WESTERN UNIVERSITY
LONDON, ONTARIO, CANADA

A Study on Deep Convolution Neural Networks for Salient Object Detection

Sitong Li, sli2332@uwo.ca

Abstract

Devising efficient and effective methodologies for Salient Object Detection (SOD) is an important research topic in computer vision. Recently published paper at CVPR 2019 titled "BASNet: Boundary-Aware Salient Object Detection" achieves the state-of-the-art results at salient object detection on benchmark datasets. In this study, we dive into the neural net architecture they used to obtain those superior results. We begin by introducing the problem of salient object detection with discussion on its closely related problems. We provide a brief literature review on previous endeavours to SOD, describe how the neural net approaches finally become the dominant method in computer vision. Ultimately, we conduct a concentrated study on BASNet — the most recent best-performing neural net for SOD — by reproducing its results and performing more analysis on experimental results to further understand its behavior. Finally, we experimentally investigate the possibility of using a recently proposed focal loss to further improve the performance of BASNet.

1 INTRODUCTION

The phenomenon that humans are capable of detecting visually distinct, that is salient, objects/regions rapidly and effortlessly from presented scenes has been long studied by cognitive scientists. While research from these subjects are still mainly focus on understanding the biological/cognitive mechanism, scientists in computer vision community have already tried to devise computational machinery for automatically detecting salient objects from digital images. Several well-studied topics closely related to visual saliency include:

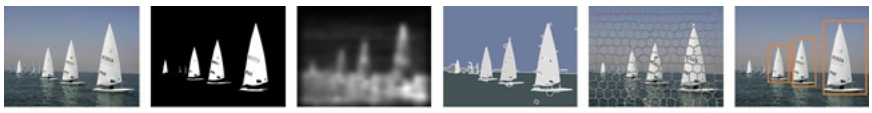

Figure 1.1: An sampled example showing the difference between various research topics. The leftmost is the original image. The rests are results produced by different research target; from left to right are respectively obtained by salient object detection, fixation prediction, image segmentation with regions in various sizes, image segmentation with super-pixels and object proposal. The example is from [7].

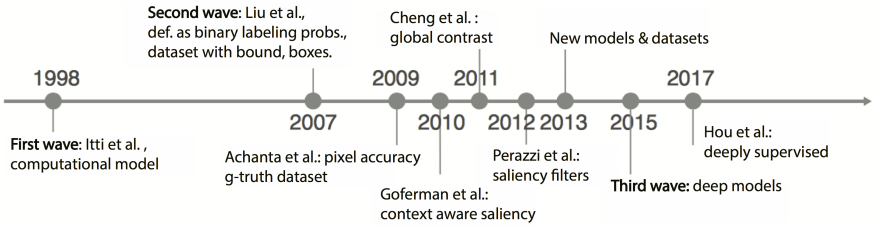

Figure 1.2: A chronological summary about previous work on salient object detection utill 2017 [7].

- Salient Object Detection [1] — detecting the most prominent object from an image;
- Fixation Prediction [8] — predicting the regions focused by human eyes;
- Object Proposal [37] — outline an bounding box that contains the candidate object.
- Semantic Segmentation [14] – detecting all objects in an scene.

It is easy to see that SOD is a special case of semantic image segmentation, where only the most prominent object is of interest. Semantic image segmentation is more challenging than SOD in the sense that for each pixel, more than two labels exist for prediction, whereas in SOD, each pixel only has two-classes representing whether it is a part of the salient object or not. Another observation is that fixation prediction is closely related to SOD as humans often pay attention on the most prominent object in the scene. In this study, we are predominantly interested in Salient Object Detection, which can be interpreted as a processes of two stages:

1. detecting the most salient object;
2. accurately segmenting the region of that object.

Historically speaking, the developments of SOD modeling might be roughly divided into three waves. The first wave is due to Itti et al. [41] who initiated the

research area by discussing it from multiple disciplines including cognitive psychology, neuroscience, and computer science. Psychological theories inspired bottom-up attention-based methods were developed. However, SOD in computer vision, by design, does not necessarily need to simulate human's vision system whose underlying mechanism is still largely obscure. The second wave is brought by Liu et al [57], who explicitly defined salient object detection as a binary segmentation problem. After such an clear definition, a number of work were subsequently proposed and obtained improved results on this objective. Many datasets were developed. The third wave is brought by the revival of deep learning [49, 73], specially due to the outstanding results of using deep convolutional neural networks for image classification [48] with GPU for back-propagation [72]. The revolution spreads to semantic image segmentation, resulting to the development of fully convolution neural networks [59]. In contrast to previous work that relies heuristic cues, FCNNs do not use hand-crafted features/heuristics at all, alleviating the fundamental bias brought by heuristic external knowledge. Figure 1.2 chronologically summarizes the research until 2017.

We are primarily focused on research from the third-wave. The rest of the report is organized as follows. In Section 2, we review how deep neural networks gradually become the dominant approach in computer vision starting from image classification, after that we review important SOD works before BASNet. Section 3 contains an detailed explanation of BASNet. Section 4 are empirical studies. Finally, we conclude in Section 5 by commenting on a few future directions for developing more efficient and effective neural net models for SOD.

2 Fully Convolution Neural Networks for SOD

Before discussing recent advances of using fully convolutional neural networks for SOD, we first review the recent development of using neural nets for image classification, which can be regarded as a problem closely related to image segmentation. Due to their intrinsic relation, we shall see that generic techniques in improving image classification are very likely to be able to bring improvement in SOD.

2.1 Deep Learning for Image Classification

The advancement of deep learning techniques has resulted breakthrough in many research directions in computer vision. In effect, deep learning is about credit assignment in adaptive systems with long chains of potentially causal links between actions and outcomes. The notion "Deep Learning" was first used by Dechter in 1986 [20], and in the context of Artificial Neural Networks (ANNs) by Aizenberg et al. in 2000 [2]. Contemporary meaning of deep learning specifically refers to neural network architectures of many stacked layers including deep belief networks [35], deep fully connected neural networks, deep convolutional neural networks [50] and recurrent neural networks [36]. See a summary in [5]. Starting from the 1960s, deep or shallow neural networks have been used in conjunction with supervised, unsupervised, and reinforcement learning; see the detailed survey by Schmidhuber [73] and

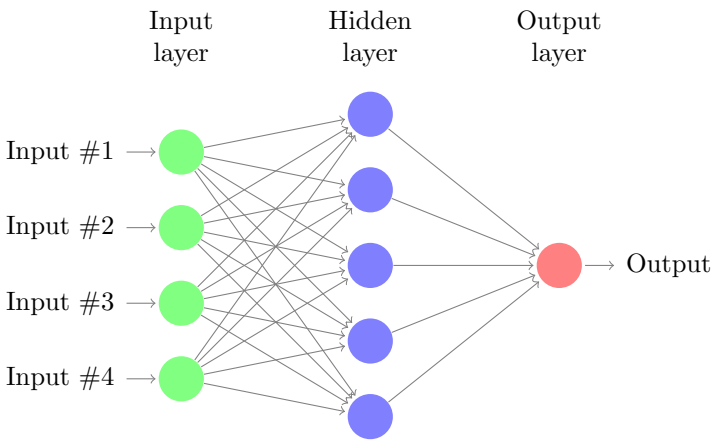

Figure 2.1: A fully connected neural net with one hidden layer; it has 4 inputs and one output.

a more focused and recent overview by LeCun et al. [49].

