Explaining Deep Learning Hidden Neuron Activations using Concept Induction

Abhilekha Dalal[1], Md Kamruzzaman Sarker[2], Adrita Barua[1], and Pascal Hitzler[1]

[1] Department of Computer Science, Kansas State University, USA
[2] Department of Computing Sciences, University of Hartford, USA
adalal@ksu.edu, sarker@hartford.edu, adrita@ksu.edu, hitzler@ksu.edu

Abstract. One of the current key challenges in Explainable AI is in correctly interpreting activations of hidden neurons. It seems evident that accurate interpretations thereof would provide insights into the question what a deep learning system has internally *detected* as relevant on the input, thus lifting some of the black box character of deep learning systems.

The state of the art on this front indicates that hidden node activations appear to be interpretable in a way that makes sense to humans, at least in some cases. Yet, systematic automated methods that would be able to first hypothesize an interpretation of hidden neuron activations, and then verify it, are mostly missing.

In this paper, we provide such a method and demonstrate that it provides meaningful interpretations. It is based on using large-scale background knowledge – a class hierarchy of approx. 2 million classes curated from the Wikipedia Concept Hierarchy – together with a symbolic reasoning approach called *concept induction* based on description logics that was originally developed for applications in the Semantic Web field.

Our results show that we can automatically attach meaningful labels from the background knowledge to individual neurons in the dense layer of a Convolutional Neural Network through a hypothesis and verification process.

1 Introduction

The origins of Artificial Intelligence trace back several decades ago, and AI has been successfully applied to multiple complex tasks such as image classification [25], speech recognition [11], language translation [2], drug design [31], treatment diagnosis [9], and climate sciences [21], as an instance for just a few. Artificially intelligent machines reach exceptional performance levels in learning to solve more and more complex computational problems by possessing the capabilities of learning, thinking, and adapting – mimicking human behavior to some extent, making them crucial for future development.

Despite their success in a wide variety of tasks, there is a general distrust of their results. Powerful AI machines particularly Deep Neural Networks, are

hard to explain and are often referred to as "Black Boxes" simply because there are no clear human-understandable explanations as to why the network gave the particular output. Many cases have been reported; for example, In 2019 Apple co-founder Steve Wozniak accused Apple Card of gender discrimination, claiming that the card gave him a credit limit that was ten times higher than that of his wife, even though the couple shares all property.[3]. In CEO image search, while 27% of US CEOs were women, only 11% of the top image results for "CEOs" were featured as women.[4] In continuation to the mentioned observation, the output of a network's classification can be altered by introducing Adversarial examples [6], and there are many more attack techniques. It becomes a need to understand the reasoning behind how a system behaves and generates an output in a human-interpretable way, especially since the popularity of these systems has grown to such an extent that these systems are responsible for decisions previously taken by human beings in safety-critical situations, for example like self-driving cars [8], drug discovery and treatment recommendations [27,12].

Explainable AI has been pursued for several years already, and the quest for efficient algorithms to generate human-understandable explanations has led to a significant number of contributions based on different approaches. or internal unit summarizing [36,6]. Improvements in deep learning show that neurons in the hidden layer of the neural network can detect human-interpretable concepts that were not explicitly taught to the network, such as objects, parts, gender, context, sentiment etc [4,18,24].

In our approach which we present in this paper, we make central use of *concept induction* [20], which has been developed for use in the Semantic Web field and is based on deductive reasoning over description logics, i.e., over logics relevant to ontologies, knowledge graphs and generally the Semantic Web field [17,16]. In a nutshell – and more details are given below – a concept induction system accepts three inputs, a set of positive examples P, a set of negative examples N, and a knowledge base (or ontology) K, all expressed as description logic theories, and where we have x occurring as instances (constants) in K for all $x \in P \cup N$. It then returns description logic class expressions E such that $K \models E(p)$ for all $p \in P$ and $K \not\models E(q)$ for all $q \in N$. If no such class expressions exist, then it returns approximations for E together with a number of accuracy measures. In this paper, for scalability reasons, we use the heuristic concept induction system ECII [28] together with a background knowledge base that consists only of a class hierarchy, however with approximately 2 million classes, as presented in [29]. Given a hidden neuron, P is a set of inputs to the deep learning system that activate the neuron, and N is a set of inputs that do not activate the neuron. Inputs are annotated with classes from the background knowledge for concept induction, however these annotations and the background knowledge are not part of the input to the deep learning system.

