On the Explainability of Convolutional Layers for Multi-Class Problems

Joe Townsend,[1] **Mateusz Kudla,**[2] **Agnieszka Raszkowska,**[2] **Theodoros Kasioumis,**[1]

[1]Fujitsu Research of Europe LTD, 4th Floor, Building 3, Hyde Park Hayes, 11 Millington Road, Hayes, UB3 4AZ, England
[2]Fujitsu Technology Solutions Sp. z o.o. .KTW, ul. Roździeńskiego 1 40-202 Katowice, Poland
firstname.lastname@fujitsu.com

Abstract

Neuro-symbolic reasoning systems support the goal of making the behaviour of trained neural networks more explainable. ERIC and SRAE are two such methods for CNNs that are similar in that they both provide decompositional, layer-wise explanations that can be extracted post-hoc and deployed as classifiers in their own right. However the two methods differ in how they represent knowledge and reason over those representations; ERIC reduces the layer's behaviour to a discrete logic program for symbolic reasoning over a vocabulary at most as large as the number of kernels in that layer; and SRAE reduces the layer's output to more limited but concise vocabulary represented by a set of continuous, orthogonal and sparse features. We compare both methods and show that despite these differences they yield similar results with respect to fidelity when deployed as approximations of the original CNN. SRAE offers marginally stronger fidelity than ERIC but in sacrificing some fidelity ERIC is able to offer a larger and more discrete set of symbols that more closely match what individual kernels actually see. Neither method has previously been demonstrated on multi-class problems but we show for the first time that under certain conditions they may yield high fidelity in such cases. However for both methods fidelity drops for those multi-class datasets in which images have less distinct edges. Similar results under different representations suggest challenges for layer-wise knowledge extraction in general and invite further investigation from the neuro-symbolic community, with our results offering an early benchmark for such research.

One goal of explainable AI is to understand the reasoning behind decisions made by trained convolutional neural networks or other models when making a classification, and to present this reasoning in an interpretable way. A popular approach is to visualise regions of an input that are important for assigning it to a particular class or for activating a given convolutional kernel (Simonyan, Vedaldi, and Zisserman 2013; Zeiler and Fergus 2014; Binder et al. 2016; Shrikumar, Greenside, and Kundaje 2017; Zintgraf et al. 2017). However relatively few attempts have been made to explore the relationships between those features in the same way that earlier neuro-symbolic reasoning systems constructed rules or other relational structures between symbols that approximate the meaning of neurons (Andrews, Diederich, and Tickle 1995; d'Avila Garcez, Broda, and Gabbay 2002; Hammer and Hitzler 2007; Besold et al. 2017; Townsend, Chaton, and Monteiro 2019).

Some of these models are evaluated by asking humans to perform tasks based on their interpretation of the rules. This is a good way to test interpretability but good explainable models should also be accurate, and this is best evaluated by what we refer to as *independent* reasoners that can be used as classifiers themselves without human intervention. Some such models for CNNs have been tested on 'one-versus-all' classification in which a multi-class task (e.g. MNIST (LeCun et al. 1998)) is treated as a two-class problem (e.g. '1 or not 1') (Qi, Khorram, and Fuxin 2021; Zhang et al. 2018a; Zhang, Nian Wu, and Zhu 2018). ERIC (Townsend, Kasioumis, and Inakoshi 2020) was shown to support multiple classes but only in the minimal sense that it was tested on three. We build on previous work by applying ERIC to more classes and comparing to another method *SRAE*, which like ERIC yields a set of sentence-like explanations but represented in an entirely different way and had only previously been tested on the one-vs-all case. We show that under certain circumstances both, despite employing different representations for their explanations, can be used to reason over multi-class tasks without reducing them to the one-vs-all case. The two methods are comparable in that they both explain the behaviour of any convolutional layer regardless of whether the CNN is designed or trained to be explainable, in other words they are *decompositional* with respect to layers and *post-hoc* as opposed to *explainable-by-design (XBD)*.

We begin by covering some preliminaries and reviewing related work. In the method section we describe our implementations of ERIC, SRAE, and our experimental method. We then present empirical results and some examples of extracted rules. We end with a discussion on the implications of our findings and on opportunities for future work.

Preliminaries

We consider a set of input images $\boldsymbol{x}$ and target outputs $\boldsymbol{t}$ both indexed by i, a CNN M and its explainable approximation M^*. Let $f(M, \boldsymbol{x})$ and $f(M^*, \boldsymbol{x})$ denote the classification output of each. *Accuracy* is the percentage of samples for which $f(\cdot, \boldsymbol{x}_i) = \boldsymbol{t}_i$ and *fidelity* the percentage of samples for which $f(M^*, \boldsymbol{x}_i) = f(M, \boldsymbol{x}_i)$. Let $l \in \{l^e, l^o\}$ refer to a layer on M; l^o to the softmax output and l^e to the layer

based on whose behaviour we wish to explain the output at l^o (in general explanations may use multiple layers, but in our work we only use 1). $A_{i,l,k} \in R^{w \times w}$ denotes an activation matrix output for a unit, where w is a natural number and a unit is a kernel for the case of l^e or a softmax neuron for l^o, with $w = 1$. l^e and l^o have units indexed by $k = 1, \ldots, K_l$.

