Neurosymbolic Hidden Neuron Analysis in Convolutional Neural Networks

Abhilekha Dalal[a], Moumita Sen Sarma[a], Avishek Das[a], Samatha E. Akkamahadevi[a], Eugene Y. Vasserman[a], Pascal Hitzler[a]

[a]*Department of Computer Science, Kansas State University, Kansas, USA*

Abstract

This tutorial introduces a step-by-step, deductive pipeline for making the inner workings of neural networks more transparent by assigning human-understandable concepts to hidden neuron activations. The approach automatically maps neuron behavior to symbolic concepts drawn from structured knowledge sources and attaches an error margin to each label, providing a measure of confidence in its precision. While demonstrated in detail on the ADE20k scene dataset—including single-concept neurons, multiple neurons contributing to the same concept, and multi-concept neurons—the method is also applied to the SUN2012 dataset and adapted for a text classification task, highlighting its generalizability across modalities. The chapter is designed to be practical and educational, focusing on a replicable methodology that readers can adapt to varied applications. Through worked examples, visualizations, and evaluation strategies, the tutorial offers a clear, reusable framework for concept-based neuron analysis in both vision and language models.

Keywords: Explainable AI, Hidden Neuron Activations, Concept Induction, Knowledge Graphs, Cross-Modal Interpretability, Neurosymbolic Methods

1. Introduction

Deep Neural Networks (DNNs) have achieved remarkable success in domains such as natural language processing, computer vision, bioinformatics, and recommender systems. Yet, they remain largely opaque due to the "black box" nature of their internal processes. Hidden neurons encode intricate and abstract features that are difficult to interpret, limiting transparency, trust, and systematic knowledge transfer (Saranya and Subhashini, 2023). This lack of interpretability is especially critical, particularly in high-stakes areas like healthcare, finance, autonomous systems, and law, where fairness, reliability, and auditability are essential (Angelov et al., 2021).

Explainable AI (XAI) has emerged to address these challenges by improving transparency, accountability, and bias detection (Wang et al., 2021). A wide range of post-hoc methods exists, from feature-based attribution e.g., SHAP: SHapley Additive exPlanations (Lundberg and Lee, 2017) and LIME: Local Interpretable Model-agnostic Explanations (Ribeiro et al., 2016) to pixel-based visualization explanation method like Grad-CAM: Gradient-weighted Class Activation Mapping (Selvaraju et al., 2017) and ReLU: Rectified Linear Units (Nair and Hinton, 2010). While widely adopted, these approaches can oversimplify models, produce unstable or noisy explanations, and often fail to reveal what the network's internal units actually represent (Aysel et al., 2025; Arrieta et al., 2020; Nazir et al., 2025; Dalal et al., 2024). This leaves a practical gap between input-level highlights and mechanistic understanding: saliency tells us where a model looked in a given example, not what its neurons consistently mean or whether those meanings transfer across data, tasks, or architectures. For auditing, safety reviews, and debugging, practitioners need neuron-level explanations that are human-interpretable, stable across examples, and amenable to quantitative checks (e.g., target vs. non-target activations). This motivates concept-level methods that can state "what a neuron detects" in shared, reusable vocabulary, rather than only "where the model looked."

Moreover, completeness is a key requirement in neurosymbolic interpretability, meaning that the symbolic explanation captures all behaviors of the neural model rather than offering only partial or potentially misleading patterns. In this regard, (Wang et al., 2023) introduces a method for extracting sound and complete symbolic rules from Differentiable Rule Mining models by formally ensuring rule–model alignment using counting-aware, multipath Datalog rules.

In contrast, Concept Induction takes a neurosymbolic approach: instead of highlighting pixels or input features, it describes what a neuron detects in terms of high-level, human-readable concepts. This is achieved through symbolic reasoning grounded in description logics and the Web Ontology Language (OWL), allowing neuron activations to be expressed in transparent and logically structured terms (Dalal, 2024; Sarker and Hitzler, 2019a).

To reduce computational overhead, this tutorial uses the heuristic Concept Induction system ECII (Sarker and Hitzler, 2019a). ECII uses a large-scale background knowledge base, such as a Wikipedia-derived ontology with approximately 2 million classes, to systematically generate explanatory concepts without requiring manual selection, thus avoiding biases inherent in predefined explanation categories (Dalal, 2024).

Concept Induction demonstrates its broad applicability across various use cases in explainable AI, highlighting its generalizability. (1) It has been used to analyze CNN hidden neurons by linking them to human-understandable concepts. (2)

It is used for error margin analysis, quantifying how reliably a neuron represents a given concept by measuring target vs. non-target concept activations, thereby improving model interpretability. (3) It has been extended to text-based Long Short-Term Memory (LSTM) networks, demonstrating its ability to interpret hidden layer operations in sequential data processing. Together, these applications show how Concept Induction generalizes across architectures and data types, while remaining grounded in symbolic, human-readable explanations.

1.1. Plan of the Chapter

This chapter is written as a *how-to guide*, walking readers through an end-to-end pipeline for Concept Induction, a neurosymbolic XAI method, and demonstrates its practical use through cross-domain case studies. Beyond the presentation of results, we show the steps and reasoning behind them, so the workflow can be reproduced or adapted in new contexts. Following this guide enables mapping of hidden neuron activations to human-readable, symbolic explanations.

In the rest of this chapter, Section 2 introduces foundational concepts and background, establishing a common terminology for the methodology. Section 3 offers a detailed, step-by-step tutorial on the Concept Induction methodology, guiding through its practical implementation. Section 4 presents case studies demonstrating the application of Concept Induction across various neural network architectures and data modalities. Section 5 analyzes observations from these case studies, evaluates the trade-offs between interpretability and complexity, and explores the method's cross-domain applicability. Section 6 provides practical guidance on the appropriate contexts for applying Concept Induction, selecting suitable evaluation techniques, and integrating it into model debugging workflows for real-world AI systems. Finally, we conclude in Section 7.

2. Background & Preliminaries

This section establishes the foundational concepts needed to follow our neurosymbolic approach. We assume readers have basic familiarity with neural networks (layers, weights, activations, backpropagation) but may need refreshers on specific architectures or knowledge representation concepts. The coverage is intentionally concise—designed as a reference for terminology and key ideas rather than comprehensive instruction. Readers seeking a deeper background should consult the provided references for each topic.

2.1. Neural Networks (Recap)

Artificial Neural Networks (ANNs) process inputs through layers of neurons connected by weighted edges. During training, weights are updated (e.g., via backpropagation and gradient descent) to minimize a loss and improve predictions. We briefly recap two architectures used in this tutorial.

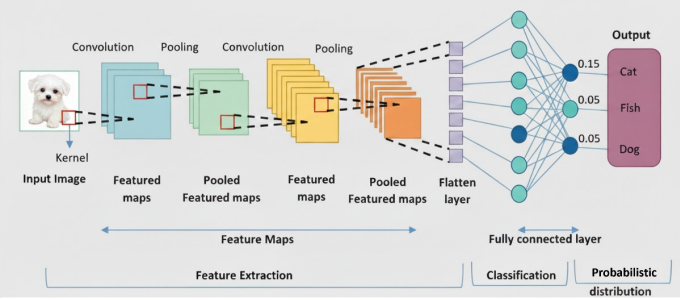

Figure 1: Architecture of a simple convolutional neural network.

2.1.1. Convolutional Neural Networks (CNNs)

CNNs are specialized for visual data (images, video). The key insight is hierarchical feature learning: early layers tend to respond to simple patterns (edges, textures), while deeper layers respond to more abstract visual parts (object fragments, materials) (Zeiler and Fergus, 2014; LeCun et al., 2015). A typical block applies learnable *convolutions* (filtering local patches using shared weights), a pointwise nonlinearity such as ReLU (Nair and Hinton, 2010), and optionally pooling or striding to aggregate spatial information (LeCun et al., 2002); final layers typically flatten the learned spatial features and map the learned features to class scores (Figure 1).

Each convolutional filter produces a *feature map* indicating where in the image that filter "fires." Intuitively, a filter (often called a *neuron*) detects the presence of a specific visual motif across locations (Zeiler and Fergus, 2014). As we go deeper, receptive fields grow, and filters tend to integrate information over broader spatial contexts and capture semantically meaningful patterns (Luo et al., 2016). While this hierarchical representation enables high accuracy, it obscures what individual filters actually encode, contributing to the model's perceived opacity (Lipton, 2018).

Throughout this chapter, we use "*neuron*" to mean a single spatial unit in a feature map and summarize its spatial activations to decide whether an input *activates* that neuron.

2.1.2. Long Short-Term Memory Networks (LSTMs)

LSTMs are specialized for data where order and context matter, addressing the vanishing gradient problem in standard RNNs (Bengio et al., 1994) and enabling effective learning over long sequences. Each LSTM unit maintains a *cell state* regulated by three gates: the *forget gate* removes outdated information, the *input gate* selectively incorporates new information, and the *output gate* controls what cell state information reaches the hidden output (Hochreiter and Schmidhuber, 1997).

