Representation Learning in Biological and Computer Vision

Ellen Blaine, Nick Hershey, Debnil Sur
Stanford University

June 2017

Abstract

Traditionally, computer and biological vision were two separate entities. Classical techniques construct image representations from either handcrafted features or those learned through artificial neural networks designed without biological inspiration. However, recent advances in computational power have brought ConvNets based on human vision architectures into the computer vision landscape. Here we review the foundations and research frontiers of representation learning in the context of three domains: classical computer vision tasks, biological modeling of human vision systems, and new hybrid vision pipelines rooted in neuroscience. For most vision tasks, image representations learned from the hybrid approach outperform their classical counterparts.

1 Introduction

The first layer of any architecture for biological or computer vision must convert preprocessed image "data" into a representation that is compatible with the rest of the system. This representation can take many forms, from low-level features, such as raw digital pixel arrays or their gradients, to higher-level object segmentations and scene graphs. In each case, the success of an algorithm on a computer vision task depends heavily on its data representation. For decades, representations typically consisted of carefully engineered hierarchical transformations of pixel arrays. On the other hand, deep learning pipelines employ representation learning to adjust some randomly initialized representation during training [5]. While this approach involves less feature engineering, researchers still have to choose the appropriate model architectures to ensure that useful representations are learned.

Many successful models incorporate representation learning algorithms that utilize uninformed priors and have relatively simple architectures with no basis in neurophysiology. But advances in our understanding of representation for human vision can inform representation theory for computer vision. In this literature review, we lay out important concepts and foundations from classical computer vision and the biology of vision, including a summary of the current state of research for both. Finally, we discuss recent attempts to merge the two and create better networks for computer vision tasks.

2 Classical Computer Vision

Classical computer vision techniques rely on features handcrafted or learned from pixel data to infer a representation of objects in the world. The methods outlined in Section 4 generally outperform this class of tools; however, classical techniques dominated the computer vision landscape before biologically inspired ConvNets became computationally tractable. In the following sections, we highlight a few examples of representations learned for various vision problems.

2.1 Matching Features

A machine cannot infer a representation of an object's structure from a single view. Matching describes a domain of classical computer vision that computes correspondences between images. For example, consider the task of creating a panorama photo from two images: One approach might involve extracting a set of handcrafted features invariant to factors such as scale and illumination, running a nearest-neighbors algorithm to compute matches, and finally using those matches to piece the images together. In other words, we can represent an the photo's subject as a set of matched keypoint descriptors. Alternatively, we might use matching features for object recognition based on a library of training data, where an object's representation consists of low-level features. A popular feature extraction and matching algorithm is outlined in [19]. Its high-level feature extraction steps operate as follows:

1. Search over all scales and orientations efficiently to identify candidate image points invariant to scale and orientation.

2. Localize candidates and choose those with greatest location stability as "keypoints" of interest.

3. Assign orientations to keypoints based on local image gradients.

4. Using gradients around the keypoints and their assigned locations, orientation, and scales, create a descriptor that allows for significant levels of local shape distortion and change in illumination.

The result allows us to compute candidates for matched pairs of points using a nearest-neighbors routine detailed in the SIFT paper [19], which can then be refined using RANSAC or another fit method. Figure 3 shows the results visually.

One can also use lower-level features for matching problems, such as histogram of oriented gradient (HOG) features. This approach involves building a dictionary of HOG descriptors at different positions, scales, and orientations of small portions of objects. Then when new images are presented, they can be broken into smaller parts, and those parts can be matched to the dictionary templates [1]. This approach is of course only invariant to scale and other distortions with a large and diverse dictionary of image data.

2.2 Supervised and Unsupervised Learning

In addition to handcrafted features like those described above, supervised and unsupervised learning can also be employed to learn a representation. We summarize a few popular techniques below.

(a) Before RANSAC (b) After RANSAC

Figure 1: A sample instance of matching, where matched features are SIFT keypoints in the image. Second image shows result of filtering bad matches with RANSAC [24].

