On the Study of Reformulating Level Sets as Deep Recurrent Neural Network Approach to Semantic Segmentation

Mengda Li, Ethan Cohen, Adrien Bardes

Abstract

This is the project report of **MVA** Master[1] course *Medical Image Analysis* on the study of *Reformulating Level Sets as Deep Recurrent Neural Network Approach to Semantic Segmentation* by Ngan Le, Kha Gia Quach, Khoa Luu, Marios Savvides, Chenchen Zhu [1]. The project is divided in 3 parts. First, we recall the *Classic Level Set method* (**CLS**) proposed by Chan-Vese in 2001 in [2]. Secondly, the proposed *Recurrent Level Set* (**RLS**) by [1] will be introduced. At last, we present some experimental results and make an analysis of the place of this paper in the current state-of-the-art in semantic segmentation.

Frankly speaking, this article is rather for the purpose of image segmentation in *"classical" computer vision* than in medical image analysis. It is inspired by the *Variational Level Set* method which is commonly used in medical segmentation to design a recurrent neural network for *general* image segmentation. However, classical *Active Contour*-based level set methods have low segmentation accuracy on images collected in the wild conditions. To solve this issue, the authors of [1] propose the *Recurrent Level Set* (**RLS**) method powered with the *learning ability* of deep neural network.

1 Classical level set method: active contour

Many *Active Contour*-based approaches have good performance under some constraints of image, e.g. resolution, illumination, shape, noise, occlusions, etc. Among them, we review a classical and influential active contour method proposed by Chan and Vese in [2].

First, we initialize a contour randomly, then the curve is updated iteratively by minimizing an *energy function*. Finally, the converging curve would focus on segmented object in the image. See figure 1.

1.1 Description of the model

We want to have an active contour C segmenting a specific part of image. To do so is to minimize the *image intensity "variance"*s inside and outside the contour.

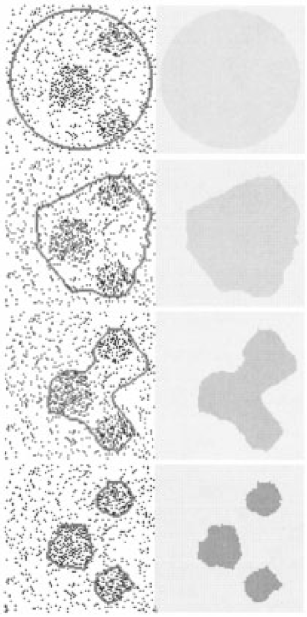

Figure 1: Detection of a simulated minefield, with contour without gradient

Let c_1, c_2 be the image intensity *means* inside and outside the contour C, and $F_1(c_1, c_2, C)$ the principal energy functional to be minimized with fixed parameters $\lambda_1, \lambda_2 > 0$ with $u(x,y)$ the original image pixel at (x,y) :

$$\begin{aligned} F_1(c_1, c_2, C) = \lambda_1 \iint_{inside(C)} |u(x,y) - c_1|^2 \, dx\, dy \\ + \lambda_2 \iint_{outside(C)} |u(x,y) - c_2|^2 \, dx\, dy \end{aligned} \tag{1}$$

Adding regularization term $F_2(C)$ with fixed parameters $\mu \geq 0, \nu \geq 0$ on the length and the area of contour would avoid over-fitting artifacts:

$$F_2(C) = \mu \cdot \text{Length}(C) + \nu \cdot \text{Area}(inside(C)) \tag{2}$$

Therefore, our optimization problem is to minimize the final energy functional $F(c_1, c_2, C) = F_1(c_1, c_2, C) + F_2(C)$.

