Neural Networks 141 (2021) 145–159

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Unsupervised foveal vision neural architecture with top-down attention

Ryan Burt, Nina N. Thigpen, Andreas Keil, Jose C. Principe *

Computational NeuroEngineering Laboratory, Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32601, United States of America

ARTICLE INFO

Article history:
Received 8 October 2020
Received in revised form 23 February 2021
Accepted 2 March 2021
Available online 20 April 2021

Keywords:
Unsupervised Learning
Foveal vision
Top-down saliency
Deep learning

ABSTRACT

Deep learning architectures are an extremely powerful tool for recognizing and classifying images. However, they require supervised learning and normally work on vectors of the size of image pixels and produce the best results when trained on millions of object images. To help mitigate these issues, we propose an end-to-end architecture that fuses bottom-up saliency and top-down attention with an object recognition module to focus on relevant data and learn important features that can later be fine-tuned for a specific task, employing only unsupervised learning. In addition, by utilizing a virtual fovea that focuses on relevant portions of the data, the training speed can be greatly improved. We test the performance of the proposed Gamma saliency technique on the Toronto and CAT 2000 databases, and the foveated vision in the large Street View House Numbers (SVHN) database. The results with foveated vision show that Gamma saliency performs at the same level as the best alternative algorithms while being computationally faster. The results in SVHN show that our unsupervised cognitive architecture is comparable to fully supervised methods and that saliency also improves CNN performance if desired. Finally, we develop and test a top-down attention mechanism based on the Gamma saliency applied to the top layer of CNNs to facilitate scene understanding in multi-object cluttered images. We show that the extra information from top-down saliency is capable of speeding up the extraction of digits in the cluttered multidigit MNIST data set, corroborating the important role of top down attention.

1. Introduction

Neural networks and deep learning (DL) architectures are the current state-of-the-art for image classification and recognition. They have been shown to reliably distinguish between as many as 10,000 different classes of objects (Deng et al., 2009). These networks, however, currently fall well short of human capabilities in two areas: recognizing objects based on a relatively small number of examples and localizing and detecting multiple objects in a single scene. One of the culprits is that computer architectures rasterize the image into a long vector with a size given by the number of pixels in the image. This approach, universally accepted as the standard, simplifies the processing in neural network algorithms but it is a brute force procedure that loses local information.

In order to move towards more autonomous vision systems, we need an architecture that can extract features from a wide range of objects in cluttered scenes without labels. Humans have the remarkable ability to view a scene and form an overall representation in a very short time (Roelfsema, 2006). However, due to the complexity of visual scene understanding, it is reasonable to assume that humans do not process an entire scene at once, or even fixate on and process every small region in an image. Instead, the human vision system (HVS) consists of a number of brain areas that operate in a massively parallel fashion, and which often are grouped in two subsystems based on neuroanatomy (Goodale & Milner, 1992; Lee, 2011; Wang, Sporns, & Burkhalter, 2012): one for object recognition and one for spatial localization. The ventral stream, or "what" pathway, consists of visual areas V1, V2, V4, and continues to the inferior temporal cortex. Among other functions, it performs neurocomputations for identifying and semantically representing visual objects (Riesenhuber & Poggio, 1999; Treisman & Kanwisher, 1998). The dorsal stream (Fig. 1), or "where" pathway, goes through V1, V2, the dorso-medial area to the posterior parietal cortex where object locations in internal coordinates are computed, for example to enable control of eye movements (saccadic changes of gaze location) by the oculomotor system (Goodale & Milner, 1992). Humans saccade to a new fixation location at an average 3 times a second, varying with task demands and complexity of the visual scene (Fernández, Denison, & Carrasco, 2019). This is adaptive because the combination of optical and neurophysiological features in the visual pathway yields a full

* Corresponding author.
E-mail address: principe@cnel.ufl.edu (J.C. Principe).

https://doi.org/10.1016/j.neunet.2021.03.003
0893-6080/

resolution of the scene only for the foveal and parafoveal portion of the visual field, corresponding to approximately 4 to 5 degrees of the visual angle (Treisman & Kanwisher, 1998, 1998; Yarbus, 1967). Thus, saccades enable the brain to sequentially sample information from the full image field, using high acuity foveal vision (Deza & Konkle, 2020). Recently, it has been proposed that the initiation of a saccade is also the beginning of a visual processing cycle aimed at actively sensing the environment, and ultimately recognizing objects in the ventral pathway (Schroeder & Lakatos, 2009), very much like sniffing for odors. The advantage of this complex sensory motor coordination is to reduce the substantial complexity of the global visual scene to a series of simpler perceptual decisions made at the local level. Saccade control during active exploration involves the entire brain, in a dynamic complex process that is not fully characterized (Einhauser, Kruse, Hoffmann, & Konig, 2006; Norman, 2002). However, there is agreement that two processes are at play: a bottom-up process that selects "interesting" local patches based on their saliency, i.e. sharp change of luminance, contrast, color or texture in a local region of the image (Fernández et al., 2019); and a top-down process that guides the eye to relevant visual details to disambiguate the scene with respect to the current goal (Posner, Walker, Friedrich, & Rafal, 1987). These two processes are commonly called overt visual attention (Posner et al., 1987) and eliminate the need to operate with the full visual scene to save limited resources, and to facilitate the computation of motor output.

Here, we propose to design a comprehensive, end-to-end machine learning architecture that can be applied to realistic data sets employing fovea-based image processing inspired by the HVS and incorporating two pathways and an attention module including both bottom-up and top-down saliency. The foveal image patch will be delivered to a redundancy reduction object recognition algorithm (Chalasani & Principe, 2015) which extracts features in a self-organizing way (i.e. without labels) and stores them in an external memory for future use. The key characteristics of our approach relate to the algorithmic pipeline, which is controlled by foveation; the construction of a video stream with fixed retina to improve the unsupervised recognition of objects in the foveated patch; and the inclusion of top-down attention to modify the order of the foveated patches according to the analysis goal.

The paper is organized as follows: Section 1 reviews the literature, Section 2 presents the overall architecture and discusses primarily the bottom up and top down focus of attention, while Section 3 presents results with the two attention mechanisms in both realistic and synthetic data set. Section 4 presents the conclusions.

2. Related work

2.1. Attention systems

One of the hallmarks of foveal vision is its high dependence on the coordination of separate streams for object recognition and selection of salient areas in the image space. Not even the brain has enough bandwidth to understand the continuous 180 degrees video stream that reaches the eyes. Hence, it sparsifies selectively and intentionally the video input for real time interaction with the world. By using this divide-and-conquer approach, the brain can quickly process large, unwieldy images and break them into smaller pieces that require extensive computation required to extract and recognize object features from an image. This patch selection approach is compatible with the fact that the eye can only perceive the world in high resolution in a narrow cone (∼5 degrees of visual angle) centered around the center of gaze. The

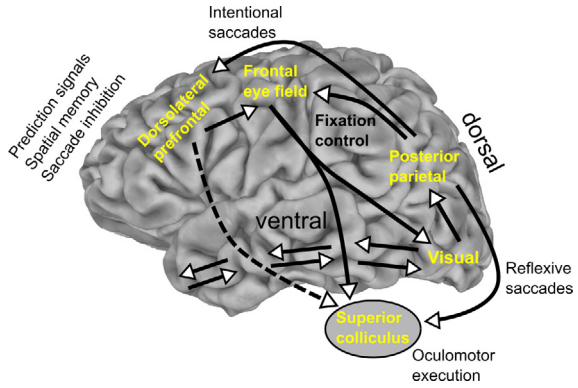

Fig. 1. The "what" and "where" paths in the human visual system.

selection of where to look (the saccades) involves the entire brain, e.g. executive, associative and motor areas (Fig. 1) and is rather complex. The discovery of simplifying assumptions for bottom-up and top-down processes are enabling inspirations for machine learning architectures, as discussed below.

Bottom-up Attention: The HVS dual-stream organization introduces new problems: finding the regions that contain relevant information, i.e. implementing visuospatial attention, on top of recognizing the patch content. Recently, machine learning methods have been proposed to suggest regions both within the structure of the network (Jaderberg, Simonyan, Zisserman, et al., 2015) or as a separate mechanism based on image features (Girshick, Donahue, Darrell, & Malik, 2014), but they still need to analyze the full image. They also require labels to choose regions that contain data most relevant for the task. Since the introduction of Itti's method in 1998 (Itti, Koch, & Niebur, 1998), saliency has become a popular way to predict visual attention in images, which could therefore be used to segment out, without labels, the interesting image regions for faster processing. Saliency is defined as the state or quality by which an object stands out relative to its neighbors. An object tends to be more salient if it is brightly colored, flashy, and altogether different from its surroundings. Employing saliency as a proxy for visual attention enables the design of unsupervised systems that can: (1) quickly select regions of interest; (2) dedicate resources for more computationally intensive processing only to the selected regions; (3) combine these representations into an overall understanding of a complex scene in much the same way the HVS works. Practically, the human visual cortex must remember and infer parafoveal and peripheral information, or use a combination of the two, to estimate regions of interest for future fixation locations. As shown from empirical research on saccadic exploratory eye movements, these future fixations will target the regions in the visual periphery (Collins, Rolfs, Deubel, & Cavanagh, 2009; Norman, 2002).

Most saliency measures work by combining a number of simple features such as color, intensity, and orientation to find distinct regions in images that could attract the human eye. Three competing views of saliency are the center–surround methods that compare a local center to a neighborhood (Burt, Santana, Principe, Thigpen, & Keil, 2016; Itti et al., 1998; Li, Zhou, Xu, Yang, & Yang, 2009; Seo & Milanfar, 2009; Walther & Koch, 2006), the global context methods that compare regions to other regions from any location in the image (Garcia-Diaz, Fdez-Vidal, Pardo, & Dosil, 2009; Goferman, Zelnik-Manor, & Tal, 2012), and the normal image methods that compare an image to a standard ideal (Achanta, Hemami, Estrada, & Susstrunk, 2009; Hou & Zhang, 2007; Kanan, Tong, Zhang, & Cottrell, 2009; Li, Levine, An, & He, 0000; Schauerte & Fink, 2010). Saliency metrics have

been used in an effort to reduce computation in image and video processing, often in lossy compression algorithms that keep high resolution data only in salient areas (Guo & Zhang, 2010; Itti, 2004). For a recent review see Cong et al. (2018).

Top-down saliency: It turns out that bottom-up attention does not explain all the visual attention mechanisms discovered in human perception. Cognitive scientists have empirically demonstrated that when the same individual views the same scene twice, she/he will change their saccadic exploration paths (Bradley, Houbova, Miccoli, Costa, & Lang, 2011; Yarbus, 1967). Therefore, this indicates that there is also a top-down attention process from the executive cortex (frontal) that changes visual processing, by conditioning the extraction of relevant information from the scene (see Fig. 1). Conceptually this selection is also based on some sort of saliency, but instead of working on image pixels, saliency is applied to more abstract sets of knowledge representations. The details are far from being fully understood (Deza & Konkle, 2020; Wolfe & Horowitz, 2017), but the existence of deep learning architectures and its multiscale learned representations opens the door for experimenting with saliency algorithms on their top layers. It has been experimentally verified (Girshick et al., 2014) that deep learning networks tend to cluster features of specific objects in their top layers. Hence, we can attempt to apply similar saliency algorithms, not to the pixels but to the feature maps created by convolution neural networks (CNNs). The Top-Down attention becomes another input to the system that can change the priority of the search in image space, by modifying the bottom-up saliency maps.

