Semantic Interpretability of Convolutional Neural Networks by Taxonomy Extraction

Vitor A. C. Horta[1,*,†], Robin Sobczyk[2,†], Maarten C. Stol[3,†] and Alessandra Mileo[1,†]

[1] *Insight Centre for Data Analytics at Dublin City University, Ireland*

[2] *École Normale Supérieure de Paris-Saclay, France*

[3] *BrainCreators, The Netherlands*

Abstract

Interpretability of Convolutional Neural Networks (CNNs) is often crucial for their application in real world scenarios. We aim to provide such interpretations in terms of the semantic content and conceptual structure CNNs acquire from their training data. Recent advances in Explainable AI have shown that CNNs are capable of learning hierarchical relationships between semantic categories in the form of taxonomic classifications. However, accurate evaluation of this ability is an open challenge, of which two aspects are non-trivial: constructing symbolic representations of semantic content after training, and quantification of its adequacy with respect to the semantics of the application domain. Existing evaluation methods are typically restricted to standard CNN performance metrics and do not take into account the underlying decision-making process in terms of explicit semantic structure in the domain. In this paper we propose a taxonomy extraction method for supervised CNN classifiers to capture how symbolic class concepts and their hypernyms from a given domain are hierarchically organised in the model's subsymbolic representation. In addition, we propose a taxonomy ground truth comparison method to evaluate the "semantic adequacy" of the extracted hierarchy of class concepts. Our approach is tested using VGG-16, ResNet-18, ResNet-152 and trained on CIFAR-100 and ImageNet. Results show the influence of the dataset quality and architecture depth on semantic adequacy, as suggested by recent literature [1]. We also observe that existing techniques for injecting external knowledge to the models during the training phase may lead to better taxonomies. This suggests that the hierarchical-aware models may have a semantic advantage over their respective original architectures. Finally, we provide a fine grained approach for analysing CNN interpretability in terms of its semantic content.

Keywords

Explainable AI, Taxonomy Extraction, Convolutional Neural Networks

1. Introduction

Since their inception in 1989 [2], Convolutional Neural Networks (CNNs) are still widely used and considered to be the state of the art in computer vision tasks. Each year, more sophisticated and accurate CNN architectures are developed by researchers in both academia and industry [3] and more real-world applications are making use of them [4].

The adoption of Convolutional Neural Networks (CNNs) in critical domains such as in healthcare [5, 6] is yet to reach its full potential, mainly due to limitations in their domain-specific semantic interpretability. Such limitations draw attention to the tradeoff between performance accuracy and the ability of a model to provide motivations for its decisions.

In the field of Explainable AI (XAI), the issue of evaluating and comparing deep learning models beyond their accuracy over a test set has been tackled by methods that provide global interpretations.

[*] Corresponding author.

[†] These authors contributed equally.

Examples include detection of concept importance [7, 8] and class hierarchy visualisation [9]. While providing some form of explanation, methods like these can not be sufficient for an objective comparison between different models, since the resulting interpretations do not provide a metric that compares CNN decision making with some external semantic ground truth.

In this work, we provide methods for semantic interpretability of CNNs, based on how hierarchical relationships between semantic concepts are captured by the model's internal representation. To this aim, we propose a taxonomy extraction method to derive a domain taxonomy from a trained CNN. For example, for a model trained over the ImageNet dataset [10], our method is capable of extracting taxonomic axioms such as ($GermanShepherd \subset Dog \subset Mammal \subset Vertebrate \subset Organism \subset Entity$).

By comparing such taxonomic axioms with a relevant ground truth class hierarchy (e.g., WordNet for ImageNet) we can evaluate quantitatively how faithful the extracted taxonomy is to the selected groundtruth. In the current context this metric is referred to as *semantic adequacy*. Note that, independent of model accuracy, a higher semantic adequacy score implies better transparency by allowing interpretation of model behaviour in terms of user-accepted concepts. Moreover, it makes objective semantic comparison between models possible, provided that their training data is drawn from classes in similar or related taxonomies.

