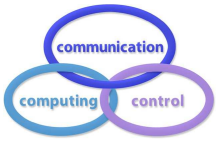
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 19, Issue: 1, Month: February, Year: 2024
Article Number: 6392, https://doi.org/10.15837/ijccc.2024.1.6392



CCC Publications

Signs and Supersigns in Deep Learning

R. Andonie, B. Muşat

Răzvan Andonie
1. Department of Computer Science
Central Washington University, USA
2. Department of Electronics and Computers
Transilvania University of Braşov, Romania
*Corresponding author: razvan.andonie@cwu.edu

Bogdan Muşat
Department of Mathematics and Informatics
Transilvania University of Braşov, Romania
bogdan_musat_adrian@yahoo.com

Abstract

Semiotics is the study of signs and sign-using behavior. Computational semiotics is an interdisciplinary field which proposes a new kind of approach to intelligent systems, where an explicit account for the notion of sign is prominent. Our fundamental thesis is that information concentration processes appear in successive layers of deep learning models: each layer aggregates information from the previous layer of the network. In computational semiotics, this information concentration is known as superization, and it is accompanied by a decrease of entropy: signs are aggregated into supersign. Our interdisciplinary approach enables us to depict superization processes within deep learning models. This is a novel semantic interpretation of deep learning. We use concepts from computational semiotics to explain decision processes in deep learning. Semiotic tools can be used to optimize the architecture of deep neural networks. Interpretability/explainability and architecture optimization of neural models are currently among the hottest topics in machine learning. We illustrate our semiotic approach with several applications. Our contribution can be seen as the initial move in establishing a cohesive semiotic framework for deep learning models.

Keywords: deep learning, computational semiotics, neural network explainability, neural architecture optimization

1 Introduction

Semiotics[1] is the study of signs and sign-using behavior. A *sign* is anything that communicates a meaning, that is not the sign itself, to the interpreter of the sign. Semiotics as an interdisciplinary field of study emerged in the late 19th and early 20th centuries with the independent work of Ferdinand

[1]Derived from the Greek word "semeiotikos" which means interpreter of signs.

de Saussure and Charles Sanders Peirce[2] (Fig. 1). More refined definitions of semiotics can be found in [1, 2, 3]. *Semiosis* is any process that involves signs, including the production of meaning.

A fundamental assumption in semiotics is that signs do not convey a meaning that is inherent to the object being represented. In Peirce's theory of sign, we have an irreducible triadic relation (Fig. 2) between Sign-Object-Interpretant [4]:

> "I define a sign as anything which is so determined by something else, called its Object, and so determines an effect upon a person, which effect I call its interpretant, that the later is thereby mediately determined by the former."

Figure 1: Charles Sanders Peirce (1839–1914) by Unknown author - New York Public Library

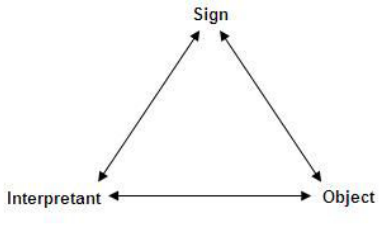


Figure 2: Peirce's theory of signs [4]

Peirce developed several classifications of signs. According to Antony Jappy [5], perhaps the most profound aspect of Peirce's work in semiotics was that Peirce defined in 1903 the means of discovery – inference – and the objects of the processes of discovery – signs – as elements of the same semiotic system, and made them subject to the same constraints and definitions. This is similar to what John von Neumann did for the computer when he suggested that both data and instruction be formulated in the same code.

Peirce argued for semiosis or triadic mediation as the sole source and end of cognition. In contrast, Charles W. Morris attempted to synthesize in semiotics pragmatism with logical positivism [6]. He grouped semiotics into three branches [7]:

- *Syntactics:* relations among or between signs in formal structures without regard to meaning.
- *Semantics:* relation between signs and the things to which they refer; their signified denotata, or meaning.
- *Pragmatics:* relations between the sign system and its human (or animal) user.

Semiotics had an early impact on computer science in the 1960s, with the introduction of the syntactic, semantic, and pragmatic distinctions into programming language theory [8]. Subsequent developments in this intersection are explored in more recent findings presented in [9].

Max Bense tried to apply Peircean semiotics to aesthetics [10]. According to Bense, information and sign processes go ahead of each communication process: information is important in the formation of signs. Bense and Abraham Moles [11] applied Shannon's information theory to aesthetics: From unstructured material successive emergence of structures is achieved by stochastic selections and aesthetic information is transmitted as complex supersigns selected from repertoires of elementary signs. This aesthetics was introduced in the 1960s, as an attempt to establish a mathematically rigorous aesthetic theory without subjective elements. Some of its concepts turned out to be reductionist and schematic, which led to their eventual disappearance [12].

[2] https://www.britannica.com/science/semiotics

However, this theory was continued by others [13, 14]) and created the basis for Generative Aesthetics, in which art could be objective and not externally influenced. In current terms, we call this Generative AI Art. A relatively recent review of informational aesthetics measures based on information theory and Kolmogorov's complexity can be found in [15].

In the context of computational sciences, *computational semiotics* was used as a mathematical framework of concepts from semiotics. Gudwin and Domide [16] stated that semantic networks can implement computational intelligence models (fuzzy systems, neural networks, and evolutionary computation algorithms). Several computational models of Peirce's triadic notion of meaning processes were proposed [17, 18, 19]. Baxter *et al.* introduced a framework for the interpretation of medical image segmentation as a sign exchange in which each sign acts as an interface metaphor [20]. In our endeavors, we employ computational semiotics in the realm of artificial intelligence, emphasizing the significance of the concept of signs.

As we know, there is a huge interest in the field of interpretability and explainability of deep learning models. For many years, we used the black box paradigm interpretation of deep learning. Things have partially changed, and presently we do have techniques to interpret these models [21, 22] or theoretical insights [23, 24], even if we still miss a fundamental theory that can elucidate all underlying aspects. Some of these methods are based on visualization, integrating the benefits of artificial intelligence, machine learning, and visual analytics [25, 26].

Our thesis is that the concept of sign and semiotics offers a tempting fundamental conceptual basis for building, training, and interpreting/explaining neural models. There are very few machine learning models designed by rigorous semiotic principles. To the extent of our knowledge, the only applications of computational semiotics in the analysis and interpretation of deep neural networks are the ones reported by us [27, 28, 29, 30].

Our current contribution is a comprehensive analysis of the semiotic infrastructure used in deep learning. The approach is interdisciplinary, at the intersection of Peirce's theory and its information theory interpretation by Max Bense [10] and Helmar Frank [13]. We also integrate visualization of semiotic deep learning processes. The practical applications focus on the interpretation/explanation of decision processes in Convolutional Neural Networks (CNNs), but also on the architecture optimization of these models.

The paper proceeds as follows. Section 2 introduces the concept of superization, a semiotic aggregation operation that serves as a central element in our framework. Section 3 focuses on the core of our approach, defining superization in deep learning models. In Section 4 we demonstrate with three applications involving CNN learning and architecture optimization, wherein we offer semiotic interpretations of the underlying processes. Section 5 encompasses our concluding remarks.

2 Superization - A Semiotic Aggregation Process

This section introduces a semiotic aggregation procedure known as "superization", which is at the foundation of our methodology.

In semiotics [10, 13, 27], the usual signs designate unconsciously perceived material entities, the so-called *first level signs*. At the next hierarchical level, these signs may be agglomerated into *second level supersigns*. Iterating the process, we obtain from *k-th level supersigns* $(k+1)$-th level supersigns. The transition from k-th level to $(k+1)$-th level supersigns in called *superization* [10, 13, 31, 32, 33].