[1] M2 MVA Mathématiques / Vision / Apprentissage

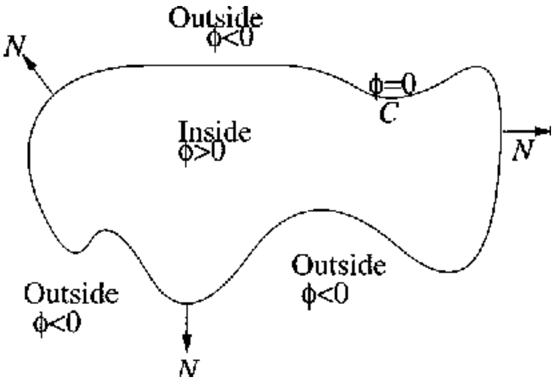

Figure 2: Curve $C = f(x; y) : (x; y) = g$ propagating in normal direction.

1.2 Model formulation by level set

The image is segmented by a Lipschitz function ϕ such that:

$$C = \{(x, y) : \phi(x, y) = 0\} \quad (3)$$
$$\mathit{inside}(C) = \{(x, y) : \phi(x, y) > 0\} \quad (4)$$
$$\mathit{outside}(C) = \{(x, y) : \phi(x, y) < 0\} \quad (5)$$

with

$$\text{Length}\{\phi = 0\} = \int \delta(\phi(x, y))|\nabla\phi(x, y)|\, dx\, dy \quad (6)$$
$$\text{Area}\{\phi \geq 0\}^{2} = \int H(\phi(x, y))\, dx\, dy \quad (7)$$

where δ is the Dirac delta function (distribution) and H is the Heaviside function.

The weighted image mean inside or outside of the contour could be defined as:

$$\begin{aligned} c_1(\phi) &= \text{average}(u) \text{ in } \{\phi > 0\} \\ &= \frac{\int u(x, y)H(\phi(x, y))\, dx\, dy}{\text{Area}(\mathit{inside}(C))} \\ c_2(\phi) &= \text{average}(u) \text{ in } \{\phi < 0\} \\ &= \frac{\int u(x, y)(1 - H(\phi(x, y)))\, dx\, dy}{\text{Area}(\mathit{outside}(C))} \end{aligned}$$

1.3 Optimization: Euler-Lagrange equation and gradient descent

So far, the energy functional F only depends on ϕ which is not differentiable and there is no *time* evolution dependence in ϕ. Recall that solving Euler-Lagrange equation could lead to good guess of optimum i.e. functions for which a given functional is stationary. By extending $\phi(x, y)$ to $\phi(t, x, y)$ (with $\phi(0, x, y) = \phi_0(x, y)$ the initial contour) and regularizing H, δ with smooth approximations $H_\epsilon, \delta_\epsilon$ as $\epsilon \to 0$, we deduce the associated Euler-Lagrange equation for ϕ keeping c_1, c_2 fixed and minimizing F with respect to ϕ:

$$\frac{\partial\phi}{\partial t} = \delta_\epsilon(\phi)\Bigg[\mu \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \nu \\ - \lambda_1(u - c_1)^2 + \lambda_2(u - c_2)^2\Bigg] = 0 \quad (8)$$

Hence, we can perform gradient descent by 9 to vanish the $\frac{\partial\phi}{\partial t}$:

$$\phi_{t+1} = \phi_t - \eta\frac{\partial\phi_t}{\partial t} \quad (9)$$

with a learning rate $\eta > 0$.

2 A brief introduction to RNN and GRU

RNN. A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. RNN creates the networks with loops in them, which allows it to persist the information. This loop structure allows the neural network to take the sequence of input. As you can see in the unrolled version (figure 3), first it takes the x(0) from the sequence of input and then it outputs h(0) which together with x(1) is the input for the next step. So, the h(0) and x(1) is the input for the next step. Similarly, h(1) from the next is the input with x(2) for the next step and so on. This way, it keeps remembering the context while training. RNN helps wherever we need context from the previous input

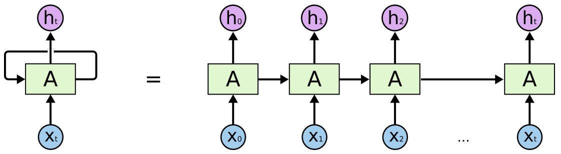

Figure 3: unrolled version of RNN

GRU (Gate Recurrent Unit). The GRU is a type of RNN that has gating units that modulate the flow of information inside the unit, but without having separate memory cells. That means the GRU fully exposes its memory content each timestep and balances between the previous memory content and the new memory content strictly using leaky integration, even though its

[2] Note that here the choice of strict inequality is not important, same for the integrals in c_1.

adaptive time constant is controlled by the update gate. The state updating process is illustrated as in figure 4.