Though M^* may take make forms, *logic programs* defined as follows are of particular relevance to the paper. Each symbol that can be expressed in a rule is a logical *atom*, which may be expressed as a positive ($\mathcal{L}_i$) or negative ($\neg\mathcal{L}_i$) literal to express whether that atom is *true* or *false*. Rules over a set of literals are denoted $R = \{R_r = (D_r, C_r)\}_r$, where D_r is a set of conjoined literals acting as the conditions for rule R_r, and C_r is a literal that evaluates as true iff those conditions are met. For example, if $D_r = P_1 \wedge P_2 \wedge \neg P_3$ and $C_r = Q$, the rule $P_1 \wedge P_2 \wedge \neg P_3 \rightarrow Q$ states that if P_1 and P_2 are true and P_3 is false then Q is true.

Background

Multiple classifications and taxonomies have been proposed for neuro-symbolic learning and reasoning systems (Andrews, Diederich, and Tickle 1995; Bader and Hitzler 2005; Townsend, Chaton, and Monteiro 2019). Below we elaborate on some of these categorisations before discussing related work in these terms. We are interested only in what we refer to as *independent* reasoners that are capable of replicating the behaviour of M without human intervention.

Properties of Neuro-Symbolic Reasoning Systems

Pedagogical methods consider M to be a black box and observe the relationships between the inputs and the outputs without concern for the architecture between. In general these are not limited to neural networks but may be applied to any class of model. On the contrary, *decompositional* methods explain M in terms of its parts (e.g. neurons, kernels or layers). These are less transferable as some assumptions about the model's architecture must be made, but the advantage is that they allow one to understand the model's internal knowledge representation.

If M^* is *global*, it explains the overall behaviour of M across arbitrary samples, whereas *local* explanations explain classifications of specific samples or subsets thereof (Ribeiro, Singh, and Guestrin 2016).

We consider a method to be *explainable-by-design* if it includes or assumes a means of designing or training M to be at least partly interpretable. XBD methods will be crucial for scenarios in which M must be held accountable, especially in safety-critical situations. However accountability is not the only motivation; explainability is also useful for aquiring new knowledge. Models constrained to reason like humans may only ever think like humans themselves, thus XBD restricts what the models are capable of learning. To learn from models trained independently of such assumptions (in particular those trained before XBD methods were invented), we require *post-hoc* methods which can be applied after-the-fact under less restrictive assumptions. Besides, post-hoc methods could still be applied to XBD networks by virtue of the fact that they are more generalisable. These reasons motivate our exploration of them in this paper.

Neuro-Symbolic Reasoning for CNNs

ERIC yields global explanations of CNN behaviour for one or more convolutional layers. It maps kernel outputs to single activations via L1 or L2 values of the kernel outputs and thresholds these values to yield binary approximations of those kernel outputs (Townsend, Kasioumis, and Inakoshi 2020). M^* is a logic program constructed to approximate the behaviour of the network after and including l^e and in terms of the binarised kernel activations extracted from training data. ERIC was previously demonstrated to work on the multi-class case but only in that it is shown to distinguish between three classes. A correlation between the accuracy of M and fidelity of M^* was also observed. Interpretability is measured based on the size of the extracted logic program, on the basis that smaller rule sets are easier for humans to read, but no human evaluation is performed.

Like ERIC, (Odense and Garcez 2020) explores global, layerwise-explainability and translates kernel outputs to literals in a logic program of so-called *M-of-N* rules in which a rule evaluates as true iff M literals out of all N in the body are true. Kernels are represented by the output of that neuron which yields the maximum information gain with respect to the output of the network, and the rule extraction process is a heuristic search that priorises literals according to the weight between their corresponding neuron and the target neuron represented by the head. Tests for CNNs are conducted on multiple datasets including MNIST.

For SRAE (Qi, Khorram, and Fuxin 2021), M^* is a smaller neural network called an XNN (explainable neural network). SRAE maps l^e to a low-dimension set of n so-called *x-features* in M^*, which is trained to learn those features such that they are sparse, orthogonal and yield faithful approximations of the output at l^o. The explanations are a series of visual projections of the x-features back onto the original image. In theory l^e may be any layer, though published results mostly focus on recreating the first dense layer, with some treatment of convolutional layers with respect to the CUB dataset (Welinder et al. 2010) and MNIST. Furthermore, SRAE is only demonstrated on 'one-vs-all' classification. The interpretability of SRAE is tested via human evaluation in which participants are asked to cluster samples based on visualisations presented.

Prototype methods introduce a layer of filters which learn prototypical examples of various parts against which input image regions can be compared (Chen et al. 2019; Hase et al. 2019; Kim et al. 2021; Rymarczyk et al. 2021). The approach is designed so that networks are trained end-to-end and so the prototype layers cannot be 'plugged in' to the original model and used for post-hoc reasoning.

Zhang et al. proposed a post-hoc approach in which part representations are disentangled from the trained CNN and rearranged into a hierarchical 'AND-OR' graph (AOG) (Zhang et al. 2018a). Interpretability is demonstrated qualitively and quantitively, but the explanations are not used as an independent classifier and so fidelity cannot be measured. The same authors also developed a means of extracting a decision tree (Zhang, Nian Wu, and Zhu 2018), but this method assumes that kernels have been trained to yield interpretable filter activations according to a loss function proposed in

previous work (Zhang et al. 2018b).

