FINDING OBJECTS IN COMPLEX SCENES WITH TOP-DOWN AND BOTTOM-UP INFORMATION

By

RYAN M. BURT

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2017

To my parents

ACKNOWLEDGMENTS

Thanks to everyone that helped me during my time in graduate school. I would especially like to thank my advisor, Dr. Jose Carlos Santos Carvalho Principe. His patience and advice was invaluable. The support of Dr. Andreas Keil was also very helpful. I would also like to thank Dr. Tan Wong for being on my committee.

During my undergraduate studies at Kettering University two professors were extremely supportive of my burgeoning academic studies - Dr. Doug Melton and Dr. Dan Russell. I would like to thank them both for stoking my curiosity beyond what was required for the classes. The experience of conducting research and engineering a product made me look past simple engineering and toward graduate school.

My time at the University of Florida was mostly shaped by the time I spent in CNEL. I would like to thank Austin, Evan, and Goktug for helping to ease my transition to lab and grad school in general. Matt, Catia, Carlos, and Eder were helpful as my contemporaries, always ready to share ideas and complaints. In addition, thank you to everyone else that was in CNEL during my time here.

I'd also like to thank the friends that helped me survive my years in Florida: Marty, David, Matt, and David. Thank all of you for keeping me grounded during my time here and being willing to listen to me rant endlessly about my work. I appreciate your efforts to make me go outside, even if you occasionally had to drag me.

Lastly, I'd like to thank my family: my parents and siblings (along with their families). Thank you for being supremely understanding of my time here, and not always asking when I was going to graduate. Being able to escape Florida to go back north was a necessary reset, especially when that involved playing with nieces and a nephew.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

FINDING OBJECTS IN COMPLEX SCENES WITH TOP-DOWN AND BOTTOM-UP INFORMATION

By

Ryan M. Burt

December 2017

Chair: Jose C. Principe
Major: Electrical and Computer Engineering

Humans have the ability to view a scene and form an overall representation in a remarkably short length of time. However, due to the complexity of visual search, it is reasonable to assume that humans do not fixate on and process every small region in an image. Instead, the entire image is quickly sent through a pyramidal processing mechanism that selects fixation regions for more attention. In contrast, current computer vision methods such as convolutional neural networks employ a sliding window based method that take small patches across the entire image. By selecting regions in the image which are interesting and processing only those we can avoid convolving over the entire image, which should reduce the correlated dimensions in the network by skipping over large uniform regions. In the same way that convolving learned filters over an image is a step beyond scanning pixel-by-pixel, processing still images as videos of small frames composed of visually interesting regions could be a further step that simply discards large regions of the image that have little to no effect on classification.

Rather than convolving across the entire image, it is possible to take inspiration from the human vision system and only process interesting regions that have significance to the viewer. Humans use a mixture of bottom-up attention (edges and bright colors that attract eyes) and top-down goals (searching for specific objects) in order to choose where to fixate next. These top-down goals are informed by both memories and current emotional states that affect the visual cortex and cause it to respond to different stimuli.

Using the human vision system as inspiration, we use a novel saliency metric based on gamma kernels as the basis of a simple bottom-up saccade and fixation system that is capable of finding interesting objects in scenes that is both faster to compute and more accurate than alternatives. By processing images with this focus of attention before forming representations with the network of choice, it is possible to both speed the computation and improve classification results by removing background data. We can also use the attention to augment the data and provide a form of supervision to the feature extraction network without using explicit labels.

Lastly, we can create a visual search mechanism by using the convolutional layers in the feature extraction network to pre-process the image, then learning a set of weights on the feature maps that correspond to specific objects. By doing this, we are sharing information between the attention and the feature extraction methods, where the output of each informs the input of the other. This mimics the "what-where" pathways in the brain, where the pathways are separate but interconnected.

CHAPTER 1
INTRODUCTION

Humans have the ability to view a scene and form an overall representation in a remarkably short length of time. However, due to the complexity of visual search, it is reasonable to assume that humans do not fixate on and process every small region in an image. Instead, the entire image is quickly sent through a pyramidal processing mechanism that selects fixation regions for more attention. In contrast, current computer vision methods such as convolutional neural networks employ a sliding window based method that take small patches across the entire image. By selecting regions in the image which are interesting and processing only those we can avoid convolving over the entire image, which should reduce the correlated dimensions in the network by skipping over large uniform regions. In the same way that convolving learned filters over an image is a step beyond scanning pixel-by-pixel, processing still images as videos of small frames composed of visually interesting regions could be a further step that simply discards large regions of the image that have little to no effect on classification.

However, choosing these regions is not a trivial task. Humans have a complex saccade and fixation system that fixates on and forms a representation of a single object at a time before composing these fixations into an entire scene. To drive the saccades and fixations in the human vision system (HVS), there is a mixture of bottom-up information that comes from the environment as well as top-down goals that come from memory and current emotional state [1].

Rather than convolving learned filters across an entire image which may consist of many objects in a complex scene, a computer vision system can emulate the HVS and fixate on certain regions, process them serially to form a representation, and fuse these together into an understanding of the scene. In order to accurately choose important fixation locations, it will be necessary to use both the information contained in the scenes as well as prior information from learned memories and appropriate responses to the objects contained in the fixations.

1.1 Human Vision System

The first component of the HVS that we are attempting to emulate are the eyes. The eyes provide full access to high resolution data only in a small region called the fovea where the focus of attention is centered, approximately 3 degrees of visual angle around the point at which gaze is directed during a given moment in time. Thus, in addition to bottom-up saliency, the human brain must remember and infer parafoveal and peripheral information, or use a combination of the two, to compute targets of interest for future fixation locations. As shown from empirical research on saccadic exploratory eye movements, these future fixations will target the blurry, low resolution regions in the visual periphery.

The visual cortex consists of distinct regions that serve separate but interconnected functions. V1 functions as an edge detector and encodes the spetial information in a scene. V2 takes the inforamation from V1, finds slightly more complex patterns, creates associations, and begins forming representations to be stored in the memory. V4 encodes the salient stimuli and is a large driver of attention, feeding the selective attention signals back through V2 and V1 to drive fixations [2], [3]. V5 helps drive eye movement by tracking and predicting the location of moving objects.

Current theory holds that the visual system is broken into two separate pathways, the ventral stream consists of V1, V2, V4, and continues through to the inferior temporal cortex, while the dorsal stream goes through V1, V2, the dorsomedial area and then the posterior parietal cortex [4]. The ventral stream is responsible for forming the representations and identifying objects, while the dorsal stream is associated with location of objects and controlling the saccades.

The ventral stream receives input mainly from the fovea and uses this high resolution area to form a representation of the fixated object [5]. It also has strong connections to the temporal cortex, which stores the long terms memories. In contrast, the dorsal stream takes information from across the field of vision and encodes a spatial map of low-level features and checks for any motion that would change these features.

Neuroscientists have shown that images with affective stimuli cause large event related potentials in both the occipital lobe (where the visual cortex is located) and the temporal cortex, where the memories are stored [6]. The stronger the stimuli is, the higher the potentials evoked in the visual areas of the brain, meaning visual attention may be allocated to stimuli depending on the significance of those stimuli [1]. These potentials act on a working memory that combines knowledge of the scene at hand with clues from a working memory to help drive attention in the visual cortex [7]. By combining the spatial attention from features in the scene with a working memory that predicts stimuli and provides reactions, the HVS achieves an attention system that is both bottom-up and top-down.

1.2 Computer Vision Systems

In recent years, convolutional networks have become the standard for image classification [8]. These networks have relatively few shared parameters compared to traditional neural networks, which allows researchers to train them on larger datasets with more images and data without overfitting or underfitting as quickly. However, these models work by convolving small learned image patches across an entire test image and many correlated dimensions are included in the network, which can also slow learning and complicate classification [9]. The complexity of the databases and the networks is a growing problem that is only practical when using fast GPUs.

When extending the image analysis beyond static images to videos, or image time series, CNNs have to be extended by also convolving across the time dimension [10]. An alternative approach is to use a state space or dynamic network to handle the dynamics of the CNNs across time. Principe and Chalasani proposed in [11] an implementation of this idea as an unsupervised network called the Deep Predictive Coding Network (DPCN). DPCNs are multilayer sparse coding networks where each layer interchangeably encodes either the sparse activations or the variance of the activations of the layer below. During training, DPCNs use the codes from the previous times as well as activation from upper layers as a prior, thus exploiting temporal structure during learning with top-down flow through the hierarchy.

The deep predictive coding network (DPCN) is a hierarchical generative deep model whose architecture is shown in Figure 1-1 [12]. The DPCN learns an unsupervised representation of the input data using bottom-up and top-down flow through the deep architecture. Thus, this is different from standard architectures where the information only flows from the bottom to the top layers of the model; DPCN combines that feedforward (or bottom-up) flow with top-down priors learned through time. This constrains the representation to a higher order of temporal smoothness.

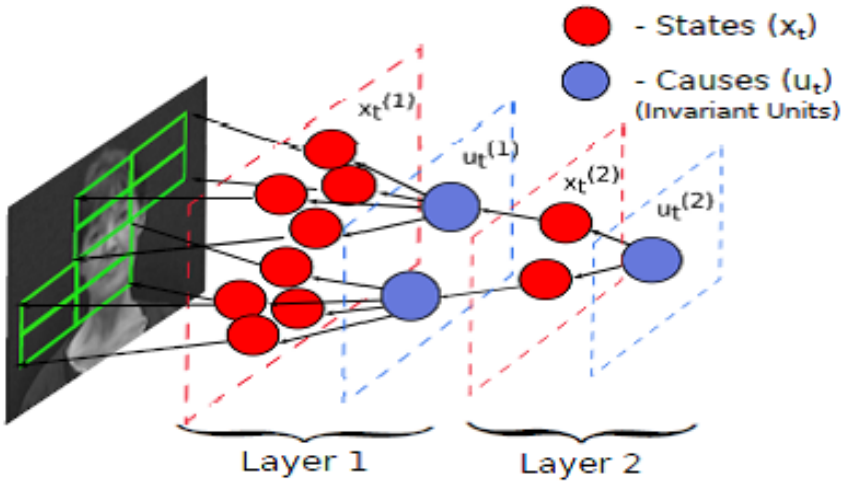


Figure 1-1. The DPCN architecture.

Unfortunately, DPCNs work directly on small image patches (28x28), so they do not scale up well to practical images. They have been integrated with CNNs [12] and their performance was shown to be better than traditional techniques. However, convolution with the full image seems brute force. All parts of input images have to be represented with convolutional sparse codes, even if that referenced part is background information common to all in the images in the dataset. We propose to avoid this problem by using the framework of visual attention.

1.3 Using Regions for Image Classification

Despite these recent advances in image processing, classification results on image datasets with multiple objects in complex scenes has stagnated when using only these convolutional

methods. Recently, research has begun into breaking images down into regions, then performing classification on these rather than the entire image [13]. 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% [14].

Most of the region classification methods proposed at this time were designed to be trained in conjunction with deep convolution networks, such as OverFeat [15]. 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. This was the first attempt at using a convolutional network trained on labels for localization.

Alternatively, the R-CNN proposed combining a region proposal method (selective search [16]) before separately sending these regions to a CNN for feature extraction, an SVM for classification and then finally recombining similar regions [14]. This proved to be effective, although it requires the selective search to suggest as many as 2000 possible objects, making it an extremely slow brute force system. This was sped up by removing the SVM [17] and later the separate region proposal method [18]. This and other similar methods such as YOLO [19] and ION [20] effectively use brute force, classifying objects over many heuristically defined bounding boxes. Despite the brute force, these methods have proven effective at localizaing objects within scenes given enough labeled data.

However, the HVS does not exhaustively scan a scene, but instead picks a few interesting regions to process fully. Following this, processing smaller regions of images has the potential to be the next breakthrough in computer vision by reducing the brute force sliding windows in the CNN-based methods. To do this, we will need to predict the location of objects without exhaustively classifying each region of an image.

1.4 Bottom-Up Attention

One way to predict fixation regions for a computer vision system is the saliency map. Saliency is defined as the quality by which an objects stands out relative to its neighbors, so it

is able to be computed from an image without any prior knowledge. In addition, it has been shown to correlate with fixation data from eye-tracking studies [21].

Since the introduction of Itti's method in 1998 [22], saliency has become a popular way to predict visual attention in images and could therefore be used to segment out the interesting regions for faster processing. 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 [22], [23], [24], [25], the global context methods that compare regions to other regions from any location in the image [26], [27], and the normal image methods that compare an image to a standard ideal [28], [29], [30], [31], [32], [33].

In recent years, many methods have been applied to saliency such as graph theory [34], information theory [35], deep networks [36], and Bayesian frameworks [37]. It is important to note, though, that these methods solely use bottom-up information and use no prior information about the scene and the target in choosing fixation locations. This means that the visual attention is predicted solely based on features contained in the current scene. Despite this limitation, these measures have been shown to be very effective in predicting saccades in a free viewing task.

Applications for these saliency measures have mostly been limited to image compression for speeding up visual processing [38], [39]. However, Torralba and Oliva showed that combining a bottom-up saliency measure with prior knowledge about objects and common background scenes can speed object detection [37], [40]. By building on this principle, it should be possible to create a system that uses bottom-up saliency as an input to help detect interesting objects in complex scenes, especially when a top-down attention mechanism is added to help direct it, such as in the human vision system.

1.5 Outline

Taking inspiration from the HVS, we plan to implement a system that is capable of finding stimuli in a multi-resolution environment, fixating on this stimuli, forming a representation of it and saving it to a memory, then driving the next fixation via a search. Chapter 2 details the current work done on bottom-up drivers of visual attention. Chapter 3 then discusses the interfacing between the bottom-up attention with feature extractors and other neural networks and tests this combination on benchmark datasets in different domains of image processing. Chapter 4 completes the loop by adding a top-down search to the attention mechanism, and tests this in a search paradigm. Finally, Chapter 5 concludes this dissertation.

CHAPTER 2
BIO-INSPIRED FOCUS OF ATTENTION

This chapter details the creation of a bottom-up focus of attention mechanism that can be used to separate and identify objects in scenes. To create this system, we start by introducing a new center-surround saliency metric based on the 2d gamma kernel that is capable of providing reliable saliency information at high speeds. After creating this, we test it in multi-resolution environments that simulate the fovea and periphery found in the human vision system. Next, we incorporate this saliency measure into a saccade and fixation system that identifies salient objects, fixates on them, then breaks them into small frames for processing by the DPCN. Finally, we show that this system is capable of both speeding computation times and improving results by ignoring unnecessary information that can be found in the background of images.

2.1 Saliency Measures

There are many existing saliency measures that compute saliency in a variety of ways. These include center-surround methods, information theory based methods, and methods that calculate deviations from a normal, expected image. Some popular bottom-up saliency metrics are detailed in the following subsections.

2.1.1 Itti-Koch Saliency

In 1998 Itti introduced the original saliency metric [22]. This model extracts color, intensity, and orientation data from the image, then compares it to neighboring regions and weights the contrast. Finally, this data is combined then normalized to create the overall saliency map. A winner-take-all structure was added to the final map to predict the next fixation point, with an inhibition added to avoid returning to previous fixation points.

2.1.2 AIM Saliency

In 2005 Bruce and Tsotsos created Attention Based on Information Maximization (AIM) [41]. AIM defines salient regions in an image by finding regions that maximize the information

gained in comparison to the neighboring regions. Shannon's information is computed on comparing patches of the image project onto a basis obtained by performing ICA.

2.1.3 Torralba Saliency

In 2006 Torralba et al. defined a bottom-up salience as the probability of finding a set of local features within an image [37]. Each color channel is passed through a set of steerable pyramids, then the distribution of the local features is modeled with a power-exponential distribution. Torralba then combined this with prior knowledge of target locations in similar scenes, but this paper only uses the bottom-up portion of the saliency metric for comparison.

2.1.4 GBVS

In 2006 Harel et al. created Graph-Based Visual Saliency (GBVS) [34]. GBVS extracts a set of three features (intensity, color, and orientation) and creates a full connected graph over each feature map. Each graph is treated as a Markov chain to build an activation map derived from the pairwise contrast between pixels. Mass is concentrated on each activation map by introducing another Markov chain, and this accumulation forms the overall saliency map.

2.1.5 FES

In 2011 Tavakoli et. al used a Bayesian framework to create a center-surround saliency method that they called Fast and Efficient Saliency (FES) [42]. FES uses kernel density estimation to compute a feature distribution. This distribution is convolved by a circular averaging filter to emphasize distinct features in each region, with the scaling of the circular feature change to find salient objects of multiple sizes.

2.1.6 RARE2012 Saliency

In 2012 Riche et al. updated the RARE algorithm to its current state [43]. RARE2012 extracts color data using a PCA decomposition and orientation data before applying different filters and attenuations to create 24 maps. These maps are combined using a multi-scale rarity mechanism that accentuates both locally contrasted and globally rare regions.

2.1.7 RCS

In 2013 Erdem and Erdem created Region Covariance Saliency by computing saliency based on covariance matrices extracted from local image patches [44]. The distance between each matrix was computed and saliency values are assigned based on a dissimilarity measure.

2.2 Gamma Saliency

Outside of saliency, gamma kernels have been used for target detection [45], specifically in the Gamma CFAR detector for synthetic aperture radar [46]. The circular shape of the gamma kernels is ideal for comparing a center region to a local neighborhood, and the size of each can easily be controlled through the use of two parameters, which allows for easily changed scales. In addition, the gamma kernel has many properties such as the ability to be computed recursively and the smoothness of the neighborhood that make it well suited to signal processing methods.

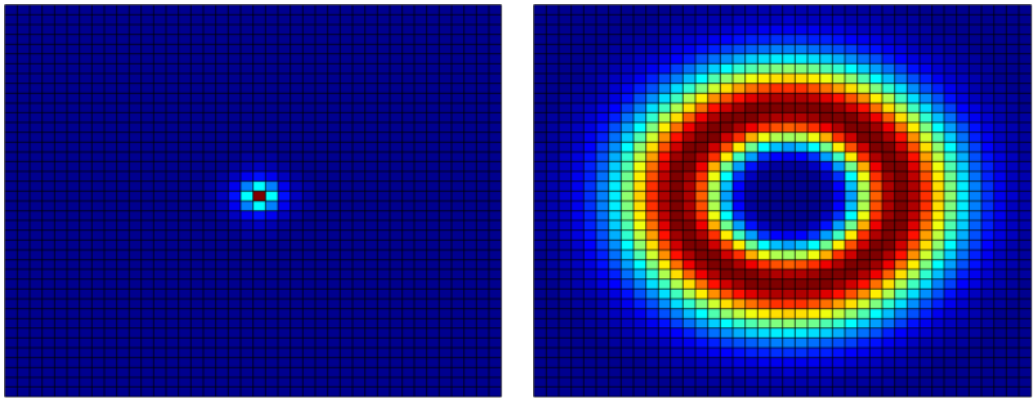
Similar to the Gamma CFAR, Itti method, and others, Gamma saliency is based on the center surround principle: a region is salient if it is different from the neighborhood. In order to compute these local differences, we use a 2D gamma kernel that emphasizes a center while contrasting it with a local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

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

For this kernel, n_1 and n_2 are the local support grid, μ is the shape parameter, and k is the kernel order. Using μ and k, we can control the shape of the kernel: when $k = 1$ the kernel peak is centered around zero. For larger kernel orders, the peak is centered k/μ away from the center. In addition, smaller values of μ will increase the bandwidth of the peak.

With these parameters we can construct a 2D shape that compares a center region to a surrounding neighborhood by subtracting a kernel with order $k > 1$ from a kernel with order $k = 1$. The 1st order kernel functions as the center while the higher order kernel forms the surrounding neighborhood. By adjusting the shape parameter and order of the neighborhood kernel we can control the size and location of the neighborhood relative to the center, and

as well as adjust the size of the center by using the shape parameter for the center kernel. Figure 2-1 shows an example of a center kernel with parameters $\mu = 1$ and $k = 1$ along with the surround kernel with parameters $\mu = 1$ and $k = 10$.



A B

Figure 2-1. Visual representation of the center (A) and surround kernels (B).

For a multiscale saliency measure, we simply combine multiple kernels of different sizes before the convolution stage 2–2. A kernel with a larger center scale is subtracted by a surround kernel with a larger and further removed neighborhood, effectively searching for larger objects by comparing more overall area in the image. By summing all the kernels before the convolution stage, we create a system which is capable of computing saliency at different scales without adding extra computation beyond a simple summation. The kernel summation is described in 2–2, where all k for even m are 1 to create the center kernels. The number of different scales is $m/2$.

$$g_{total} = \sum_{m=0}^{M-1} = (-1^m) g_m(k_m, \mu_m) \tag{2–2}$$

In addition to the circular shape of the neighborhood, the gamma kernel has other useful properties that can be exploited. The shape of the neighborhoods is smooth, which is in contrast to other methods which choose neighborhoods that sample at a fixed radius. Also, the gamma kernel can be computed recursively. Though we don't make use of the recursive

computation here in favor or pre-computing the kernel for speed, the recursive property could be exploited to extend this method to work in a temporal structure such as video saliency.