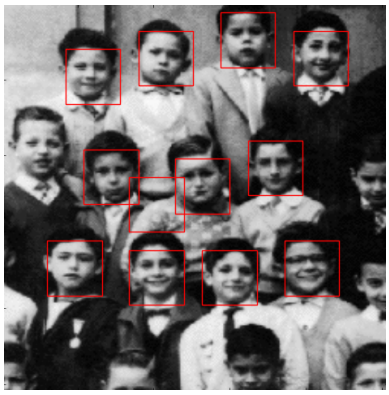

Figure 2: A sample instance of object detection via supervised learning. Features extracted using histogram of oriented gradients [24].

2.2.1 Supervised Methods

Supervised methods learn a representation of images by iteratively making predictions and comparing those predictions to some "ground truth" labels. The HOG template-matching algorithm above has a supervised representation learning analog. The input is the same, i.e. data consists of many images of parts of objects at different orientations. To decide whether two patches of images match, we input those patches into a decision network to make a prediction. Then the label is used to compute loss and update the decision network to improve future predictions [29]. See Figure 2 for an example.

2.2.2 Unsupervised Methods

There are many unsupervised representation learning techniques, but most use deep learning, which will be discussed in further detail later. Here we use sparse coding as a simple example. Sparse coding takes a large amount of unlabeled images as input, and then outputs a set number of features that best convey the features of the entire original dataset. This contrasts with the handcrafted HOG dictionary analog, because there is no longer an explicit lookup method for a certain image or part of an image. Instead, the representation of some image becomes a linear combination of the learned features. An improvement to this method proposed by Yu et al. hierarchically encodes images, and then uses the signals

from the top layer of the hierarchy to output the final representation [28].

Other classical methods might include a standard feedforward neural network to encode images. One example is autoencoders (which, again, can be improved using techniques from Section 4) [22]. The simplest autoencoder might have a forward layer consisting of a stacked linear activation function and nonlinearity, where the learned representation of an input image is the nonlinearity output vector. This output can have whatever length the researcher chooses. The weights in this layer are updated according to quality of reconstructions based on these output vectors. This differs from the previous method in that the original images are used as "labels," so no expensive data labeling is necessary.

2.3 Object Representations

The previous sections discussed representations learned for small parts of objects, which are useful for matching problems. Here we outline some techniques for representing whole objects, in order to build object recognition systems. This class of problems includes object categorization (e.g. "building"), object instance classification (e.g. "Empire State Building"), and object segmentation (e.g. "these pixels form a building").

For any of these tasks, it is important that the representation learned is robust to intra-class variability, illumination, scale, deformation, occlusion, and background clutter. One option is to use the learning algorithms discusses above (e.g. autoencoders, template-based approaches) to learn a representation that explains our observations in a given class. We might also separate an object into parts before doing so. This bag-of-words representation was proposed in [12] for image search. In practice, accuracy on the categorization task is improved if we maintain some sense of the original structure of the parts with respect to each other. Absolute position data with not be invariant to scale or deformation. Instead, we might use a hierarchical bag-of-words approach to preserve information about which features we originally close together, as well as how "words" at a given level are arranged with respect to each other [15]. For even more structure, many groups use graphs to represent how parts are organized. For example, Yu et al. approach human posture representation with a star-shaped graph, with a person's torso as the graph root and their extremities as the surrounding nodes [27].

The approaches in Section 4 consistently outperform these techniques.

2.4 Scene Representations

A scene representation describes the objects in an images and their relationships to each other, which involves object localization and segmentation. We may think of the segmentation task in a bottom-up (group pixels based on local similarity) and/or top-down sense (group pixels based on belonging to the same object). Gdalyahu et al. combine these ideas in an algorithm that clusters image components hierarchically for scene segmentation by making minimal cuts to a graphical scene representation. An example bottom-up approach involves clustering via K-means or a similar algorithm, which results in an imprecise but fast-to-compute segmentation (see Figure **??** for an example). This technique is useful for person segmentation [2].