LIME (Ribeiro, Singh, and Guestrin 2016) is a post-hoc, pedagogical explainer based on features and was later extended to rule-based explanations in the form of *Anchors* (Ribeiro, Singh, and Guestrin 2018). However both are local explainers only. Frosst and Hinton introduce a global, post-hoc approach designed for CNNs though is pedagogical in nature and so perhaps could be applied more generally (Frosst and Hinton 2017). In this case M^* is a decision tree capable of multi-class classification and is mainly demonstrated on MNIST.

Of the methods discussed, ERIC, SRAE and the M-of-N method are all post-hoc, layerwise-decompositional, global; and may all be applied as independent classifiers so that fidelity can be objectively compared. One difference is that the M-of-N method is dependent on the weights of connections in the original layers, which are only defined between adjacent layers. Both ERIC and SRAE are able to bypass the layers between any l^e and l^o and so we chose to compare these two as representatives of different models of explanation: the former being discrete and rule-based and the latter being continuous and feature-based. Furthermore both have yet to be applied to scenarios with large number of classes.

Though not explicitly called *x-features* in the literature, the conditions in rules are equivalent to x-features as defined for SRAE and so we refer to them as such henceforth. A notable difference between the rule based methods and SRAE is that for the latter, a single x-feature is derived from a function over multiple kernels in l^e whereas in the rule-based methods each x-feature is derived from a single kernel in l^e. In other words, visually the set of K_{l^e} *possible* x-features in rule-based methods are more faithful representations of l^e than the n x-features used by SRAE. This also means that SRAE's explanation space is limited to the use of these n x-features only. For rule-based methods a *single* explanation may still be limited to a length of n but each x-feature may selected from a *vocabulary* of $K_{l^e} \geq n$ possible atoms.

Method

We evaluate the fidelity of ERIC and SRAE on a wider range of datasets in multi-class contexts. For each dataset we train one CNN, on which we perform all further experiments for that dataset. Then for both methods we perform 5 trials of knowledge extraction on the same CNN at the final (13th) convolutional layer. All tests were performed on an Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz with 200 GB RAM, and Ubuntu 18.04. Tests were implemented using Keras 2.4 from Tensorflow 2.3.1, in Python 3.6.9 and spread across 8 x NVIDIA GeForce RTX 2080 Ti, with Nvidia Driver 460.32.03, Cuda 10.1 and CUDNN 7. Datasets and configurations for training and extraction are described below.

Datasets

Where datasets do not provide testing annotations publically or validation data is not pre-defined, we rearrange the partitions as described below.

First, we test on *MNIST* (LeCun et al. 1998) as a standard benchmark and one which is built of multiple classes. We split the standard training set into training and validation, with the latter composed of the last 10,000 images. We use the normal test set.

We also test on the *German Traffic Sign Recognition Benchmark (GTSRB)* (Stallkamp et al. 2012) as it is an example of one with many classes (43). Each class contains multiple instances of a physical signpost and multiple images are taken for each of these signposts. Therefore we split with respect the signposts and not individual images. Thus, we apply a 8:2 training:validation split. We use the original test set as provided.

We select *PascalVOC 2010* (Everingham et al.) as our third dataset as it has been used by other datasets in the related literature, though mainly for just the animal classes which make up 6 out of the total 20 (Zhang, Nian Wu, and Zhu 2018; Zhang et al. 2018a,b). We test on both the 6 and 20 class cases. In both cases we split the validation set in half to form the new validation and test sets.

Lastly, we select a subset of room classes from *Places* (Zhou et al. 2017) partly for its use in the SRAE experiments but also because ERIC was previously tested on another subset of *Places*. We also use this to explore the effect of varying the total number of classes used, in particular $\in \{2, 3, 5, 10\}$, starting with bathroom and bedroom, then adding kitchen, then dining and living rooms, and finally home office, office, waiting room, conference room and hotel room. For each set of classes we derive the test set from the training data as the provided validation set is quite small, at 100 images per class. This gives us 4,000 and 1,000 training and test images respectively per class.

CNN training

For all datasets we mostly use the same configuration for training the CNN, with some variations that we choose because they yielded better results in preliminary trials; the important point is that for each dataset, all explanation methods are applied to the same CNNs. As default, we use VGG16 pretrained on imagenet and replace the softmax layer with a number of softmax neurons equal to the appropriate number of classes. We train in batches of 32 for 100 epochs using the Adam optimiser, with the exception of the Pascal cases, for which we found RMSProp to be the better optimiser. We applied class weights where there was imbalance in the training data. We use L2 regularisation of 0.005 in all layers and a learning rate of $5 * 10^{-7}$ in all cases except MNIST, where we used 10^-6. We applied a decay factor of 0.5 and patience of 10 epochs. For all datasets images are centre-cropped where not necessarily square in dimension and resized to 224×224. For Places we augmented the dataset by randomly applying vertical and horizontal flips and a channel shift of range 20 at each epoch, according to uniform distributions.

Rule Extraction

For both ERIC and SRAE we set $l^e = 13$, so that M^* takes input from the final convolutional layer of M. As SRAE was proposed to be optimal for $n = 5$ x-features, we apply the same for both SRAE and ERIC. For both models we conduct 5 trials and average the results for two reasons. Firstly,

SRAE is a stochastic solution. ERIC's rule extraction algorithm is deterministic, however it requires a large number of kernel activations to be stored in memory at a single time and so for larger datasets we perform extraction based on a uniform random selection of samples from the training set (except for *Pascal*, which is small by comparison). For inference we can more easily batch process all samples and so random sampling is unnecessary. For fairness we also train SRAE on random subsets.

