Reproducibility Justification and Simplification of Stacked U-Nets

Kaiwen Yuan
81169138
University of British Columbia
Department of Electric and Computer Engineering
kaiwen@ece.ubc.ca

He Zhang
83857169
University of British Columbia
Department of Electric and Computer Engineering
hezhang@ece.ubc.ca

Abstract

U-Nets is a popular and powerful light-weighted architecture for image classification and segmentation tasks, working well on small datasets, but retaining the limitation of performance degrade on natural images. Recently, an innovative structure of Stacked U-NETS has been proposed to tackle the limitation, in paper "STACKED U-NETS: A NO-FRILLS APPROACH TO NATURAL IMAGE SEGMENTATION" submitted to ICLR2019. In this project, we aim at justifying the reproducibility of the work in the paper by reimplementing Stacked U-Nets (SUNet-64) with tensorflow on image classification tasks. With exact same network structure and training hyperparameters of the original Stacked U-Nets, we could not reproduce the convergence of the image classfication task on ILSVR2012 (task3). Several limitations of the original Stacked U-Nets have been discovered. With these issues addressed, we further introduce two simplified Shallow Stacked U-Nets to work with small size images datasets. Experiment results show that the simplified models can successfully achieve good results on CIFAR-10 and CIFAR-100 but no advantages of adding U-Net blocks discovered. Comparisons between simplified Stacked U-Nets and ResNet-20 also show no benefits of this stacked U-Nets architecture on CIFAR datasets. Source code for our project can be found in github: https://github.com/KleinYuan/tf-stacked-unets.

1 Introduction

1.1 Background

Computer vision is one of the major topics in the artificial intelligence area, including classic tasks such as image classification, object detection, and image segmentation. Image classification is the fundamental and most mature task compared with other more complicated ones. Traditional approaches to tackle those tasks include hand-crafted feature extraction such as Scale-Invariant Feature Transform (SIFT) Lowe *et al.* [1], Histogram of Oriented Gradient (HOG) McConnell *et al.* [2], followed by second stage classification algorithm such as Support Vector Machine (SVM) Maji *et al.* [3]. One of the main constraints for those classic approaches is that features are hand-crafted, utilizing lower-level features with low generalization abilities.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

In 2012, with the development of more powerful Graphic Processing Unit (GPU), Deep Convolutional Neural Network (DCNN), which is capable of learning hierarchical layer-wise representation features, made breakthrough Krizhevsky *et al.* [4] in the performances of image classification. After that, a series of more sophisticated deep neural network algorithms have been proposed, such as ResNet He *et al.* [5], YOLO Redmon *et al.* [6], Faster R-CNN Ren *et al.* [7], Mask R-CNN He *et al.* [8], U-Nets Ronneberger *et al.* [9].

One common problem for DCNN based approach is that both training and inference processes are expensive due to large number of parameters, especially for image segmentation models such as Mask R-CNN or DeepLab Chen *et al.*[10]. Large amount of annotated datasets sometimes are not always feasible or accessible and thus many sophisticated model may not be able to fit.

U-Nets, a convolutional neural network based encoder-decoder structure is not only light-weighted but also designed for small amount of available dataset. Namely, it is able to utilize available annotated data more efficiently comparing with other state-of-the-art methods. This advantage benefits the research in fields such as bio-medical area which is significantly restrained by limited data resources.

The success of U-Nets also makes it a building module for more complicated structures such as DeepLabV3 Chen *et al.*[11], Pixel2Pixel Isola *et al.*[12]. However, one potential issue for bare-bone U-Nets is that it does not perform well on natural images with complex color profile and light effects.

To overcome the shortcoming and make U-Nets suitable for more dynamic scenarios, Stacked U-Nets [13] has been proposed. The main idea of Stacked U-Nets is to utilize U-Nets as a building block and stack the U-Nets module to form a hierarchical U-Nets structure. Besides existing advantages of conventional U-Nets structure, motivation of this paper is that more complicated U-Nets with stacked structure can generalize the performance to more complex natural images.

