Vision by Logic: a Naïve Visual Recognition Model with Non-Axiomatic Logic

Bowen Xu[1,3][0000−0002−9475−9434], Boyang Xu[2][0009−0004−8194−1343], and Pei Wang[1][0000−0002−1066−0454]

[1] Department of Computer and Information Sciences, Temple University, USA
[2] Technical University of Denmark, Lyngby, Denmark
[3] bowenxu.agi@gmail.com

Abstract. Logic-based systems were commonly considered unsuitable for perceptual tasks. In this work, we explore an alternative approach in which visual recognition is treated as a process of logical inference. Building on Non-Axiomatic Logic (NAL), a logic for reasoning under uncertainty, we represent objects as compositional concepts defined by parts and their spatial relations, while local observations provide graded evidential support for competing hypotheses. Recognition is thus formulated as an incremental process of evidence accumulation and belief revision, unifying perceptual processing and logical reasoning within a single framework. To demonstrate the feasibility of this perspective, we construct a minimal model, the Convolutional Concept Network (CCN), which combines convolution-like operations with NAL-based inference. The model maintains explicit, interpretable intermediate representations. Experiments on MNIST show that the system learns hierarchical concepts, suggesting that logic can be applied to visual recognition effectively.

Keywords: Visual Recognition · Non-Axiomatic Logic · Convolutional Concept Network.

1 Introduction

Vision has long been a central topic in artificial intelligence and related fields. A wide range of approaches have been developed for visual recognition, modeling the problem from different perspectives.

In this work, we study vision from a logical perspective, formulating visual recognition as a process of logical reasoning. From this viewpoint, objects are represented as concepts composed of parts and their relative relations, while local observations provide evidence for competing hypotheses. The system integrates such evidence and incrementally updates its beliefs, forming an interpretation of the observation. In this way, the transformation from pixels to objects is expressed within a unified representational and inferential framework.

The idea of vision as a logical process has important precedents in psychology. Helmholtz described vision as "unconscious inference", arguing that perception

draws conclusions about the most likely external cause from fragmentary sensory input. [5] Rock likewise proposed that perception is a process of intelligent problem-solving that follows a logic similar to higher-level thought. [4] These support the view that perception itself can be understood as a form of reasoning.

To formalize this perspective, we adopt Non-Axiomatic Logic (NAL) [7] as the underlying basis. NAL is a form of logic designed for reasoning under conditions of insufficient knowledge and resources, where conclusions are supported and revised based on accumulated evidence. Within this framework, perceptual processing and logical inference can be formulated in a unified manner.

Although symbolic methods have traditionally been associated with high-level cognitive processing and are often considered unsuitable for low-level perceptual tasks, systems grounded in NAL should not be understood as symbolic in the conventional sense. NAL adopts *experience-grounded semantics* [6], in which the meaning of a concept depends on its relations to other concepts and evolves through interaction. From this perspective, a NAL-based system can be viewed as a *concept network* [10], where both concepts and their relations are dynamically updated.

We construct a minimal visual model, the *Convolutional Concept Network* (CCN). The model combines convolution-like operations with logical inference: local regions are matched in a spatially distributed manner, while the matching results are represented, combined, and revised using NAL. All intermediate representations are expressed in logical form, making each processing step interpretable and allowing recognition outcomes to be traced back to their contributing components and spatial structure.

This work does not aim to compare with or improve upon existing methods, nor to compete with highly optimized deep neural networks, but instead explores visual recognition from a different theoretical foundation, demonstrating the feasibility of applying Non-Axiomatic Logic to visual recognition.

2 Basis

This section introduces the conceptual foundations underlying the proposed model. We first summarize the NAL-based account of object recognition that provides the theoretical foundation of the paper. We then describe how image pixels are encoded as primitive concepts and give a simple edge-detection example to illustrate how part-whole object representations can already operate at the level of basic visual features.

2.1 Preliminaries: a Brief Introduction to NAL

"Logic" in NAL is about the laws (or rules) of valid inference (or reasoning), or the "laws of thought". A *concept* is a the core unit for organizing knowledge. Each concept is identified by a term, like *bird*, and *animal*. Relations between/among concepts are represented by *statement*. The simplest form of *statement* in NAL

is *inheritance statement*, written as $S \to P$, which can be informally understood as "S is a specialization of P" and "P is a generalization of S". For example, $bird \to animal$ indicates that the concept *bird* is included in the concept *animal* (to a degree). NAL can express *instance*, denoted by $\{S\}$, and *property*, denoted by $[P]$. For example, $\{apple1\} \to [red]$ means "*the* apple is red". NAL also includes other types of relations, such as *implication*, which expresses conditional dependence between statements, though the present work mainly relies on inheritance.

