Attention, Salience, and Convolutional Neural Networks

SYDE 552/750 Final Project

Peter Duggins

April 6, 2016

1 Introduction

The primate visual system is responsible for processing an immense amount of information about the external world. Images which enter the brain through the retinas not only have high resolution and cover a large visual field; they are also dynamic, contain multiple color channels, and have intensities which vary over many orders of magnitude. Without a significantly larger visual system, it is impossible to process this information in full detail. One standard method the brain uses to combat this is compression through feature extraction, a mechanism which has been explored at length in deep neural networks (DNNs), particularly convolutional networks (CNNs). The other method utilizes a kind of selective processing for salient parts of the image, deemphasizing data which are deemed less relevant for the systems current goals and instead focusing on features of interest. This salience-based focus is commonly referred to as attention.

Attention requires the visual system to convey signals about the salience of particular features, or parts of the image, backwards through the visual hierarchy in order to affect how the system processes future inputs. There is abundant evidence that high-level cognitive areas in biological brains generate and feedback such salience signals. The frontoparietal attention network projects widely to visual areas Padmala and Pessoa (2008); through its strong connections to, and activation alongside, evaluative sites such as the amygdala, hypothalamus, and cingulate cortex, it is hypothesized that emotional salience information calculated in these areas is fed back to the visual system Pessoa (2008). Similarly, the pulvinar is thought to receive input

from biological areas that represent salience information organized in map-like structures, then send control signals which selectively route information in the visual hierarchy Olshausen et al. (1993). Though the exact mechanisms and areas involved in computing and routing salience information are still unclear, it is generally accepted that top-down signals from cortical areas drive visual attention.

Combining bottom-up visual processing, as implemented in DNNs, and top-down attentional, as realized through feedback signals, is a difficult but potentially highly-rewarding pursuit. Most DNNs are designed as feedforward classifiers which utilize backpropagation for training. Incorporating top-down connections in these networks creates loop, requiring specialized cells, such as recurrent neural networks (RNNs), to backpropagate error, but these networks are limited when it comes to simulating dynamical systems with feedback loops. Attention is often cited as the next step for improving DNN performance, and an increasing number of researchers are building models that incorporate mechanisms to focus on salient regions of the input.

My objectives for this project are the following. First, compute the salience of features present in visual images using bottom-up classification, external input from the user, and top-down projections from higher-order features. Second, feedback salience signals through the visual hierarchy, such that they reactivate the corresponding features, enhancing the presence of that feature relative to other less-relevant features. Third, observe the effects of this feedback on feedforward image classification. This should be assessed through the statistics of feature map activation (across classes and layers), the classification of ambiguous images (which contain multiple features on which the network was trained), and visualization of the image as seen by the network (highlighting the attended features). Fourth, and most generally, I hope to build a flexible and biologically plausible model which is capable of implementing different CNN structures, salience feedback mechanisms, and selective tuning.

2 Background

The use of salience information to focus on particular areas of an image was popularized by Koch and Ullman (Koch and Ullman, 1987). The authors used what they referred to as salience maps, spatially organized layers which represent the salience of each location in the image, to focus on particularly important areas. The

concept of salience maps and feedback was expanded by Tsotsoss Selective Tuning model Tsotsos et al. (1995), in which features in a visual hierarchy compete with each other for dominance, one is selected by a winner-take-all procedure and said to become salient, then information is fed back through the network to activate neurons associated with that feature. The selection of these neurons is achieved through a selective tuning mechanism, in which the activated neurons engage an attentional suppressive annulus, inhibiting the activity of nearby neurons that are not associated with the attended location or feature, as shown in figure 1. Tsotsos and colleagues have shown that this mechanism can indeed highlight areas of the image associated with a particular class, and that psychophysical evidence supports the biological plausibility of some of these computations Rothenstein and Tsotsos (2008); Hopf et al. (2006). This model has been adopted and modified by other researchers in biological and computer vision to find the features associated with salient parts of the image. A typical approach is the one taken by Olshaussen Olshausen et al. (1993), in which the original image is blurred, the brightest region is selected, its boundaries are routed to the top level of the classification hierarchy, the features associated with that region are perceived then inhibited, and the process is repeated for each object of attention in the image.

Implementing these mechanisms in DNNs has proved more difficult, and it is only in the last few years that backpropagating salience has been integrated into visual classification systems. Recent models have sent salience information back through context vectors into specialized long short term memory (LSTM) cells. The result is that certain areas of the image are emphasized in future processing, which some researchers have realized through a fovea-like area of high-resolution perception Mnih et al. (2014). Perhaps the most successful application of salience-based attention in DNNs was by Xu et al Xu et al. (2015), who used a CNN in combination with LSTMs to highlight areas in the image corresponding to each word in a caption the network generated for that image (figure 2). Although their results are quite impressive, it is not obvious that they are biologically plausible. In particular, such networks often (a) assign salience directly to areas of the image itself, rather than the features identified in the visual hierarchy; (b) calculate winner-take-all salience and enforce selective tuning using mechanisms that cannot be implemented in neurons; and (c) send salience signals back down the visual hierarchy using connections and transformations not possible in brains.

Walther and Koch Walther and Koch (2006) propose a hybrid model in which specific circuits of excitatory and inhibitory neurons implement WTA and selective tuning using biologically-plausible local connectivity.