(a) Original image

(b) Segmented image

Figure 3: A sample instance of scene segmentation using K-means, with 15 clusters [24].

3 Neural Foundations of Vision

3.1 Anatomical Structure

Our current understanding of the neurology of vision is dictated a pathway from the retina through to various regions of the visual cortex that process the input in different ways. Once a stimulus hits the retina, where it is transformed into an electric signal. This signal then passes through the optic nerve to the optic chasm, which separates each eye's processing of left and right. The signal is then passed onto the lateral geniculate nucleus, a region in the thalamus which preprocesses the electrical signal before passing into the visual cortex in the occipetal lobe.

The portion of the visual cortex that first receives this transformed image stimulus is the primary visual cortex (V1), the functional portion of which is called the striate cortex. We see precise retinotopy occurring in V1: stimulation in precise regions of the visual field corresponds to stimulation in precise regions of V1. Because this processing is so low-level, the receptive field for a fixed number of V1 neurons is significantly smaller than in later processing. That said, the neurons per region in the visual field is not equal: indeed, a large portion of V1 is mapped to the center of the visual field. [4] Neurons tuned to detecting similar facets of stimuli (e.g. color, orientation, etc.) tend to cluster in cortical columns.

The prestriate cortex (V2) receives feedforward connections from V1 and starts the higher-level processing. There are neurons in V2 that are tuned to differences in binocular disparity and figure-ground differences. Although still debated, the prevailing theory suggests that object-recognition memory (ORM) begins with a layer of cells in V2.

From here, research has found a marked divergence in the pathway into two streams: the ventral stream and the dorsal stream. Although debate is still active as to the exact processing differences between the two pathways, the prevailing theory posits that the ventral stream is responsible for object representation and perception and the dorsal stream is responsible for visual conceptualization during skilled actions. [8]. In the feed-forward model of visual processing, the pathways are as follows:

- Ventral - V1 to V2 to V4 to IT
- Dorsal - V1 to V2 to V3 to MT to MST

However, a two conditions must be said. First, the divisions between these areas and some researchers suggest the existence of more or fewer regions. Second, there are connections between regions that both skip this order and that run backward and modulate the signals from earlier on.

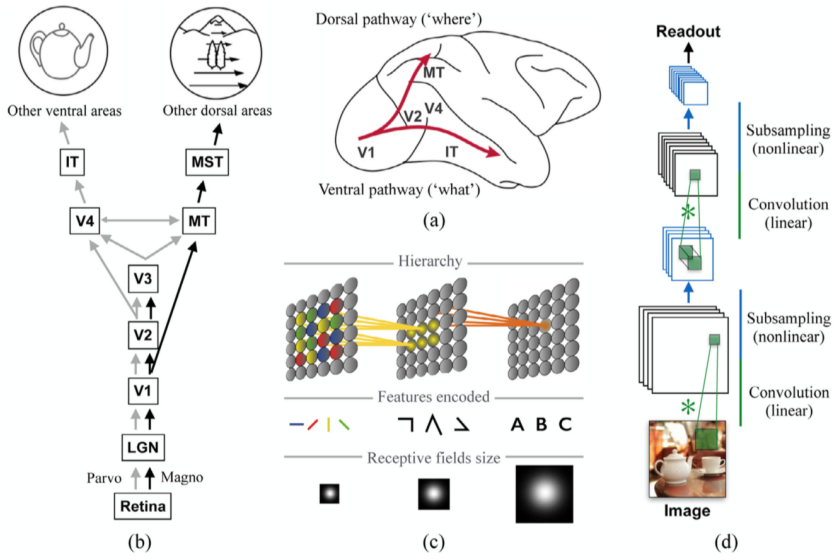

Figure 4: (a) The visual pathway in the occipital lobe. (b) The neural connections in the dorsal and ventral paths. (c) Abstractions and increasing visual fields as we move up the pathway. (d) A sample neural net schematic as used in computer vision. [20].