2.2. Object recognition for foveal vision

With saliency functioning as a factor that constraints guided search, the next processing goal is to form representations of and extract features from foveated patches. Therefore, an end-to-end architecture needs to integrate fovea vision and top-down attention with object recognition and memory. Therefore, the traditional deep learning single module architecture needs to be improved. In principle, the ventral stream receives patches of visual data from the fovea and builds a representation through the ventral visual pathway that is then sent to a temporary (working) memory to enable inference about the scene composition. Finally, objects that are relevant for the state of the subject are permanently stored in a visual long-term memory. It is well accepted that human memory is content addressable (Hasanbelliu & Principe, 2008), which is very efficient because it utilizes the metric of the internal representations, instead of an address bus as in our digital computers. Normally in computer vision architectures, there is no use of external memory blocks. A neural network builds its own internal long-term memory of the input data in its parameters through learning, or a short-term memory in its recurrent connections, like LSTMs (Schmidhuber, 2015). But this internal memory is not shared with other modules in the architecture. Therefore, we propose to implement explicitly content addressable memories in our architecture.

With regard to object recognition for foveal vision, we will employ a deep learning neural network. But notice that by focusing the representation on only foveated patches, rather than the entire scene, we save computation and may even improve recognition because background and other objects become distractors. By segmenting objects around highly salient points found by the attention mechanism, the network role is simplified to find invariant representations of the objects in isolation. These deep networks are generally trained on large datasets such as Imagenet (Deng et al., 2009) or CFAR100 that contain tens of thousands up to millions of labeled images. By backpropagating the errors in the class labels through the network, the network is able to learn the relevant features for predicting the label associated with the image. However, this learning becomes harder when multiple objects are contained within each image, each with its own label. In addition, supervised training requires labels for each image, which requires curating these large datasets and hampers their ability to be implemented in autonomous vision, where the number of classes are unknown.

Despite recent advances in single object recognition, classification results on image datasets with multiple objects in complex scenes require either state-of-the-art deep convolutional methods such as RESNET (He, Zhang, Ren, & Sun, 2015) or approaches that learn to extract salient features using CNN networks. One limitation is that the majority of the methods still require labels for training. In this group we consider the Spatial Transformer Network (STN), which integrates a differentiable image transform into the overall network structure that is capable of learning which features in an image best discriminate objects by their labels, focusing in on these objects accordingly (Jaderberg et al., 2015). The CNN based saliency detection architectures have become the main stream, using convolution blocks (Woo, Park, Lee, & So Kweon, 2018), an attention aware concentrated network (Li, Xing, Xu, Cai, & Cheng, 2021) as well as recurrent networks (Zhang, Wang, Qi, Lu, & Wang, 2018), perhaps using as inspiration the neurodynamics of V1 (Berga & Otazu, 2020). We can consider these as "parametric saliency" methods, because the CNN learns saliency from the full image, as opposed to the ones mentioned above that simplify the processing of information using nonparametric saliency algorithms that are much faster and never see the full picture, as inspired by the HVS. Of course, there are internediate methods that break images down into regions, then perform classification on these rather than the entire image (Gu, Lim, Arbelaez, & Malik, 2009). By fusing these region detection algorithms with the recent advances in convolutional networks, classification performance on datasets such as VOC2012 have improved by up to 30% (Girshick et al., 2014). Most of the region classification methods were designed to be trained in conjunction with deep convolution networks, such as OverFeat (Sermanet et al., 2013). OverFeat consists of a single convolutional network that is applied at multiple locations via a sliding window before producing a distribution that predicts the bounding box containing the targeted object. Alternatively, the R-CNN uses a separate region method (selective search), before separately sending these regions to a CNN for classification and then finally recombining similar regions (Girshick et al., 2014). The stacked what-and-where autoencoder is a different approach of implementing divide and conquer in a deep architecture (Zhao, Mathieu, Goroshin, & LeCun, 2016) without supervision, and exploits the same discriminative-generative principles of our approach. Despite the different paradigms, processing smaller image regions has the potential to be the next breakthrough in computer vision by reducing the brute force sliding windows in the CNNs, but further work is necessary.

Training deep learning architectures without explicit class labels has been a growing area of research (Walther, Rutishauser, Koch, & Perona, 2005). In an effort to expand these techniques beyond datasets that are fully labeled, there have been effort toward learning features based on other forms of supervision such as temporal and ego motion. Goroshin, Bruna, Tompson, Eigen, and LeCun (2015) and Wang and Gupta (Wang & Gupta, 2015) learned short term dependencies between subsequent frames in video. Agrawal et al. modeled the ego motion of the camera in order to provide a form of supervision other than labels (Agrawal, Carreira, & Malik, 2015). A different approach uses the input data as the desired response i.e., it creates a generative model of the system that created the data. Helmholtz in the XIX century wrote that the role of the visual system was exactly to model the external

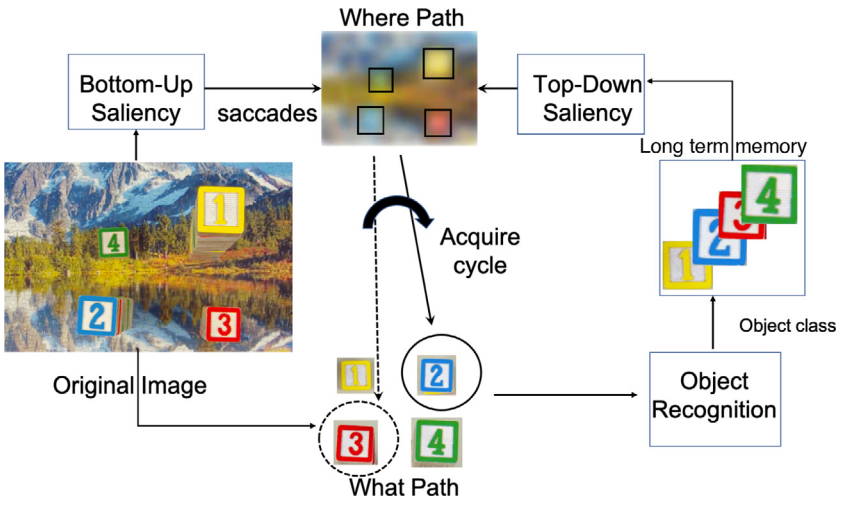

Fig. 2. Within the framework of the what–where paths, we present our vision system architecture using sequential acquire cycle for scene understanding. The "where" path is implicitly included by the search for salient image patches (bottom-up saliency), as well as by the top-down attention mechanism. The what path recognizes objects selected by the acquire cycle. Note that the explicit memory module is also needed for inference and storage of object identity.

sources that created the sensory stimuli (Von Helmholtz, 1867). A productive way to extract a generative model that created the data is through redundancy reduction principles. The deep predict coding network (DPCN) by Chalasani and Principe (Chalasani & Principe, 2015; Principe & Chalasani, 2014) uses temporal predictions to learn in an unsupervised manner features through time and build object representations of video streams. This work was later extended to the recurrent winner take all autoencoder (RTWA) (Santana, Emigh, Zerges, & Principe, 2016), which use a dual-stream autoencoder structure with a recurrent bottleneck layer to represent the current frame and predict the next frame (Fig. A.1 in the Appendix). Both methods are self-organizing generative models, but use different algorithms: the DPCN uses a naïve Bayes approach to maximize free energy on a distributed multilayer topology, while the latter uses more traditional machine learning methods (stacked autoencoder and recurrent neural networks). This framework must be integrated to learn features from unlabeled image datasets by taking advantage of the sequence structure introduced through a prescribed movement of a retina (creating a video) covering the saliency patch (Cudic & Principe, 2019).

In terms of DL applications of foviation and attention, we mention visual question answering (VQA), where the bottom-up mechanism (based on Faster R-CNN) proposes image regions, while the top-down mechanism determines feature weightings (Anderson et al., 2018). The push for explainability benefits from saliency and attention (Gilpin et al., 2019), the recent DeepFovea by Facebook speeds up and brings low-power solutions to rendering in virtual reality environments (Kaplanyan et al., 2019), and to implement gaze prediction with DL (Bazzani, Freitas, & Ting, 2011), or DeepFix (Kruthiventi, Ayush, & Babu, 2017) and efficient egocentric machine perception (Ozimek, Balog, Wong, Esparon, & Siebert, 2017).

3. Methods

Humans experience the world's static scenes through movement, whether by moving fixations across a painting or walking around a still landscape. Despite the lack of change in the physical properties of the scene, the information sent to the visual cortex through the eyes is constantly changing at a slow pace as the viewpoint is updated. The temporal coherence builds the full understanding of the scene as objects are recognized and placed into working memory as the brain searches out new fixations. Time disambiguates space, and we will take advantage of this in our approach.

3.1. Proposed architecture

As in HVS, we propose an end-to-end machine learning architecture to merge information from separate paths to simplify processing: one path for attention, which selects visually informative regions (bottom-up and top-down saliency modules), and one for representing object properties in an unsupervised manner, organizing and storing extracted objects in a dedicated long-term and working memory modules. Fig. 2 shows our proposed architecture, where the object recognition module never sees the full scene. Bottom-up saliency works as the dispatcher or the executive module for the full architecture that initiates and drives the transferring of data from the source (video or image), operating in cycles and sending the data to the object recognition module that works only on patches. Meanwhile, the extracted objects must be temporary stored in an internal canvas, a.k.a working memory, that summarizes the visual scene, and can be used for further inference. After training and during testing, the system still has the ability to speed up recognition of a scene if it is instructed by the user (or the task itself) to search for a certain object type in the scene. Currently, the top-down attention module can modify, as a prior, the bottom-up module with the characteristics of the preferred object. This architecture is rather different from CNN based saliency (Goodfellow, Bulatov, Ibarz, Arnoud, & Shet, 2013), and requires several innovations as described below.

The diagram of Fig. 3 shows the visual processing cycle, and it is complemented by Fig. 4 with a demonstration. The process starts with an image (Fig. 4A) by foveation to the center of the image (Fig. 4B), to select the highest saliency point (Fig. 4C). Multiple saliency patches may exist and they should be forwarded to the object recognition module one at a time. Hence, the process of selecting salient regions needs to be automatically repeated until the saliency map is featureless, achieved by a hyperparameter to define what is "sufficiently different" from background. Foveal vision requires centering and refocus in the highest saliency (Fig. 4D) because of the intrinsic blurring produced by the masks. Moreover, the saliency algorithm must not only find the most salient point in the image, but also the extent of the proto-object.