Unlike classical logic, where meaning and truth are defined with respect to a fixed knowledge-base, NAL adopts *experience-grounded semantics*, in which both the meaning of concepts and the evaluation of statements are determined by accumulated experience and may change over time. Accordingly, each statement is associated with a *truth-value* that reflects the system's current assessment based on available evidence. Rather than being binary, a *truth-value*, denoted as $\langle f; c \rangle$, consists of two components: a *frequency*, indicating how often the statement is supported by evidence, and a *confidence*, indicating how reliable this estimate is. Formally, given positive w^+ and negative evidence w^-, as well as total evidence w, frequency is defined as $f = w^+/w$, and confidence is defined as $c = w/(w+k)$ (where k is a constant and usually $k = 1$). For comparison and decision-making, the numbers in $\langle f; c \rangle$ can be merged into a single scalar called *expectation*, defined as

$$e = 0.5 + c \cdot (f - 0.5), \tag{1}$$

which provides an overall measure of plausibility.

Reasoning in NAL proceeds by combining and revising statements according to available evidence. Since knowledge is incomplete and resources are limited, conclusions are tentative and subject to revision when new evidence becomes available. In this sense, inference in NAL can be understood as a process of evidence accumulation and belief updating. In this paper, the following inference rules are involved.

Name	Rule	Truth-Value Function
Deduction	$M \to P,\ S \to M \vdash S \to P$	$f = f_1 f_2;\ c = f_1 f_2 c_1 c_2$
Induction	$M \to P,\ M \to S \vdash S \to P$	$w^+ = f_1 f_2 c_1 c_2;\ w = f_2 c_1 c_2$
Abduction	$P \to M,\ S \to M \vdash S \to P$	$w^+ = f_1 f_2 c_1 c_2;\ w = f_1 c_1 c_2$

Additionally, when the same statement gets two different truth-values, the *revision* rule can be applied to merge them into one truth-value ($w^+ = w_1^+ + w_2^+;\ w^- = w_1^- + w_2^-$), or the *choice* rule can be applied to choose one of them as the output. [7]

2.2 Object Recognition by NAL

Recent work by Xu and Wang [11] treats an object not as a pre-given external entity, but as a constructed concept that summarizes relatively stable relations among parts. Formally, an object prototype P is represented as a set of part–whole beliefs $P \to [C_i](p_i)$, where C_i denotes a component concept and p_i denotes its position relative to the other parts. In this representation, the identity

of an object is carried not by any single feature alone, but by the organized pattern of features and their relative positions.

When an image instance S provides evidence for one of the parts ($\{S\} \rightarrow [C_i](p_i)$), and an object prototype also include the part ($P \rightarrow [C_i](p_i)$), the system obtains evidence that the whole object may be present as well ($\{S\} \rightarrow P$) according the the abduction rule [11]. With more parts present or absent, the truth-value of $\{S\} \rightarrow P$ is thereby updated using the revision rule Recognition is therefore an evidential accumulation process: multiple local observations contribute graded support to candidate object hypotheses, and the accumulated support is revised as more evidence arrives. This view is especially attractive for perception because it naturally handles partial evidence, uncertainty, and missing parts. This formulation also introduces spatial tolerance in a principled way. Parts with the same type but nearby relative positions can be treated as similar [11], so recognition need not rely on exact pixel-perfect coincidence.

The present paper adopts this part-based, position-sensitive account as the theoretical foundation of the model. A prototype stores its parts and their relative positions, the forward pass accumulates evidence from observed parts to candidate wholes, and learning revises the prototype according to the evidence supplied by new experience.

2.3 Image Encoding

To use the above formalism on images, each pixel must first be converted into primitive concept(s). This step serves as the interface between raw sensory values and logical object recognition. Rather than feeding continuous pixel intensities directly into the object model, the system quantizes them into a small vocabulary of concept types and attaches them to spatial positions.

Figure 1 illustrates the general idea. For a gray-scale image, intensity can be represented on a black-white axis. In the simplest setting, the pixel is binarized into the concepts *black* and *white*; more generally, the axis can be discretized into several intermediate levels. For color images, the same idea extends to channels such as red/non-red, green/non-green, and blue/non-blue. The lower part of the figure shows that familiar colors can be interpreted as combinations of channel concepts: for example, white activates red, green, and blue together, whereas black corresponds to their complements.