With this local difference measure, the rest of the saliency measure is constructed similarly to the other center surround methods: the image is broken into feature matrices, each matrix is convolved with the multiscale kernel, the matrices are combined and exponentiated to accentuate peaks, then postprocessing is performed to boost results using a Gaussian blur and a center bias.

The feature matrices are composed of the CIELab color space, which has three matrices - one luminence matrix and two color opponency matrices. In CIELab space, the distance between two colors can be calculated using simply 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 2–3. In the following equations, $\bullet$ is the convolution operator.

$$S = \frac{|g \bullet L| + |g \bullet a| + |g \bullet b|}{3} \tag{2–3}$$

Once we have the overall combined saliency map, there are a few common postprocessing mechanisms used to improve results. First, the main peaks in the measure are accentuated by raising the combined map to a power $\alpha > 1$. Next, it is well known that humans tend to fixate on the center of images, 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 effects of noise and created a more streamlined map, the map is blurred using a small Gaussian kernel 3–4 as in [42].

$$S = (S^{\alpha} G(\sigma^2)) \bullet G(.5) \tag{2–4}$$

2.2.1 Eye-Tracking Results

To compare this new saliency metric with other common methods, results were computed on the Toronto dataset [35] and the CAT2000 training database [47]. 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 $\alpha = 5$. This gives center surround differences at three scales, as in [42], set to neighborhood sizes of 13, 25, and 38 pixels. All images are resized to $128x171$ to speed processing time. α was selected by performing a grid search on the integers between 1 and 20.

The maps were then compared to the collected fixation data using these five metrics: the area under receiver-operating characteristic (ROC) curve created by Judd [48], the area under ROC curve by Borji [49], the similarity measure [50], the correlation coefficient, and the normalized scanpath saliency [51]. 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 postives 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 scanpath 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.

To calculate the computation time, each algorithm was set to produce a saliency map sized 128x171 to ensure that algorithms that downsample don't have an inherent advantage

for computation time. All times were computed on PC running Matlab R2012a on an i5-2310 clocked at 2.9GHz.

Fig 2-2 and Fig 2-3 show the ROC curves for different scales and the area under the curve for different values of alpha, respectively. Table 2-1 shows the full results from comparing the saliency maps with the fixation maps in the Toronto 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 2-4 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 2-2 shows the results for the CAT2000 database. Once again Gamma saliency performs the best in 4 of 5 metrics and computes the saliency maps in the fastest times. A qualitative example is shown in Figure 2-5.

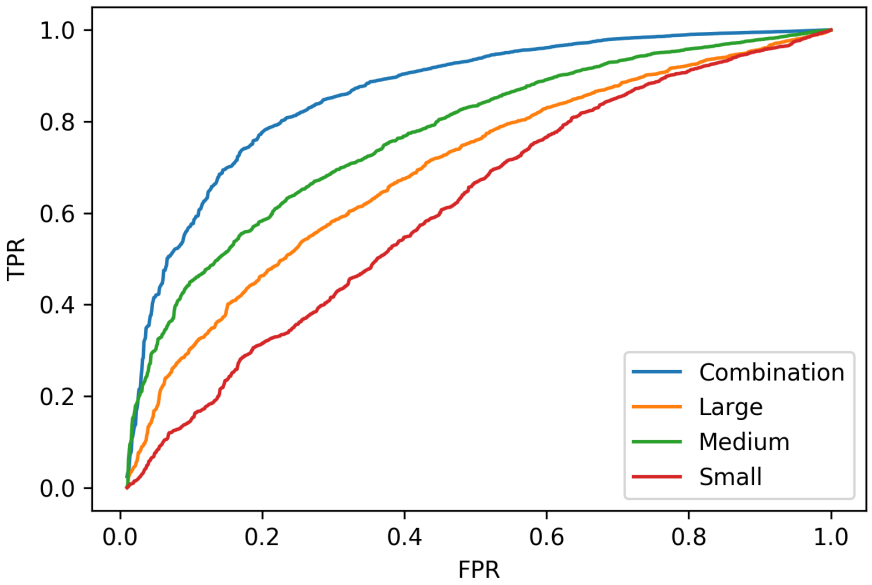


Figure 2-2. ROC curves for different scales of gamma saliency on the Toronto Saliency Dataset.

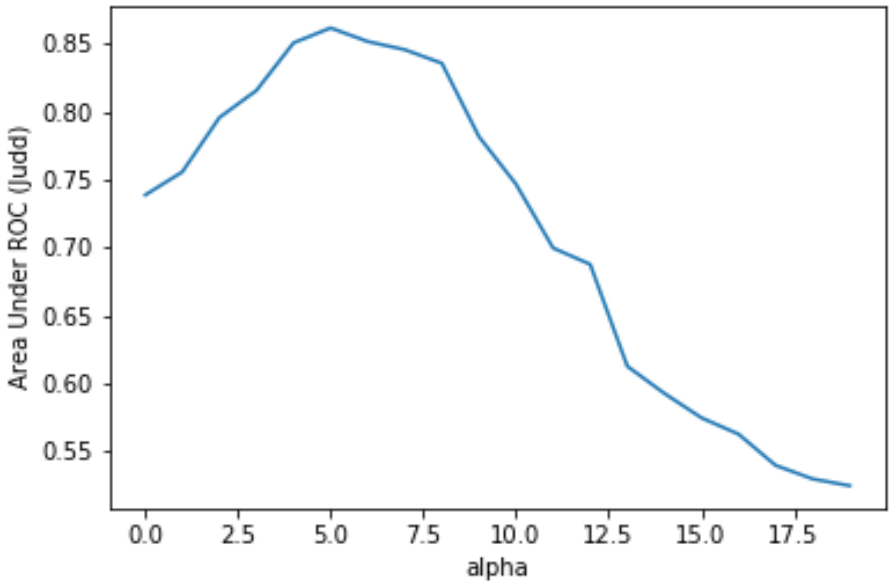


Figure 2-3. The area under the ROC curve for different values of alpha.

Table 2-1. Attention Prediction Results on the Toronto Database

Method \ Metric	ROC (Judd)	ROC (Borji)	Similarity	Correlation	NSS	Time (s)
Itti	.712	.597	.384	.275	.341	.280
AIM	.746	.632	.403	.363	.479	1.10
Torralba	.684	.600	.374	.292	.360	.78
GBVS	.848	.677	.488	.570	.638	1.03
FES	.847	.586	.520	.572	.446	.21
RARE2012	.785	.625	.477	.551	.489	1.39
RCS	.747	.609	.431	.414	.413	15.84
Gamma	.862	.695	.588	.581	.546	.21

2.3 Foveation

There are fundamental differences between how these 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, in order for saliency metrics to properly mimic the human vision system they must therefore be able to find regions of interest 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

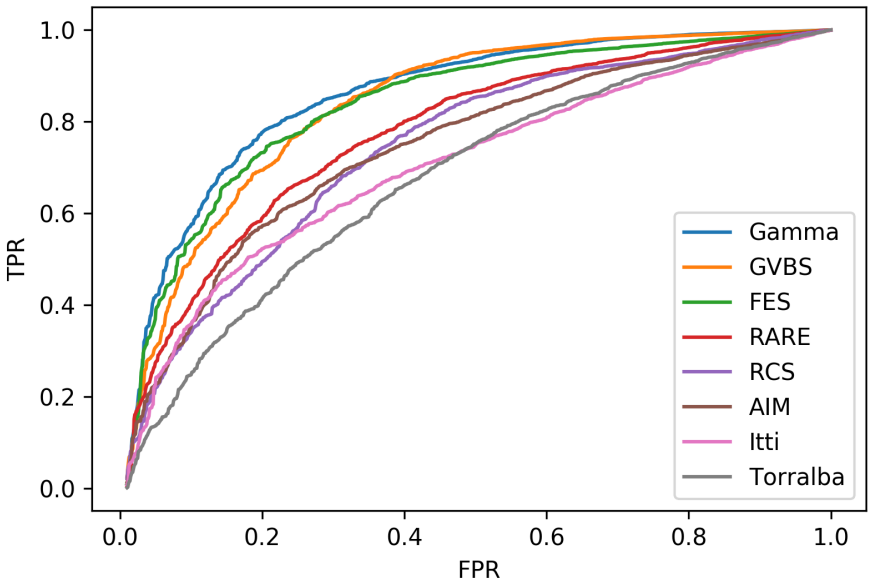


Figure 2-4. ROC curves on the Toronto Saliency Dataset.

Table 2-2. Attention Prediction Results on the CAT2000 Database

Method \ Metric	ROC (Judd)	ROC (Borji)	Similarity	Correlation	NSS	Time (s)
Itti	.700	.570	.377	.206	.258	.25
AIM	.772	.628	.437	.335	.497	1.04
Torralba	.770	.619	.437	.324	.448	1.20
GBVS	.844	.642	.498	.486	.510	1.05
FES	.812	.576	.562	.628	.368	.29
RARE2012	.822	.643	.466	.408	.511	1.37
RCS	.763	.593	.431	.292	.352	14.91
Gamma	.852	.676	.592	.633	.468	.21

initial glimpse can be an improvement over knowledge of an entire scene a priori [52]. 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 the biological and computational study, a framework is needed to transform images from single resolution to multi-resolution. Using images with a clear field of focus and a blurred periphery is called foveated imaging. Foveated imaging has been used in other areas in image and video processing to this date, mainly for

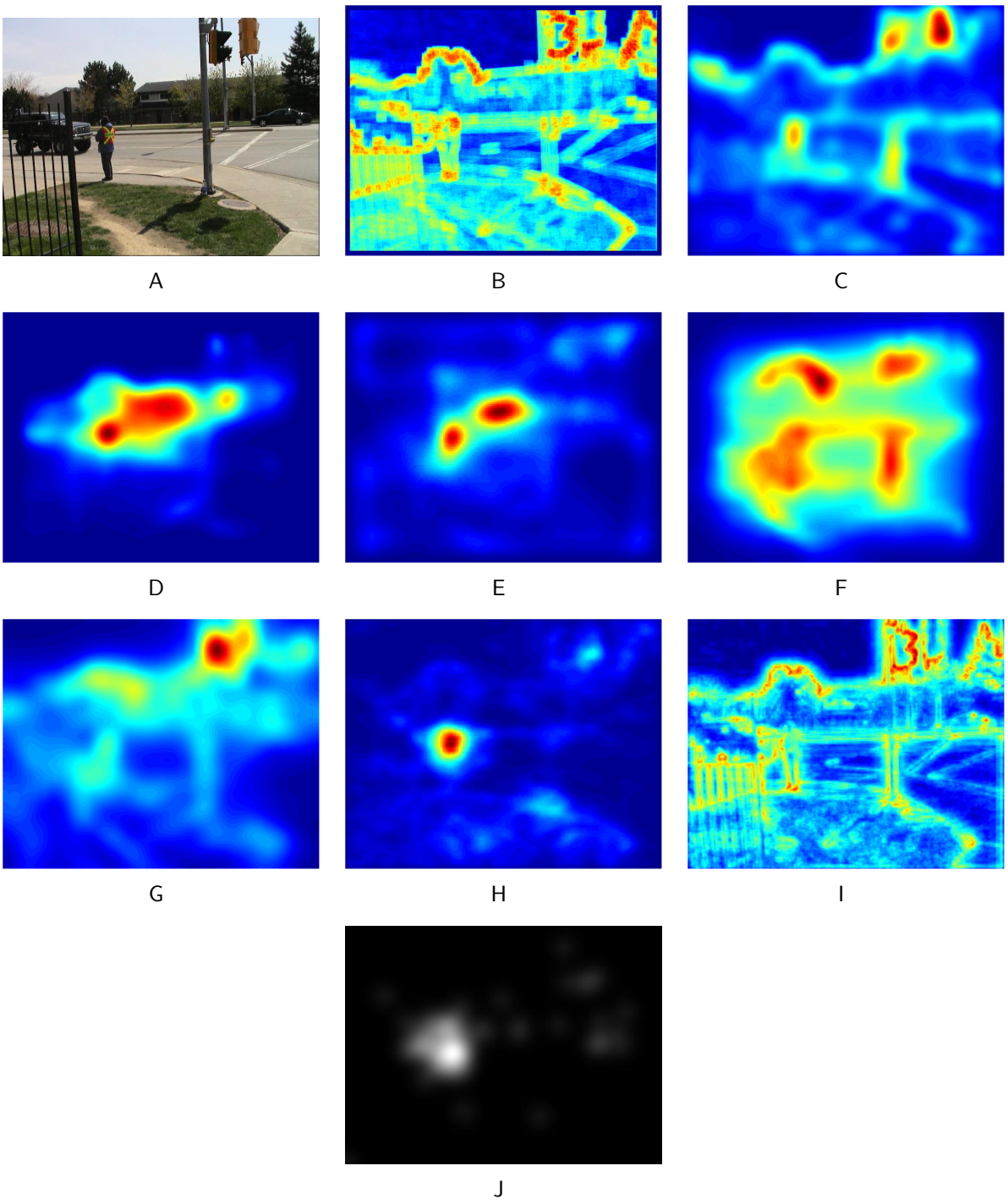


Figure 2-5. Comparison of different saliency measures for an image in the Toronto dataset. A) The original image. B) AIM Saliency. C) Region Covariance Saliency. D) Fast and Efficient Saliency. E) Gamma Saliency. F) Graph Based Visual Saliency. G) Itti-Koch Saliency. H) RARE2012 Saliency. I) Torralba Saliency. J) Fixation Map.

compression and faster processing [38], [39]. In addition, some saliency metrics have been tested in multi-resolution images in an attempt to speed computation and improve results [53], [54], but study in this area is still limited.

We propose that to build a vision system that adheres as closely as possible to the human standard, the saliency measure should be capable of predicting regions of interest outside the initial focal area. In the final system, we will use the Lytro Illum camera, which is capable of using light fields to alter the focus in an image according to the depth [55]. However, before building this final system, is it important to further study the idea of saliency in unfocused regions, since the images from the Illum will obscure any objects at a different depth than the current focus. To this end, we have selected several current saliency metrics and will study them in a standard fixation database, but the images will be foveated before calculating the saliency maps.

To mimic the effect of the fovea, we created images that are increasingly blurred around a small high-resolution area (artificial fovea). To create these images, we used the fast method developed by Geisler and Perry for images and videos in 2002 [56]. This method creates arbitrary visual fields in displays that allows for relatively high frame rates so that the visual field in the displays can be controlled in real time.

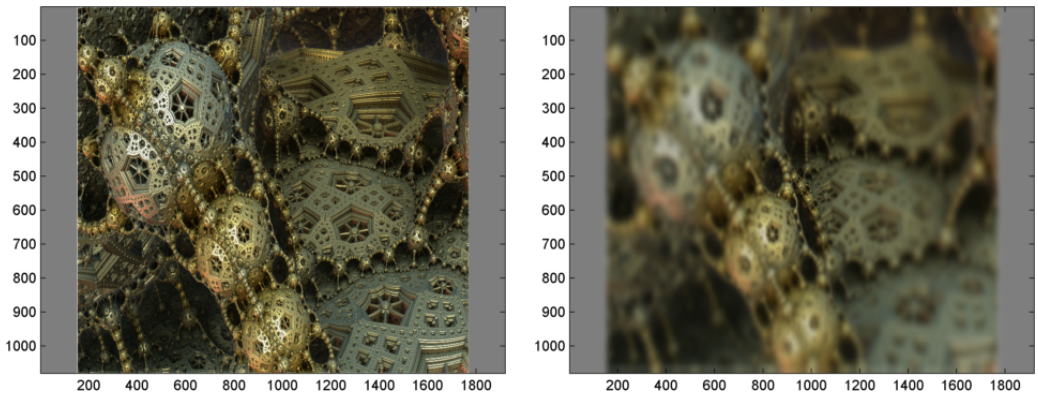


Figure 2-6. Comparison of a normal resolution image (A) and a foveated image (B).

This method involves creating 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 multiresolutional pyramid creating by first blurring the original image with a small kernel (such as 3x3), then downsampling 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. This process results in a small high resolution area surrounded by a blur that becomes stronger as it moves further away from the center region. 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. Figure 2-6 shows an example of an image before and after foveation.

In addition, models that are biased to the center of the image and blurred tend to match human fixations more closely [57]. In most eye tracking studies, users typically fixate on the center of the screen before the presentation of each visual stimulus. To best parallel present work, the present study began the analysis of each image with the highest resolution (the artificial fovea) at the center of the screen. To achieve the best results, most metrics are post-processed by training blurring and centering parameters on a subset of images with known fixations. However, foveation applies both blurring and center to the original image, which could approximate this effect with no post-processing and training necessary.

2.3.1 Eye-Tracking Results

Table 2-3 shows the results for each saliency measure on the foveated Toronto database. Gamma saliency still performs the best across most of the metrics, which shows that it could be used in a fixation system that approximates the HVS by using foveated inputs. Interestingly, the foveation actually improves the results obtained by most saliency measures, possibly by natually adding a blur and center bias that has been shown to improve results in previous studies.

Table 2-3. 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
RARE2012	.841	.656	.525	.632	.591
RCS	.819	.629	.517	.595	.517
Gamma	.858	.684	.607	.649	.483

2.4 Focus of Attention

With the study of saliency in foveated images complete, we can create a bottom-up attention framework for the overall vision system. To do this, we use an image that can be refocused at different depth level from the Lytro Illum. The image start as being focused on the center, computes the saliency map, then refocuses on that area. A frame is sampled from the image, then the process is repeated with the addition that the previously scanned areas are return inhibited.

Figure 2-7 shows the flowchart for the new focus of attention (FOA) mechanism. This system is meant to approximation the saccade and fixation system found in the HVS. The system starts by focusing around the central point in a given image, computes the most salient point in the image, and refocuses around this region. In this local region of high resolution, the saliency is computed again in order to center the fixation around the salient object, effectively performing a corrective saccade such as is found in the HVS. After this corrective saccade, the scale of the object is found by taking the connected salient region and expanding it by 10% to ensure coverage. This local image patch is broken into a number of 28x28 frames for the DPCN to process. In these initial tests, prior information about the desired objects was used to set the size of the fixations at 25 frames in a 5x5 configuration. In future tests, the size information will be provided by different scales in the saliency measures.

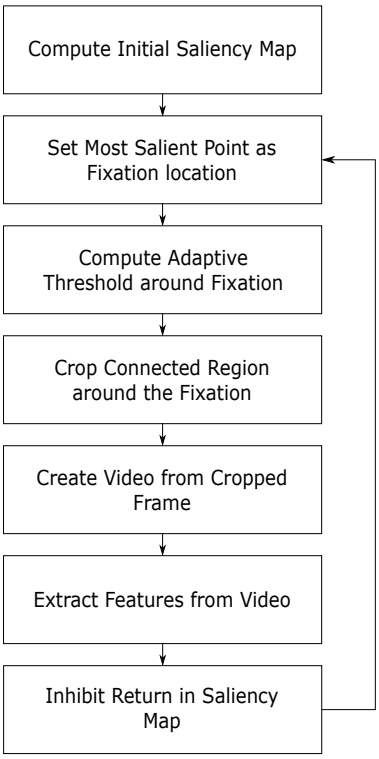


Figure 2-7. Flowchart showing the focus of attention algorithm.

Figure 2-8 shows the FOA system in action for a simple Lytro image. The numbered blocks are placed at different depths in the image to facilitate the light-field refocusing so that only one block is clearly in focus at a time. The images in the figure show the first and beginning of the second iteration, from the initial center fixation all the way through to finding the second salient fixation after inhibiting the original location.

Figure 2-9 shows the results from this system in an environment with a more complex background. The image from the Lytro Illum is focused at different depth levels (a-d), the eye tracking data overlaid on the image (e), the original saliency map computed through GBVS on the center focused image (f), and the first four patches sampled from the image (g-j). The eye tracking data comes from a single free-viewing session consisting of 10 seconds. The four frames all match fixations from the eye tracking data. These frames would be broken into pieces to scan the entire frame, then this video would be processed by the DPCN. In the same way that the human viewer is fixating on the different interesting objects to identify them, the

focus of attention system breaks this complex problem into multiple easier problems before forming a representation of the scene.