Functionally, visual area 4 (V4) is related to selective attention. Current research indicates that visual area 4 is the only region of visual pathway in which stimulation varies dramatically based on attention [21]. Neurons in V4 are tuned to simple figures, such as geometric shapes but not complex ones like faces.

The middle temporal visual area (MT), which receives a significant feed-forward input from V1, is primarily implicated in perceiving motion. As indicated by lesion studies, unlike V1, which processes simple motion, MT processes movement of entire objects.

From here, the pathways diverge further into ventral and dorsal areas. However, research into the exact processing done in these regions is tenuous and ongoing. And although representations might be changing at each stage, the primary representation happens earlier at the retina, LGN, and V1. Therefore, we end our discussion of the anatomy of the visual system here.

3.2 Feed-Forward Model

3.2.1 The Model

Given the primarily feed-forward nature of the visual processing system, the natural most popular model is a hierarchy of feed-forward model. In a a landmark paper in computer vision, Marr proposed what is essentially a feed-forward convolutional neural network that captures the simplified pathway. Figure 4, which is taken from Medathati et. al, frames the vision pathway under this model of a feed-forward convolutional neural network. As we move down the pathway, receptive fields increase and the features encoded grow in complexity.

This model well captures the essence of object recognition. Anatomically, a layer in the neural network corresponds approximately to a layer of tissue in the visual cortex.

3.2.2 Shortcomings

However, there are many conceptual differences that this simplification does not capture:

- Retina and LGN Transforms - The feed-forward model takes images as inputs rather than the transformations that come from the retina or lateral geniculate nucleus. These models tend to dismiss the retina and LGN to roads for the stimuli while it is transported to the visual cortex, where the real work is done. But recent research has indicated the LGN does major transformation to the input [13].
- Cortical Magnification - The current feed-forward model treats all pixels in an image equally rather than magnifiying the center.
- Backward and lateral connection - There is ample evidence of connectivity in the feedback direction from layers further along in the feed-forward model to layers earlier on in the network. It is hypothesizes that these inputs will modulate the input from earlier sections. Furthermore, within a given region of the visual cortex, there are lateral connections that affect the perception and processing at that level.
- Intracellular effects - This model, like all computationally feasible neurological models of a cognitive process, abstracts away what is happening within a given cell. However, the cells in V1 have different firing rates, internal dynamics, chemical compositions, and physical structure. Although they are the same in that they fire, other differences could have an impact.

All in all, the model lacks the full complexity of image processing. However, if any model were to truly capture the visual pathway in full detail, it would be computationally infeasible, attempting to reason about a nonlinear dynamical system that is incredibly complex.

3.3 Task-Based Approach

In comparing a human approach and a computational approach to problem-solving, a human is limited to being a generalist while a computer can be programmed to achieve a specific task. For example, a human is limited in its ability to do arithmetic partially because it also must reason about many other things. A computer, on the other hand, can be built for this specialized purpose given we a priori know what its use will be: a calculator. And given this fundamental difference, when considering biological vision for the purposes of inspiration for computer vision, it makes sense to break biological vision into its various tasks. For example, rather than tracking the entire system as multiple processes run in parallel (as in Section 3.1), we can examine exactly how the biological vision system manages object recognition, figure-ground separation, motion tracking, and more. Then, for each task, the biological methods can be examined for comparison with computer vision task and for possible integration into a generalist computer vision approach.