In the object formalism, a pixel value yields beliefs of the form $\{\#\} \rightarrow P$, where P is the quantized concept assigned to that pixel, and $\{\#\}$ is an anonymous instance generated by the system. For the current implementation, the encoder uses the minimal black/white setting, so each pixel becomes one primitive instance written into the feature-map.

Figure 2 further shows that the encoded concepts may be equipped with graded similarities. Discrete levels closer to white, for example, can have a stronger similarity to the white concept than darker levels. Such similarities are useful because perceptual evidence is rarely exact. They provide a natural route for generalization: a prototype learned from one intensity level may still partially match a nearby level, allowing recognition to remain robust under small

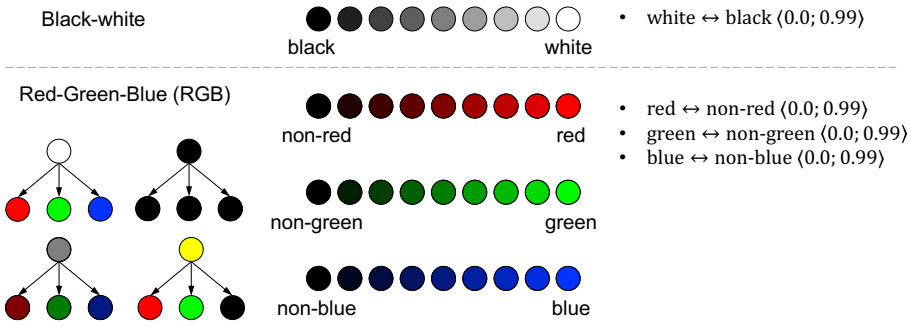

Fig. 1. Examples of concept-level encodings for gray-scale and RGB pixels. Gray-scale values are arranged on a black–white axis, while color pixels are decomposed into channel-wise conceptual oppositions such as red/non-red, green/non-green, and blue/non-blue.

changes in illumination or quantization. In the present paper we use a deliberately simple binary encoding to isolate the central logical mechanism, while the figures illustrate a more general way of conceptually encoding color that leaves room for future extensions.

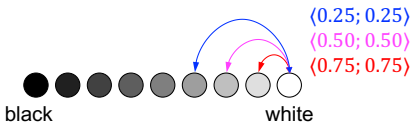

Fig. 2. Similarity among discretized encoding levels. Neighboring levels on the black–white axis can be treated as conceptually similar to different degrees, which later enables tolerant matching rather than exact equality only.

2.4 Example: Edge Detection

To illustrate, we consider a small edge-detection example. The purpose is not to introduce a separate algorithm, but to show how even a classical low-level visual feature can be expressed as object recognition in the NAL sense. Each edge prototype is defined as a sparse arrangement of black and white components within a 3×3 receptive field.

Figure 3 shows two hand-specified prototypes, *Prewitt*-0 and *Prewitt*-90, rotated by 90° relative to one another. Each prototype contains only the informative positions: one side of the patch is expected to be black and the opposite side white, while the dashed cells are left unspecified. In other words, the detector is

represented not as a numeric filter kernel, but as a set of high-confidence beliefs of $P \rightarrow [black](p)$ and $P \rightarrow [white](p)$.

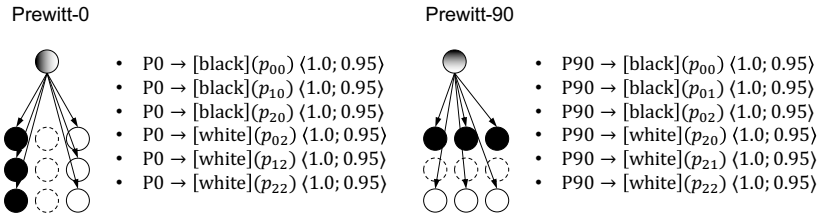

Fig. 3. Two oriented edge prototypes represented as sparse concept arrangements. Filled circles indicate explicitly modeled black or white components, and dashed circles denote positions left unspecified in the prototype.

When such prototypes are scanned over an image, a local patch supports whichever edge concept best explains its pattern of black and white evidence. Figure 4 (a) shows the responses of the two prototypes on a handwritten digit, where the intensity at each location represents the *expectation* of the truth-value of the corresponding match. The two maps emphasize complementary oriented structures: one responds strongly to one dominant local contrast orientation, and the other to the orthogonal orientation. This small example illustrates the general strategy of the model. A primitive edge is already treated as an object-like concept defined by parts and relative positions; higher-level shapes can then be learned compositionally from these lower-level edge concepts in exactly the same manner.