In 1970s, inspired by studies from neurophysiology, the idea of convolutional neural network (CNN) architecture was described by Fukushima [28] in “neurocongintion”. Such architectures are widely used in a number of scientific areas today, in particular, computer vision. In CNNs, (usually square) *receptive field* of a unit with some given weight vector (called *filter* or *kernel*) is shifted step by step across the array of input values, e.g., the pixels of an image. See Figure 2.2 for a demonstration. There are often many filters for capturing different “features” of the input information, resulting a stacked output of many 2d maps.

The output can be further processed by applying a point-wise “activation function”; a popular choice is rectified non-linearity function $f(x) \triangleq \max\{0, x\}$ [30, 64]. The resulting array of output subsequently provides input to units in the next “layer” repeatedly. Compared to fully-connected nets, because of the massive weight replication, relatively few parameters are needed to describe convolutional neural nets. Downsampling layers are used to reduce dimension of the input from previous layer. Early downsampling units use "Spatial Averag”. Weng [85] replaced Spatial Averaging by “Max-Pooling" and obtained better results.

A fundamental problem is how to update the connection weights of a neural net, given the observed difference between the predicted output and the “true” result. Algorithm used in most feed-forward and recurrent neural nets is called back-propagation (BP), whose continuous form was first derived in the early 1960s [44, 11]. Dreyfus [23] published the derivation of BP based on the chain rule only. The efficient implementation in FORTRAN for discrete sparse networks was first published by Linnainmaa [54]. Dreyfus [24] adopted BP in a controller experiment. Werbos [87] published the first application of BP to NNs, extending his earlier thoughts in 1974 thesis [86]. From 1980 to 1990, computers widely accessible in academic labs had gradually became 10^4 times faster than those of 1960 to 1970. A notable computa-

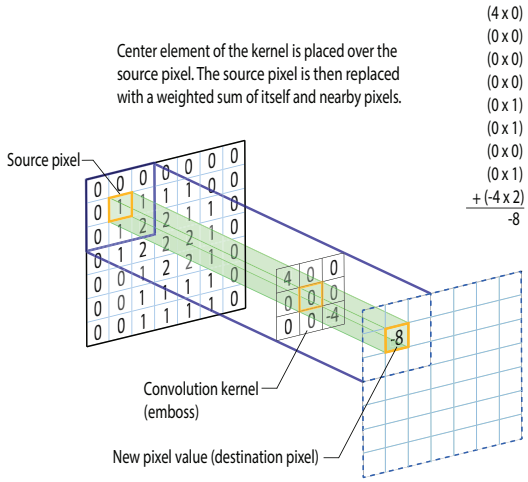

Figure 2.2: Illustration of convolution operation in 2-dimension image. Each filter (kernel) contains a vector of weights, applying which step by step horizontally and vertically using the same filter transforms the original picture into a new image. Here the filter is with size 3×3; the input image was has an extra border padding with 0, therefore the resultant image remains the same size. More generally, for a $W \times W$ image, suppose the padding is P, convolution size is $F \times F$, stride is S, then the resulting image will have dimension $W' \times W'$ where $W' = \frac{W+2P-F}{S} + 1$.

tional experiments by Rumelhart et al. [72] demonstrated that BP in NNs can indeed yield useful internal representations in hidden layers of neural nets. LeCun et al. [50] (called LeNet) first applied BP to Neocognitron-like CNNs, obtaining impressive results on a handwritten digit dataset called MNIST.

In the 2000s, computing hardware had again become 10^4 times faster than those of 1980s. The advantage of Graphics Processing Units (GPUs) (originally used for video games) in numerical computation began to be harnessed by neural net researchers: fully-connected neural net implemented on GPU were 20 times faster than on CPU [66]. Ranzato et al. [60] first applied BP to Max-Pooling CNNs. A fully-connected NN trained by BP obtained a new record of 0.35% error rate [17] on the MNIST handwritten digit dataset. Later, more efficient and parallel GPU implemented CNNs with Max-pooling further improved the MNIST record dramatically [18], finally achieving human performance (around 0.2% error rate) for the first time [16].

The record-breaking winner [48] (now commonly known as AlexNet) of the 2012 ImageNet [21] classification contest significantly popularized the use of deep neural nets implemented with GPUs in the computer vision community. As deep neural nets can easily overfit on the training dataset, AlexNet adopts dropout to alleviate overfitting, whose merit as a regularizer was investigated in [75]. Since then, the challenge of ImageNet has been dominated by deep CNNs. Th winner of 2013 is ZFNet [90] whose architecture resembles AlexNet but with more fine-tuned hyper-

Table 2.1: Evolution of winning neural net architectures on ILSVRC ImageNet object recognition competition from 2012 to 2015.

Year	CNN	Author	Rank	Top-5 Error Rate
2012	AlexNet	Krizhevsky et al.	1st	15.3%
2013	ZFNet	Zeiller and Fergus	1st	14.8%
2014	GoogLeNet	Google	1st	6.6 %
2014	VGG-Net	Simonyan et al.	2nd	7.3%
2015	ResNet	He et al.	1st	3.6%

parameters. A even deeper GoogLeNet (also called Inception v1) [77] became the winner in 2014, with VGG Net as the runner up [74]. These empirical successes suggest that depth of neural networks is a crucial ingredient for achieving high performance. However, the accompanied problem is that network training becomes more difficult with increasing depth due to the phenomena of *exploding* and *vanishing* gradients. He et al. [33] proposed Residual Net (ResNet), which adopts *short-cut* connections to tackle the gradient vanishing/exploding problem, making it possible to train neural nets over 100 layers. ResNet is the underlying architecture of the winner in the 2015 ImageNet competition. A similar architecture called Highway Network also uses short-cut connections [76].

Although theory behind BP shows how to update neural net connection weights, in practice, for a large dataset, it is often infeasible to optimize the neural net each iteration by feeding all data altogether. Specifically, denote the neural net as a θ parameterized function f_θ, given a dataset $\mathscr{D}$ of n images, the optimization target can be defined as an empirical risk function $R(\theta) = \frac{1}{n}\sum_{i=1}^{n} error(y_i, f_\theta(x_i))$. Challenge arises when n is very large because it requires expensive evaluations of the gradients from all summed error functions. The solution is using stochastic (or "on-line") gradient descent, which approximates the true gradient of $R(\theta)$ by a gradient at a single example: $\theta \leftarrow \theta - \alpha\nabla error(y_i, f_\theta(x_i))$. The algorithm can be implemented by sweeping through the training set, and perform the update for each training example. Several passes could be conducted over the training set until convergence. A more popular choice is not computing the true stochastic gradient at a single example, but against a mini-batch of training example at each step. With appropriate adjustment of the learning rate α, training by SGD converges to either optima or local optimal given the function is either convex or non-convex; see the analysis in [9]. Some new SGD variants that may accelerate the convergence have also been proposed, e.g., Adam [45], RMSProp [79] and AdaGrad [25]. See [10] for a comprehensive survey.