In order to achieve these goals, we build upon the idea of co-activation graph [11], a data structure based on correlation coefficients of neuron activation values. Semantic notions are introduced by relating CNN output neurons to nodes in the graph. In this work, we combine co-activation graphs with a taxonomy extraction method originally designed for knowledge graphs by [12]. An overview of the approach can be seen in Figure 1. Finally, the resulting taxonomy can be compared to a ground truth taxonomy to measure the semantic adequacy of the CNN classifier. The main contributions of the paper are:

- A vector representation for semantic relationships of output classes, obtained by co-activation analysis of a trained model. Such class embeddings are general purpose representations. We use them for taxonomy extraction by means of hierarchical clustering.
- A method for taxonomy extraction from a trained model. The resulting taxonomies provide symbolic explanations for sub-symbolic decision making.
- An evaluation metric for comparison between extracted and ground truth taxonomies, measuring what we call the semantic adequacy of a model. In turn, this metric allows us to compare different models in semantic terms, instead of merely their performance metrics.

Experimental results indicate that the CNNs used can learn direct and transitive subclass relationships reasonably well, especially considering that they were not explicitly trained for that and did not have any a priori information about semantic hierarchies. An encouraging finding from our results is that hierarchy-aware ResNets achieved the best results for both datasets. This support the intuition that if we inject external knowledge into a neural network during the training process, we can make the model more interpretable by making such knowledge explicit in relation to the network's decision process. Our experiments also show that the taxonomy extraction and the idea of semantic adequacy can enhance global interpretability of deep models in terms of the semantics pertaining to class hierarchy encoded in their internal representations.

The rest of this paper is organised as follows. Section 2 presents related work for the topics of global interpretation of CNNs and taxonomy extraction. Section 3 elaborates on the methodology for extracting taxonomies from co-activation graphs built for CNNs. In Section 4 an experiment using CNNs trained for CIFAR-100 and ImageNet is conducted. In Section 5 we present our conclusions and discuss the obtained results.

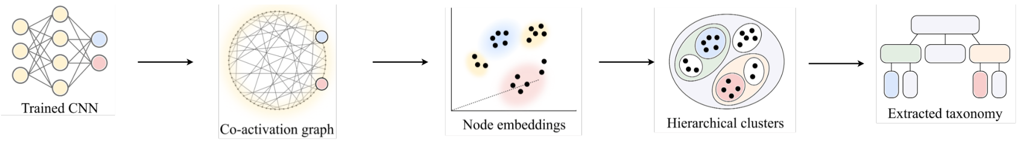

Figure 1: Taxonomy extraction. After optimizing for class discrimination without hierarchical structure, we extract a taxonomy by means of co-activation analysis and hierarchical clustering. This enables a comparison between extracted taxonomy and original semantic class relationships.

2. Related work

2.1. Semantic Global Interpretation of CNNs

In XAI, global interpretations aim to provide insights into the behaviour of a model as a whole, instead of explaining individual inference events. E.g., TCAV [7] identifies the most important concepts for predicting each output class in a classification problem. TCAV is able to interpret classification outcomes in terms of semantic concepts but fails to work well in treating correlations between concepts, often present in real-world image datasets. This issue is addressed by [13], with a method to detect cause-effect relations between concepts and the model predictions. Then, [14] proposes a method to discover concepts and measure concept importance for predictions. Interpreting cause-effect relationships between concepts and output classes in this way is useful, but requires each concept to be measured individually. As a result it does not easily allow for a direct and quantitative comparison between models.

Using a different approach, [9] explores how classes are hierarchically organised by CNNs. A visual approach to discover hierarchical relationships between classes is combined with an hierarchical-aware CNN architecture. This work is closely related to ours in the way we both explore hierarchical structures. However, while their method requires visual and interactive analysis, our approach extracts the hierarchical relationships automatically from the model. This is an important prerequisite for the assessment of the semantic adequacy: without automatic extraction of hierarchical information, the scale of modern CNNs makes semantic global interpretation infeasible. In addition, while [9] relies only on the confusion matrix, our taxonomy extraction explores the internal representations of a model. This makes our approach suitable for interpreting separate layers in the model and understanding the role of such layers in the overall decision making process.