For text classification, word embeddings are processed sequentially through LSTM layers; the final hidden state captures a summary representation of the entire sequence, which then feeds into classification layers (Figure 2). Like CNN filters, LSTM hidden states encode complex patterns, but determining what linguistic concepts they represent remains opaque, motivating concept-based analysis approaches. We refer to LSTM "neurons" as the hidden units at each timestep, with activations representing the learned sequential patterns encoded at

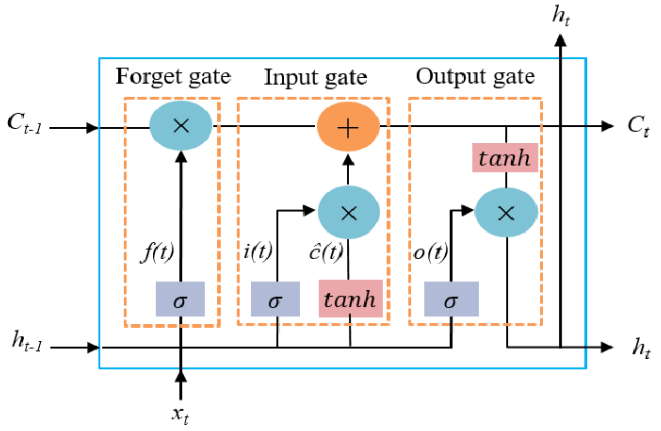

Figure 2: Simple architecture of a Long Short-Term Memory (LSTM) unit (Mohsen et al., 2021).

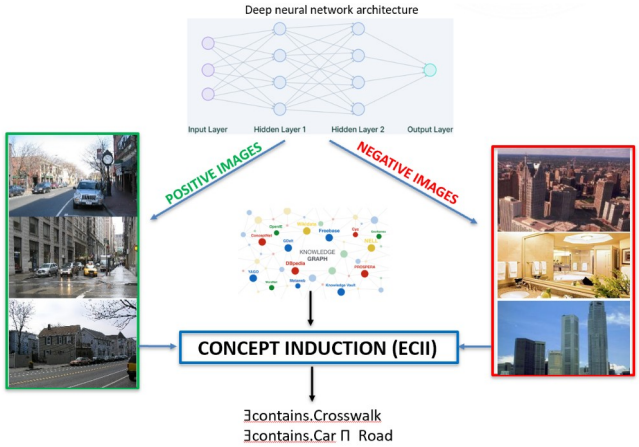

Figure 3: Illustrated Concept Induction process. Positive images (left) and negative images (right) are selected based on neuron activation patterns. A knowledge graph is employed to perform semantic reasoning and generate neuron "labels," which then serve as interpretable concepts.

that point in processing.

Further reading: original LSTM (Hochreiter and Schmidhuber, 1997), applications (Graves, 2012), RNN fundamentals (Goodfellow et al., 2016), and intuitive explanations (Karpathy, 2015).

2.2. Knowledge Graphs

Knowledge Graphs (KGs) provide structured, human-readable representations of knowledge, making them ideal bridges between abstract neuron activations and understandable concepts. KGs are directed, edge-labeled graphs where nodes represent entities and edges represent relationships, typically organized within formal ontological schemas that enable logical reasoning (Hogan et al., 2021).

A Knowledge Graph consists of two main components: the *TBox*, which serves as the schema or ontology layer defining classes, properties, and their relationships, and the *ABox*, which contains the instance-level facts about specific entities. Information is represented as triplets following the structure (subject, predicate, object), such as (street, has, crosswalk). This triplet-based representation enables the storage of explicit facts while also supporting the inference of implicit relationships through reasoning frameworks like RDF, RDFS, and OWL (Berners-Lee et al., 2023).

Vision tasks: We use a large taxonomy derived from Wikipedia's categories and links, organizing millions of concepts from general to specific. This structure lets our algorithms identify the most specific common ancestor concept across activated examples—particularly useful for visual data.

Text tasks: WordNet groups words into synonym "setswords" linked via semantic relations (Miller, 1995). Mapping neuron-activated words/phrases to "synsets" enables clustering and generalization to broader, human-readable concepts.

Further reading: KG fundamentals (Hogan et al., 2021; Hitzler et al., 2010; Berners-Lee et al., 2023; Hitzler, 2021), Wikipedia taxonomy (Sarker et al., 2020; Ponzetto and Strube, 2007), and WordNet structure (Miller, 1995).

2.3. Concept Induction

Concept Induction (Lehmann and Hitzler, 2010) is a technique that automatically generates human-interpretable concept labels by analyzing positive and negative examples within a structured knowledge base. Originally developed for ontology engineering, it relies on deductive reasoning over description logics (Hitzler et al., 2010; Hitzler, 2021) and provides the critical bridge between raw neural activation patterns and symbolic, explainable concept labels in our neuron analysis framework.

The core idea is straightforward: given examples of what activates a neuron (positive examples) and what doesn't (negative examples), Concept Induction searches through a knowledge graph to find the most precise concept that covers all positive cases while excluding all negative ones (Figure 3). For instance, if a neuron consistently activates for "cars," "trucks," and "motorcycles" but not for "bicycles" or "pedestrians," the system may induce the concept "motor vehicle."

Concept Induction system accepts three inputs: (1) Positive examples P (high-activation images/texts), (2) negative examples N (low-activation cases), and (3) a knowledge base (or ontology) K. It returns description logic class expressions E such that all the positive examples and none of the negative examples are contained in each of the class expressions, represented as $K \models E(p)$ for all $p \in P$ and $K \not\models E(q)$ for all $q \in N$, with accuracy measures when perfect separation isn't possible.

We use the ECII (Efficient Concept Induction and Integration) system (Sarker and Hitzler, 2019b), designed for scalability over large ontologies. ECII systematically explores concept space using refinement operators and ranks candidates by precision and coverage to output the best-fitting labels.

Further reading: concept learning theory (Lehmann and Hitzler, 2010), ECII system (Sarker and Hitzler, 2019b), description logics (Hitzler et al., 2010), neuron analysis (Zeiler and Fergus, 2014; Luo et al., 2016), and its applications (Dalal et al., 2024).

2.4. Key Terminology

The following terms are used throughout this tutorial:

Positive Set: Input examples (images/texts) that strongly activate a neuron above a chosen threshold, representing patterns the neuron detects.

Negative Set: Inputs that do *not* strongly activate the same neuron (below threshold), used as contrastive evidence.

Target Label / Concept Label: Human-interpretable concept assigned to a neuron through Concept Induction (e.g., "vehicle," "building").

Non-Target Label: Any label not matching the assigned target/concept label for that neuron.

Target Label Activation Percentage (TLA): Percentage of inputs *belonging to the target concept* that activate the neuron carrying that label.

Non-target Label Activation Percentage (Non-TLA): Percentage of images *not* matching the target label that still activate the neuron.

Error Margin: The difference (TLA vs. Non-TLA), reflecting how reliably the neuron is associated with its concept label. Larger margins indicate more precise neuron-concept mapping.

3. Step-by-Step Methodology Walkthrough

This section presents the complete workflow of our approach for concept-based hidden neuron activation analysis, using the ADE20K dataset as a worked example, which was the basis of our initial study (Dalal et al., 2024). The pipeline includes five sequential stages: (1) dataset selection and preparation, (2) model training, (3) identification of neuron activation patterns, (4) mapping these patterns to symbolic concepts (Concept Induction), and finally (5) validating these concept assignments. The overall workflow is depicted in figure 4.

Step 1: Begin by selecting and preparing the dataset.
The ADE20K dataset (Zhou et al., 2019) contains over 27,000 images across 365 scene categories, with extensive pixel-level annotations of objects and object parts. While these annotations are not used during CNN training, they are leveraged later for generating label hypotheses in *Step 4*. For our approach, we select 10 scene categories with the largest number of images to ensure a balanced and computationally manageable subset: *bathroom, bedroom, building_facade, conference_room, dining_room, highway, kitchen, living_room, skyscraper, street.* This results in a total of 7,557 images. This selection is intentionally diverse and includes categories sharing common object types (e.g., "table," "chair") to observe how neurons can encode overlapping visual concepts. Images were resized to

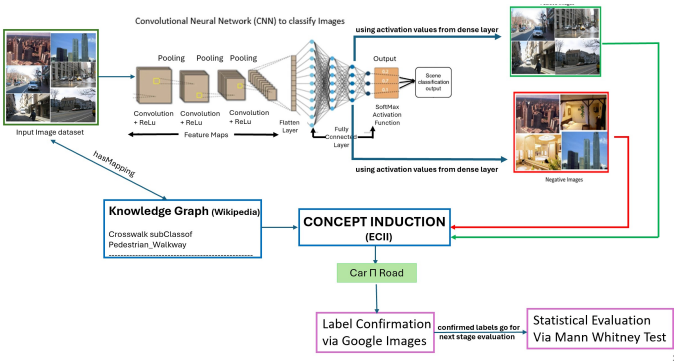

Figure 4: The overview of the workflow for Neurosymbolic Hidden Neuron Analysis in Convolutional Neural Networks.

224 × 224 pixels (standard input size for most CNN architectures). The pixel values are normalized using channel-wise mean and standard deviation normalization (commonly applied with ImageNet-pretrained models) to leverage pre-trained feature representations, and divided into training (5,500), validation (687), and test (1,370) sets. The users can choose any standard train/validation/test ratio.

Step 2: Train the neural network model for analysis.
Having prepared our dataset, the next step is establishing a trained neural network whose internal representations we can analyze. To prepare the model for interpretability analysis, we trained several widely used benchmark CNN architectures on the ADE20K subset, including VGG16 (Simonyan and Zisserman, 2015), InceptionV3 (Szegedy et al., 2016), ResNet50, ResNet152, ResNet101, and ResNet50V2 (He et al., 2016a,b). These are widely used baseline models with complementary design choices: plain deep stacks in VGG, multi-branch filters in Inception, and residual/skip connections in ResNets. They provide robust ImageNet-pretrained weights, making them ideal for our approach of comparing neuron behaviors across architectural families. Each architecture was fine-tuned on the dataset. The training configuration includes:

1. Input resolution: 224 × 224;
2. Optimizer: Adam (learning rate: 0.0001);
3. Loss function: Categorical cross-entropy;
4. Batch size: 32;
5. Epoch: 30;
6. Early stopping: monitoring val loss, patience = 3, restoring best weights.