[3] https://worldline.com/en/home/knowledgehub/blog/2021/january/ever-heard-of-the-aI-black-box-problem.html

[4] https://www.mckinsey.com/featured-insights/artificial-intelligence/tackling-bias-in-artificial-intelligence-and-in-humans

As we will see below, this approach is able to provide meaningful explanations for hidden neuron activation.

The rest of this paper is organized as follows. Section 2 discusses relevant work in the filed of generating explanations using knowledge graph. Sections 3 present our study design and Section 4 discusses the results of our study along with the findings and their implications. Finally, Section 5 sums up the paper and proposes some possibilities for future research.

2 Related Work

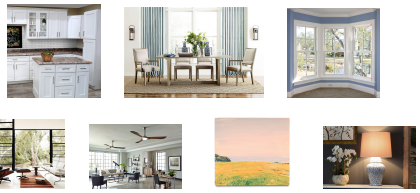
Explainable AI has been intensively studied since the 1970s [22]; and the model's explainability can be translated in many ways - interpretable, understandable, justified, and evaluable.

The segment of explainable AI methods focuses on interpreting the inner workings of black box models, such as identifying input features by training explanation networks that generate human-readable explanations [15] or create models alternatives to summarize the behavior of a complex network [26]. Other approaches include such as the use of salience maps where the explanations summarize the contribution of each pixel to predictions [3] or visual cues [35,32] or counterfactuals [7].

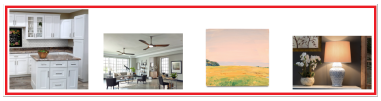
The literature demonstrates that combinations of neurons can encode meaningful and insightful information [19,5].Justifying the result of a neural network requires a defined language that incorporate elements of reasoning that use knowledge bases to create human-understandable, yet unbiased explanations[10].

Knowledge graphs and the structured web represent a valuable form of machine – readable, domain – specific knowledge; available connected datasets can serve as a knowledge base for an AI system to explain its decisions to its users in a better way. The Web Ontology Language (OWL) provides a basis for verbose descriptions of entities and their relationships through description logics [1]. Deep deductive reasoning can be described as one of generating complex description logic class expressions over the knowledge graph and is based on rich concept hierarchies that play an important role in generating human – readable satisfactory explanations. We briefly discussed some recent works doing logical reasoning using deep networks.

[37,19,36], methods have been proposed and demonstrated that adding semantic annotations to label objects that activate neurons in the hidden layers of common CNN architectures provides human-readable explanations. Nonetheless, these approaches need to improve in terms of producing deeper explanations generated over more expressive background knowledge. [23] follows the effort of [30], by semi-automating the DL Learner tool, which provides explanations to ML algorithms using semantic background knowledge. However, while DL-Learner is a very useful system in producing theoretically correct results has significant performance issues in some scenarios, such as a single run of DL-Learner can easily take over two hours; in contrast the scenario easily necessitates thousands of such runs.



(a) Examples of images collected for neuron-5 from google using the lists of concepts for Case-1.



(b) Images that didn't activate the neuron-5 for Case-1.

(c) Images that activate the neuron-5 for Case-1.

Fig. 1: Case - I

The main motivation of the proposed work is to automate the assignment of human-interpretable explanations for the activations of neurons in the hidden – dense layer of CNNs; Using Wikipedia's rich class hierarchy of around 2 million classes with an improved concept induction approach (in terms of running time by 1-2 orders of magnitude while maintaining accuracy of results products) known as ECII.

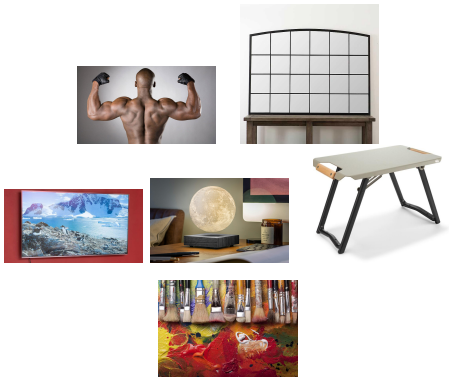
3 Research Method

This work includes the implementation of explaining the activation pattern of neurons in hidden layers of CNN i.e. dense layer in this case, using Resnet50V2 architecture and ECII – concept induction explanation generation algorithm. We also tested other architectures to achieve better accuracy and found that Resnet50V2 gives the highest accuracy. The subsections discuss the steps followed for implementing the system in a more detailed manner.