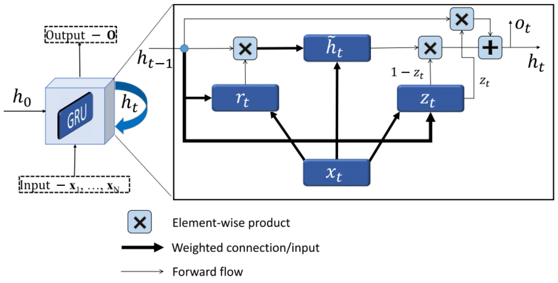

Figure 4: The unfolding GRU in time of the computation involved in its forward computation

3 Proposed method

3.1 Recurrent Level Set (RLS)

In this section, we take the CLS evolution introduced as an instance to demonstrate the idea of how to reformulate LS as an end-to-end trainable recurrent framework, named RLS. However, the proposed RLS can be applied to reform any LS approach once it successfully reforms CLS model because they share similar properties of curves moving over time. The first two columns in the Table 1 summarizes the coherence between GRU and CLS problems under three aspects: input, update rule and output. As shown in Table 1, the first difficult part of reformulating CLS as recurrent network is data configuration. Recurrent network works on sequence data while both the input and the output of the CLS approach are single images. The critical question is how to generate sequence data from a single image. Notably, there are two inputs used in CLS, i.e. an input image $\mathbf{I}$ and an initial LS function φ_0, which is updated by Eqn.(9) In the proposed RLS, we need to generate a sequence data x_t $(t = 1, ..., N)$ from single image $\mathbf{I}$. In order to achieve this goal, we define a function $g(\mathbf{I}, \varphi_{t-1})$ as in Eqn.(14)

$$x_t = g(\mathbf{I}, \varphi_{t-1}) = \kappa(\varphi_{t-1}) - U_g((\mathbf{I} - c_1))^2 + W_g((\mathbf{I} - c_2))^2 \tag{10}$$

In Eqn. (14), c_1 and c_2 are average values of inside and outside of the contour presented by the LS function φ_{t-1} .κ denotes the curvature $(\kappa(\varphi_t) = \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right))$. U_g and W_g are two matrices that control the force inside and outside of the contour. Clearly, during the curve evolution, the input at iteration t^{th} , x_t, is updated based on the input image $\mathbf{I}$ and the previous function φ_{t-1} which is in the same fashion as in LS . In the proposed RLS, x_t is considered to be input sequence whereas LS function φ_t is treated as hidden state. Notably, the initial LS function plays the role as initial hidden state.

So far, we have answered the question of generating input sequence from the single image $\mathbf{I}$. That means we use the same input as defined in LS problem, namely, the input image $\mathbf{I}$ and the initial φ_0. From the input, we are able to generate sequential data x_t. The next important task is generating the hidden state φ_t from the input data x_t and the previous hidden state φ_{t-1}. Under the same intuition of GRU , the procedure of generating hidden state φ_t is based on the updated gate z_t, the candidate memory content o_t and the previous activation unit φ_{t-1} as the rule given in Eqn.(15)