Helmar Frank [13] considered the following two types of superization:

- **Type I.** By class formation: building equivalence classes and thus reducing the number of signs. The characters within a text can be viewed as first level signs. The collective class encompassing all variations of the letter "a" (handwritten, uppercase, etc.) constitutes a second level supersign.
- **Type II.** By compound formation: building compound supersigns from simpler component supersigns. Revisiting the earlier example, we can derive words from letters, sentences from words, and progressively build more complex and abstract syntactic-semantic structures thereafter.

Let us consider the Shannon entropy computed at two successive layers: H_k and H_{k+1}. The extracted information by the interpretant can be measured by the difference $H_k - H_{k+1}$. Helmar

Frank proved that both types of superization tend to concentrate information by decreasing entropy [13].

This is a particular case of a more general result: A measure-preserving function can map several points to the same point, but not vice versa, so this change in entropy is always a decrease. Since we do not introduce any additional randomness, the entropy can only decrease, and we can talk about the 'information loss' associated with the function [34].

The fact that the entropy decreases is a simplified mathematical result. In a communication model, from an informational psychology perspective, the entropy is not necessarily monotonically decreasing, since an information adaptation process takes place. The entropy increases until it reaches its peak value, a phase linked to the perceiver's adaptation to the information [32, 33]. A subsequent entropy decrease is related to the processing of structural information, and the decrease rate depends on the amount of structural information. The entropy falls quickly when little structural information is available, whereas when major structural information is present, the entropy will remain high over most of its range [32, 33].

Superization is a semiotic aggregation process characterized at each perception level by a specific repertory of supersigns. For example, we may consider a Gaussian or Laplacian image pyramid, a multiresolution image representation obtained by succesive convolution and sampling on each resulting image [35]. If we consider each pixel of an image as a sign at the given hierarchical level, we may find a similarity between this hierarchical aggregative representation and superization. We can derive a resolution-dependent Shannon entropy from the probability distribution of grey-level events observed at that level [36]. Using the newspaper's reading analogy, at the magnified level, where only white and black patches are visible, the entropy H will be low. As the picture is brought to normal focusing distance, a great variety of grey levels become apparent, and consequently, the entropy increases. As the picture is moved further away from the eyes, the entropy decreases (see Fig. 3). Finally, it may become nearly uniformly grey in appearance, with $H \approx 0$.

Figure 3: At the magnified level, where only white and black patches are visible, the entropy is low. It increases up to a maximum (the normal focusing distance and then decreases to 0 as we zoom out.

Superization is not a simple combinatorial process, but subtle syntactic-semantic perception frame related to Peirce's triadic model of semiosis. Moreover, in a semiotic communication process, from an informational psychology view, there are further omitted details.

In the newspaper's reading analogy, the entropy increases until it reaches its peak value. This phase may be associated to the informational adaptation of the perceiver [32, 33]. The subsequent entropy decrease is related to the processing of structural information [36]. The rate of decrease depends upon the amount of structural information in the picture. The entropy falls quickly when little structural information is available, whereas when major structural information is present, the

entropy will remain high over most of its range. The variation of entropy can indicate the type and quantity of structural information in the picture in terms of size and relationships to detailed features. We may associate the peak value of the entropy to one of the most meaningful observations of the picture. However, because of other factors, the maximum entropy is not always associated with the "optimal" resolution [32, 33].

In the image pyramid representation, we omit an important fact pertaining to human vision that reduces the significance of the overcompleteness [35]: The vast majority of the transform coefficients represent information in the highest spatial frequency bands where people have poor visual resolution. Therefore, we can quantize these elements very severely without much loss in image quality.

In the following, we will illustrate semiotic aggregation with an example from generative art [37]. Our goal is to produce geometric structures with aesthetic appeal, similar to the ones in Fig. 4.

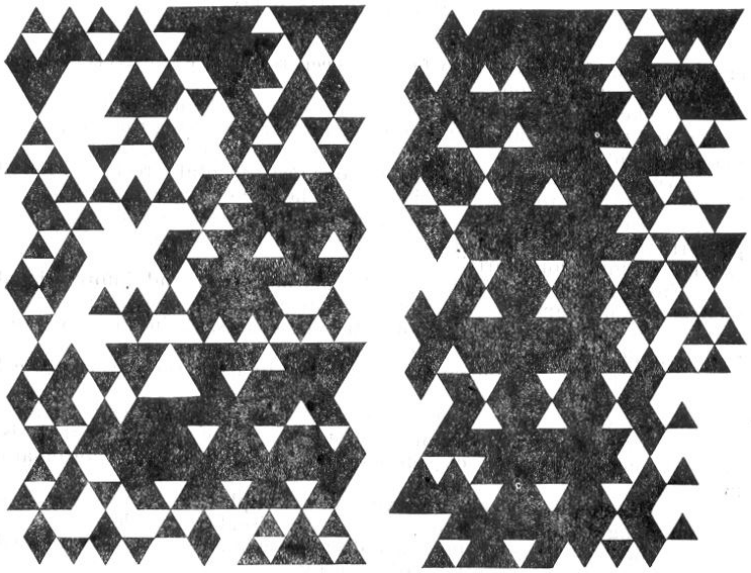


Figure 4: Generated structures [37]

For the generation of geometrical structures, we choose as the basic element the triangle. It is generated as a result of a point-line tension relationship (Fig. 5).

Figure 5: The basic element [37]

The equilateral triangle performs best as the transition from the static state of the square to the dynamism of the circle, and it is the symbol of the active equilibrium, open to external tensions. The five signs chosen as the first level signs, depicted in Fig. 6, results from dividing the triangular field into four internal fields obtained by translation of each side up to the half of the two sides.

Figure 6: The first level signs [37]

An increase from white to black results on the basis of a measurable proportion scale, achieving in this way the conversion of the passive from the white triangle into the active of black one. This

graduation allows to obtain an exact quantitative control of the degrees of tonality between clear and obscure (passive-negative). When comparing the outlines, surface tensions result. The signs have a ternary symmetry and are disposed in a (first generation) grid structure. In the structure, the signs interact locally.

The second level supersigns are obtained by type I superization (by class formation, see Fig. 7). The signs are reduced from five to three according to the maximum fields domination principle (Fig. 8).

Figure 7: The second level signs [37]

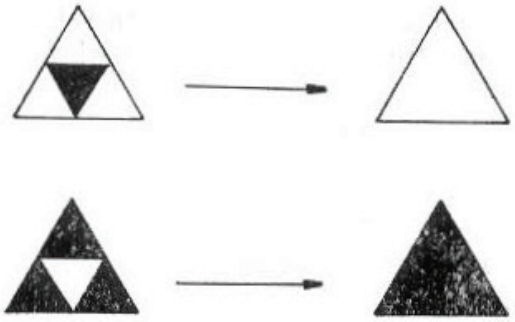


Figure 8: Superization to second level signs [37]

The second generation structure is probabilistically generated by a 2D Markov chain describing the local interactions between the second level signs. The transition matrices used to generate the structure are determined by some specific artistic creative schemes(see [37].

The process continues, and we obtain level three supersigns by superization of level two supersigns, reduced from three to two according to the transition in Fig. 9. This superization is justified by the active tendency of the black field to increase its surface perceived when it lies on a light field. This principle can be verified experimentally.

Figure 9: The third level signs [37]

The third generation structure is generated by a similar 2D Markov chain which models the interactions between third level signs.