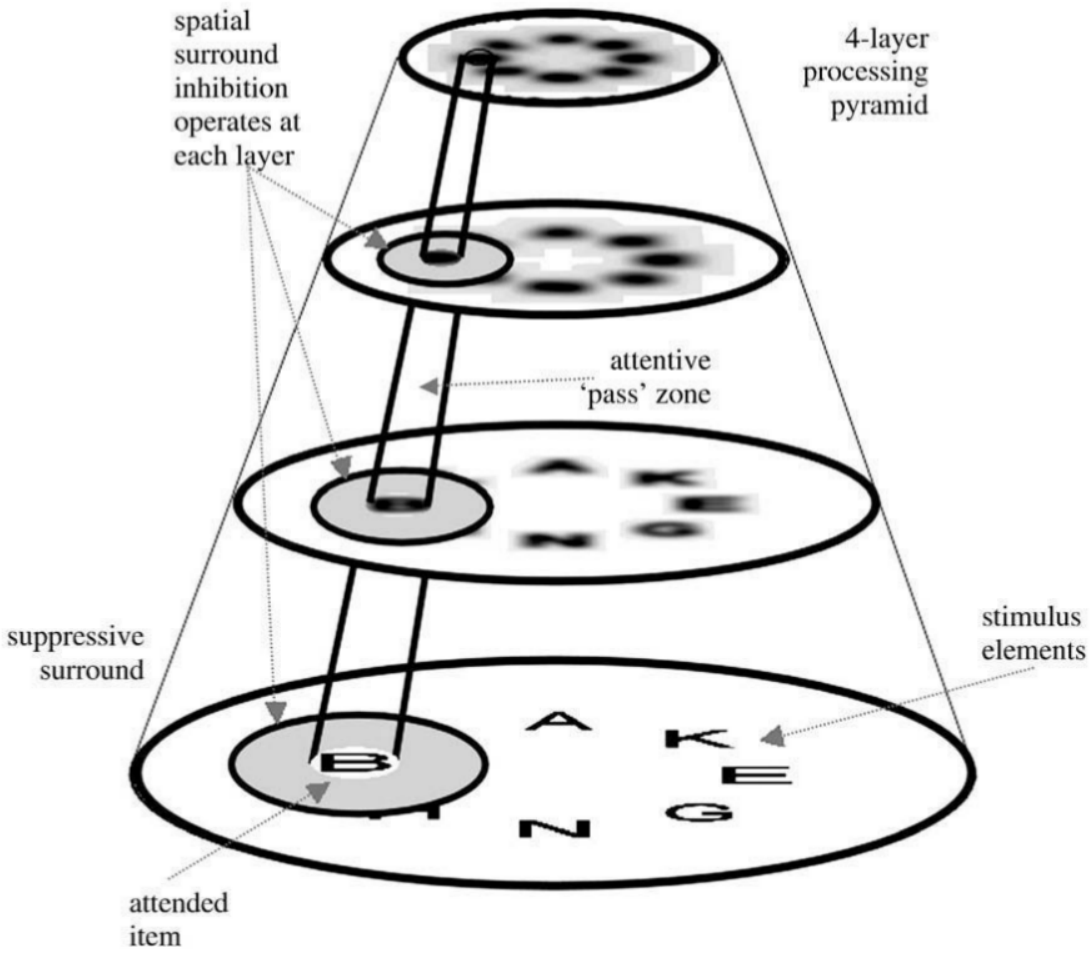

Figure 1: Diagram describing the Selective Tuning Model Rothenstein and Tsotsos (2008). An image is presented to the bottom of the network and feeds forward through the visual hierarchy. At the top, the feature is identified, and feedback signals propagate back down the hierarchy. At each layer, units activated by backpropagation create spatial surround inhibition, maintaining high activity while suppressing neighboring units that do not encode the feature. At the bottom layer, the attended item appears against a suppressive surround in the image.

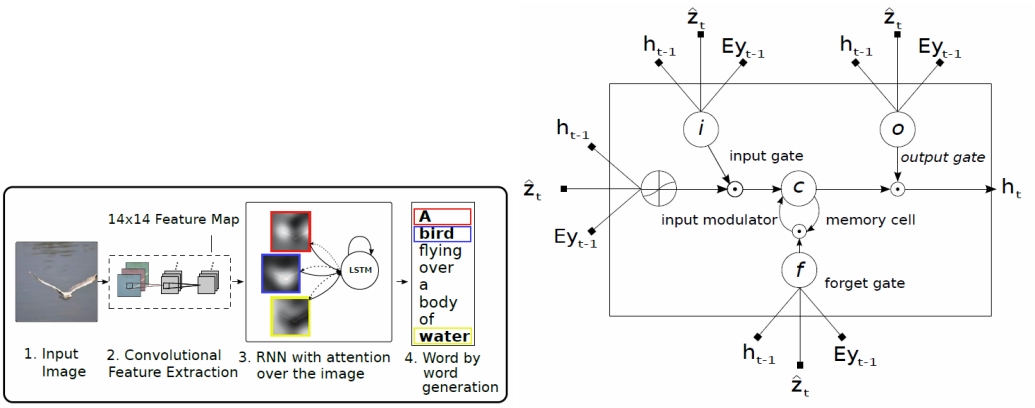

Figure 2: Sketch of the Xu et al Xu et al. (2015) CNN, which utilize LSTMs to backpropagate a type of attentional signal. Details are interesting but irrelevant to the current model.

These types of circuits have been identified in the visual system, and are thought to produce the center-surround inhibition necessary for contrast sensitivity and selective tuning Dragoi and Sur (2000). The image is fed into the classifier which identifies low level features, whose spatial activations are sent to "conspicuity maps. The maps from different channels (color, intensity, orientation) compete using the neural WTA in the salience layer, which feeds back the most salient location to each of the feature maps. The feature with the greatest activation in that location is activated using neural selective tuning, and the resulting map is used to mask the network output, producing an area of attentional focus. Figure 3 shows a diagram of this network. The main shortcoming of this model is that the feature maps are hard-coded to select the features chosen by the authors, and only a single layer of feature extraction occurs, prohibiting advanced classification and attention to high-level features.

3 Model Description

I expand the network of Walther and Koch to include a learned CNN, a salience map associated with each layer, and local constraints on the transmission of information from salience maps (SMs) to feature maps (FMs). Though the model is designed to work with an arbitrarily structured CNN, I found the following hyperparameters gave excellent performance on both my custom dataset LINE (described in Results) and the MNIST dataset:

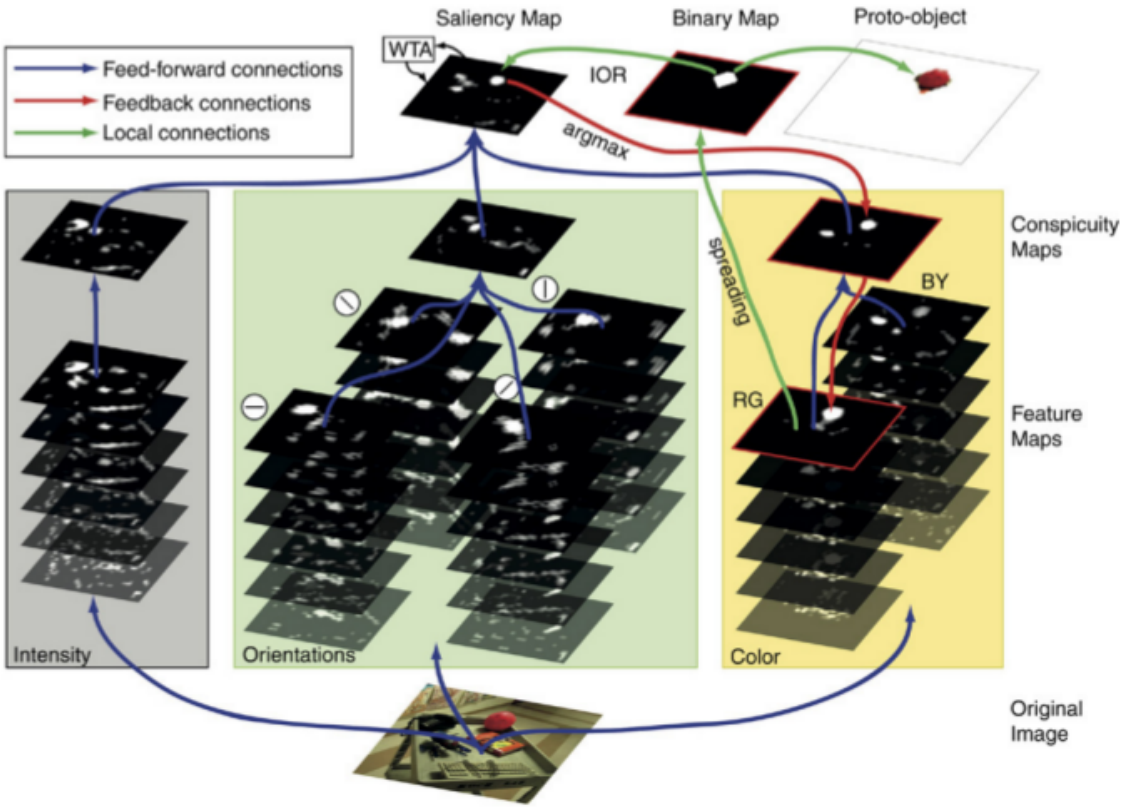

Figure 3: Diagram describing the Attention to Proto-Objects model designed by Walther and Koch Walther and Koch (2006). See text for details.

- 3 convolutional layers
- 8 kernels of size 7x7, 5x5, and 3x3
- 2 dense layers of size 128 and $n_{classes} = 3(LINE), 10(MNIST)$
- dropout with probability 0.5 after each layer
- relu activation in all layers (except the last dense layer with softmax activation)

The SM at each convolutional layer has a number of units equal to the number of features in that layer (8). First, each of these salience units computes the sum (or average) activity over all units in its corresponding FM. This calculation gives a rough estimate of the bottom-up salience of that feature by indicating the total activity in the FM, but (importantly) removes all spatial information. For example, if the input image contained a tall rectangle, and the FMs encoded information about horizontal (H), vertical (V), and diagonal (D) lines, then the salience unit corresponding to the V FM would have the highest (scalar) value, the unit for H would have an intermediate value, and the unit for D would be zero.

Second, the SM receives arbitrary external stimulus that differentially activates each unit. These signals originate in other parts of the brain, such as the frontoparietal attention network or pulvinar, and represent the salience of the chosen features given their emotional or cognitive relevance. For instance, when tasked with identifying all the H lines in an image filled with lines at every orientation, an external input would increase the salience of units associated with kernels that picked out Hs in feedforward classification. The manner in which these stimuli are routed to these features is not addressed in the model, except as described below.

Third, the SM receives feedback input from higher-level SMs through learned connections. This feedback causes highly activated features to spread activation backwards to the features which compose them. This mechanism should realize a top-down cascade of attention that identifies and selectively activates those units in the SM which indicate the presence of an attended high-level feature. For example, if the input image was a mixture of geometric shapes, and rectangles became salient at later stages in visual processing, then salience would feed down to lower levels, increasing the salience of straight, but not curved, lines. This mechanism is something like the backpropagating context vector in Xu et al.s LSTMs in the context of Walther and Kochs SM/FM hierarchy.

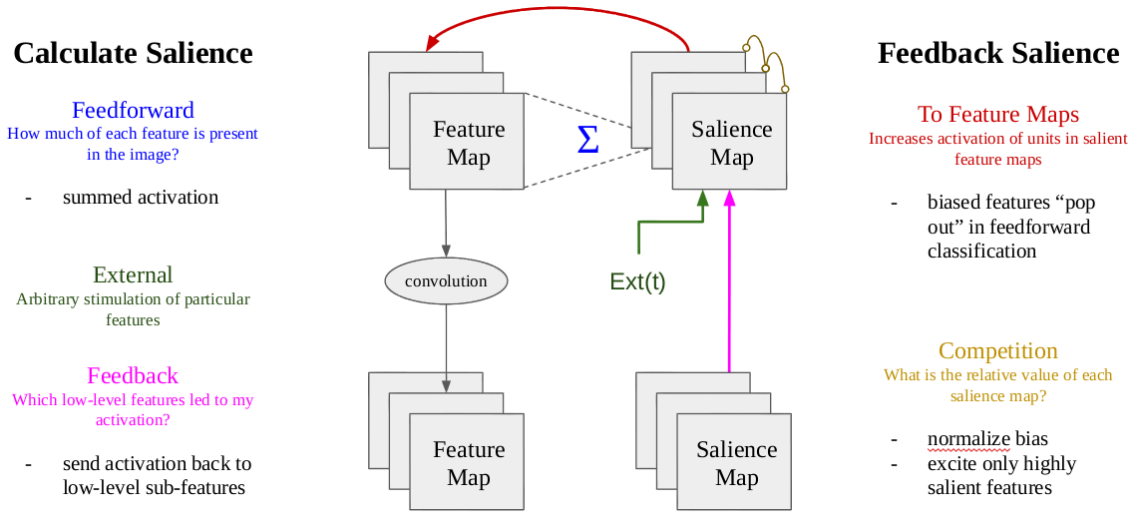

Figure 4: Model architecture as implemented in Nengo, using convolutional filters learned in Keras.