$$\varphi_t = z_t \circ \varphi_{t-1} + (1 - z_t) \circ o_t \tag{11}$$

	CLS	GRU	RLS
Input	Image $\mathbf{I}$ Initial LS function φ_0	Sequence $x_1, x_2, ..., x_N$ Initial state hidden h_0	Image $\mathbf{I}$ Initial LS function φ_0
Update	$\varphi_{t+1} = \varphi_t + \eta\frac{\partial\varphi_t}{\partial t}$ $\frac{\partial\varphi}{\partial t} = \delta_\epsilon(\varphi[\nu\kappa(\varphi - \mu)\lambda_1(\mathbf{I} - c_1)^2 + \lambda_2(\mathbf{I} - c_2)^2])$	$z_t = f(U_z x_t + W_z h_{t-1})$ $r_t = f(U_r x_t + W_r h_{t-1})$ $o_t = tanh(U_h x_t + W_h(h_{t-1} \circ r_t))$ $h_t = (1 - z_t)h_{t-1} + z_t o_t$	$x_t = \kappa(\varphi_{t-1}) - U_g((\mathbf{I} - c_1))^2 + W_g((\mathbf{I} - c_2))^2$ $z_t = \sigma(U_z x_t + W_z\varphi_{t-1} + b_z)$ $r_t = \sigma(U_r x_t + W_r\varphi_{t-1} + b_r)$ $o_t = tanh(U_o x_t + W_o(\varphi_{t-1} \circ r_t) + b_o)$ $\varphi_t = z_t \circ \varphi_{t-1} + (1 - z_t) \circ o_t$
Output	φ_N	$softmax(Vh_N)$	$softmax(V\varphi_N + b_V)$

Table 1: Comparison beetween CLS, GRU and RLS

The update gate z_t, which controls how much of the previous memory content is to be forgotten and how much of the new memory content is to be added is defined as in Eqn. (16)

$$z_t = \sigma(U_z x_t + W_z\varphi_{t-1} + b_z) \tag{12}$$

where σ is the sigmoid function and b_z is the update bias. The RLS, however, does not have any mechanism to control the degree to which its state is exposed, but exposes the whole state each time. The new candidate memory content o_t is computed as in Eqn.(17)

$$o_t = tanh(U_o x_t + W_o(\varphi_{t-1} \circ r_t) + b_o) \tag{13}$$

where b_o is the hidden bias.When r_t is close to 0 (off), the reset gate effectively makes the unit act as if it is reading the first symbol of an input sequence, allowing it to forget the previously computed state. The output $\mathbf{O}$ is computed from the current hidden states φ_t and then a softmax function is applied to obtain foreground/background segmentation $\hat{y}$ given the input image as follows,

$$\hat{y} = softmax(\mathbf{O}) = softmax(V\varphi_t + b) \tag{14}$$

where V is weighted matrix between hidden state and output. The proposed RLS model is trained in an end-to-end framework and its learning processes of the forward pass and the backward propagation are described as follows. The proposed RLS in folded mode is given in Fig 5 where the input of the network is defined as the same as the CLS model, namely, an input image $\mathbf{I}$ and an initial LS function φ_0. The LS function is initialized via checkerboard function. In the proposed RLS, the curve evolution from φ_{t-1} at time t-1 to the next step φ_t at time t is designed in the same fashion as the hidden state in GRU and is illustrated in Fig 5 where φ_t depends on both φ_{t-1} and the input x_t. We have summarized CLS, GRU and the proposed RLS in Table 1. It is easy to see that RLS shares the same input as CLS while updating procedure and output in the proposed RLS follows similar fashion as in GRU. With such design, the next parts show how to train the proposed RLS.

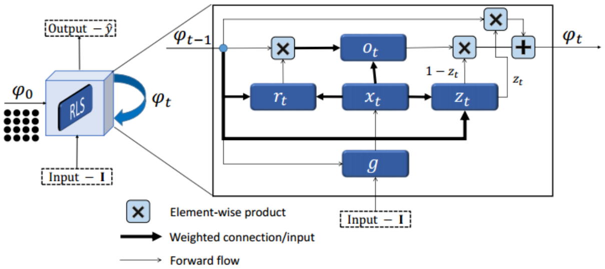

Figure 5: The proposed RLS network for curve updating process under the sequential evolution and its forward computation of curve evolution from time t-1 to time t

3.2 Contextual Recurrent Level Sets (CRLS)

The paper introduces CRLS for semantic segmentation. The model is constructed in 3 parts; Since the main contribution of the article is the second part that we develloped previously, we will not give details . They begin with an object detection using Region Proposal Network (RPN) [3] and a VGG-16 network [4]. Then they use 13 convolution layer as shown in the first part of the Figure 6 in order to get series of pre-defined boxes, or anchors, at each location. In the second part, they implemented the object segmentation as the second part of Figure 6. For each predicted box, the algorithm first extract feature via ROI warping layer [5], then the extracted features are passed through the proposed RLS with a randomly initial φ_0 to generate a sequence input data x_t based on Eqn (14).The curve evolution procedure is performed via LS updating process. The final part is the object classification using fully-connected network represented in the third part of the Figure 6.