3.1 Training Convolutional Neural Network

Dataset 1) The ADE20K [38] semantic segmentation dataset from the Massachusetts Institute of Technology contains more than 27K scene-based images from the SUN and Places databases, extensively annotated with pixel-level objects and object part labels. There are 150 semantic categories including sky, road, grass, and discrete objects like person, car, and bed. The current version of the dataset contains the following:

- 25,574 for training and 2,000 for testing from 365 scenes.
- 707,868 unique object along with their WordNet definition and hierarchy.
- 193,238 parts of annotated objects and parts of parts.
- Polygon annotations with attributes, annotation time, and depth order.



(a) Examples of images collected for neuron-5 from google using the lists of concepts for Case-2.

(b) Images that activate the neuron-5 for Case-2.

(c) Images that didn't activate the neuron-5 for Case-2.

Fig. 2: Case - II

We only considered the subset of scenes in the ADE20k Dataset; the classes that were considered for this work are ten classes with the highest number of images – bathroom, bedroom, building facade, conference room, dining room, highway, kitchen, living room, skyscraper, and street.

2) For verification purposes of the activation pattern of each neuron in corresponding to identified concepts for that respective neuron, we used Google images – simply because the system should be easy to use for any user. It should be able to detect concepts and give us the reasoning for its classification category of any random image from the largest crawling search engine.

Tested Networks We analyzed many Convolutional neural network (CNN) architectures to achieve better and higher accuracy such as Vgg16 [33], InceptionV3 [34]; in Resnet we tried different versions like – Resnet50, and Resnet50V2, Resnet101, Resnet152V2 architecture [13,14].

Each neural network was fine-tuned with a dataset of 6187 images (training and validation set) of size 224*224 for 20 epochs to classify images into 10 scene categories using the ADE20K dataset. The optimization algorithm used was Adam, with a categorical cross-entropy loss function and a learning rate of 0.001. The accuracy achieved by each architecture along with validation accuracy is summarized in table 1.

Clearly, ResNet50V2 achieved the highest accuracy – 92.47% on the training dataset and 87.50% on the validation dataset, proving to be the best network out of all.

Architectures	Training acc	Validation acc
Vgg16	80.05%	46.22%
InceptionV3	89.02%	51.43%
Resnet50	35.01%	26.56%
Resnet50V2	**92.47%**	**87.50%**
Resnet101	53.97%	53.57%
Resnet152V2	94.53%	51.04%

Table 1: Performance of different architectures on ADE20K dataset

Activations of Trained Network We tested the Resnet50V2 with 1370 images and retrieved the activations of the dense layer, i.e., the layer before the output layer. Though technically, the layer before the output layer is the dropout layer, we chose not to analyze the activations of the dropout layer since the dropout layer is a mask that negates the contribution of some neurons towards the next layer and leaves all others unmodified.

The activations of 1370 images for the dense layer comprise 64 neurons contributing to the final decision of classifying each image as one of 10 classes.

Candidate Set of Neurons Next, out of 64 neurons, we chose some candidate sets of neurons based on the following criteria – only the neuron having more than 50% of activation values > 0 i.e., the neuron should have at least 680 values ($=$ 1370/2, 1370 being total images) that are greater than 0. Choosing such neurons would simply mean that these are frequently activated nodes, which would be a good choice to analyze before exploring any other possibilities. Following the idea, the neurons selected for analysis were neuron numbers – 4, 5, 6, 7, 9, 11, 12, 13, 15, 16, 22, 23, 27, 29, 34, 35, 36, 37, 39, 45, 52, 54, 55, 56, 58, 59, 60, 62, 63.

ECII - Preliminaries As mentioned concept induction is an explanation generation algorithm over description logic which takes in three inputs – a positive set of images, a negative set of images and a knowledge base. For our approach, we use ECII –improved on DL Learner by the magnitude of order 2.

