4.4 Explanations on Misclassification

In this section, we show explanations for a misclassification case to identify the features causing a wrong prediction. The contextual importance for the perturbation variable $(C, \{i\})$ is computed for the predicted class and for the true class.

Then, both classes are visualized to demonstrate the relevant features for the incorrect class $out_j(C)$ and analyse why the model predicts as $out_j(C)$ instead $out_c(C)$. We present LIME and contextual importance explanations for example of true class number 9 and predicted class number 4 in Fig. 8. This input example classified as number 4 by 51% and the actual class is 9 by 19% probability. LIME gives explanations for class number 4 (12%) while it failed to find any relevant features for the actual class (6%). The contextual importance features represented for number 4 (by 99%) are resembles to explanations produced by LIME as shown in Fig. 8. Contextual importance for ground truth (9) (by 26%) highlights all the pixels as important. Although the prediction score is low for the ground truth, it still provides discriminative features for each class.

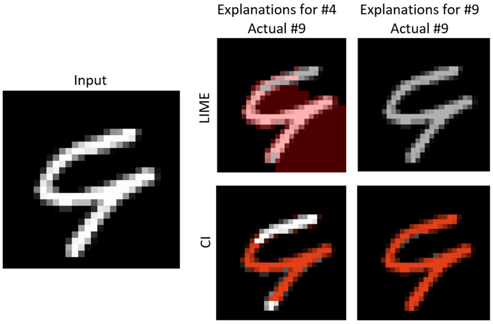

Fig. 8. CI and LIME comparisons on misclassification

This kind of result gives no direct explanation about why a model makes a wrong prediction, however, they could potentially enhance trust as it helps to identify the contributing features and evaluate whether the model is performing in an arbitrary way. Hence, understanding the features learned by a model will give an opportunity to improve the dataset and correct the model.

5 Conclusion

In this paper, we proposed contextual importance to provide visual explanations for DNN predictions. The method presents region-wise explanations for image classification results by visualizing the contribution of each region based on the degree of importance. The visual comparison results show that the contextual importance provides explicit visual justification for the DNN predictions. The results also demonstrate that the region with the high importance gives a class score close to the initial prediction score. This suggests that the proposed method is able to extract the most relevant features for the prediction to justify an outcome. The idea is further supported by providing explanations for wrong predictions to investigate the regions contributing to misclassification. The results indicate that contrastive explanations offer a way to analyze the misclassified examples and identify class discriminative features. Being limited to only a hand-written digit dataset, this study lacks the implementation of contextual importance for the multi-object classification explanations, which is an important consideration for our future work. Another interesting research direction could be exploring contextual importance for text-based explanations in a multi-object classification task.

Acknowledgements. This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012)
2. Alvarez-Melis, D., Jaakkola, T.S.: Towards robust interpretability with self-explaining neural networks. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 7786–7795. Curran Associates Inc. (2018)
3. Anjomshoae, S., Främling, K., Najjar, A.: Explanations of black-box model predictions by contextual importance and utility. In: Calvaresi, D., Najjar, A., Schumacher, M., Främling, K. (eds.) EXTRAAMAS 2019. LNCS (LNAI), vol. 11763, pp. 95–109. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30391-4_6
4. Anjomshoae, S., Kampik, T., Främling, K.: Py-CIU: a python library for explaining machine learning predictions using contextual importance and utility. In: IJCAI-PRICAI 2020 Workshop on Explainable Artificial Intelligence (XAI) (2020)
5. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS one **10**(7), e0130140 (2015)
6. Bodria, F., Giannotti, F., Guidotti, R., Naretto, F., Pedreschi, D., Rinzivillo, S.: Benchmarking and survey of explanation methods for black box models. arXiv preprint arXiv:2102.13076 (2021)
7. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-CAM++: generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 839–847. IEEE (2018)
8. Du, M., Liu, N., Hu, X.: Techniques for interpretable machine learning. Commun. ACM **63**(1), 68–77 (2019)
9. Du, M., Liu, N., Hu, X.: Techniques for interpretable machine learning. Commun. ACM **63**(1), 68–77 (2019). https://doi.org/10.1145/3359786
10. Eykholt, K., et al.: Robust physical-world attacks on deep learning visual classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1625–1634 (2018)
11. Främling, K.: Modélisation et apprentissage des préférences par réseaux de neurones pour l'aide à la décision multicritère. Ph.D. thesis, Institut National de Sciences Appliquées de Lyon, Ecole Nationale Supérieure des Mines de Saint-Etienne, France (1996)

12. Hohman, F., Park, H., Robinson, C., Chau, D.H.: Summit: scaling deep learning interpretability by visualizing activation and attribution summarizations. arXiv preprint arXiv:1904.02323 (2019)
13. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
14. Li, O., Liu, H., Chen, C., Rudin, C.: Deep learning for case-based reasoning through prototypes: a neural network that explains its predictions. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
15. Liu, M., Shi, J., Li, Z., Li, C., Zhu, J., Liu, S.: Towards better analysis of deep convolutional neural networks. IEEE Trans. Visual Comput. Graphics **23**(1), 91–100 (2016)
16. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. Artif. Intell. **267**, 1–38 (2019)
17. Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., Clune, J.: Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In: Advances in Neural Information Processing Systems, pp. 3387–3395 (2016)
18. Ribeiro, M.T., Singh, S., Guestrin, C.: 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. ACM (2016)
19. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626 (2017)
20. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 3145–3153 (2017). JMLR.org
21. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
22. Yoo, C.H., Kim, N., Kang, J.W.: Relevance regularization of convolutional neural network for interpretable classification. Network **50**, 5 (2019)
23. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53
24. Zhang, Q., Nian Wu, Y., Zhu, S.C.: Interpretable convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8827–8836 (2018)
25. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929 (2016)