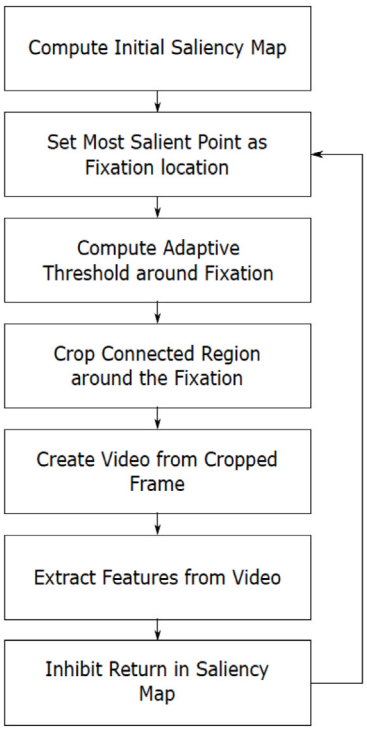

Fig. 3. The cycle of acquiring and processing information in our architecture, initiated with the saccade to the center of the image.

This is accomplished by the Gamma Saliency with its multi-scale approach because the underlying rings contain information to estimate the object extent, which leads to object segmentation matched to its size (Fig. 4E-F). Working from this base, we fine tune the saliency patch to avoid nearby objects and recompute saliency and crop a tighter patch around the object (Fig. 4G), create the video according to the selected scan (Fig. 4H), and send it to the RTWA for representation and/or recognition. After this, the saliency map is locally inhibited by applying an inverted Gaussian that corresponds to the extent of the foveal area and with an amplitude estimated from the saliency value (Fig. 4I). Below we will explain these modules one by one.

3.2. Gamma saliency

An effective attention mechanism should meet a few basic requirements to bring two pathways into a single vision system pipeline. First, the calculation should be done quickly so that the attention works to speed scene recognition, not slow it by compounding the data. Second, the bottom-up attention system should trigger the object recognition module, not in reverse order, i.e. be driven by recognizing objects and then assigning saliency scores. Since the dorsal stream of the HVS uses the peripheral, and therefore blurred, vision as the input to determine fixations, the system should be able to work only with low-level features.

To accomplish this list, we use a simple center surround saliency method that computes local differences in regions at different scales. Although high level saliency methods exist which predict human fixations very well, these often require extensive training, require full object recognition, and are slower to compute than the more classic signal processing methods. We adopt here the Gamma kernels, which have been used for target detection (Kim, Fisher III, & Principe, 1996). The circular shape of the kernel is ideal for comparing a center region to a local neighborhood, and the size of the saliency patch can easily be controlled to fit the object size through two parameters (the decay μ and the order k of the Gamma function), which allows for easy change in spatial scales.

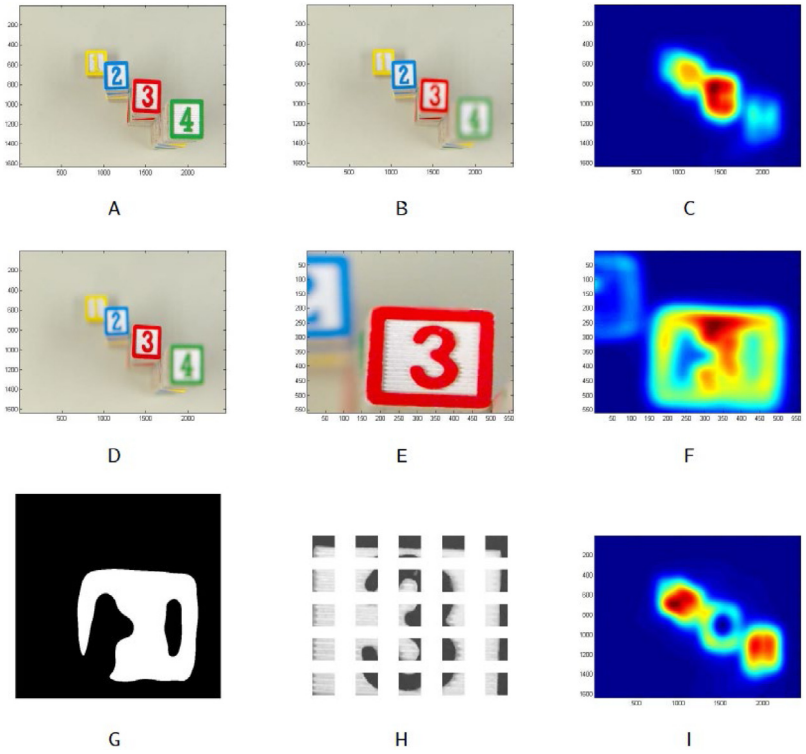

Fig. 4. A series of images showing the progression of the focus of attention algorithm. (A) The original image. (B) The image focused on the center point. (C) Saliency map created from center-focused image. (D) The image refocused on the most salient point. (E) The local patch containing the point. (F) Local saliency map. (G) The segmented object. (H) A set of scanned frames. (I) Saliency map around the new focus point with Gaussian inhibition at previously scanned locations.

Similar to the Itti method and others, the Gamma saliency (Burt et al., 2016) is based on the center surround principle: a region is salient if it is sufficiently different from the surrounding neighborhood. In order to compute these local differences, we use a 2D gamma kernel that emphasizes a central area while contrasting it with a local neighborhood through convolution:

$$g_{k,\mu}(n_1, n_2) = \frac{\mu^k}{2\pi(k-1)!}\left(\sqrt{n_1^2 + n_2^2}\right)^{k-1} e^{-\mu\sqrt{n_1^2+n_2^2}} \quad (1)$$

For this kernel, n_1 and n_2 are the local support grid coordinates, μ is the shape parameter, and k is the kernel order. Using μ and k, we can control the shape and extent of the kernel: when $k = 1$, the kernel peak is centered around the origin of the patch (exponential decay). The scale parameter μ affects the metric of the coordinates in the same way, which means that the peak of kernels k >1 is centered at k/μ away from the origin, while for $k = 1$ it controls the exponential decay. With these two parameters we can construct a 2D circularly symmetric shape that compares a center region to a surrounding neighborhood by subtracting the kernel with order $k > 1$ from the $k = 1$ kernel, i.e. $\gamma_m(n_1, n_2) = g_{1,\mu}(n_1, n_2) - g_{k,\mu}(n_1, n_2), m = k + 1$. The peak of $g_{1,\mu}(n_1, n_2)$ functions as the center of the image area to be tested, while multiple higher order kernels form the surrounding neighborhood. By adjusting the shape parameter and order of the ring kernel we can control the size and location of the neighborhood relative to the center, as well as adjust the size and location of the neighborhood relative to the center. The 2D kernel is then slid over the full image, with a stride equal to twice the radius of the highest order kernel.

One of the advantages of the Gamma kernel is that it easily allows for the estimation of the object size, by utilizing multiple rings and implementing successive subtractions over consecutive rings. For a multiscale saliency measure, we simply combine multiple kernels of different order m before the convolution stage, as shown in (2), creating a multiscale template. By creating the multiscale template before the convolution stage, we create a method which is capable of computing saliency at different scales adding little extra computation beyond a single scale $\gamma_m(n_1, n_2)$. The number of different scales is M -1.

$$\gamma_{total}(n_1, n_2) = \sum_{m=1}^{M} \gamma_m(n_1, n_2) = Mg_{1,\mu}(n_1, n_2) - \sum_{m=1}^{M} g_{m,\mu}(n_1, n_2) \quad (2)$$

In this work we do not take advantage of the recursive computation of the gamma function, because we precompute the kernel. However, if one is interested in space time saliency, the recursive computation of the gamma function becomes very efficient.

Apart from the local functions, the rest of the saliency measure is constructed similarly to the other center surround methods (Lee, Grosse, Ranganath, & Ng, 2009): the image is broken into local feature matrices of predetermined size, each matrix is convolved with the multiscale kernel, the matrices are combined and exponentiated to accentuate peaks, then post processing is performed to boost results using a Gaussian blur and a center bias.

The feature matrices to compute the saliency are composed in the CIELab color space, which has three matrices — one luminance matrix and two color opponency matrices. In CIELab space, the distance between two colors can be calculated using the Euclidean distance, which is a useful property that we take advantage of in the convolution. Each of these matrices is convolved with the multiscale gamma kernel to get the saliency measure in each channel (3). In the following equations, $*$ is the convolution operator.

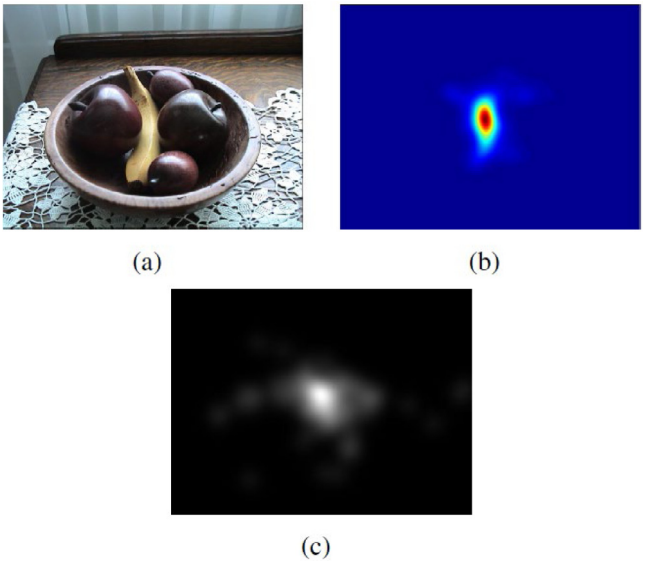

Fig. 5. Example image from the Toronto Saliency Dataset (A), saliency map produced by Gamma Saliency (B) and the ground truth fixation map (C).

$$S = \frac{|\gamma * L| + |\gamma * a| + |\gamma * b|}{3} \quad (3)$$

Once we have the overall combined saliency map, there are a few common postprocessing mechanisms used to improve results (Judd, Durand, & Torralba, 2012). First, the main peaks in the measure are accentuated by raising the combined map to a power $\alpha > 1$, applied elementwise. It is well known that humans tend to fixate on the image center, so a Gaussian weighting is applied to the center of the image where the variance of the Gaussian is dependent on the image size. Finally, to reduce the noise effects and create a more streamlined map, the map is blurred using a small Gaussian kernel (4) as

$$S = \left(S^{\alpha} G_{\sigma^2}(.)\right) * G_{0.5}(.) \quad (4)$$

Fig. 5 shows an example of the gamma saliency map on a foveated image in the Toronto data set, along with the human fixation map.

3.3. Foveal vision

There are fundamental differences between how saliency measures are tested and how the human vision system uses saliency to direct attentive exploration of the surrounding scene. Since human vision only has access to full resolution in the fovea area, the saliency module to properly mimic the human vision system must be able to find regions of interest in low resolution, outside the initial focal area. Interestingly, studies have shown that initial full previews of the scene can often hinder relevant object detection, meaning that the blurred initial glimpse can be an improvement over knowledge of an entire scene a priori (Litchfield & Donovan, 2016). However, saliency algorithms applied to digital images have per definition access to the full resolution across the field of view.