Having computed salience from bottom-up, top-down, and external sources, the features now compete for prominence. Rather than implement the WTA competition described in Walther and Koch, I begin by assuming the competition results in a softmax normalizations of saliences. I then feedback the activity of the SMs to the corresponding FMs. There are two ways I implement this feedback. I began with the simplest option: feedback the salience value as a constant bias term to each unit in the FM. This undoubtedly activates this FM more strongly, but has significant downsides discussed below. I extend this feedback by implementing center-surround inhibition within each FM, such that units that are already active will receive a boost in activation, but inhibit neighboring units to keep them inactive: `x[i][j] += -1*np.average(x[i-rad:i+rad+1][:,j-rad:j+rad+1) + 2*x[i][j]`.

As discussed in the introduction, CNNs are ideally suited to visual feature extraction and learned image classification, but are not equipped to deal with feedback (FB) signals due to the linear nature of backpropagation. RNNs provide a possible solution by unfolding time for a certain number of steps, turning the loop created between FMs and SMs into a long temporal chain. However, RNNs would be difficult or impossible to implement in this network, because the state of the SMs depends not only on their previous state, but also on the state of higher-level features. Thus, the W matrix would have to account for the implementation of $FB_{SM \to FM}$ (selective tuning), $FB_{SM \to SM}$ (learned salience backpropagation), and

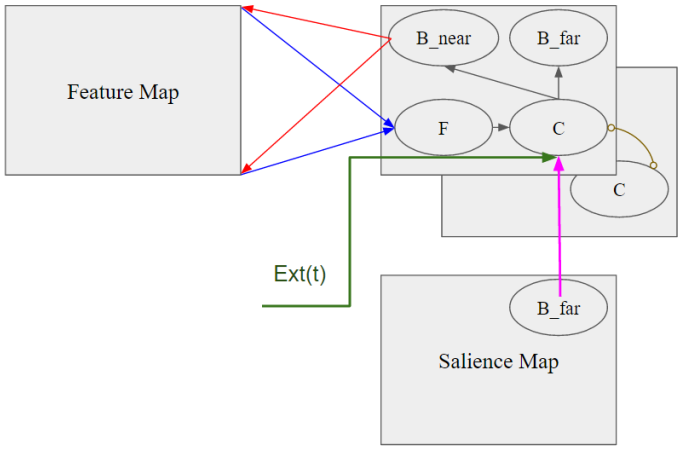

Figure 5: Salience Map architecture as implemented in Nengo. Blue arrows indicate summation over all dimensions of the FM; red arrows indicate projection back into the dimension of the FM: other arrows indicate addition or scaling of one-dimensional vectors. The pink arrow is a learned connection.

$FF_{FM \to FM}$ (learned convolutional kernels).

A better approach is to implement the circuit in software which was build to accommodate arbitrary dynamics and connections, such as nengo. To incorporate the strengths of both the CNN and nengo approach, I trained the CNN using stochastic gradient descent in Keras (details in documentation in `keras_CNN_v5.py`), imported the learned filters (kernels) and model architecture into nengo, rebuilt the feedforward structure, added the SMs, and added the feedback connections.[1] Figure 4 summarizes the model architecture, including FMs, SMs, feedforward visual classification, SM competition, and feedback salience; a closeup of the SM is shown in figure5. Any Keras structure composed of standard CNN layers (convolution, average/max pooling, dropout, and dense in any order with appropriate parameters) can be converted to nengo using this code, allowing for rapid architecture prototyping and experimentation. [2] I am currently using nodes rather than neuron ensembles for the simulation, to simplify the coding/analysis and reduce the runtime.

The simulation proceeds as follows. The user specifies the dataset, type and magnitude of feedback, SM competition, and external stimulation for the network in `main()`. An image is fed into the first convolutional layer using a stimulus node for *pt* (presentation time) timesteps. Activation spreads forwards and backwards, and the state of the output node is dynamically tracked. Once the simulation is over, I compute the statistics

[1] The flattening and dropout layers not depicted for clarity, since they do not perform computationally relevant transformations
[2] Pooling and strides have a bug which leads to poor classification in nengo; see discussion

of FM and SM activation, and the classification produced by the output, after the first time step and at the end of the presentation time. These two values reflect what the standard feedforward network would produce (salience has not yet had time to propagate due to nonzero synaptic time constants), and what effect the feedback had on feature activation and image classification.

4 Results

I began by validating that the conversion from Keras to Nengo preserved the network structure and feedforward classification. To do so, I create the LINE dataset, a collection of 1x64x64 images that contain H, V, and D lines that have been randomly rotated, blurred, and gaussian noised (see documentation in `create_lines.py`). I found that the the nengo classifier performed comparably to Keras on the LINE dataset: Keras error was 0.005, while Nengo error was 0.01 for the test data. Interestingly, the nengo network performs slightly worse on the MNIST dataset than Keras does, but this difference depends on the hyperparameters chosen, indicating the presence of a small bug. Although the trends reported below hold for the MNIST data (with more noise), I report results on the LINE dataset in the rest of the report.

Next, I confirmed that the SMs had different activations when the network was presented with images of different classes, and that this variance was significantly greater than the within-class variation. [3] Figure 6 shows the average activation of the eight feature maps in layer 0 and 1 of the network when presented with H and D lines, as computed in the SM at each layer. Some features are highly active for both H and D lines, such as 1/5 in layer 0 and 2/6 in layer 1, though their magnitudes are significantly different. Similarly, some features are not active for either line, such as 4/5 in layer 1. Finally, some features activate strongly only in the presence of either H or D, such as 0/4 in layer 0 for H and (coincidentally) 0/4 in layer 1 for D. This provides preliminary evidence of differential FM/SM activation that can be utilized as for attentional control through selective tuning. It also supports the hypothesis that individual features do not solely indicate the presence of H or D lines, but that the weighted combination of those features strongly indicates their presence or absence, as we would expect from a DNN.