More concretely, as illustrated in Fig. 4 (b), each component in the receptive field contributes local abductive evidence to the hypothesis $\{S\} \rightarrow P$: when a component in the observation agrees with that specified in the prototype, it yields a positive contribution (*e.g.*, $P \rightarrow [black](p_{00})\ \langle 1; 0.95\rangle$ and $\{S\} \rightarrow [black](p_{00})\ \langle 1; 0.95\rangle$ derive $\{S\} \rightarrow P\ \langle 1; 0.47\rangle$), while mismatched or unsupported components provide negative (*e.g.*, $P \rightarrow [white](p_{22})\ \langle 1; 0.95\rangle$ and $\{S\} \rightarrow [white](p_{22})\ \langle 0; 0.95\rangle$ derive $\{S\} \rightarrow P\ \langle 0; 0.47\rangle$) or null contributions (*e.g.*, $P \rightarrow [black](p_{11})\ \langle 1; 0.95\rangle$ and $\{S\} \rightarrow [black](p_{11})\ \langle 0; 0\rangle$ derive $\{S\} \rightarrow P\ \langle 0; 0\rangle$). After all components are evaluated, these truth-values are combined through the *revision* rule in NAL, resulting in a single truth-value for the match (*e.g.*, $\{S\} \rightarrow P\ \langle 0.83; 0.84\rangle$).

3 Pragmatic Simplifications

The model presented in this paper adopts several simplifying assumptions to isolate the core logical mechanisms of perception. These assumptions are introduced to make the problem feasible in the current implementation.

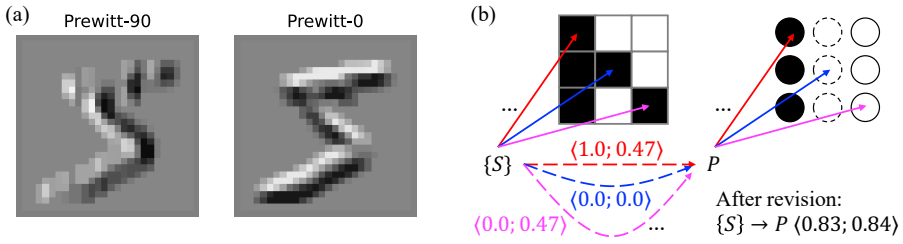

Fig. 4. (a) Responses of the two oriented prototypes on a sample digit, showing how local contrast patterns are turned into orientation-specific concept activations. (b) Illustration of component-wise matching, where local abductive evidence from individual parts is combined through revision to produce the final recognition result.

First, we consider the problem of visual recognition as follows: given a large set of sensory observations distributed over space, and a large set of candidate prototypes (each defined by a set of components with relative positions), the system must determine which prototype best explains the observations at each location. In general, this matching problem is challenging not merely due to its computational cost, but because it must be performed under resource constraints. In realistic settings, the system does not have sufficient time to evaluate all possible matches. Instead, decisions often need to be made based on partial evidence, before a complete set of candidate explanations can be considered. Moreover, visual perception is inherently coupled with sensorimotor processes [3], but the current model does not consider such processes and only handles passive observation.

Furthermore, learning introduces an additional challenge: the system must decide which prototypes to revise, and based on which new observations. This involves a dynamic allocation of computational resources, as well as a mechanism for selecting informative experiences. In a general setting, such processes are expected to rely on heuristic strategies and attentional control, allowing the system to focus on relevant regions of the input and relevant candidate concepts over time.

In the present work, we adopt a significantly simplified strategy. Prototype matching is performed exhaustively: at each spatial location, the system evaluates all candidate prototypes. This procedure is functionally analogous to the convolution operation in deep learning. In addition, we assume that each input is given sufficient observation time. That is, all candidate explanations are evaluated before a final decision is made. After the accumulation of evidential support, a competition mechanism is applied to select the best explanations, which are then used to revise the corresponding prototype. This assumption removes the need to address time-pressure and incremental decision-making, which remain important topics for future work.

Finally, we assume that spatial structure is given *a priori*. The relative positions of components are directly available to the system and are used as part

of the object representation. While this assumption is common in many vision models, it is not fundamental. As discussed in [11], spatial sense itself may be learned from sensorimotor experience, and integrating such mechanisms into the present framework is an important direction for future research.

4 Convolutional Concept Network

We now describe the implemented *Convolutional Concept Network* (CCN), a minimal visual model that combines the convolutional operation with Non-Axiomatic Logic.

CCN represents knowledge through structured, interpretable prototypes. Matching a receptive field against a prototype yields a truth-value, and learning refines prototype structure directly. Generally, CCN alternates between two routines: a forward operation and a learning operation. In the forward routine, each local input patch is matched against a bank of prototypes, producing a new feature map of concept instances. In the learning routine, the winning prototypes are revised using the local evidence that supported their recognition. Perception therefore unfolds as a repeated cycle of *accommodation* and *assimilation* as termed by Piaget [2].