To address this crucial difference between biology and computational study, we use foveated imaging, which uses images with a clear field of focus and a blurred periphery to mimic the HVS. Foveated imaging has been used mainly for compression and faster processing (Cong et al., 2018; Geisler, Perry, & Najemnik, 2006). In addition, some saliency metrics have been tested in

multi-resolution images in an attempt to speed computation and improve results (Advani, Sustersic, Irick, & Narayanan, 2013; Itti, 2004), but study in this area is still limited. Currently the Lytro Illum camera, a light field camera (Ng, 2006), can be used for this purpose. But for comparisons with data sets in the literature, we will need to create foveated imaging by software.

To mimic the effect of the fovea, we created images that are increasingly blurred around a small high-resolution area (artificial fovea), employing the fast method developed by Geisler and Perry for images and videos in 2002 (Geisler & Perry, 2002). This method creates a variable resolution map around a center point (either pre-selected or input in real time by the user). The map is composed of a multi resolution pyramid created by first blurring the original image with a small kernel (such as 3×3), then down sampling and blurring with the same kernel, then repeating the process to create 6–7 layers. These layers are blended with weights corresponding to the distance from the center point, thus creating the newly foveated image. The foveation mechanism contains a resolution parameter that controls the distance weights, which in turn affect both the size of the fovea and the amount of blur in the periphery.

3.4. RTWA

Rather than using explicit labels, our vision module uses architectural constraints along with the structure inherent in a video stream in order to extract robust features from images. Here we use the RTWA (Santana et al., 2016), which uses a combination of a feedforward convolutional autoencoder and an RNN on the bottleneck layer that encodes a dynamic state that describes the change between consecutive frames (Fig. A.1 in the Appendix). By using the same decoder at the end of each stream, the representations are forced to project to the same space and the error can be minimized. The cost function for the RTWA is given by

$$L_t = \mathbb{E}[(x_{t-1}D(E(x_{t-1}))^2 + (x_t D(R(x_{t-1}))^2] \quad (5)$$

where x_t is the video stream at time t, the stateless encoder is E, the shared decoder is D, the RNN is R, and E denotes the expectation operator. The architecture is trained using backpropagation through time and the architecture details are in the Appendix.

As already stated, the Gamma Saliency works on still images while the RTWA uses both spatial and temporal context to form images representations. The reason we selected this combination relates to two difficulties: First, the small amount of data available for training in most complex real-world scenes. We hypothesize and confirmed (Cudic & Principe, 2019) that rasterizing an image of an object into a video of small image patches extracts more discriminatory information from the imagery than just the spatial structure of the pixels, as normally done in CNNs. The reason is that the next image patch appears naturally as a label, in a self-organizing process, that targets to learn the changes from patch to patch i.e., time helps disambiguate space. Second, a disadvantage of foveal vision is that the size of the saliency patch is not known a priori and would create difficulties for the RTWA, which is built around an autoencoder with a pre-fixed pixel size. These two aspects can be overcome with a segmentation of the saliency patch in fixed 28×28 pixel frames, which becomes the input size of the RTWA (small squares in Fig. 4H), immaterial of the size of the input image as in HVS. The attention mechanism provides, not only the salient point on which to focus, but a structured series of frames encompassing the object e.g., a video. There are multiple techniques that could be used for creating the video from image patches: scan the frame over the saliency patch, either in a circular or zig-zag path. Both provide reasonable results; the important aspect is that the scan must be kept constant such that the RTWA's learned representations do not change across the frames from training to testing. This sequence of frames leads to a more invariant set of features learned by the vision system when compared to a simple cutout of pixels, as classified by a CNN (Cudic & Principe, 2019).

In order to train the RTWA, one frame is used for input, and two frames are used as the desired response: the input frame will be used to train the autoencoder part of the RTW and the next frame trains the recurrent part for prediction (See Fig. A.1 in the Appendix). Notice that this training is unsupervised i.e., does not require a desired response. The system is trained simply to represent inputs, but given enough parameters it can represent a large class of different classes as we demonstrate in Sermanet et al. (2013). After training stops, the outputs of the bottleneck layer are the internal representations that need to be stored in the external memory for permanent storage to represent the input class with a significant savings in storage. The RTWA is capable of reproducing the input when needed by placing the memory codes at the input of the decoder part, and even predict the next image in the sequence.

3.5. Top-down saliency module

The last module of the architecture is the top-down visual attention, which implements a goal driven approach to guide the selection of scene objects. Top-down saliency is an extra input to the vision system that can be used to modify the operation of a trained network to meet some other constraint. In the HVS, the executive cortex may want to direct the visual cortex to search for a piece of information needed to complete an inference. In general, this is very difficult to achieve in machine learning, so we need to simplify what is meant by top-down attention. Here we hypothesize that the user or the environment provides the learning machine the simplest hint of what is the goal of the processing. With this extra information the idea is to modify the bottom-up processing with a top-down prior that automatically changes saliency priorities with the information contained solely in the learning machine memories. In machine learning, example applications are video question answering, where the question prioritizes the search for objects in a scene. Current solutions in these domains (Anderson et al., 2018) can still be largely improved. We propose to train a model that is capable of discriminating different input objects based on top layer network activations by leveraging the Gamma Saliency framework. Our proposed method is similar to the work of Frintrop, Backer, and Rome (2005), except that in our approach the feature maps are extracted from a network trained on the scene objects, instead of using pre-defined feature maps designed by hand, which does not qualify as a simple hint. In bottom-up Gamma Saliency, the features maps were naturally the channels of a LAB image, as shown in (3). However, to implement top-down Gamma Saliency, we propose to use a set of feature maps $C_n^i(a_1, a_2)$ from a fully convolutional neural network (CNN), where n is the feature map index for object i, and a_1, a_2 are coordinates in the activations of the convolutional layers (Erhan, Bengio, Courville, & Vincent, 2009). Unlike fully-connected layers, convolutional layers of a neural network are representation layers agnostic to the size of the input. Note that the feature maps are themselves discriminative for the objects in the training set, however, with the RTWA or DPCN, this discrimination does not necessarily mean human provided labels. Therefore, it is possible to train a system external to the CNN to discriminate amongst object classes solely based on the top layer activations of a network trained on a standard dataset (such as MNIST or Imagenet). One just needs to present an object class, find out which feature maps are activated in the convolution layers, and then learn how these maps differ from other objects' feature maps.

We selected for simplicity and to quickly test the reasonableness of the approach, a linear discrimination model that weights (w_n^i) the Gamma Saliency for each object i. We propose to learn the weights w_n^i as follows: Over a training set with an exemplar of the ith class as input, we compute each raw saliency map directly from the Gamma Saliency feature maps $\left|\gamma * C_n^i\right|$. We may need to use foveal vision to individualize which object is being observed, for multi-object imagery. We then calculate the ratio (contrast) of the mean raw saliency (a scalar) inside the bounding box $S_{I_n}^m$ of the object of interest to the mean raw saliency of the background $S_{O_n}^m$, and find the mean value across different image exemplars of the same object in the training data, i.e.

$$w_n^i = \sum_{m=1}^{M} \frac{S_{I_n}^m / S_{O_n}^m}{M} \tag{6}$$

where m is the exemplar image index for the class and n is the corresponding feature map index for each image. This scalar value gives the average weight for each feature map (n) of object i across different images of the same object. We then compute a set of Gamma Saliency feature maps S_i that can be used to distinguish among i objects in the training set, i.e.

$$S_i = \frac{\sum_{n=1}^{N} w_n^i \left|\gamma * C_n^i\right|^\alpha}{N} \tag{7}$$

where $\left|\gamma * C_n^i\right|$ is the Gamma Saliency for feature map C_n^i, α is an enhancement parameter applied elementwise, and w_n^i are the vector coefficients for object i. The procedure creates a weight matrix W that contains a vector of weights for each object of interest. Feature maps that always activate for a certain object are given a high weight, while feature maps with fewer activations are given a lower weight. Hence, each entry in this weight matrix corresponds to how highly each specific feature map activates for each class. These saliency maps S_i, when utilized in conjunction with the bottom-up saliency templates, will modify by multiplication, e.g. as a prior, their bottom-up saliency values and lead to object ranking according to a top-down goal, which potentially decrease the number of fixations necessary to find a particular object.

4. Results

This section presents the experimental results. First, we will demonstrate the results of the gamma saliency when compared with other techniques. Then we apply foveated vision to the Street View House Numbers (SVHN) dataset to access the performance of the method in a realistic environment. Finally, we will show results with mixed saliency to speed up the understanding of visual scenes with multiple objects when a ranking of object importance is given. Information on the parameters used for the attention mechanism and the classifiers are presented in the Appendix.

Validation of Gamma saliency.

Remember that our goal is to develop a methodology that uses only low-level features in the periphery (low resolution), and the goal is to be fast to compute, and agree with the human foveation. Therefore, we ultimately compare the methods with respect to human eye tracking. Saliency can be thought as object detection, and as such it is important to use detection theory as the underlying theory to compare different saliency detectors. To compare this new saliency metric with other common methods, results were computed on the Toronto dataset (Bylinskii et al., 2015) and the CAT2000 training database (Borji & Itti, 2015). The Toronto database consists of 120 images shown to 20 students for four seconds of free viewing. The CAT2000 database has 2000 images drawn from 20 different categories for a wide variety of image foregrounds and backgrounds, as well as the fixation data from 18 different observers. The observers were given the task of free viewing each image for five seconds with one degree of visual angle corresponding to roughly 38 pixels in each image. Each set of saliency maps were computed with the default set of parameters recommended by the algorithms.

For Gamma Saliency, the parameters used were $k = [1, 26, 1, 25, 1, 19]$, $\mu = [2, 2, 1, 1, .5, .5]$, and $\alpha = 5$. Note that α is also a function of the μ selected, so it was selected by performing a grid search on the integers between 1 and 20. This gives center surround differences at three scales, as in Tavakoli, Rahtu, and Heikkilä (2011), set to neighborhood sizes of 13, 25, and 38 pixels. All images are resized to 128×171 to speed processing time. The maps were then compared to the collected fixation data using the following five metrics: the area under receiver operating characteristic (ROC) curve created by Judd et al. (2012), the area under ROC curve by Borji, Sihite, and Itti (2013), the similarity measure (Le Meur & Baccino, 2013), the correlation coefficient, and the normalized scanpath saliency (Peters, Iyer, Itti, & Koch, 2005). The area under ROC curve by Judd is measured as the proportion of saliency map values above a threshold at the fixation locations to the number of values below the threshold at the fixation locations. In contrast, Borji's version of the area under ROC curve measure the proportion of true positives to false positives, which are the values in the saliency map above a threshold that do not correspond to a fixation location. The similarity measure treats each map as a distribution and computes the histogram intersection. The correlation measure is Pearson's linear coefficient between the two maps. Lastly, the normalized scan path saliency refers to the mean value of the normalized saliency map at fixation locations. In each of the metrics, the higher number indicates a better result. Also, note that these metrics only deal with finding the location of the fixation, not determining what the object is or its size. We compare the Gamma Saliency with the following competing algorithms: the original center surround Itti's work (Itti et al., 1998), graph-based saliency (GBVS) (Harel, Koch, & Perona, 2007), Torralba method (Oliva & Torralba, 2001), attention based on information maximization (AIM) (Bruce & Tsotsos, 2007), the free energy saliency (FES) (Gu, Zhai, Lin, Yang, & Zhang, 2015), Rare (Riche, Mancas, Gosselin, & Dutoit, 2012), and random center surround saliency (RCS) (Vikram, Tscherepanow, & Wrede, 2012).

