Automating CNN Neuron Interpretation using Concept Induction

Samatha Ereshi Akkamahadevi[1,*], Abhilekha Dalal[1] and Pascal Hitzler[1,*]

[1]*Department of Computer Science, Kansas State University, Manhattan, KS, USA*

Abstract

This paper presents an automation pipeline for interpreting hidden neuron activations in Convolutional Neural Networks (CNNs), a crucial objective of Explainable AI (XAI). Previously, our research group addressed this objective by employing concept induction and semantic reasoning using a concept hierarchy derived from the Wikipedia knowledge graph. However, the process was executed manually, taking several days to complete. In this study, we have fully automated the workflow, achieving consistent results while significantly reducing the execution time. The automation pipeline streamlines model training, data preparation, concept induction, image retrieval, classification, and statistical validation, thereby completely eliminating the manual intervention. This automation enables us to efficiently interpret and validate CNN neuron activations by modifying parameters, such as incorporating a broader range of training images and classes and examining additional concept induction results across various neuron layers using different analytical tools.

Keywords

Explainable Artificial Intelligence, Deep Learning, Knowledge Graph, Semantic Web, Automation in AI

Demo video: https://youtu.be/a_tHVwexlEE, **Github:** https://bit.ly/ExAI_Automation_DaSe

1. Introduction and Related Work

Deep learning has revolutionized the field of artificial intelligence (AI), achieving breakthroughs in fields such as image recognition, speech recognition, drug discovery, robotics etc. [1]. However, its "black box" nature poses challenges, especially in critical domains needing transparency and explainability [2]. Explainable AI steps in to address these issues, striving to make AI systems more interpretable and their decision-making processes more transparent [3]. Previously, Dalal et al. has demonstrated that hidden neuron activations in CNNs could be meaningfully interpreted using structured background knowledge and ontology reasoning [4, 5, 6]. This approach utilized a large-scale knowledge base derived from Wikipedia's concept hierarchy [7] and employed concept induction [8, 9] to generate interpretable class labels for hidden neurons. Building on this foundation, the current study automates the entire interpretability process to enhance efficiency and ensure reproducibility, eliminating the need for human intervention. We optimized resource allocation and implemented parallel processing to significantly reduce the execution time. This paper provides a detailed description of our automated approach, its technical components, performance evaluation results, and broader implications for XAI.

Posters, Demos, and Industry Tracks at ISWC 2024, November 13–15, 2024, Baltimore, USA

*Corresponding author.

samatha94@ksu.edu (S. E. Akkamahadevi); adalal@ksu.edu (A. Dalal); hitzler@ksu.edu (P. Hitzler)

0009-0001-6333-8004 (S. E. Akkamahadevi); 0000-0002-7047-5074 (A. Dalal); 0000-0001-6192-3472 (P. Hitzler)

2. System Architecture and Implementation

Our system uses automation in four stages (Figure 1) below to streamline processes and to enhance efficiency.

2.1. Stage 1: Model Training and Data Configuration

Initially, our automation pipeline trains and configures a CNN model using the ADE20K dataset [10]. This process is executed on Beocat [11], a high-performance computing environment optimized for managing extensive datasets. A Bash script automates job scheduling, resource allocation via SLURM, initializes the Python environment, securely clones the stage 1 repository from GitHub, and installs the necessary dependencies to establish the training environment. We employ a ResNet50V2 architecture implemented in TensorFlow, fine-tuned to enhance model performance using techniques such as data augmentation, early stopping, and batch normalization. Our model is trained on 6,187 images, using Adam optimization algorithm (learning rate 0.001) and categorical cross-entropy as the loss function. Post-training, the model is saved and used to analyze activations within the dense layer across 1,370 ADE20K images and it generates positive and negative example sets based on activation thresholds. P consists of images activating a neuron above 80% of maximum activation, while N includes images activating below 20%. These sets are annotated with classes from background knowledge and will generate configuration files for each neuron which are pivotal for the Concept Induction analysis in Stage 2, providing structured input data for generating and validating label hypotheses.

2.2. Stage 2: Parallelized Concept Induction and Label Hypothesis Generation

We used the concept induction process to generate label hypotheses for each of the 64 neuron activations in the CNN's dense layer using the heuristic Concept Induction system ECII [9]. We automated the simultaneous execution of tasks for all 64 neurons by employing parallel processing with a SLURM-configured Bash script in Beocat. The script initializes the environment, installs necessary Java and Maven dependencies, and clones the latest stage 2 repository from GitHub. Each neuron-specific configuration file from Stage 1 was used to generate semantic concepts, producing output concept files with hypothesized labels and coverage scores using a background knowledge base from the Wikipedia concept hierarchy.