In this section we look at figure-ground separation. The principle of figure-ground separation was first outlined by Koffka in his landmark paper on Gestalt principles [26]. Establishing a figure is important for numerous reasons: identifying an object, tracking an object through time, establishing motion, and more. The core challenge here is extracting what feature(s), such as color, edge, luminance, depth, location, or something else, define a figure and separate it from the background. The human brain tends to making these distinctions early in V1 and V2 via contour detection in the early field. Specific neurons are tuned

toward edge detection in the visual field. Their receptive fields have distinct excitatory and inhibitory regions, which cancel out across neurons to cause mutual antagonism. Therefore, a net effect is only seen when an edge is present. Given the clear retinotopic matching, we can place edges at specific orientations in the visual field and achieve excitation in very specific neurons. The firing of these neurons follows that of a Gabor filter, which is a normal distribution imposed over a sine wave. Once these edges have been detected, however, they then must be integrated into a concept of a figure. This, too, begins in V1 with suprgranular lateral connections in V1. Furthermore, some research postulates grouping cells that fire when any undirected, convex boundary appears in a given radius. Nesting these in a recurrent feedback cycle could allow for quick detection of a shape. Indeed, temporal evidence suggests that the cortex first quickly decides about border ownership (BOwn) and direction and then refines the border and the interior by labelling which object is which. This suggests a type of visual computation that starts at the borders, labels them, and then spreads inward, recurrently refining what is figure and ground as it goes.

3.4 Visual Summary

In summary, the biological vision system achieves a variety of tasks useful in survival (object recognition, figure segregation, motion perception) in its pathway from retina/LGN and through the visual cortex. When simplified, the system can be viewed as primarily feed-forward through various specific visual regions primed for particular stimulation and into a dorsal and ventral pathway. However, much more is going on: significant feedback and lateral connections exist, exact purpose within regions is not well-defined, intracellular differences exist, and more. Thus, computational challenges hinder our full understanding of the visual system. That said, by examining how the visual system tackles specific tasks, like figure-ground separation, we can gain inspiration for computer vision tasks.

4 The Intersection: Biologically-Inspired Computer Vision

4.1 Deep Learning

Handcrafted representations dominated computer vision for decades. They worked quite well for many applications, including image retrieval, structure-from-motion, and other canonical tasks in the field. Nevertheless, they had several shortcomings. Chiefly, it can be difficult to find the discriminative signature for some problems, and even if found, this signature can be hard to approximate. Rule-based approaches quickly grow in complexity in these cases and thus cannot be used to build a generalizable, scalable system. Consequently, learning these representations becomes critical, either through a supervised or unsupervised approach.

Deep learning has significantly outperformed other machine learning methods for learning representations. These methods have multiple levels of representation, found by composing simple, non-linear modules that each transform the representation at one level (starting with raw input) into a representation at a higher, slightly more abstract level [16]. With enough such transforms, we can learn very complex functions. As a result, neural networks are called universal function approximators. The general motivation behind each layers mirrors human-based approaches. For example, the learned features in the first layer of representation typically represent the presence or absence of edges at particular orientations

and locations in the image. The next layers detect motifs through spotting certain edge-invariant arrangements, assemble them into larger combinations corresponding to parts of objects, and detect objects as combinations of these parts. Most importantly, this process is data-driven using general purpose learning, rather than chosen by humans. It can thus detect more abstract, generalizable similarities than people can, suggesting far better performance on many tasks in computer vision than hand-crafted approaches.

Indeed, even basic uses of deep learning vastly outpaced handcrafted features. Early research in the 1980s demonstrated that simple stochastic gradient descent could be used to easily train multilayer architectures using backpropagation. This strand of research stalled due to insufficient hardware to conduct the matrix multiplications needed to determine weights for large batches of input data. The advent of fast graphics processing units (GPUs) let researchers conveniently program deep, feedforward networks (those with many more layers) and train them 10 or 20 times faster. This was first used in speech recognition, achieved record-breaking results on a large vocabulary task in 2009 [7], and were being deployed in Android phones by 2012 [10]. One type of deep, feedforward network was much easier to train and generalized much better than traditionally architected networks with full connectivity between adjacent layers. This was the convolutional neural network (ConvNet), which achieved many theoretical and practical successes while neural networks were out of favor and has shown stunningly good results in computer vision [17].