To estimate the computation time, each algorithm was set to produce a saliency map sized 128×171 to ensure that algorithms that down sample do not have an inherent advantage for computation time. All times were computed on PC running Matlab R2012a on an i5-2310 clocked at 2.9 GHz. Fig. 6 shows the ROC curves for different scales. Table 1 shows the full results from comparing the saliency maps with the fixation maps in the CAT2000 database across five different metrics along with the mean time to create a saliency map from a single image in the database, with the best results for each metric in bold. Fig. 7 shows the ROC curves calculated with the Judd method for each metric. Gamma saliency performs the best in four of five metrics, with the closest competitor being GBVS. Gamma saliency is also the fastest since it is based on a convolutional filter. Table 2 shows the results for the Toronto database. Once again Gamma saliency performs the best in 4 of 5 metrics and computes the saliency maps in the fastest times.

Foveal Vision Results

Table 3 shows the results for each saliency measure on the foveated Toronto database. Gamma saliency still performs the best across 5 of the 6 metrics, which shows that it could be used in a fixation system that approximates the HVS. Interestingly, when compared with Table 2, the foveation actually improves the results obtained by most saliency measures, possibly because the

Table 1
Attention prediction results on the CAT2000 database.

Method \ Metric	ROC (Judd et al., 2012)	ROC (Borji et al., 2013)	Similarity (Le Meur & Baccino, 2013)	Correlation	NSS (Peters et al., 2005)	Time (s)
Itti (Itti et al., 1998)	.700	.570	.377	.206	.258	0.25
AIM (Bruce & Tsotsos, 2007)	.772	.628	.437	.335	.497	1.04
Oliva and Torralba (2001)	.770	.619	.437	.324	.448	1.20
GBVS (Harel et al., 2007)	.844	.642	.498	.486	.510	1.05
FES (Gu et al., 2015)	.812	.576	.562	.628	.368	0.29
RARE (Riche et al., 2012)	.822	.643	.466	.408	**.511**	1.37
RCS (Vikram et al., 2012)	.763	.593	.431	.292	.352	14.9
Gamma (Burt et al., 2016)	**.852**	**.676**	**.592**	**.633**	.468	**0.21**

Table 2
Attention prediction results on the Toronto database.

Method\ Metric	ROC (Judd)	ROC (Borji)	Similarity	Correlation	NSS	Time (s)
Itti	.712	.597	.384	.275	.341	0.28
AIM	.746	.632	.403	.363	.479	1.10
Torralba	.684	.600	.374	.292	.360	0.78
GBVS	.848	.677	.488	.570	**.638**	1.03
FES	.847	.586	.520	.572	.446	**0.21**
RARE	.785	.625	.477	.551	.489	1.39
RCS	.747	.609	.431	.414	.413	15.8
Gamma	**.862**	**.695**	**.588**	**.581**	.546	**0.21**

Table 3
Attention prediction results on the foveated Toronto database.

Method\ Metric	ROC (Judd)	ROC (Borji)	Similarity	Correlation	NSS
Itti	.737	.597	.403	.314	.369
AIM	.794	.657	.433	.458	.561
Torralba	.784	.650	.433	.469	.539
GBVS	.839	.664	.502	.603	**.594**
FES	.846	.571	.487	.536	.403
RARE	.841	.656	.525	.632	.591
RCS	.819	.629	.517	.592	.517
Gamma	**.858**	**.684**	**.607**	**.649**	.583

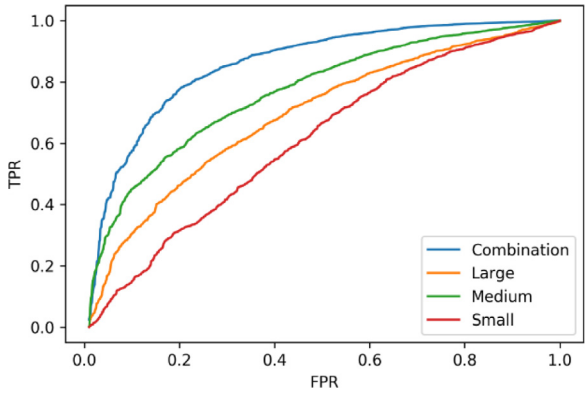

Fig. 6. ROC curves for different scales of gamma saliency on the Toronto Saliency Dataset.

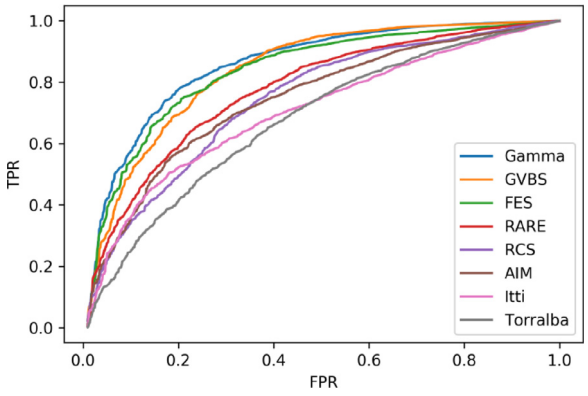

Fig. 7. ROC curves on the Toronto Saliency Dataset. The images contain 3.8 times as many negative locations as positives.

addition of a blur and center bias improves results, as shown in previous studies.

Classification Results: Street View House Numbers

The Street View House Numbers (SVHN) dataset offers a tough localization and classification challenge. It consists of over 73,000 training digits and over 23,000 testing digits in images from Google Street View. There are two main formats to the database — one cropped into 32 × 32 MNIST like digits with the additions of color, variable contrast, and some confusing data and the full images which contain extensive backgrounds and multiple digits in addition to the challenges in the cropped format.

In this dataset, the foveate vision implemented with Gamma saliency is used to localize and separate each number, turning the task into one resembling MNIST rather than training a single CNN to recognize both the number of digits and the classification of each. By using this divide-and-conquer approach the unsupervised feature extraction is able to focus on representing relevant parts of the image rather than trying to explain both the digit and the noise, leading to more useful features.

Fig. 8 shows the foveate vision working on an example SVHN image. It first segments a salient area (Fig. 8B-D), then breaks that area up into the digits that compose the two objects found in that location (Fig. 8E-J). By separating the digits in this manner, we are able to extract features that correspond to a single object at a time, rather than attempting to learn a network that explains an entire scene with multiple labeled objects.

Table 4 shows the classification accuracies, segmentation accuracies, and time per training epoch on the fully image dataset with the enlarged bounding boxes created by the procedure outlined in Geisler and Perry (2002). Segmentation accuracies are

Table 4
SVHN results on the bounded box dataset.

Method	Unsupervised				Supervised		
	RWTA FOA	TDN FOA	Autoencoder	VAE	CNN Full	CNN FOA	STN Full
Classification	92.51	92.28	15.58	17.46	94.47	96.06	96.30
Segmentation	83.67	83.67	76.92	76.92	76.92	83.67	NA
Time (sec)	12638	2015	2372	2784	1426	1195	NA

Table 5
SVHN results on the full image dataset.

Method	Unsupervised				Supervised		
	RWTA FOA	TDN FOA	Autoencoder	VAE	CNN Full	CNN FOA	STN Full
Classification	73.30	71.59	5.13	8.34	68.15	80.58	28.03
Segmentation	72.43	72.43	NA	NA	NA	72.43	NA
Time (sec)	22658	2943	3689	4016	2397	1506	4549

calculated by dividing the intersection of the true and predicted bounding boxes by their union. The STN and CNN results are reported by the authors (Goodfellow et al., 2013; Jaderberg et al., 2015) respectively, and do not include segmentation data or time information. We implemented an autoencoder (Vincent, Larochelle, Bengio, & Manzagol, 2008) and a stacked autoencoder (Zeng, Yu, Wang, Li, & Tao, 2017), which were not competitive. RWTA and TDN (the recurrent part of the RWTA is substituted by a time delay neural network (Waibel, Hanazawa, Hinton, Shikano, & Lang, 1989)), which greatly simplifies the training because it can be trained by backpropagation. In our model that generates a priori the video from a sequence of images it is easy to set the TDN size, and explains the similar results. Both vastly outperform traditional unsupervised learning strategies that do not use attention. In addition, the focus of attention also improves the performance of a CNN close to the state-of-the-art results reported by the STN. This means that the focus of attention can be used with any neural network architecture for vision.

Fig. 9 shows the ROC curves for the bottom-up attention system on the full SVHN dataset. These curves were created by setting the digit locations as fixations and computing the ROC using the Judd method (Bylinskii et al., 2015).

Most results reported on the SVHN dataset use cropped digits, and even the few ones that try to classify the full address at once use an enlarged bounding box instead of the full image. In this next test, however, we use the full collected images with no additional preprocessing nor information about the image contents. This means that our bottom-up attention mechanism based on foveate vision implemented with the Gamma saliency must find the address location, segment, then identify each digit for success. This is a much harder problem than simply classifying boxed digits since it combines the problems of localization and classification in a paradigm that does not have fixed output size.

Table 5 shows the classification, segmentation, and timing results for the full dataset. In this case, adding an attention mechanism is imperative to success as the task involves classifying numbers in what are often extremely large background compared to the size of the numbers. It is gratifying to see that pure unsupervised techniques using space–time information are outperforming the standard deep architectures based on CNNs in curated datasets. This shows that the curated datasets used by the ML community are still very artificial and do not portray the reality of autonomous vision. It is also important to note that the foveal vision helps as a preprocessor for CNNs, because it avoids the cluttered background, even though there are still errors in the saliency algorithms.

Multi-digit, Cluttered MNIST Search with Top-Down Saliency

To test the combined bottom-up and top-down framework, we used a cluttered multi-digit MNIST environment (Cudic, 2016; Cudic, Burt, & Principe, 2018). This environment is useful for a proof of concept since it is built upon a dataset for which we know classification performance. This ensures that the complexity in the problem comes from the multiple objects and the clutter found in the scene. For a first test, the environment consists of 128 x 128 pixels canvas with 5 random MNIST digits placed at random locations, without the clutter. The end-to-end architecture employed the standard MNIST 28 x 28 pixels input and was a simple convolutional network trained on the original MNIST data base, augmented with the bottom up and top-down Gamma Saliency previously described. The bottom-up saliency used parameters $k = (1, 9)$ and $\mu = (.2, .5)$ because the digits are small, and the background is uniform, so a single scale is sufficient. The 64 feature maps for each of the 10 digits were extracted at the end of the final convolutional layer before pooling. The top-down saliency mask used the same k and μ parameters, and the discriminatory weights in (6) and (7) were computed after the neural network was trained in the full original MNIST data set. During operation, the user first selects the number to be searched (the external hint), and then the system automatically biases the bottom-up saliency according to this extra information as described above.