We selected ResNet50V2 for the neuron activation analysis after evaluating the performance of several models. Its high classification accuracy (training accuracy: 87.60%; validation accuracy: 86.46%) and reasonable complexity for interpretability served as the basis for this decision. To maintain the emphasis on the interpretability workflow rather than reaching the highest classification accuracy, we purposefully steer clear of extensive hyperparameter tuning. This choice yields a competent model whose hidden units are rich enough to analyze while keeping the analysis tractable.

Tip: If computation is the bottleneck, start with ResNet50V2; if memory is tight, VGG16 is simpler to run but typically less data-efficient.

Step 3: Identify and extract neuron activation patterns. With our trained model in place, we now turn to the core question: *Which images cause each neuron to activate strongly, and which cause minimal activation?* Answering this gives us the foundation for mapping neurons to symbolic concepts. For the ADE20K experiment, we used the test split of 1,370 images and passed them through the trained ResNet50V2 model and recorded the activations from the dense layer. We focused on this layer because prior studies (Olah et al., 2017) show that: early CNN layers tend to detect low-level features (edges, textures, colors) while later CNN layers detect higher-level features (faces, roads, furniture, etc.), which better align with the semantic richness of our background knowledge.

Tip: For hidden neuron analysis, you can choose any mid-to-late convolutional or dense layer, depending on the semantic level you want to interpret. Early layers will give you primitive visual patterns while deeper ones reveal semantic content.

The dense layer in our model contains 64 neurons (a design choice balancing computational efficiency with sufficient representational capacity for our 10-class problem), and we analyze each neuron separately. For each neuron:

1. Extract activation values across all 1,370 test images;
2. Find the maximum activation value for that neuron;
3. Compute positive threshold = 80% of the max activation;
4. Compute negative threshold = 20% of the max activation;
5. Positive set: Images with activation $\geq$ positive threshold;
6. Negative set: Images with activation $\leq$ negative threshold.

We selected the 80% / 20% thresholds to capture neurons with strong activation preferences while filtering out noise from marginal responses. The 80% threshold ensures we focus on images that genuinely excite the neuron, while the 20% threshold identifies images the neuron actively ignores. This creates a clear contrast for Concept Induction while maintaining sufficient sample sizes for statistical analysis.

The positive set contains images the neuron is most responsive to, while the negative set contains those it responds to the least. These sets are the direct input for Concept Induction in *Step 4*, where we attempt to find symbolic descriptions that explain the neuron's behavior.

Step 4: Apply Concept Induction to map neuron activation to symbolic concepts. Having identified which images activate each neuron, we now face the central challenge: *What visual concepts do these activation patterns represent?* To bridge the gap between neural activations and human-interpretable concepts, we employ symbolic reasoning to assign meaningful labels to each neuron based on its activation behavior. To accomplish this, we utilize the Efficient Concept Induction and Integration (ECII) system, a Java-based application for Concept Induction from background knowledge and labeled examples. ECII supports ontology-based operations such as contextual data analysis, ontology creation, merging, pruning, and entity similarity measurement. The tool, along with detailed documentation, is publicly available at https://github.com/md-k-sarker/ecii.

In our workflow, ECII is used to induce logical concept expressions that differentiate the positive set and negative set of images associated with each neuron (as derived in *Step 3*), relative to a background ontology constructed from ADE20K object annotations and a Wikipedia-derived class hierarchy (Sarker and Hitzler, 2019a).

The complete setup for Concept Induction involves two main phases: preparing the background knowledge that will inform our concept reasoning, and then applying ECII to generate concept hypotheses for each neuron.

Preparation of Background Knowledge:

Ontology Creation from ADE20K Annotations: For each image in the ADE20K dataset, we create a lightweight ontology containing only the objects annotated as present in the image (e.g., "chair," "window," "table"). Richer information such as segmentation masks or part-whole relationships is excluded to keep the symbolic representation tractable.

All object labels are matched to corresponding classes in the Wikipedia-derived concept hierarchy using Levenshtein string similarity with edit distance 0, ensuring only exact lexical matches are retained (Levenshtein, 1975). For instance, an image annotated with "door" is linked to the "door" class in the hierarchy. No mapping is performed for the scene label (e.g., "kitchen") itself; images are linked to the ontology exclusively through their constituent objects.

Ontology Merging with Wikipedia: After generating ontologies for each image, we use ECII's Combine Ontologies feature to merge these with the Wikipedia-derived class hierarchy. This results in a single, integrated background ontology that can provide symbolic context for all images in the dataset. For large-scale hierarchies, ECII's Strip Down Ontology functionality can retain only dataset-relevant classes, improving computational efficiency.

Running ECII for Concept Induction: Once the background ontology is prepared, we invoke ECII's Contextual Data Analysis module to induce concept labels for each neuron. The inputs for each neuron are:

- P: Positive set (images resulting in high neuron activation);
- N: Negative set (images resulting in low neuron activation);
- K: Knowledge base (constructed from ADE20K annotations and the Wikipedia hierarchy).

The output is a concept expression that, according to ECII's reasoning, best distinguishes P from N using the background knowledge K. These expressions are logical class descriptions (often in description logic), which we refer to as the *target label* for the neuron. To assess the quality of each induced concept,

ECII computes a *Coverage Score*:

$$\text{coverage}(E) = \frac{|Z_1| + |Z_2|}{|P \cup N|}$$

where $Z_1 = \{ p \in P \mid K \models E(p) \}$ and $Z_2 = \{ n \in N \mid K \not\models E(n) \}$, P is the positive set, N is the negative set, and K is the knowledge base. In plain terms, coverage measures how well the induced concept expression matches the neuron's observed activation behavior across both positive and negative sets. A higher score indicates that the symbolic label faithfully captures the neuron's activation pattern.

In some cases, the induced concept may be a conjunction of multiple classes, for example: "mountain $\sqcap$ bush" (as description logic expression) or "mountain(x) $\wedge$ bush(x)" expressed in first-order predicate logic. This corresponds to the intuitive interpretation that the neuron tends to activate when both concepts are present in the image. In practice, we record such labels as comma-separated terms, e.g., "mountain, bush."

We consider these symbolic expressions to be *target label* or *label hypotheses* for the corresponding neuron's activation behavior. These hypotheses are subject to further evaluation in the subsequent sections.

Step 5: Evaluate the induced concepts through external validation and statistical testing.
Following the Concept Induction process described in *Step 4*, each neuron is assigned a symbolic *target label* (hypothesis) that characterizes its activation behavior in terms of high-level concepts from a structured ontology. In this section, we evaluate the validity and interpretability of these labels using two complementary strategies:

1. Verify semantic consistency through external visual confirmation;
2. Assess statistical separation of neuron activations across concept-based groups.

These methods work together to provide comprehensive validation: the first confirms that our labels make intuitive sense when tested against independent image sources, while the second ensures the observed patterns are statistically robust rather than coincidental.

Label Confirmation via Image Retrieval:
To test the validity of each neuron's *target label*, we perform a retrieval-based evaluation using Google Images. The idea is simple: if the concept label assigned to a neuron is accurate, then images associated with that concept from an external source (i.e., the web) should also trigger that neuron's activation. For each *target label*:

1. We search Google Images using the *target label* as the query;
2. If the label is a conjunction (e.g., "mountain, bush"), we require that all keywords be matched in the search results;
3. We download up to 200 JPEG images (readers may use 100–400 images depending on computational resources and desired statistical power) from the search results using the Imageye Chrome extension;[1]
4. We filter results to include only images with a resolution $\geq 224 \times 224$ pixels to match the CNN input size.

Of the retrieved images, 80% are used for label confirmation and the remaining 20% are reserved for the statistical evaluation. Each image from the 80% split is passed through the trained CNN, and we record the activations from the dense layer and assess whether the neuron assigned to the label activates, and whether other neurons also activate for these same images. To quantify this:

1. We compute the *Target Label Activation* (TLA) %, i.e., the percentage of retrieved images for a concept where the assigned neuron activates above a pre-defined threshold;
2. We compute the *Non-Target Label Activation* (Non-TLA) %, i.e., the percentage of those same images where any other neuron activates above the threshold.

We define a concept label for a neuron to be *confirmed* if the neuron activates in at least 80% of the retrieved images for its *target label* (TLA $\geq$ 80%). We chose this relatively high threshold to ensure strong neuron-concept associations while accounting for the inherent noise in web-retrieved images. Lower thresholds (e.g., to 60%) may capture weaker but still meaningful patterns, while higher thresholds (e.g., 90%) risk rejecting valid concepts due to image quality variations or retrieval noise.

Statistical Evaluation with the Mann-Whitney U Test:
While the image retrieval method confirms neuron-label associations in practical terms, we complement this with a formal statistical hypothesis test to measure whether the difference in activations is significant. For each neuron-label pair, we formulate the hypothesis:

1. Null hypothesis: There is no significant difference between the neuron's activation on target images (those retrieved using its label) and non-target images (those retrieved using other labels).
2. Alternative hypothesis: The neuron activates more strongly for target images.

Since we cannot assume the activation distributions are normal, and since sample sizes vary, we use the Mann-Whitney U test, a non-parametric statistical test that does not require distributional assumptions (McKnight and Najab, 2010). This test is particularly suitable for our scenario because: (1) neural activations often follow skewed or multimodal distributions, (2) different concept labels may yield different numbers of retrieved images, and (3) the test focuses on rank differences rather than absolute values, making it robust to outliers. We apply this test using the remaining 20% of images retrieved in the previous subsection. A *target label* is considered statistically validated if the test demonstrates that the neuron's response to target images is significantly higher than its response to non-target images. Specifically, we require: p-value < 0.05, and a negative

[1] https://chrome.google.com/webstore/detail/image-downloader-imageye/agionbommeaifngbhincahgmoflcikhm

z-score, indicating that activations for target images are significantly higher than for non-target images. This dual evaluation framework provides confidence that our neuron-concept assignments are both intuitive to humans and statistically robust, forming a solid foundation for interpreting the network's internal representations.