2.2 Fully Convolutional Networks for SOD

The drastic improvements in image classification due to deep CNNs also inspired the research in image segmentation. For visual tasks such as biomedical image processing and image manipulation, precise image segmentation with pixel-level labelling is preferred [71, 62]. The large receptive fields and max-pooling layers in

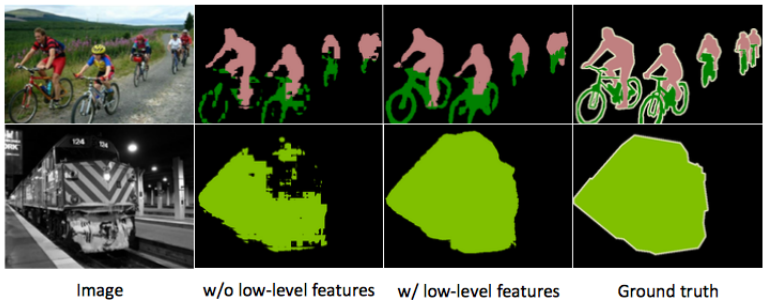

Figure 2.3: The prediction results with low-level features and without low-level features [78]. ResNet-50 is used as the backbone.

CNNs highly limit the fine segmentation of output [59, 12]. The direct strategy is to make a prediction at every pixel, in which each pixel is labeled with the class of its enclosing object or region [65, 68, 15, 26, 31, 32, 29]. This methods usually require pre- and post- processing complications, including superpixels [26, 32], proposals [31], or post-hoc refinement by random fields or local classifiers [26, 32]. Besides, pre-trained CNN could not be reinterpreted in these methods.

To address these problems, FCNNs (Fully Convolutional Neural Networks) were introduced and firstly trained end-to end for pixelwise prediction and from supervised pre-training in [59]. FCNN transfers fully connected layers into convolution layers for pixelwise prediction [59]. FCNNs are now popularly used for image segmentation, and show gratifying improvement in accuracy with various structures and strategies developed for refinement or upsampling [22, 4].

It is widely accepted that the top layers of deep neural networks contain high-level semantic information, while the bottom layers learn low-level fine details. As figure 2.3 shows, the downsampled low-level features are still able to refine the segmentation prediction substantially. Many SOD methods are developed based on the idea of fusing deep features from upper layers to lower layers [58], among which U-net is a representative [71]. U-net keeps a large number of feature channels in the upsampling part, which allows the network to propagate context information to higher resolution layer. Besides, the concatenation of the feature maps from correspondingly contracting path infused lower level features into expansive path. This U-shaped architecture is frequently used in other work on image segmentation [63, 46, 80]. UCF uses reformulated dropout to facilitate probabilistic training and inference and a hybrid upsampling method to reduce the artifacts of deconvolution operations [92].

HED (holistically-nested edge detection) [88] — first presented for edge detection — was later introduced for saliency detection [13, 38]. HED performs multi-scale prediction fusion by combining all side output predictions linearly — they can be deeply supervised. Those deep supervisions at the intermediate sides generate side-output predictions; the final result is a linear combination of all side-output predictions. Hou et al. (DSS+) [38] adopt HED by introducing short connections to

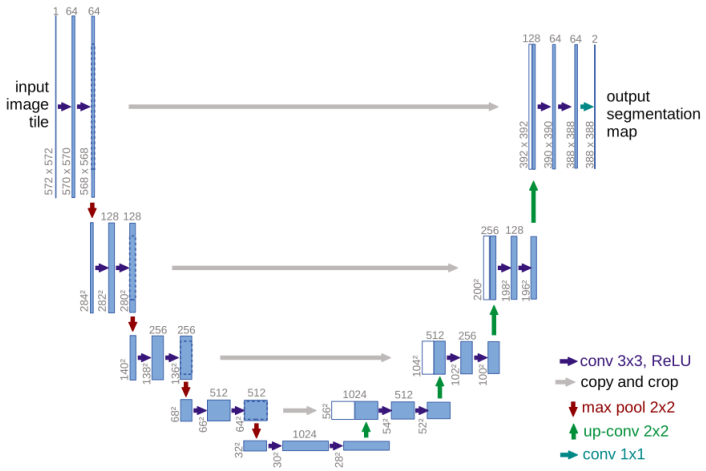

Figure 2.4: U-net architecture (example for 32x32 pixels in the lowest resolution) [71]. Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

its skip-layers for saliency prediction. Chen et al. (RAS) [4] use HED by refining its side-output iteratively using a reverse attention model [13].

The advantage of U-Net and HED could be easily combined by imposing deep supervision at each decoder stage of U-Net. Many recent saliency models fall into this category [94, 83, 56, 3, 91, 55, 34, 52, 43], where different fusion strategies are applied. One notable similarity of these models is that the final prediction is produced by a linear aggregation of side-output predictions. Hence the multi-scale learning is achieved in two aspects: i) the U-Net aggregates multi-level convolutional features from top layers to bottom layers in an encoder- decoder form; ii) the multi-scale side-output predictions are further linearly aggregated for final prediction. Current research in this field mainly focuses on the first aspect, and state-of-the-art models have designed very complex feature fusion strategies for this [56, 91].

On the other side, numerous refinement strategies have been proposed to capture finer structure and boundaries. A novel hierarchical refinement model (HRCNN) is proposed to hierarchically and progressively refine saliency maps to recover image details by integrating local context information without using over-segmentation methods [55].

Inspired by [67], Amirul et al. [40] proposed a refinement units to recurrently refine the saliency map produced by earlier layers by learning context-aware feature. In [22], a recurrent residual refinement network (R^3Net) is proposed to progressively refine saliency map by building a sequence of novel residual residual block (RRBs) to alternately use the low-level and high-level features. Although these methods raise the bar of salient object detection greatly, there is still a large room for improvement

in terms of the fine structure segmenting quality and boundary recovery accuracy.

Two main aspects to consider when predicting precise boundary are: (1) integration of high-level and low-level features [6]; (2) developing learning strategies and corresponding loss function to capture different level losses [27]. To introduce communications between hierarchical features, skip connections [88], short connections [38] and feature aggregations [91] have been proposed.

3 BASNET: BOUNDARY-AWARE SOD

In this section, we review BASNet [69]. We begin from the motivation, then explain how each component of BASNet is designed in detail; finally we show its advantage by comparing against existing architectures.

3.1 MOTIVATION

The major drawback of previous FCNN type networks is that their predicted saliency map is defective in two perspectives:

- fuzzy in fine structures;
- blurry in boundaries.

As shown in a demonstration example in Figure 3.1 from PiCANetR [56], one recent well-performing architecture for SOD. BASNet is proposed to address such deficiencies. It achieves the goal by novel combination and modification of following techniques:

1. encoder-decoder module [71],
2. residual refinement module [22],
3. hybrid loss consists of BCE [19], IoU [61] and SSIM [84],
4. deeply-supervised training [51].

The encoder-decoder is inspired from U-net [71]; refinement module has been adopted in [22]; BCE, IoU losses are widely used in computer vision, and SSIM has been argued as an very effective measurement in expressing structural similarities [84]; deeply-supervised learning is a general approach for better training deep neural networks [51].