The first goal is to test the "search" of the top-down saliency against the original bottom-up version of Gamma Saliency by comparing the number of saccades required by each algorithm to find target digits, specified by the user. Fig. 10 shows an example image of the data base, as well as the bottom-up and top-down saliency measures for one image, where the target digit is "1". With the bottom-up saliency, each digit was found in an average of 3 saccades. Adding the top-down framework reduces the number of average saccades to 1.73 saccades per digit, meaning the saliency was successfully biased towards the desired object we want to find.

The more demanding task adds clutter as shown in Fig. 11. In this case, the system is answering a simple question for each new input image: does the image contain a specific (randomly selected) digit?

The test uses three different architectures, shown in Fig. 11. The first is a network given the full image augmented with a one-hot vector with the target digit (Fig. 11A). The second network preprocesses the image with bottom-up Gamma Saliency and classifies the patch using a pre-trained MNIST network (Fig. 4B). Finally, the third architecture uses both bottom-up and top-down Gamma Saliency to find the target image before sending the extracted image patch to the same pre-trained MNIST classifier (Fig. 4B). We created 100,00 training images, as well as 25,000 validation and 25,000 test images to train the classifier in Fig. 11.A, but the classifier for Fig. 11B does not need retraining because the digits are from the original MNIST data set.

Table 6 shows the results for this experiment. The full network trained on the scene has trouble combining information from the

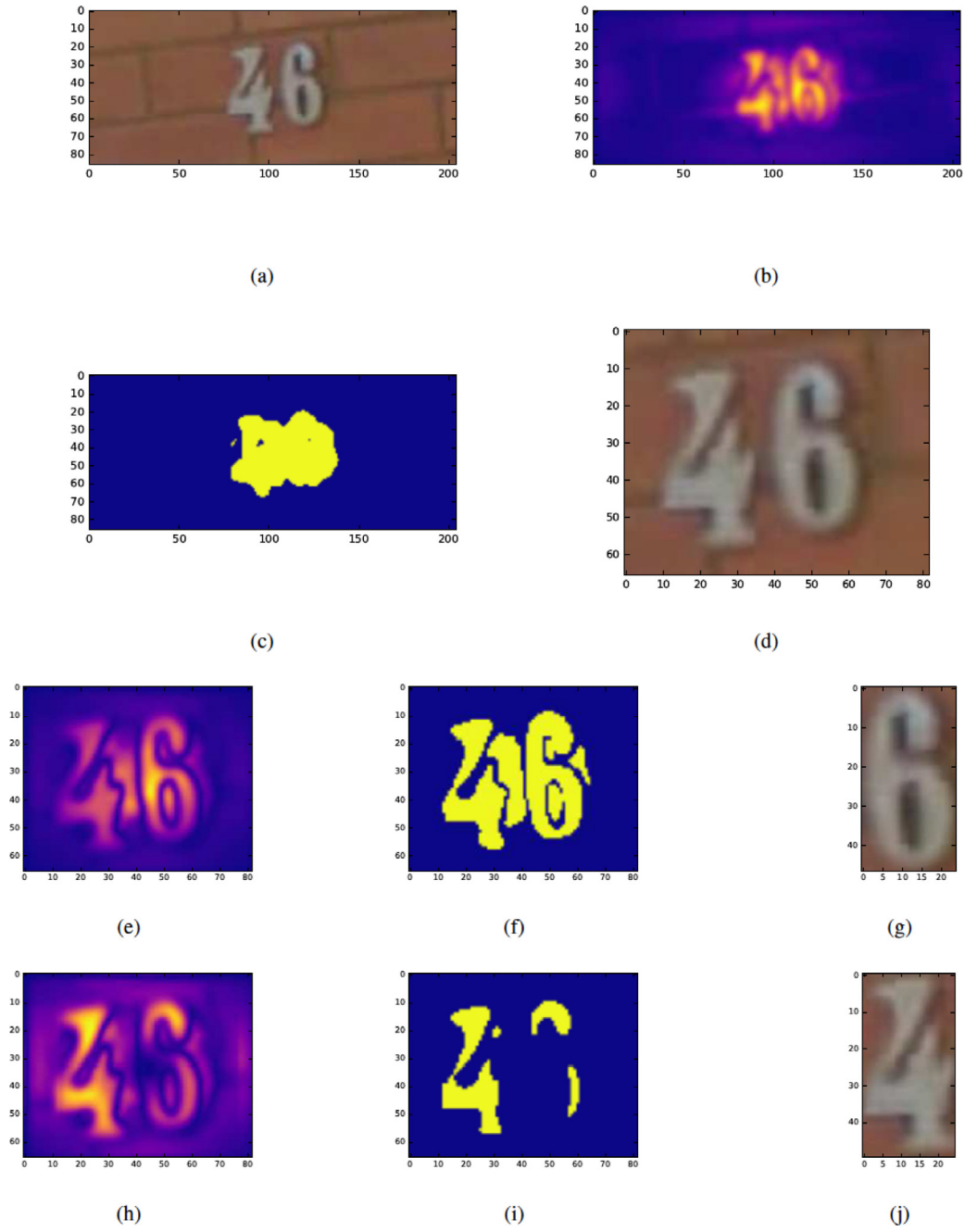

Fig. 8. Initial SVHN image (A), saliency map produced by Gamma Saliency (B), thresholded saliency map (C), the cropped patch around the house numbers (D), the initial saliency map from the crop (E), the thresholded version of that map (F), the patch extracted around the object with the highest saliency (G), the saliency map with return inhibition around the most salient point (H), the thresholded map (I), and the cropped second object (J).

cluttered image and the target, correctly identifying whether the target is present in less than 72% of test images. The bottom-up saliency network performance in one saccade is only 61% correct targets, and it improves slowly with two saccades. Since this attention mechanism is purely input driven, it often misses the target digit, passing saccades of irrelevant digits to the classifier.

The top-down saliency, however, improves on both the bottom-up saliency and the full canvas neural network, correctly identifying at the first saccade with 84% accuracy that the target number is present. With 2 saccades, the result is over 90%. Therefore, we conclude that the top-down information biases the bottom-up saliency sufficiently to make the selected digit more probable, which speeds up the understanding task. If the number of saccades is increased to 5 (the number of digits on each canvas), the bottom-up saliency network approaches the performance of the mixed saliency network, as expected. The reason the mixed saliency network performance is higher when compared with the conventional network is the benefit of being

Table 6
Classification results for finding the target digit.

Input	Full network	Bottom-up	Top-down
Full image	72.3	NA	NA
One saccade	NA	61.4	84.1
Two saccades	NA	68.3	92.3

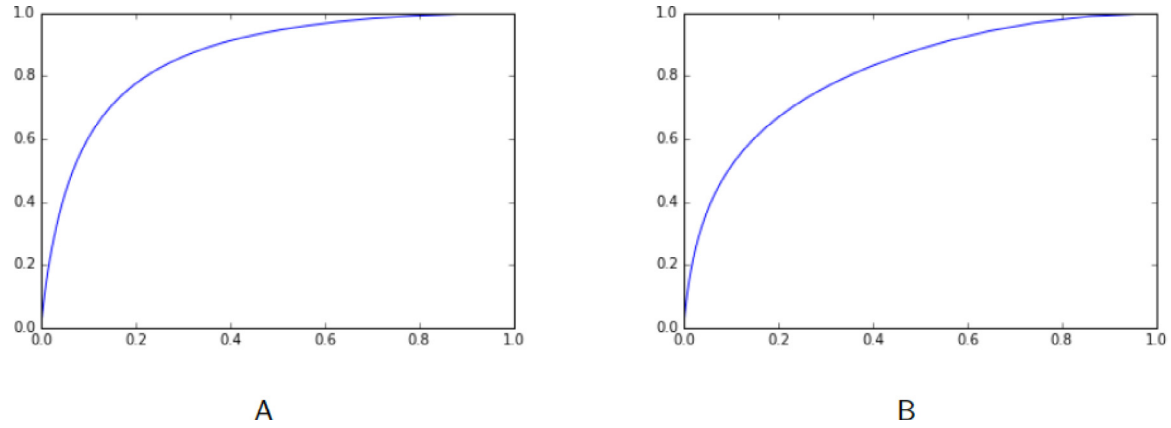

Fig. 9. ROC curve for finding the bounding box containing all numbers from the SVHN data set using foveated vision (A) and without blurring and centering (B).

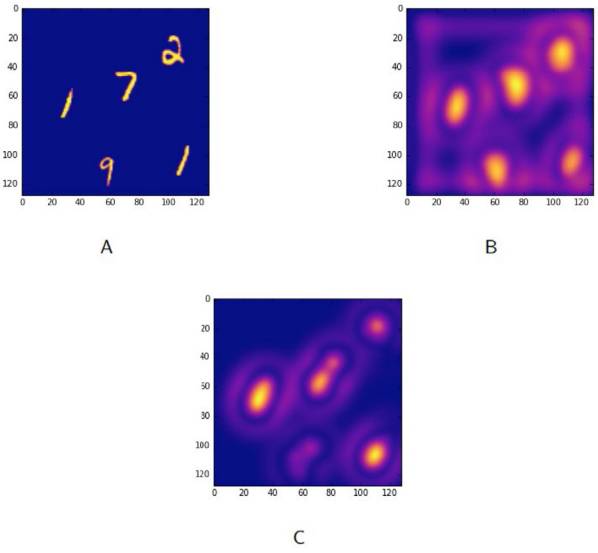

Fig. 10. Example feature maps from the convolutional layers of the MNIST classifier, with saliency weighted for the digit 1.

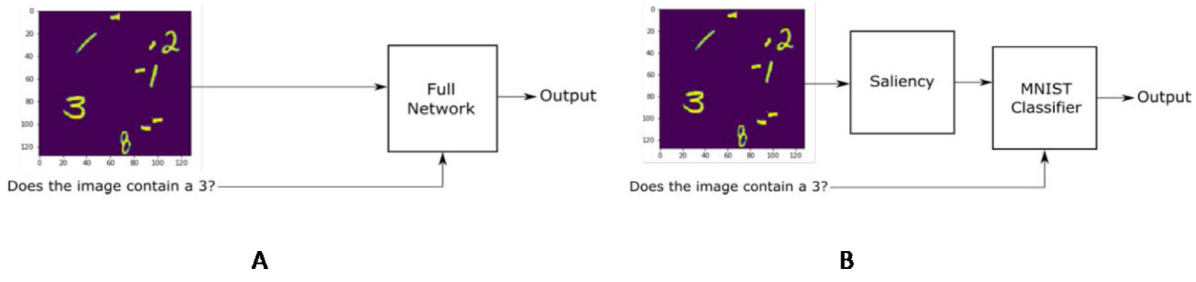

Fig. 11. Training of the full network (A) and the end-to-end architecture with saliency (B).

able to remove background information, while also concentrating on the correct foreground data. Recall that an added advantage is that the classifier of the proposed end-to-end architecture is also much smaller and can run in microprocessors for Internet of Things (IoT) applications, since the network only processes a 28 x 28 pixel input instead of learning the entire canvas. The details of both networks are presented in the Appendix.