2.2. Taxonomy extraction from graph structures

In complex domains, constructing taxonomies by hand is an expensive task. Some approaches have looked into taxonomy extraction from non-textual, structured and semi-structured data, such as knowledge graphs. Authors in [12, 15] propose an unsupervised method to find hypernym axioms. Vector embeddings for each node in the knowledge graph are calculated, resulting in a class centroid and radius in the latent space. Then, based on the distance between the centroids, they form axioms and construct their transitive closures to build a taxonomy.

The main drawback of the approach in [15] is that concepts need to be directly represented as nodes in the knowledge graph. The method proposed in [12] can help overcome this issue. Based on node embeddings, their approach uses hierarchical clustering over the latent space to find a hierarchical structure. By mapping concepts (or types) to cluster-nodes in this structure, a typed tree is formed from which to construct a taxonomy. The advantage is that the clustering phase does not use any information regarding the types. This makes it more suitable in our setup, where these types are typically not represented by nodes in the graph. In this work, we use co-activation graphs as a graph representation for CNNs and adapt the method in [12] to this graph for the taxonomy extraction. In the next section, the methodology to combine the two is presented in detail.

2.3. Hierarchy-aware architectures

Recent works have shown that CNNs can learn the underlying hierarchical structure between classes during the training phase even though they were not explicitly trained for this specific task. Motivated by this observation, authors in [9, 16] have shown how a CNN architecture can be modified in order to help the model in the learning of such hierarchical structure during the training phase.

In their work, the authors in [9] show how a modified Alex-Net architecture [17] can improve accuracy and accelerate the process of learning class hierarchies. Their method works by adding extra classification branches between some of the convolutional layers from the original architecture.

On other hand, the method proposed by [16] adds extra classification layers after the final classification layer of the original architecture. We consider the method from [9] more suitable for a benchmark on semantic adequacy because the hierarchical structure is learned directly from the convolutional layers and not from the final classification layer.

Following [9] and by adapting their method for ResNets, a hierarchy-aware ResNet can be constructed as follows: Given a hierarchy of depth n (root excluded), we add $n - 1$ branches in the architecture that learn group level classifications and optimise for error at each level of the class hierarchy. The branches are each composed of two fully-connected layers, and are evenly spread along the residual blocks. These additions are sufficient for our current experiments. We leave further architectural optimisations (e.g., dimension of extra modules) to future work. One hypothesis that can emerge from this is that hierarchy-aware CNNs may lead to better taxonomies than their corresponding original architectures, since they were explicitly trained for learning the hierarchical relationships between classes. To test this hypothesis, in Section 4 we compare the semantic adequacy from hierarchy-aware CNNs constructed following [9] against their corresponding original architecture.

3. Methodology

We assume class concepts are organised in a taxonomy, and that a neural network was optimised only to discriminate between leaf node classes. Our goal is to reconstruct, into symbolic form, the full taxonomy from the internal sub-symbolic structure of the model. To this end, we introduce a novel extraction method, designed such that the extracted taxonomy reflects how the model organises output classes and their hypernyms hierarchically in its internal representation.

We use co-activation graphs [11] as an intermediate representation to correlate neuron activities with classes in a trained model. Embedding graph nodes as vectors results in a latent representation of semantic relationships between classes as learned by the model. We then build on the taxonomy extraction method of [12] and hierarchical clustering to transform latent representations back into a semantic structure. An overview of the procedure is shown in Figure 1. The remainder of this section is devoted to describing each step in further detail.

3.1. From deep representations to co-activation graph

Given a trained neural network model, the nodes of its co-activation graph stand in one-to-one correspondence with its neurons. Each pair of nodes in the graph is connected by a weighted edge determined by the Spearman correlation coefficients between neuron activation values on a test data set. For neurons in dense layers this process is straightforward: there is a single activation value per data sample. For neurons in convolutional layers, a single activation value is obtained by applying average pooling on the feature map.

The resulting graph provides relevant information on how dependencies between internal representations impact the global behaviour of a classifier. This makes co-activation graphs a suitable intermediate representation to analyse how a model transforms sub-symbolic information from pixels, via its internal representations, into an assignment to a semantic class. Next, we will discuss how to transform the statistical information in the co-activation graph into a semantic structure.