The Stacked U-Nets paper has been submitted to ICLR 2019 and the work of the Stacked U-Nets include: 1) proposing an innovative Stacked U-Nets structure; 2) achieving comparable performances on image classification tasks on ImageNet (ILSVR2012 datasets); 3) achieving better performances on image segmentation tasks on PASCAL VOC 2012 datasets compared with ResNet-101, with higher accuracy and fewer parameters.

In this project, also as part of ICLR 2019 Reproducibility Challenge [14], we focus on justifying the reproducibility of the Stacked U-Nets paper mainly on image classification task, which is the prerequisite step for further image segmentation task and evaluating the simplified Stacked U-Nets on CIFAR datasets. We describe several related works in Section 2. In Section 3, we review the architecture of U-Nets, Stacked U-Nets and simplified Shallow Stacked U-Nets. We describe experiments and analysis in Section 4, followed by Section 5 drawing conclusions and discuss future work.

1.2 Contributions

The main contributions of the project are summarized as follows:

- Stacked U-Nets proposed in the original paper has been reimplemented from scratch with Tensorflow.
- Experiments on SUNet-64 (one of the Stacked U-Nets structure proposed) have been performed with training and validating on ImageNet (ILSVR 2012 task3) datasets and with exact same training hyper-parameters and network structure, we are not able to get SUNet-64 converged.
- Problems of sensitivity to image size have been addressed and a simplified Stacked U-Nets has been experimented to successfully get promising results on CIFAR-10 and CIFAR-100 datasets.
- ResNet-20 has been utilized as baseline to verfiy that the stacked simplied unets structure do not perform better on CIFAR-10
- This project is also part of ICLR 2019 Reproducibility Challenge [14] to spread the spirits of reproducing academical publications.

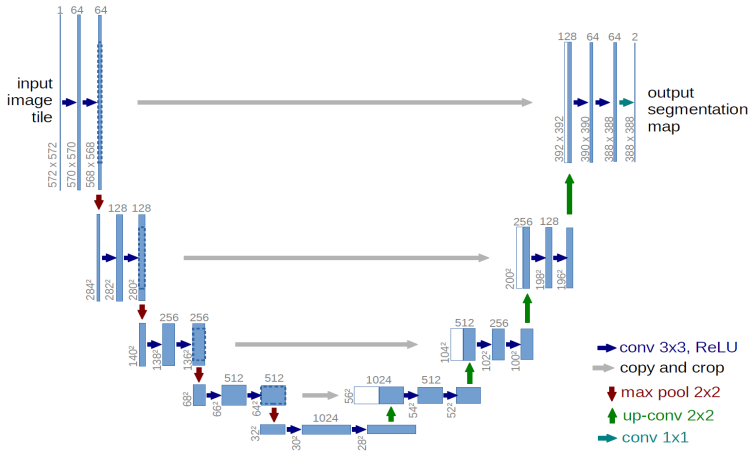


Figure 1: Basic architecture of U-Net Ronneberger *et al.*[9].

2 Related Work

In Krizhevsky *et al.* [4], a typical DCNN is proposed by taking the advantage of more powerful GPU. This network implements five convolutional layers, some of which are followed by max-pooling layer, and a fully connected layer at the end. The dropout method is applied to achieve efficiency. In Lin *et al.* [15], an even deeper architecture is built by proposing an inception module. In He *et al.* [5], the author propose a ResNet and present residual learning framework to enable deeper network architecture, which is used in this project as baseline. Two main challenge for those DCNN methods are that parameters are heavy and and rely on large datasets.

To tackle those challenges, an encoder-decoder structure is designed in several works to enable more efficient utilization of the limited amount of datasets with fewer parameters. U-Net is one of the successful structures designed by Ronneberger *et al.* [9]. This network consists of an encoding path and decoding path, shown in Fig. 1. With the convolutional layer and max-pooling layer, the image is contracted in the encoding path. Then, with upsampling (deconvolution, bilinear or nearest-neighbour), the image is expanded to the same size in the decoding path. The concatenation is implemented between encoding path and decoding path so that the high resolution features at the encoding path are combined with up-sampled output at the decoding path. In this case, the structure of u-nets makes it possible to capture context information at multiple scales and propagate them to the higher resolution layers.