[3] I calculated these statistics initially in Keras before building the nengo model, to confirm the differential existed before constructing a model which utilized this different to implement attention. I visualize the statistics in the nengo code. The code for calculating statistics in can be found near the end of the Keras file, and includes more information on the spatial differences within FMs than reported here

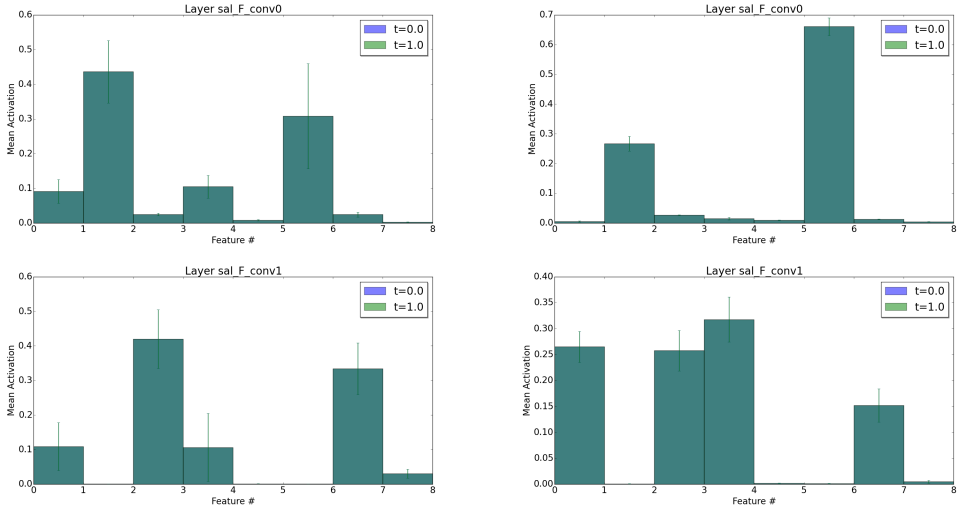

Figure 6: Salience map activations in layers 0 (top) and 1 (bottom) for images containing horizontal (left) and vertical (right) lines, without feedback. Error bars represent standard deviation over 200 images in the test set, while the legend at the top indicates that the presentation time of the input image is only one time-step of the simulation - an insufficient time for feedback to occur, but long enough for classification.

To show that the feedback signals from SM to FM have an effect on image classification, I presented the network with an H for $pt = 5$ timesteps and allowed the SMs to compete and feedback to FM. I chose the simplest version of competition (softmax) and feedback (constant bias) for this initial test, and implement feedback only in layer 0, with a feedback constant of $k_{FB-near} = 10$. Figure 7 shows the SM activation in layer 0 and 1, as well as the output layer activation, at the beginning and end of the simulation. As expected, feedback in layer 0 increased the activation of the feature which is identified most strongly in the image (bottom-up salience) while proportionally decreasing the others. These new values propagated forward through the network, changing the salience of layers 1 and 2, but in a more differentiated manner (some features increase in activity, some decrease, others stay constant). Initially, the network correctly classifies the input as H (class 0), but after $pt = 10$, the feedback signals causes the network to classify the image as V (class 1). This further supports the hypothesis that the weighted combination of features are necessary to correctly identify the output: if only feature 5 in layer 0 encoded H, then feedback which increased 5 relative to all other features would lead to a stronger selection of H in the output.

A change in output can also be induced by externally stimulating certain features in layer 1 (those

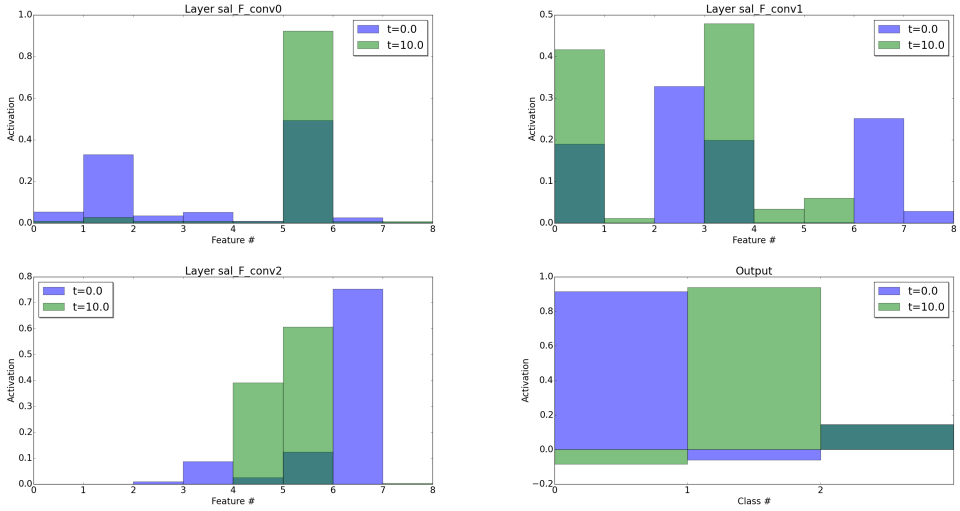

Figure 7: Salience map activations before and after the application of external stimulus and feedback from salience maps in higher layers. Presented input was a horizontal line (class 0), but after feedback, the network classifies the image as a vertical line (class 1).

associated with H but not D) [4] allowing those saliences to backpropagate to the layer 0 SM, then allowing $feedback_{SM \rightarrow FM}$ in layer 0. This causes features in layer 0 associated with H to become more active (feature 3) and features associated with D to become less active (feature 5). This top-down salience interacts with the bottom-up salience from the image itself to produce new activation in each layer of the network, as shown in Figure 8. Although the top-down attention to H features is not sufficiently strong to overwrite the clear bottom-up presence of D in the image, it does shift the output significantly towards classifying the image as H. Note that SM $\rightarrow$ FM feedback in layer 1 was not allowed, so this result relies on the SM $\rightarrow$ SM feedback that is one of the unique features of this model.