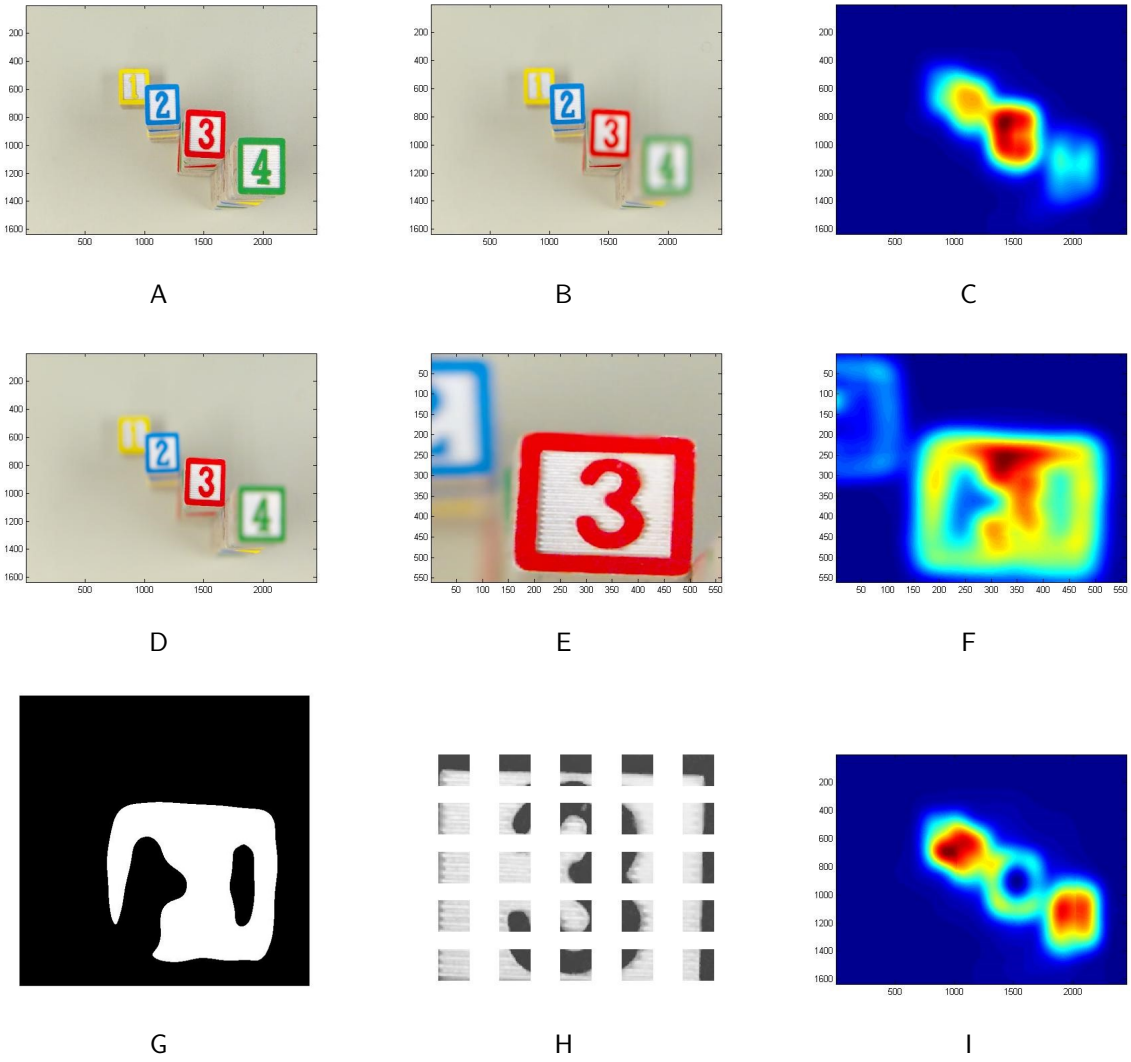


Figure 2-8. A series of images showing the progression of the focus of attention algorithm. A) The original image. B) The image focused on the center point. C) Saliency map created from center-focused image. D) The image refocused on the most salient point. E) The local patch containing the point. F) 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.

2.4.1 Computation Improvement Results

After showing that the saliency metrics could work in vision environments consistent with the HVS, the next test served as a proof of concept for the focus of attention mechanism.

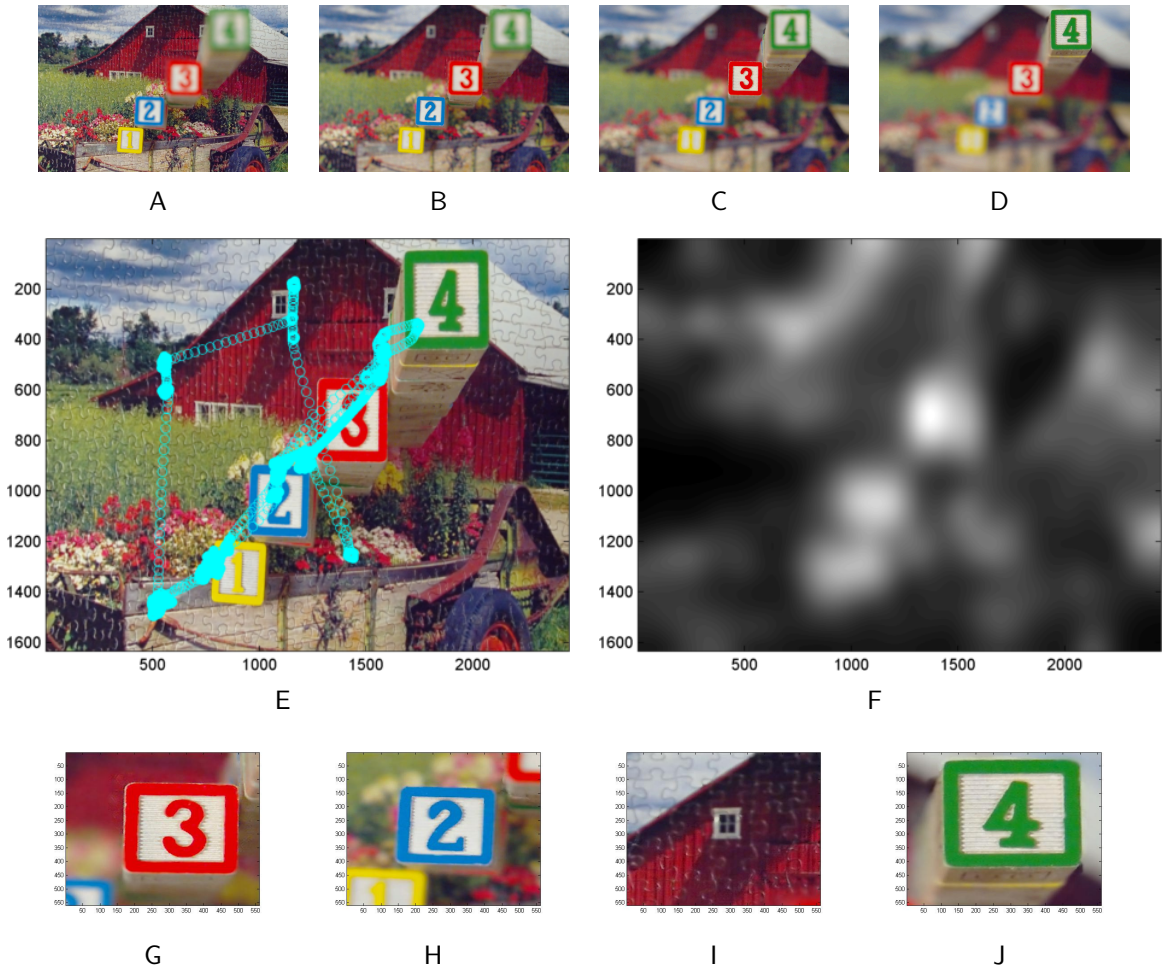


Figure 2-9. A preliminary result on a simple visual attention framework that could create videos to be processed by a DPCN. A) The image focused on furthest depth level. B) The image focused on second depth level. C) The image focused on third depth level. D) The image focused on highest depth level. E) The image with overlaid eye-tracking data. F) Initial saliency map from center focused image. G) The first extracted frame. H) The second extracted frame. I) The third extracted frame. J) The fourth extracted frame.

To test this, we created a dataset consisting of a single object against a blank background as shown in Figure 2-10

Each of the 108 644x644 pixel images contained one of four different numbers (1-4) placed at a random location. To test the new architecture integrating the DPCN with the focus of attention framework, each image was scanned into 28x28 non-overlapping patches for processing. The entire image (529 frames) was tested against a smaller 25 frame video

Table 2-4. Computation time (in seconds) required to process all objects in a scene.

	FOA	DPCN	Total
Full Images	0	5741	5741
Sampled Patches	43	269	312

generated automatically for processing by the FOA mechanism. Each set of videos created by the frames was trained for 100 batches on the DPCN. The time information for the study is shown in Table 2-4. In all, the architecture with the FOA trained in 5.4% of the time it took to train the on the larger videos. However, this ratio is dependent on the number of objects within a scene; adding more objects will increase the total amount of time by requiring the system to extract and train the DPCN on more objects. Since the time saved by processing small regions instead of the entire scene is so large, this system should scale to separately processing multiple regions in a complex scene while still saving time.

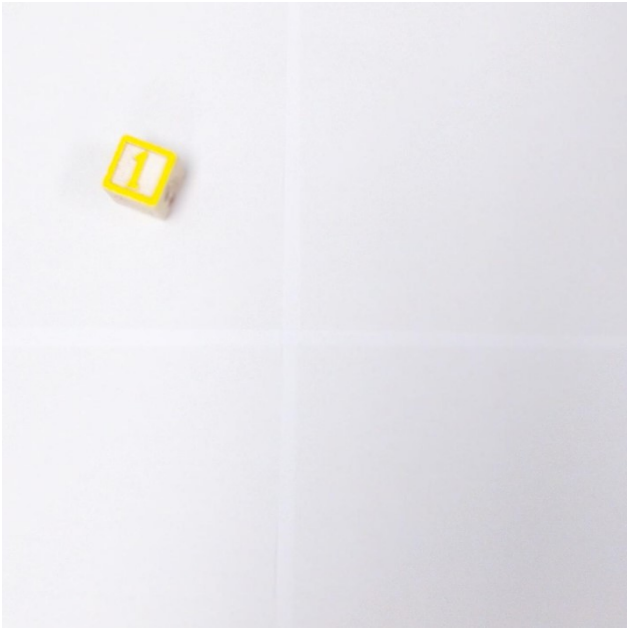


Figure 2-10. An example image for the computation improvement test.

Figure 2-11 shows the causes of the DPCN trained on the patched sampled using the focus of attention projected down to two dimensions using PCA. The causes for this DPCN are clearly separable, and training a simple classifier quickly resulted in a classification rate of 98%. In comparison, the DPCN was unable to differentiate between the classes for the full

images (which included a large amount of background useless for the classification task). For the same projection, the causes were not clearly separable and the classification rate was 42%. This shows that in some cases, focusing on only relevant areas can improve both computation time and classification results.

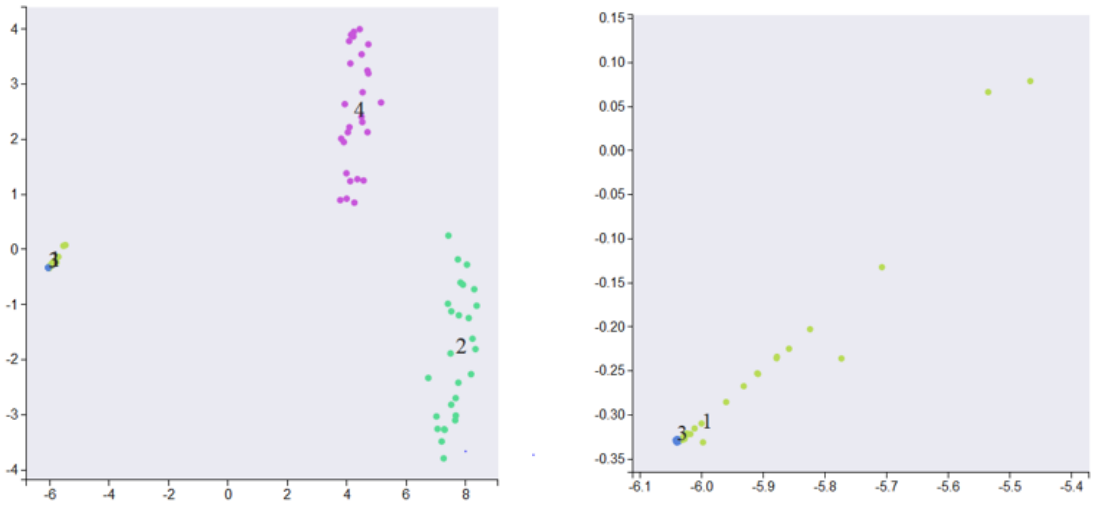


Figure 2-11. A 2D projection of the principle components of the causes of the DPCN after being trained on the image patches from the focus of attention. A) The four clusters with each color representing a different input class. B) A zoomed view showing separation between two closely placed clusters. NOTE: CLEAN THESE AND ADD FIGURES FOR DPCN TRAINED ON FULL IMAGES

CHAPTER 3
UNSUPERVISED FEATURE EXTRACTION

3.1 Background

Neural networks and deep learning architectures are the current state-of-the-art for image classification and recognition. They have been shown to reliably distinguish between as many as 1000 different classes of objects [58]. 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. In order to move towards more fully autonomous systems, we need an architecture that can extract features from a wide range of objects in cluttered scenes with minimal labels in training.

Humans have the remarkable ability to view a scene and form an overall representation in a short length of time. However, due to the complexity of visual search, 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) is broken into two broad systems or pathways: one primarily devoted to spatial neuro-computations and another for object perception. The ventral stream, or the "what" pathway, includes V1, V2, V4, and continues through to the inferior temporal cortex. It is responsible for forming higher-level representations of the visual scene and identifying objects. The dorsal stream, or the "where" pathway, goes through V1, V2, the dorsomedial area and then the posterior parietal cortex. It is associated with processing the location and spatial relations objects, guiding the focus of spatial attention, and controlling exploratory eye movements [4].

We propose to build a two-stream perception system that shares this broad organization with the human vision system (Figure 3-1). It will have one stream that processes an entire scene to find potentially interesting data, and a separate stream dedicated to processing the fixations given by the first stream. By using this divide-and-conquer approach, we can quickly process large, unwieldy images and break them into smaller pieces that require the more intense computation required to extract features from an image.

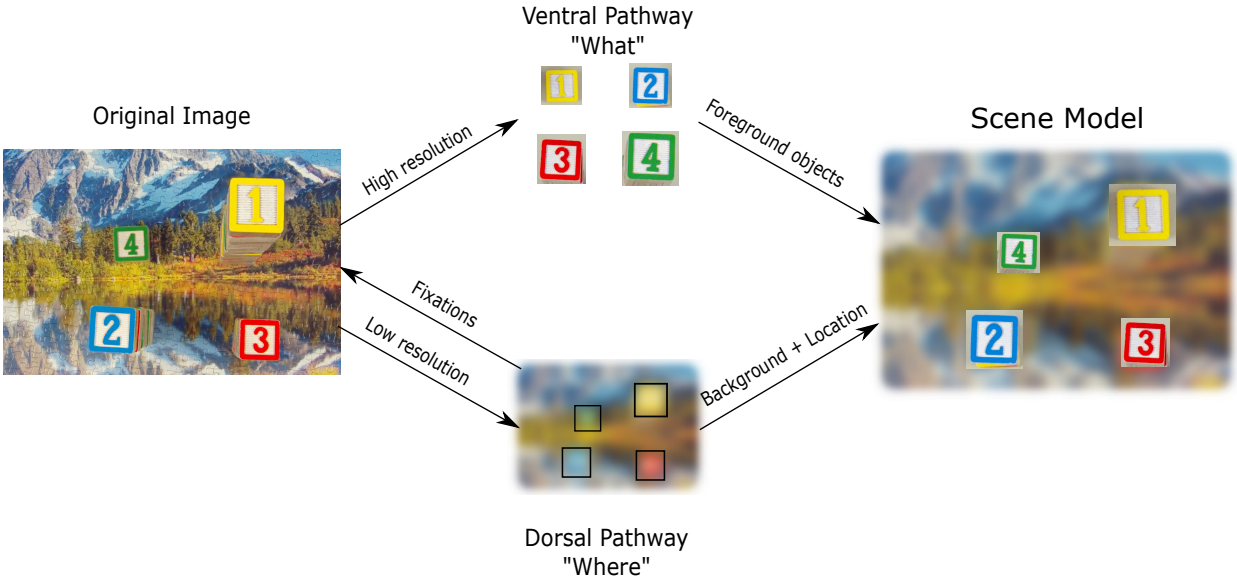


Figure 3-1. Diagram of the proposed architecture with its two pathways. The dorsal pathway extracts spatial information and uses this to find fixations for the ventral pathway, which extracts features from the relevant objects. In the HVS, there would be another feedback connection where the ventral pathway inform fixation driver used by the ventral pathway.

The first processing pathway will replicate the dorsal stream, which deals with spatial attention over an entire scene [4]. In the HVS, the eyes provide full access to high resolution data only in a small region called the fovea where the focus of attention is centered, approximately 3 degrees of visual angle around the point at which gaze is directed during a given moment in time [5]. Thus, the human brain must remember and infer parafoveal and peripheral information, or use a combination of the two, to compute targets of interest for future fixation locations. As shown from empirical research on saccadic exploratory eye movements [5], these future fixations will target the regions in the visual periphery.

In contrast, current techniques in computer vision tend to process entire images by convolving them with learned filters [59]. By preprocessing visual data with an attention mechanism, we can focus processing only on the subregions that contain interesting data and use these to form an overall representation of the scene. In the same way that convolving learned filters over an image is a step beyond scanning pixel-by-pixel, processing still images

as videos of small frames composed of visually interesting regions could be a further step that simply discards large regions of the image that have little to no effect on classification.

However, this introduces a new problem: finding the regions that contain the relevant information. Recently, methods have been proposed that suggest regions both within the structure of the network [60] or as a separate mechanism based on image features [14]. These approaches are supervised and are trained to choose regions that contain data most relevant to the label.

Since the introduction of Itti's method in 1998 [22], saliency has become a popular way to predict visual attention in images and could therefore be used to segment out the interesting 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, high in contract, or categorically different from its surroundings. By using saliency as a proxy for bottom-up visual attention, it could be possible to create an unsupervised system that quickly selects regions of interest for more computationally intensive processing, then combine these representations into an overall understanding of a complex scene in much the same way the HVS works.

With saliency functioning as a bottom-up attention mechanism, the second pathway of Figure 3-1 will approximate HVS ventral stream functions and will form representations of and extract features from objects. The ventral stream in the HVS receives visual data from the fovea and builds an active representation through the visual cortex. This is the role that neural networks and deep learning have traditionally played in image processing. By focusing the representation on only specific objects rather than the entire scene, we save computation on background and other non-informative data as well as ambiguous data that could be other, separate objects. By segmenting objects around highly salient points found by the attention mechanism, we can restrict the role of the network to finding invariant representations of the objects that it encounters.

These neural networks are generally trained on large datasets such as Imagenet [58] or MNIST [61] 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 to extract 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 outside of certain situations.

Recently, there has been study on forms of supervision other than class labels, which often consist of large datasets curated by hand. Temporal supervision [62] [63], egomotion [64], and other self-supervised or un-supervised learning techniques can be used to extract features from data before fine-tuning a network for a specific task based on a much smaller labeled set. By reducing the number of labels needed to still produce acceptable results, they have moved the networks one step closer to wider implementation in a range of problems without nicely labeled sets.

Alternatively, Chalasani and Principe introduced a framework that uses inference in time between subsequent frames of a video to learn relevant features in an unsupervised manner [12]. Using this and other similar frameworks, it is possible to learn to extract features from unlabeled datasets by taking advantage of the structure introduced through video.

In this paper we propose the foundation for a new unsupervised dual pathway architecture for vision systems that separates the spatial perception element from the object recognition Figure 3-1. The attention system will be based on a saliency measure, while the feature extraction will come from a deep learning method that uses temporal supervision. Section II discusses related work, Section III outlines the methods used for the full vision system, Section IV presents the results, and Section V concludes the paper.

3.2 Related Work

3.2.1 Attention Systems

Despite these recent advances in image processing, classification results on image datasets with multiple objects in complex scenes has advanced little when using convolutional methods. Recently, research has begun into breaking images down into regions, then performing classification on these rather than the entire image [13]. 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% [14].

Most of the region classification methods proposed at this time were designed to be trained in conjunction with deep convolution networks, such as OverFeat [15]. 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 proposal method (selective search), before separately sending these regions to a CNN for classification and then finally recombining similar regions [14]. This original R-CNN work is closest to our paper. However, our approach is both unsupervised and uses the initial attention mechanism to create a video to provide extra information to the feature extraction.

Despite the different paradigms, processing smaller regions of images has the potential to be the next breakthrough in computer vision by reducing the brute force sliding windows in the CNNs. The Spatial Transformer Network, on the other hand, 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 [60].