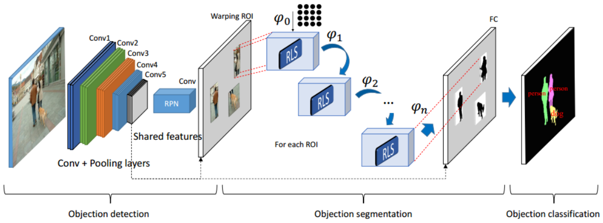

Figure 6: The entire process: Detection, segmentation and classification

4 Experimental results

The authors conducted several experiments to show the efficiency of their approach. The simple RLS approach is compared to the classical approach proposed by [2]. Then the full CRLS approach is compared to the state-of-the art methods in semantic segmentation.

RLS Object segmentation. Experiments with RLS have been conducted on a dataset containing the dataset of synthetic and medical images used by [2], and new real images from [6]. The dataset is augmented with transformed version of the original images, like rotation, translation, scale or flip.

Three approaches are compared. The classical approach of [2] presented in section 1 (CV), a simple method based on neural network (NN) and the proposed RLS approach. The performance measure is the F-measure, defined as the harmonic mean of precision and recall. The results obtained with respect to two ground truth are presented in the following table:

Methods	GT1	GT2	Test Time (s)
CV	88.51	87.51	13.5
FCN	93.3	93.26	0.001
RLS	99.16	99.17	0.008

The neural network methods are able to model the curve evolution of level-sets much better and have the ability of fine-tuning the final segmented shape.

CLRS Semantic segmentation. In order to compare with state-of-the-art semantic segmentation methods, a full experiment of the approach is conducted on the Pascal VOC dataset and on the more challenging MS COCO dataset.

5 State-of-the-art and Analysis

The most interesting thing about this paper is the iterative approach inspired by classical level set methods. Indeed, most deep learning based methods such as [7] rely on an end-to-end training where all the magic happens inside the network. Level-sets methods leverage the fact that drawing a precise boundary can be done iteratively by sharpening the proposed contour. Recurrent units have been designed to tackle this kind of sequential problems where one element in the sequence is highly correlated to its neighbors in the sequence. It is thus a logical choice for modeling level-sets approaches. After this paper introduced the main concepts, other attempts have been made to demonstrate the usefulness of the approach in medical image analysis and to improve over it.

A missing point of the article we study is the lack of analysis for medical images. Indeed, level-sets methods are very popular in the medical image analysis field [2, 6] and we would except from a level-set based method to have consistent improvement over standard medical image baselines. This missing work has been done by [8]. They evaluate the method on a challenging glioblastoma cell nuclei segmentation dataset and show the improvement over classical baselines. In addition, to the recurrent unit approach, they propose a novel focal loss function, and achieve state-of-the-art results on different cell segmentation tasks.

Unfortunately, this kind of approach based on recurrent units remains quite involved and does not yield state-of-the-art results on standard benchmarks such as Pascal VOC and COCO MS. The limitation of these approaches in semantic segmentation could be explained by the fact that it is designed for object segmentation and does not provide any benefit when it comes to understanding the scene. Moreover, these algorithms are usually computationally expensive. Indeed, minimizing the level-set energy as well as training recurrent units are expensive operations. Based on many contributions in the field of semantic segmentation, it appears that simpler approaches that does not try to model the problem as a sequential problem are able to achieve better performance while being easier to understand, implement, use and are computationally more tractable. Some of these approaches are still based on the idea of level-sets, or inspired by it.