5. Conclusion

In this paper we propose an architecture that mimics the function of two fundamental mechanisms of the human vision system: a saliency-based method for the spatial attention (approximating functions of the HVS dorsal pathway) and an object recognition network (approximating the function of the HVS ventral pathway). By separating these pathways, we can achieve greater computational efficiency by quickly selecting subregions of the image for full processing, as well as improve the feature extraction by eliminating non-discriminatory data.

In addition to removing irrelevant information, the attention pathway is a way to also create videos from still images, which adds data augmentation to the new architecture and in spite of being unsupervised, gets performance close to fully supervised techniques. Here we use a RTWA network that has both spatial

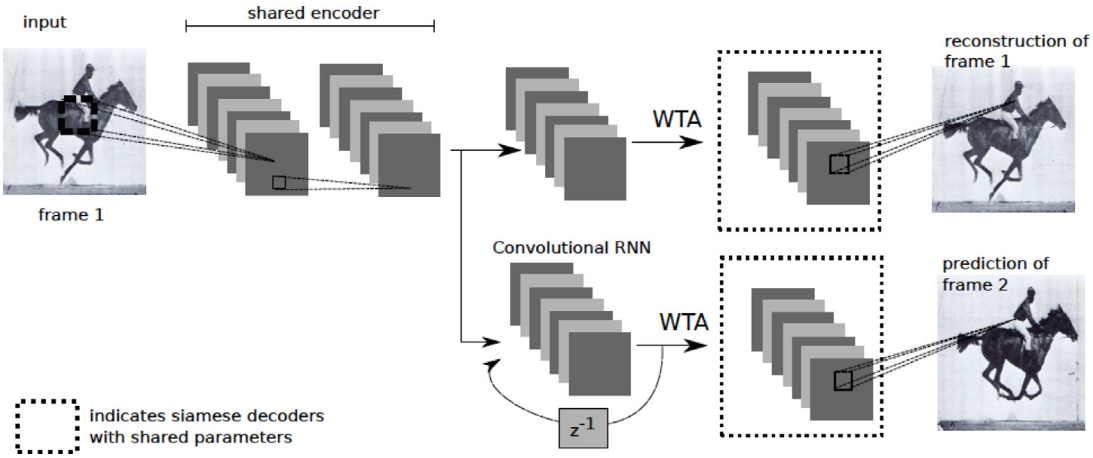

Fig. A.1. Block diagram of the RTWA.

and temporal pathways to learn more discriminant data structure. This combination extracts robust features that cluster the relevant information in objects without the need of labels. One major drawback to spatial time features is the computation time, especially the temporal component that uses an RNN. To mitigate these issues, we simply combined the features from different frames using learned weights (TDN), which produced similar results at much faster speeds. We believe this was possible since the underlying data in the frames were the same, i.e. the videos were created simply with rotation. In a true video, the RNN would be necessary to learn the temporal relationship between the frames.

The inclusion of the top-down saliency is a major advantage in conditions of high background clutter, and in scene understanding with many objects. Fovea vision (Fig. 3) is the dispatcher for scene understanding, and it must go over all of the salient points which can be very computation intensive and the visitation order is dictated by pixel information only. Top-down attention is able to change this ordering, when the information is made available to the system. Basically, the vision system must use fovea vision to individualize each object, and store the extracted object features in an external memory (content addressable memory — CAM) as shown in Fig. 2. This problem requires the use of extra information to disambiguate the images, which the brain does naturally through attention mechanisms. Since we are using machine learning methods, we employ a traditional CNN trained on a large dataset to discriminate among classes, although the system does not need human labels as stated above. Our proposal to extract object specific information in the top layers of CNNs uses Gamma saliency applied to the top layers of a trained deep learning architecture. The method is simple, effective, and it shows that the local activation of units in the top layer of CNNs indeed carry global information about the image type in a distributed manner. We only use a linear model to make the decision, and nonlinear functions would likely improve the results. Very few computer vision systems today could achieve the performance of our end-to-end architecture. This is an exciting avenue of future research.

Future work includes extending and testing the proposed method in scene understanding, particularly occluded objects where deep learning is worse than human performance. In addition, future research may aim to further improve the attention mechanism to make it focus not only on areas of the image that are locally different, but ones that offer the greatest scene understanding when combined with the information already extracted from the image. Finally, further enhancing the bio-realism and autonomy of the architecture requires inclusion of some form of supervision to extract object affordances for scene understanding, using reinforcement learning techniques.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was partially funded by ONR-N00014-14-1-0542, N00014-18-1-2306 and DARPA FA9453-18-1-0039.

Appendix

All networks were created using Keras [92] using the Theano backend. We adopt the notation that conv [N, w, s] denotes a convolutional layer with N filters of size w x w, and stride s; fc [N] is a fully connected layer with N units; and max[s] is a s x s max-pooling layer with stride s. The CNN model is conv [64,9,9]- max [2]- conv [32,7,7] - max [2] - fc [256] – fc [10] with rectified linear units following each weight layer and a softmax layer at the end for classification. For the CNN with STN, the STN network is max [2]- conv [20,5,5] – max [2]-conv [20,5,5] – fc [50] – fc [6].

The spatial encoder in the RWTA model is conv [64,3,3] – conv [64,3,3], while the convolutional time encoder is conv [64,3,3] with a time sequence of 5 frames (Fig. A.1). A linear SVM is learned on the latent states of the RWTA to produce the classification scores.

Since the images and digits in this dataset are uniformly sized, a single scale attention model was used. The center kernel has an order of $k = 1$ and a shape parameter of $\mu = 0.2$. The neighborhood kernel has an order of $k = 9$ and $\mu = 0.5$. A single frame was extracted from each image since each image contained only a single digit and contained no location information.

Each network was trained for 500 epochs on a Tesla K80 GPU.

A.1. SVHN

The CNN model is: conv [48,5,1] – max [2] – conv [64,5,1] – conv [128,5,1] -max [2]-conv [160,5,1]- conv [192,5,1] – max [2] – conv [192,5,1] – conv [192,5,1] – max [2] – conv [192,5,1] – fc [3072] -f c [3072] – fc [3072], with rectified linear units following each weight layer, followed by five parallel fc [11] and softmax layers for classification. There are 11 outputs in the final layer to account for the digits 0–9 and an extra class for noise classification. The ST-CNN has a single spatial transformer before the first convolutional layer of the CNN model the STNs

localization network architecture is: conv [32,5,1] – max [2] – conv [32,5,1]-fc [32] – fc [32].

The spatial encoder in the RWTA model is conv [64,3,3] – conv [64,3,3] – conv [128,3,3], while the convolutional time encoder is conv [64,3,3] with a time sequence of 5 frames. A linear SVM is learned on the latent states of the RWTA to produce the classification scores.

Since the images and digits in this dataset have different sizes, a multi scale attention model was used. The center kernels have an order of k = 1 and a shape parameter of $\mu = 0.1$; $\mu = 0.3$; and $\mu = 0.8$. The neighborhood kernel has an order of k = 13; k = 9; k = 5 and $\mu = 0.3$; $\mu = 0.5$; $\mu = 0.7$. These parameters were used to create the initial saliency maps and find the main fixation points. For the local saliency, the largest scale was removed to focus on finer details, leaving a two-scale kernel. Each network was trained for 10000 epochs on a Tesla K80 GPU.

A.2. Cluttered MNIST

The network in Fig. 11A was a conv [32,3,3] – conv [64,3,3] – pool [2,2]- conv [64,3,3] – conv [128,3,3] – pool [2,2] – conv [128,3,3] – conv [256,3,3] – pool [2,2] – fc [500] – fc [100] – fc [2] with the one-hot target vector concatenated to the features before the first dense layer.

The network in Fig. 11b used for the saliency-based architectures is conv [32,3,3] – conv [32,3,3] – pool [2,2] – conv [64,3,3] – conv [64,3,3] – pool [2,2] – fc [100] – fc [10].

References

Achanta, R., Hemami, S., Estrada, F., & Susstrunk, S. (2009). Frequency-tuned salient region detection. In *IEEE computer vision and pattern recognition* (pp. 1597–1604).

Advani, S., Sustersic, J., Irick, K., & Narayanan, V. (2013). A multi-resolution saliency framework to drive foveation. In *Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on* (pp. 2596–2600). IEEE.

Agrawal, P., Carreira, J., & Malik, J. (2015). Learning to see by moving. In *Proceedings of the IEEE International conference on computer vision* (pp. 37–45).

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., et al. (2018). Bottom-up and top-down attention for image captioning and visual question answering. arXiv:1707.07998v3.

Bazzani, L., Freitas, N., & Ting, J. (2011). Learning attentional mechanisms for simultaneous object tracking and recognition with deep networks. In *International Conference on machine learning*.

Berga, D., & Otazu, X. (2020). Modeling bottom-up and top-down attention with neurodynamic model of V1. *Neurocomputing*, http://dx.doi.org/10.1016/j.neucom.2020.07.047.

Borji, A., & Itti, L. (2015). A large scale fixation dataset for boosting saliency research. arXiv:1505.03581.

Borji, A., Sihite, D. N., & Itti, L. (2013). Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Transactions on Image Processing*, *22*(1), 55–69.

Bradley, M., Houbova, P., Miccoli, L., Costa, V., & Lang, P. (2011). Scan patterns when viewing natural scenes: Emotion, complexity, and repetition. *Psychophysiology*, *48*(11), 1544–1553.

Bruce, N., & Tsotsos, J. (2007). Attention based on information maximization. *Journal of Vision*, *7*(9).

Burt, R., Santana, E., Principe, J. C., Thigpen, N., & Keil, A. (2016). Predicting visual attention using gamma kernels. In *IEEE international conference on acoustics, speech and signal processing* (pp. 1606–1610). IEEE.

Bylinskii, Z., Judd, T., Borji, A., Itti, L., Durand, F., Oliva, A., et al. (2015). Mit saliency benchmark. http://saliency.mit.edu/.

Chalasani, R., & Principe, J. C. (2015). Context dependent encoding using convolutional dynamic networks. *IEEE Transactions on Neural Networks and Intelligent Systems*, *26*(9), 1992–2004.

Collins, T., Rolfs, M., Deubel, H., & Cavanagh, P. (2009). Post-saccadic location judgments reveal remapping of saccade targets to non-foveal locations. *Journal of Vision*, *9*(5), 29 1–9.