4.2 Convolutional Neural Networks

ConvNets initially process data in the form of multiple arrays. This primes them for color images, which are composed of three 2D arrays containing pixel intensities in three color channels. Four key ideas behind ConvNets take advantage of significant properties of natural signals: local connections, shared weights, pooling, and natural hierarchies [16]. Architecturally, a ConvNet is a series of stages. The first few stages consist of two types of layers: convolutional and pooling. In a convolutional layer, units are organized in feature maps. A set of weights called a filter bank connects each unit to local patches in the previous layer's feature maps. All units in a feature map share the same filter bank, and different feature maps in a layer use different feature maps. This locally weighted sum is then passed though a nonlinearity to the next layer.

Using local connections has two major benefits. First, local groups of values are often highly correlated in array data like images, so distinctive local motifs are easily detected. Second, local statistics of images and similar signals are invariant to location. A motif in one part of an image can appear anywhere else, so sharing weights between units at different locations allows the detection of the same pattern in different parts of the array. This filtering method is, mathematically, a convolution, thus giving the network its name.

While the convolutional layer detects local conjunctions of features from the previous layer, the pooling layer merges similar features together. A typical pooling unit computes the maximum of a local patch of units in a feature map. To reduce the dimension of the representation and create invariance to small alterations, neighboring pooling units take input from patches shifted by multiple columns. Multiple stages of convolution, non-linearities, and pooling are stacked and then followed by more convolutional and fully connected layers.

Deep networks exploit the compositional hierarchy inherent in many natural signals: higher-level features are obtained by composing lower-level ones. In images, edges combine to form motifs; motifs form parts; and parts form objects. Pooling lets us form representations that are invariant to changes in position and appearance in prior layers.

CNNs directly take advantage of hierarchies in the brain to record excellent performance on many different tasks in image understanding. This setup directly recalls the classic notions of simple and complex cells, respectively, in visual neuroscience. Similarly, the overall architecture resembles the visual cortex ventral pathway's LGN-V1-V2-V4-IT hierarchy. Moreover, basic ConvNet models approximate neural visual processes quite well: for example, when such models and monkeys are shown the same picture, the activations of high-level units in the ConvNet explain half the variance of random sets of neurons in the monkey's inferotemporal cortex [6].

It should come as no surprise, then, that CNNs have significantly outperformed even other neural networks on many image-based tasks. In particular, they have demonstrated state of the art results in object recognition and detection. In the former, the AlexNet posted a record-setting performance in the 2012 ImageNet Challenge [14]. Further advances like the ResNet showed the utility of additional convolutional layers in extracting features of higher abstraction, almost equaling human performance [9]. Though systems like AlexNet performed quite well, there remained vigorous debate within the computer vision community regrading the generalizability of these classification results to more difficult tasks in computer vision. Girshick et al. showed that high-capacity convolutional neural networks could be applied to bottom-up region proposals to localize and segment objects. This technique gave a 30% relative improvement over the previous best results on PASCAL VOC 2012. Moreover, when labeled training data is scarce, a network that has been pre-trained for an auxiliary task with abundant data (image classification) can then be fine-tuned for the target task (object detection). This use of transfer learning has significantly aided the widespread use of pre-trained convolutional nets on many different image-based tasks.

Interestingly, even the best advances and optimizations in CNN performance can be directly traced to advances in the neuroscience community. Physiological data from the inferotemporal cortex has demonstrated that the brain uses max-pooling, rather than winner-take-all pooling [23]. Subsequent implementation in pooling layers has provided impressive performance gains. Surround suppression has been modeled in classical receptive fields and then applied through local non-max suppression in edge detection and object recognition [11]. Rectified linear units draw directly from neuronal interrelationships and have significantly outperformed other activation functions in many visual problems [14]. Similar interplay between neuroscience and computer vision has driven other changes in architecture, layer, and neural network design, and are likely to continue to do so into the future.