3.2. From co-activation graph to taxonomy

We modify the extraction method in [12] such that it applies to co-activation graphs. Three phases can be distinguished: embedding of graph nodes into vector representation, followed by agglomerative clustering, and finally assignment of semantic types to clusters. We now discuss further details of the method. For the embedding of nodes from the co-activation graph to vector representations we make use of existing embedding functions [18, 19]. The results from the first phase is a data set D containing a vector for each node in the co-activation graph.

Agglomerative clustering starts by creating a leaf cluster for every vector in D. At each iteration, the two closest clusters are merged according to some chosen distance metric. Clustering terminates when there is a single cluster containing every vector in D, resulting in a tree structure over the vectors. Assigning semantic types to the tree will turn it into a taxonomy.

In order to assign types to clusters, the method calculates the F-score $F(C, t)$ for each cluster C and type t, which indicates how well C represents the entities in t. The F-score can be calculated as shown in Equation 1, where $N_{C,t}$ is the number of entities with type t in cluster C, N_C is the number of entities in C and N_t is the number of entities with type t. A high F-score(C,t) indicates that cluster C contains mostly entities of type t and the highest number of entities of type t is contained in C.

$$F(C, t) = 2 \cdot \frac{N_{C,t}}{N_C + N_t} \tag{1}$$

We remove clusters that are not associated to a type and evaluate the extracted taxonomy by precision, recall and F-score over edges of the taxonomy (“direct” evaluation) and its transitive closure (“transitive” evaluation).

4. Experimental Analysis

The experimental evaluation has two distinct goals. First, to measure correlations of class similarity in the embedding space with semantic similarity in the ground truth taxonomy. This indicates the ability of learned representations to separate symbolic concepts in the taxonomy. Second, to evaluate the semantic adequacy of the extracted taxonomy with respect to its ground truth. This indicates the degree of transparency allowed by the particular models as expressed in semantic terms, and the suitability of our method for semantic interpretation in general.

4.1. Experiment Setup

Experiments were conducted using CIFAR-100 and ImageNet datasets. For CIFAR-100 we trained VGG-16, ResNet-18 and hierarchy-aware ResNet-18 (noted HA-ResNet-18). For ImageNet we trained

VGG-16, ResNet-152 and hierarchy-aware ResNet-152 (noted HA-ResNet-152).

For each dataset and model, a co-activation graph containing only those connections with correlation higher than 0.3 was built. We extracted the taxonomy using the pipeline in Figure 1. Node embeddings were generated using two different embedding methods in order to check if the results are consistent across different strategies: Node2Vec [19] and Fast Random Projection (FastRP) [18]. For both Node2Vec and FastRP we used the algorithm implemented by Neo4j.

After calculating the node embeddings using the two methods above, the next phase is to apply the agglomerative clustering algorithm. For this phase we have tested different distance criteria and metrics, which influence the merging strategy of the agglomerative clustering. The metrics were euclidean and cosine (when applicable) while the distance criteria used were: *average (UPGMA)*, *weighted (WPGMA)*, *complete*, *centroid* and *ward*.

At this point, we end up with a hierarchical clustering tree, and the next phase is to assign types to the clusters that best represent the entities from each type. Because we want to compare the extracted taxonomy with the WordNet hierarchy, we extracted the types directly from WordNet using the *nltk* python package. Then, for each type t and each cluster C, we have calculated the $Fscore(C, t)$ using Equation 1 and assigned types to clusters by solving the corresponding Linear Sum Assignment problem. Finally, the clusters and types that are not associated are removed, and the resulting tree hierarchy represents the extracted taxonomy.

4.2. Correlating class embedding with class similarity

The goal of this analysis is to check whether the produced node embeddings can encode semantic similarity between classes represented in that space. For this analysis, we have first calculated the semantic similarity for every pair of classes in the dataset. This process was done by using the path similarity metric available from the nltk package, which calculates how similar two concepts are based on the shortest path that connects them on the WordNet hierarchy. We have then calculated the spearman correlation between semantic similarity among classes and cosine similarity between the corresponding classes in the embedding space.

Dataset	Model	Correlation (Node2Vec)	Correlation (FastRP)
CIFAR100	VGG16	**0.13**	0.09
CIFAR100	ResNet18	**0.36**	0.27
CIFAR100	HAResNet18	**0.32**	0.30
ImageNet	VGG16	0.44	**0.45**
ImageNet	ResNet152	0.43	**0.54**
ImageNet	HAResNet152	0.48	**0.61**

Table 1
Benchmark of each node embeddings method using the spearman correlation between pairwise class semantic similarity and their cosine similarity in the embedding space.