2.3. Stage 3: Parallelized Image Retrieval and Classification

Image retrieval and classification were automated for all neurons to validate the label hypotheses generated in Stage 2. A Bash script manages parallel task execution using SLURM, generating indices for neurons with configuration files. It clones the Stage 3 project repository, sets up the environment, installs dependencies. The script runs a Python program that utilizes the pygoogle_image library to extract labels from the top 3 solutions for each neuron, retrieves 100 images per label from Google, and classifies them using the trained CNN model. Retrieved images are divided into evaluation and verification sets for statistical analysis.

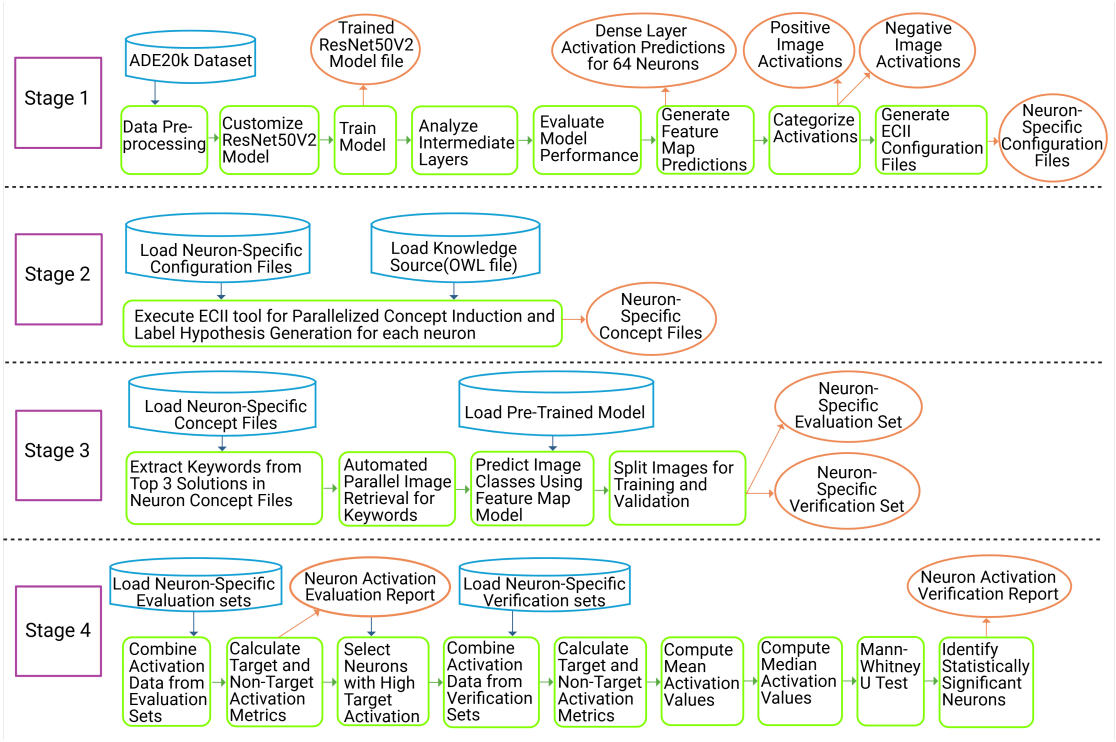

Figure 1: Automated four-stage pipeline for analyzing neuron activations, inducing concepts, and evaluating neuron significance using a ResNet50V2 model and ECII tool, created with BioRender.com.

2.4. Stage 4: Statistical Analysis and Verification of Neuron Activations

Label hypotheses are validated through statistical analysis of neuron activations. A Bash script sets up the environment, clones the stage 4 repository, and installs dependencies. The script runs a Python program that combines activation data from evaluation and verification sets, generates summary statistics, and conducts a Mann-Whitney U test [12] to compare activation values for target and non-target images. Evaluation sets, containing images that strongly activate neurons, provide initial activation metrics. Verification sets undergo further statistical testing to confirm the accuracy and robustness of the label hypotheses.

3. Results and Conclusion

The automation pipeline, executed to enhance the interpretation of hidden neuron activations in CNNs, achieved significant performance improvements.