There are other methods that use multiple pathways in deep learning. One such example is the work by by Wang et. al [65], where the input images are broken into separate channels for assessing the aesthetics. However, these pathways are different from those presented in this work since they are not separating tasks explicitly as much as providing an initial set of features to the network. Other work is by Zhao et. al [66], where information from the pooling

layers is fed to the corresponding layer in the decoder to preserve the spatial information lost through encoding the image. This method does not reduce the input dimensionality as does ours, nor would it create separate feature vectors for separate objects - it still describes each image at once using a single hidden state. The work of Bazzani et. al [67] uses separate pathways for object tracking and object recognition.

Saliency is often used as a predictor of bottom up attention. 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 [22], [23], [24], [25] [68], the global context methods that compare regions to other regions from any location in the image [26], [27], and the normal image methods that compare an image to a standard ideal [28], [29], [30], [31], [32], [33].

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 [54] [39]. Walther et. al [69] also used bottom-up saliency as an attention mechanism to extract features from unlabeled images, though our attention approach is extended by creating videos from the still image.

Saliency has also recently come to prominence in terms of explaining the inner workings of neural networks. The works of Simonyan et al. [70] and Montavon et al. [71] both use saliency maps as explanations of localizing pieces of images that excite certain regions of the networks. In contrast to our saliency maps, these are computed with the trained networks and used to for further analysis of the inner workings of the network. Our saliency maps are computed by a static method computed before the network is involved - the network is only presented salient objects, as opposed to using the trained network to later decide which objects are salient.

3.2.2 Feature Extraction

Training deep learning architectures without explicit class labels has been a growing area of research [63] [72]. In an effort to expand these techniques beyond datasets that come with

an excess of labeled examples, there have been effort toward learning features based on other forms of supervision such as temporal and egomotion, as well as the traditional self-supervision used by autoencoders [73].

Goroshin et. al [62] and Wang and Gupta [63] learned short term dependencies between subsequent frames in video. Agrawal et al. modeled the egomotion of the camera in order to provide a form of supervision other than labels [64].

The deep predictive coding network (DPCN) by Chalasani and Principe [12] [11] used temporal predictions to learn features through time and build representations of video streams. This work was later extended into the recurrent winner-take-all convolutional networks (RWTA) [74], which use a dual-stream autoencoder structure to represent the current frame and predict the next frame as will be explained shortly.

3.3 Attention Based Model for Scene Understanding

Humans experience even static scenes through movement, whether by moving fixations across a painting or walking around a still landscape. This motion is inherent in understanding our environment; 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 in a coordinated manner through top-down attention mechanisms at a slow pace as the viewpoint is updated. Since the brain is in control of the saccades, the temporal coherence builds the full understanding of the scene as objects are recognized and placed into memory as the brain searches out new fixations.

To mimic the dual pathways of the HVS, we propose to use two separate systems: one for attention and detection of visually salient regions, and one for representing these regions and extracting useful features [75]. Additionally, to avoid the use of class labeling and to leverage the capabilities gained by combining these two systems, we will use an unsupervised saliency based detection system with a temporally supervised learning structure. Combining these two pieces makes the system self-organizing; the system creates the supervision that it then uses to extract robust features.

In order to successfully take advantage of the temporal supervision, the attention mechanism must provide not only a salient point on which to focus, but a structured series of frames encompassing the object. By collecting frames with each giving a slightly altered view of the subject, a temporally supervised learning structure is able to learn representations that persist across the frames. This leads to a more invariant set of features learned by the vision system that can later be used for classification or other tasks, but it requires the quantification of the time structure, i.e. it can not be a static model.

The procedure for creating these videos is outlined in Figure 3-2. Given an input image, the fixation is defined via the most salient point. The object around this point is segmented using information from the attention mechanism. Since Gamma Saliency is a multi-scale measure [68], the underlying feature maps contain information on which scale the object was different from its surrounding. Working from this base, we crop a patch around the object, create the video according to the data, and send the sequence to the RWTA. There are multiple techniques that could be used for creating the videos in a structured manner. Two of these (rotating the patch around the objects and translating the frame) have been shown to be successful in previous applications of the RWTA [74]. Here we will use rotation.

Once the video of the object has been created, the fixation is moved to the next most salient point, the next object viewed and sampled, and the process continued. At each previous fixation point, the saliency map is inhibited by applying an inverted gaussian that corresponds to the foveal area and the inhibition of fixation return found in the HVS.

This system differs from the others in that it is unsupervised unlike the STN and OverFeat and it just once identifies distinct locations for processing a rather than extracting features from multiple overlapping regions such as the R-CNN. Our approach has the benefits of speeding computation by reducing the dimensionality of the input and not requiring labels, but it also has the drawback of possibly picking salient information that is not relevant to the labels. To account for the irrelevant information, we include a "junk" class in our final classifiers that will learn to classify the false positives provided by the attention system.

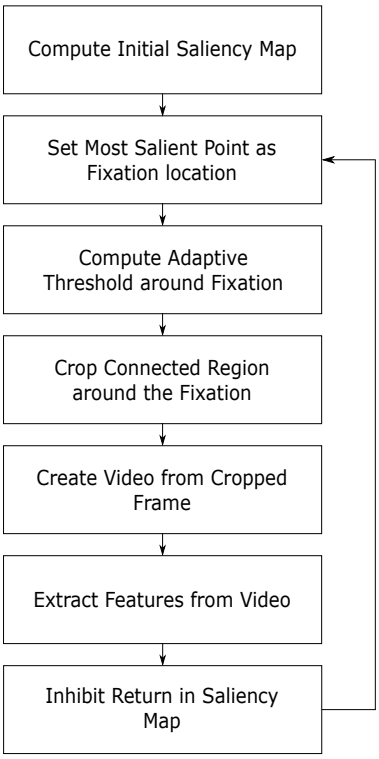


Figure 3-2. Flowchart showing the focus of attention algorithm.

The junk class is created by including patches from non-relevant areas in the training of the RWTA. By doing this, we train the RWTA to represent not only the digits, but also patches that do not contain relevant information. By doing this, we can ensure that the features we learn are able to not only distinguish the different types of relevant information, but whether the information is relevant at all.

The next subsections detail the main components of the attention, video creation, and feature extraction systems.

3.3.1 Gamma Saliency

An effective attention mechanism in a dual pathway vision system should meet a few basic requirements. 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 system should function as an accessory to the recognition system which means that the attention will not 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 with only low level features and work as a detector.

To accomplish this, we will 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 ones.

The 2D gamma kernels selected here have been used for target detection [46]. Their circular shape is ideal for comparing a center region to a local neighborhood, and the size of each can easily be controlled by two parameters, which allows for multiscale regions of support. In addition, the gamma kernel has the ability to be computed recursively to save computation and smooth the neighborhood. This it well suited to frontend signal processing applications such as saliency.

Similar to Itti and other methods, Gamma saliency is based on the center surround principle: a region is salient if it is different from the surrounding neighborhood. In order to compute these local differences, we use a circularly symmetric gamma kernel that emphasizes a center while contrasting it with a local neighborhood through convolution:

$$g_{k,\mu}(n_1, n_2) = \frac{\mu^{k+1}}{2\pi k!}\sqrt{n_1^2 + n_2^2}^{k-1} e^{-\mu\sqrt{n_1^2+n_2^2}} \tag{3–1}$$

For this kernel, n_1 and n_2 are the coordinates of the local support grid, μ is the shape parameter, and k is the kernel order. We can control the shape and size of the kernel using μ and k. When $k = 1$ the kernel peak is centered around zero, but for larger kernel orders, the peak is centered at a distance k/μ away from the origin. In addition, increasing μ will decrease the width of the peak. By using these parameters we can construct a 2D shape that creates a local difference comparison by subtracting a neighborhood kernel from a center kernel. To create a multiscale measure, we simply combine multiple kernels of different sizes before the convolution stage 3–1 [22], [42]. The kernel summation is described in 3–2, where all k for

even m are 1 to create the center kernels. Many multi-scale saliency measures use three scales designed to find small, medium, and large targets in the images.

$$g_{total} = \sum_{m=0}^{M-1} = (-1^m) g_m(k_m, \mu_m) \tag{3–2}$$

With this local difference measure, the rest of the saliency measure is constructed similarly to the other center surround methods [57]: the image is broken into feature matrices, each matrix is convolved with the multiscale kernel to compute local differences, the matrices are combined and exponentiated to accentuate peaks, then postprocessing is performed to boost results using a Gaussian blur and a center bias. The feature matrices are composed of the CIELab color space, which has three dimensions - one luminence dimension and two color opponency dimensions. In this space, the distance between two colors is the Euclidean distance. Each feature is convolved with the multiscale kernel to create a local difference measure, which are then combined into a full saliency map.

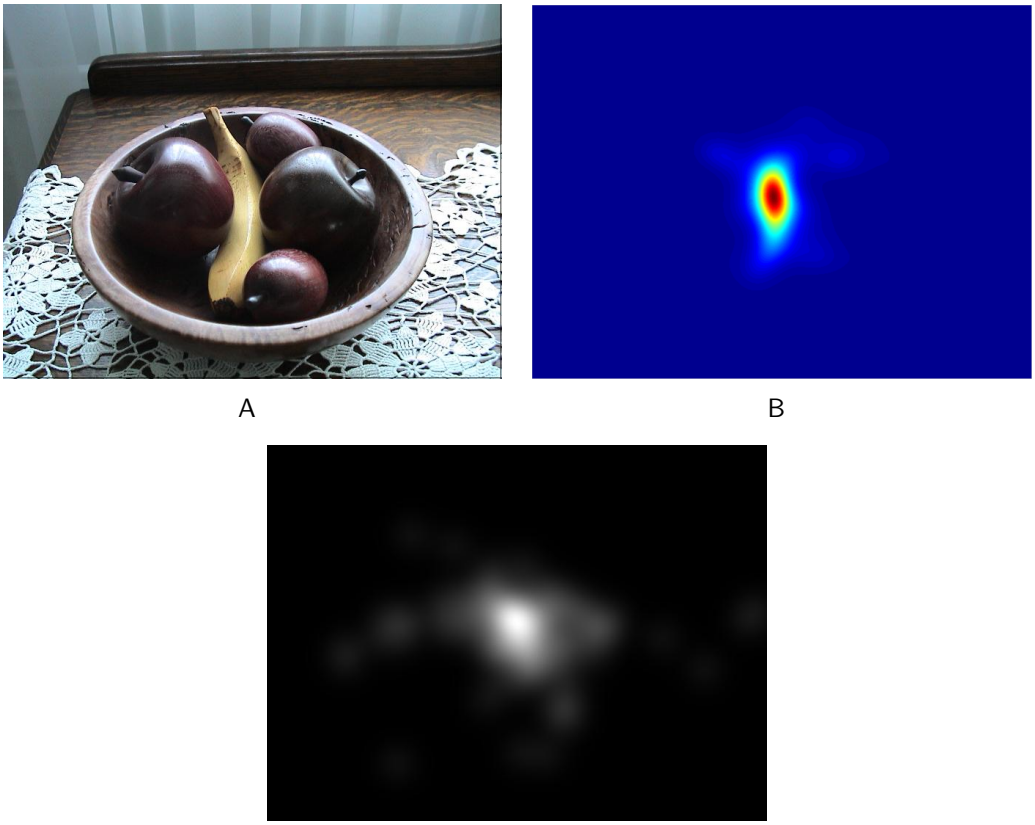
$$S = \frac{|g \bullet L|^\alpha + |g \bullet a|^\alpha + |g \bullet b|^\alpha}{3} \tag{3–3}$$

Lastly, we postprocess the final combined map with techniques shown to boost the saliency scores [57]. These including accentuating the peaks, weighting the center of the map, and blurring the final results 3–4.

$$S = (S * G(\sigma^2)) \bullet G(.5) \tag{3–4}$$

3.3.2 Video Creation

After creating the saliency map, the next step is to segment the object and create a video using an adaptive threshold [76] on the saliency map to segment out regions of interest. Since the creation of the saliency maps emphasizes edges and then blurs the results, the connected high saliency regions often contain an entire object. The borders around the connected region that contains the highest saliency point are expanded by 10% and local saliency within this



A B C

Figure 3-3. Example image from the Toronto Saliency Dataset (A), saliency map produced by Gamma Saliency (B) and the ground truth fixation map (C).

point is computed to center around the main object or separate multiple objects within the saccade. This is similar to the corrective saccades in HVS that perform small amplitude adjustments to the larger saccadic movements [77]. Each local object is then cropped from the image and resized for the feature extraction network.

From here, we need to create a video from a single static frame. In this paper, we rotate the cropped patch from a range of -45° to 45° in six steps and treat each rotation as a frame. There are other possible methods of creating videos, such as scanning frames across an object, but for this paper we followed the procedure outlined in [74].

These videos are then used as an input to a recurrent winner-take-all network that uses the temporal structure afforded by the video to learn robust features, essentially providing the link between the attention mechanism and the feature extraction to form a combined system.

The extra temporal element adds an extra layer of structure to the data and can function as a form of self-supervision. This makes the system self-organizing - it takes an unlabeled image and creates a relevant label to be used (in the form of a subsequent frame) rather than using a provided label or simply attempting to minimize reconstruction error on the image itself.

Once the video is created from the initial patch, the process is repeated by moving to the second most salient region and creating a second video there, and this process repeats until the regions above the threshold have all been visited. To ensure that the algorithm does not return to an area previously visited, each local saliency map is multiplied by an inverted Gaussian around the previous fixation. This mimics the inhibition return found in the HVS.

3.3.3 RWTA

Rather than using explicit labels in the form of class supervision, the recurrent winner-take-all (RWTA) model [74] will use architectural constraints along with the structure inherent in a video stream in order to extract robust features from images. The end goal of this feature extraction is to create a sparse set of latent codes that can be used to differentiate between objects. Using a Winner-Take-All method enforces this sparseness in the latent codes by using aggressive dropout [78], detailed in 3–5, where f, r, and c are the rows, columns, and channels of an image. In addition, by creating a video and then using an RNN to learn the temporal structure of the video, we can learn robust features that describe the object [64].

$$\begin{aligned} WTA(x_{f,r,c}) = x_{f,r,c} \ \textit{if} \ x_{f,r,c} = \max_{r,c}(x_{f,r,c}) \\ WTA(x_{f,r,c}) = 0, \ \ \textit{else} \end{aligned} \tag{3–5}$$

RWTA combines a stateless convolutional autoencoder and a convolutional RNN that creates a dynamic state that describes the change between two frames, as shown in Figure 3-4. With this architecture, we are backpropagating errors from two sources: the reconstruction error from the input image as well as the reconstruction error from the subsequent frame. The RWTA accomplishes this by adding a recurrent layer to the hidden state, which attempts to

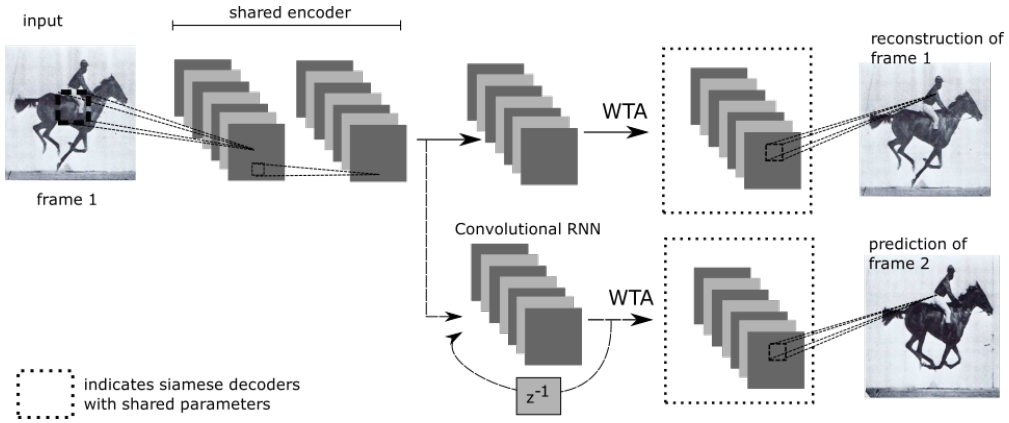


Figure 3-4. Dual stream structure of the RWTA

predict the hidden state of the next frame. 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. Minimizing the two errors together allows us to learn more robust features through the concept of slowness. In a coherent video, the frames change slowly, so the corresponding features that represent the frame should also change slowly (or at least be related). We can learn this relation through the RNN.

The cost function for the RWTA is given by

$$L_t = \mathbb{E}[(x_{t1} D(E(x_{t1})))^2 + (x_t D(R(x_{t1})))^2], \tag{3–6}$$

where x_t is the video stream, the stateless encoder is E, the shared decoder is D, the RNN by R, and $\mathbb{E}$ denotes the expectation operator. As each frame is input to the RWTA, the top (spatial) stream attempts to recreate the current frame while the bottom (temporal) stream attempts to predict the subsequent frame of the video. The shared decoder forces the final reconstructions to be made from the same space so the difference between the frames is encoded in the hidden states rather than the decoders themselves. The parameters of the RNN and spatial autoencoder are trained using backpropagation-through-time [79]. More details and tests are described in [74].

Table 3-1. Unsupervised Cluttered MNIST Results

Method \ Metric	RWTA w/ Attention	TDN w/ Attention	Autoencoder	VAE
Classification	88.59	86.25	25.82	24.65
Time (s)	1600	68	107	146

Table 3-2. Supervised Cluttered MNIST Results

Method \ Metric	CNN	CNN w/ Attention	STN Full
Classification	88.00	92.76	95.69
Time (s)	38	23	498

3.4 Results

This section presents the experimental results. Full information on the parameters used for the attention and the classifiers is presented in the appendix.

3.4.1 Cluttered MNIST

For an initial test we use the cluttered MNIST dataset. This consists of handwritten digits in a 28x28 bounding box dropped into a 60x60 canvas with six pieces of correlated clutter, which are randomly sampled 6x6 frames from other digits. This functions as a very basic test of the dual-pathway vision system; the attention system should ignore the empty background and clutter to focus in only on the relevant data. This should in turn speed computation by eliminating computation over large portions of the images and help improve final classification results by helping the network focus on learning only relevant features. Since this dataset only contains labels for the digits and not their locations within the background, this is an end-to-end test of the dual-pathway system: both the attention and feature extraction will be validated by the classification results.

This second portion could prove to be a benefit for self-supervised learning, since the training will not include explicit labels that the network can use to inform which features are best for classification. By using the attention mechanism to filter some of these out, we hope to create a system that learns a more robust and useful set of features than one that would try to explain the entire scene, which often contains a large amount of useless data.

Figure 3-5 shows an example image from the Cluttered MNIST dataset, the corresponding saliency map, and finally the cropped image patch containing the salient point. By preprocessing

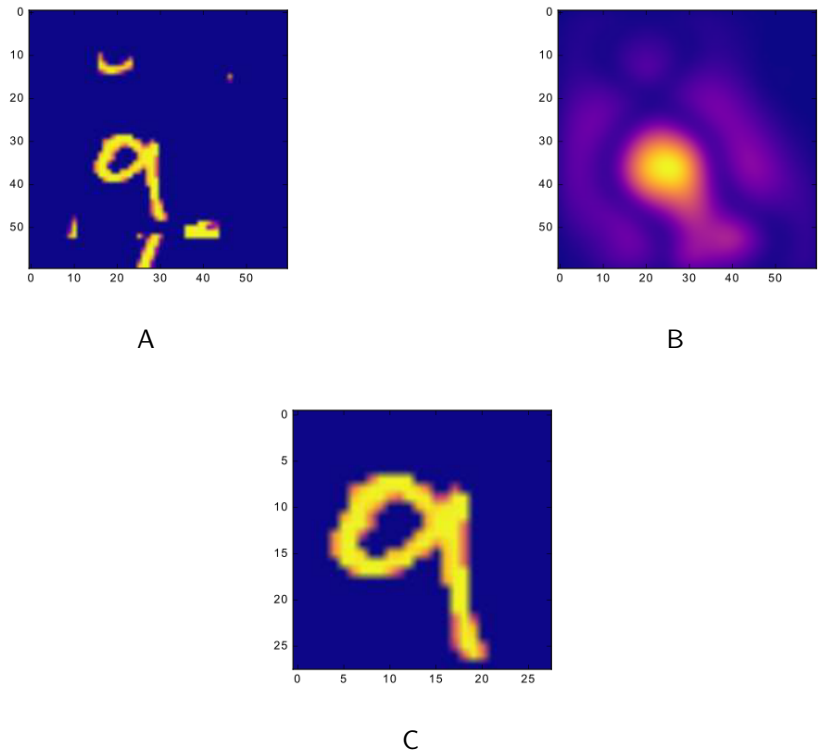


Figure 3-5. Example of FOA on a digit. Initial image with MNIST digit and clutter (A), saliency map produced by Gamma Saliency (B) and patch cropped around the salient region (C).

the image in this way, we reduce the dimensionality of the input to the feature extractor as well as eliminate confusing clutter.