U-Net then becomes a building module for more complicated structures with the benefits of being light-weighted and less dependent on large datasets. For example, the combination of U-Net and generative adversarial network (GAN) has proven to be able to achieve more accurate performance on the image to image translation task. In Isola *et al.*[12], conditional GAN is combined with U-Net for image-to-image translation problem. In this work, the U-Net based architecture is utilized as the generator of the image-to-image translation framework. This architecture is proven to be effective for a wider range of problems. Motivated by the idea of applying U-Net as the generator of GAN, more advanced architectures have been proposed. In Di *et al.* [16], a gender preserving GAN (GP-GAN) is proposed to leverage the advantages of U-Net and DenseNet for face synthesis. Besides applying U-Net in GAN, the U-Net module has been implemented for new architecture design to tackle challenges in image segmentation. In Chen *et al.*[11], DeepLab is designed by applying the convolution with upsampled filters, and an atrous spatial pyramid pooling, to robustly segment objects at multiple scales. In Jegou *et al.* [17], the authors proposed an equivalent U-Net based on the DenseNet to reduce the number of parameters. In Newell *et al.* [18], the authors apply multiple stacks of u-net modules for human-pose estimation.

However, U-Net also has a known issue that conventional structure does not perform well on natural images. Different from the work in [18], Stacked U-Nets [13], retains the basic u-net structure from Ronneberger *et al.* [9], operates without any intermediate supervision and processes features by

progressively downsampling, trying to further generalize U-Net to more complex natural images with the stacked deeper U-Nets.

3 Architecture of Stacked U-Net

This section describes the architecture of the Stacked U-Nets we are going to implement. We first depict the original Stacked U-Nets architecture in paper [13]. Then, we introduce the modify/simplified Stacked U-Nets as shallower architecture.

3.1 Original Architecture

In this section, we describe the original Stacked U-Nets architecture introduced in [13]. We first describe the U-Net module, based on which, the Stacked U-Nets is described.

3.1.1 U-Net Module

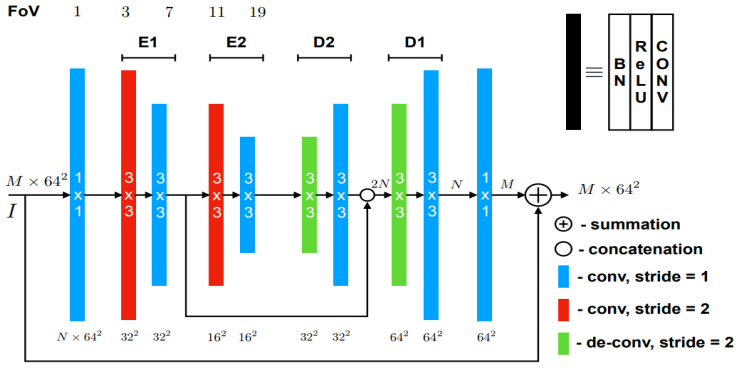


Figure 2: Basic architecture of U-Net for Stacked U-Nets [13].

Different from the original U-Net architecture introduced in Ronneberger *et al.*[9], the U-Net module for Stacked U-Nets is modified as shown in Fig. 2. The depth of the network is reduced to ten layers. Each layer consists of a batch normalization, a ReLU nonlinearty activation, and a convolution. The first convolution layer, with a 1×1 kernel, is responsible of transforming the input image with M features into a fixed number of features, N, in the following layers. The second and the third layers constitute the first encoder E1, in which the convolution layer with stride 2 is responsible of down-sampling and replacing the functionality of max-pooling. The convolution layer with stride 2 is followed by a convolution layer with stride 1. This layer is responsible of mitigating high frequency noise from sampling operation. Encoder E2 has the same architecture as encoder E1. After the two encoders E1 and E2. The up-sampling path consists of two decoders D2 and D1 with similar architecture. Each decoder consists of a deconvolution layer with stride 2 and a convolution layer with stride 1. The deconvolution layer with stride 2 is responsible of up-sampling of the feature map. The deconvolution layer is followed by a 3×3 convolution layer. After the two decoder D1 and D2, the image passes through a convolution layer with 1×1 kernel, which is used to transform the image with N feature channels to the image with M feature channels to remain the original size of the input image, so that the image can pass through the next U-Net module. Moreover, the concatenation is performed between the outputs of the encoder E1 and decoder D2. The concatenation helps to merge the high resolution features at the output of E1 and the low resolution features at the output of D2 together.