ERIC We implement ERIC mostly according to the literature (Townsend, Kasioumis, and Inakoshi 2020) but simplified to extract from a single layer and with a slightly different approach to binarising the kernels (Equation (4)). Pseudocode of the implementation is provided in the technical appendix in the supplementary materials. Let $b_{i,l,k} \in \{1, -1\}$ denote a binary truth value assigned to $A_{i,l,k}$ as in Eq. (1). $b_{i,l,k}$ may be expressed as positive and negative literals $\mathcal{L}_{i,l,k} \equiv (b_{i,l,k} = 1)$ and $\neg\mathcal{L}_{i,l,k} \equiv (b_{i,l,k} = -1)$ respectively. For a rule R_r, conditions in D_r correspond to a subset of kernels in l^e and C_r corresponds to the class to be assigned if those conditions are met. These rules are extracted using a tree-based extraction algorithm similar to C4.5 (Quinlan 1993). Let us denote the training data $Z_l = \{(\boldsymbol{z}_i, t_i) \mid i = 1, ..., m\}$ where $\boldsymbol{z}_i \in \{True, False\}^{2K_{l^e}}$ and $t_i \in \{True, False\}$. $z_{l^e,k} = True$ if it corresponds to a positive literal and its binary value is 1 or if it represents a negative literal and its binary value is -1. It is *false* otherwise. Each path from the root to a leaf of the tree represents a rule, with nodes branching on rule conditions based on the pearson criterion. A pruning parameter ρ prevents overfitting such that we stop branching if $|Q|/|P| < \rho$, where P, Q represent sets of training instances that satisfy the path of conditions leading to a parent node and child node, respectively. If a leaf represents multiple outcomes, the class is chosen to be the modal value of Q.

$$b_{i,l^e,k} = Q(A_{i,l^e,k}, \theta_{l^e,k}) \tag{1}$$

$$Q(A_{i,l,k}, \theta_{l,k}) = \begin{cases} 1, & \text{if } a^{tr}_{i,l,k} > \theta_{l,k} \\ -1, & \text{otherwise} \end{cases} \tag{2}$$

$$a_{i,l,k} = \|A_{i,l}\|_2 \tag{3}$$

$$\theta_{l,k} = \alpha \cdot \overline{a^{tr}_{l,k}} + \gamma \sqrt{\frac{1}{n} \sum (a^{tr}_{i,l,k} - \overline{a^{tr}_{l,k}})^2} \tag{4}$$

We set the kernel threshold according to Equations (4) and (3). Values for α, γ and ρ are chosen for each dataset according to a grid search over validation fidelity with $\alpha, \gamma \in \{0.1 * t | t \in [0..1]\}$ and $\rho \in \{0, 0.001, 0.003, 0.006, 0.01\}$. For *Places* and *Pascal*, the grid search is performed on the largest instances of each (i.e. 10 and 20 classes respectively). Final parameters are given in table 1.

SRAE We implement the XNN as shown in Fig. 1, with a single encoder and decoder layer pair and the output of the encoder max-pooled to yield the x-features, which then feed into a softmax classification layer. We implement the loss function for SRAE according to the literature (Qi, Khorram, and Fuxin 2021). The loss function has parameters β and η for weighting sparsity and orthogonality respectively, plus another q used for calculating the sparsity term. We use a grid search to choose $\beta, \eta \in [1..5]$, $q \in [1..10]$ with respect to training fidelity on the largest instances of each dataset. The chosen values are given in table 1. For each dataset the XNN is trained for 100 epochs using the Adam optimiser with a learning rate of 0.001.

		ERIC			SRAE		
Dataset	% of train. set	α	γ	ρ	q	β	η
GTSRB	30	0.6	0.7	0.0	1	5	1
MNIST	20	1.0	0.6	0.0	1	1	1
Pascal	100	1.0	1.0	0.001	3	3	2
Places	20	0.6	0.7	0.006	5	2	3

Table 1: Dataset-specific rule extraction configurations.

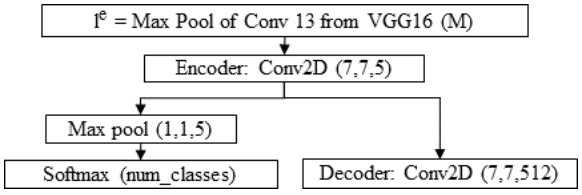


Figure 1: Architecture for our implementation of the SRAE XNN, with brackets giving layer dimensions (channels last).

Results

Empirical results

Fig. 2 and table 2 show results for all datasets with respect to both M and M^*. SRAE slightly outperforms ERIC in terms of accuracy and fidelity though ERIC is slightly superior on GTSRB and *Pascal* (All). We regard 70% or over to be a reasonable score for test accuracy with respect to explainable models. Both methods yield this or higher for MNIST, GTSRB, and Places with 2 or 3 classes. SRAE also crosses this threshold for Pascal animals. Fig. 3 shows that in both cases the fidelity of M^* shares an almost linear relationship with the test accuracy of M, a finding which is consistent with previous findings using ERIC (Townsend, Kasioumis, and Inakoshi 2020) but the first such observation for SRAE.

In general it would seem that it is difficult to extract for a large number of classes in the Pascal and Places datasets, with accuracy and fidelity dropping as the number of classes increases. However fidelity cannot only be dependent on the number of classes both methods are capable of extracting relations for a 10-class case (MNIST) and a 43-class case (GTSRB). The better performance on these datasets compared with Pascal and Places are likely because the images in the former two are more symbolic with more defined edges, whereas the latter two are more photographic.