In Table 1 it is possible to observe that there is a positive correlation between semantic similarity and cosine similarity among classes in the embedding space. This is a first indication that the CNNs have learned semantic relationships between classes, which supports the idea that it may be possible to reconstruct the taxonomical relationships between them. The HAResNet152 architecture trained on ImageNet achieved the highest correlation, which gives a first evidence that the additional information injected during the training phase helped this model to better capture the semantic relationships between classes. It can also be noted that the results from Node2Vec are higher than FastRP for CIFAR-100 while FastRP performs better for ImageNet, which indicates that the choice of the embedding method may be incluenced by the density of the graph. However, the correlation only is not a strong metric to compare the semantic adequacy between each architecture since they do not expose which semantic relationships were learned by the model. The semantic adequacy comparison between models is done in a more detailed way in the second analysis.

4.3. Evaluating extracted taxonomies

In this second analysis we evaluate the semantic adequacy of the taxonomies extracted using our method by comparing them with the ground truth hierarchy extracted from WordNet. This evaluation is conducted following the same principles used in [12], which uses both a direct and a transitive evaluation. For example, an axiom such as *GermanShepherd* $\sqsubset$ *Dog* is going to cause a negative effect in the direct evaluation, because, according to WordNet, the direct hypernym for a *GermanShepherd* is *ShepherdDog*. In the transitive evaluation, the goal is to evaluate high level axioms. Using the previous example, *GermanShepherd* $\sqsubset$ *Dog* causes a positive effect in the transitive evaluation because there is a transitive relationship between types *GermanShepherd* and *Dog*, also according to WordNet.

For this analysis, we also evaluated the semantic adequacy of an untrained model using the VGG-16 architecture initialised using random parameters, with the purpose of providing a lower bound baseline. The precision, recall and F-score are calculated using the edges of the graphs representing the extracted taxonomies and the edges of the graph representing the ground truth, as described in Equation 2. For the direct evaluation we calculate the evaluation metrics based directly on the respective graphs whereas for the transitive evaluation we consider the transitive closure.

$$\text{Ground truth}: G_t = (V, E_t),\ \text{Experimental}: G_e = (V, E_e)$$
$$\begin{cases} precision = \dfrac{|E_t \cap E_e|}{|E_e|} \\ recall = \dfrac{|E_t \cap E_e|}{|E_t|} \\ f\text{-}score = 2 \cdot \dfrac{|E_t \cap E_e|}{|E_t| + |E_e|} \end{cases} \tag{2}$$

Table 2 reports the best F-scores achieved by each model for CIFAR-100 using both FastRP and Node2Vec. The untrained VGG-16 had the lowest semantic adequacy for both direct and transitive evaluations, which was expected since this model was provided as a lower bound baseline. It can be observed that HA-ResNet-18 achieves the highest F-scores for both FastRP and Node2Vec, although ResNet-18 also obtains the best value for the direct evaluation using FastRP. VGG-16 gets the lowest results from all the three models.

From Table 3 we can observe a similar behaviour when considering the setup for the ImageNet dataset. Again, the untrained VGG-16 achieved a much lower semantic adequacy for both direct and transitive evaluations, when compared to the trained models. The HA-ResNet variation had the best scores for both direct and transitive evaluations with VGG-16 getting the lowest values. We can also note how the semantic correlation from Table 1 influences the semantic adequacy, since most of the setups that obtained the highest semantic correlations also led to the best semantic adequacies.

The analyses obtained from Table 2 and Table 3 consider only the best result achieved by each model. In order to analyse which architectures are the most semantic adequate in a consistent way we have performed a statistical test. For this, we first obtained all the F-scores for each model. The F-scores were obtained by varying the embedding methods, clustering criterion and distance metrics. The setups that produced F-score values lower than twice the standard deviation from the median were considered as outliers and thus removed from this analysis. Then, for each pair of models we tested if there were a difference in their F-score distribution using the student t-test. The null-hypothesis is that there is no difference between two distributions and p-value ≤ 0.05 rejects the null hypothesis and indicates there is a statistical difference between the two distributions.