The U-Net module has several properties. First, the sampling and de-sampling, handled by the convolution and deconvolution layers, facilitates information exchange between the lower and higher resolution features. Second, every layer uses 3×3 kernels and outputs a fixed number of feature map, N. Third, the max-pooling operation designed in the original U-Net is replaced with strided convolution. The strided convolution enables different filters in each U-Net module to operate at different resolutions. Fourth, the number of features at the input and at the output of the U-Net module are the same, which enables stacking multiple U-Net modules without loosing the resolution.

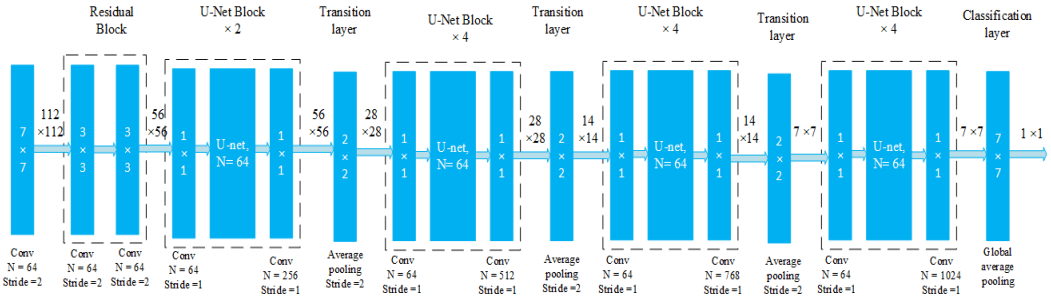
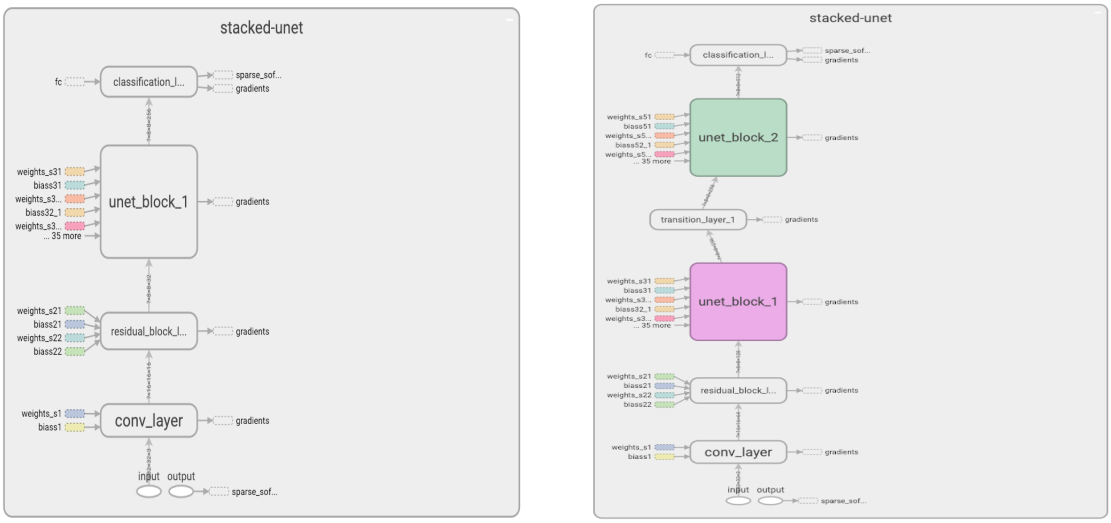


Figure 3: Architecture of Stacked U-Nets, SUNet-64, for image classification [13].