Fig. 4 shows that the number of classes does appear to affect the size of M^*. For SRAE this is clear because M^* is a neural network and a dense output layer will always increase in size linearly with the number of classes. For ERIC the correlation is likely because with more classes, more rules and

		M	$M^* = ERIC$					M^* = SRAE		
Dataset	Part.	Acc.	Accuracy	Fidelity	Atoms	Rules	Size	Accuracy	Fidelity	Conns
MNIST	Train	1.00	0.93 ± 0.00	0.93 ± 0.00	147 ± 2	86 ± 4	413 ± 18	0.97 ± 0.00	0.97 ± 0.00	5170
	Valid	1.00	0.93 ± 0.00	0.93 ± 0.00				0.97 ± 0.01	0.97 ± 0.01	
	Test	1.00	0.92 ± 0.00	0.93 ± 0.00				0.96 ± 0.01	0.96 ± 0.01	
GTSRB	Train	1.00	0.81 ± 0.01	0.81 ± 0.01	232 ± 3	125 ± 5	626 ± 28	0.80 ± 0.02	0.80 ± 0.02	5335
	Valid	0.99	0.74 ± 0.01	0.74 ± 0.01				0.71 ± 0.03	0.71 ± 0.03	
	Test	0.98	0.73 ± 0.01	0.73 ± 0.01				0.72 ± 0.02	0.72 ± 0.02	
Pascal Animals	Train	1.00	0.71 ± 0.00	0.71 ± 0.00	86 ± 0	52 ± 0	246 ± 0	0.94 ± 0.02	0.94 ± 0.02	5150
	Valid	0.90	0.62 ± 0.00	0.62 ± 0.00				0.71 ± 0.03	0.72 ± 0.03	
	Test	0.90	0.61 ± 0.00	0.62 ± 0.00				0.70 ± 0.02	0.71 ± 0.02	
Pascal All	Train	1.00	0.79 ± 0.00	0.79 ± 0.00	239 ± 0	145 ± 0	690 ± 0	0.82 ± 0.03	0.82 ± 0.03	5220
	Valid	0.89	0.64 ± 0.00	0.64 ± 0.00				0.58 ± 0.03	0.59 ± 0.02	
	Test	0.88	0.63 ± 0.00	0.64 ± 0.00				0.61 ± 0.02	0.63 ± 0.02	
Places (2)	Train	1.00	0.90 ± 0.01	0.90 ± 0.01	11 ± 1	12 ± 2	52 ± 8	0.96 ± 0.01	0.97 ± 0.01	5130
	Valid	0.97	0.91 ± 0.01	0.91 ± 0.02				0.94 ± 0.02	0.94 ± 0.02	
	Test	0.97	0.89 ± 0.01	0.89 ± 0.01				0.95 ± 0.00	0.95 ± 0.00	
Places (3)	Train	1.00	0.84 ± 0.01	0.84 ± 0.01	33 ± 4	25 ± 2	118 ± 13	0.91 ± 0.04	0.91 ± 0.04	5135
	Valid	0.96	0.82 ± 0.01	0.80 ± 0.02				0.85 ± 0.06	0.85 ± 0.05	
	Test	0.94	0.81 ± 0.01	0.82 ± 0.01				0.87 ± 0.04	0.88 ± 0.04	
Places (5)	Train	1.00	0.66 ± 0.01	0.67 ± 0.01	57 ± 3	36 ± 2	171 ± 10	0.74 ± 0.02	0.74 ± 0.02	5145
	Valid	0.87	0.64 ± 0.02	0.65 ± 0.02				0.68 ± 0.03	0.69 ± 0.03	
	Test	0.85	0.63 ± 0.01	0.65 ± 0.01				0.67 ± 0.02	0.70 ± 0.02	
Places (10)	Train	0.97	0.38 ± 0.01	0.39 ± 0.01	85 ± 6	42 ± 3	208 ± 16	0.46 ± 0.04	0.47 ± 0.04	5170
	Valid	0.73	0.37 ± 0.02	0.41 ± 0.02				0.42 ± 0.05	0.44 ± 0.04	
	Test	0.70	0.36 ± 0.01	0.39 ± 0.01				0.41 ± 0.04	0.43 ± 0.04	

Table 2: Results over 5 trials. Note that since for Pascal each trial uses 100% of the training set, the results for ERIC, which is otherwise deterministic, show no variation. For ERIC size is measured as the sum length of rules in the extracted program.

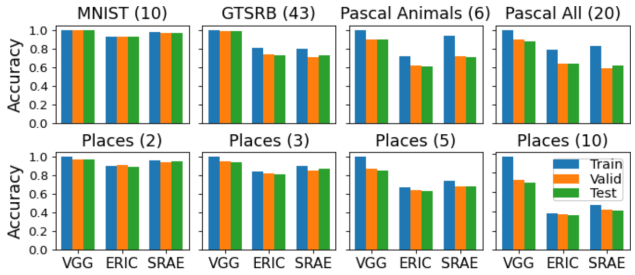
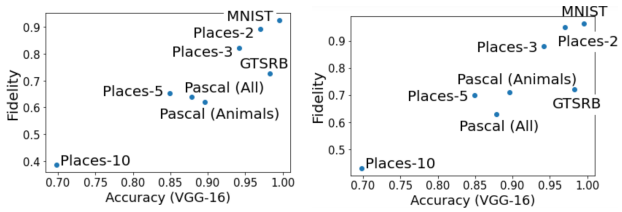


Figure 2: Results for all datasets. The number in brackets show the number of classes in each dataset.