Table 3-1 shows the classification accuracies and the time per training epoch for the unsupervised methods, including a standard autoencoder and the variational autoencoder [80]. A linear SVM was trained on the encoded versions of the digits to produce the accuracy percentages. The RWTA with Attention produces the highest accuracies by only focusing on the relevant information, rather than trying to explain both the digits and noise in the latent states. However, training the RWTA is time-consuming due to the RNN used for the temporal encoder. To mitigate this, instead of an RNN, we simply learned weights to combine the features produced by each frame in a single set of features in a Time Delay Network (TDN). This is possible because we are also controlling the creation of the videos, so we know the

exact number of delays present, which informs the network architecture. In these artificial videos produced from still images, the accuracy is similar with the benefit of greatly reducing the computation.

Table 3-2 shows how the attention mechanism can be added to other feature extraction and classification approaches to improve results. By combining this attention mechanism with a CNN, we are able to improve the results over a standard CNN as well as reduce the time, as opposed to other localization methods which can increase training time by introducing more parameters.

3.4.2 Cluttered MNIST network

All networks were created using Keras [81] using the Theano backend. We adopt the notation that conv[N,w,s] denotes a convolutional layer with N filters of size $w \times w$, and stride s; fc[N] is a fully connected layer with N units; and max[s] is a $s \times s$ 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. 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 $\mu = .2$. The neighborhood kernel has an order of $k = 9$ and $\mu = .5$. 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.

Table 3-3. Unsupervised SVHN Results on the Bounded Dataset

Method \ Metric	RWTA w/ Attention	TDN w/ Attention	Autoencoder	VAE
Classification	92.51	92.28	15.58	17.46
Segmentation	83.67	83.67	NA	NA
Time (s)	12638	2015	2372	2784

Table 3-4. Supervised SVHN Results on the Bounded Dataset

Method \ Metric	CNN Full	CNN FOA	STN Full
Classification	94.47	96.06	96.30
Segmentation	NA	83.67	NA
Time (s)	1426	1195	NA

3.4.3 Street View House Numbers

The Street View House Numbers (SVHN) dataset offers a tougher 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 32x32 MNIST like digits with the additions of color, variable contrast, and some confusing data and the full images which contain extensive backgrouds and multiple digits in addition to the challenges in the cropped format.

In this dataset, the attention mechanism 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.

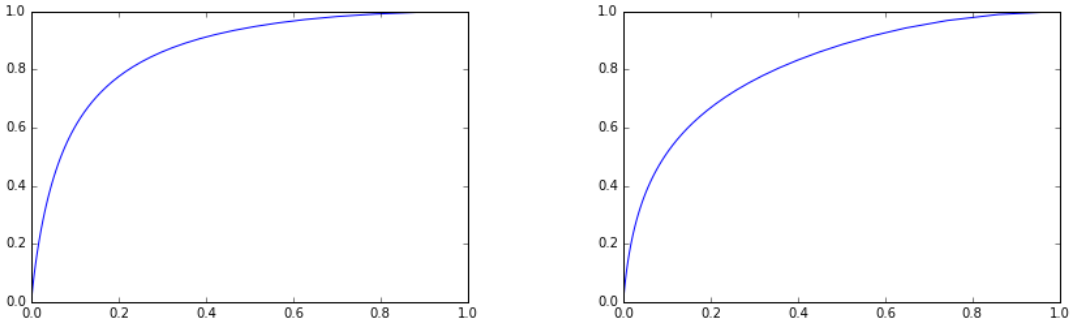
Tables 3-3 and 3-4 show the classification accuracies, segmentation accuracies, and time per training epoch on the dataset with the enlarged bounding boxes created by the procedure

Table 3-5. Unsupervised SVHN Results on the Full Image Dataset

Method \ Metric	RWTA w/ Attention	TDN w/ Attention	Autoencoder	VAE
Classification	73.30	71.59	5.13	8.34
Segmentation	72.43	72.43	NA	NA
Time (s)	22658	2943	3689	4016

Table 3-6. Supervised SVHN Results on the Full Image Dataset

Method \ Metric	CNN Full	CNN FOA	STN Full
Classification	68.15	80.58	28.03
Segmentation	NA	72.43	NA
Time (s)	2397	1506	4549



A B

Figure 3-6. ROC curve for finding the bounding box containing all numbers from the SVHN dataset (A) and ROC curve without the postprocessing (blurring and centering) (B).

outlined in [82]. Segmentation accuracies are calculated by dividing the intersection of the true and predicted bounding boxes by their union. The STN results are reported by the authors [60] and do not include segmentation data or time information.

RWTA and TDN both vastly outperform traditional unsupervised learning strategies that do not use attention. Once again there is only a small difference between the RWTA and TDN since the videos are artificially produced from still images. In addition, it also improves the performance of a CNN to the state-of-the-art results reported by the STN.

Most results reported on this dataset uses the cropped digits, and even 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 images with no additional data about bounding boxes or the number of digits contained within the image. This means that our attention mechanism must localize the address location, segment, then identify each digit for a success. This is a

much harder problem than simply classifying boxed digits since it has combines the problems of localization and classification in a paradigm that does not have fixed output size.

Tables 3-5 and 3-6 show 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.

Figure 3-6 shows the ROC curves for the 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 [83]. The axes were normalized for the purpose of showing the curve, but the negatives outnumber the positives in the dataset by a factor of 2.46.

Figure 3-7 shows the working of the attention system on an example SVHN image. It first segments a salient area, then breaks that area up into the digits that compose the two object found in that location. 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.

3.4.3.1 SVHN network

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] - fc[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 STN's 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 has an order of $k = 1, k = 1, k = 1$ and a shape parameter of $\mu = .1, \mu = .3, and \mu = .8$. The neighborhood kernel has an order of $k = 13, k = 9, k = 5$ and $\mu = .3, \mu = .5, \mu = .7$. 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.

3.4.4 VQA

For a second test of the combination of a bottom-up attention mechanism and a network-based feature extractor, we will use visual question answering (VQA). In VQA, the system is provided an image and a corresponding question, then must take the two and provide a correct answer. In contrast to description of visual content, which has been well studied, VQA must take information from both a question and an image, and combine them to find an answer. This is a more complex task that image annotation or question answering using just text because it involves different types of information being processed together.

The traditional technique for VQA is combining a CNN [84] for image feature extraction, an RNN for question language processing [85], and an MLP to combine the features and output and answer. This technique is also known as Neural-Image-QA [86], and a basic diagram is shown in Figure 3-8a. However, as shown in the VQA challenge, this approach begins to fail when confronted with many objects and a question that requires higher level reasoning [87].

To help mitigate these concerns, we propose to augment this architecture with a visual attention system. This attention system will separate each object into separate crops of the image. By doing this, we allow the CNN to focus on identifying single objects rather than trying to extract features from the entire image at once that explain both the objects and their spatial information. Figure 3-8b shows the new architecture including the attention system.

By cropping each object out of the image, the location and scale information is also extracted simply by storing the location and size of the cropped region. By also passing this information to the MLP, we can relieve the CNN of extracting features that explain spatial relations between object and make them explicit. By simplifying the task of the CNN to object recognition, we can ensure the MLP has the relevant information to solve the question.

The results were computed on the cluttered MNISTVQA dataset [88]. This is a flexible test environment that places MNIST digits into a larger background along with clutter, as well as generates questions from nine different categories to correspond with the images. This is useful both because it has a finite set of objects (which allow the CNN to easily be trained to extract features), as well as gives the ability to evaluate the performance of the architecture on different types of questions. In addition, the amount of clutter, number of digits, and their size are all customizable.

The questions are broken down into three major groups: content, relation, and arithmetic. Each group then has three subgroups of questions. The content group focuses on object recognition, asking if a certain number is present in the image, whether the number present is odd or even, and how many numbers are in each image. The relation questions deal with the spatial and numerical relations between the digits, such as their positions, values, and scales. Finally, the arithmetic questions ask for the total sum of the digits contained in the image, their total product, and the size of the range between the maximum and minimum value. Figure 3-9 shows example images and possible questions from the dataset.

The architectures trained were as follows. We adopt the notation that conv[N,w,s] denotes a convolutional layer with N filters of size $w \times w$, and stride s; fc[N] is a fully connected layer with N units; and max[s] is a $s \times s$ max-pooling layer with stride s. The baseline architecture CNN consisted of conv[8,3,3] - max[2] - conv[16,3,3] - max[2] - conv[32,3,3] - max[2] - conv[128,3,3] - max[2] - conv[128,3,3] - max[2]. The RNN for question embedding was a single LSTM. These outputs were merged and fed into an MLP consisting of fc[50] - fc[50] - fc[779], with 779 being the number of words in the vocabulary. Each

convolutional and fully-connected layer was followed by a RELU activation except for the final fully-connected layer which had a softmax for the output. The focus of attention used the same CNN, RNN, and MLP architectures with the exception that the CNN had one extra fc[11] to output the digit class (0-9 and a noise class).

Since the images in this dataset are relatively small at 64x64, we used a two scale gamma kernel. The gamma saliency parameters for the larger scale were a center kernel with $k = 1, \mu = .2$ and a surround kernel with $k = 8, \mu = .5$. The second scale has a center kernel with $k = 1, \mu = .5$ and a surround kernel with $k = 5, \mu.5$. The focus of attention was used to create three crops per image since that is the maximum number of digits, which necessitates the use of an extra noise class in the CNN used to classify each crop.

Figure 3-11 shows an example image from this dataset containing three digits. For this work, the possible number of digits was constrained to be from one to three, normal 6x6 clutter was used, and the scales set to be uniformly random from .6 to 1.2 times their original MNIST size.

As an initial test, we validated the visual attention by computing the ROC curve shown in Figure 3-10. Despite being unsupervised, the visual attention via gamma saliency is able to identify most digits. As a second test, we tested the output of the CNN with visual attention with the output of the CNN used in the baseline architecture. To have the baseline CNN output classes, three parallel fc[11] layers with a softmax activation were added to the top. The CNN with visual attention were all able to classify all three digits correctly 89.5% of images, while the baseline CNN could only correctly classify 86.0%.

Table 3-7 shows the results of the two architectures on this dataset. Each architecture was trained for 200 epochs on a training set of 100,000 image/question pairs, validated on 25,000, and tested on another 25,000. Despite using an unsupervised attention mechanism to crop image patches, the results of the architecture including the focus of attention improve the results in eight of nine categories. Overall, it performs roughly 6% better on the entire dataset. Also, since the focus of attention quickly crops the image into three patches, the training

Table 3-7. Supervised SVHN Results on the Full Image Dataset

Method	Contains	Odd/Even	#Imgs	Position	Value	Size	Sum	Product	Range	Total
Baseline	81.5	93.4	98	57.7	76.4	58.5	43.7	60.5	52.4	69.3
FOA	80.6	93.7	99.8	60.8	85.0	61.8	57.9	66.7	70.9	75.3

Table 3-8. Results on Cluttered MNISTVQA

time is decreased from 164 seconds per epoch to 88 seconds, training twice as quickly for an architecture of similar size.

The categories that show the most improvement are the ones that require the higher level reasoning, while the simpler content questions perform roughly the same as the baseline architecture. Interestingly, though, despite explicitly providing information about the location and scale of the digits from the focus of attention, other categories see larger improvements, such as the range and sum categories. One possible explanation for this is the embedding of the features sent to the MLP: since each image patch produces a vector containing the encoding of the class, location, and scale, 10 of the 13 features are the class. This domination by the class of the digit could be limiting the impact of providing the location and scale.

3.5 Conclusion

In this chapter we propose an architecture that mimics the function of two fundamental mechanisms of the human vision system - a saliency based mechanism for the spatial attention (approximating functions of the HVS dorsal pathway) and a network based representer (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. As currently constructed, the system is not a true dual pathway architecture - the attention mechanism feeds information directly into the feature extractor without receiving feedback or incorporating information from both pieces into a whole without post-processing. However, this does provide a foundation for a true dual pathway system, one that could

eventually include more pieces such as feedback between the two systems and a memory that could incorporate the two sets of information together.

In addition to removing irrelevant information, the attention pathway also is a way to create videos from still images, which adds data augmentation to the new architecture. To take advantage of this, we use a feature extraction network that has both spatial and temporal pathways to learn the structure of the data. Combining all of these produces leads to learning robust features that explain the relevant information in images without the need of supervised labels.

One major drawback to this technique is the computation time required to train both the two pathways, especially the temporal pathway that uses an RNN. To mitigate this, we simply combined the features from different frames using learned weights, which produced similar results at much faster speeds. We believe this was possible since the underlying data in the frames was the same - the videos were created simply with rotation. In a true video, the RNN would be necessary to learn the temporal relationship between the frames.

Future work includes extending and testing the proposed method on video. The dual spatial-temporal architecture of the RWTA makes it uniquely suited to extracting features from videos, so pairing it with an appropriate attention mechanism that takes time into account should learn robust features. In addition, we could research improving 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.

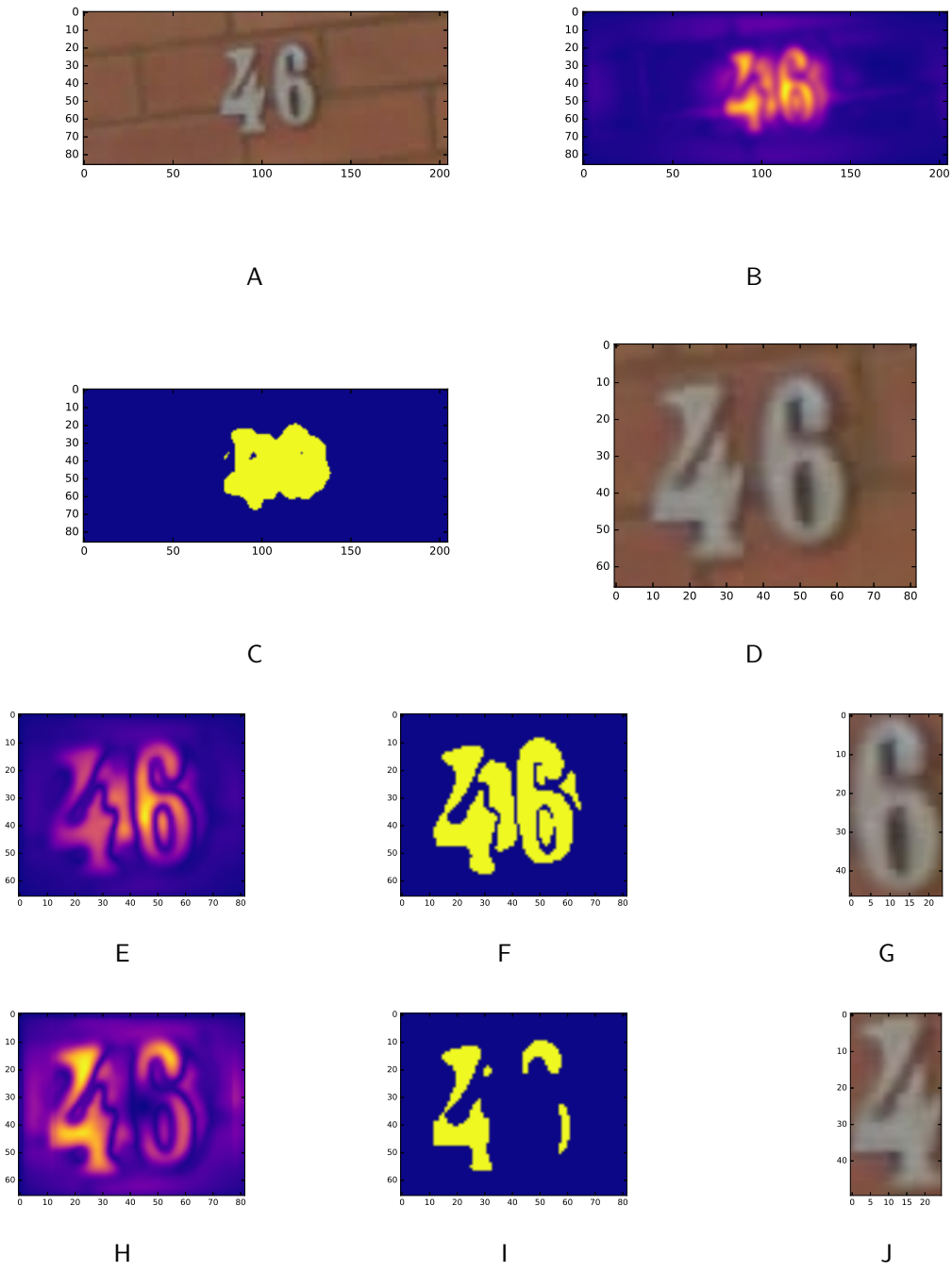
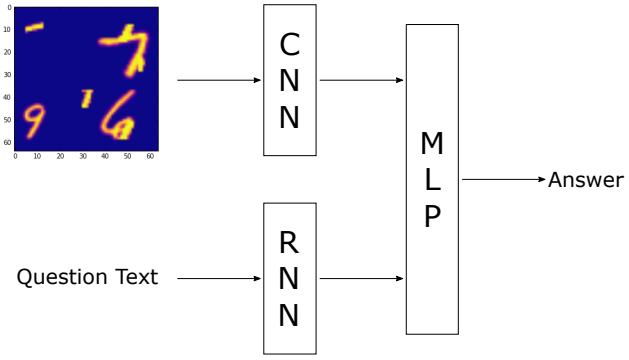
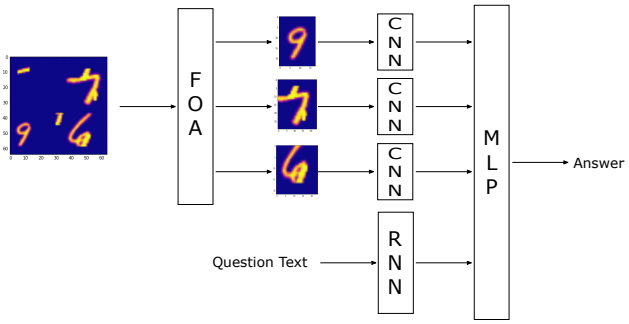


Figure 3-7. Initial SVHN image (A), saliency map produced by Gamma Saliency (B), thresholded saliency map (C), the cropped patch around the house numbers (D), the initial saliency map from the crop (E), the thresholded version of that map (F), the patch extracted around the object with the highest saliency (G), the saliency map with return inhibition around the most salient point (H), the thresholded map (I), and the cropped second object (J).



A The baseline architecture



B The architecture with visual attention

Figure 3-8. VQA network architectures.

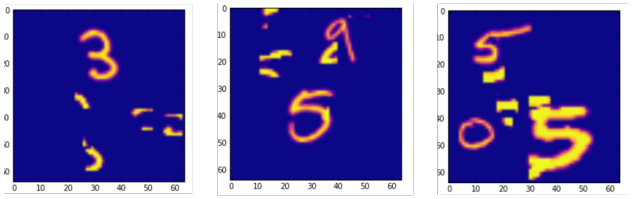


Figure 3-9. Example images and questions from the Cluttered MNISTVQA dataset. Each question type has multiple possible wordings.

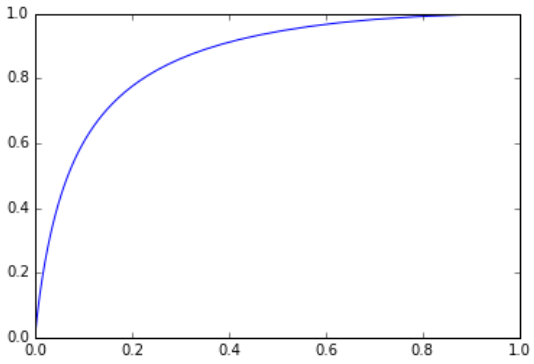
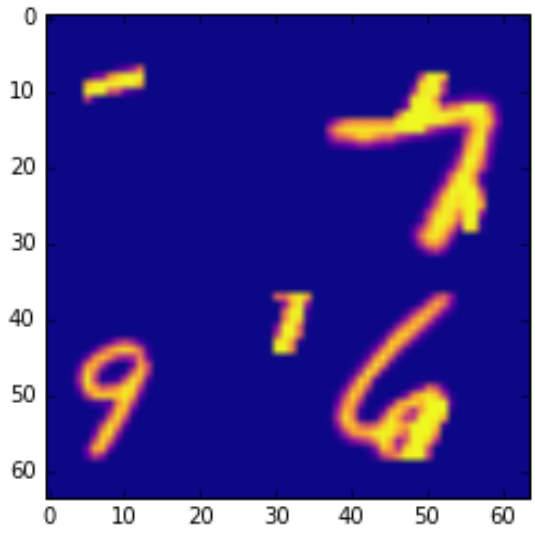
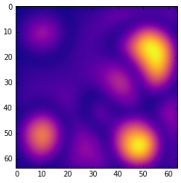


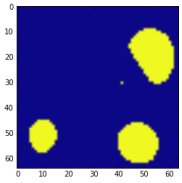
Figure 3-10. ROC curve for Gamma Saliency on the cluttered MNISTVQA dataset.