Cong, R., Lei, J., Fu, H., Cheng, M., Lin, W., & Huang, Q. (2018). Review of visual saliency detection with comprehensive information. arXiv:1803.033912v2 [cs.CV].

Cudic, M. (2016). Mnistvqa. https://github.com/mihaelcudic/mnistvqa.

Cudic, M., Burt, R., & Principe, J. (2018). A flexible testing environment for visual question and answering with performance evaluation. *Neurocomputing*, *291*, 128–135.

Cudic, M., & Principe, J. (2019). Using a Recurrent Kernel Learning Machine for Small-Sample Image Classification. In *IEEE Proc. IEEE IJCNN 2019, Budapest*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: a large-scale hierarchical image database. In *IEEE computer vision and pattern recognition* (pp. 248–255).

Deza, A., & Konkle, T. (2020). Emergent properties of foveated perceptual systems. arXiv:2006.07991v1.

Einhauser, W., Kruse, W., Hoffmann, K., & Konig, P. (2006). Differences of monkey and human overt attention under natural conditions. *Vision Research*, *46*(8–9), 1194–1209.

Erhan, D., Bengio, Y., Courville, A., & Vincent, P. (2009). *Visualizing higher-layer features of a deep network (Vol. 1341)* (p. 3). University of Montreal.

Fernández, A., Denison, R., & Carrasco, M. (2019). Temporal attention improves perception similarly at foveal and parafoveal locations. *Journal of Vision*, *19*(1), 12.

Frintrop, S., Backer, G., & Rome, E. (2005). Goal-directed search with a top-down modulated computational attention system. In *Pattern recognition* (pp. 117–124). Springer.

Garcia-Diaz, A., Fdez-Vidal, X. R., Pardo, X. M., & Dosil, R. (2009). Saliency based on decorrelation and distinctiveness of local responses. In *Computer analysis of images and patterns* (pp. 261–268). Springer.

Geisler, W. S., & Perry, J. S. (2002). Real-time simulation of arbitrary visual fields. In *Proceedings of the 2002 symposium on eye tracking research & applications* (pp. 83–87). ACM.

Geisler, W. S., Perry, J. S., & Najemnik, J. (2006). Visual search: The role of peripheral information measured using gaze contingent displays. *Journal of Vision*, *6*(9), 1.

Gilpin, L., Bau, D., Yuan, B., Bajwa, A., Specter, M., & Kagal, L. (2019). Explaining explanations: An overview of interpretability of machine learning. cs.AI.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE computer vision and pattern recognition* (pp. 580–587).

Goferman, S., Zelnik-Manor, L., & Tal, A. (2012). Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(10), 1915–1926.

Goodale, M. A., & Milner, A. D. (1992). Separate visual pathways for perception and action. *Trends in Neuroscience*, *15*(1), 20–25.

Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., & Shet, V. (2013). Multi-digit number recognition from street view imagery using deep convolutional neural networks. ArXiv preprint arXiv:1312.6082.

Goroshin, R., Bruna, J., Tompson, J., Eigen, D., & LeCun, Y. (2015). Unsupervised learning of spatiotemporally coherent metrics. In *Proceedings of the IEEE international conference on computer vision* (pp. 4086–4093).

Gu, C., Lim, J. J., Arbelaez, P., & Malik, J. (2009). Recognition using regions. In *IEEE computer vision and pattern recognition (CVPR)* (pp. 1030–1037).

Gu, K., Zhai, G., Lin, W., Yang, X., & Zhang, W. (2015). Visual saliency detection with free energy theory. *IEEE Signal Processing Letters*, *2*(#10), 1552–1555.

Guo, C., & Zhang, L. (2010). A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE Transactions on Image Processing*, *19*(1), 185–198.

Harel, J., Koch, C., & Perona, P. (2007). Graph-based visual saliency. *Proceedings NIPS*.

Hasanbelliu, E., & Principe, J. (2008). Content addressable memories in reproducing kernel Hilbert spaces. In *Proc. IEEE Workshop on machine learning for signal processing*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. arXiv:1512.03385v1 [cs.CV].

Hou, X., & Zhang, L. (2007). Saliency detection: A spectral residual approach. In *IEEE computer vision and pattern recognition* (pp. 1–8).

Itti, L. (2004). Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing*, *13*(10), 1304–1318.

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (11), 1254–1259.

Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015). Spatial transformer networks. *Advances in Neural Information Processing Systems*, 2017–2025.

Judd, T., Durand, F., & Torralba, A. (2012). A benchmark of computational models of saliency to predict human fixations.

Kanan, C., Tong, M. H., Zhang, L., & Cottrell, G. W. (2009). Sun: Top-down saliency using natural statistics. *Visual Cognition*, *17*(6–7), 979–1003.

Kaplanyan A. Sochenv, A., Leimkuhler M. Okunev, T., Goodall, T., & Rufo, G. (2019). DeepFovea: Neural reconstruction for foveated rendering and video compression using learned statistics of natural videos. research.fb.com.

Kim, M., Fisher III, J. W., & Principe, J. C. (1996). New cfar stencil for target detections in synthetic aperture radar imagery. In *Aerospace/defense sensing and controls* (pp. 432–442). International Society for Optics and Photonics.

Kruthiventi, S., Ayush, K., & Babu, R. (2017). Deepfix: A fully convolutional neural network for predicting human eye fixations. *IEEE Transactions on Image Processing*, *26*(9), 4446–4456.

Le Meur, O., & Baccino, T. (2013). Methods for comparing scan paths and saliency maps: strengths and weaknesses. *Behavior Research Methods*, *45*(1), 251–266.

Lee, B. (2011). Visual pathways and psychophysical channels in the primate. *Journal of Physiology*, *589*(Pt 1), 41–47.

Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning* (pp. 609–616). ACM.

Li, J., Levine, M. D., An, X., & He, H. (0000). Saliency detection based on frequency and spatial domain analysis. In *BMVC 2011*.

Li, P., Xing, X., Xu, X., Cai, B., & Cheng, J. (2021). Attention-aware concentrated network for saliency prediction. *Neurocomputing*, *429*, 199–214.

Li, Y., Zhou, Y., Xu, L., Yang, X., & Yang, J. (2009). Incremental sparse saliency detection. In *IEEE image processing (ICIP)* (pp. 3093–3096).

Litchfield, D., & Donovan, T. (2016). Worth a quick look? Initial scene previews can guide eye movements as a function of domain-specific expertise but can also have unforeseen costs. *Journal of Experimental Psychology, Human Perception and Performance*.

Ng, R. (2006). *Digital light field photography* (Ph.D. dissertation), Stanford University.

Norman, J. (2002). Two visual systems and two theories of perception: An attempt to reconcile the constructivist and ecological approaches. *Behavioral and Brain Sciences*, *25*(01), 73–96.

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, *42*, 145–175.

Ozimek, P., Balog, L., Wong, R., Esparon, T., & Siebert, J. (2017). Egocentric Perception using a Biologically Inspired Software Retina Integrated with a Deep CNN. In *ICCV 2017 workshop on egocentric perception, interaction and computing*.

Peters, R. J., Iyer, A., Itti, L., & Koch, C. (2005). Components of bottom-up gaze allocation in natural images. *Vision Research*, *45*(18), 2397–2416.

Posner, M., Walker, J., Friedrich, F., & Rafal, R. (1987). How do the parietal lobes direct covert attention? *Neuropsychologia*, *25*(1A), 135–145.

Principe, J., & Chalasani, R. (2014). Cognitive architectures for sensory processing. *Proceedings of the IEEE*, *102*(4), 514–525.

Riche, N., Mancas, M., Gosselin, B., & Dutoit, T. (2012). Rare: A new bottom-up saliency model. In *IEEE Int. conf. image proc. Orlando FL*.

Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, *2*(11), 1019–1025.

Roelfsema, P. (2006). Cortical algorithms for perceptual grouping. *Annual Review of Neuroscience*, *29*, 203–227.

Santana, E., Emigh, M., Zerges, P., & Principe, J. C. (2016). Exploiting spatio-temporal structure with recurrent winner-take- all networks. ArXiv e-prints, Oct.

Schauerte, B., & Fink, G. A. (2010). Focusing computational visual attention in multi-modal human–robot interaction. In *International conference on multimodal interfaces* (p. 6). ACM.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117.

Schroeder, C., & Lakatos, P. (2009). Low-frequency neuronal oscillations as instruments of sensory selection. *Trends in Neuroscience*, *32*(1), 9–18.

Seo, H. J., & Milanfar, P. (2009). Nonparametric bottom-up saliency detection by self-resemblance. In *IEEE computer vision and pattern recognition workshops* (pp. 45–52).

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. ArXiv preprint arXiv:1312.6229.

Tavakoli, H. R., Rahtu, E., & Heikkilä, J. (2011). Fast and efficient saliency detection using sparse sampling and kernel density estimation. In *Image analysis* (pp. 666–675). Springer.

Treisman, A., & Kanwisher, N. (1998). Perceiving visually presented objects: Recognition, awareness, and modularity. *Current Opinion in Neurobioloy*, *8*, 218–226.

Vikram, T., Tscherepanow, M., & Wrede, B. (2012). A saliency map based on sampling an image into random rectangular regions of interest. *Pattern Recognition*, *45*(9), 3114–3124.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on machine learning* (pp. 1096–1103). ACM.

Von Helmholtz, H. (1867). Handbuch der physiologischen Optik: Mit 213 in den Text eingedruckten Holzschnitten und 11 Tafeln. Voss.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *37*(3), 328–339.

Walther, D., & Koch, C. (2006). Modeling attention to salient proto objects. *Neural Networks*, *19*(9), 1395–1407.

Walther, D., Rutishauser, U., Koch, C., & Perona, P. (2005). Selective visual attention enables learning and recognition of multiple objects in cluttered scenes. *Computer Vision and Image Understanding*, *100*(1), 41–63.

Wang, X., & Gupta, A. (2015). Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision* (pp. 2794–2802).

Wang, Q., Sporns, O., & Burkhalter, A. (2012). Network analysis of corticocortical connections reveals ventral and dorsal processing streams in mouse visual cortex. *Journal of Neuroscience*, *32*(13), 4386–4399.

Wolfe, J., & Horowitz, T. (2017). Five factors that guide attention in visual search. *Nature Human Behaviour*, *1*(3), 0058.

Woo, S., Park, J., Lee, J.-Y., & So Kweon, I. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision* (pp. 3–19).

Yarbus, A. (1967). *Eye movements in vision*. Plenum Press.

Zeng, K., Yu, J., Wang, R., Li, C., & Tao, D. (2017). Coupled deep autoencoder for single image super-resolution. *IEEE Transactions on Cybernetics*, *47*(1), 27–37.

Zhang, X., Wang, T., Qi, J., Lu, H., & Wang, G. (2018). Progressive attention guided recurrent network for salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 714–722).

Zhao, J., Mathieu, M., Goroshin, R., & LeCun, Y. (2016). Stacked what-where auto-encoders. arXiv:1506.02351.