4. Case Studies

This section demonstrates the generalizability of our concept-based neuron analysis approach across different datasets and modalities. We apply the five-step methodology from Section 3 to two additional scenarios: SUN2012 image data (extending to different visual domains) and AG News text classification (extending to natural language processing). These case studies validate that our approach scales beyond the initial ADE20K demonstration while highlighting key adaptations needed for different data types.

4.1. Case 1: SUN2012 Image Dataset Analysis

We replicate the methodology on a second image dataset: SUN2012,[2] an expanded version of the SUN database (Xiao et al., 2010), designed for large-scale scene understanding with object context.

Step 1: Data Selection & Preparation.

The SUN2012 dataset stands out for its scale and scene diversity compared to ADE20K. With about 131,000 images and 908 scene categories, it provides much broader coverage of both indoor and outdoor environments than ADE20K (over 27,000 images across 365 scene categories). It has images ranging from indoor environments like bookstores and kitchens to outdoor settings like beaches and streets. SUN2012 also includes annotations for over 3,800 object categories along with scene attributes, enabling research that combines scene recognition, object detection, and contextual reasoning. Unlike ADE20K with pixel-level object annotations, SUN2012 has object annotations with polygon masks, which suffice for our analysis as we are focused solely on object presence within images.

Following our established approach, we select 10 scene categories with the highest image counts: "bathroom," "bedroom," "building facade," "dining room," "highway," "kitchen," "living room," "mountain snowy," "skyscraper," and "street." This results in a subset of 3,950 images (considering 10 categories maintains dataset balance and computational manageability; readers can adjust the number of categories based on available computational resources and research objectives), from which 3,157 are used for training and validation (at a 90:10 ratio) and 793 are reserved for testing (20% of the entire subset).

Step 2: Model Training.

With the dataset subset prepared, the next step is to train neural network models suitable for interpretability studies. To keep the workflow consistent with Section 3, we train the same benchmark architectures from our ADE20K study: VGG16, VGG19 (Simonyan and Zisserman, 2015), InceptionV3, ResNet50, ResNet101, ResNet152, and ResNet50V2 -ensuring consistent evaluation criteria and enabling cross-dataset performance comparison. These networks are fine-tuned using a set of 3,157 images, divided into training and validation sets. The training configuration is as follows:

1. Input resolution: 299×299 (default for InceptionV3) and 224×224 (default for the other models);
2. Optimizer: Adam (learning rate: 0.001);
3. Loss function: Categorical cross-entropy;
4. Batch size: 32;
5. Epoch: 30
6. Early stopping: monitoring val loss, patience = 3, restoring best weights.

We select InceptionV3 for neuron activation analysis due to its superior classification performance compared to other tested models, achieving 96.83% training accuracy and 92.71% validation accuracy.

Step 3: Neuron Activation Extraction.

After model training, we conduct the neuron-level analysis on the dense layer of the trained InceptionV3 model using 793 test images from the SUN2012 dataset. As stated in Section 3, we select this layer, containing 64 neurons, because later network layers typically encode more abstract and semantically meaningful patterns. For each neuron, activation responses across the test dataset are evaluated, and we apply identical thresholds for positive ($\geq$ 80% of the maximum activation) and negative ($\leq$ 20%) set creation, leveraging the same rationale established in our main methodology. These contrasting sets are used as input to the Concept Induction system ECII.

Step 4: Apply Concept Induction.

As in Section 3 *Step 4*, we use ECII to induce symbolic concept labels for each neuron's activation pattern by contrasting its positive and negative image sets. Here we only note the SUN2012-specific setup:

Preparation of Background Knowledge:

Ontology Creation from SUN2012 Annotations: For each image, a lightweight ontology is built using only annotated objects mapped to exact lexical matches in a Wikipedia-based concept hierarchy.

Ontology Merging with Wikipedia: The per-image ontologies are combined with the Wikipedia hierarchy using ECII, resulting in a unified background ontology.

Running ECII for Concept Induction:

We apply identical Concept Induction procedures, with ECII's Contextual Data Analysis module: given each neuron's positive/negative sets and the SUN2012-adapted knowledge base, ECII returns a concise class expression scored by coverage score that distinguishes positive from negative activation sets. We consider these generated *target labels or label hypotheses* for subsequent evaluation.

Table 1 shows the labels received from ECII for the neurons of the hidden layer. If we take a closer look in the table, Neuron 7 is assigned the concepts "snow" and "mountain" with a coverage score of 1.0, a TLA of 80%, and Non-TLA of 39.55%.

[2] https://groups.csail.mit.edu/vision/SUN/hierarchy.html

This indicates that the neuron is highly specialized for snowy mountain features, with strong alignment to its target concept. This highlights how ECII uncovers interpretable semantic roles of individual neurons within the network. In this step, we obtain 32 confirmed neurons having TLA ≥ 80%.

Table 1: Concept Induction for Case 1. **Bold** rows indicate TLA % ≥ 80.

Neuron ID	ECII Concepts	Coverage Score	TLA %	Non-TLA %
0	**snowy_mountain**	**0.99**	**95.00**	**52.57**
1	microwave, microwave	0.92	37.84	35.27
2	dock, dock	0.94	51.19	42.71
3	river	0.96	49.20	53.34
4	shower, wicket	0.99	38.75	34.51
5	spice_rack, mitten	0.97	59.43	60.42
6	sky	0.93	78.50	68.51
7	**snow, mountain**	**1.00**	**80.00**	**39.55**
8	toy, plaything	0.99	30.00	32.21
9	**field**	**0.97**	**93.75**	**74.96**
10	person_walking, rubbish	0.99	48.75	30.20
11	bath, candle	0.99	45.00	53.96
12	side, sculpture	0.98	70.00	63.98
13	**dish_rack**	**0.89**	**81.25**	**31.12**
14	cutter	0.98	70.00	53.92
15	**pillow, ceiling_fan**	**0.99**	**88.75**	**57.15**
16	**skyscraper, river_water**	**1.00**	**80.00**	**42.05**
17	balcony	0.88	51.25	59.18
18	**city**	**0.99**	**87.50**	**68.62**
19	**snowy_mountain**	**0.98**	**97.50**	**46.58**
20	walk, crossing	0.99	63.75	43.94
21	**bedroom, duvet**	**0.97**	**85.00**	**47.67**
22	**dishwasher**	**0.94**	**96.20**	**24.58**
23	**fence**	**0.96**	**98.75**	**77.85**
24	**sink**	**0.94**	**96.25**	**52.59**
25	toilet_lid, tissue_box	0.98	26.25	24.87
26	**toilet_tissue**	**0.98**	**95.00**	**39.49**
27	**toilet**	**0.96**	**95.00**	**55.33**
28	**cars**	**0.90**	**97.50**	**72.79**
29	bathtub, cup	0.99	72.50	48.56
30	hanger, apparel	0.99	72.50	58.42
31	**snowy_mountain**	**0.95**	**97.50**	**67.00**
32	**shell**	**0.95**	**80.00**	**46.64**
33	**bed, pillow**	**0.95**	**80.00**	**34.41**
34	little_bear, cabinet_dresser	0.96	0.00	54.89
35	bottle_soap, faucet	0.99	33.75	39.02
36	**snowy_mountain**	**0.97**	**98.75**	**68.33**
37	display_window	0.99	17.50	34.27
38	cutter	0.93	70.00	67.53
39	food, decorative_plate	0.99	18.75	23.99
40	**plant**	**0.97**	**83.75**	**51.00**
41	**pole, handrail**	**1.00**	**85.00**	**70.35**
42	**skyscraper**	**0.97**	**93.75**	**49.47**
43	**skyscraper**	**0.98**	**100.00**	**66.88**
44	pole, field	1.00	75.00	59.65
45	car_rear	0.98	67.50	61.64
46	clothes_hook, bidet	0.97	61.25	54.73
47	**crosswalk**	**0.98**	**81.25**	**23.42**
48	**bidet**	**0.97**	**98.75**	**57.35**
49	**ironing_board, shoe**	**0.97**	**93.75**	**63.09**
50	**field**	**0.98**	**83.75**	**66.54**
51	jug	0.96	32.50	41.93
52	display_window	0.92	47.50	25.14
53	candlestick, chest_of_drawers	0.99	65.00	31.00
54	**snowy_mountain**	**0.96**	**92.50**	**40.80**
55	rucksack	0.99	75.00	49.02
56	tree, streetlight	1.00	18.75	38.38
57	knives	0.96	57.50	44.08
58	**snowy_mountain**	**0.97**	**97.50**	**55.05**
59	sink	0.96	72.50	44.25
60	**brocoli, cheese**	**0.98**	**93.75**	**39.00**
61	**cars**	**0.98**	**90.00**	**45.51**
62	**air_conditioning, chest_of_drawers**	**0.98**	**87.50**	**61.18**
63	crosswalk	0.97	65.00	26.97

Step 5: Evaluate the induced concepts.

Label Confirmation via Image Retrieval:

To evaluate the validity of each neuron's target label, we employ a retrieval-based approach using Google Images, following *Step 5* of Section 3. For each target label, up to 100 images are retrieved at a minimum resolution of 299×299 pixels, with 80% allocated for label confirmation and the remaining 20% reserved for statistical evaluation, mentioned in *Step 5* of Section 3. A label is considered confirmed if TLA ≥ 80%, a threshold chosen to ensure robust neuron-concept associations despite variability in web images.