(a) SUNet-64-u1 (b) SUNet-64-u2

Figure 4: Structure of SUNet-64-u1 and SUNet-64-u1

Sixth, the U-Net module is not as deep as that of the original U-Net architecture, so that more U-Net modules can be stacked.

3.1.2 Stacked U-Nets Architecture for Image Classification

With the U-Net module, the Stacked U-Nets architecture can be designed for image classification. The designed Stacked U-Nets is shown in Fig. 3, which corresponds to the SUNet-64 in the original paper. We note that each convolution layer consists of batch normalization, ReLU activation, and convolution. Different layers differ in the number of output feature map, N, and total number of U-Net modules in each block. The input images are processed using a 7×7 convolution filter followed by a residual block. The feature maps are processed bottom-up and as well as top-down by multiple stacks of U-Nets at different scales and with regularly decreasing resolutions. Moreover, since that the size of input image to block 4 is 7×7, the U-Net in block 4 removes the encoder E2 and decoder D2.

3.2 Simplfied Architecture: Shallow Stacked U-Net

In this section, we modify the original Stacked U-Nets architecture, SUNet-64, and obtain two simplified versions: SUNet-64-u1 and SUNet-64-u2. The structure of those two simplified Stacked U-Nets is shown in Fig. 4.

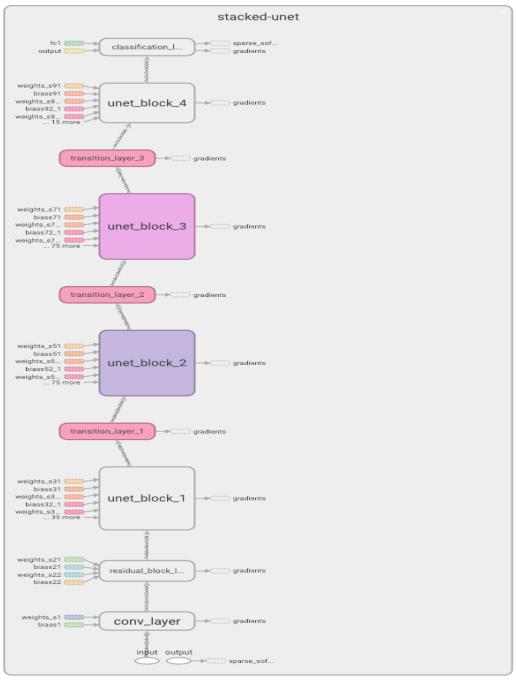


Figure 5: Tensorflow Graph of Reimplemented SUNet-64

In the first simplified Stacked U-Nets architecture, we only keep until the first U-Net block. In the second simplified Stacked U-Nets architecture, we only keep until the second U-Net block, but only consists of two repeated U-Net modules and pooling with 1×1 kernel and stride 1.

The motivation behind simplifying the original Stacked U-Nets is that the original one is deep and complicated, which may make the network sensitive to the size of the input image. The experiment results showed in the following section illustrate that the original Stacked U-Nets is not reproducible as the loss does not converge, but the simplified versions of Stacked U-Nets can obtain be converged easily.

4 Experiments and Analysis

4.1 Experiments Setup

A single NVIDIA GeForce GTX 980ti GPU has been used as the training device on an Ubuntu 14.04 system. This project is built based on Tensorflow framework, reimplementated from scratch referring to original paper descriptions. For all experiments, stochastic gradient descent (SGD) optimizer with a momentum of 0.9 (and 0.95) has been utilized and cross entropy loss with weight decay (L2-regularizer) of 10^{-4} has been used to evaluate the loss of each model. Data augmentation for all datasets has been implemented including on the fly random flip and rotate.