Level-sets based methods can be classified into region-based and edge-based methods. The approach presented in this paper as well as the classical approach of [2] belongs to the region-based category. The idea of [9] is to smartly combine both region and edge based methods approaches into one powerful framework. Instead of proposing a weighted loss function as it was already done before, they proposed theoretical foundations of a new level-set framework that unifies edge and region segmentation methods. It consists in "hybrid level set models using a normalized intensity indicator function that allows the region information easily embedding into the edge-based model". The energy weights of region and edge terms can be then constrained by the global optimization condition deduced from the framework.

Moderns powerful deep learning based segmentation algorithms learn from a large set of annotated data. It is not the case of level-set methods that are a form of unsupervised-learning. The idea of [10, 11] is to use a standard deep learning procedure combined with a loss function inspired from the level-set framework. The latter is particularly interesting in the sense that the loss is designed to refine spatial details of segmentation results such as small objects and fine boundary, which is something very hard to achieve with conventional deep learning methods. Moreover, the method is designed from the beginning to tackle semantic segmentation tasks instead of simple object segmentation tasks. Finally, combining this loss function with a standard loss function make this approach generic.

Although modeling the semantic segmentation problem with level-sets seems to be an appropriate approach, it appears that state-of-the-art performance is obtained by other methods based on deep learning. In particular [7] introduced the concept of *atrous* convolutions to capture the multi-scale aspect of the problem.

The observation made by [7] is that responses at the final layer of deep convolutions neural networks (DCNN) are not sufficiently localized for accurate object segmentation. They thus proposed to combine the response at the final DCNN layer with a fully connected conditional random field. This work done in 2014 called DeepLab already surpass the performance of the method proposed by the paper we study, with a score of 71.6% above 0.5 threshold mean intersection over union on the Pascal VOC semantic segmentation task, whereas the level-set method only achieve 66.7%. Several iterations have been made over DeepLab. First in 2016, [12] proposed atrous spatial pyramid pooling to control the resolution at which feature responses are computed within DCNN. They achieve 79.7% on the Pascal VOC challenge. Then in 2017 and 2018, The work of [13, 14] proposed to get rid of conditional random field and replace it by an encoder-decoder structure able to capture sharper object boundaries by gradually recovering the spatial information. This final iteration named DeepLabv3+ is the current state-of-the-art in semantic segmentation, achieving 89.0% accuracy on the Pascal VOC benchmark.

References

[1] Ngan Le, Kha Gia Quach, Khoa Luu, Marios Savvides, and Chenchen Zhu. Reformulating Level Sets as Deep Recurrent Neural Network Approach to Semantic Segmentation. *arXiv e-prints*, page arXiv:1704.03593, Apr 2017.

[2] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, Feb 2001.

[3] He K. Girshick R. Ren, S. and J Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NIPS*, 2015.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.

[5] He K. Dai, J. and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. *CVPR*, 2016.

[6] Chunming Li, Chiu-Yen Kao, J. C. Gore, and Zhaohua Ding. Minimization of region-scalable fitting energy for image segmentation. *Trans. Img. Proc.*, 17(10):1940–1949, October 2008.

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

[8] Thomas Wollmann, Manuel Gunkel, I Chung, Holger Erfle, Karsten Rippe, and Karl Rohr. Gruu-net: Integrated convolutional and gated recurrent neural network for cell segmentation. *Medical image analysis*, 56:68–79, 2019.

[9] Weihang Zhang, Xue Wang, Wei You, Junfeng Chen, Peng Dai, and Pengbo Zhang. Resls: Region and edge synergetic level set framework for image segmentation. *IEEE Transactions on Image Processing*, 29:57–71, 2019.

[10] Boah Kim and Jong Chul Ye. Multiphase level-set loss for semi-supervised and unsupervised segmentation with deep learning. *CoRR*, abs/1904.02872, 2019.

[11] Youngeun Kim, Seunghyeon Kim, Taekyung Kim, and Changick Kim. Cnn-based semantic segmentation using level set loss. pages 1752–1760, 01 2019.

[12] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016.

[13] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[14] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018.