To better understand the effect feedback has on FMs in the CNN, I investigate the kernels associated with features that are excited or inhibited, as well as the image as perceived by specific layers of the network, which I approximate by summing the (salience weighted) feature maps in that layer. The top row of Figure 9 shows the kernel of feature 5 as well as the spatial activation of FM 5 before and after the $pt = 5$ application of feedback. Firstly, the kernel itself seems totally unrelated to either H or D: the only useful filtering it

[4] External stimuli: Feature 3 in layer 1 = -1, feature 3 in layer 1 = 1. Feedback: $FB_{SM \rightarrow FM}$ in layer 0 = constant, $k_{FB} = 5$; $FB_{SM \rightarrow FM}$ in layer 1 = none; $FB_{SM \rightarrow SM}$ = inverse of the dense matrix, $k_{FB} = 20$

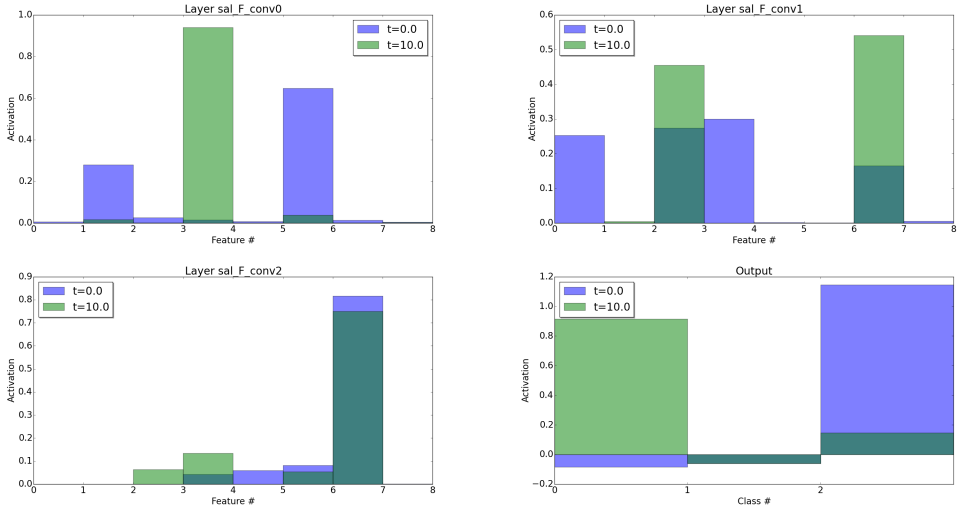

Figure 8: Salience map activations before and after the application of external stimulus and feedback from salience maps in higher layers. Presented input was a diagonal line (class 2), but after feedback, the network *almost* classifies the image as a horizontal line (class 0).

appears to do is take the negative of the original image. When feedback from unit 5 of the SM reaches the FM, it sends a constant positive bias to each unit. The result is a uniformly high level of activation that washes out the image entirely. The effect of this oversaturation on the transformed image is shown in the bottom row: before feedback, the summed activation of FMs helped reduce background noise and sharpen H; after feedback the line is still present, but harder to distinguish. It is probable that whichever class looks most like a white square to the network is the one that will be chosen as the label for this image.

This oversaturation is a direct effect of the type of $feedback_{SM \to FM}$ chosen: because all spatial information is lost when calculating salience, trying to increase the presence of that feature by feeding back a constant value will not work. Another mechanism is needed to transform a constant bias signal into spatially-relevant changes in FM activation. Drawing from the selective tuning model, I implemented a simple form of center-surround inhibition within the FMs: each spatial unit excites itself by positively multiplying its activation, then subtracts from its activation the average activation of neighboring units in radius r (see documentation in `keras_nengo_layers.py` and the toy script `center_surround.py`). As shown in Figure 10, this center-surround inhibition helps reduce washout, but does not selectively tune those units in the FM whose activity represents the presence of that feature. Incorporating this simple center-surround is ineffective in solving the

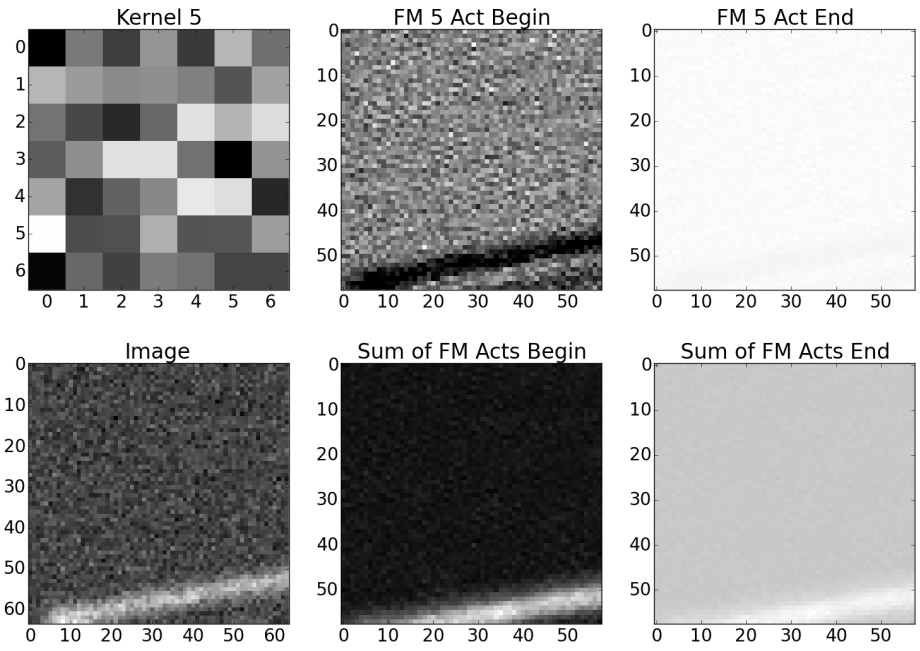

Figure 9: Visualizations of the kernel, feature map, and layer activation for layer 0 when presented with a horizontal line. Note the high degree of oversaturation at the end of the presentation time (right column).