Table 2
Direct and transitive F-scores for CIFAR-100.

Best F-scores for CIFAR100				
	FastRP		Node2Vec	
Model	Transitive	Direct	Transitive	Direct
UntrainedVGG16	0.13	0.23	0.13	0.24
VGG16	0.28	0.40	0.24	0.37
ResNet18	0.34	0.44	0.38	0.50
HAResNet18	0.36	0.44	0.40	**0.55**

Table 3
Direct and transitive F-scores for ImageNet.

Best F-scores for ImageNet				
	FastRP		Node2Vec	
Model	Transitive	Direct	Transitive	Direct
UntrainedVGG16	0.02	0.22	0.03	0.22
VGG16	0.41	0.47	0.34	0.44
ResNet152	0.41	0.48	0.40	0.50
HAResNet152	**0.43**	**0.49**	**0.46**	**0.53**

From Table 4 it is possible to observe that, for the transitive evaluation, all HA-ResNet variations are statistically more semantic adequate than pure ResNets and VGG-16. This is an encouraging evidence that the injection of external knowledge during the training phase may help achieving more interpretable models. We can also see that VGG-16 does not perform better than any other model, which may indicate that the architecture depth can have an effect on semantic adequacy.

Dataset	Model one	Model two	direct pvalue	transitive pvalue
CIFAR100	ResNet18	VGG16	**<0.05**	**<0.05**
CIFAR100	HAResNet18	VGG16	**<0.05**	**<0.05**
CIFAR100	HAResNet18	ResNet-18	>0.05	**<0.05**
ImageNet	ResNet152	VGG16	>0.05	>0.05
ImageNet	HAResNet152	VGG16	>0.05	**<0.05**
ImageNet	HAResNet152	ResNet152	**<0.05**	**<0.05**

Table 4
Statistical comparison on the semantic adequacy between different architectures.

Overall, our evaluation shows that the taxonomies extracted from CNNs using our method can achieve reasonable direct and transitive F-scores, even when the models were not trained explicitly for that. The best results in our analysis were generated by the hierarchical-aware architectures, as proposed by [9] and adapted for ResNets in this paper.

5. Conclusion

We proposed a method for semantic interpretability of deep representations by extracting taxonomies from the internal structure of trained CNNs. Our approach represents a CNN as a co-activation graph and adapts the taxonomy extraction method in [12] to such graph. We then introduce the concept of semantic adequacy to measure how well a model captures the hierarchical relationship between classes from a given domain by comparing its extracted taxonomy to a ground truth such as WordNet.

The proposed taxonomy extraction method and semantic adequacy together can help in comparing and choosing among different CNNs by exposing how well each model learned the semantic relationships from a given dataset instead of relying purely on performance metrics. The next steps include adapting the semantic adequacy in order to provide more fine-grained information, since the value is currently associated to the extracted taxonomy as a whole. In this case we expect the metric to inform which specific parts of the taxonomy are more or less adequate according to the ground truth. For example, the extracted taxonomy may be more adequate for a specific subtree (e.g. *dogs*) but less adequate for another (e.g. *primates*). This information can be useful not only for deciding when to trust a given model but also for transfer learning, where a model may not be suitable for a given task even though it may provide high accuracy.

Acknowledgements

This publication has emanated from research supported by Science Foundation Ireland (SFI) Grant Number SFI/12/RC/2289_P2, co-funded by the European Regional Development Fund.

References

[1] D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network dissection: Quantifying interpretability of deep visual representations, 2017. URL: https://arxiv.org/abs/1704.05796. doi:10.48550/ARXIV.1704.05796.

[2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation 1 (1989) 541–551. URL: https://doi.org/10.1162/neco.1989.1.4.541. doi:10.1162/neco.1989.1.4.541. arXiv:https://direct.mit.edu/neco/article-pdf/1/4/541/811941/neco.1989.1.4.541.pdf.

[3] A. Khan, A. Sohail, U. Zahoora, A. S. Qureshi, A survey of the recent architectures of deep convolutional neural networks, Artificial Intelligence Review 53 (2020) 5455–5516. URL: https://doi.org/10.1007/s10462-020-09825-6. doi:10.1007/s10462-020-09825-6.