3.2 ARCHITECTURE

The architecture of BASNet is depicted in Figure 3.2. The encoder part has an input convolution layer; it is followed by six stages using residual blocks. The input convolution layer and the first four stages are adapted from ResNet-34 [33], with the difference that in BASNet the input layer has 64 convolution filters with size of 3×3 and stride of 1, rather than 7×7 with stride of 2. No pooling after the input layer; this

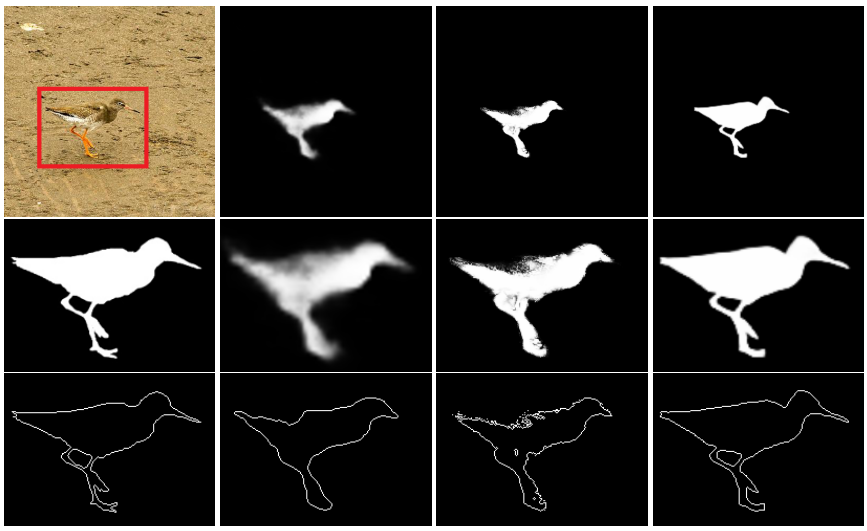

Figure 3.1: Sample result of BASNet in comparison to PiCANetR [56]. The first column shows the input image, zoom-in view of ground truth (GT) and the boundary map, respectively. The rest three columns are results of PiCANetR, PiCANetRC (PiCANetR with CRF [47] post-processing), and BASNet. Result from PiCANetR is blurry in fine structure and poor at outlining object boundary.

ensures that the feature maps before the second stage have exactly the same spatial resolution as the input image, which is also different from the original ResNet-34 who has quarter scale resolution in its first feature map. Arguably, this adaptation can make the network obtain higher resolution feature maps, meanwhile decreasing the overall receptive fields. To have the same receptive field size as ResNet-34, two more stages after the fourth stage are added, each of which has three basic res-blocks with 512 filters after a non-overlapping max pooling layer with size of 2.

To capture global information, BASNet adds a bridge stage between its encoder and decoder; it is made by three convolution layers with 512 dilation 2 [89] 3×3 filters; each layer is followed by a batch normalization [39] and a ReLU activation function. The decoder is almost symmetrical to the encoder, where each stage has three convolution layers followed by a batch normalization and a ReLU activation function. Input to each stage is the upsampled output from previous stage appended by its corresponding stage in the encoder. To obtain side-output saliency maps for deeply-supervised training, each output from the bridge stage and every decoder stage is fed to a plain 3×3 convolution layer followed by a bilinear upsampling then a sigmoid function. By such, given an input image, BASNet produces eight saliency maps in the training process. Note that although every saliency map is upsampled to the size of input image, the last one is taken as the final output of the predict module; it is then passed forward for refinement.

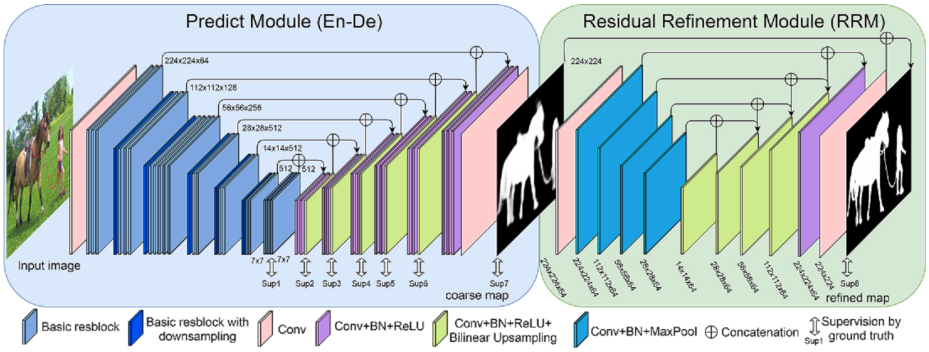

Figure 3.2: Architecture of BASNet. It consists of an Encoder-Decoder (En-De) prediction module and a Residual Refinement Module (RRM).

Refinement Module (RM) [40, 22] is typically designed as a residual block that tries to refine the predicted coarse saliency map S_{coarse} by learning the residual $S_{residual}$ between the saliency map and the ground truth:

$$S_{refined} = S_{coarse} + S_{residual}. \tag{3.1}$$

The definition of the term "coarse" includes two aspects: 1) the blurry and noisy boundaries; 2) unevenly predicted regional probabilities. Usually, real predicted coarse saliency maps exhibit both characteristics. Residual refinement module based on local context (RRM_LC) was proposed for boundary refinement in [67]. Because the receptive field is small, Islam *et al.* [40] and Deng *et al.*[22] either iteratively or recurrently use it for refining saliency maps on different scales. In [82], pyramid pooling module is adopted; it concatenates three-scale pyramid pooling features. To avoid lost information by applying pooling, RRM_MS instead adopts convolutions with various kernel sizes and dilation [89] for capturing multi-scale context.

The commonality of all these modules is that they are shallow. Therefore it could be hard to capture high level information during refinement. To tackle the drawbacks in region and boundary coarse saliency maps, BASNet instead adopts a new residual refinement module that uses the residual encoder-decoder architecture, denoted by RRM_BASNet. It is similar to but simpler than the described predict module; it is made by an input layer, an encoder, a bridge, a decoder and an output layer. Both encoder and decoder have four stages, which is slightly different from the prediction module; each stage has only one convolution layer containing 64 filters of size 3×3 followed by a batch normalization and a ReLU activation function. The bridge stage is made by a convolution layer with 64 filters of size 3×3 followed by a batch normalization and ReLU activation. Non-overlapping max pooling and bilinear interpolation are respectively used for downsampling in the encoder and upsampling in the decoder. Output of this RM module is the final saliency map of BASNet.