4.2 SUNet-64 on ILSVRC 2012

We first reimplement SUNet-64 using tensorflow, with exact the same network structure and hyper parameters of the original SUNet-64 introduced in [13], the structure and tensorflow graph of which is shown in Fig. 3 and Fig. 5. In order to align with original paper work and consider the cost of computational resources, Imagenet ILSVRC2012 Task 3, which is a subset of Task 1 and Task 2, has been used to train and validate the model. This datasets contains 120 categories of dogs, each of which has dynamic number and image sizes. Therefore, each image has been resized to 224 x 224 and normalized between 0 to 1. Various initial learning rate from 0.01 to 0.0002 with exponential decay of different rates from 0.96 to 0.1 has been explored. However, unfortunately, none of our trials is successful, namely, the significant decrease trend of the loss has not been observed.

4.3 Limitations and Hypotheses

Considering both ILSVRC 2012 is a large datasets and SUNet-64 is a complicated stacked structure, there could be few reasons this happened, listed as following:

- This structure makes neural network more sensitive to initialization and during our experiments, the initialization, which original paper does not described in details, could lead to a bad local optima.
- Gradient vanishing happens due to the complicated structure
- Some training tricks are not disclosed in original paper

Besides the problem of reproducibility, following limitations and confusions also have been disclosed:

- The network structure is sensitive to input tensor/image size. For example, SUNet-64 cannot work with CIFAR-10 or CIFAR-100 due to the series of pooling layers eventually keeping reducing image size from 32 to 1 repeatedly. Thus, it is difficult to scale on different or small datasets.
- Each U-Nets block is actually directly chained and there is no skip connection from previous U-Nets block to later one. For example, in SUNet-64, U-Nets block 1 is not connected with U-Nets block 4. Even though in the U-Nets block there could be minimal information loss, from disconnected U-Nets block may still confront of resolution loss due to the pooling. Therefore, firstly, it is confusing that this structure has been named as Stacked U-Nets; secondly, it seems more promising to add one connection between U-Nets block 2 to U-Nets block 3 and U-Nets block 1 to U-Nets block 4.

4.4 Shallow Stacked U-Nets on CIFAR

ILSVRC 2012 is a super large datasets, with total amount to be 1.2 million images for the training process. And it is very expensive to feed all those datasets into memory and run fast training, which is one of the main reason that in this project, only task 3 (a small amount subset) has been selected to experiments. However, MNIST and CIFAR-10/100 are more light-weighted and friendly for researchers to verify the ideas of improvements. Even though there are growing concerns that depending too much on those small simple datasets may result in model optimization bias on the limited datasets, considering the benefit of short period of experimental iterations, it is still worth to vaildate the model on those datasets.

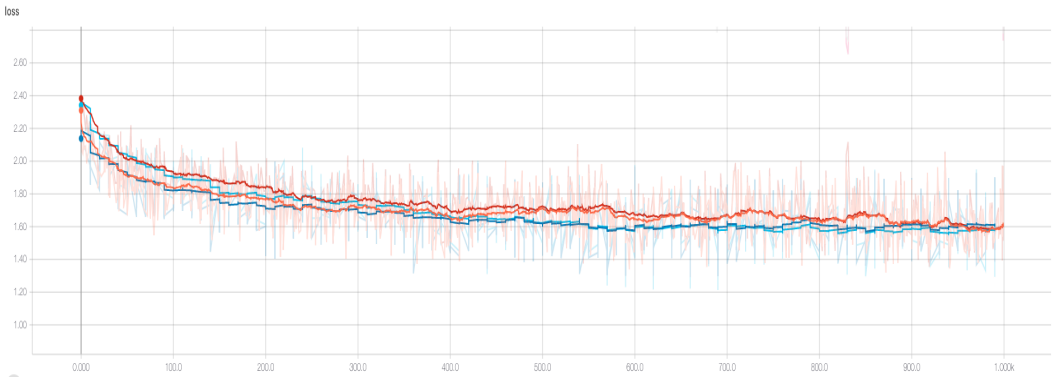
Addressed the limitation of Stacked U-Nets being sensitive to input image size, we explored two simplified Stacked-UNet structure, SUNet-64-u1 and SUNet-64-u2, to get trained on CIFAR-10 and CIFAR-100 successfully.