For a given neuron, a positive set of images that activates the said neuron, and a negative set of images that do not activate the given neuron. How do we decide that an image activates a neuron and therefore that image is positive, and in the same way for negative set of images. To decide on activation, we considered and analyzed – a threshold value around the activation values with the following three different criteria:-

- CASE-I – positive set will have images with $>= 50\%$ activation of the highest value, lets say if the highest activation value is 12 for neuron_x then all images (1370, is the total number) having an activation value of 6 or more than 6 will be positive set and so negative set will have images that were $< 50\%$ i.e all the images with less than 6 including 0.

- CASE-II – positive set will have images with >= 50% activation of the highest activation value and negative set as the images that were just zero, i.e excluding images that are 0 < images < 50%.
- CASE-III – positive set will have images with anything > zero i.e this will include all images that are 0 < images <= highest value and negative set as the images that were just zero, i.e excluding images that are 0 < images.

For the knowledge base, we mapped all the 1370 images with Wikipedia's rich class hierarchy of 2 million classes.

ECII - Analysis Now that we have a knowledge base and a set of positive and negative images based on three cases, we run ECII with all three inputs from each case defined above for all chosen candidate sets of the neurons. ECII returns a list of class expressions such that it best describes the positive set of images while excluding all negative images, sorted by coverage score.

Coverage score can be formulated using the following formula:

$$coverage(E) = \frac{|P \cap Z1| + |N \cap Z2|}{|P \cup N|}$$

Where,

$$Z1 = K \models E(p) \text{ for all } p \in P,$$
$$Z2 = K \not\models E(n) \text{ for all } n \in N,$$
P is the set of all positive instances,
N is the set of all negative instances, and
K is the knowledge base provided to ECII as input.

We chose to look at the first 50 expressions out of all returned by ECII in text format, simply because the list of expressions could have many duplicate concepts.

Example 1. An example of the – explanation ECII came up with looks like
solution 1 ∃ imageContains.((WN_Table) ⊓ (Bed))
solution 3 ∃ imageContains.(:WN_Table)

indicating the presence of a table and bed in one of the images from the positive set. We collected all distinctive keywords as concepts(in this case – Table and Bed) from the list since solutions could have overlapping concepts, resulting in a reduced list of concepts.

This returned list of concepts for each neuron would give us the intuition of what contributes towards the activation of the respective neuron.

Example 2. As an example lets see the the list of class expression returned by ECII for neuron 5 and its corresponding reduced list of concepts.
solution 1: ∃ :imageContains.(:WN_Table)
solution 2: ∃ :imageContains.(:Floor)