3.3 Loss Function

Let θ be the neural net, the final loss is simple summation over all intermediate losses:

$$\mathscr{L}(\theta) = \sum_{k=1}^{K} \ell_k(\theta_k) \tag{3.2}$$

where l_k is the loss of the k-th side output, K denotes the total number of the outputs and α_k is the weight of each loss. As mentioned earlier, the salient object detection model is deeply-supervised with $K = 8$ outputs where seven are from the prediction model and one from the refinement module. To obtain high quality prediction, each l_k is a hybrid loss:

$$\ell_k = \ell_{bce}^{(k)} + \ell_{ssim}^{(k)} + \ell_{iou}^{(k)} \tag{3.3}$$

where $\ell_{bce}^{(k)}$, $\ell_{ssim}^{(k)}$ and $\ell_{iou}^{(k)}$, as the name indicates, are respectively BCE, IoU and SSIM losses. The BCE is for binary classification, defined as:

$$\ell_{bce} = -\sum_{(r,c)} [G(r,c)\log(S(r,c)) + (1-G(r,c))\log(1-S(r,c))] \tag{3.4}$$

where $G(r,c) \in \{0,1\}$ is the ground truth of the pixel (r,c), $S(r,c)$ is the predicted probability of being salient object. The drawback of BCE is that it may be poor at differentiating pixels closing to boundaries. That is, saliency maps predicted by models trained solely with BCE often produce blurred boundaries. On the other hand, SSIM, originally proposed for image quality assessment [84], can capture the structural information in an image. Denote $\mathbf{x} = \{x_j : j = 1, ..., N^2\}$ and $\mathbf{y} = \{y_j : j = 1, ..., N^2\}$ as the pixel values of two corresponding patches (size: $N \times N$) cropped from the predicted probability map S and the ground truth mask G respectively, then SSIM loss of $\mathbf{x}$ and $\mathbf{y}$ can be expressed as:

$$\ell_{ssim} = 1 - \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3.5}$$

where μ_x, μ_y and σ_x, σ_y are the mean and standard deviations of $\mathbf{x}$ and $\mathbf{y}$ respectively; σ_{xy} is their covariance; $C_1 = 0.01^2$ and $C_2 = 0.03^2$ are used to prevent zero division.

IoU is a metric for measuring the similarity of two sets [42]. It is now used as a standard evaluation measure for object detection and segmentation. Using it as training loss has been explored in [70, 61]. For differentiability, the IoU proposed in [61] is borrowed:

$$\ell_{iou} = 1 - \frac{\sum_{r=1}^{H}\sum_{c=1}^{W} S(r,c)G(r,c)}{\sum_{r=1}^{H}\sum_{c=1}^{W} [S(r,c) + G(r,c) - S(r,c)G(r,c)]} \tag{3.6}$$

where $G(r,c) \in \{0,1\}$ is the ground truth label of the pixel (r,c), $S(r,c)$ is the predicted probability of being salient object.

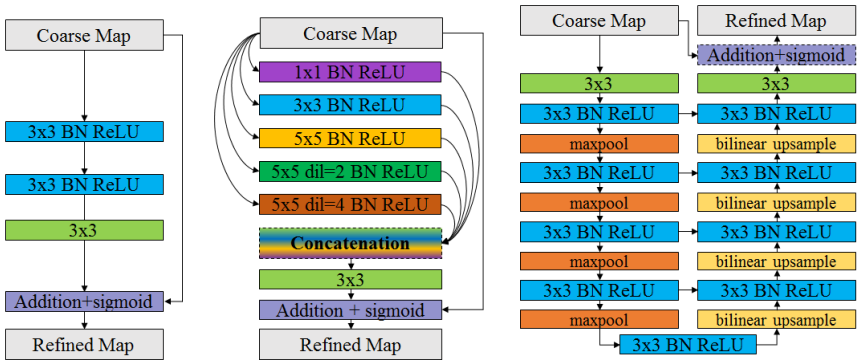

Figure 3.3: Comparison of different residual refine modules: local boundary refinement module RRM_LC; multi-scale refinement module RRM_MS; and encoder-decoder refinement module RRM_BASNet. RRM_BASNet is significantly deeper and larger than the other two.

3.4 Contributions

The major contribution of BASNet is a novel combination of existing techniques as well as its carefully designed architecture in integrating these technologies. The refinement module RRM_BASNet itself could also be a contribution. Figure 3.3 shows the comparison against two other existing refinement modules.

The advantage of a combined loss can be demonstrated by seeing the dissimilar impact of each individual measurement. See Figure 3.4. They represent three different measurements in three hierarchies, summarized as follows:

- BCE is pixel-level;
- SSIM is patch-level;
- IoU is map-level.

The task of SOD requires accurate prediction in all of these three levels. Therefore combing these three metrics together as the training loss is intuitively reasonable and technically sound.

4 Empirical Study

In this section, we reproduce the experiments of BASNet by rerunning its code on our computer. We then investigate the effectiveness of a recently proposed FOCAL loss. Our code is directly downloaded and modified from the authors' website [1]. The code is implemented in Pytorch. We run the code on a GTX-1080ti GPU with 11GB RAM.

[1] https://github.com/NathanUA/BASNet

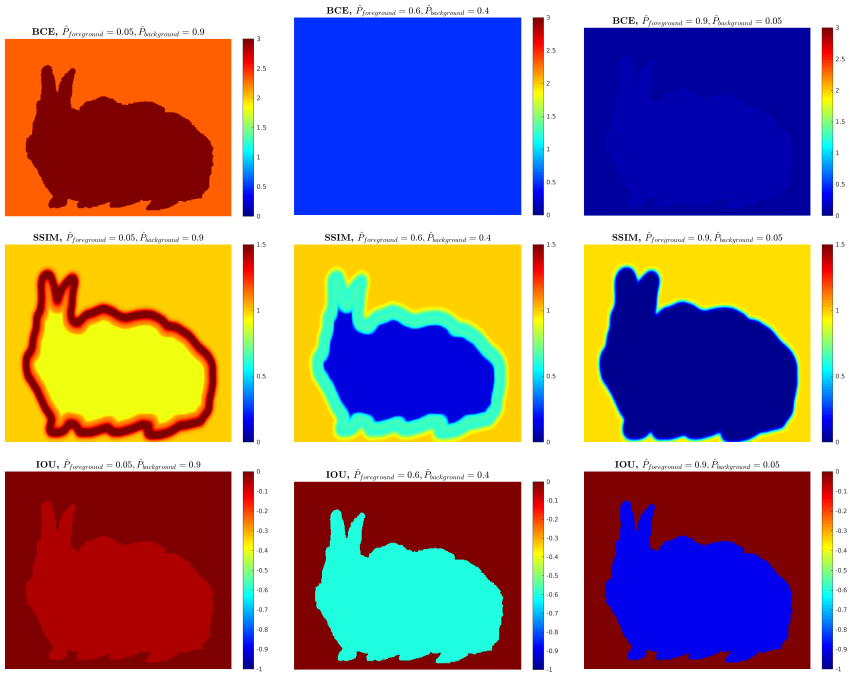

Figure 3.4: Demonstration of the difference between BCE, IoU and SSIM. $\hat{P}_{fg}$ and $\hat{P}_{bg}$ are respectively the assumed prediction probabilities of the foreground and background; color represents the loss of each pixel location. The leftmost column assumes predictions that are almost "totally" incorrect; rightmost is the opposite. In effect, reducing BCE error "equally" forces the learning model to predict correctly on both foreground and background; however, reducing IoU and SSIM has more impact on foreground and background. IoU loss is sensitive to boundary prediction.

Each training is stopped after 600,000 iterations, taking about 5 days. Each iteration contains a mini-batch of 8 images. The optimization algorithm is Adam [45], with parameter setting as follows: learning rate $\alpha = 0.001$, $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-8$, weight_decay = 0. Each image is re-sized into 256×256 then randomly cropped to 224×224. Training is conducted on DUTS-TR dataset that contains 10553 images. The dataset is augmented by horizontal flipping into 21106 images. DUTS [81] (available in [2]) is currently the largest saliency detection dataset, split into DUTS-TR and DUTS-TE. DUTS-TR is for training and DUTS-TE has 5019 images for testing.