For each generated structure we can compute its entropy: H_k is the entropy of the generation k structure, composed of k-th level supersigns. In our experiments, it can be observed that the entropy decreased monotonically, according to our expectation.

This algorithm iterates several times, producing "super...supersigns", etc. A good question is when to stop: When we are left with one supersign and the entropy reaches its minimum value (zero)? From the informational psychology perspective, the process can end when the entropy reaches 160 bits, which is the capacity of the short-term memory of the viewer (the art consumer in this case) [13].

In this generative art application, our goal is to optimize the aesthetic pleasure of the viewer. The subjective "aesthetic pleasure" characterizing the transition between level k and level $k+1$ of supersigns can be quantified by the following quantity [10, 13, 31]:

Figure 10: Superization to third level signs [37]

$$\frac{order(redundancy)}{complexity(entropy)} = \frac{\frac{H_k - H_{i+1}}{H_k}}{H_{k+1}}$$

The transition to a higher supersign level is a called a Birkhoff[3] synthetic process. It is accompanied by information reduction of the contemplated art object (from the viewer's perspective). The synthetic process is characterized by the superization phase. The synthetic process is followed by an analytical process which takes us back to inferior order supersigns. The inverse transition is called a Moles[4] analytical process.

When perceiving art, we iterate back and forth through several synthetic-analytical processes, adapting the perceived information to our informational capability. The "aestetic pleasure" is a subjective feeling of the viewer who "discovers order" (see the above formula). Details about this semiotic application in generative art, along with similar applications can be found in our previous work [31, 37, 38, 39].

The "aestetic pleasure" concept was a "heroic experiment" of the 1960s (to quote Frieder Nake [12]), since it is too reductionist, and was later abandoned. However, what remained, also according to Nake, was the semiotic approach to aesthetics, not the numeric. In our example, the semiotic aggregation illustrates well the complex sequence of choices made to optimize a loss function (in this case, the "aesthetic pleasure").

3 Superization in Deep Learning

In this section we define our core problem: How can we design or optimize a CNN, the most common deep learning model, based on semiotic principles? The layers of a CNN can be interpreted as multi-resolution representations of the input images [40, 41, 42]. We may consider the multi-resolution image representation example in the context of a semiotic recognition process, where the interpretant attempts to classify an input image. Can we interpret the neural layers as combinations of supersigns obtained by superization from the previous layer?

In a CNN, complex objects are composed of simpler object parts as the receptive field of the network grows and combines multiple neurons from previous layers [21]. By extension, it is interesting to observe if any form of superization is present in the training process of a CNN.

In [27], we considered a CNN model in the context of a semiotic recognition process, where the machine (the interpretant) attempts to classify an input pattern. We imagined multi-layered classification as a semiotic process where each layer performs a superization of the previous layer and the superization information is made available to the interpretant. Upon the completion of a successful recognition process, the entropy of the output layer reaches zero, indicating that no additional information extraction is required.

In the following, we will describe the interpretation of superization in the learning phase of a CNN with respect to the relationships between successive neural layers.

A type I superization appears when we reduce the spatial resolution of a layer $k+1$ by subsampling layer k. This bears resemblance to class formation as it involves minimizing the variation of the input values, by reducing the number of distinctive signs.

The pooling operator in a CNN partitions the input image into non-overlapping rectangles and performs downsampling (max or average pooling). For instance, max pooling applied to a feature map F at layer k and locations (i, j) with a kernel of 2×2 is computed as:

[3] George David Birkhoff.
[4] Abraham Moles.

$$O_{i,j}(F) = max(F_{i,j}, F_{i+1,j}, F_{i,j+1}, F_{i+1,j+1}) \tag{1}$$

Pooling, however, involves more than just downsampling; it is a fusion of filtering and subsampling. Filtering can be interpreted as type II superization. Hence, pooling represents a synthesis of the two forms of superization.

A type II superization is produced when applying a convolutional operator to a neural layer k. As a consequence, layer $k+1$ will concentrate on discerning more intricate objects, formed by the combination of objects already identified by layer k. For a 3×3 kernel W, the convolutional operator for feature map F at layer k and pixel (i, j) is:

$$O_{i,j}(F) = \sum_{x=0}^{2} \sum_{y=0}^{2} F(i+x, i+y) W(x, y) \tag{2}$$

The output O of the convolutional operator is a linear combination of the input features and the learned kernel weights. Therefore, a neuron in the resulting layer can identify a combination of simpler objects by composing the previously detected supersigns.

Our intuition is that using the semiotic superization analogy, there should be a tendency of entropy decrease at successive pooling and convolutional CNN layers. Our experiments confirmed that, in general, the entropy decreases when progressing through the depth of the network, but this is not always happening [27, 28].

During training a CNN, both types of superizations are effective. For type I superization, the pooling operation combines signs (scalar values) by criteria like average value or maximum value, reducing their number and building equivalence classes.

For type II superization, it is known that CNNs compose whole objects starting from simple object parts [21]. This precisely characterizes the second form of superization, as it constructs composite supersigns by combining simpler component supersigns. The receptive fields are gradually enlarged after each convolutional layer is applied. As the receptive field expands, a neuron within a hidden layer can encompass a significantly broader area of interest of the input image, thereby becoming activated for increasingly intricate objects.

The challenge arises from the simultaneous operation of both superizations in some layers, making it difficult to disentangle their individual effects.

Our hypothesis is that to decrease the entropy noticeably, the first type of superization is more effective, while the second type is more responsible with building supersigns with semantic roles.

In a simplified representation, a multi-layered classifier can be understood as a progression: from syntactics to semantics and finally to pragmatics. The last layer is connected to the outer world of classes and, at the end of a successful recognition process, the entropy of this layer becomes 0. This could be regarded as the pragmatic level in Morris' semiotic theory, as it elucidates the connection between input signs and output objects, which in turn can be linked to decisions and actions.

4 Applications

This section summarizes our previous work on semiotic techniques for CNN training and architecture optimization, which are based on the semiotic interpretation of aggregation in CNNs [27, 28, 29, 30]. In addition, we provide an in-depth analysis of the semiotics aspects.

4.1 CNN Optimization by Layer Pruning

A significant trend in deep learning involves optimizing these networks to adhere to various hardware limitations [43]. *Pruning* is the process of reducing the size of a neural network by removing unimportant weights, neurons, or filters, without significant loss in accuracy [44]. The goal of pruning is to achieve more efficient models with fewer parameters and faster inference time.

The practice of pruning has recently become more pertinent and in demand. This is due to the tendency of modern network architectures to be overparametrized, providing more opportunities for

optimization [45]. Recent advancements in pruning methods have notably progressed in the past few years, demonstrating the capability to significantly reduce the computational workload of a deep neural network multiple times over without compromising accuracy [46]. A large suite of methods for pruning have been proposed in the last years. Comprehensive surveys on neural network pruning can be found in [47, 48].

There are two main pruning methods: weight pruning and structural pruning. Weight pruning involves setting small weights to zero, which can lead to sparse connectivity patterns [49]. Structural pruning removes entire neurons or filters, which preserves the dense connectivity patterns but reduces the model size [50]. Pruning individual parameters in an unstructured way is hard to deploy on existing hardware. Pruning an entire filter is friendly to hardware implementation, and is the dominant filter pruning method.

In our problem statement, the main objective is the reduction of the total number of floating point operations per second (FLOPS) and parameters, by removing redundant weights from the network. The ratio between the number of FLOPS after compression and the number of FLOPS before compression measures the sparsity of a network. Typically, the initial network is large and accurate, and the goal is to produce a smaller network with similar accuracy.