(a) ERIC (b) SRAE

Figure 3: The fidelity of the extracted program correlates with the accuracy of the original CNN. This is shown for test data but similar trends are observed for other partitions.

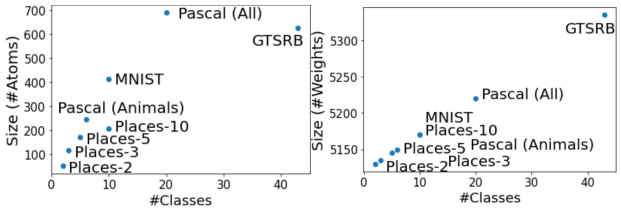
x-features must be employed to distinguish between them.

Rule and kernel samples

We now visualise some of the explanations and rules generated by both methods by taking the raw activation from relevant x-features, generating a mask over values ≤ 0.03 and applying this to the original image (e.g. Fig. 5).

Fig. 6 presents an example of a rule from *Places 5* that correctly identifies a kitchen. Each kernel is represented by 5 training images that activate that kernel's threshold, with the strongest at the top, the weakest at the bottom and the remaining three evenly spaced between these two extremes with respect to the kernel's activation level (Equation (3)). Kernels BQ and LM as identified by ERIC both respond to cabinets. Kernel labels are assigned automatically, using double-letters as up to 512 atoms would be needed for all kernels. The negated condition kernels FT and SA mostly respond to beds (FT) and sinks or toilets (SA). Cabinets may be found in many rooms but in a kitchen it would be unlikely to also find upholstry or ceramic bathroom utilities.

Example explanations under SRAE for the test set are shown in Fig. 7. In ERIC rules may use different x-features but in SRAE all explanations share the same x-features and so explanations and kernel visualisations are combined in this figure. The fifth x-feature responds to four different things: a toilet, a bed, a dining chair and an oven. SRAE's reasoning process is still distributed across a set of network weights, and so the representations are less discrete than ERIC and the general interpretation of an x-feature is less specific. Recall also that each x-feature is derived from functions over multiple kernels in l^e, rather than from a single kernel, and so are less faithful representations over the ker-



(a) ERIC: Size = #Atoms (b) SRAE: Size = #Weights

Figure 4: The size of the extracted program, measured as the total number of atoms (ERIC) or connections (SRAE) used, shows strong correlation with the number of classes.

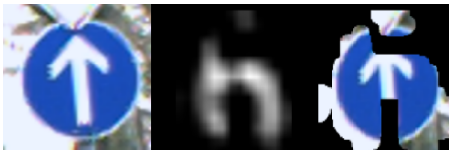


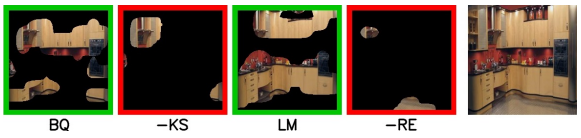
Figure 5: Visualising a kernel by normalising its output, thresholding at > 0.03 and then using this to mask the input.

nels in l^e than ERIC would offer. However none of this is to say that the explanations are bad as in most cases the masked regions are reasonable evidence of their classes and overall SRAE's explanations yield higher fidelity than ERIC's.

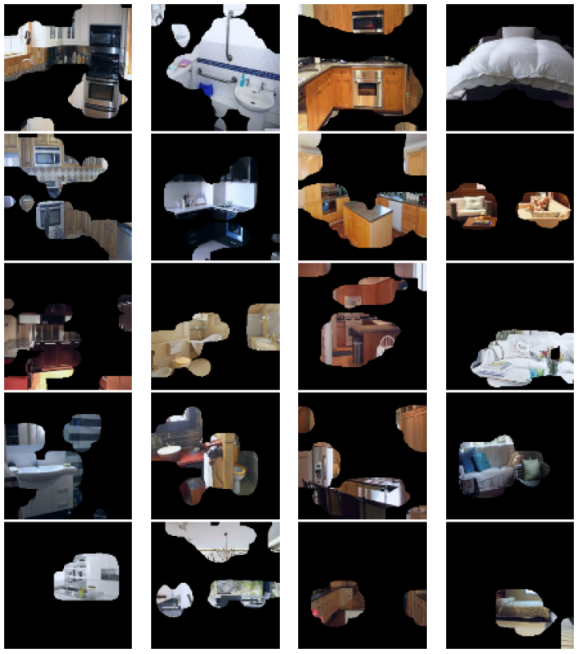
Figs. 8 and 9 show examples from GTSRB. Despite the simplicity and symbolic nature of traffic signs, the explanations are not very inuitive. This is interesting because it demonstrates that CNNs trained in the traditional way (i.e. not explainable by design) do not necessarily reason like humans do despite yielding reasonable accuracy. Fig.8 presents ERIC's reasoning behind the correct classification of a 'national speed limit' sign. Humans would recognise this as a white circle with a diagonal black line. The rule contains one kernel (CO) which appears to react to white circles in 4/5 samples shown (Fig. 8c), but of the two other positive conditions, one appears to represent arrows in blue circles and the other triangles. Both do contain white diagonal edges, however, as does the national limit sign. For SRAE (Fig. 9) some signs are somehow recognised purely by their edges, without considering the symbols on the signs.