Encoding. The input image is first converted into a spatially distributed set of observations of the form $\{\#\} \rightarrow P$: For a gray-scale image, each pixel value at location p_j is quantized into a discrete sensory type. In the simplest setting used here, the encoder uses two sensory prototypes, $P_0^{(0)} = \textit{black}$ and $P_1^{(0)} = \textit{white}$, although the model can, in principle, be extended to support a larger number of quantization levels. For each location, the system creates a temporary anonymous instance with a type and a truth-value, so each observation has the form $\{\#\} \rightarrow P_i^{(0)}\ \langle f; c\rangle$, which is stored into a cell, of a feature map, corresponding to position p_j.

Feature maps and prototypes. The observations from encoded image as well as a lower layer are stored in a *feature map*, a two-dimensional grid whose cells correspond to spatial locations in the image. Rather than storing a single value, each cell maintains a set of candidate instances, since the input at a location may match multiple prototypes simultaneously. Each match gives rise to an anonymous instance together with its type, with the form $\{\#\} \rightarrow P$. An observation from a lower layer, $\{\#^{(l-1)}\} \rightarrow P$, is converted to a component of an instance observed in the current layer, *i.e.*, $\{\#^{(l)}\} \rightarrow [P](p)$, so that abduction can be applied with it and a prototype in layer l as premises.

An object prototype in layer l is represented as a set of typed components anchored to relative positions within a finite receptive field. Formally, a prototype $P_r^{(l)}$ in layer l is composed of statements of the form

$$P_r^{(l)} \rightarrow [P_i^{(l-1)}](p_j)\ \langle f; c\rangle\ , \tag{2}$$

meaning that the prototype expects a component of type $P_i^{(l-1)}$ at relative position p_j in layer $l-1$. Each component carries a truth-value indicating how strongly and how confidently that component belongs to the prototype. Consequently, a prototype is not a rigid template, but an arrangement of parts with graded evidential support.

A prototype stores only its occupied components up to a fixed capacity, reflecting a resource constraint that prevents unbounded growth. When this capacity is reached, the weakest component is discarded when inserting a new one, allowing the prototype to remain adaptive while maintaining a bounded size.

Prototype matching. Prototype matching should first be understood at the level of a *single* concept hypothesis. Given a feature-map from layer $l-1$, the model slides one prototype over the map using a convolution-like schedule defined by a kernel shape, stride and padding. At every output position, the local input patch is interpreted as evidence for an anonymous instance in layer l, and the prototype is evaluated as a possible explanation of that patch.

Matching is performed component-wise. Suppose the current patch contains an observed feature $\{\#\} \to [P_i^{(l-1)}](p_j)$ and the prototype contains the statement $P_r^{(l)} \to [P_i^{(l-1)}](p_j)$. Their agreement supports the abductive conclusion that the anonymous local instance belongs to the prototype:

$$P_r^{(l)} \to [P_i^{(l-1)}](p_j),\ \{\#\} \to [P_i^{(l-1)}](p_j) \vdash \{\#\} \to P_r^{(l)}\ \langle F_{\text{abd}} \rangle\ . \tag{3}$$

The total match between a patch and a prototype is then obtained by revising the abductive evidence accumulated across all occupied component locations in the receptive field.

This single-prototype matching process is graded rather than exact: the result is a truth-value, not a binary decision. Moreover, the implementation allows missing or mismatched local evidence to contribute negative support rather than causing the entire match to fail abruptly. The present or absence of an anonymous instance is therefore evaluated by the balance of positive and negative evidence across its receptive field, quantified by truth-value.

Prototype learning. Prototype learning is likewise easiest to define first for a *single winning concept*. Once a winner $\{\#\} \to P_r^{(l)}$ has been selected, the corresponding prototype is revised using the input patch that supported it. For each observed component in the lower layer, the model performs induction,

$$\{\#\} \to [P_i^{(l-1)}](p_j),\ \{\#\} \to P_r^{(l)} \vdash P_r^{(l)} \to [P_i^{(l-1)}](p_j)\ \langle F_{\text{ind}} \rangle. \tag{4}$$

If the corresponding component already exists in the prototype, its truth-value is revised; otherwise, a new component is inserted. In this sense, learning is not parameter optimization in a continuous vector space, but the incremental revision of an explicit part-based hypothesis.