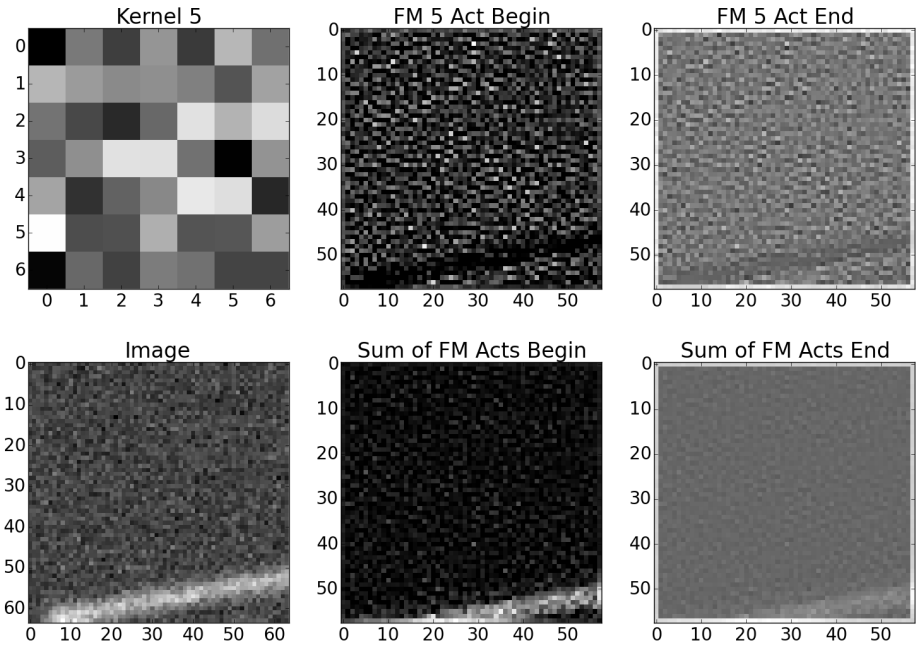

Figure 10: Visualizations of the kernel, feature map, and layer activation for layer 0 when presented with a horizontal line, but with center-surround dynamic inhibition present in the feature maps. The oversaturation problem is lessened, but the areas corresponding to the feature are not selectively tuned.

selective-tuning problem, though it does not decrease classification accuracy on its own.

In one last attempt to find the ghost of attention lurking in the network, I presented the network with an image containing multiple both H and D, externally activated features associated with the D class, permitted feedback, and observed the classification made by the network. Without feedback, the network classified 70% of the images as H and 30% of the images as D. I used figure 6 to identify the features associated with D but not H, then externally stimulated various combinations of them in each SM. Despite the strength and specificity of the external feedback[5], I was unable to force the network to classify as D any images that were originally labeled as H. The visualizations did not highlight areas the image composing D. The result was the same for the MNIST data, and did not improve with alterations to the CNN architecture. Sad Peter.

[5]I also tried tweaking of the feedback constant and mechanisms and retraining the Keras network on different line data

5 Discussion and Conclusion

There exists strong biological evidence that salience information from higher-order cognitive systems feeds back to the visual system and implements a selective tuning of neurons. However, translating what we know about anatomical connectivity, visual microcircuits, and neurotransmitter action into a computational framework requires many simplifying assumptions about how salience is represented and how it is backpropagated to the visual system. Modelling visual systems using such feedback is a difficult problem that has been approached in many different ways over the last twenty years. Though several DNNs have used WTA algorithms and masking to create images that are filtered by the presence of specific features, it is not clear the methods they use can be implemented in biological brains, or than they can be generalized to spreading activation between feature maps. In this project, I have incorporated several framework into a dynamic, recurrent CNN that classifies images of horizontal, vertical, and diagonal lines.

I first showed that the feature maps in the convolutional network were differentially activated for different classes of inputs (both for the LINE and MNIST datasets), a necessary precondition for using these activities to calculate salience. I then showed that calculating salience from, and feedback to, the feature maps produced new patterns of activation that caused the network to classify input images differently. Upon closer inspection, however, these mechanisms did not perform the selective tuning achieved by other models. I was unsuccessful in my attempts to fine-tuning parameters or implement center-surround inhibition to achieve any sort of multiplicative gain within the feature maps. I was also unable to attend to specific features in a manner which led to changes in classification for mixed-class images (images containing horizontal and diagonal lines). Visualization of the feature map activations confirmed my suspicion that the external activation of particular features at different layers, when combined with bottom-up and top-down salience, did not highlight those areas of the image corresponding to the lines.

There are several clear reasons why this project failed to receive positive results. Firstly, the model framework I proposed for salience feedback was not firmly based on any single paper discussed in the Background section. It was an attempt to incorporate certain strengths from each: selective tuning from Tsotsos; visual feature extraction using convolutional neural networks; and biologically-plausible algorithms and circuits from Walther and Koch. The result was lacking specific detailed mechanisms mentioned in

each model: these features may have been essential in realizing their positive results. For example, I was unsure whether implementing a constant feedback from the salience to the feature maps, in combination with center-surround inhibition, would enhance the spatial activation of units in that feature map encoding the current feature. I had hoped that, in combination with the dynamic input of the image to the feature maps (bottom-up), this could produce a multiplicative-like selective tuning effect as time increased. Unfortunately, all the evidence I collected points against this being so easy.

To address a similar problem, Walther and Koch implemented a specific circuit (Figure 11 within their feature maps, which allowed backpropagating information to spread positively to those spatial units associated with the object, but inhibit all other units. Given more time, I would have implemented this circuit within my FMs to address this problem. Another biologically-motivated solution would be to use biophysical neuron models which could be controlled by neuromodulators, such that their activity (voltage, spike rate) did not change except in response to inputs from lower layers. The neurotransmitter Acetylcholine is known to increase the sensitivity of visual neurons to future inputs Thiele (2013); Picciotto et al. (2012), and to originate from areas associated with attentional control (amygdala, anterior cingulate cortex, etc.) Phelps and LeDoux (2005); Menon and Uddin (2010). However, such a biophysically detailed mechanism was beyond the scope of this project.

On the other hand, there may be more fundamental problems with the architecture, especially in regards to the flow of salience signals. This feature differed wildly between the models discussed above, so I did not hold my implementation strictly to any one of them. One departure is that I trained a feedforward connection between the salience maps in Keras, then inverted that matrix in Nengo to feedback activation between the SMs. This is a strong assumption[6], but seems to me a necessity for transmitting salience signals down the visual hierarchy dynamically. I do find it significantly more plausible than inverting an identical kernel at every location in the feature maps, or using highly specialised LSTM cells, and I believe the biological evidence supports the claim that salience information is stored and transmitted separately from the features themselves.

Another reason the model performed poorly was unresolved bugs. It is difficult to describe how frustrating

[6]Inverting matrices is implausible in general, but the brain is unconstrained to the backpropagation learning rule that inherently limits DNNs: it can probably learn the connection weights between salience maps "on the fly in the top-down direction as the maps function dynamically in the visual hierarchy and cortical.