Discussion and Future Work

Some existing works reduce multi-class problems to one-versus-all probems when generating explanations that many be deployed as classifiers in their own right (Qi, Khorram, and Fuxin 2021; Zhang et al. 2018a; Zhang, Nian Wu, and Zhu 2018). We have shown that both ERIC and SRAE may be applied to convolutional layers for at least some multi-class cases without reducing to one-vs-all and we hope to see more knowledge extraction methods evaluated without reducing to one-vs-all. The 10-class MNIST appears to be an exception to the tendency to reduce to one-vs-all (Frosst and Hinton 2017; Odense and Garcez 2020) and we have now shown that a set of at least 43 road signs could also be a suitable benchmark, with a fidelity of at least 73% achievable for the test data (table 2).



(a) A test image satisfying the rule. Green and red borders show conditions which must evaluate as true or false respectively.



(b) BQ (c) KS (d) LM (e) RE

Figure 6: ERIC rule $BQ \wedge \neg KS \wedge LM \wedge \neg RE \rightarrow Kitchen$, with visualisations of each kernel based on the training set. A possible interpretation is $Cabinets1 \wedge \neg Ceramics \wedge Cabinets2 \wedge \neg Upholstry \rightarrow Kitchen$ - 'If there are cabinets and no ceramics or upholstry then it is a kitchen.'

Our findings suggest that however an explanation is represented, factors affecting fidelity of M^* are ultimately affected by factors that affect the test accuracy of M. The two scenarios for which we observe the best success are MNIST and GTSRB, both of which are very symbolic and have very distinct edges, whereas the weakest performance was observed for photographic datasets which tend to have less distinct edges. The fact that explainability methods are unsuitable for less accurate original models is less of a concern when one considers that such models are less likely to be deployed in real-world applications anyway, especially for safety-critical scenarios. For example, the consequences of an autonomous vehicle misclassifying a road sign could be fatal and some investigation would be required. Software such as ERIC or SRAE could be useful here.

ERIC's fidelity is not quite so strong as that of SRAE. However there are innumerable ways of configuring both models, especially SRAE, which may take on extra neural network layers, types of layers, etc.

Furthermore the results of both of the configurations used are comparable and the explanations take completely differ-

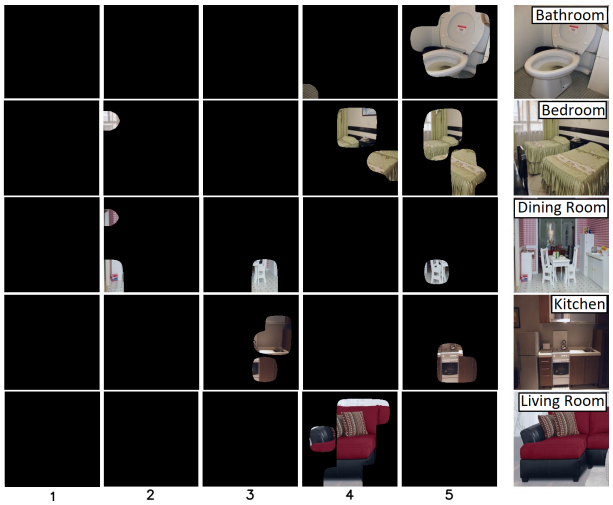
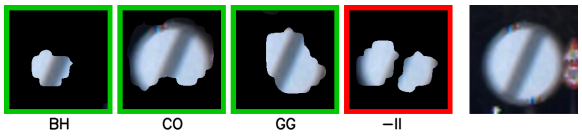


Figure 7: Places explanations from SRAE. x-feature 5 activates for multiple things (toilet, bed, chair and oven)

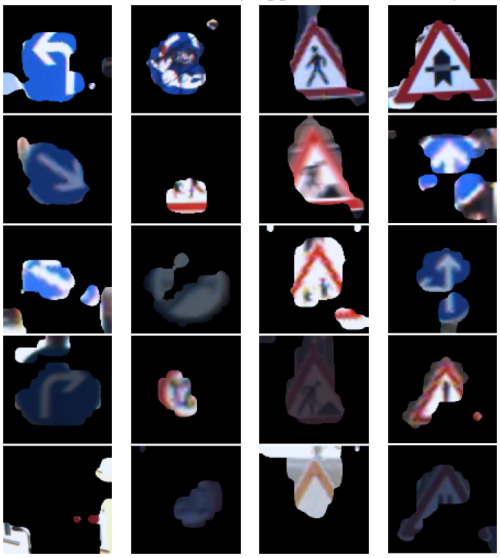
ent forms, offering users some choice. In SRAE the reasoning used by the explanation model is still distributed across a set of continuous network weights, albeit a smaller set than is used in the original CNN. ERIC's reasoning is more discrete and expressive. Also, in SRAE, all explanations of length n for all samples must be selected from the same set of n x-features. ERIC also offers explanations of length n but x-features may be selected from a larger number of $K_{l^e} \geq n$ atoms. In both cases, it is still up to a human observer to assign their own labels or meaning to the x-features. This will likely be easier for ERIC in the sense that x-features cluster concepts selectively, but more difficult in the sense that there are more x-features. In SRAE an x-feature is less likely to isolate a single concept across the space of all possible explantions; though for a single instance the visualised features nonetheless represent legitimate evidence of the conclusions they draw. SRAE would be preferred for reasoning with higher fidelity and a more compact and continuous explanation space. ERIC would be preferred for more discrete and expressive explanations with a clearer vocabulary and more faithful representation of kernels. A comparative study based on human interaction as was originally performed for SRAE would be welcome.