SUNet-64-u1 discards all layers after U-Nets block 1 and SUNet-64-u2 discards all layers after U-Nets block 2 with trainsition layer 1 using kernel size 1. With this structure, CIFAR datasets, which consists of 32×32 colour images in $10/100$ classes, are able to fit those structures.

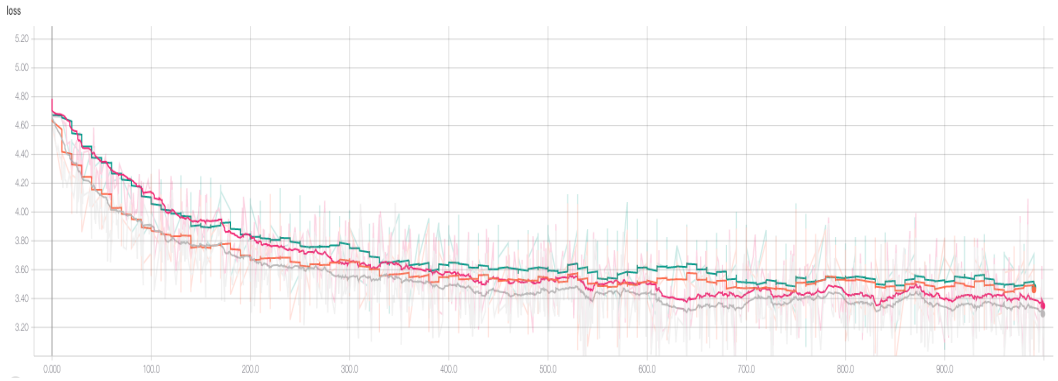
In the experiment, initial learning rate is set to 0.0001 with exponential decay of 0.96 and with batch size of 64, the (smoothed) training and validation curve is shown in Fig. 6.

As we can see that for CIFAR-10 datasets, both models achieve cross entropy with L2 regularization loss around 1.60; for CIFAR-100, around 3.40. The differences of performances between two different models are not obvious. By comparing the training errors and validation errors, it is implied that within the training range, the model is not over-fitted. Since we have not done any hyper-parameters tuning on validation datasets, we believe that the validation process could reflect the real test very well.

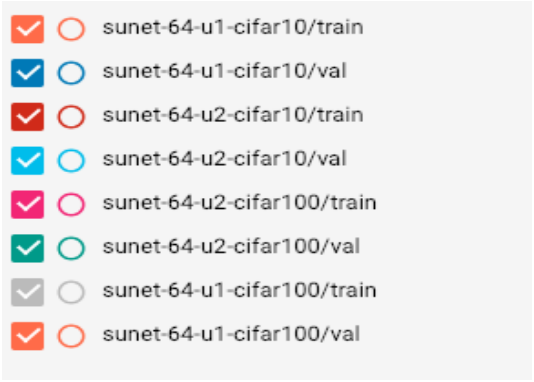
ResNet-20 with exact same training hyperparameters setup has been trained and validated on CIFAR-10 as baseline to compare, as shown in Fig. 7., which also implies that SUNet-64-u1 and SUNet-64-u2 do not perform better than ResNet-20. With same epochs, ResNet-20 achieves both lower training error and validation error and no obvious overfitting has been observed. In original publication, it is shown that SUNET-64 performs no better but comparable than ResNet on ImageNet, which to some degree aligns with the results of our experiments.