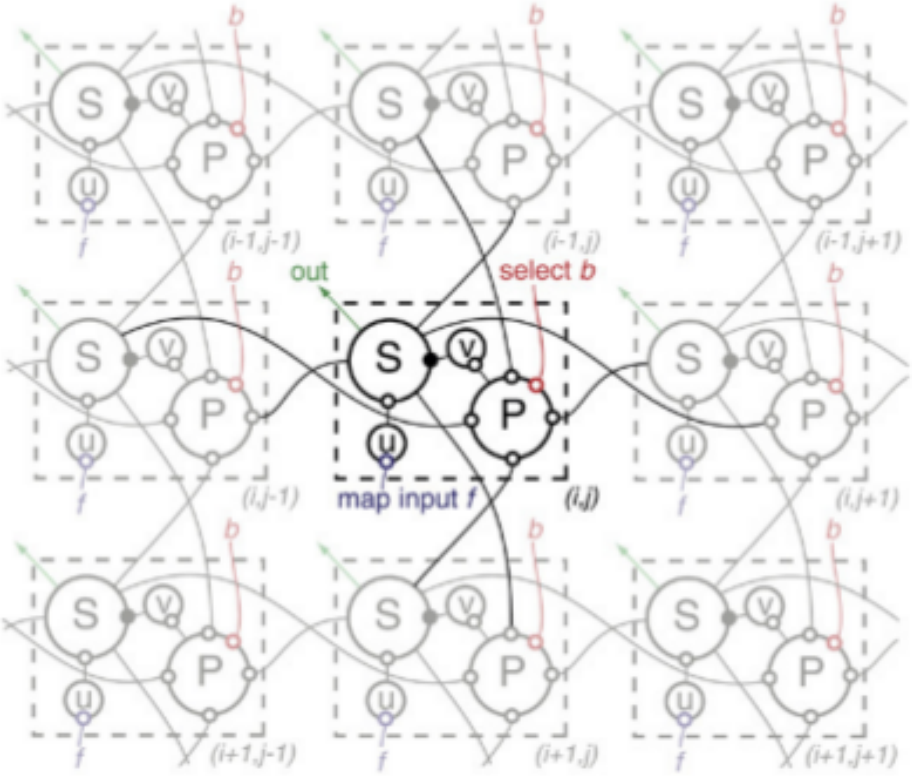

Figure 11: Walther and Koch's Linear Threshold Unit implementation of a segmentation operation within their feature maps. Backpropagation b inactivates a pooling unit P, causing disinhibition of the spreading unit S which also receives feedforward input f. S activates disinhibits nearby S units, but only as far as the object being segmented extends.

these are, as well as the fact that they still exist despite the staggering among of time in the last two weeks spent fixing them. One especially important bug arises from how Keras performs rounding on odd-dimensional feature maps, producing weight matrices of a different shape than I expected given normal implementations of pooling layers (including Eric Hunsburgers cuda/nengo code). This bug keeps me from effectively utilizing max and average pooling in the rebuilt nengo architecture. Ignoring pooling in the Keras training probably produces filters that are not translation-invariant, and therefore have less distinctive kernels even for simple features like lines.

The final reason the project failed was inefficient use of time. The majority of my hours spent on the project ended up being devoted to interfacing Keras and Nengo, and getting Nengo working properly. I recognized that doing so would involve a far greater effort that building some simple additions into Keras, but I believed that a Nengo implementation would ultimately be more robust and flexible. To this end, I did build a framework which allows for the versatile manipulation and testing of features related to feedforward visual feature extraction and feedback salience signals. Implementing the model in nengo allowed the exploration of multiple dynamically active feedback mechanisms, and will hopefully permit the inclusion of multiplicative or modulatory salience modulation in future work. It also provides the means for substituting more biological details into the network itself, especially the use of spiking neurons and the NEF. However, due to my unfamiliarity with Nengo and the intricacies of model dynamics, it took me significant effort to add in new features and hunt down bugs, leaving less time for exploration of the nuanced feature (as mentioned in the background) that might be necessary to turn the attention signal from a classification-disruptor into a classification-enhancer. Nonetheless, by building this model, struggling to resolve bugs, and revisiting the way information flows through the system, I have learned a great deal about manipulating convolutional networks and the difficulties inherent in the implementation of visual attention.

References

Dragoi, V. and Sur, M. (2000). Dynamic properties of recurrent inhibition in primary visual cortex: contrast and orientation dependence of contextual effects. *Journal of Neurophysiology*, 83(2):1019–1030.

Hopf, J.-M., Boehler, C., Luck, S., Tsotsos, J., Heinze, H.-J., and Schoenfeld, M. (2006). Direct neurophysio-

logical evidence for spatial suppression surrounding the focus of attention in vision. *Proceedings of the National Academy of Sciences of the United States of America*, 103(4):1053–1058.

Koch, C. and Ullman, S. (1987). Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer.

Menon, V. and Uddin, L. Q. (2010). Saliency, switching, attention and control: a network model of insula function. *Brain Structure and Function*, 214(5-6):655–667.

Mnih, V., Heess, N., Graves, A., et al. (2014). Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, pages 2204–2212.

Olshausen, B. A., Anderson, C. H., and Van Essen, D. C. (1993). A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *The Journal of Neuroscience*, 13(11):4700–4719.

Padmala, S. and Pessoa, L. (2008). Affective learning enhances visual detection and responses in primary visual cortex. *The Journal of Neuroscience*, 28(24):6202–6210.

Pessoa, L. (2008). On the relationship between emotion and cognition. *Nature reviews neuroscience*, 9(2):148–158.

Phelps, E. A. and LeDoux, J. E. (2005). Contributions of the amygdala to emotion processing: from animal models to human behavior. *Neuron*, 48(2):175–187.

Picciotto, M. R., Higley, M. J., and Mineur, Y. S. (2012). Acetylcholine as a neuromodulator: cholinergic signaling shapes nervous system function and behavior. *Neuron*, 76(1):116–129.

Rothenstein, A. L. and Tsotsos, J. K. (2008). Attention links sensing to recognition. *Image and Vision Computing*, 26(1):114–126.

Thiele, A. (2013). Muscarinic signaling in the brain. *Annual review of neuroscience*, 36:271–294.

Tsotsos, J. K., Culhane, S. M., Wai, W. Y. K., Lai, Y., Davis, N., and Nuflo, F. (1995). Modeling visual attention via selective tuning. *Artificial intelligence*, 78(1):507–545.

Walther, D. and Koch, C. (2006). Modeling attention to salient proto-objects. *Neural networks*, 19(9):1395–1407.

Xu, K., Ba, J., Kiros, R., Courville, A., Salakhutdinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*.