Though CNN architectures have significantly improved, many drawbacks remain, particularly in the realm of representation. For one, humans and animals can learn visual concepts quite quickly, from even single training examples and fleeting exposure to small groups [3]. However, deep learning approaches require hundreds of thousands of labeled images and many different exposures. The brain deals far better with low amounts of unlabeled image data than even the best neural net architectures can at the moment. Additionally, most computer vision test benchmarks are 'closed set' problems, in which a system knows all classes it will encounter. But the vast variety of real-world objects makes it impossible to enumerate all possible negative classes of objects. Systems that cannot see all possible classes fail on even largely solved datasets, like the MNIST hand-written digit recognition set (over 99% accuracy) [25]. These signal continued difficulty in learning generalizable, flexible representations.

4.3 Mixing Representations

Mixing representations has been suggested to overcome the drawbacks of current deep learning representations. This can transfer information across tasks, form a multi-task representation, and improve single-task representations. Four major classes of approaches have been suggested: (1) fine-tuning an existing neural network on a new task, (2) duplicating an existing neural network and fine-tuning it on a new task, (3) using a pre-trained convolutional net for feature extraction and only training a new classifier, and (4) jointly re-training a pre-trained convolutional net on multiple tasks. A summary of these approaches' relative performance, determined empirically, is below [18].

	Fine Tuning	Duplicating and Fine Tuning	Feature Extraction	Joint Training
new task performance	good	good	X medium	best
original task performance	X bad	good	good	good
training efficiency	fast	fast	fast	X slow
testing efficiency	fast	X slow	fast	fast
storage requirement	medium	X large	medium	X large
requires previous task data	no	no	no	X yes

How can we use insights from the brain to more effectively integrate representations from different models? Curriculum learning is such a neurally inspired approach to mixing representations that improves convergence and minima. Guided learning helps train humans and animals; children often start from simpler examples and easier tasks. In machine learning, though, it is assumed that a model should learn from a training set of examples sampled from the same distribution as the test set. Bengio et al propose organizing examples via a sequence of training distributions of increasing difficulty. Models may initially peak on easier and simpler ones, but gradually weighting more difficult ones over training helps reach the target distribution. A series of experiments on diverse tasks demonstrates faster convergence to better local minima of a non-convex training criterion, with particular boosts in test set performance. Thus, imitating the learning process of the brain alone significantly boosts models' ability to learn better representations.

4.4 Generic Representations

Ultimately, we wish to learn generic representations: those that generalize beyond what they're trained for. Given the abundance of training data for certain tasks, supervised learning could generate excellent representations. Namely, instead of providing supervision over the desired tasks, one could provide supervision over some selected foundational tasks to improve generalization to novel tasks and abstraction. Zamir et al follow this principle in an approach inspired by the developmental stages of vision skills in humans [30]. Normally, humans see foundational, widely generalizable tasks first in building visual representations. These are then generalized to more complex tasks through interaction with the world. Similarly, the authors learn a generic 3D representation through solving two supervised proxy 3D tasks: object-centric camera pose estimation and wide baseline feature matching. They empirically demonstrate that the internal representation of the ConvNet used for the first problems generalizes well to more complex unseen 3D tasks without fine tuning and also

shows traits of abstraction. Though this recent finding demonstrates the utility of biological inspiration, it remains unclear which tasks in vision are fundamental and which are secondary. But further advances in partitioning and understanding the visual cortex could develop better, more complete 'vision complete' representations moving forward.

5 Conclusion

Representations are the basis of visual processing in both biological and computer systems. Improving them can help our human-guided approaches succeed as well as the visual cortex. In this literature review, we have laid out classical algorithms in computer vision, current models in visual neuroscience, and their intersection in modern deep learning methods. First, we showed that human-crafted algorithms can capture many of the most significant features of an image but are ultimately limited in their generalizability. Next, we reviewed the significant explanatory capacity of feed-forward models and understood continuing computational challenges hindering full understanding of the visual system. Finally, we examined the incredible effectiveness of deep learning - especially convolutional neural networks - in creating flexible, generalizable features. Some of the best optimizations to these systems take direct inspiration from visual neuroscience. This indicates that moving forward, deeper understanding of the brain is likely to continue to improve computer vision - creating a synergistic connection that will drive advances in both fields.