In Table 2, the results of this Google images-based validation are shown. For instance, Neuron 23, labeled "fence," achieves a TLA of 98.75%, while Neuron 47, labeled "crosswalk," achieves a TLA of 81.25%. After this step, a total of 32 neurons demonstrate consistency between their assigned labels and independently retrieved web images.

Table 2: Label confirmation via Google Images for Case 1 showing the confirmed neurons (TLA % ≥ 80).

Neuron ID	ECII Concepts	Coverage Score	TLA %	Non-TLA %
0	snowy_mountain	0.986	95.00	52.57
7	snow, mountain	0.998	80.00	39.55
9	field	0.971	93.75	74.96
13	dish_rack	0.892	81.25	31.12
15	pillow, ceiling_fan	0.992	88.75	57.15
16	skyscraper, river_water	0.997	80.00	42.05
18	city	0.993	87.50	68.62
19	snowy_mountain	0.975	97.50	46.58
21	bedroom, duvet	0.967	85.00	47.67
22	dishwasher	0.945	96.20	24.58
23	fence	0.965	98.75	77.85
24	sink	0.939	96.25	52.59
26	toilet_tissue	0.979	95.00	39.49
27	toilet	0.961	95.00	55.33
28	cars	0.903	97.50	72.79
31	snowy_mountain	0.948	97.50	67.00
32	shell	0.950	80.00	46.64
33	bed, pillow	0.945	80.00	34.41
36	snowy_mountain	0.973	98.75	68.33
40	plant	0.972	83.75	51.00
41	pole, handrail	0.997	85.00	70.35
42	skyscraper	0.969	93.75	49.47
43	skyscraper	0.978	100.00	66.88
47	crosswalk	0.983	81.25	23.42
48	bidet	0.966	98.75	57.35
49	ironing_board, shoe	0.969	93.75	63.09
50	field	0.976	83.75	66.54
54	snowy_mountain	0.965	92.50	40.80
58	snowy_mountain	0.969	97.50	55.05
60	brocoli, cheese	0.979	93.75	39.00
61	cars	0.976	90.00	45.51
62	air_conditioning, chest_of_drawers	0.982	87.50	61.18

Statistical Evaluation:

Next, we carry out a statistical evaluation of each neuron's label using the remaining subset of Google Image samples. Specifically, we apply the non-parametric Mann–Whitney U test to assess whether there is a statistically significant difference in activation levels between target and non-target labels. Following *Step 5* of Section 3, a label is considered statistically supported if $p < 0.05$ and the z-score is negative, indicating that activations for target images are significantly stronger than those for non-target images. For instance, in Table 3, Neuron 19, labeled *snowy_mountain*, achieves a z-score of -6.55 with $p < 0.00001$, showing strong statistical support for the label, as target images consistently triggered higher activations. In contrast, Neuron 28, labeled as *cars*, records $p = 0.268$, which does not meet the statistical threshold and thus fails to reject the null hypothesis. Out of 32 confirmed neurons, we fail to reject the null hypothesis in only 3 cases, implying that the majority of neuron-concept associations are statistically well-supported and reflect meaningful distinctions in activation patterns.

Table 3: Statistical evaluation of Case 1. **Bold** rows indicate neurons with p-value ≥ 0.05, where we **cannot reject the null hypothesis**.

Neuron ID	ECII Concepts	TLA %	Non-TLA %	Target Median	Non-Target Median	Target Mean	Non-Target Mean	z-score	p-value
0	snowy_mountain	95	53.02	5.99	0.23	5.58	1.22	-6.18	< 0.00001
7	snow, mountain	70	39.68	2.10	0.00	1.74	0.68	-3.04	0.00061
9	field	100	75.48	4.38	1.43	4.03	1.97	-3.64	0.00025
13	dish_rack	90	32.62	1.65	0.00	1.68	0.56	-4.66	< 0.00001
15	pillow, ceiling_fan	90	54.21	3.03	0.21	2.36	0.85	-4.40	< 0.00001
16	skyscraper, river_water	70	43.41	1.54	0.00	1.44	0.45	-3.15	0.00052
18	city	80	71.11	2.17	0.82	2.03	1.10	-2.85	0.00398
19	snowy_mountain	100	45.95	5.91	0.00	5.39	1.09	-6.55	< 0.00001
21	bedroom, duvet	65	45.95	0.86	0.00	1.91	0.57	-2.70	0.00332
22	dishwasher	80	26.11	1.60	0.00	1.87	0.29	-5.04	< 0.00001
23	fence	100	78.73	3.07	1.61	2.96	1.80	-3.40	0.00063
24	sink	95	53.25	3.14	0.11	3.14	0.81	-5.60	< 0.00001
26	toilet_tissue	100	38.65	2.47	0.00	2.36	0.51	-6.29	< 0.00001
27	toilet	95	56.98	4.06	0.33	3.88	0.94	-6.10	< 0.00001
28	**cars**	**95**	**72.78**	**1.58**	**1.03**	**1.66**	**1.59**	**-1.10**	**0.26788**
31	snowy_mountain	90	67.62	4.58	0.62	3.97	1.35	-5.05	< 0.00001
32	shell	85	46.19	1.65	0.00	1.55	0.70	-3.77	0.00004
33	bed, pillow	75	35.87	3.19	0.00	2.60	0.43	-4.62	< 0.00001
36	snowy_mountain	100	67.62	4.72	0.93	4.60	1.48	-6.24	< 0.00001
40	**plant**	**70**	**51.90**	**1.00**	**0.07**	**0.86**	**0.68**	**-1.44**	**0.12808**
41	pole, handrail	90	69.21	2.23	1.05	2.05	1.51	-2.18	0.02735
42	skyscraper	95	50.79	2.91	0.05	2.78	1.01	-4.49	< 0.00001
43	skyscraper	100	66.51	2.94	0.81	3.01	1.10	-5.65	< 0.00001
47	crosswalk	95	22.94	3.20	0.00	3.20	0.29	-6.74	< 0.00001
48	bidet	100	58.02	3.00	0.43	2.91	1.01	-5.35	< 0.00001
49	ironing_board, shoe	90	62.06	1.98	0.55	2.20	1.02	-3.37	0.00053
50	**field**	**85**	**66.83**	**0.98**	**0.73**	**1.15**	**1.11**	**-0.89**	**0.36257**
54	snowy_mountain	100	39.68	4.53	0.00	4.01	0.72	-6.58	< 0.00001
58	snowy_mountain	95	52.94	4.56	0.18	4.77	1.12	-6.33	< 0.00001
60	brocoli, cheese	95	39.52	1.79	0.00	1.93	0.50	-5.89	< 0.00001
61	cars	85	45.95	1.41	0.00	1.32	0.55	-4.07	< 0.00001
62	air_conditioning, chest_of_drawers	85	62.22	2.12	0.50	2.59	0.95	-3.92	0.00006

4.2. Case 2: AG News Text Dataset Analysis

Concept-based hidden neuron activation analysis, originally applied to images using datasets like ADE20K and SUN2012, can also be adapted to text. The overall pipeline remains the same—identifying neuron activations, forming positive and negative sets, and inducing concepts—but with modality-specific adjustments. While image-based methods rely on ECII to generate semantic concepts from visual features, in the text domain we instead use *WordNet*, a lexical database that captures linguistic and semantic relations (e.g., hypernyms, hyponyms). This shift preserves the core idea of linking neuron activations to interpretable concepts, while adapting the concept-generation step to the linguistic domain. Figure 5 shows the overall workflow of this approach.

Step 1: Data Selection & Preparation.

For the text case study, we use the AG News topic classification dataset (Zhang et al., 2015), a standard benchmark in NLP. It consists of about 127,600 news snippets, categorized into four topical domains: World, Sports, Business, and Science/Technology. In the original dataset the training portion includes 30,000 samples per class, while the test portion contributes 1,900 per class totaling 120,000 training examples and 7,600 test examples. We take another set of 7,600 samples as validation from the training set. Each instance is a brief piece of news text—often a headline with an opening sentence—capturing topical cues. Articles in the World category focus on global events and conflicts, Sports samples cover games and athletes, Business entries report on corporate and financial updates, and Sci/Tech covers technology releases and scientific developments. The dataset's balanced structure makes it well-suited for analyzing how neurons respond to distinct, semantically coherent topics. Representative samples from the AG News topic classification are shown in Table 4. The relevant concepts are shown in bold in the sentences.

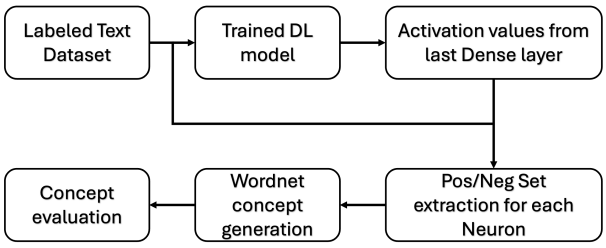

Figure 5: Workflow for generating concepts from text data.

Step 2: Model Training.

We train three common architectures for text classification: LSTM (Bengio et al., 1994), BiLSTM (Graves et al., 2005), and 1D CNN (Kim, 2014). To ensure a fair comparison, all models are trained with the same number of layers and identical hyperparameters, and without extensive tuning; the goal is not to chase peak accuracy but to obtain a capable model with hidden units which can be analyzed consistently across architectures. The training configuration is as follows:

1. Input resolution: 956×100 tensor shape (Max sentence length × Embedding vector length);
2. Optimizer: Adam (learning rate: 0.0001);

Table 4: Data samples from AG News topic classification dataset. **Bold** words refer to the relevant concepts.