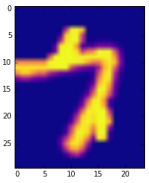
A The multi-digit cluttered MNIST image



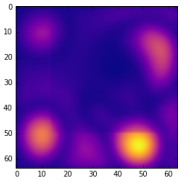
B The initial saliency map



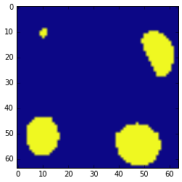
C The thresholded saliency map



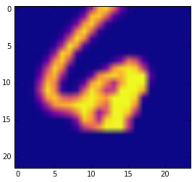
D The first extracted patch



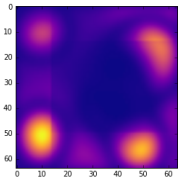
E The saliency map with return inhibition around the first patch



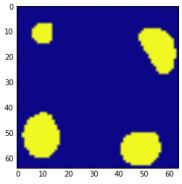
F The thresholded map



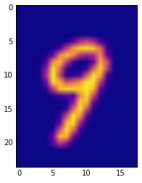
G The second extracted patch



H The saliency map with return inhibition around the second patch



I The thresholded map



J The third extracted patch

Figure 3-11. The focus of attention separating and extracting areas of relevant information.

CHAPTER 4
DIRECTED VISUAL SEARCH

4.1 Background

Searching for objects in complex, cluttered scenes is a difficult problem in computer vision. Neural networks have proven to be useful at classifying objects in images, but most classification datasets still deal with single object images, which excludes a large portion of natural images.

Recently, neural networks have been applied to localizing specific objects, though different frameworks such as R-CNN [14] and YOLO [19]. These approaches train convolutional networks to select from a grid of heuristically defined bounding boxes [18] before extracting features that are then sent to a standard dense classifier. These are a step beyond simple classification networks because they can find multiple objects within a single image, but they suffer from being brute force algorithms that use heuristically defined bounding boxes as priors. They also take a large amount of data to train.

On the other hand, the Human Vision System is capable of processing complex scenes in an incredibly small amount of time. It is able to accomplish this by separating the tasks of localization and identification between two separate pathways [4]. A fast, low-resolution pathway localizes interesting areas and drives fixations to these areas, while a second, high-resolution pathway takes these fixations and identifies the contents [3].

However, we know that the Human Vision System does not choose fixations solely on visual data, but these saliency measures are all bottom-up measures, meaning they assign saliency values based only on the input image. Often a person is searching for a specific object in a scene for a variety of reasons. This means a specific target must be found and extra information about the target must be incorporated. Torralba et. at accomplished this by learning mean vertical positions of household objects and biasing the saliency maps towards these locations [37]. Kanan et. al [30] used a probabilistic model guided by coarse features that predict object locations. Frintrop et. al [89] augmented the classic Itti-Koch saliency with

a set of learned weights on top of the input feature maps. Top-down measures overall prove much more effective at finding specific objects and predicting search patterns.

In addition, the concept of saliency has been more recently used to study activations in neural networks. The first work in this area attempted to find the images that maximized activations at each layer of the network [90]. The work by Simonyan et. al [70] learns the images that most activate the nodes for specific classes. It can also provide a weak localization for class based on backpropagating the gradients through to the input image.

Since neural networks have proven adept at learning features to distinguish objects while saliency measures are fast, reliable indicators of human attention, we propose to combine these two approaches. Rather than input a traditional RGB image to our saliency measure, we can use feature maps based on the learned convolutional filters of a neural network. By doing this, we are starting with a set of features that has been optimized to distinguish the objects. Furthermore, by learning a set of weights on these filters, we can bias the saliency towards specific objects that activate certain filters in the network.

Section II will present the method used for the top-down attention mechanism. Section III will present the results, and Section IV will conclude the paper.

4.2 Top-Down Gamma Saliency

The visual search system is based on the same Gamma Saliency as the bottom-up visual attention [68], detailed in Chapter 2. This is a center-surround method that takes advantage of the gamma kernel to compute a fast, multi scale difference between a center and surrounding regions.

$$S = \frac{\sum_{n=1}^{N} w_n^i |g \bullet C_n|^\alpha}{N} \tag{4–1}$$

In bottom-up Gamma Saliency, these features maps would be the channels of an RGB or LAB image. However, to make top-down Gamma Saliency, we propose to use a set of feature maps C from a fully convolutional network [91]. Unlike fully-connected layers, convolutional layers of a neural network are agnostic to the size of the input. Therefore, it is possible to train

a classification network on a standard dataset (such as MNIST), then strip the convolutional layers from the network and use them to preprocess images of any size. In doing this, a set of feature maps are created that can be used to distinguish between the objects in the training set.

In addition, a set of weights w can be learned on these maps to bias the saliency measures towards a target object. Since the maps were trained as part of a classification network, the features contained in these maps have already been optimized as a part of a system that separates the set of objects in the images. By learning a set of weights corresponding to each object, we are changing a bottom-up saliency measure to a top-down measure capable of finding objects in fewer fixations.

Eqn 4–2 shows a simple way of learning of learning the weights w_n for each object i. Over a training set that contains the locations of the objects, we can compute each raw saliency maps as in Eqn 4–1 assuming each $w_n^i = 1$. We can calculate the ratio of the saliency inside the bounding box $s_{I_n}^m$ to the saliency outside the box $s_{O_n}^m$ and find the mean value across the training data. In this equation, m is the specific image and n is the object class. This gives us the average weight for each object for each feature map, with maps that have higher saliencies inside the bounding boxes as compared to outside receiving higher weights. This is similar to the work of Frintrop et. al [89], except instead of using pre-defined feature maps designed by hand, the feature maps are extracted from a network trained on the objects in the scene.

By doing this, we create a weight matrix w_n that contains a weights for each feature map and each class. Each entry in this weight matrix corresponds to how highly each specific feature map activates for each class. Feature maps that always activate for certain digits are given a high weight, while feature maps with fewer activations are given a lower weight. Using these, we can quickly process an image and gain an idea as to the location of certain objects.

$$w_n = \sum_{1}^{M} \frac{\frac{s_{I_n}^m}{s_{O_n}^m}}{M} \tag{4–2}$$

After weighting each map, we process and combine them in the same manner as bottom-up Gamma Salinecy. By processing them in this way, we end up with a set of feature maps that has been trained to separate the objects found in the scene. We can convolve the maps with the gamma kernel, then instead of averaging them, we can learn a set of weights on top of these. By learning the weights, we can bias the saliency towards certain objects, giving them a higher saliency than they would have in a pure bottom-up metric.

4.3 MNIST Search Results

To test this framework, we used a multi-digit MNIST environment [92] [88], shown in Figure 4-1. This environment is useful for a proof of concept since it is a built on a simple, easily learning dataset in MNIST [93]. This ensures that the complexity in the problem comes from the extra space and multiple objects found in the scene.

For the first test, the environment consists of 128x128 backgrounds with 5 random MNIST digits placed at random locations. In this initial test, the 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 specific digits.

The bottom-saliency used parameters $k = (1, 9)$ and $mu = (.2, .5)$. The top-down saliency used the same k and mu parameters. The network used to compute the feature maps used as inputs to the top-down saliency was a simple convolutional network trained on MNIST. Figure 4-2 shows an example of the feature maps used as input to the saliency. Figure 4-3 shows an example image, as well as the bottom-up and top-down saliency measures for that image. For this figure, the saliency was weighted to find the "1" digit.

For the bottom-up saliency, each digit was found in an average of 3 saccades. Using the top-down framework reduces that to an average of 1.73 saccades per digit, meaning we successfully biased the saliency towards the object we want to find.

For a second test, we used a version of Multi-digit MNIST with correlated clutter. Fig 4-4 shows an example image of this environment. We created 100,00 training images, as well

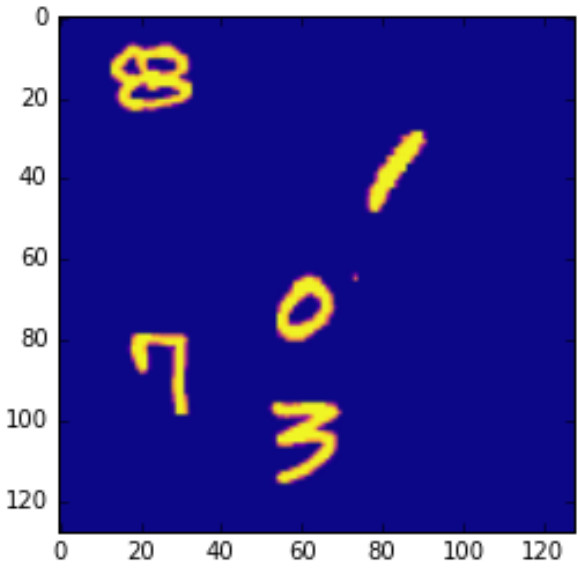


Figure 4-1. Example Multi-digit MNIST image.

Input	Full Network	Bottom-up	Top-down
Full Image	72.3	N/A	N/A
One Saccade	N/A	61.4	84.1
Two Saccades	N/A	68.3	92.3

Table 4-1. Classification Results for Finding the Target Digit

as 25,000 validation and 25,000 test images. In this case, we are trying to answer a simple question for each canvas - does the image contain a specific (random) digit?

For this test, we are using three different architectures, shown in Fig 4-5. The first is simply a network given the full image and a one-hot vector with the target digit. The second network preprocesses the image with bottom-up Gamma Saliency before using a pre-trained MNIST classifier on the extracted image patch. Finally, the third architecture uses top-down Gamma Saliency to attempt to find the target image before also sending the extracted image patch to a pre-trained MNIST classifier.

Table 4.3 shows the results from this experiment. The full network has trouble combining the information from the cluttered image and the target, correctly identifying whether the target is present in less than $3/4$ of the test images. However, this is an improvement over

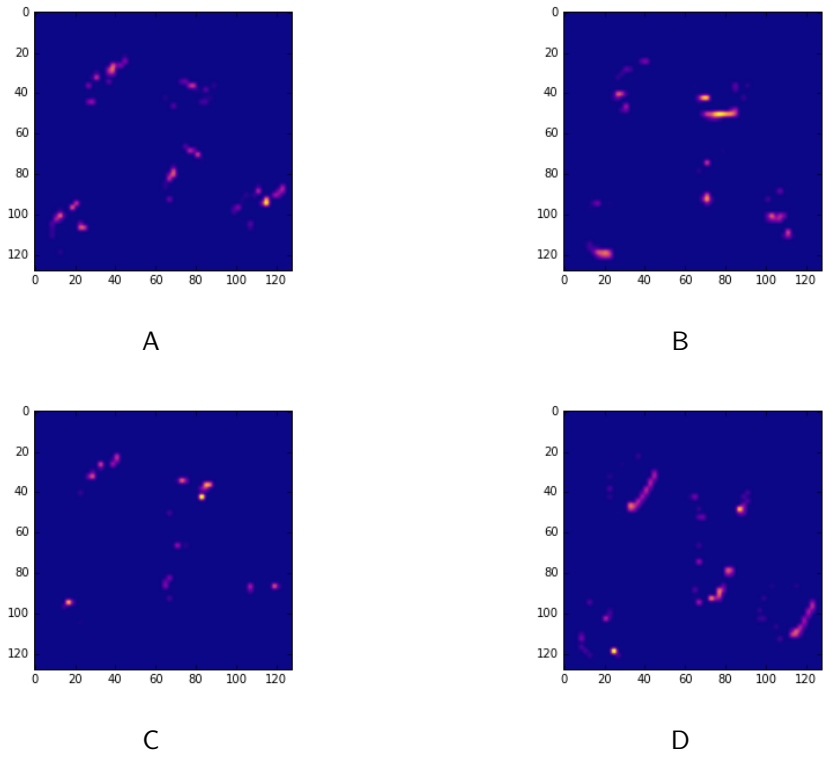


Figure 4-2. Example feature maps from the convolutional layers of the MNIST classifier.

the bottom-up saliency. The bottom-up saliency is unable to identify the target often unable to identify the target in either one or 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 pure neural network, correctly identfying whether the target is present in over 90% of the images when given two saccades. It is able to accomplish this because information about the target is used in the attention mechanism. This gives it the benefit of being able to remove background information, unlike the full network, while also concentrating on the correct foreground data, unlike the bottom-up saliency. In addition, it is able to achieve these results despite using a smaller neural network, since the network needs to focus only on one task (digit classification) instead of learning the entire problem end-to-end.

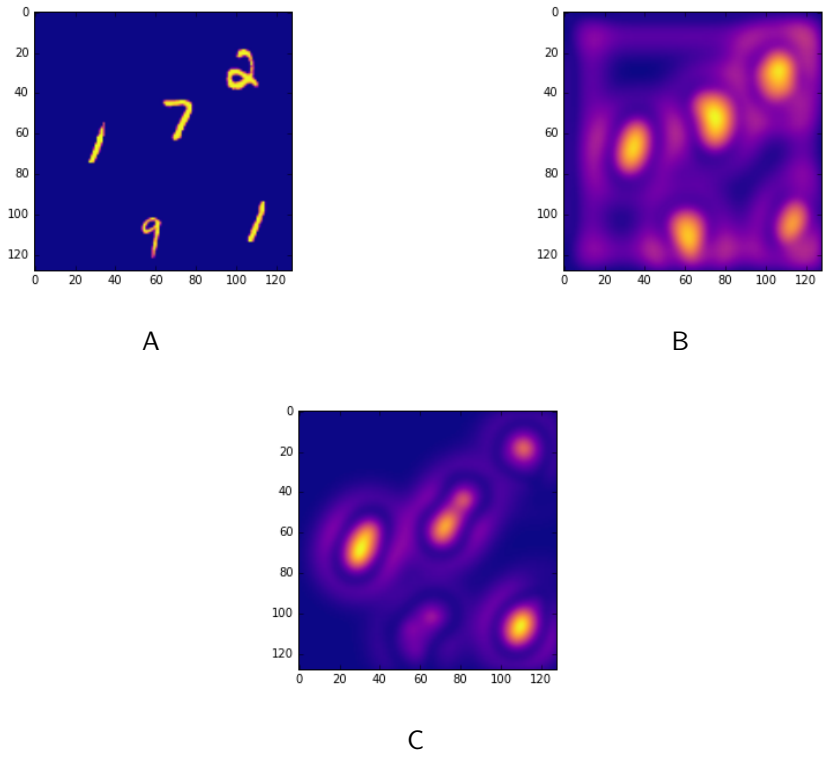


Figure 4-3. Top-down vs. bottom-up search. The initial image (A), the bottom-up saliency map (B), and the top-down saliency map (C).

The networks used were as follows, where conv[X,Y,Z] is a convolution layer with X filters of size YxZ, fc[X] is a fully-connected layer with X nodes, and pool[X,Y] is a max pooling layer with a pool size of XxY. The full network 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 MNIST classifier used for the saliency based architectures 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]. The feature maps for the top-down saliency were extracted at the end of the final convolutional layer before pooling.

4.4 Naturalistic Search Results

In addition to testing on synthetic environments such as cluttered MNIST, we test the new search technique on a set of naturalistic images containing animals. This dataset consists

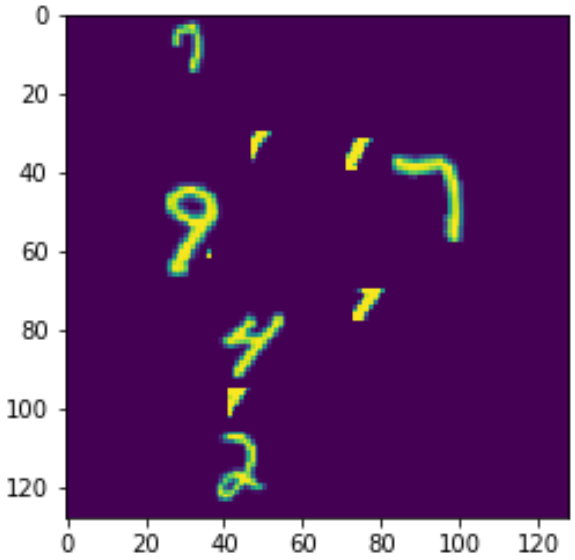


Figure 4-4. Example Cluttered Multi-digit MNIST image.

of 80 images of nature: 20 images containing birds, 20 more with similar background but no birds, 20 with snakes, and 20 similar images without snakes. Search in these images is much more difficult than the synthetic environments due to the backgrounds being much more complex as well as the targets having natural camouflage. Fig 4-6 shows example images from this dataset.

Since we are dealing with such a small number of images it would be impractical to train an entire neural network to differentiate birds from snakes from background. Therefore we will use transfer learning [94] where we will use the feature extractor from a network pre-trained on a much larger dataset. For this task, we will use the VGG16 network [95] which is a 16 layer network with 13 convolutional layers and three fully-connected layers trained on Imagenet [96]. The output of the final convolutional layer (before the subsequent max-pooling) will be used to generate the feature maps used as inputs to the top-down saliency. This produces 512 feature maps.

To test the top-down saliency measure against the bottom-up saliency measure, first we used each algorithm to attempt to find the animal in the 40 images that contain a target.

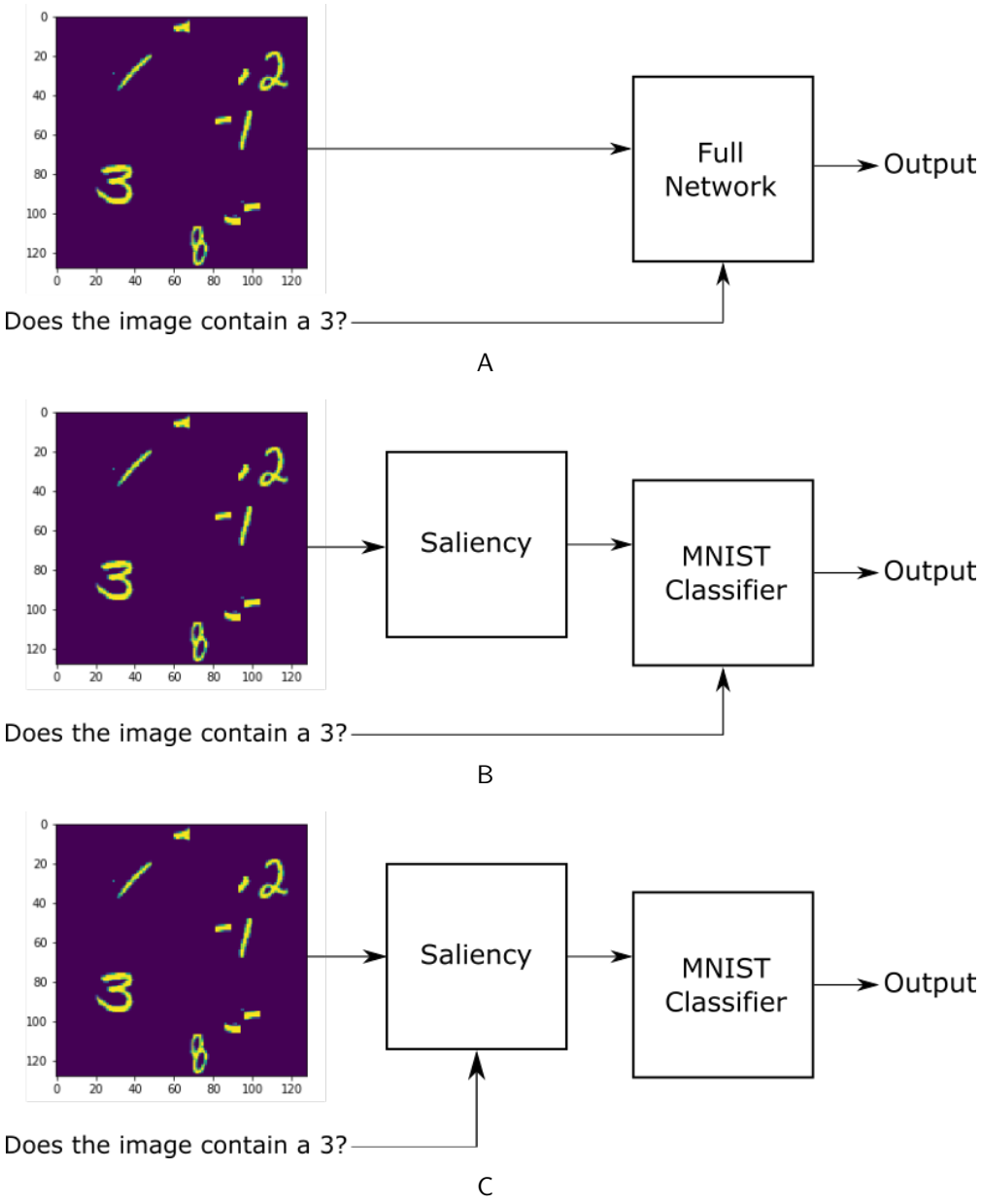


Figure 4-5. Top-down vs. bottom-up search. The initial image (A), the bottom-up saliency map (B), and the top-down saliency map (C).