4.1 Reproduced Results

In the original paper, learning curve was not plotted. To show the learning process, we retrain BASNet and plot the learning curve in Figure 4.1. There are 8 supervision heads in BASNet, therefore 8 hybrid losses. The loss on the final output is named as `loss0`; the rest are named from `loss1` to `loss7` according to their distances to `loss0` the final output. One observed phenomenon is that it is generally true that $loss0 < loss1 < \ldots < loss7$ throughout the training. This is, however, not surprising since these supervision "heads" before the final layer are with outputs upsampled from bilinear interpolation. The shallowest head has the smallest output map, therefore containing more coarse up-sampled predictions in its feature map, making it most difficult to produce an accurate saliency map.

During training, BASNet intermediately saves its neural net model. To see the learning progress, after the training has finished, we wrote a script to test the Mean Absolute Error (MAE) on the test data DUTS-TE, shown in Figure 4.2. Our results are consistent with those in the authors' paper, confirmed the superior performance of BASNet. Strikingly, we even obtained better MAE in our testing — the best error we obtained is 0.044 while the best reported in [69] is 0.047. Note that before BASNet, the best MAE reported on DUTS-TE is around 0.048 [69]; PiCANetR [56] obtained an MAE of 0.050.

4.2 Will Focal Loss be Useful in BASNet?

Recall how BCE is defined in Eq. 3.4. Recently, a focal loss has been proposed [53], defined as follows in the context of SOD:

$$\ell_{focal} = -\sum_{(r,c)} [G(r,c) \cdot (1 - S(r,c))^{\gamma} \cdot \log(S(r,c)) + (1 - G(r,c)) \cdot (S(r,c))^{\gamma} \cdot \log(1 - S(r,c))] \tag{4.1}$$

where γ is a tuning parameter usually set to 2. The advantage of Focal loss can be visualized in Figure 4.3. An implementation in Python is depicted in below. In contrast to cross entropy, focal loss assigns small weight to "well-classified" examples, forcing the learning model to spend more effort on fitting these poorly classified examples.

Listing 1: Pytorch implementation of focal loss

[2] http://saliencydetection.net/duts/

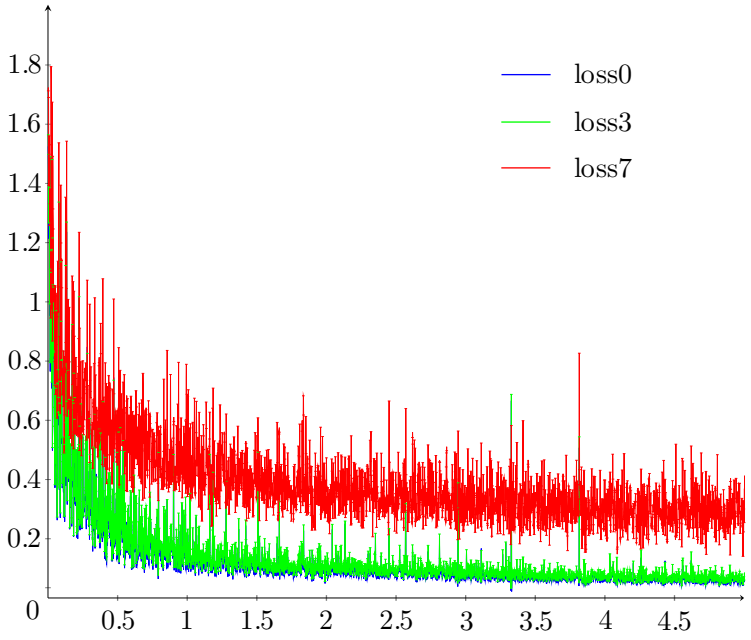

Figure 4.1: The training losses. There are 8 hybrid losses in total; loss0 is from the final layer; loss7 is the from shallowest supervision layer. To make the plot not too fuzzy, we only plot three of them. Results are obtained from running BASNet training on a GTX 1080ti GPU for around 5 days.

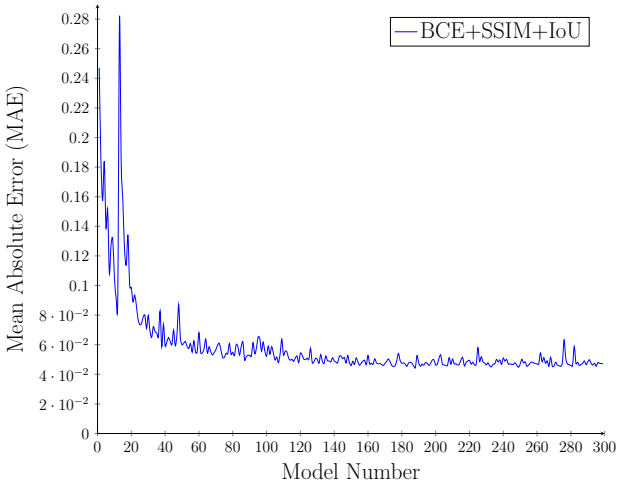

Figure 4.2: The learning progress measured by MAE on DUTS-TE. Consistent to Figure 3.4, it seems the learning has converged after half of the total training time.

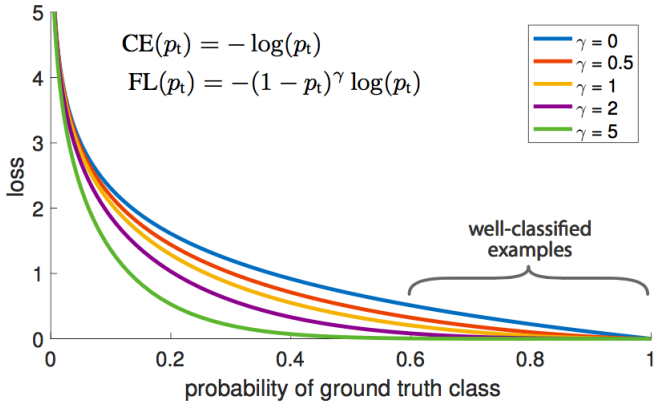

Figure 4.3: A comparison between focal loss and cross entropy. Focal loss provides very small error on "well-classified" examples, therefore may force the learning model to focus on improving its accuracy on "had" examples. Image from [53].

```
class FocalLoss(nn.Module):
    def __init__(self, alpha=1, gamma=2, logits=False, reduce=
        True):
        super(FocalLoss, self).__init__()
        self.alpha = alpha
        self.gamma = gamma
        self.logits = logits
        self.reduce = reduce

    def forward(self, inputs, targets):
        b1=F.binary_cross_entropy_with_logits(inputs, targets,
            reduce=False)
        b2=F.binary_cross_entropy(inputs, targets, reduce=False)
        if self.logits:
            BCE_loss = b1
        else:
            BCE_loss = b2
        pt = torch.exp(-BCE_loss)
        F_loss = self.alpha * (1-pt)**self.gamma * BCE_loss

        if self.reduce:
            return torch.mean(F_loss)
        else:
            return F_loss
```

As we have discussed in previous sections, one primarily reason behind BASNet's success is that its well-tuned loss functions for the task of SOD. Since focal loss can be regarded as a refined version of cross entropy, it is a nature question to wonder