solution 3: ∃ :imageContains.(:WN_Floor)
solution 4: ∃ :imageContains.(:WN_Flooring)
solution 5: ∃ :imageContains.(:Window)
solution 6: ∃ :imageContains.(:WN_Window)
solution 7: ∃ :imageContains.((:WN_Flooring) ⊓ (:Window))
solution 8: ∃ :imageContains.((:Window) ⊓ (:Floor))
solution 9: ∃ :imageContains.((:WN_Flooring) ⊓ (:Floor))
solution 10: ∃ :imageContains.((:Ceiling) ⊓ (:WN_Table))
solution 11: ∃ :imageContains.(:Ceiling)
solution 12: ∃ :imageContains.(:WN_Ceiling)
solution 13: ∃ :imageContains.(:WN_Windowpane)
solution 14: ∃ :imageContains.(:WN_Leg)
solution 15: ∃ :imageContains.(:Picture)
solution 16: ∃ :imageContains.(:WN_Painting)
solution 17: ∃ :imageContains.(:WN_Picture)
solution 18: ∃ :imageContains.(:Leg)
solution 19: ∃ :imageContains.((:WN_Table) ⊓ (:Leg))
solution 20: ∃ :imageContains.((:WN_Painting) ⊓ (:WN_Ceiling))
solution 21: ∃ :imageContains.((:WN_Leg) ⊓ (:WN_Window))
solution 22: ∃ :imageContains.(:Chair)
solution 23: ∃ :imageContains.(:WN_Chair)
solution 24: ∃ :imageContains.(:WN_Lamp)
solution 25: ∃ :imageContains.((:WN_Lamp) ⊓ (:WN_Floor))
solution 26: ∃ :imageContains.((:WN_Windowpane) ⊓ (:WN_Painting))
solution 27: ∃ :imageContains.(:Back)
solution 28: ∃ :imageContains.(:WN_Back)
solution 29: ∃ :imageContains.((:Back) ⊓ (:WN_Flooring))
solution 30: ∃ :imageContains.((:WN_Floor) ⊓ (:WN_Back))
solution 31: ∃ :imageContains.((:WN_Windowpane) ⊓ (:WN_Ceiling))
solution 32: ∃ :imageContains.((:Ceiling) ⊓ (:Leg))
solution 33: ∃ :imageContains.((:Floor) ⊓ (:Table))
solution 34: ∃ :imageContains.(:Table)
solution 35: ∃ :imageContains.((:WN_Back) ⊓ (:WN_Windowpane))
solution 36: ∃ :imageContains.((:Chair) ⊓ (:Ceiling))
solution 37: ∃ :imageContains.(:Arm)
solution 38: ∃ :imageContains.(:WN_Arm)
solution 39: ∃ :imageContains.((:WN_Window) ⊓ (:WN_Lamp))
solution 40: ∃ :imageContains.((:Back) ⊓ (:Window))
solution 41: ∃ :imageContains.((:WN_Floor) ⊓ (:WN_Windowpane))
solution 42: ∃ :imageContains.((:Back) ⊓ (:Floor))
solution 43: ∃ :imageContains.((:WN_Window) ⊓ (:WN_Floor))
solution 44: ∃ :imageContains.((:Chair) ⊓ (:WN_Table))
solution 45: ∃ :imageContains.(:Top)
solution 46: ∃ :imageContains.(:WN_Top)
solution 47: ∃ :imageContains.((:Table) ⊓ (:WN_Chair))

solution 48: ∃ :imageContains.((:Floor) ⊓ (:WN_Chair))
solution 49: ∃ :imageContains.((:Leg) ⊓ (:Picture))
solution 50: ∃ :imageContains.(:WN_Cabinet)

And after eliminating duplicate concepts from the above class expressions, we get a reduced list of concepts as –

arm, back, cabinet, ceiling, chair, floor, flooring, lamp, leg, painting, picture, table, top, window, windowpane

The next step would be to verify the activation of neurons by collecting some more data that is more generic and could serve as solid verification and test it through the model and see if we get the same activations for the neurons; we collected google images corresponding to the resultant keywords of the list for each neuron. In this case – collect images of arm, back, cabinet, ceiling, chair, floor, flooring, lamp, leg, painting, picture, table, top, window, and windowpane from the google search engine.

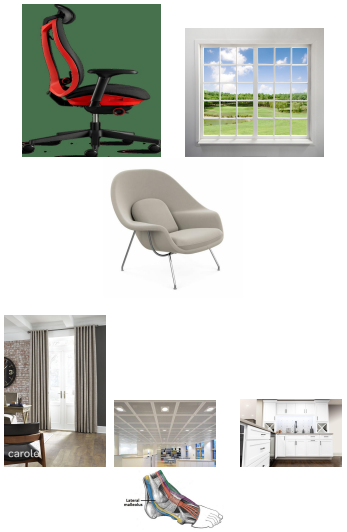
Collection of Google Images We used a python script to download Google images for each keyword in the list. For each keyword, it collects the first 200 images that appear in the google search. After that, we manually checked for duplicates and removed them. After cleaning the duplicates we have at least 140 images for each keyword. For example, for the keyword, 'base' google search comes up with all kinds of images including bed frames to military bases. However, some search for keywords like 'edifice' collects images of a particular model of watch named edifice, which in this case is not what we wanted. But we take the google results as it appears and evaluate our model on them.

Activations on Google Images Once we have the dataset from google ready for all neurons (29 being total as chosen candidate set), we test this new dataset for each neuron through our trained model – Resnet50V2 and get the activations of the dense layer.