Conclusions

We presented a study of two post-hoc methods for translating CNNs into independent, explainable neuro-symbolic reasoning systems, extending earlier work to show that the robustness of both of these methods extends to multiple classes as long as the original CNN itself yields high test accuracy. We observed that the high-accuracy scenarios on which ERIC and SRAE perform better tend to be those in which images have more discrete edges. Finally, we discussed the differences between the two methods: SRAE offers marginally stronger fidelity than ERIC, but in sacrificing some fidelity ERIC is able to offer a much larger and more discrete vocabulary.



(a) The rule as correctly applied to a test image.



(b) BH (c) CO (d) GG (e) II

Figure 8: The rule $BH \wedge CO \wedge GG \wedge \neg II \rightarrow National\_Limit$. Each kernel is represented by five training images that activate it.

Figure 9: Traffic sign explanations from SRAE.

Acknowledgements

The authors would like to thank colleagues Hiroya Inakoshi, Yu Shanshan, Tetsu Yamamoto, Karolina Szafarz and Tomasz Raszkowski; and collaborators Artur Garcez and Kwun Ngan for their support and comments.

References

Andrews, R.; Diederich, J.; and Tickle, A. B. 1995. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-based systems*, 8(6): 373–389.

Bader, S.; and Hitzler, P. 2005. Dimensions of neural-symbolic integration-a structured survey. *arXiv preprint cs/0511042*.

Besold, T. R.; d'Avila Garcez, A. S.; Bader, S.; Bowman, H.; Domingos, P.; Hitzler, P.; Kühnberger, K.; Lamb, L. C.; Lowd, D.; Lima, P. M. V.; et al. 2017. Neural-symbolic learning and reasoning: A survey and interpretation. *arXiv preprint arXiv:1711.03902*.

Binder, A.; Bach, S.; Montavon, G.; Müller, K.; and Samek, W. 2016. Layer-wise relevance propagation for deep neural network architectures. In *Information Science and Applications (ICISA) 2016*, 913–922. Springer.

Chen, C.; Li, O.; Barnett, A.; Su, J.; and Rudin, C. 2019. This looks like that: deep learning for interpretable image recognition. *arXiv preprint arXiv:1806.10574*.

d'Avila Garcez, A. S.; Broda, K.; and Gabbay, D. M. 2002. *Neural-Symbolic Learning Systems: Foundations and Applications*. Springer Science & Business Media.

Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A. ???? The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

Frosst, N.; and Hinton, G. 2017. Distilling a Neural Network Into a Soft Decision Tree. *arXiv preprint arXiv:1711.09784*.

Hammer, B.; and Hitzler, P. 2007. *Perspectives of neural-symbolic integration*, volume 8. Springer Heidelberg:.

Hase, P.; Chen, C.; Li, O.; and Rudin, C. 2019. Interpretable image recognition with hierarchical prototypes. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, 32–40.

Kim, E.; Kim, S.; Seo, M.; and Yoon, S. 2021. XProtoNet: Diagnosis in Chest Radiography with Global and Local Explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15719–15728.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Odense, S.; and Garcez, A. d. 2020. Layerwise Knowledge Extraction from Deep Convolutional Networks. *arXiv preprint arXiv:2003.09000*.

Qi, Z.; Khorram, S.; and Fuxin, L. 2021. Embedding deep networks into visual explanations. *Artificial Intelligence*, 292: 103435.

Quinlan, J. R. 1993. C4.5: Programming for machine learning. *The Morgan Kaufmann Series in Machine Learning, San Mateo, CA: Morgan Kaufmann*, 38: 48.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. 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*, 1135–1144. ACM.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2018. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence*.

Rymarczyk, D.; Struski, Ł.; Tabor, J.; and Zieliński, B. 2021. ProtoPShare: Prototypical Parts Sharing for Similarity Discovery in Interpretable Image Classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1420–1430.

Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Stallkamp, J.; Schlipsing, M.; Salmen, J.; and Igel, C. 2012. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, (0): –.

Townsend, J.; Chaton, T.; and Monteiro, J. M. 2019. Extracting Relational Explanations From Deep Neural Networks: A Survey From a Neural-Symbolic Perspective. *IEEE transactions on neural networks and learning systems*.

Townsend, J.; Kasioumis, T.; and Inakoshi, H. 2020. ERIC: Extracting Relations Inferred from Convolutions. In *Proceedings of the Asian Conference on Computer Vision*.

Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; and Perona, P. 2010. Caltech-UCSD birds 200.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhang, Q.; Cao, R.; Shi, F.; Wu, Y. N.; and Zhu, S. 2018a. Interpreting CNN knowledge via an explanatory graph. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhang, Q.; Nian Wu, Y.; and Zhu, S. 2018. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8827–8836.

Zhang, Q.; Yang, Y.; Wu, Y. N.; and Zhu, S. 2018b. Interpreting CNNs via decision trees. *arXiv preprint arXiv:1802.00121*.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zintgraf, L. M.; Cohen, T. S.; Adel, T.; and Welling, M. 2017. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*.