A typical approach [49] is to first train the network. Afterwards, each parameter or structural element in the network is issued a score, and the network is pruned based on these scores. Pruning reduces the accuracy of the network, so it is trained further (known as fine-tuning) to recover. The process of iteratively pruning and fine-tuning is commonly repeated, progressively decreasing the size of the network.

Pruning is a trade-off between model efficiency and quality, with pruning increasing the former while (typically) decreasing the latter. It can improve the time or space vs. accuracy trade-off of a given architecture, sometimes even increasing the accuracy [48].

4.1.1 Grad-CAM Saliency Map

In visual recognition, a saliency map (e.g., shown in Fig. 11) functions to highlight the crucial or standout features (pixels) within an input image that drive a specific decision. In the context of a CNN classifier, this decision correlates with identifying the class that achieves the highest likelihood score. As depicted in Fig. 11, the saliency map takes the form of a heatmap, visually conveying the significance of individual features through varying intensity levels.

Figure 11: A saliency map generated [27] using the Grad-CAM method. It highlights the most important pixels that contribute to the prediction of the class "boxer" (dog). Red denotes important regions.

The concept of a saliency map is not new and predates the rise of CNNs [51]. Within the realm of CNNs, Simonyan *et al.* [52] were among the pioneers investigating saliency maps. They employed backpropagated gradients concerning the input image as a signal, where higher gradient tensor magnitudes indicated greater importance of corresponding pixels. However, in deep CNNs, the gradient signal related to specific classes diminishes as it moves backward through the network. Consequently, in [51], saliency maps for numerous images tended to be noisy and challenging to interpret. The same

study introduced a technique for generating class-specific images: employing gradient descent on a randomly initialized noise image until it converges, aiming to maximize the likelihood of a particular class. The resultant images successfully captured certain semantic elements characteristic of images belonging to that class.

Grad-CAM, a widely adopted and modern technique for visualizing saliency maps [22], leverages gradient information derived from backpropagating the error signal through the loss function concerning a specific feature map $A^{(l)} \in R^{w \times h \times c}$ at any given layer l of the network. Here, w, h, and c denote the width, height, and number of channels of that particular feature map. The gradient signal is averaged across the spatial dimensions $w \times h$ to generate a c-dimensional vector denoting the importance weights α_k. These weights are utilized to perform a weighted combination across channels with the feature maps $A_k^{(l)}$ and then passed through a ReLU activation function:

$$O_{Grad-CAM}^{(l)} = ReLU\left(\sum_{k=0}^{c} \alpha_k A_k^{(l)}\right) \tag{3}$$

Grad-CAM can show what parts of different layers in a deep network are active. In [22], it was only used for the last layer to understand its decisions. In our tests, we used Grad-CAM for all CNN layers, to make maps that highlight important areas.

Having computed the saliency map for each layer using Grad-CAM, we need an efficient method to compute the entropy of the structures for which spatial relationships are important.

4.1.2 Image Spatial Entropy

A saliency map aggregates information from the previous layer and this information is measured by the spatial entropy of the map.

We summarize here the main formulas. Details can be found in [27]. The joint probability of pixels at spatial locations (i, j) and $(i + k,\ j + l)$ to take the value g, respectively g' is:

$$p_{gg'}(k, l) = P(X_{i,\,j} = g, X_{i+k,\,j+l} = g') \tag{4}$$

where g and g' are pixel intensity values $(0 - 255)$. If we assume that $p_{gg'}$ is independent of $(i,\ j)$ (the homogeneity assumption [53]), we define for each pair $(k,\ l)$ the entropy:

$$H(k,\ l) = -\sum_{g}\sum_{g'} p_{gg'}(k,\ l) \log p_{gg'}(k,\ l) \tag{5}$$

where the summations are over the number of outcome values. A standardized relative measure of bivariate entropy is [53]:

$$H_R(k,\ l) = \frac{H(k,\ l) - H(0)}{H(0)} \in [0,\ 1] \tag{6}$$

The maximum entropy $H_R(k,\ l) = 1$ corresponds to the case of two independent variables. $H(0)$ is the univariate entropy, which assumes all pixels as being independent, and we have $H(k,\ l) \geq H(0)$.

The Aura Matrix Entropy (AME, see [54]) is:

$$H_{AME}(\mathbf{X}) \approx \frac{1}{4}\Big(H_R(-1,\ 0) + H_R(0,\ -1) + H_R(1,\ 0) + H_R(0,\ 1)\Big) \tag{7}$$

Starting from a map obtained by the Grad-CAM method, we compute the probabilities $p_{gg'}$ in Equation 4, and finally the AME in Equation 7, which results in the spatial entropy quantity of a saliency map.

The Mutual Information (MI) between two saliency maps $M1$ and $M2$ is:

$$I(M1, M2) = H(M1) + H(M2) - H(M1, M2) \tag{8}$$

where $H(\cdot)$ is the (spatial) entropy of a variable, and $H(\cdot,\cdot)$ is the joint (spatial) entropy between two variables. We modify the simplified aura matrix entropy to be applicable for joint entropy calculation by changing Equation 4 to:

$$p_{gg'g''g'''}(k,l) = P(M1_{i,j} = g, M1_{i+k,j+l} = g', M2_{i,j} = g'', M2_{i+k,j+l} = g''') \tag{9}$$

where g, g', g'', g''' are pixel intensity values. The upcoming equations from the spatial entropy computation will use $p_{gg'g''g'''}$ instead of $p_{gg'}$. The final modification will be to Equation 7, where we take into consideration four spatial positions instead of two, the first two from $M1$ and the last two from $M2$:

$$H_{AME}(\mathbf{X}) \approx \frac{1}{4}\Big(H_R(-1,\ 0,-1,\ 0) + H_R(0,\ -1,0,\ -1) + H_R(1,\ 0,1,\ 0) + H_R(0,\ 1,0,\ 1)\Big) \tag{10}$$

We apply Equation 10 to compute the joint spatial entropy between two saliency maps and we can use Equation 8 to compute the Mutual Entropy (MI).

4.1.3 Layer Pruning

In [27], we highlighted the statistical aspects related to how information concentrates in saliency maps across consecutive CNN layers. We approached the analysis of these saliency maps through a semiotic lens. Throughout the process of superization, we noticed a reduction of the spatial entropy, signifying the aggregation of signs into supersigns. Each saliency map consolidated information from the preceding network layer. Our investigation delved into the potential application of semiotic tools to optimize deep learning neural network architectures by employing a semiotic greedy technique on saliency maps. Therefore, we monitored the entropy dynamics of the saliency maps at each layer. We then selectively removed layers where the reduction in entropy was not substantial compared to the previous layer, disrupting the intended process of superization.

Fig, 12 illustrates how superization leads to a decline in spatial entropy across the layers of the network due to repeated downsampling. Nevertheless, certain convolutional layers exhibit a consistent maintenance of spatial entropy.

AlexNet				
Layer	Pretrained	Random	Pretrained	Fine-tuning
	ImageNet	ImageNet	Caltech101	Caltech101
conv1	0.6830	0.6816	0.6786	0.6829
relu1	0.6806	0.6802	0.6746	0.6795
maxpool1	0.5252	0.5113	0.5264	0.5356
conv2	0.5311	0.5100	0.5395	0.5352
relu2	0.5231	0.5096	0.5297	0.5191
maxpool2	0.4147	0.3952	0.4241	0.4116
conv3	0.4423	0.3861	0.4508	0.4474
relu3	0.4326	0.3864	0.4437	0.4454
conv4	0.4272	0.3867	0.4375	0.4292
relu4	0.4214	0.3934	0.4222	0.4304
conv5	0.4056	0.3934	0.4019	0.3925
relu5	0.3928	0.3949	0.3878	0.3784
maxpool3	0.3114	0.3038	0.3077	0.3071

Figure 12: Entropy values for saliency maps for AlexNet at different levels in the network. Similar results were obtained on VGG16 and ResNet50 [27].