The implementation also incorporates negative evidence within a winning prototype. When the winner confirms one component type at a relative position, alternative component types already stored at the same position receive negative inductive support. For example, when $\{\#\} \to [P_0^{(1)}](p_0)$ is observed, and both $P_0^{(2)} \to [P_0^{(1)}](p_0)$ and $P_0^{(2)} \to [P_1^{(1)}](p_0)$ are contained in prototype $P_0^{(2)}$, a negative observation $\{\#\} \to [P_1^{(1)}](p_0)$ $\langle 0.0; 0.5\rangle$ is generated, and *induction* is then applied to generate a negative belief of $P_0^{(2)} \to [P_1^{(1)}](p_0)$. Through negative observation, the system strengthens not only what tends to be present, but also what tends to be absent.

Competition across concepts. Inspired by [1], we introduce the competition mechanisms in matching and learning. The single-prototype mechanisms described above are embedded in a second level of organization: competition among multiple concept hypotheses. In the forward pass, the same local patch is matched against the entire bank of prototypes at the current layer. Several candidate concepts may therefore be supported at the same output position. Rather than collapsing these responses immediately, the model stores all matched candidates at that position and ranks them by *expectation* of truth-value.

This ranking implements *point-wise inhibition*. Multiple concepts may compete to explain the same local evidence, but only the strongest explanation remains active at that location. The lower-ranked candidates are not discarded; they are preserved as latent alternatives. The winners are used for next layer's recognition.

Competition is extended further during learning. After the forward pass, the system repeatedly selects the strongest candidate in the whole feature-map and then applies a lateral inhibition window around that winner, suppressing nearby candidates before the next selection step. The result is a set of spatially distributed winners. Consequently, only a coherent subset of concept instances is allowed to revise its prototypes at each learning step. This procedure combines local evidence accumulation with inter-concept competition and plays the role of a winner-take-all mechanism over both concept identity and spatial location.

In a nutshell, the matching routine decides which concepts are plausible explanations for each patch; the learning routine decides which of those explanations are sufficiently strong and sufficiently non-redundant to drive learning.

Pipeline. A sensory image is first quantized into primitive concept instances and written into the input feature-map. For a given layer, the prototype bank is then scanned over the incoming feature-map by convolution, producing a ranked list of candidate concepts with the form $\{\#\} \to P$ at each location. In the learning routine, winner selection and lateral inhibition determine which instances are allowed to revise the prototype bank. Point-wise competition keeps only the strongest candidate active, and they are passed upward as the representational output of the current layer.

The resulting architecture is naturally hierarchical. The sensory interface produces a feature-map of primitive sensory types. A first learned layer can then acquire local prototypes corresponding to simple patterns such as oriented edges or corners. A subsequent layer treats those lower-level recognitions as components and learns larger compositions. The same representational scheme is reused across layers: a higher-level concept is always represented as an arrangement of lower-level concepts with explicit relative positions and truth-values.

5 Experimental Results

Experiments are conducted on the MNIST handwritten-digit dataset. The purpose of this section is not primarily to optimize classification accuracy, but to examine whether the proposed hierarchical learning process acquires meaningful intermediate concepts. To provide a simple quantitative check, we attach a linear classifier (SVM) to the learned representation. We train the first layer using 10,000 images, then train the second layer and the SVM classifier using 20,000 images. During layer-2 training, the first layer is frozen; during classifier training, both learned layers are frozen. The final system is evaluated on the full test set. Under this protocol, the model reaches 99.3% training accuracy and 95.5% test accuracy.

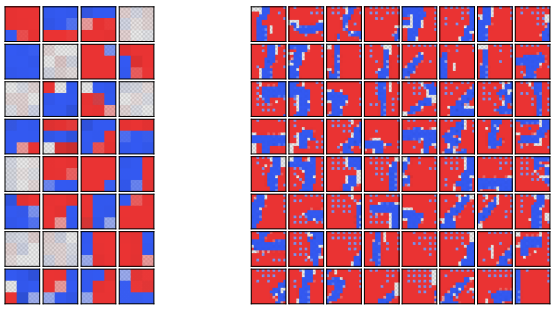

Fig. 5. Learned prototypes in the first two layers. Left: layer-1 prototypes. Each spatial position is colored by the concept type with the highest expectation of truth-value at that location, and the opacity indicates the corresponding *confidence* of truth-value; fully transparent cells denote positions where no concept is retained. Right: layer-2 prototypes, visualized by replacing each layer-1 component with the image of its corresponding layer-1 prototype. The first layer mainly captures local edge-like patterns, while the second layer composes them into larger stroke fragments.