4 Results and Discussion

We analyzed three different criteria as mentioned in subsection 3.1 – ECII-Preliminaries, to see what we could call the best scenario for the activation of the neuron –

- CASE-I – positive set will have images with $>= 50\%$ activation of the highest value, and the negative set will have images that were $< 50\%$.
- CASE-II – positive set will have images with $>= 50\%$ activation of the highest activation value and negative set as the images that were just zero.
- CASE-III – positive set will have images with anything $>$ zero and negative set as the images that were just zero.



(a) Examples of images collected for neuron-5 from google using the lists of concepts for Case-3.

(b) Images that activate the neuron-5 for Case-3.

(c) Images that didn't activate the neuron-5 for Case-3.

Fig. 3: Case - III

ECII was run for all neurons taking each case at a time along with Wikipedia as a Knowledge base; in total we did 29*3 = 87 ECII analysis.

From each ECII analysis, we got a list of class expressions sorted by coverage score for the respective neuron – looked at the first 50 expressions, and reduced it to a shorter list by eliminating any duplicate keywords. These keywords indicate the activation of neurons by the presence of these concepts. Table 2 lists the concepts we got from ECII corresponding to each case, representing the neuron's activation for neuron number 5.

To verify if these concepts actually play a role for neurons in deciding the output for the network, collected google images corresponding to the reduced list of concepts for each neuron.

In case of neuron number 5 CASE–I, google images were collected corresponding to **arm, back, cabinet, ceiling, chair, floor, flooring, lamp, leg, painting, picture, table, top, window, windowpane.**

For CASE–II, google images were collected corresponding to **arm, back, cabinet, ceiling, chair, floor, flooring, lamp, leg, painting, picture, table, wall, windowpane.**

For CASE–III, google images were collected corresponding to **cabinet, ceiling, chair, curtain, cushion, drapery, floor, flooring, lamp, leg, painting, picture, shade, table, table_lamp, wall, windowpane.**

Case-I	Case-II	Case-III
arm	arm	cabinet
back	back	ceiling
cabinet	cabinet	chair
ceiling	ceiling	curtain
chair	chair	cushion
floor	floor	drapery
flooring	flooring	floor
lamp	lamp	flooring
leg	leg	lamp
painting	painting	leg
picture	picture	painting
table	table	picture
top	wall	shade
window	windowpane	table
windowpane		table_lamp
		wall
		windowpane

Table 2: List of concepts activating neuron_5 for case I, II, III

Around 200 images were collected for each concept in the list, making a total of around 4000-5000 images for each neuron. In total 4000*87 = 348000 images were collected.

The new google image dataset was divided into 80-20 ratio and 80% of them were tested by the trained model for verification and activations of the dense layer (n-1 layer) were analyzed for each neuron. In total there were 87 dense layer activations; each dense layer activation consists of 64 neurons; we only look for the activation value of the desired neuron number. The activation percentage for each neuron is summarized case–wise in table 3. The figure shows the examples of the google image dataset collected for neuron_5 in each case, along with images that activated the neuron and those that didn't activate the neuron.

Some observations from the table –

- 11 neurons – neuron number 4, 9, 11, 12, 15, 16, 23, 27, 60, 62, and 63 got activated by more than 90% activations in all the three cases.
- 10 neurons – neuron numbers 6, 13, 29, 36, 37, 39, 45, 52, 54, 59 were below 1% activations in all three cases.
- the rest activations are in the range of 1 – 56.52%.

We can say that the criteria we chose for deciding the activation for the positive and negative set of images – as two inputs for ECII, doesn't have much impact on the activation percentage of the neurons as there is a slight difference in the percentage value of Ist, IInd and IIIrd Case.

Table 4 shows the evaluation of 29 neurons for the remaining 20% of the Google Image Dataset. The activation percentage for each neuron is listed for all three cases.