Using the VGG16 architecture as a baseline, we iteratively applied the following greedy algorithm: (i) train the network on CIFAR-10 using the SGD optimizer with a learning rate of 0.01; (ii) compute the spatial entropy for each saliency map; (iii) remove a layer for which the entropy does not decrease; and (iv) repeat steps (i)-(iii) until the performance does not degrade too much.

Network	Number of parameters	Accuracy
VGG16	15.245.130	89.55%
VGG11	9.750.922	87.83%
VGG16 after 4 layers removed	9.345.354	89.57%
VGG16 after 8 layers removed	2.118.346	89.49%

Table 1: Comparisons on CIFAR-10 - top 1 accuracy between VGG16, VGG11 (the smallest configuration from the VGG family), VGG16 after 4 layers removed (which has roughly the same number of parameters as VGG11) and VGG16 after 8 layers removed (which is the smallest configuration which maintains the accuracy within 1% difference) [27].

The tests showed that removing up to 8 layers from the network didn't really change its performance much, dropping by less than 1%. But when we removed the 9th layer, the accuracy went down a lot. So, we stopped removing layers at that point.

An interesting discovery was that the sequence of layer removal significantly influenced the outcome. Removing small layers with fewer parameters from the initial part of the network led to a 2% decrease in accuracy after the third removal. However, when eliminating larger (over-parametrized) layers starting from the mid-end section of the network, the accuracy remained stable. Particularly, removing the second convolutional layer with 64 output channels resulted in a notably rapid decline in accuracy.

Our interpretation was that the first two convolutional layers played a crucial role in the network's subsequent performance. This initial section of the network, termed the "stem" in the literature [55], occurs before a subsampling operation. Some variations of ResNets implement this stem as three 3×3 convolutional layers or a single large 7×7 layer. These early layers are responsible for detecting low-level features like edge patterns. Having just a single 3×3 convolutional layer, instead of two or three, implies that the receptive field before the initial max-pooling operation is limited to 3×3, potentially hindering the proper detection of basic strokes and edges.

The resulting network configuration was: 64, 64, M, 128, M, 256, M, 512, M, M, where "M" indicates max-pooling, and the integers represent a convolutional layer with the corresponding number of output channels, followed by a ReLU non-linearity. There were no alterations to the fully connected layers from the original architecture. The outcomes are detailed in Table 1. It's evident that even with a reduction in network capacity by approximately $7.5\times$ from the original network, the accuracy is preserved, indicating the network's excessive overparametrization for this task. Also, when comparing the original VGG11 with a pruned version of VGG16 with the same number of parameters obtained using our method, VGG16 outperformed VGG11.

To confirm if this configuration applies to other tasks, we trained the network on CIFAR-100 and contrasted it with the performance of the full VGG16. The full VGG16 achieved an accuracy of 62.61%, whereas the optimized VGG architecture obtained 63.78%. Remarkably, the smaller network slightly improved the performance compared to the full network, despite being significantly smaller.

We visualized this iterative layer pruning process in [27] by plotting the saliency maps at different key layers, where the spatial entropy value dropped significantly between successive layers. In compliance with the theory of semiotic superization, it became visible how supersigns were gradually formed, layer by layer. Interesting, semiotic superization took place inside a CNN regardless of the architecture of the network.

4.2 Information Bottleneck in Deep Learning - A Semiotic Approach

In [28], we investigated semiotic superization in the context of the Information Bottleneck (IB) principle. We studied the evolution of spatial entropy of CNN saliency maps to validate/invalidate this principle and applied the results to pruning.

The IB concept was introduced in [56] as an information-theoretic approach designed to identify the optimal balance between the predictive accuracy of a variable Y and the compression of the input random variable X within the code T. This objective is achieved through the minimization of the following Lagrangian [56]:

$$\min_{P_{T|X}} I(X;T) - \beta I(Y;T) \tag{11}$$

where $I(\cdot,\cdot)$ is the MI of two random variables and β is a trade-off parameter. In a CNN architecture, MI computation can suffer from high computational cost and sensitivity to noise. Therefore, it is critical use an efficient MI estimation method.

Recently, the IB principle was applied to deep learning by Tishby and Zaslavsky [23], offering a theoretical framework to elucidate the fundamental mechanisms governing modern deep learning architectures. Extending this concept, Shwartz-Ziv and Tishby [24] conceptualized the layers within a deep neural network as a sequential Markov chain representing internal representations of the input X. Each latent representation T is characterized through an encoder $P(T|X)$ and a decoder $P(\hat{Y}|T)$, where $\hat{Y}$ signifies the neural prediction. They defined the Information Plane (IP) as the coordinate plane displaying the mutual information quantities $I_X = I(X;T)$ and $I_Y = I(T;Y)$ across multiple training epochs. In the context of a multi-layer perceptron tackling a synthetic data problem, they observed two critical phases during training: a fitting phase characterized by simultaneous increases in $I(X;T)$ and $I(T;Y)$, and a compression phase where $I(X;T)$ starts declining while $I(T;Y)$ remains relatively constant. They linked the reduction in $I(X;T)$ to the compression of input X into the latent space T, preventing overfitting and thereby elucidating the effective generalization achieved by overparameterized CNNs.

Based on the IB theory, the two distinct phases (fitting and compression), characterize the MI of: *a)* the input X and the internal representation T; and *b)* the internal representation T and the output Y. According to this principle, a good internal representation produced by a neural model should maximally compress the input data, while preserving sufficient information about the output. This is similar to what happens in visual information adaptation.

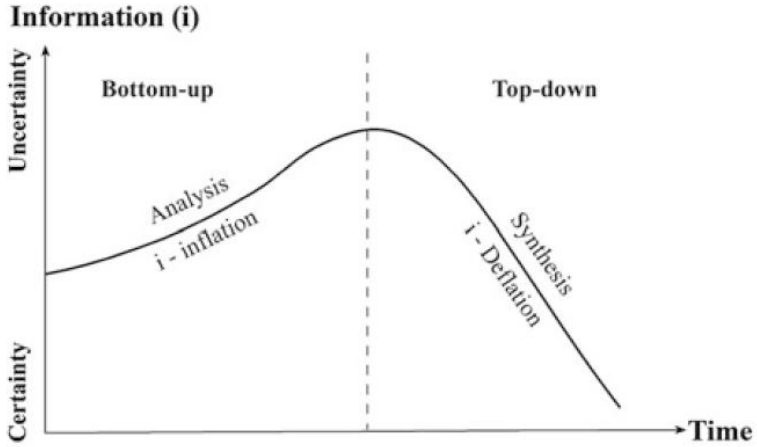


Figure 13: The 2-stages process of information adaptation [57]: It starts with a bottom-up process of information inflation and continues with a top-down process of information deflation until the appropriate quantity of information is adapted to the required task.

Information adaptation can be interpreted as a process characterized at least in its later stage by a gradual reduction of the number of alternatives from which the answer is to be selected to respond to

a given question, until eventually one of the alternatives is established as the unique answer. In some cases, the process is characterised by a shift from a pre-established unique alternative to another new alternative. In such a case, we have an unlearning stage, followed by a pure learning stage [58, 59]. Therefore, we can regard the CNN layers as learning layers characterised by the evolution of their information content with (see Fig. 13).