(a) Training Curve of SUNet-64-u1 and SUNet-64-u1 on CIFAR-10

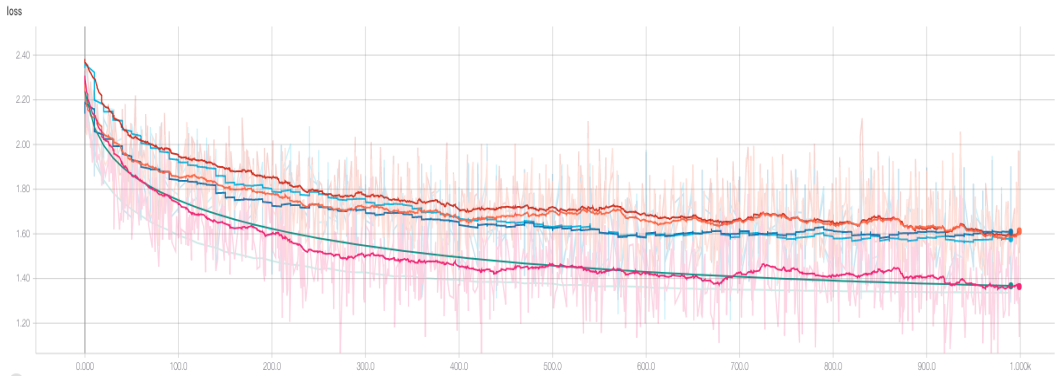


(b) Training Curve of SUNet-64-u1 and SUNet-64-u1 on CIFAR-100



(c) Color code

Figure 6: Training Curve of SUNet-64-u1 and SUNet-64-u1 on CIFAR-10 and CIFAR-100



(a) Training Curve of ResNet-20, SUNet-64-u1, SUNet-64-u1 on CIFAR-10

(b) Color code

Figure 7: Training Curve of ResNet-20, SUNet-64-u1 and SUNet-64-u1 on CIFAR-10

5 Discussion and Future Work

In this project, we have reimplemented Stacked U-Nets, but failed to reproduce the results of SUNet-64 training on ILSVRC 2012. Hypothesis of the cause has been raised that the initialization methods could be the key. However since the details have not been elaborated besides potential tricks may not exposed, we are not able to verify it. Limitation of this structure has been addressed that it is sensitive to input image size and therefore we have explored two simplified, trimmed Stacked U-Nets structures successfully getting trained on CIFAR-10 and CIFAR-100 without over-fitting. The performances between those two structures are not significantly different, thus we do not see advantages of adding more U-Nets blocks directly at least with CIFAR datasets. At last, we compare the simplied Stacked U-Nets with ResNet-20 and it implies that ResNet-20 performs better.

Even though the reproduce experiment is not successful, there are still several future works can be done to make further contributions:

- Contact author to verify the the reproduce experiments to exposed potential non-disclosed details
- Explore structure of connecting U-Nets block 1 to block 4 and block 2 to block 3
- Monitoring the statistics of weights on each layer to verify whether gradient vanishing is the cause
- Compare SUNet-64-u1 and SUNet-64-u2 on larger datasets
- The original paper has not compared their performances with conventional U-Nets which is worth adding an extra experiment to disply the advantages

References

[1] Lowe, David G. "Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image." U.S. Patent No. 6,711,293. 23 Mar. 2004.

[2] McConnell, Robert K. "Method of and apparatus for pattern recognition." U.S. Patent No. 4,567,610. 28 Jan. 1986.

[3] Maji, Subhransu, Alexander C. Berg, and Jitendra Malik. "Classification using intersection kernel support vector machines is efficient." Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008.

[4] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[5] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[6] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[7] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems. 2015.

[8] He, Kaiming, et al. "Mask r-cnn." Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017.

[9] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[10] Chen, Liang-Chieh, et al. "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs (2016)." arXiv preprint arXiv:1606.00915 (2016).

[11] Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." IEEE transactions on pattern analysis and machine intelligence 40.4 (2018): 834-848.

[12] Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." arXiv preprint (2017).

[13] "Stacked u-nets: a no-frills approach to natural image segmentation." https://openreview.net/forum?id=BJgFcj0qKX (2018).

[14] `https://github.com/reproducibility-challenge/iclr_2019/issues/40`

[15] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9.

[16] X. Di, V. A. Sindagi and V. M. Patel, "GP-GAN: Gender Preserving GAN for Synthesizing Faces from Landmarks," 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, 2018, pp. 1079-1084.

[17] Simon Jegou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation". In Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on, pp. 1175–1183. IEEE, 2017.

[18] Alejandro Newell, Kaiyu Yang, and Jia Deng. "Stacked hourglass networks for human pose estimation". In European Conference on Computer Vision, pp. 483–499. Springer, 2016.