The target is considered found if the image patch produced by the saliency map has an intersection over union of greater than .5 with the true bounding box. The patches are obtained using the method described 3.3.2 without the extra augmentation used to create videos for the RWTA. The weights for the maps in the top-down saliency were created by randomly selecting an image with a bird and another with a snake and using these. Both

A B

C D

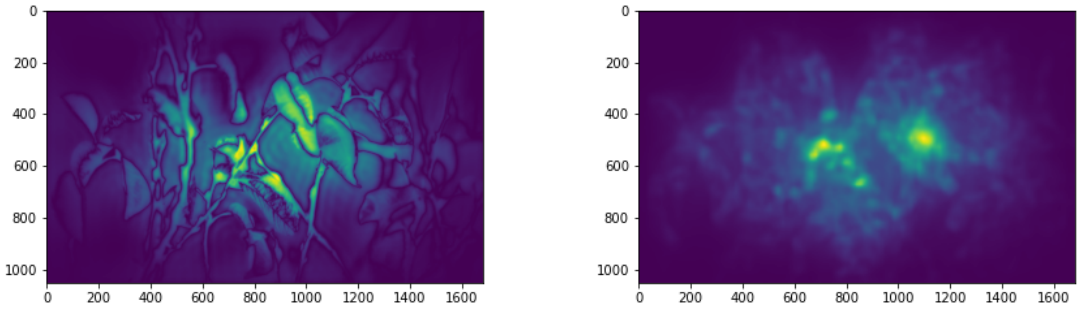
Figure 4-6. Example images from the naturalisitic search dataset. An image containing a bird (A), an image containing a snake (B), an image with no bird (C), and an image with no snake (D).

the top-down and bottom-up saliencies are computed using $k = (1, 60, 1, 38, 1, 19)$ and $\mu = (.05, .5, .1, .5, .5, .5)$, with $n_1 = n_2 = 400$.

Across the 38 remaining images that contain targets, the bottom-up saliency finds the target in an average of 39.8 saccades, while the top-down saliency finds the target in an average of 5.9 saccades. The top-down saliency both finds the targets faster and produces much more realistic maps. Fig 4-7 shows an example image containing a bird along with the bottom-up and top-down saliency maps. The bottom-up saliency map highlights the edges of the central objects fairly equally, while the top down saliency focuses on the bird even though it is partially hidden by leaves in the foreground.

For a second example, Fig 4-8 shows the patches extracted using bottom-up saliency, where it takes 6 fixations to find the bird. In contrast, Fig 4-9 shows the same image with

A



B C

Figure 4-7. Top-down vs. bottom-up saliency on the naturalistic dataset. An image containing a bird (A), the bottom-up saliency map (B), and the top-down saliency map (C).

the patches extracted using a top-down saliency method. It only takes 4 fixations to find the bird for the top-down saliency, and it localizes the bird with a higher accuracy. In this case the numbers are relatively close, but there are other images where the bottom-up saliency takes over 80 fixations to locate the target, while the maximum number of fixations for the top-down saliency was 16.

In addition to how quickly the targets are found, we can analyze the top-down search by how well it matches a human searching the same scene. To do this, we will analyze

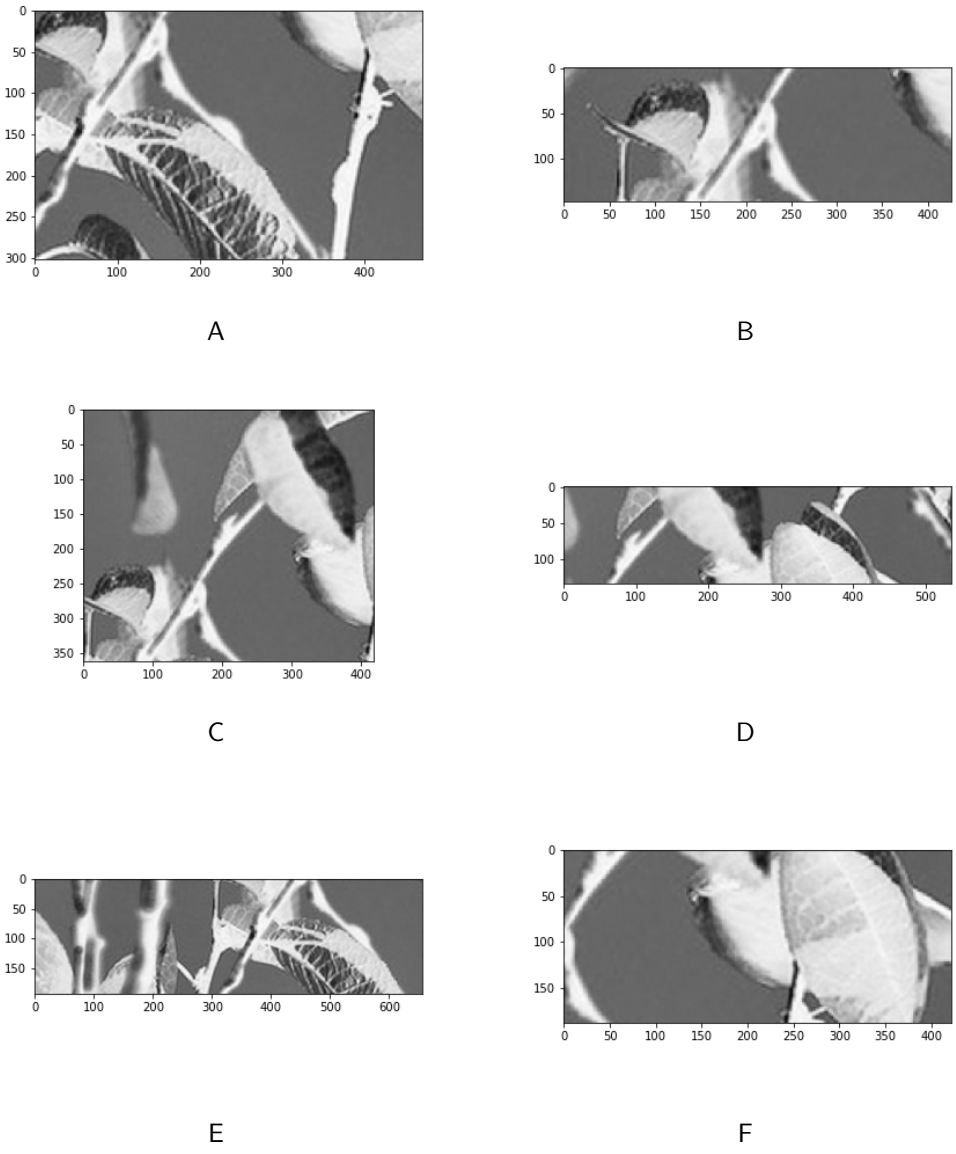


Figure 4-8. Patches extracted using bottom-up saliency. The first patch (A), the second patch (B), the third patch (C), and the fourth patch which contains the bird (D).

the saliency maps as a standard saliency measure. The dataset of 80 images also includes eye-tracking data where participants were asked to find the bird or snake contained in each image (including the images with no target). From this eye-tracking data, we can create fixation maps that we can compare to saliency maps as in Chapter 2. Fig 4-10 shows an example image with the eyetracking data of a single person overlaid. The person scans most

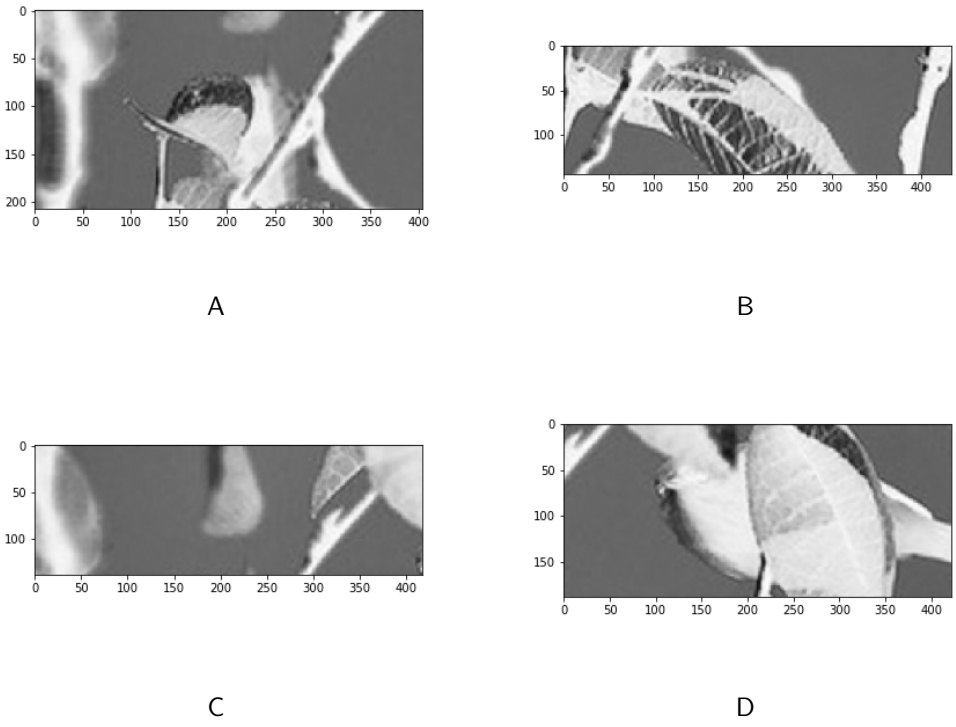


Figure 4-9. Patches extracted using top-down saliency. The first patch (A), the second patch (B), the third patch (C), the fourth patch (D), the fifth patch (E), and the sixth patch which contains the bird (F).

of the image before finding the target rather than localizing it instantly, showing that this is a hard task even for humans.

Since the participants in the study were told to find the animals, we expect the top-down saliency the outperform the bottom-up saliency because we are specifically biasing it towards finding the targets. Table 4-2 shows the results of the two saliency measures with the commonly used metrics, while Fig 4-11 shows the ROC curve for each. The top-down saliency does outperform the bottom-up version greatly in all of the metrics, showing that the top-down saliency does approximate human search better than the purely bottom-up approach. Fig 4-12 shows an example fixation map along with the associated top-down and bottom-up maps.

Table 4-2 does not show how successful either bottom-up or top-down saliency are at finding the targets, but how well the fixation locations match the locations searched by

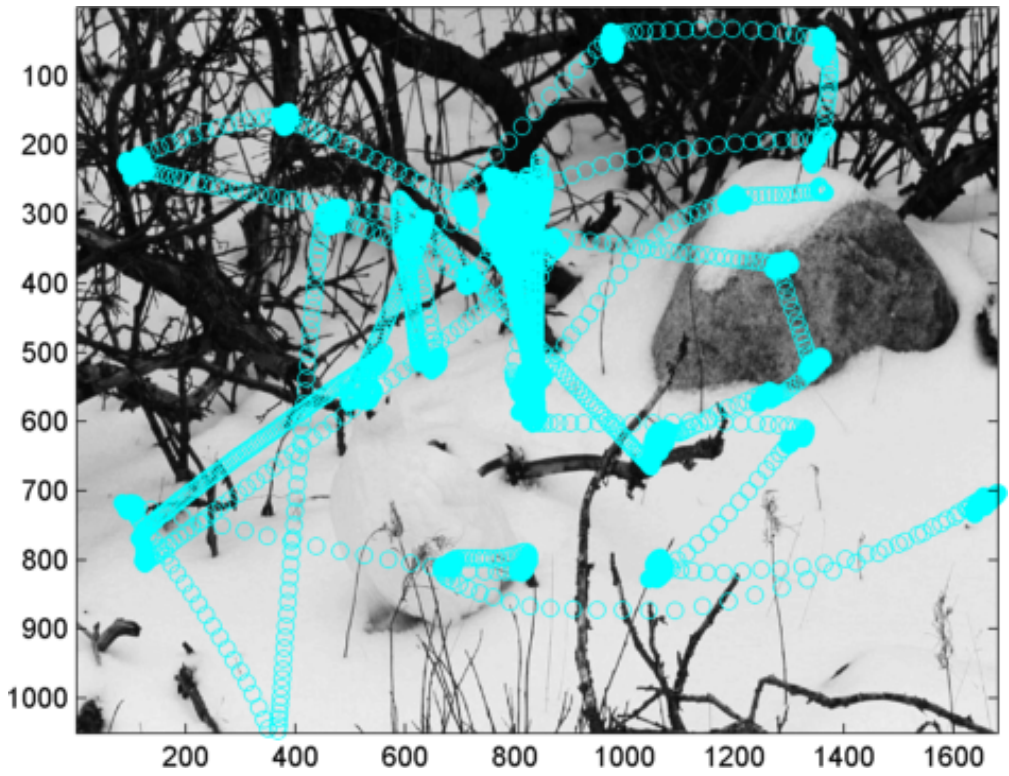


Figure 4-10. Search image with overlaid eye-tracking.

Table 4-2. Comparison of Human Search with Top-Down and Bottom-Up Saliency

Method \ Metric	ROC (Judd)	ROC (Borji)	Similarity	Correlation	NSS
Bottom-Up	.586	.427	.362	.267	.398
Top-Down	.792	.615	.562	.331	.461

humans. Bottom-up saliency measures have been shown to correlate with free-exploration in images [22], but fail when the fixation maps are produced by humans that are searching for specific objects. The top-down saliency mitigates this problem by using features to search for these targets. Doing this not only finds the targets in less fixations, but searches the images in a way that more closely matches human exploration than the bottom-up method.

4.5 Network Structure

In addition to using this framework for visual search, it can be used to learn about the structure of the network. By learning which feature maps correspond to each object, we can ascertain what each filter in the network has learned from the data. This opens up many

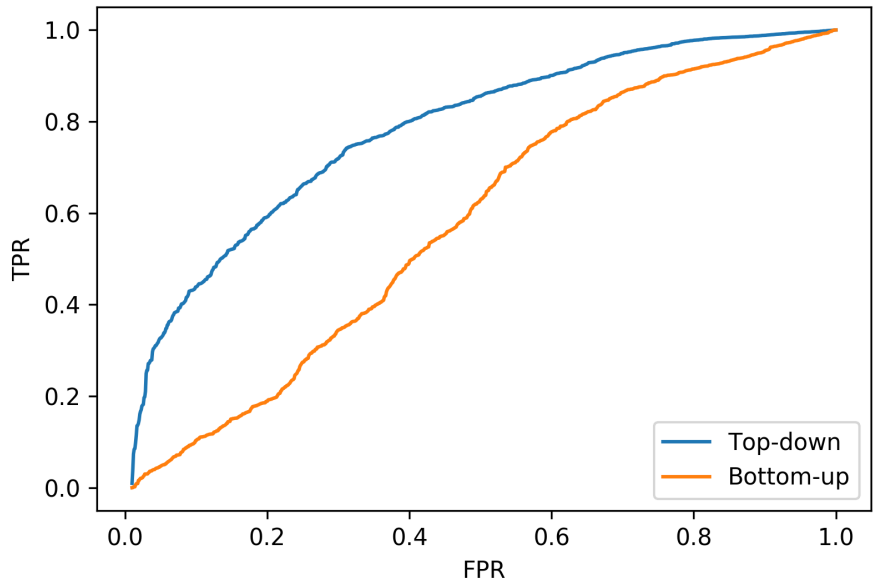


Figure 4-11. ROC curves for top-down and bottom-up saliency for a search task.

possibilities, including partitioning the network by which filters are useful for which objects and features.

To illustrate this, we consider the MNIST classifier used in 4.3. The output of the final convolutional layer outputs 64 feature maps. A set of ten weights (one per class) is learned on each feature map. By visualizing these weights, we can learn which feature maps correlate with each label. Fig 4-13 shows the weights learned on the Cluttered MNIST dataset.

This could be useful in demystifying neural networks. Since these networks are nonlinear and often very deep, the flow of information between the input and labels is rarely understood. This provides a simple method of the relevance of the feature maps from each convolutional layer to the outputs of the network. In this example, it shows that certain feature maps have a high correlation with a single digit (such as the 63rd feature map with digit 8), while others occur across most or all of the classes.

In this example, it is clear that for a majority of the maps, the linear weights learned are not useful for separating the objects via the top-down saliency. However, this network was able to achieve over 99% test accuracy while classifying MNIST. Therefore, it stands to reason

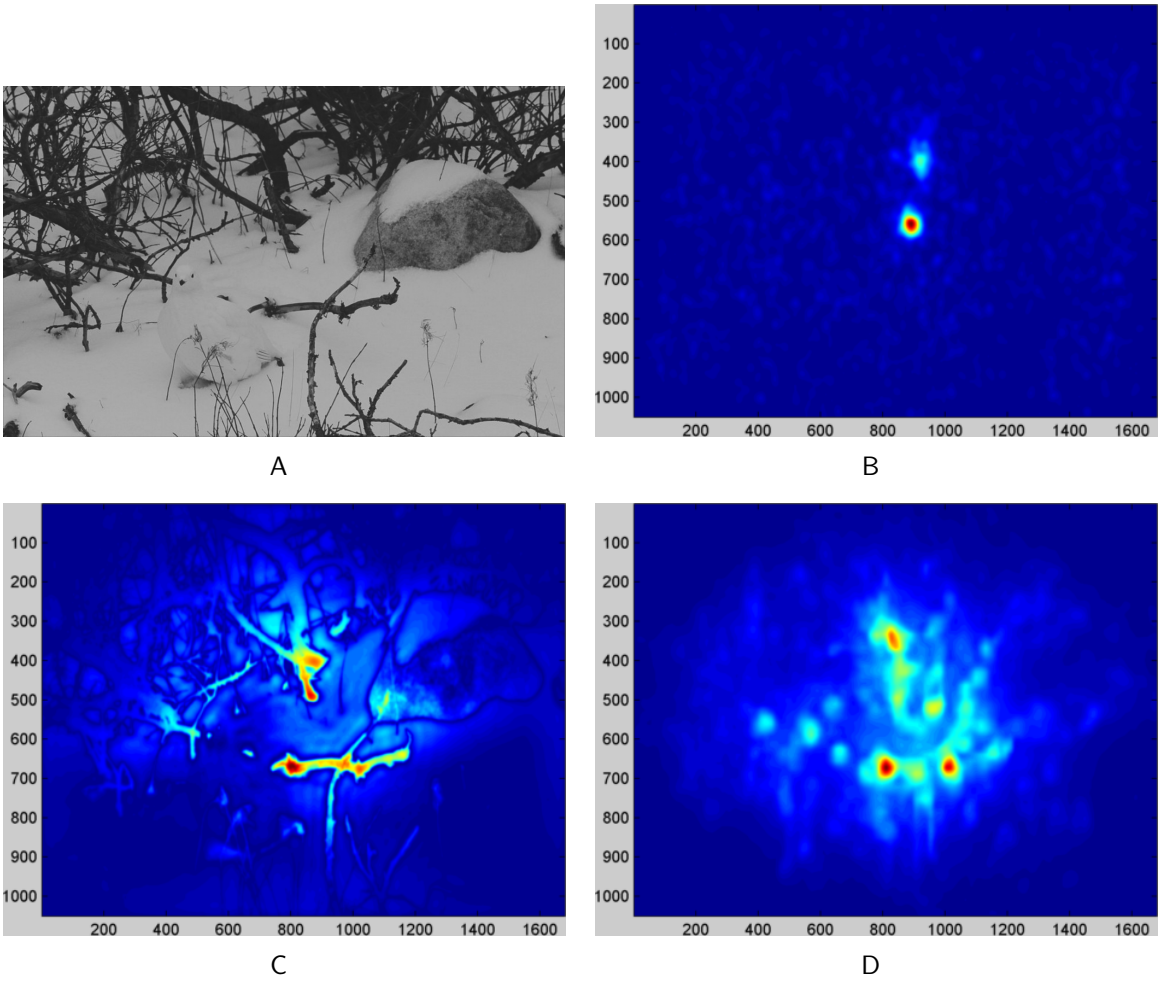


Figure 4-12. A fixation map from the naturalistic images compared with the top-down and bottom-up saliency maps. The original image (A), the fixation map (B), the bottom-up saliency map (C), and the top-down saliency map (D).

that using nonlinear weights instead of the current linear weights could further improve the localization for each class. This idea could be implemented with fully-convolutional networks [97], which could be used to produce a full map for each class.

Another potential use for this that requires further exploration is determining the amount of feature maps necessary in each layer. In this example, many of the maps have the same linear weights across all the classes. This could be a sign that maps are learning redundant or irrelevant information, and that the size of the network could be reduced. However, this requires testing in a more complex environment.