We explored within the CNN model this semiotic aspect of information adaptation. Our experiments revealed that throughout the training process, the entropy of each neural layer steadily increases until it reaches its peak value. This phase appears to correspond to the information adaptation of the model. The subsequent decline in entropy is linked to the processing of structural information. The rate of this decline significantly hinges on the quantity of structural information present in the input layer. In scenarios with minimal structural information, entropy diminishes rapidly, whereas in cases with substantial structural elements, the neural layers maintain consistently high entropy across a significant range. The fluctuation in entropy reflects the nature of the information encoded within the input layer [32, 33]. We believe that this phenomenon of information adaptation aligns with the two distinct phases outlined in the IB principle—fitting and compression.

Next, we studied variation of the spatial entropy of saliency maps through the whole training process in conjunction with its layer-wise behaviour. Our investigation [28] focused on two aspects:

- The progression of MI between input and intermediate saliency maps, as well as between intermediate and output saliency maps, throughout the training process.
- The changes in spatial entropy within saliency maps over the course of training.

The IB plane analysis, as described in [24], tracked the two MI quantities $I(X;T)$ and $I(Y;T)$ and noticed the fitting and compression patterns emerging. As such, we analyzed the information planes between $I(X;T)$ and $I(Y;T)$ by computing the MI from Equation 8 between the first and an intermediate saliency map, and between the last and the same intermediate saliency map. The proposed experiment was meant to uncover any resemblance to the original results from [24], but applied to a different concept like saliency maps.

We tested if there is a possible link between the IB fitting and compression patterns and the variation of the spatial entropy in saliency maps. Different than in [27], where the spatial entropy was studied at a single point in time (after training), going along the depth of the network, we captured this time the dynamics of the entropy during the whole training to see if it is governed by the same patterns.

Using the CIFAR-10 dataset, we trained a standard VGG16 architecture and applied the Grad-CAM method to generate saliency maps at each layer. Analyzing the behavior of MI derived from these maps after each epoch, we enhanced estimation accuracy by averaging MI values across 50 randomly chosen samples from the training set. Similar to observations in [24], we noted an increase in $I(X;T)$ and $I(Y;T)$ during the initial epochs. However, subsequent epochs showed no discernible pattern indicative of compression, leading us to empirically conclude the absence of the IB concept in the mutual information of these saliency maps.

When we derived a bit from the study of MI and analyzed the evolution of spatial entropy for saliency maps over time for the same VGG16 architecture, we noticed a pattern. The spatial entropy increased during the initial phase of the training and at some point plateaued. We observed the same patterns on other well-known network architectures: ResNet [60], DenseNet [61], and GoogleNet [62]. However, we encountered some exceptions to the patterns, which were present only for a few layers.

We observed that early layers experienced a more sudden increase in entropy values during the initial few epochs. This phenomenon could be explained by the fact that the initial layers of a CNN specialize in recognizing simpler concepts like edges. Consequently, the network learned these tasks more swiftly compared to later layers, which handle more intricate concepts such as complete object parts [21].

We hypothesized a correlation between the variation of spatial entropy and the evolution of the training process. To test this hypothesis, we trained the VGG16 model on the CIFAR-10 dataset. We then froze the layers where the average spatial entropy of the saliency map, calculated over the last

five epochs, entered a phase of compression with minimal variance and fell below a threshold value, which we will denote as ϵ. Subsequently, we observed if the model achieved comparable accuracy to a fully trained network within the same or fewer number of epochs.

For a large ϵ, the layers were frozen early in the training, prohibiting learning. On the opposite side, with a small ϵ, layers were generally left unfrozen, allowing the network to undergo regular training. In our practical experiments, we found that an ϵ value of $5e-05$ yielded the best results. Our empirical results compared a fully trained VGG16 network with another VGG16 network where certain layers were frozen during training. Remarkably, even when trained with certain layers frozen, the network preserved the same or even achieved superior performance compared to its counterpart, which had all layers continuously trained. This training approach can be viewed as a form of early stopping applied at the layer level, a technique commonly used to prevent network overfitting. Thus, we establish a connection between the observed patterns in the spatial entropy of saliency maps and the training dynamics of a CNN.

We observed an intriguing property of spatial entropy in saliency maps within a neural network. Following the superization process, where entropy drops, we found that the magnitude after compression closely resembles the initial magnitude before superization. This pattern of continuation across layers suggests a connection between entropy evolution and the superization process, hinting at an inherent property in a CNN's training dynamics: the necessity to elevate spatial entropy to an upper bound set by previous layers through superization.

Driven by these observations, we identified a link between IB theory and superization. Our research in [27] indicates that superization within a CNN reduces spatial entropy, requiring subsequent layers to increase entropy to reach the initial levels from previous layers. This increase follows a trend similar to the fitting-compression phases observed in IB theory. The empirical evidence suggests a mutual dependency between IB theory and superization, providing insights into the information-theoretical aspects governing modern CNNs' training dynamics.

Although cases exist where superization does not cause a spatial entropy decrease, we noted the presence of fitting and compression phases. Most modern CNNs involve some form of subsampling, indicating the existence of the superization process during training, even though its manifestation may differ.

The semiotic superization process mirrors the IB theory's fitting (entropy increase) followed by compression (entropy decrease). In our model, spatial entropy measured the neural layers' information content. However, in our visual experiments, the entropy increase and decrease occur distinctly. While spatial entropy increased during the fitting phase, we observed entropy decrease only after the superization process, particularly in subsequent layers employing subsampling.

4.3 Pruning Convolutional Filters via Reinforcement Learning and Entropy Minimization

AutoML, a powerful technique for various tasks such as neural architecture search (NAS), hyperparameter search, data preparation, and feature engineering [63, 64], aims to automate these processes to find optimal solutions more quickly than manual methods allow. Recent advancements in AutoML involve using it for network compression via pruning [65]. This approach employs a reinforcement learning (RL) agent [66] to determine sparsity per layer and implements a magnitude-based pruning heuristic that removes filters with the smallest magnitude.

The focus has expanded beyond using accuracy alone as the reward criterion for AutoML network compression. A significant contribution involves introducing an information-theoretical reward function, specifically entropy minimization, diverging from the accuracy-centric approach [66]. In neural networks, while cross-entropy measures error, exploring the entropy of hidden layers is less common. This observation led to investigating the impact of layer entropies on network pruning, hypothesizing that minimizing entropy preserves crucial information and reduces uncertainty.

The study's novelty lies in establishing a connection between entropy minimization and structural pruning, similar to the concept of structural entropy in previous research [67]. Utilizing an AutoML framework [65], the proposed optimization approach involves minimizing spatial entropy at each convolutional layer. Empirical findings suggest that this minimization indirectly maintains accuracy,

highlighting a more principled approach to network pruning beyond solely optimizing the accuracy in the reward function.

The AMC framework operates as an AutoML tool dedicated to pruning neural networks by selecting sparsity percentages for each layer individually. This process involves an algorithm utilizing L_2 magnitude to identify and remove filters with the lowest magnitude, guided by a non-differentiable accuracy function. A DDPG agent [66] is employed, trained through actor-critic methods [68]. The agent's task is to optimize this accuracy criterion, treated as a reward function, using metrics computed from a distinct dataset, either from a split within the training or validation set. By promoting actions that yield higher rewards and discouraging those with poor outcomes, the DDPG agent drives the selection of pruning percentages.