Text	Class
GAZA CITY: The Israeli army demolished 13 Palestinian houses during an incursion in the southern **Gaza strip** town of Rafah on Thursday, Palestinian security sources and witnesses said.	World
BAGHDAD (Reuters) - At least 110 people were killed across **Iraq** on Sunday in a sharp escalation of violence that saw gun battles, car bombs and bombardments rock the capital.	World
AP - Arsenal extended its unbeaten streak in the Premier **League** to 48 games Saturday, getting two goals from Thierry Henry in a 4-0 victory over Charlton and bouncing back from a Champions **League** tie.	Sports
AP - Three New York Giants have filed complaints with the NFL Players **Association** after being fined by coach Tom Coughlin for not being "early enough" to team meetings.	Sports
Stocks fell on Wednesday after investment bank Morgan Stanley (MWD.N: Quote, Profile, Research) said quarterly profit fell, casting doubt on corporate profit growth, while a brokerage downgrade on Cisco Systems Inc.	Business
US **stocks** fell as setbacks for drugmakers, including a study showing Pfizer Inc.'s Celebrex painkiller increased the risk of heart attacks, sent health-care shares tumbling.	Business
CHICAGO - Hewlett-Packard (HP) has moved its Active Counter Measures network security **software** into beta tests with a select group of European and North American customers in hopes of readying the product for a 2005 release, an HP executive said at the HP World conference here in Chicago Wednesday.	Sci/Tech
AUGUST 18, 2004 (IDG NEWS SERVICE) - A majority of US home **Internet** users now have broadband, according to a survey by NetRatings Inc.	Sci/Tech

3. Loss function: Categorical cross-entropy;
4. Batch size: 32;
5. Epoch: 15;
6. Early stopping: monitoring val loss, patience = 3, restoring best weights.

The LSTM model demonstrates the strongest performance, achieving about 90% weighted precision, recall, and accuracy. For the subsequent neuron-level analysis, we proceed with LSTM as the primary text model.

Tip: If using pretrained embeddings (e.g., GloVe), fix the embedding matrix during training to reduce variance across runs; if training embeddings from scratch, set a random seed and report it alongside the Max_Sentence_Length.

Step 3: Neuron Activation Extraction.

Following model training, we pass the test set through the model and record activations from the final dense layer of 64 ReLU neurons. As in Section 3 *Step 3*, for each neuron, the maximum observed activation is computed over the test set and used to define two thresholds: 80% of the maximum (positive) and 20% of the maximum (negative). Instances with activations exceeding the positive threshold are assigned to the positive set P, while those below the negative threshold are assigned to the negative set N. This procedure provides a structured summary of neuron responsiveness for text inputs (derived from the model's sequence representation) and forms the basis for subsequent concept generation and analysis.

Step 4: Apply Concept Induction.

As an adaptation of Section 3 *Step 4* for text, we induce human-interpretable concept labels using *WordNet* rather than ECII. The procedure is contrastive (positive vs. negative sets) and straightforward as can be seen below:

1. For each neuron's positive set, we first tokenize, lowercase, lemmatize, keep content words, retain nouns via POS tagging, and extract the most salient terms from its positive set using TF-IDF weighting;
2. Remove terms that are frequent in the neuron's negative set to ensure discriminative relevance (e.g., drop terms with similar TF-IDF in negatives);
3. Map the remaining terms to their most representative WordNet noun synsets automatically by retrieving their noun senses using NLTK (Bird et al., 2009), producing semantically coherent concept labels. For ambiguous terms, choose the synset with the highest corpus frequency; multiword terms use the head noun.

The resulting synsets form the neuron's *target labels* or *label hypotheses*—concise, linguistically grounded descriptors of the conditions that most strongly elicit that neuron's activation. This mirrors the image pipeline's goal (linking activations to concepts) while aligning Concept Induction with the structure and semantics of language.

Step 5: Evaluate the induced concepts.

Label Confirmation:

For concept evaluation, we exclude neurons that show no activation across the dataset or fail to produce valid WordNet synsets, resulting in 53 candidate neurons. For each of these, we generate 20 target sentences (containing its induced target label) and 20 non-target sentences (excluding the target label) using GPT-4. The number of generated texts is chosen empirically: it provides enough examples to observe consistent

activation patterns while keeping the generation and evaluation pipeline computationally manageable (readers can use any number of sentences depending on computational resources).

Following the procedure from Section 3 *Step 5*, half of this dataset is used for label confirmation and the remaining half for statistical evaluation. Neuron responses are quantified using two measures: Target Label Activation (TLA) and Non-Target Label Activation (Non-TLA). A label is considered confirmed if TLA $\geq$ 80%, a threshold chosen to ensure robust neuron-concept associations. Table 5 presents the outcomes of the label confirmation step along with the assigned labels. For example, the *Target %* column indicates the proportion of target sentences that triggered each neuron, defined as activations exceeding 80% of the neuron's maximum value. Out of 53 neurons, 31 are confirmed, indicating strong alignment between the WordNet-derived labels and observed activation patterns.

Statistical Evaluation:
We now reevaluate the 31 confirmed neurons using the held-out half of the generated samples (Section 3 *Step 5*). Of these, 25 neurons consistently maintain $\geq$ 80% on the held-out set and proceed to statistical testing. We now apply the non-parametric Mann-Whitney U test to assess whether there is a statistically significant difference in activation levels between target and non-target sentences under the null hypothesis that there is no difference in activation levels, and the alternative hypothesis that the neuron activates more strongly for target sentences.

Statistical testing confirms that 23 of these 25 neurons show significantly higher activation for target samples ($p < 0.05$), indicating clear separation between target and non-target activations. This confirms that the majority of neurons examined are not only interpretable but also robust in capturing semantically meaningful concepts. Table 6 summarizes Target and Non-Target level activation rates (TLA and Non-TLA) with statistical validation, focusing on neurons where TLA % $\geq$ 80.

Neurons 52 and 62, both linked to *company, institution*, fail to reach significance $p = 0.141$ and $p = 0.076$. In contrast, neuron 55 (*software*, *code*, *internet*) is activated on 90% of target cases and only 20% of non-target sentences, with high statistical significance ($p = 0.0002$), demonstrating strong and reliable concept encoding within the Sci/Tech class. To better understand the difference between target and non-target sentences, some sample target and non-target sentences are shown in Table 7. The associated labels are marked in bold in the target sentences. Overall, our findings confirm that most neurons respond selectively to specific semantic concepts, which in turn guide the model's classification behavior and support both *interpretability* and *robustness* of the induced labels.

Table 5: Label confirmation for Case 2. **Bold** rows indicate confirmed neurons (TLA % $\geq$ 80).

Neuron	Labels	TLA %	Non-TLA %
1	**iraq, gaza_strip, baghdad**	**100**	**10**
2	time_period, league, association	70	10
3	**iraq, gaza_strip, baghdad**	**90**	**10**
4	**iraq, gaza_strip, baghdad**	**100**	**10**
5	company, institution, corporation	10	20
6	league, association, activity	70	10
7	oil, lipid	60	10
8	stocks, framework	30	10
10	**company, institution**	**80**	**20**
11	freestyle, race, thorpe	0	0
12	**game, activity, time_period**	**100**	**10**
13	**time_period, activity, league**	**90**	**30**
14	time_period, league, association	40	40
15	gaza_strip, iraq, baghdad	50	10
17	software, code	50	10
18	**software, code, internet**	**100**	**30**
19	time_period, league, association	10	10
21	**oil, lipid, monetary_value**	**80**	**20**
23	**iraq, corporate_executive**	**80**	**10**
24	**software, code, internet**	**80**	**10**
25	**time_period, activity, league**	**80**	**0**
28	time_period, league, association	70	10
30	iraq, corporate_executive	60	10
31	time_period, league, association	60	0
32	**time_period, league, association**	**90**	**10**
33	**activity, league, association**	**90**	**10**
34	**gaza_strip, iraq, baghdad**	**80**	**20**
35	**iraq, gaza_strip, baghdad**	**90**	**0**
36	**gaza_strip, iraq, baghdad**	**80**	**0**
37	**space, attribute**	**100**	**20**
38	**company, institution**	**80**	**20**
39	iraq, baghdad	50	20
40	**time_period, activity, league**	**80**	**40**
41	**activity, time_period, rest_day**	**100**	**20**
42	**iraq, baghdad, gaza_strip**	**100**	**10**
43	iraq, baghdad	70	30
45	stocks, framework	60	20
46	chromatic_color, score, high_status	20	0
48	**game, activity, league**	**100**	**10**
49	**oil, lipid, monetary_value**	**90**	**20**
50	**space, attribute, internet**	**100**	**10**
51	**software, code**	**100**	**40**
52	**company, institution**	**80**	**30**
53	stocks, framework	60	20
54	high_status, association, ordering	10	0
55	**software, code, internet**	**90**	**20**
57	oil, lipid	50	20
58	company, institution	60	20
59	**league, association, activity**	**100**	**20**
60	**software, code, company**	**100**	**10**
61	**software, code, internet**	**80**	**20**
62	**company, institution**	**90**	**20**
63	**gaza_strip, iraq, israeli**	**80**	**10**

5. Discussion

Our cross-domain case studies showcase the adaptability of concept-based hidden neuron activation analysis and reveal some challenges and trade-offs when transferring the methodology across modalities. Both studies progressed through Concept Induction and statistical validation, providing insights into neuron-concept alignment, but reliability assessments via error-margin analysis remains beyond the current scope.