Figure 5 visualizes the learned prototypes in the first two layers. In the first layer, many prototypes resemble familiar local primitives: vertical and horizontal contrasts, and corner-like transitions. Many cells remain unspecified, while the informative cells often carry high confidence. However, the transparency in these visualizations should not be given a single blanket interpretation. Some

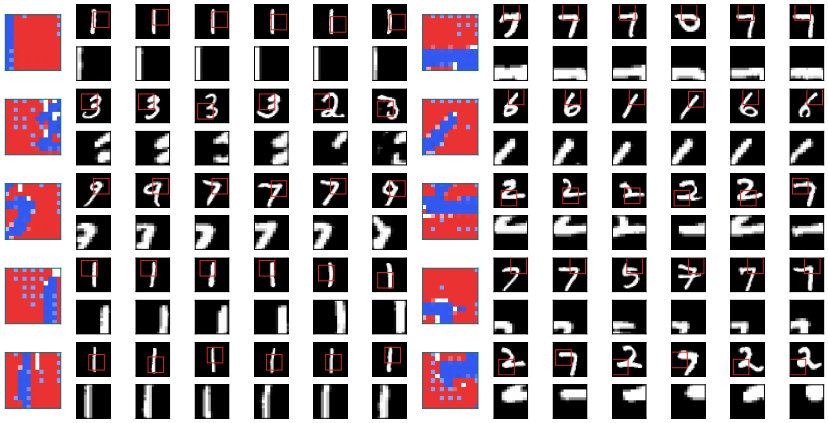

Fig. 6. Top matches of selected layer-2 prototypes. For each prototype, the upper row shows digit images with the matched receptive field marked in red, and the lower row shows the corresponding local patches. The same prototype can recur in different digit classes whenever a similar local stroke configuration is present.

layer-1 prototypes are almost completely transparent because, after receiving tiny random confidences at initialization, they were never selected for revision during training; under the current learning mechanism, these prototypes never won the competition against other candidate prototypes. By contrast, isolated transparent cells inside otherwise learned prototypes have a different origin: the prototype itself was selected and revised, but at those particular positions no component survived the competition against alternative components at other positions.

The right side of Fig. 5 shows the second-layer prototypes by expanding each lower-level component into its corresponding first-layer visualization. Compared with the first layer, these concepts are clearly more structured. Rather than isolated local contrasts, they capture longer line segments. This qualitative change matches the intended hierarchical interpretation of CCN: higher-level concepts are learned as explicit spatial compositions of lower-level concepts.

Figure 6 further illustrates how these second-layer concepts are used in recognition. Each prototype is matched by patches drawn from different digit classes, yet the matched local structures remain visually consistent. Some prototypes recur on near-vertical strokes, others on diagonal segments, horizontal caps, or curved stroke endings. At the same time, the matches are not pixel-identical: variations in thickness, curvature, and surrounding context are tolerated as long as the same local part-whole structure is present. This is exactly the behavior expected from a NAL-based recognition mechanism, where matching is graded evidential support rather than exact template equality.

For reference, Wang et al. (2022) [8] reported 43.66% accuracy on the MNIST dataset. However, classification accuracy alone is not the main point of comparison here. In the present work, the linear classifier serves mainly as a probe

of the learned representation; the central result is that the system acquires reusable middle-level concepts with clear visual interpretation. The two studies are more meaningfully compared in terms of their theoretical choices. The previous work [8] adopts *conjunction* [7] to represent an object, *e.g.*, $(C_1 \wedge C_2 \wedge C_3)$ denotes an object composed of three parts. In that formulation, the absence of a component directly counts as strong evidence against the whole, making partial matches difficult to preserve. In contrast, the *part-whole relation* [11] adopted here allows each component to contribute partial evidence, so the presence or absence of one part is influential but never decisive. This relation is more natural for physical objects in perception and better suited to robust recognition from incomplete local observations.

6 Conclusion & Discussion

In this work, we constructed a naïve visual recognition model based on Non-Axiomatic Logic (NAL), in which the process from pixels to higher-level interpretation is formulated as a series of logical inference steps. The main contribution of the paper is not a direct comparison with highly optimized recognition systems, but a concrete demonstration that visual recognition can be expressed within a logical framework. In the proposed model, objects are represented as compositional concepts, local observations provide evidential support to candidate hypotheses, and recognition emerges from the accumulation and revision of such evidence. The resulting system maintains explicit intermediate representations, so the learned prototypes and their matches remain interpretable throughout the hierarchy.