In contrast to the original AMC approach, our modification incorporates a reward function aimed at minimizing the average spatial entropies of convolutional activations alongside accuracy. This addition seeks to explore whether entropy minimization could substitute direct accuracy computation, potentially linking neural pruning with information theory. Thus, the agent's optimization problem shifts towards determining layer-specific sparsity levels to minimize spatial entropy. To compute the mean spatial entropy per layer, we utilized convolutional outputs from a subset of 100 samples, representing an estimation of the entire dataset, due to computational constraints.

Our hypothesis revolved around the possibility that minimizing spatial entropy might yield comparable or superior outcomes compared to maximizing accuracy. If validated, this empirical link between pruning and information theory would suggest that eliminating redundant information from a model can achieve comparable accuracy to direct accuracy maximization strategies.

We started by training a standard VGG16 on the CIFAR10 dataset. For that, we trained for 200 epochs using the SGD optimizer with a learning rate of 0.01 and cosine annealing scheduler [69]. Pruning was applied afterwards on the pretrained network after which the new network configuration is fine-tuned, as is standard in pruning literature.

In order to establish a baseline to compare our method with, we used the original formulation of the AMC framework and optimize first the network using the accuracy criterion. To achieve a certain level of pruning, AMC pushes up the level of sparsity until only a predefined percentage of the total FLOPS are maintained. The ratio between the number of FLOPS after compression and the number of FLOPS before compression can measure indirectly the amount of sparsity in a network.

We noticed that with entropy minimization we achieved the same performance as when accuracy is used as a reward. The solution found by this method has $10\times$ less FLOPS and $\approx 38\times$ less parameters than the original VGG-16 network. For entropy maximization, the framework produced a solution which has indeed fewer parameters, but used the same number of FLOPS as the method with entropy minimization. We could see though that the resulting network architecture has a much poorer accuracy performance.

In order to test the generality of our method for various other architectures, we repeated the same experiments for other popular networks: MobileNetV2 [70] and ResNet50 [60]. Our method was on par with the original AMC framework for various architectures and FLOPS preservation percentages. The only noticeable drop in performance was for ResNet50, which was previously observed to contain less redundancy [71] and was the most difficult to compress, even when using accuracy as a criterion.

Using an information-theoretical optimization criterion, which aims to minimize entropy, we achieved the same performance as when we optimize directly the accuracy of the model. We were able to reduce the total number of FLOPS of a VGG-16 architecture by $10\times$ and the number of parameters by $\approx 38\times$, while incurring minimal accuracy drop, with similar results for other popular architectures.

In this application, the semiotic interpretation of the pruning procedure is the following:

- We minimize the entropy of convolutional activations. In a semiotic interpretation, convolution is similar to type II superization. This means that we attempt to minimize the entropy of the subsequent level of supersigns.

- Pruning is achieved by removing (canceling) unimportant weights from the convolutional filters. This means that type II superization compounds obtained after convolution are less structured. Therefore, pruning accelerates the type II superization process, maximizing information reduction for the subsequent layer.

5 Conclusions

Our fundamental thesis is that information concentration, expressed as semiotic superizations, appear in successive layers of deep learning models: Each layer aggregates information from the previous layer of the network. Our experiments generally confirm this statement. Our approach can be extended to other neural networks, since semiotic superization appears to be present in many architectures.

We can build and optimize neural models using this semiotic framework. We were able to significantly simplify the architecture of CNNs, by pruning layers and filters. While this optimization process can be slow, our work tries to use the notion of computational semiotics to prune existing state of the art networks top-down, instead of constructing the network bottom-up, a standard neural architecture search procedure.

We also can go from effect to cause, and interpret neural models as sequences of semiotic processes. The obtained interpretation can be used to visualize and explain decision processes within CNN models.

According to Mihai Nadin, in a relaxed interpretation, most inference engines utilized in contemporary machine learning incorporate semiotic elements [72, 73]. However, this is more like a general principle than a practical tool. In the best case scenario, designers are guided more by semiotic intuition than by a comprehensive understanding of semiotic principles and we are actually very far from a complete computational model of Peirce's semiosis. Our results may be the starting point of a general unifying semiotic framework for the design, optimization, and interpretation/explanation of deep neural models.

References

[1] U. Eco, *A Theory of Semiotics*. Indiana University Press, 1976.

[2] T. Sebeok, *Signs: An Introduction to Semiotics*, ser. Toronto Studies in Semiotics. University of Toronto Press, 1994.

[3] D. Chandler, *Semiotics: The basics*. Taylor & Francis, 2017.

[4] C. S. Peirce, *Collected papers of Charles Sanders Peirce*. Harvard University Press, 1960, vol. 2.

[5] A. Jappy, "Iconicity, hypoiconicity," in *The Commens Encyclopedia: The Digital Encyclopedia of Peirce Studies. New Edition*. Commens, 2014.

[6] E. Rochberg-Halton and K. McMurtrey, "The foundations of modern semiotic: Charles Peirce and Charles Morris," *The American Journal of Semiotics*, vol. 2, no. 1/2, pp. 129–156, 2007.

[7] C. Morris and M. Charles William, *Writings on the General Theory of Signs*, ser. Approaches to semiotics. Mouton, 1972.

[8] H. Zemanek, "Semiotics and programming languages," *Communications of the ACM*, vol. 9, no. 3, pp. 139–143, 1966.

[9] K. Tanaka-Ishii, *Semiotics of Programming*, 1st ed. USA: Cambridge University Press, 2010.

[10] M. Bense, *Semiotische Prozesse und Systeme in Wissenschaftstheorie und Design, Ästhetik und Mathematik*. Baden-Baden: Agis-Verlag, 1975.

[11] A. A. Moles, *Information Theory and Esthetic Perception*. Urbana,: University of Illinois Press, 1966.

[12] F. Nake, "Information aesthetics: An heroic experiment," *Journal of Mathematics and the Arts*, vol. 6, no. 2-3, pp. 65–75, 2012.

[13] H. Frank, *Kybernetische Grundlagen der Pädagogik: eine Einführung in die Informationspsychologie und ihre philosophischen, mathematischen und physiologischen Grundlagen.* Baden-Baden: Agis Verlag, 1969.

[14] R. Gunzenhäuser, *Maß und Information als ästhetische Kategorien: Einführung in die ästhetische Theorie GD Birkhoffs und die Informationsästhetik.* Agis Verlag, 1975.

[15] J. Rigau, M. Feixas, and M. Sbert, "Informational aesthetics measures," *IEEE computer graphics and applications*, vol. 28, no. 2, pp. 24–34, 2008.

[16] R. Gudwin and F. Gomide, "A computational semiotics approach for soft computing," in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 4. IEEE, 1997, pp. 3981–3986.

[17] A. Gomes, R. Gudwin, and J. Queiroz, "Towards meaning processes in computers from peircean semiotics," *SEED Journal–Semiotics, Evolution, Energy, and Development*, vol. 3, no. 2, pp. 69–79, 2003.

[18] R. R. Gudwin, "Semiotic synthesis and semionic networks," *SEEDJournal (Semiotics, Evolution, Energy, and Development)*, vol. 2, no. 2, pp. 55–83, 2002.

[19] R. Gudwin and J. Queiroz, "Towards an introduction to computational semiotics," in *International Conference on Integration of Knowledge Intensive Multi-Agent Systems.* IEEE, 2005, pp. 393–398.

[20] J. S. Baxter, E. Gibson, R. Eagleson, and T. M. Peters, "The semiotics of medical image segmentation," *Medical image analysis*, vol. 44, pp. 54–71, 2018.

[21] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *CoRR*, vol. abs/1311.2901, 2013. [Online]. Available: http://arxiv.org/abs/1311.2901

[22] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-CAM: Why did you say that? Visual explanations from deep networks via gradient-based localization," *CoRR*, vol. abs/1610.02391, 2016. [Online]. Available: http://arxiv.org/abs/1610.02391

[23] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," 2015.