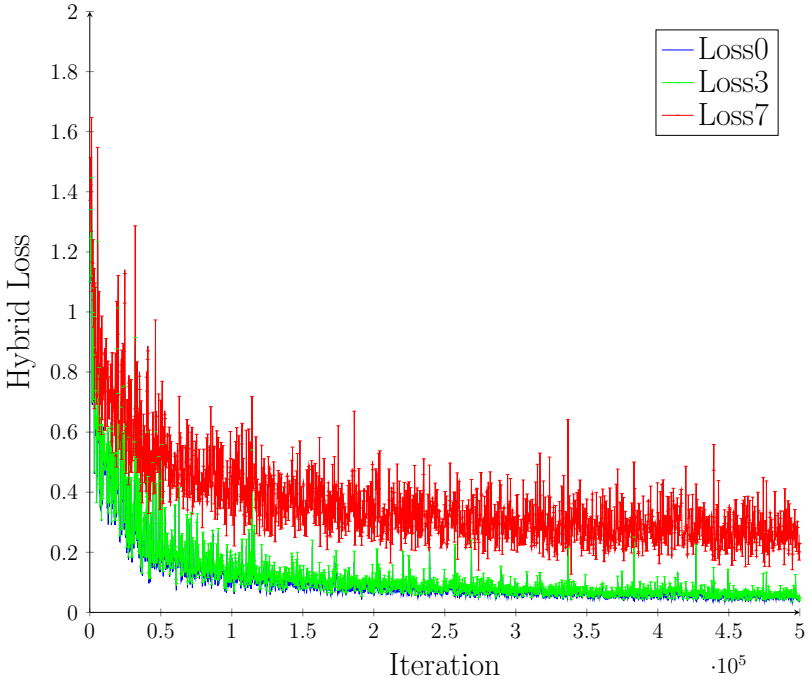

Figure 4.4: Loss curve after replacing BCE with `FocalLoss`. Same as before, results obtained after training for 600,000 iterations.

if replacing BCE to `FocalLoss` will further enhance the performance of BASNet. To answer this question, we implement `FocalLoss` in Pytorch and rerun BASNet training script. Figure 4.4 shows the resultant learning curve using around 5 days of training. It is easy to see that Figures 4.1 and 4.4 are very similar, presumably because that their training configurations are almost identical except one loss term.

Figure 4.5 shows the testing MAE of neural net models obtained by using `FocalLoss`. To compare, the curve of BCE is plotted alongside. From Figure 4.5 we see that `FocalLoss` failed to bring significant improvement, as these two curves largely overlap each other. `FocalLoss` seems to have smaller variance and converges a little faster in the early stage of the training.

Yet one more performance metric is Max F-Measure. F measure is a comprehensive measure on both precision and recall of the test. It is defined as weighted harmonic mean of the precision and recall:

$$F_\beta = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{4.2}$$

where β^2 is set to 0.3 to put more weight to precision than recall. The $maximum\ F_\beta$ during training processes is plotted in Figures 4.6. We can see that the original model reaches $maximum\ F_\beta$ above 0.8 earlier than `FocalLoss`. As the training progresses,

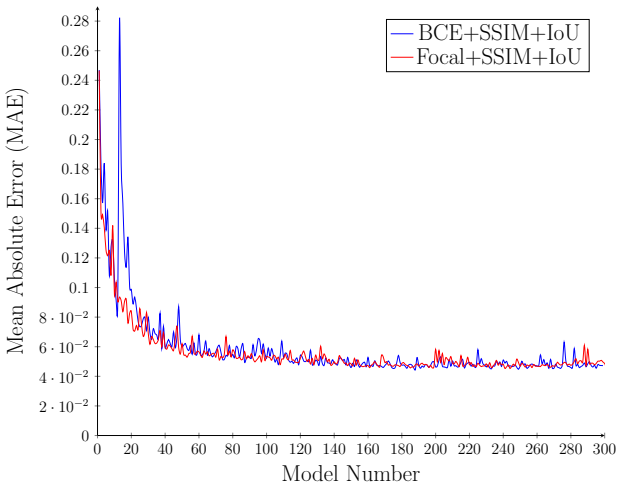

Figure 4.5: The learning progresses measured by MAE on DUTS-TE: comparison between `FocalLoss` and BCE.

the two models achieve similar $maximum\ F_\beta$ around 0.85, and both of two obtained highest $maximum\ F_\beta$ over 0.86 during the training.

5 Conclusions

We have provided a study on methods based on deep neural networks for salient object detection, with a particular focus on the state-of-the-art architecture BASNet. One major contribution of BASNet is their hybrid loss, which effectively guided the network to learn useful features in three hierarchies. We then conducted an experiment of replacing `BCE` with `FocalLoss`, a new loss metric that has been regarded as an improvement over BCE. However, in our experiment, solely replacing BCE with `FocalLoss` did not bring significant improvement in mean absolute error and maximum F measures on DUTS dataset.

On recent ICML conference, Richard Zhang proposed an anti-aliasing strategy by low-pass filtering before downsampling [93]. This strategy have been applied to several commonly-used architectures, such as ResNet, DenseNet, and MobileNet, and produced improved accuracy and robustness in ImageNet classification. The technique has been regarded as a general method in curing side-effects brought by downsampling. Since recent SOD models all use downsampling, a future direction is to integrate this idea into image segmentation models, such as BASNet, and see if it can lead to enhanced performance.

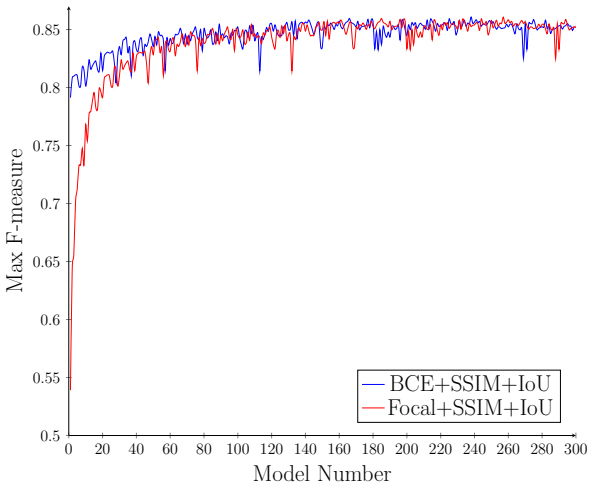

Figure 4.6: The learning progresses measured by Max F-measure on DUTS-TE: comparison between `FocalLoss` and BCE.

ACKNOWLEDGEMENT

Thanks the authors of BASNet from University of Alberta for providing helpful explanation for their source code and paper, in particular Xuebin Qin for providing Figures 3.1, 3.3, 3.2 and 3.4. Thanks Prof. Charles Ling for supervising this project.

REFERENCES

[1] Radhakrishna Achanta, Francisco Estrada, Patricia Wils, and Sabine Süsstrunk. Salient region detection and segmentation. In *International conference on computer vision systems*, pages 66–75. Springer, 2008.

[2] Igor Aizenberg, Naum N Aizenberg, and Joos PL Vandewalle. *Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications*. Springer Science & Business Media, 2013.

[3] Md Amirul Islam, Mahmoud Kalash, and Neil DB Bruce. Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7142–7150, 2018.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[5] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[6] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4380–4389, 2015.

[7] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *Computational Visual Media*, pages 1–34, 2014.

[8] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):185–207, 2012.