This work also helps clarify where the main difficulty lies in extending NAL to perception. The challenge is not primarily in knowledge representation or inference rules, but in the construction of invariant concepts from variant sensory experience. At the same time, the present model leaves open the broader problem of how a perceptual system should actively organize its own experience.

From the perspective of AGI, Wang and Hammer [9] proposed that perception should be *subjective*, *active*, and *unified*. However, the current model does not yet satisfy the *active* principle. The system passively receives an image and processes it with a preset scanning procedure; it does not yet decide what to attend to, what information to seek next, or what action to take in order to reduce uncertainty. This limitation is fundamental rather than cosmetic, and future work should address it explicitly.

The current model has several additional limitations. Spatial sense is assumed to be given in advance, though in principle it should itself be learnable [11]. The image encoder is deliberately simplified, using a minimal binary representation so that the logical mechanism can be isolated clearly. The experiments are also limited in scope: the evaluation is on MNIST, and the current use of a simple linear classifier mainly serves as a probe of the learned representation, though classification could in principle be carried out within NAL itself using its native goal-driven backward inference and operation-execution mechanisms, as in

[8]. Similarity relations among prototypes are also not considered, though such relations may affect both matching and resource competition. Most of these limitations stem directly from the simplifying assumptions mentioned in Sec 3. Thus, there are several relatively direct extensions for future work, including adding spatial generalization, introducing prototype generalization through concept similarity, testing on more complex 2D benchmarks, and extending the framework to 3D point-cloud perception.

More generally, the path from the present model to a system with complete perceptual mechanisms involves many further issues that cannot be discussed here in sufficient detail. Among them are the learning of spatial sense, perception under time pressure, scene perception, cognitive mapping, and the tighter interaction of perception with action and other cognitive processes. For this reason, the present paper should be understood as only an early step in a larger research program. A more complete plan for such a perceptual system deserves a separate paper.

Code Availability. The source code for reproducing the experimental results can be found in https://github.com/bowen-xu/logical-visual-recognition.

References

1. Kheradpisheh, S.R., Ganjtabesh, M., Thorpe, S.J., Masquelier, T.: STDP-based spiking deep convolutional neural networks for object recognition. Neural Networks **99**, 56–67 (Mar 2018). https://doi.org/10.1016/j.neunet.2017.12.005
2. Müller, U., Ten Eycke, K., Baker, L.: Piaget's Theory of Intelligence. In: Goldstein, S., Princiotta, D., Naglieri, J.A. (eds.) Handbook of Intelligence: Evolutionary Theory, Historical Perspective, and Current Concepts, pp. 137–151. Springer, New York, NY (2015). https://doi.org/10.1007/978-1-4939-1562-0_10
3. O'Regan, J.K., Noë, A.: A sensorimotor account of vision and visual consciousness. Behavioral and Brain Sciences **24**(5), 939–973 (Oct 2001). https://doi.org/10.1017/S0140525X01000115
4. Rock, I.: The logic of perception. MIT Press (1983)
5. Von Helmholtz, H.: Handbuch der physiologischen Optik, vol. 9. L. Voss (1867), title-translated: Treatise on Physiological Optics
6. Wang, P.: Experience-grounded semantics: a theory for intelligent systems. Cognitive Systems Research **6**(4), 282–302 (Dec 2005). https://doi.org/10.1016/j.cogsys.2004.08.003
7. Wang, P.: Non-Axiomatic Logic: A Model of Intelligent Reasoning. WORLD SCIENTIFIC, e-book, 2 edn. (Sep 2025). https://doi.org/10.1142/14486
8. Wang, P., Hahm, C., Hammer, P.: A Model of Unified Perception and Cognition. Frontiers in Artificial Intelligence **5** (2022)
9. Wang, P., Hammer, P.: Perception from an AGI Perspective. In: Iklé, M., Franz, A., Rzepka, R., Goertzel, B. (eds.) Artificial General Intelligence. pp. 259–269. Lecture Notes in Computer Science, Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-97676-1_25
10. Wang, P., Thórisson, K.R.: Concept-Centered Knowledge Representation: A 'Middle-Out' Approach Fusing the Symbolic-Subsymbolic Divide. In: Robertson, P., Georgeon, O. (eds.) Situated Self-guided Learning. pp. 10–52. Springer Nature Switzerland, Cham (2025). https://doi.org/10.1007/978-3-031-96325-4_2

11. Xu, B., Wang, P.: On the Essence of Spatial Sense and Objects in Intelligence. In: Iklé, M., Kolonin, A., Bennett, M. (eds.) Artificial General Intelligence. pp. 350–360. Springer Nature Switzerland, Cham (Aug 2025). https://doi.org/10.1007/978-3-032-00800-8_31