[24] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," 2017.

[25] B. Kovalerchuk, R. Andonie, N. Datia, K. Nazemi, and E. Banissi, "Visual knowledge discovery with artificial intelligence: Challenges and future directions," in *Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery.* Cham: Springer International Publishing, 2022, pp. 1–27.

[26] B. Kovalerchuk, K. Nazemi, R. Andonie, N. Datia, and E. Banissi, *Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery.* Springer, 2022.

[27] B. Muşat and R. Andonie, "Semiotic aggregation in deep learning," *Entropy*, vol. 22, no. 12, 2020. [Online]. Available: https://www.mdpi.com/1099-4300/22/12/1365

[28] B. Muşat and R. Andonie, "Information bottleneck in deep learning-a semiotic approach," *International Journal of Computers Communications & Control*, vol. 17, no. 1, 2022.

[29] ——, "Pruning convolutional filters via reinforcement learning with entropy minimization," in *Artificial Intelligence and Soft Computing*, L. Rutkowski, R. Scherer, M. orytkowski, W. Pedrycz, R. Tadeusiewicz, and J. M. Zurada, Eds. Cham: Springer Nature Switzerland, 2023, pp. 167–180.

[30] B. Muşat and R. Andonie, "Accelerating convolutional neural network pruning via spatial aura entropy," in *2023 27th International Conference Information Visualisation (IV)*, 2023, pp. 286–291.

[31] I. Stan and R. Andonie, "Cybernetical model of the artist-consumer relationship (in Romanian)," *Studia Universitatis Babes-Bolyai*, vol. 2, pp. 9–15, 1977.

[32] R. Andonie, "A semiotic approach to hierarchical computer vision," in *Cybernetics and Systems (Proceedings of the Seventh International Congress of Cybernetics and Systems, London, Sept. 7-11, 1987)*, J. Ross, Ed. Lytham St. Annes, U.K.: Thales Publication, 1987, pp. 930–933.

[33] ——, "Semiotic aggregation in computer vision," *Revue Roumaine de linguistique, Cahiers de linguistique théorique et appliquée*, vol. 24, pp. 103–107, 1987.

[34] J. C. Baez, T. Fritz, and T. Leinster, "A characterization of entropy in terms of information loss," *Entropy*, vol. 13, no. 11, pp. 1945–1957, 2011. [Online]. Available: https://www.mdpi.com/1099-4300/13/11/1945

[35] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.

[36] A. K. Wong and M. A. Vogel, "Resolution-dependent information measures for image analysis," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, no. 1, pp. 49–61, 1977.

[37] R. Andonie and A. Marian, "A probabilistic model in the automatic generation of visual structures," *Revue Roumaine de linguistique, Cahiers de linguistique théorique et appliquée*, vol. 22, pp. 3–17, 1985.

[38] ——, "Piet Mondrian – Computer aided analysis and synthesis (in Romanian)," in *Mathematical Semiotics of Visual Arts*, S. Marcus, Ed. Bucureşti: Editura Ştiinţifică şi Enciclopedică, 1982, pp. 66–72.

[39] A. Marian, P. Puşcaş, and R. Andonie, "The bases of a metalanguage in the cybernetic aesthetics (possible relationships between the visual and sonorous structures at the level of near, mid, and distant orders)," *Revue Roumaine de linguistique*, vol. 30, pp. 51–65, 1985.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 346–361.

[41] I. Kokkinos, "Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5454–5463.

[42] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[43] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," *CoRR*, vol. abs/1710.09282, 2017. [Online]. Available: http://arxiv.org/abs/1710.09282

[44] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in neural information processing systems*, 1990, pp. 598–605.

[45] S. Oymak and M. Soltanolkotabi, "Towards moderate overparameterization: global convergence guarantees for training shallow neural networks," *CoRR*, vol. abs/1902.04674, 2019.

[46] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?" 2020.

[47] T. Gale, E. Frank, and M. Johnson, "The state of sparsity in deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 8, pp. 2403–2424, 2019.

[48] D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?" *Proceedings of machine learning and systems*, vol. 2, pp. 129–146, 2020.

[49] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[50] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," in *Proceedings of the International Conference on Learning Representations*, 2017.

[51] Y. Lin, B. Fang, and Y. Tang, "A computational model for saliency maps by using local entropy," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[52] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps." *CoRR*, vol. abs/1312.6034, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1312.html#SimonyanVZ13

[53] A. G. Journel and C. V. Deutsch, "Entropy and spatial disorder," *Mathematical Geology*, vol. 25, no. 3, pp. 329–355, 1993.

[54] E. Volden, G. Giraudon, and M. Berthod, "Modelling image redundancy," in *1995 International Geoscience and Remote Sensing Symposium, IGARSS '95. Quantitative Remote Sensing for Science and Applications*, vol. 3, 1995, pp. 2148–2150.

[55] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *CoRR*, vol. abs/1602.07261, 2016. [Online]. Available: http://arxiv.org/abs/1602.07261

[56] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," 2000.

[57] H. Haken and J. Portugali, *Information adaptation: the interplay between Shannon information and semantic information in cognition.* Springer, 2014.

[58] S. Watanabe, "Learning process and inverse H-theorem," *IRE Transactions on Information Theory*, vol. 8, no. 5, pp. 246–251, 1962.

[59] ——, *Knowing and Guessing a Quantitative Study of Inference and Information.* New York: Wiley, 1969.

[60] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[61] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jul 2017, pp. 2261–2269. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.243

[62] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[63] X. He, K. Zhao, and X. Chu, "AutoML: A survey of the state-of-the-art," *Knowledge-Based Systems*, vol. 212, p. 106622, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705120307516

[64] Q. Yao, M. Wang, H. J. Escalante, I. Guyon, Y. Hu, Y. Li, W. Tu, Q. Yang, and Y. Yu, "Taking human out of learning applications: A survey on automated machine learning," *CoRR*, vol. abs/1810.13306, 2018. [Online]. Available: http://arxiv.org/abs/1810.13306

[65] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "AMC: AutoML for Model Compression and Acceleration on Mobile Devices," in *European Conference on Computer Vision (ECCV)*, 2018.

[66] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://dblp.uni-trier.de/db/conf/iclr/iclr2016.html#LillicrapHPHETS15

[67] A. Almog and E. Shmueli, "Structural entropy: monitoring correlation-based networks over time with application to financial markets," *Scientific reports*, vol. 9, no. 1, pp. 1–13, 2019.

[68] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., vol. 12. MIT Press, 1999. [Online]. Available: https://proceedings.neurips.cc/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[69] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=Skq89Scxx

[70] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 6 2018.

[71] Z. Wang, C. Li, and X. Wang, "Convolutional neural network pruning with structural redundancy reduction," in *2021 IEEE/CVF Conference On Computer Vision And Pattern Recognition (CVPR)*, 2021, pp. 14 908–14 917.

[72] M. Nadin, "Information and semiotic processes: The semiotics of computation," *Cybernetics & Human Knowing*, vol. 18, pp. 153–175, 2011.

[73] ——, "Semiotic machine," *Public Journal of Semiotics*, vol. 1, pp. 57–75, 2007.

Journal's webpage: http://univagora.ro/jour/index.php/ijccc/

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:
Andonie, R.; Muşat, B. (2024). Signs and Supersigns in Deep Learning, *International Journal of Computers Communications & Control*, 19(1), 6392, 2024.
https://doi.org/10.15837/ijccc.2024.1.6392