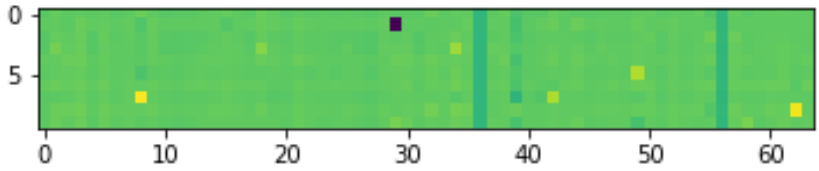


Figure 4-13. The saliency weight map from the MNIST classifier.

4.6 Top-Down Search Conclusion

In this chapter we presented a novel top-down attention algorithm based on combining a data-driven center-surround saliency metric and the convolutional filters in a trained neural network. By combining these two architectures, we are able to bias the attention mechanism towards specific objects, creating a search mechanism that finds objects faster than a purely bottom-up approach.

This is a preliminary work used to measure the effectiveness of using the feature maps of a trained classification network for localizing and recognizing objects using a small number of saccades. Further studies would include testing on more naturalistic datasets and comparing the saccade data and saliency maps to the data generated by human eye-tracking during searches. In addition, the architecture can be improved beyond learning a simple linear weighting on the feature maps, such as using an extra convolutional layer to predict locations in a feature map rather than using a separate saliency metric.

Lastly, this also can be used as a technique to visualize the relevance of each layer to the outputs of the network. By doing this, we can learn which filters in the convolution layers

correspond to each class (or feature) later in the network. This will help to make neural networks less of a "black box."

CHAPTER 5
SUMMARY AND CONCLUSIONS

5.1 The Human Vision System

The Human Vision System is capable of understanding extremely complex scenes in a very short period of time. It is able to accomplish this by separating visual information into two pathways: the dorsal pathway with takes in the entire scene in low resolution to make spatial decisions, and the ventral pathway which focuses on a small area in high resolution to recognize the objects within it. By separating these tasks, the brain is able to build a model of the scene without exhaustively processing each small region.

In contrast, state-of-the-art computer vision methods are brute force. The newest neural network methods convolve small learned filters across an entire image to search for features, but these must be trained on large labeled datasets. In order to create a more efficient system, we took inspiration from the HVS and designed a system that separates the spatial and recognition tasks.

5.2 Bio-Inspired Focus of Attention

In this chapter we proposed an unsupervised technique for finding the salient regions in images. To do this, we created a center-surround method that is able to calculate differences in an image at multiple scales to find objects that stand out from the background. This method is composed of a single convolutional filter composed of gamma kernels at multiple scales, which makes it extremely fast to compute.

In addition to testing it on standard saliency datasets, this method was tested on images that were blurred in a manner similar to what the brain receives from the eye. The brain must make the decisions about fixation locations based on this blurred data, so we want our saliency metric to work in the same environment. Interestingly, blurring the dataset improved the results for 5 out of 7 saliency methods, meaning that low resolution data may be better for making decisions about fixation locations.

5.3 Self-Supervised Feature Extraction

To continue building a full vision system, we then created a full bottom-up vision pipeline that includes the initial focus of attention and a second recognition step that is able to extract features from the salient areas. To keep this unsupervised, we used a method based on reconstruction error similar to an autoencoder. However, autoencoders do not necessarily learn robust features, so the autoencoder was extended to predict subsequent frames in a video to learn features that are more robust to small translations and rotations. To do this, we used the focus of attention to make videos from the still images that were used as inputs to the system.

Creating these videos is a form of self-supervised data augmentation. By altering the images in a way that the system knows, we can train the system with these restrictions in mind, leading to features that are much more useful than those learned through pure reconstruction error. In addition, this is a mechanism that can be further exploited through interactive environments where there are more possibilities than in a static image.

Combining the focus of attention with the self-supervised feature extraction lead to learning much more robust features than the standard unsupervised methods. In addition, this system has the added benefit of having separate feature vectors for each object in the scene, which is much more robust than using a single feature vector to describe an entire scene.

5.4 Visual Search

With the bottom-up pipeline completed, the next step was to add a visual search. The HVS is drawn to objects not only because they are salient, but also because people often desire to find specific objects in a scene. In order to do this, some mental model of the object needs to be accessed to allow the HVS to search for features deemed relevant to finding the object.

To create this system, we used the convolutional layers of the feature extractor to preprocess each input image. This gives us a set of feature maps which can be used as inputs to the Gamma Saliency. We can then learn a set of weights on the proto-saliency maps from each feature corresponding to each object. This allows us to share information between the "what" and "where" pathways in our system, much like the HVS.

A top-down saliency is able to find specific targets much faster than a purely bottom-up search. In addition, the top-down saliency correlates better with eye-tracking data from a human searching for a target better than the bottom-up search does. By exploiting these characteristics, we can make a vision system that is both more efficient and better approximates the HVS.

In addition, this framework can be used to study the representations learned by the network. By learning weights on the feature maps produced by the convolution layers, we are relating the maps to the labels at the output layer. This gives us a measure of how much each map contributes to each category of object.

5.5 Future Work

In this work we created a computer vision system that was inspired by the HVS. It divides processing between "what" and "where" pathways much like the HVS, but is still missing key components of that system. Figure 5-1 shows a more complete computer vision system. It includes multiple forms of memory as well as a Valence and Arousal mechanism to control the system.

The memory in the current version of the system is simple - we simply store the location that corresponds with each feature vector for the short term memory and have a lookup table of weights for each object. These can be vastly improved upon in a later system. Using a content-addressable memory (CAM) would expand the number of objects the system could learn, as well as group objects that have similar feature sets. In addition, the relationship between objects can be learned in scenes and also leveraged for the search mechanism.

The Valence and Arousal mechanism is a simple psychological model that explains reactions to a stimulus. Valence is a measure of how positive or negative the stimulus is, while Arousal is a measure of how strong the resulting reaction is. By placing different objects at locations in this plane, we can define a system that will react differently to different stimuli, which could then make its own "decisions" about when to search and when to free explore, as

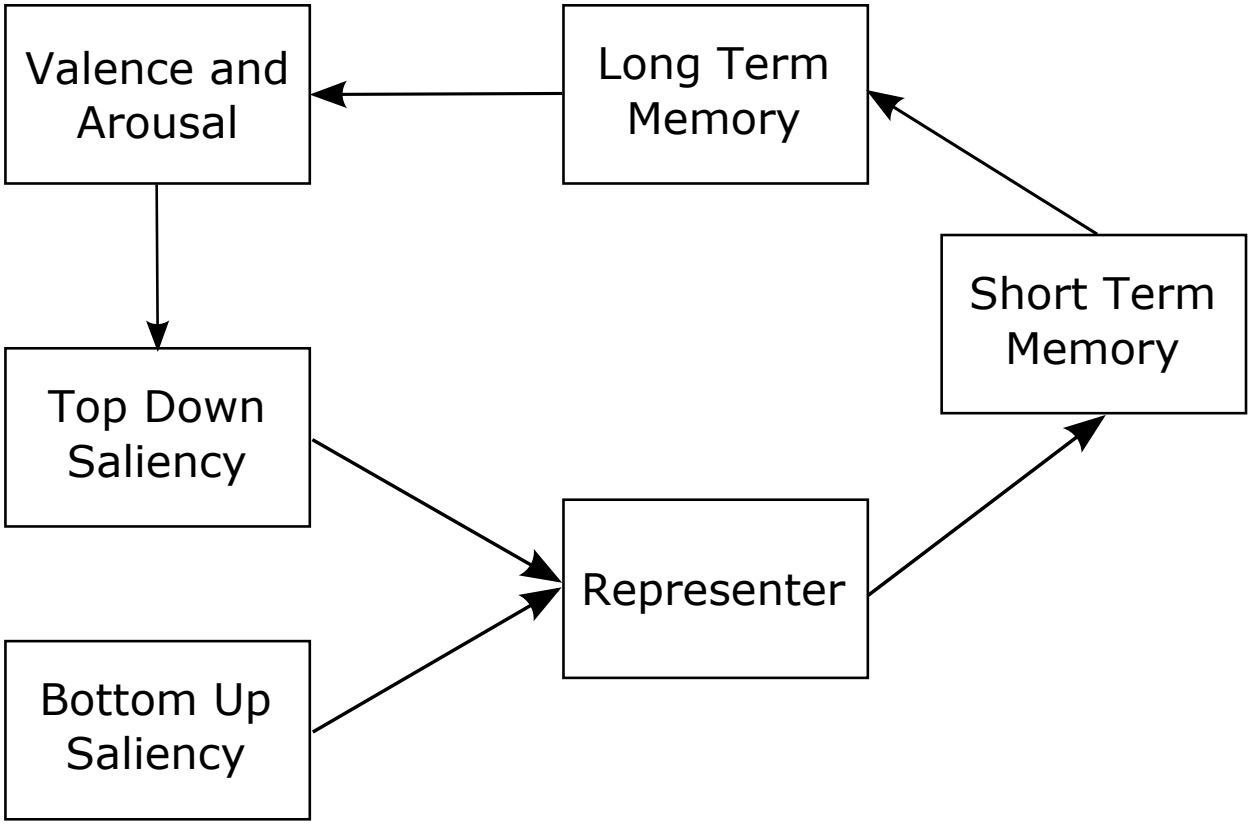


Figure 5-1. A diagram of a full vision architecture. It includes the system described in this dissertation as well as a mechanism that controls the interplay between the top-down and bottom-up searches.

well as what to look for. Adding this would be a big step towards a truly autonomous vision system.

Lastly, this system is currently a standalone vision system that works on static images. However, much of the data that humans perceive everyday comes from interacting with and changing our environments. This system provides the initial tools to interact with a dynamic environment.

REFERENCES

[1] A. Keil, T. Gruber, M. M. Müller, S. Moratti, M. Stolarova, M. M. Bradley, and P. J. Lang, "Early modulation of visual perception by emotional arousal: evidence from steady-state visual evoked brain potentials," *Cognitive, Affective, & Behavioral Neuroscience*, vol. 3, no. 3, pp. 195–206, 2003.

[2] A. M. Sillito, J. Cudeiro, and H. E. Jones, "Always returning: feedback and sensory processing in visual cortex and thalamus," *Trends in neurosciences*, vol. 29, no. 6, pp. 307–316, 2006.

[3] E. A. Buffalo, P. Fries, R. Landman, H. Liang, and R. Desimone, "A backward progression of attentional effects in the ventral stream," *Proceedings of the National Academy of Sciences*, vol. 107, no. 1, pp. 361–365, 2010.

[4] M. A. Goodale and A. D. Milner, "Separate visual pathways for perception and action," *Trends in neurosciences*, vol. 15, no. 1, pp. 20–25, 1992.

[5] J. Norman, "Two visual systems and two theories of perception: An attempt to reconcile the constructivist and ecological approaches," *Behavioral and brain sciences*, vol. 25, no. 01, pp. 73–96, 2002.

[6] A. Keil, M. M. Bradley, O. Hauk, B. Rockstroh, T. Elbert, and P. J. Lang, "Large-scale neural correlates of affective picture processing," *Psychophysiology*, vol. 39, no. 5, pp. 641–649, 2002.

[7] M. Corbetta and G. L. Shulman, "Control of goal-directed and stimulus-driven attention in the brain," *Nature reviews neuroscience*, vol. 3, no. 3, pp. 201–215, 2002.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1725–1732.

[11] J. C. Principe and R. Chalasani, "Cognitive architectures for sensory processing," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 514–525, 2014.

[12] R. Chalasani and J. C. Principe, "Context dependent encoding using convolutional dynamic networks," 2014.

[13] C. Gu, J. J. Lim, P. Arbeláez, and J. Malik, "Recognition using regions," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1030–1037.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 580–587.

[15] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[16] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[17] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[20] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2874–2883.

[21] B. Marius't Hart, J. Vockeroth, F. Schumann, K. Bartl, E. Schneider, P. Koenig, and W. Einhäuser, "Gaze allocation in natural stimuli: Comparing free exploration to head-fixed viewing conditions," *Visual Cognition*, vol. 17, no. 6-7, pp. 1132–1158, 2009.

[22] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 11, pp. 1254–1259, 1998.

[23] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural networks*, vol. 19, no. 9, pp. 1395–1407, 2006.

[24] Y. Li, Y. Zhou, L. Xu, X. Yang, and J. Yang, "Incremental sparse saliency detection," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 3093–3096.

[25] H. J. Seo and P. Milanfar, "Nonparametric bottom-up saliency detection by self-resemblance," in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*. IEEE, 2009, pp. 45–52.

[26] A. Garcia-Diaz, X. R. Fdez-Vidal, X. M. Pardo, and R. Dosil, "Saliency based on decorrelation and distinctiveness of local responses," in *Computer Analysis of Images and Patterns*. Springer, 2009, pp. 261–268.

[27] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 10, pp. 1915–1926, 2012.

[28] B. Schauerte and G. A. Fink, "Focusing computational visual attention in multi-modal human-robot interaction," in *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*. ACM, 2010, p. 6.

[29] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[30] C. Kanan, M. H. Tong, L. Zhang, and G. W. Cottrell, "Sun: Top-down saliency using natural statistics," *Visual Cognition*, vol. 17, no. 6-7, pp. 979–1003, 2009.

[31] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "Sun: A bayesian framework for saliency using natural statistics," *Journal of vision*, vol. 8, no. 7, p. 32, 2008.

[32] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on*. IEEE, 2009, pp. 1597–1604.

[33] J. Li, M. D. Levine, X. An, and H. He, "Saliency detection based on frequency and spatial domain analysis," 2011.

[34] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Advances in neural information processing systems*, 2006, pp. 545–552.

[35] N. Bruce and J. Tsotsos, "Attention based on information maximization," *Journal of Vision*, vol. 7, no. 9, pp. 950–950, 2007.

[36] M. Kümmerer, L. Theis, and M. Bethge, "Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet," *arXiv preprint arXiv:1411.1045*, 2014.

[37] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search." *Psychological review*, vol. 113, no. 4, p. 766, 2006.

[38] W. S. Geisler, J. S. Perry, and J. Najemnik, "Visual search: The role of peripheral information measured using gaze-contingent displays," *Journal of Vision*, vol. 6, no. 9, p. 1, 2006.

[39] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *Image Processing, IEEE Transactions on*, vol. 13, no. 10, pp. 1304–1318, 2004.

[40] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[41] N. Bruce and J. Tsotsos, "Saliency based on information maximization," in *Advances in neural information processing systems*, 2005, pp. 155–162.

[42] H. R. Tavakoli, E. Rahtu, and J. Heikkilä, "Fast and efficient saliency detection using sparse sampling and kernel density estimation," in *Image Analysis*. Springer, 2011, pp. 666–675.

[43] N. Riche, M. Mancas, M. Duvinage, M. Mibulumukini, B. Gosselin, and T. Dutoit, "Rare2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis," *Signal Processing: Image Communication*, vol. 28, no. 6, pp. 642–658, 2013.

[44] E. Erdem and A. Erdem, "Visual saliency estimation by nonlinearly integrating features using region covariances," *Journal of vision*, vol. 13, no. 4, p. 11, 2013.

[45] J. C. Principe, M. Kim, and J. W. Fisher III, "Target discrimination in synthetic aperture radar using artificial neural networks," *Image Processing, IEEE Transactions on*, vol. 7, no. 8, pp. 1136–1149, 1998.

[46] M. Kim, J. W. Fisher III, and J. C. Principe, "New cfar stencil for target detections in synthetic aperture radar imagery," in *Aerospace/Defense Sensing and Controls*. International Society for Optics and Photonics, 1996, pp. 432–442.

[47] A. Borji and L. Itti, "A large scale fixation dataset for boosting saliency research," arXiv:1505.03581, 2015.

[48] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba, "Mit saliency benchmark," http://saliency.mit.edu/, 2015.

[49] A. Borji, D. N. Sihite, and L. Itti, "Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study," *Image Processing, IEEE Transactions on*, vol. 22, no. 1, pp. 55–69, 2013.

[50] O. Le Meur and T. Baccino, "Methods for comparing scanpaths and saliency maps: strengths and weaknesses," *Behavior research methods*, vol. 45, no. 1, pp. 251–266, 2013.

[51] R. J. Peters, A. Iyer, L. Itti, and C. Koch, "Components of bottom-up gaze allocation in natural images," *Vision research*, vol. 45, no. 18, pp. 2397–2416, 2005.

[52] D. Litchfield and T. Donovan, "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*, 2016.

[53] S. Advani, J. Sustersic, K. Irick, and V. Narayanan, "A multi-resolution saliency framework to drive foveation," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2596–2600.

[54] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *Image Processing, IEEE Transactions on*, vol. 19, no. 1, pp. 185–198, 2010.

[55] R. Ng, "Digital light field photography," Ph.D. dissertation, stanford university, 2006.

[56] W. S. Geisler and J. S. Perry, "Real-time simulation of arbitrary visual fields," in *Proceedings of the 2002 symposium on Eye tracking research & applications*. ACM, 2002, pp. 83–87.

[57] T. Judd, F. Durand, and A. Torralba, "A benchmark of computational models of saliency to predict human fixations," 2012.

[58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[59] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[60] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2017–2025.

[61] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database of handwritten digits," 1998.

[62] R. Goroshin, J. Bruna, J. Tompson, D. Eigen, and Y. LeCun, "Unsupervised learning of spatiotemporally coherent metrics," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4086–4093.

[63] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2794–2802.

[64] P. Agrawal, J. Carreira, and J. Malik, "Learning to see by moving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 37–45.

[65] Z. Wang, S. Chang, F. Dolcos, D. Beck, D. Liu, and T. S. Huang, "Brain-inspired deep networks for image aesthetics assessment," *arXiv preprint arXiv:1601.04155*, 2016.

[66] J. Zhao, M. Mathieu, R. Goroshin, and Y. LeCun, "Stacked what-where auto-encoders," 2015.

[67] L. Bazzani, N. de Freitas, and J.-A. Ting, "Learning attentional mechanisms for simultaneous object tracking and recognition with deep networks," in *NIPS 2010 Deep Learning and Unsupervised Feature Learning Workshop*, vol. 32, 2010.

[68] R. Burt, E. Santana, J. C. Principe, N. Thigpen, and A. Keil, "Predicting visual attention using gamma kernels," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 1606–1610.

[69] D. Walther, U. Rutishauser, C. Koch, and P. Perona, "Selective visual attention enables learning and recognition of multiple objects in cluttered scenes," *Computer Vision and Image Understanding*, vol. 100, no. 1, pp. 41–63, 2005.

[70] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[71] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, 2017.

[72] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 609–616.

[73] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[74] E. Santana, M. Emigh, P. Zerges, and J. C. Principe, "Exploiting Spatio-Temporal Structure with Recurrent Winner-Take-All Networks," *ArXiv e-prints*, Oct. 2016.

[75] R. Burt and J. C. Principe, "Multi-object image classification using attention based, self-organized deep learning networks," Submitted.

[76] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[77] E. Kowler and E. Blaser, "The accuracy and precision of saccades to small and large targets," *Vision research*, vol. 35, no. 12, pp. 1741–1754, 1995.

[78] A. Makhzani and B. J. Frey, "Winner-take-all autoencoders," in *Advances in Neural Information Processing Systems*, 2015, pp. 2791–2799.

[79] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[80] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[81] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[82] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," *arXiv preprint arXiv:1312.6082*, 2013.

[83] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?" *arXiv preprint arXiv:1604.03605*, 2016.

[84] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[85] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[86] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1–9.

[87] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *International Conference on Computer Vision (ICCV)*, 2015.

[88] M. Cudic, R. Burt, E. Santana, and J. Principe, "A flexible testing environment for visual question and answering with performance evaluation," *IEEE Transactions on Computational Intelligence and AI in Games (under review)*, 2016.

[89] S. Frintrop, G. Backer, and E. Rome, "Goal-directed search with a top-down modulated computational attention system," in *Pattern Recognition*. Springer, 2005, pp. 117–124.

[90] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *University of Montreal*, vol. 1341, p. 3, 2009.

[91] R. Burt and J. C. Principe, "Top-down gamma saliency - learning to search for objects in complex scenes," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Submitted.

[92] M. Cudic, "Mnistvqa," https://github.com/mihaelcudic/mnistvqa, 2016.

[93] Y. LeCun, C. Cortes, and C. J. Burges, "Mnist handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, vol. 2, 2010.

[94] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[95] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[96] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[97] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *arXiv preprint arXiv:1703.06870*, 2017.

BIOGRAPHICAL SKETCH

Ryan was born in Joliet, Illinois in 1988. He spent his early years in Michigan before attending Kettering University, where he graduated with Bachelor of Science degree in electrical engineering in 2011 with minors in acoustics and business. During his undergraduate studies, he had many co-operative education jobs, including with Autoliv Electronics, Patrick Energy Services, and P3 North America, where he also worked for a year after graduating.

Ryan was admitted to the University of Florida in 2012 and was a teaching assistant for an undergraduate signal processing course. He then joined the Computational NeuroEngineering Laboratory in 2013 and began his research there. He earned a Master of Science degree in 2014 and a Doctor of Philosophy in 2017.