[9] Léon Bottou. Online algorithms and stochastic approximations. In David Saad, editor, *Online Learning and Neural Networks*. Cambridge University Press, Cambridge, UK, 1998. revised, oct 2012.

[10] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[11] Arthur E Bryson. A gradient method for optimizing multi-stage allocation processes. In *Proc. Harvard Univ. Symposium on digital computers and their applications*, volume 72, 1961.

[12] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[13] Shuhan Chen, Xiuli Tan, Ben Wang, and Xuelong Hu. Reverse attention for salient object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 234–250, 2018.

[14] Heng-Da Cheng, X_ H_ Jiang, Ying Sun, and Jingli Wang. Color image segmentation: advances and prospects. *Pattern recognition*, 34(12):2259–2281, 2001.

[15] Dan Ciresan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851, 2012.

[16] Dan Cireşan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. *arXiv preprint arXiv:1202.2745*, 2012.

[17] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12):3207–3220, 2010.

[18] Dan Claudiu Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[19] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

[20] Rina Dechter. *Learning while searching in constraint-satisfaction problems*. University of California, Computer Science Department, Cognitive Systems âĂę, 1986.

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[22] Zijun Deng, Xiaowei Hu, Lei Zhu, Xuemiao Xu, Jing Qin, Guoqiang Han, and Pheng-Ann Heng. R3net: Recurrent residual refinement network for saliency detection. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 684–690. AAAI Press, 2018.

[23] Stuart Dreyfus. The numerical solution of variational problems. *Journal of Mathematical Analysis and Applications*, 5(1):30–45, 1962.

[24] Stuart Dreyfus. The computational solution of optimal control problems with time lag. *IEEE Transactions on Automatic Control*, 18(4):383–385, 1973.

[25] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[26] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2012.

[27] Mengyang Feng, Huchuan Lu, and Errui Ding. Attentive feedback network for boundary-aware salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1623–1632, 2019.

[28] Kunihiko Fukushima. Neural network model for a mechanism of pattern recognition unaffected by shift in position-neocognitron. *IEICE Technical Report, A*, 62(10):658–665, 1979.

[29] Yaroslav Ganin and Victor Lempitsky. n^4-fields: Neural network nearest neighbor fields for image transforms. In *Asian Conference on Computer Vision*, pages 536–551. Springer, 2014.

[30] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

[31] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European conference on computer vision*, pages 345–360. Springer, 2014.

[32] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[34] Shengfeng He, Jianbo Jiao, Xiaodan Zhang, Guoqiang Han, and Rynson WH Lau. Delving into salient object subitizing and detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1059–1067, 2017.

[35] Geoffrey E Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009.

[36] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[37] Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? *IEEE transactions on pattern analysis and machine intelligence*, 38(4):814–830, 2015.

[38] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr. Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3203–3212, 2017.

[39] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[40] Md Amirul Islam, Mahmoud Kalash, Mrigank Rochan, Neil DB Bruce, and Yang Wang. Salient object detection using a context-aware refinement network. In *BMVC*, 2017.

[41] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1254–1259, 1998.

[42] Paul Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.

[43] Sen Jia and Neil DB Bruce. Richer and deeper supervision network for salient object detection. *arXiv preprint arXiv:1901.02425*, 2019.

[44] Henry J Kelley. Gradient theory of optimal flight paths. *Ars Journal*, 30(10):947–954, 1960.

[45] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[46] Simon Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R Ledsam, Klaus Maier-Hein, SM Ali Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A probabilistic u-net for segmentation of ambiguous images. In *Advances in Neural Information Processing Systems*, pages 6965–6975, 2018.

[47] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

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

[49] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[50] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[51] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pages 562–570, 2015.

[52] Zun Li, Congyan Lang, Yunpeng Chen, Junhao Liew, and Jiashi Feng. Deep reasoning with multi-scale context for salient object detection. *arXiv preprint arXiv:1901.08362*, 2019.

[53] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[54] Seppo Linnainmaa. The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors. *Master's Thesis (in Finnish), Univ. Helsinki*, pages 6–7, 1970.

[55] Nian Liu and Junwei Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 678–686, 2016.

[56] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3089–3098, 2018.

[57] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence*, 33(2):353–367, 2010.

[58] Yun Liu, Deng-Ping Fan, Guang-Yu Nie, Xinyu Zhang, Vahan Petrosyan, and Ming-Ming Cheng. Dna: Deeply-supervised nonlinear aggregation for salient object detection. *arXiv preprint arXiv:1903.12476*, 2019.

[59] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[60] Fu-Jie Huang MarcâĂŹAurelio Ranzato, Y-Lan Boureau, and Yann LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Proc. Computer Vision and Pattern Recognition Conference (CVPRâĂŹ07). IEEE Press*, volume 127, 2007.

[61] Gellért Máttyus, Wenjie Luo, and Raquel Urtasun. Deeproadmapper: Extracting road topology from aerial images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3438–3446, 2017.

[62] Roey Mechrez, Eli Shechtman, and Lihi Zelnik-Manor. Saliency driven image manipulation. *Machine Vision and Applications*, 30(2):189–202, 2019.

[63] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016.

[64] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[65] Feng Ning, Damien Delhomme, Yann LeCun, Fabio Piano, Léon Bottou, and Paolo Emilio Barbano. Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14:1360–1371, 2005.

[66] Kyoung-Su Oh and Keechul Jung. Gpu implementation of neural networks. *Pattern Recognition*, 37(6):1311–1314, 2004.

[67] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361, 2017.

[68] Pedro HO Pinheiro and Ronan Collobert. Recurrent convolutional neural networks for scene labeling. In *31st International Conference on Machine Learning (ICML)*, number CONF, 2014.

[69] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7479–7489, 2019.

[70] Md Atiqur Rahman and Yang Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International symposium on visual computing*, pages 234–244. Springer, 2016.

[71] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[72] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[73] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[74] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[75] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[76] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[77] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[78] Zhi Tian, Tong He, Chunhua Shen, and Youliang Yan. Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3126–3135, 2019.

[79] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[80] Caiyong Wang, Yong He, Yunfan Liu, Zhaofeng He, Ran He, and Zhenan Sun. Sclerasegnet: an improved u-net model with attention for accurate sclera segmentation.

[81] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 136–145, 2017.

[82] Tiantian Wang, Ali Borji, Lihe Zhang, Pingping Zhang, and Huchuan Lu. A stagewise refinement model for detecting salient objects in images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4019–4028, 2017.

[83] Wenguan Wang, Jianbing Shen, Xingping Dong, and Ali Borji. Salient object detection driven by fixation prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1711–1720, 2018.

[84] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[85] John J Weng, Narendra Ahuja, and Thomas S Huang. Learning recognition and segmentation of 3-d objects from 2-d images. In *1993 (4th) International Conference on Computer Vision*, pages 121–128. IEEE, 1993.

[86] Paul Werbos. Beyond regression:" new tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*, 1974.

[87] Paul J Werbos. Applications of advances in nonlinear sensitivity analysis. In *System modeling and optimization*, pages 762–770. Springer, 1982.

[88] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[89] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[90] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[91] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 202–211, 2017.

[92] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Baocai Yin. Learning uncertain convolutional features for accurate saliency detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 212–221, 2017.

[93] Richard Zhang. Making convolutional networks shift-invariant again. *arXiv preprint arXiv:1904.11486*, 2019.

[94] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 714–722, 2018.