	Case-I	Case-II	Case-III
neuron4	100%	100%	100%
neuron5	35.01%	36.84%	38.38%
neuron6	0.44%	0.40%	0.24%
neuron7	6.31%	7.48%	5.71%
neuron9	99.90%	100%	99.97%
neuron11	99.00%	99.00%	99.00%
neuron12	95.20%	95.20%	95.20%
neuron13	0.05%	0.06%	0.05%
neuron15	99.93%	99.96%	100%
neuron16	99.94%	99.97%	99.97%
neuron22	37.32%	26.00%	26.24%
neuron23	100%	100%	100%
neuron27	99.64%	99.65%	99.60%
neuron29	0.67%	0.67%	0.67%
neuron34	56.46%	56.46%	57.58%
neuron35	16.32%	16.31%	9.25%
neuron36	0%	0%	0%
neuron37	0%	0%	0%
neuron39	0.24%	0.20%	0.63%
neuron45	0%	0.09%	0%
neuron52	0.18%	0.24%	0.17%
neuron54	0%	0%	0%
neuron55	53.81%	47.88%	38.36%
neuron56	4.16%	4.06%	2.22%
neuron58	1.80%	1.80%	1.68%
neuron59	0%	0%	0%
neuron60	100%	99.96%	100%
neuron62	100%	100%	100%
neuron63	100%	100%	100%

Table 3: Activation percentage for each neuron with Google Images for all three cases.

At this point, we can say that by our hypothesis and our verification process – neurons get activated by the presence of concepts and concepts plays a role in deciding the output given by the network.

	Case-I	Case-II	Case-III
neuron4	100%	100%	100%
neuron5	34.49%	37.79%	38.10%
neuron6	0.5%	0.53%	0.19%
neuron7	0.14%	0.19%	0.20%
neuron9	100%	100%	100%
neuron11	98.83%	98.83%	98.83%
neuron12	94%	94%	94%
neuron13	0.20%	0.22%	0.20%
neuron15	100%	99.85%	100%
neuron16	100%	100%	100%
neuron22	35.13%	23.06%	25.62%
neuron23	100%	100%	100%
neuron27	99.74%	99.45%	99.77%
neuron29	1 %	1%	1%
neuron34	56.74%	56.74%	58.17%
neuron35	14.29%	17.62%	9.55%
neuron36	0.14%	0.14%	0.22%
neuron37	0.18%	0.20%	0.13%
neuron39	0.58%	0.39%	0.5%
neuron45	0.12%	0.27%	0.19%
neuron52	0.36%	0.19%	0.33%
neuron54	0.19%	0.22%	0.12%
neuron55	55.37%	46.27%	35.39%
neuron56	4.82%	4.32%	2.17%
neuron58	1.33%	1.33%	1.40%
neuron59	0.12%	0.14%	0.22%
neuron60	99.88%	100%	99.84%
neuron62	100%	100%	100%
neuron63	100%	100%	100%

Table 4: Activation percentage after doing evaluation for each neuron with Google Images for all three cases.

5 Conclusion and Future Work

This paper is an effort toward recognizing the activation pattern of neurons in the hidden layer of CNN architecture with the presence of abstract concepts. A novel approach using ECII as an explanation generation algorithm and Wikipedia as background knowledge was shown to quantify how well a concept is recognized

across the latest convolutional layer (specifically the dense layer) of a CNN. Through our verification and evaluation using Google Images, we have also reported on promising activation percentages to support our hypothesis.

Future work will incorporate the studying of the remaining neurons and will study the effect of the different thresholds for activation of the neuron. We will need to automate the whole process of getting the human-understandable explanation for the output of the network; given the classification of the network (output of the network) as an input, it should output the activation concepts for the neurons to limit the human-intervention and explain the decision of the network efficiently.

Acknowledgement. This work was supported by the U.S. Department of Commerce, National Science Foundation, under award number 2033521.

References

1. D. Allemang and J. Hendler. *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier, 2011.
2. M. Auli, M. Galley, C. Quirk, and G. Zweig. Joint language and translation modeling with recurrent neural networks. In *Proc. of EMNLP*, 2013.
3. S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.
4. A. Bau, Y. Belinkov, H. Sajjad, N. Durrani, F. Dalvi, and J. Glass. Identifying and controlling important neurons in neural machine translation. *arXiv preprint arXiv:1811.01157*, 2018.
5. D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.
6. D. Bau, J.-Y. Zhu, H. Strobelt, A. Lapedriza, B. Zhou, and A. Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020.
7. R. M. Byrne. Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning. In *IJCAI*, pages 6276–6282, 2019.
8. Z. Chen and X. Huang. End-to-end learning for lane keeping of self-driving cars. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1856–1860. IEEE, 2017.
9. H.-I. Choi, S.-K. Jung, S.-H. Baek, W. H. Lim, S.-J. Ahn, I.-H. Yang, and T.-W. Kim. Artificial intelligent model with neural network machine learning for the diagnosis of orthognathic surgery. *Journal of Craniofacial Surgery*, 30(7):1986–1989, 2019.
10. D. Doran, S. Schulz, and T. R. Besold. What does explainable ai really mean? a new conceptualization of perspectives. *arXiv preprint arXiv:1710.00794*, 2017.
11. A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772. PMLR, 2014.
12. W. Hariri and A. Narin. Deep neural networks for covid-19 detection and diagnosis using images and acoustic-based techniques: a recent review. *Soft computing*, 25(24):15345–15362, 2021.

13. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.
14. K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.
15. L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. Generating visual explanations. In *European conference on computer vision*, pages 3–19. Springer, 2016.
16. P. Hitzler. A review of the semantic web field. *Commun. ACM*, 64(2):76–83, 2021.
17. P. Hitzler, M. Krötzsch, and S. Rudolph. *Foundations of Semantic Web Technologies*. Chapman and Hall/CRC Press, 2010.
18. A. Karpathy, J. Johnson, and L. Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.
19. B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.
20. J. Lehmann and P. Hitzler. Concept learning in description logics using refinement operators. *Mach. Learn.*, 78(1-2):203–250, 2010.
21. Y. Liu, E. Racah, J. Correa, A. Khosrowshahi, D. Lavers, K. Kunkel, M. Wehner, W. Collins, et al. Application of deep convolutional neural networks for detecting extreme weather in climate datasets. *arXiv preprint arXiv:1605.01156*, 2016.
22. S. T. Mueller, R. R. Hoffman, W. Clancey, A. Emrey, and G. Klein. Explanation in human-ai systems: A literature meta-review, synopsis of key ideas and publications, and bibliography for explainable ai. *arXiv preprint arXiv:1902.01876*, 2019.
23. T. Procko, T. Elvira, O. Ochoa, and N. Del Rio. An exploration of explainable machine learning using semantic web technology. In *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*, pages 143–146. IEEE, 2022.
24. C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.
25. M. Ramprasath, M. V. Anand, and S. Hariharan. Image classification using convolutional neural networks. *International Journal of Pure and Applied Mathematics*, 119(17):1307–1319, 2018.
26. M. T. Ribeiro, S. Singh, and C. Guestrin. " 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*, pages 1135–1144, 2016.
27. A. S. Rifaioglu, E. Nalbat, V. Atalay, M. J. Martin, R. Cetin-Atalay, and T. Doğan. Deepscreen: high performance drug–target interaction prediction with convolutional neural networks using 2-d structural compound representations. *Chemical science*, 11(9):2531–2557, 2020.
28. M. K. Sarker and P. Hitzler. Efficient concept induction for description logics. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3036–3043. AAAI Press, 2019.
29. M. K. Sarker, J. Schwartz, P. Hitzler, L. Zhou, S. Nadella, B. S. Minnery, I. Juvina, M. L. Raymer, and W. R. Aue. Wikipedia knowledge graph for explainable AI. In B. Villazón-Terrazas, F. Ortiz-Rodríguez, S. M. Tiwari, and S. K. Shandilya,

editors, *Knowledge Graphs and Semantic Web - Second Iberoamerican Conference and First Indo-American Conference, KGSWC 2020, Mérida, Mexico, November 26-27, 2020, Proceedings*, volume 1232 of *Communications in Computer and Information Science*, pages 72–87. Springer, 2020.

30. M. K. Sarker, N. Xie, D. Doran, M. Raymer, and P. Hitzler. Explaining trained neural networks with semantic web technologies: First steps. *arXiv preprint arXiv:1710.04324*, 2017.
31. M. H. Segler, T. Kogej, C. Tyrchan, and M. P. Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, 4(1):120–131, 2018.
32. 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. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.
33. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
34. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
35. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.
36. B. Zhou, D. Bau, A. Oliva, and A. Torralba. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2131–2145, 2018.
37. B. Zhou, Y. Sun, D. Bau, and A. Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.
38. B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.