Table 6: Statistical evaluation of Case 2. **Bold** rows indicate neurons with p-value ≥ 0.05, where we cannot reject the null hypothesis

Neuron ID	Labels	TLA %	Non-TLA %	z-score	p-value
1	iraq, gaza_strip, baghdad	80	10	-3.7796	0.0001
3	iraq, gaza_strip, baghdad	90	10	-3.7796	0.0001
4	iraq, gaza_strip, baghdad	90	10	-3.7796	0.0001
12	game, activity, time_period	100	20	-3.7796	0.0002
13	time_period, activity, league	80	30	-3.7796	0.0002
18	software, code, internet	90	30	-3.7796	0.0002
24	software, code, internet	80	10	-3.7796	0.0002
25	time_period, activity, league	80	10	-3.7796	0.0001
33	activity, league, association	100	20	-3.7796	0.0002
34	gaza_strip, iraq, baghdad	90	10	-3.7796	0.0001
35	iraq, gaza_strip, baghdad	90	20	-3.7796	0.0001
36	gaza_strip, iraq, baghdad	90	10	-3.7796	0.0001
37	space, attribute	90	10	-3.7041	0.0002
38	company, institution	80	40	-2.0410	0.0452
40	time_period, activity, league	80	20	-3.7796	0.0002
41	activity, time_period, rest_day	100	20	-3.7796	0.0001
42	iraq, baghdad, gaza_strip	90	10	-3.7796	0.0001
50	space, attribute, internet	80	10	-3.7796	0.0001
51	software, code	100	20	-3.7796	0.0002
52	**company**, **institution**	**80**	**30**	**-1.5119**	**0.1405**
55	software, code, internet	90	20	-3.7796	0.0002
59	league, association, activity	100	20	-3.7796	0.0001
60	software, code, company	80	20	-3.6285	0.0003
61	software, code, internet	90	20	-3.7796	0.0002
62	**company**, **institution**	**80**	**30**	**-1.8142**	**0.0757**

5.1. Challenges in Cross-Domain Application

Applying the same workflow to vision and language data underscored the importance of domain-appropriate background knowledge and evaluation strategies. For the SUN2012 dataset, concepts derived from object annotations mapped smoothly to a Wikipedia-based hierarchy, producing interpretable labels such as *snowy mountain* or *toilet*. In contrast, the AG News dataset required additional preprocessing steps such as TF-IDF term extraction and WordNet mapping, which introduced ambiguity when aligning linguistic terms with symbolic concepts. For readers, this highlights an important lesson: expect to adjust preprocessing to match the domain, and recognize that different symbolic resources bring different strengths and weaknesses.

Similarly, evaluation strategies diverged: images relied on retrieval-based verification, while text required synthetic sentence generation, each with its own limitations in scalability and bias. These differences highlight that while the pipeline is generalizable, its effectiveness depends critically on tailoring the symbolic resources and evaluation procedures to the domain.

5.2. Trade-offs Between Interpretability and Complexity

The case studies also revealed a balance between the interpretability of symbolic neuron labels and the complexity of the analysis workflow:

- Interpretability gains: Both studies showed that individual neurons often encode semantically coherent concepts (e.g., "snowy mountain" in SUN2012, "software, code" in AG News), and statistical validation confirmed that these associations are not coincidental. Such labels provide human-aligned explanations of hidden layer behavior, going beyond coarse attribution maps.

- Complexity costs: These interpretability gains come with substantial computational and methodological overhead. Constructing background ontologies, aligning terms, running Concept Induction, and validating labels each require domain-specific resources and additional computation. This workflow is "heavier" than gradient-based saliency methods, but provides richer interpretability.

- Granularity vs. scalability: Neuron-level labeling offers fine-grained interpretability, but scaling this approach to hundreds of thousands of neurons in modern architectures (e.g., transformers) presents a significant challenge. In practice, sampling subsets of neurons or aggregating across layers may be necessary compromises.

5.3. Lessons Learned from Early Results

Since both case studies concluded at statistical validation without error-margin analysis, the findings represent early but meaningful evidence of concept-neuron alignment. From the SUN2012 experiments, we learned that scene-level datasets with rich object diversity produce a wide range of interpretable neuron labels, though some neurons remain ambiguous or respond to multiple concepts. From the AG News experiments, we observed that textual neurons could reliably capture semantically related clusters of terms, such as geopolitical entities or

Table 7: Target vs. non-target sentences. **Bold** words indicate the labels in the target sentences.

Neuron ID	Label	Target Text	Non-Target Text
1	iraq, gaza_strip, baghdad	A car bomb detonated near a busy market in **Baghdad**, **Iraq**, killing at least 12 civilians and injuring dozens more, Iraqi police reported.	BOSTON - A university team developed an AI algorithm to predict weather patterns with unprecedented accuracy.
		Palestinian authorities reported a ceasefire violation after an airstrike in the **Gaza Strip** wounded several civilians.	CHICAGO - A leading pharmaceutical company announced a new drug trial for a treatment aimed at combating Alzheimer's disease.
55	software, code, internet	NEW YORK - A fintech startup released a blockchain-based **software** to secure **internet** transactions.	A major retailer reported lower-than-expected quarterly earnings, citing supply chain disruptions.
		BERLIN - A new open-source **code** repository gained traction, enabling developers to collaborate over the internet.	MOSCOW - Russian diplomats met with international leaders to discuss peacekeeping efforts in conflict zones.

technology-related concepts, but with greater variability in precision compared to images. Readers should expect that text-based models may show more distributed concept representations, while CNNs often yield more localized, object-focused neurons. These differences suggest that text-based models may exhibit more distributed representations, whereas CNNs often yield more localized, object-focused neurons.

Overall, the cross-domain application highlights both the promise and the limitations of the approach:

Pros:

- Generalizable pipeline applicable to both image and text modalities,
- Symbolically grounded concepts enable intuitive human interpretation, and
- Statistical validation ensures rigor beyond anecdotal examples.

Cons:

- Dependence on external knowledge sources (Wikipedia for vision, WordNet for text) introduces variability in label quality, and
- Evaluation procedures differ across modalities, which limits direct comparability.

5.4. Useful Insights Moving Forward

A key lesson from both case studies is that concept-based neuron labeling is feasible and informative; complementary evaluation metrics are essential to build trust in practical applications. The cross-domain experiments also indicate that the symbolic grounding step serves as the cornerstone of generalizability: the choice of ontology or lexical resource directly shapes the clarity and reliability of neuron labels. Finally, these early results suggest that while neuron-level interpretability is achievable, future work must address scalability and reliability to make the approach usable for modern, large-scale neural networks. For practitioners, a practical takeaway is to begin with smaller models and datasets to establish the workflow before scaling to more complex architectures.

6. Practical Guidelines

In Section 5, we reflected on insights and limitations emerging from our cross-domain experiments. This section shifts focus to practice, where we provide actionable guidelines for practitioners and researchers who wish to apply concept-based hidden neuron activation analysis in their own work. These guidelines emphasize when the method is most useful, how to tailor it to different settings, and concrete steps for integrating it into model development workflows.

6.1. When to Use This Method

This method is particularly suitable in scenarios where:

- Interpretability is critical: For domains such as healthcare, finance, or law, where transparency and accountability are essential, neuron-level explanations grounded in human concepts offer added trust.
- Debugging model behavior: When a model underperforms or shows unexplained biases, mapping neurons to symbolic concepts helps identify which internal representations may be misaligned.
- Educational and exploratory analysis: For teaching neural network interpretability or exploring novel architectures, the method provides tangible neuron-concept mappings that aid human understanding.

6.2. Choosing Evaluation Techniques

Selecting appropriate evaluation strategies depends on the data modality and practical constraints:

- Image data: Use external image retrieval (e.g., Google Images) or curated datasets to confirm concept labels. Web retrieval offers breadth but introduces noise; curated sets improve reliability but may limit coverage. For reproducibility, fix the number of retrieved images (e.g., 100-200) and report filtering criteria (minimum resolution, query terms).

- Text data: Map salient terms from neuron activations to WordNet or other lexical resources, and generate synthetic examples using LLMs or controlled templates to test neuron-concept associations. Readers should be aware that lexical resources differ in coverage; WordNet is useful for general English but may miss domain-specific terms.
- Statistical validation: Employ non-parametric tests such as Mann-Whitney U to verify that observed activations for target concepts differ significantly from non-targets. This step helps separate genuine concept alignment from coincidental associations. Always report outcomes, such as p-values or z-scores, alongside measures, like TLA vs. Non-TLA, to provide a clear picture of the reliability of concept assignments. If differences are *not* significant, the alignment should be treated as inconclusive, and practitioners may refine their positive/negative sets or reconsider candidate concepts.

6.3. Integration into Model Debugging Workflows

Concept-based neuron analysis can be most effective when integrated into broader model evaluation and debugging workflows. In this context, it may improve the interpretability of results, highlight hidden sources of bias, and complement other evaluation techniques by providing neuron-level insights:

- Bias detection: Neuron labels can reveal when models rely on irrelevant or socially sensitive features (e.g., background artifacts in images, named entities in text).
- Model simplification: In principle, identifying neurons with overlapping or redundant concepts could inform pruning or compression strategies while retaining interpretability. Although we did not evaluate this in our case studies, it represents a promising direction for future work.
- Ensemble reasoning: Combining insights from multiple neurons associated with related concepts can yield stronger, more reliable interpretability than examining neurons in isolation.
- Iterative refinement: Practitioners can iteratively retrain models while monitoring changes in neuron-concept associations, enabling targeted adjustments to improve reliability and fairness.

6.4. Checklist for Practitioners

As a practical reference, the following checklist summarizes the steps for applying this method:

1. Select dataset and train a suitable model (CNN for images, LSTM for text, or others depending on modality).
2. Extract neuron activations from mid-to-late layers and construct positive and negative sets based on thresholds.
3. Apply a Concept Induction tool (e.g., ECII for images, WordNet mapping for text) to assign candidate labels.
4. Evaluate concept labels through modality-appropriate methods (image retrieval or text generation) and validate statistically.
5. Document neuron-concept mappings and analyze patterns for interpretability, bias detection, or debugging purposes.
6. Start with a small subset of neurons to pilot the workflow, then scale gradually; this helps manage compute and ensures insights remain interpretable.