In stage 1, it eliminated the need for manual analysis to identify and categorize the positive and negative images from the model output. It also generated neuron-specific configuration files with embedded ontology references in OWL format, which serve as input for the subsequent concept induction analysis, completing the execution under 40 minutes.

In stage 2, parallel execution of ECII tool for all 64 neurons reduced the concept induction

execution time from over 10 hours to 20 minutes. The ECII tool processed neuron-specific configuration files to generate output concept files with hypothesized labels, sorted by coverage scores, along with precision, recall, and f-measure metrics.

In Stage 3, the image retrieval and classification processes were automated to run concurrently for all neurons to validate label hypotheses from Stage 2. It extracted labels from ECII output, retrieved relevant images from the internet, and classified them using our trained CNN model from stage 1. Model generated the evaluation sets to include activations from images that activate the neuron, providing initial insights into neuron activation patterns while verification sets were generated for detailed statistical analysis in the next stage. This parallelized approach reduced processing time to about 10 minutes, compared to 16 hours without parallelization.

Finally, Stage 4 performed statistical analysis and validated the results in just 3 minutes. The system analyzed activation data, generated a detailed summary of statistics, and verified the label hypotheses. The statistical analysis showed that concept induction analysis with structured background knowledge yields meaningful labels that consistently explain neuron activation. The Mann-Whitney U test rejected the null hypothesis ($p < 0.05$), confirming significant differences in activation values between target and non-target images.

Overall, the entire pipeline was completed in approximately 1 hour 15 minutes, demonstrating substantial improvements in performance, indeed the explainability of the CNN model. The automation and parallelization strategies drastically reduced execution times, minimized manual effort, and ensured consistent and reproducible results, demonstrating the robustness and efficiency of our approach.

4. Future work

We will expand and diversify the dataset, explore various neural network architectures, and integrate various analytical tools. Additionally, we aim to enhance model interpretability by examining additional concept induction results across various neuron layers.

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444. doi:10.1038/nature14539.

[2] A. Adadi, M. Berrada, Peeking inside the black-box: a survey on explainable artificial intelligence (XAI), IEEE access 6 (2018) 52138–52160. doi:10.1109/ACCESS.2018.2870052.

[3] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, G.-Z. Yang, Xai—explainable artificial intelligence, Science robotics 4 (2019) eaay7120. doi:10.1126/scirobotics.aay7120.

[4] A. Dalal, M. K. Sarker, A. Barua, E. Vasserman, P. Hitzler, Understanding CNN hidden neuron activations using structured background knowledge and deductive reasoning, arXiv preprint arXiv:2308.03999 (2023). doi:10.48550/arXiv.2308.03999.

[5] A. Dalal, R. Rayan, A. Barua, E. Y. Vasserman, M. K. Sarker, P. Hitzler, On the value of labeled data and symbolic methods for hidden neuron activation analysis, arXiv preprint arXiv:2404.13567 (2024). doi:10.48550/arXiv.2404.13567.

[6] A. Dalal, R. Rayan, P. Hitzler, Error-margin analysis for hidden neuron activation labels, arXiv preprint arXiv:2405.09580 (2024). doi:10.48550/arXiv.2405.09580.
[7] B. Villazón-Terrazas, F. Ortiz-Rodríguez, S. M. Tiwari, S. K. Shandilya (Eds.), Wikipedia knowledge graph for explainable AI, 2020. doi:10.1007/978-3-030-65384-2_6.
[8] J. Lehmann, P. Hitzler, Concept learning in description logics using refinement operators, Machine Learning 78 (2010) 203–250. doi:10.1007/s10994-009-5146-2.
[9] M. Kamruzzaman Sarker, P. Hitzler, Efficient concept induction for description logics, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019) 3036–3043. doi:10.1609/aaai.v33i01.33013036.
[10] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, A. Torralba, Semantic understanding of scenes through the ade20k dataset, International Journal of Computer Vision 127 (2019) 302–321. doi:10.1007/s11263-018-1140-0.
[11] K. Hutson, D. Andresen, A. Tygart, D. Turner, Managing a heterogeneous cluster, in: Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning), Association for Computing Machinery, New York, NY, USA, 2019, pp. 1–6. doi:10.1145/3332186.3332251.
[12] P. E. McKnight, J. Najab, Mann-whitney u test, The Corsini encyclopedia of psychology (2010) 1–1. doi:10.1002/9780470479216.corpsy0524.