References

[1] Anish Acharya. "Template Matching based Object Detection Using HOG Feature Pyramid". In: *CoRR* abs/1406.7120 (2014). URL: `http://arxiv.org/abs/1406.7120`.

[2] Borislav Antić et al. "K-means based segmentation for real-time zenithal people counting". In: *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE. 2009, pp. 2565–2568.

[3] F Gregory Ashby and W Todd Maddox. "Human category learning". In: *Annu. Rev. Psychol.* 56 (2005), pp. 149–178.

[4] Lauren Barghout-Stein. *On differences between peripheral and foveal pattern masking*. University of California, Berkeley, 1999.

[5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[6] Charles F Cadieu et al. "Deep neural networks rival the representation of primate IT cortex for core visual object recognition". In: *PLoS Comput Biol* 10.12 (2014), e1003963.

[7] George E Dahl et al. "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition". In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.1 (2012), pp. 30–42.

[8] Melvyn A Goodale and A David Milner. "Separate visual pathways for perception and action". In: *Trends in neurosciences* 15.1 (1992), pp. 20–25.

[9] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[10] Geoffrey Hinton et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups". In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 82–97.

[11] Aapo Hyvärinen. "Statistical models of natural images and cortical visual representation". In: *Topics in Cognitive Science* 2.2 (2010), pp. 251–264.

[12] Herve Jegou, Hedi Harzallah, and Cordelia Schmid. "A contextual dissimilarity measure for accurate and efficient image search". In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE. 2007, pp. 1–8.

[13] Helen E Jones et al. "Differential feedback modulation of center and surround mechanisms in parvocellular cells in the visual thalamus". In: *Journal of Neuroscience* 32.45 (2012), pp. 15946–15951.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[15] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories". In: *Computer vision and pattern recognition, 2006 IEEE computer society conference on*. Vol. 2. IEEE. 2006, pp. 2169–2178.

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), p. 436.

[17] Y LeCun et al. "Handwritten digit recognition with a back-propagation network, 1989". In: *Neural Information Processing Systems (NIPS)*.

[18] Zhizhong Li and Derek Hoiem. "Learning without forgetting". In: *European Conference on Computer Vision*. Springer. 2016, pp. 614–629.

[19] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[20] NV Kartheek Medathati et al. "Bio-inspired computer vision: Towards a synergistic approach of artificial and biological vision". In: *Computer Vision and Image Understanding* 150 (2016), pp. 1–30.

[21] Jeffrey Moran and Robert Desimone. "Selective attention gates visual processing in the extrastriate cortex". In: *Front. Cogn. Neurosci* 229 (1985), pp. 342–345.

[22] Andrew Ng. "Sparse autoencoder". In: *CS294A Lecture notes* 72.2011 (2011), pp. 1–19.

[23] Maximilian Riesenhuber and Tomaso Poggio. "Hierarchical models of object recognition in cortex". In: *Nature neuroscience* 2.11 (1999), pp. 1019–1025.

[24] Silvio Savarese. *Problem Set 3*. URL: `http://web.stanford.edu/class/cs231a`.

[25] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. "Probability models for open set recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 36.11 (2014), pp. 2317–2324.

[26] AW Wolters and K Koffka. *Principles of gestalt psychology*. 1936.

[27] Elden Yu and Jake K Aggarwal. "Human action recognition with extremities as semantic posture representation". In: *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*. IEEE. 2009, pp. 1–8.

[28] Kai Yu, Yuanqing Lin, and John Lafferty. "Learning image representations from the pixel level via hierarchical sparse coding". In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, pp. 1713–1720.

[29] Sergey Zagoruyko and Nikos Komodakis. "Learning to compare image patches via convolutional neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 4353–4361.

[30] Amir R Zamir et al. "Generic 3d representation via pose estimation and matching". In: *European Conference on Computer Vision*. Springer. 2016, pp. 535–553.