[4] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, L. Farhan, Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, Journal of Big Data 8 (2021). URL: https://doi.org/10.1186/s40537-021-00444-8. doi:10.1186/s40537-021-00444-8.

[5] R. Zeleznik, B. Foldyna, P. Eslami, J. Weiss, I. Alexander, J. Taron, C. Parmar, R. M. Alvi, D. Banerji, M. Uno, Y. Kikuchi, J. Karady, L. Zhang, J.-E. Scholtz, T. Mayrhofer, A. Lyass, T. F. Mahoney, J. M. Massaro, R. S. Vasan, P. S. Douglas, U. Hoffmann, M. T. Lu, H. J. W. L. Aerts, Deep convolutional neural networks to predict cardiovascular risk from computed tomography, Nature Communications 12 (2021). URL: https://doi.org/10.1038/s41467-021-20966-2. doi:10.1038/s41467-021-20966-2.

[6] R. Shadmi, V. Mazo, O. Bregman-Amitai, E. Elnekave, Fully-convolutional deep-learning based system for coronary calcium score prediction from non-contrast chest ct, in: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), 2018, pp. 24–28. doi:10.1109/ISBI.2018.8363515.

[7] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, R. Sayres, Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav) (2017). URL: https://arxiv.org/abs/1711.11279. doi:10.48550/ARXIV.1711.11279.

[8] A. Ghorbani, J. Wexler, J. Y. Zou, B. Kim, Towards automatic concept-based explanations, in: Advances in Neural Information Processing Systems, 2019, pp. 9273–9282.

[9] A. Bilal, A. Jourabloo, M. Ye, X. Liu, L. Ren, Do convolutional neural networks learn class hierarchy?, IEEE Transactions on Visualization and Computer Graphics 24 (2018) 152–162. URL: https://doi.org/10.1109%2Ftvcg.2017.2744683. doi:10.1109/tvcg.2017.2744683.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[11] V. A. Horta, I. Tiddi, S. Little, A. Mileo, Extracting knowledge from deep neural networks through graph analysis, Future Generation Computer Systems 120 (2021) 109–118. URL: https://www.sciencedirect.com/science/article/pii/S0167739X21000613. doi:https://doi.org/10.1016/j.future.2021.02.009.

[12] F. Martel, A. Zouaq, Taxonomy extraction using knowledge graph embeddings and hierarchical clustering, in: Proceedings of the 36th Annual ACM Symposium on Applied Computing, SAC '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 836–844. URL: https://doi.org/10.1145/3412841.3441959. doi:10.1145/3412841.3441959.

[13] Y. Goyal, A. Feder, U. Shalit, B. Kim, Explaining classifiers with causal concept effect (cace), 2019. URL: https://arxiv.org/abs/1907.07165. doi:10.48550/ARXIV.1907.07165.

[14] C.-K. Yeh, B. Kim, S. O. Arik, C.-L. Li, T. Pfister, P. Ravikumar, On completeness-aware concept-based explanations in deep neural networks, 2019. URL: https://arxiv.org/abs/1910.07969. doi:10.48550/ARXIV.1910.07969.

[15] P. Ristoski, S. Faralli, S. P. Ponzetto, H. Paulheim, Large-scale taxonomy induction using entity and word embeddings, in: Proceedings of the International Conference on Web Intelligence, WI '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 81–87. URL: https://doi.org/10.1145/3106426.3106465. doi:10.1145/3106426.3106465.

[16] R. L. Grassa, I. Gallo, N. Landro, Learn class hierarchy using convolutional neural networks, CoRR abs/2005.08622 (2020). URL: https://arxiv.org/abs/2005.08622. arXiv:2005.08622.

[17] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, Curran Associates Inc., Red Hook, NY, USA, 2012, p. 1097–1105.

[18] H. Chen, S. F. Sultan, Y. Tian, M. Chen, S. Skiena, Fast and accurate network embeddings via very sparse random projection, 2019. URL: https://arxiv.org/abs/1908.11512. doi:10.48550/ARXIV.1908.11512.

[19] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, 2016. URL: https://arxiv.org/abs/1607.00653. doi:10.48550/ARXIV.1607.00653.