This structured workflow ensures that practitioners can systematically apply the method while adapting it to the resources and requirements of their specific domain.

7. Conclusion and Future Directions

This chapter presented a tutorial-style framework for concept-based hidden neuron activation analysis, combining neural network activations with symbolic reasoning to assign human-understandable labels. By applying the method to both image (SUN2012) and text (AG News) datasets, we demonstrated its adaptability across modalities. The results confirmed that neurons frequently align with coherent, semantically meaningful concepts, and that statistical validation can separate genuine associations from noise. Together, these findings illustrate the value of neurosymbolic approaches for enhancing the interpretability of deep learning models.

7.1. Summary of Contributions

The main contributions of this chapter were:

- Introduced a reproducible pipeline for labeling hidden neurons with symbolic concepts, grounded in structured knowledge sources.
- Demonstrated the approach on two distinct modalities—vision and language—highlighting its generalizability.
- Showed how evaluation techniques such as external image retrieval and synthetic text generation, combined with statistical testing, provide rigor in validating neuron-concept mappings.
- Provided practical guidelines for practitioners, including when to use the method, how to select evaluation strategies, and how to integrate findings into model debugging.
- Positioned the method as complementary to feature-attribution techniques, offering finer-grained, neuron-level explanations.

7.2. Future Directions

Building on these foundations, we foresee several promising directions for research and application:

- **Broader datasets and architectures:** Extending the method to multimodal datasets and large-scale architectures (e.g., transformers, vision-language models) will test its scalability and generality. Readers attempting this should expect larger compute costs and may need to apply neuron sampling strategies.

- **Advanced induction methods:** Exploring hybrid approaches that combine ontology-based induction with large language models may yield more flexible and human-aligned concept labels.

- **Interactive tools:** Developing visualization platforms where users can query neurons and view associated concepts in real time would make the approach more accessible to practitioners. Such tools would also help integrate neuron-level analysis into everyday model debugging workflows.

- **Bias and fairness auditing:** Applying neuron-level labels to systematically detect unintended correlations in sensitive domains such as healthcare, law, or finance could enhance trust and accountability. This use case is especially suited for practitioners who must provide regulatory explanations.

- **Error-margin and reliability analysis:** Incorporating precision-oriented metrics, such as error-margin evaluation, will strengthen confidence in neuron labels for real-world deployment. Readers replicating the method should consider adding these metrics early, even in pilot studies.

Concept-based hidden neuron analysis exemplifies how neurosymbolic AI can bridge the gap between opaque neural computations and human-understandable reasoning. Grounding neuron behavior in symbolic knowledge and validating with statistical rigor, the method offers a pathway toward more transparent and trustworthy AI systems. As models grow in size and complexity, reproducible neurosymbolic methods like this one can provide interpretable anchors for practitioners and researchers alike.

8. Declaration on Generative AI

During the preparation of this work, the author(s) used X-GPT-4 for grammar/spelling check and sentence correction. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

9. Acknowledgments

This work was funded in part by the Kansas State University Game-changing Research Initiation Program (GRIP), the Kansas State University GRIPex: AI in the Disciplines program, and U.S. National Science Foundation awards 2119753 and 2333782. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or Kansas State University.

References

Angelov, P.P., Soares, E.A., Jiang, R., Arnold, N.I., Atkinson, P.M., 2021. Explainable artificial intelligence: An analytical review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 11, e1424.

Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al., 2020. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information fusion 58, 82–115.

Aysel, H.I., Cai, X., Prugel-Bennett, A., 2025. Explainable artificial intelligence: Advancements and limitations. Applied Sciences (2076-3417) 15.

Bengio, Y., Simard, P., Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks 5, 157–166.

Berners-Lee, T., Hendler, J., Lassila, O., 2023. The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities, in: Linking the world's information: Essays on Tim Berners-Lee's invention of the World Wide Web, pp. 91–103.

Bird, S., Klein, E., Loper, E., 2009. Natural Language Processing with Python. 1st ed., O'Reilly Media, Inc.

Dalal, A., 2024. Understanding Hidden Neuron Activations Using Structured Background Knowledge and Deductive Reasoning. Kansas State University.

Dalal, A., Rayan, R., Barua, A., Vasserman, E.Y., Sarker, M.K., Hitzler, P., 2024. On the value of labeled data and symbolic methods for hidden neuron activation analysis, in: International Conference on Neural-Symbolic Learning and Reasoning, Springer. pp. 109–131.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org.

Graves, A., 2012. Supervised sequence labelling with recurrent neural networks. volume 385 of *Studies in Computational Intelligence*. Springer.

Graves, A., Fernández, S., Schmidhuber, J., 2005. Bidirectional LSTM networks for improved phoneme classification and recognition, in: International conference on artificial neural networks, Springer. pp. 799–804.

He, K., Zhang, X., Ren, S., Sun, J., 2016a. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

He, K., Zhang, X., Ren, S., Sun, J., 2016b. Identity mappings in deep residual networks, in: European conference on computer vision, Springer. pp. 630–645.

Hitzler, P., 2021. A review of the semantic web field. Commun. ACM 64, 76–83. doi:10.1145/3397512.

Hitzler, P., Krötzsch, M., Rudolph, S., 2010. Foundations of Semantic Web Technologies. Chapman and Hall/CRC Press. URL: http://www.semantic-web-book.org/.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9, 1735–1780.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., et al., 2021. Knowledge graphs. ACM Computing Surveys (CSUR) 54, 1–37.

Karpathy, A., 2015. The unreasonable effectiveness of recurrent neural networks. https://karpathy.github.io/2015/05/21/rnn-effectiveness/. Blog post, accessed [9/14/2025].

Kim, Y., 2014. Convolutional neural networks for sentence classification, in: Moschitti, A., Pang, B., Daelemans, W. (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar. pp. 1746–1751. doi:10.3115/v1/D14-1181.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. nature 521, 436–444.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 2002. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324.

Lehmann, J., Hitzler, P., 2010. Concept learning in description logics using refinement operators. Mach. Learn. 78, 203–250. doi:10.1007/s10994-009-5146-2.

Levenshtein, V.I., 1975. On the minimal redundancy of binary error-correcting codes. Inf. Control. 28, 268–291. doi:10.1016/S0019-9958(75)90300-9.

Lipton, Z.C., 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue 16, 31–57.

Lundberg, S.M., Lee, S.I., 2017. A unified approach to interpreting model predictions, in: 31st Conference on Neural Information Processing Systems (NIPS 2017).

Luo, W., Li, Y., Urtasun, R., Zemel, R., 2016. Understanding the effective receptive field in deep convolutional neural networks. Advances in neural information processing systems 29.

McKnight, P.E., Najab, J., 2010. Mann-Whitney U test, in: The Corsini Encyclopedia of Psychology. Wiley.

Miller, G.A., 1995. WordNet: A lexical database for English. Commun. ACM 38, 39–41. doi:10.1145/219717.219748.

Mohsen, S., Elkaseer, A., Scholz, S., 2021. Industry 4.0-oriented deep learning models for human activity recognition. IEEE Access PP. doi:10.1109/ACCESS.2021.3125733.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted Boltzmann machines, in: Proceedings of the 27th international conference on machine learning (ICML-10), pp. 807–814.

Nazir, M.A., Evangelista, E., Bukhari, S.M.S., Sharma, R., 2025. A survey of feature attribution techniques in explainable AI: taxonomy, analysis and comparison. Annals of Mathematics and Computer Science 28, 115–126.

Olah, C., Mordvintsev, A., Schubert, L., 2017. Feature visualization. Distill doi:10.23915/distill.00007.

Ponzetto, S.P., Strube, M., 2007. An API for measuring the relatedness of words in Wikipedia, in: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pp. 49–52.

Ribeiro, M.T., Singh, S., 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, pp. 1135–1144.

Saranya, A., Subhashini, R., 2023. A systematic review of explainable artificial intelligence models and applications: Recent developments and future trends. Decision analytics journal 7, 100230.

Sarker, M.K., Hitzler, P., 2019a. Efficient concept induction for description logics, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 3036–3043.

Sarker, M.K., Hitzler, P., 2019b. Efficient concept induction for description logics. Proceedings of the AAAI Conference on Artificial Intelligence 33, 3036–3043. doi:10.1609/aaai.v33i01.33013036.

Sarker, M.K., Schwartz, J., Hitzler, P., Zhou, L., Nadella, S., Minnery, B.S., Juvina, I., Raymer, M.L., Aue, W.R., 2020. Wikipedia knowledge graph for explainable AI, in: Villazón-Terrazas, B., Ortiz-Rodríguez, F., Tiwari, S.M., Shandilya, S.K. (Eds.), Proceedings of the Knowledge Graphs and Semantic Web Second Iberoamerican Conference and First Indo-American Conference (KGSWC), Springer. pp. 72–87. doi:10.1007/978-3-030-65384-2_6.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 618–626. doi:10.1109/ICCV.2017.74.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations (ICLR). URL: https://arxiv.org/abs/1409.1556.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818–2826.

Wang, L., Wang, C., Li, Y., Wang, R., 2021. Explaining the behavior of neuron activations in deep neural networks. Ad Hoc Networks 111, 102346.

Wang, X., Cucala, D.J.T., Grau, B.C., Horrocks, I., 2023. Faithful rule extraction for differentiable rule learning models, in: The Twelfth International Conference on Learning Representations.

Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A., 2010. SUN database: Large-scale scene recognition from abbey to zoo, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3485–3492.

Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer. pp. 818–833.

Zhang, X., Zhao, J., LeCun, Y., 2015. Character-level convolutional networks for text classification. Advances in neural information processing systems 28.

Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A., 2019. Semantic understanding of scenes through the ADE20K dataset. International Journal of Computer Vision 127, 302–321.