COMPRESSING U-NET USING KNOWLEDGE DISTILLATION

Karttikeya Mangalam

Master's Semester Project
Computer Vision Lab
EPFL, Switzerland

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Mentored By: Dr. Mathieu Salzmann

17th January 2018

Introduction

Objective

Why?

Compressing the U-net architecture using Knowledge Distillation techniques with minimal loss in performance on Image Segmentation tasks.

Motivation

What?

Many state of the art models are too heavy to be used on mobile devices such as smartphones and other IoT products. Compression techniques are needed to be able to use these models in actual production.

Timeline : October 2017 – January 2018

Dataset Description

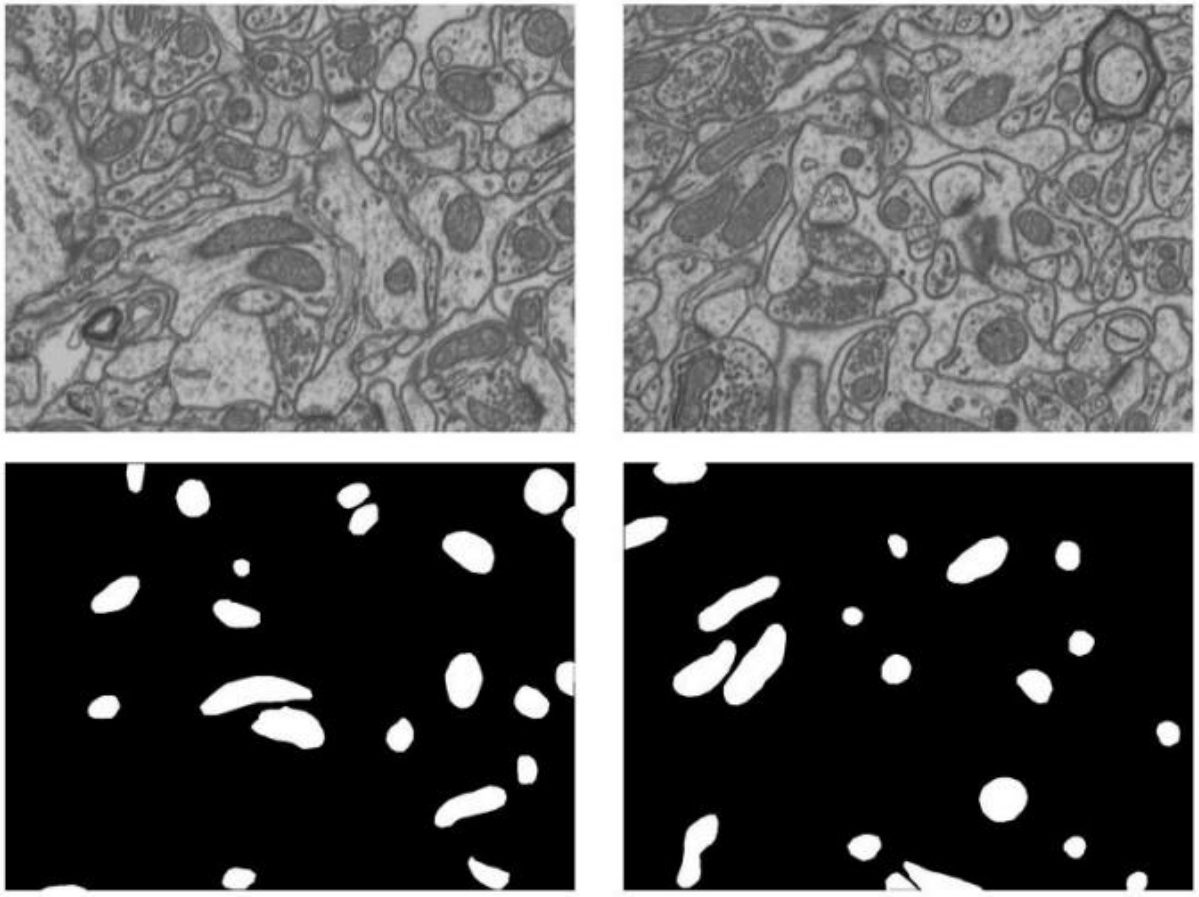
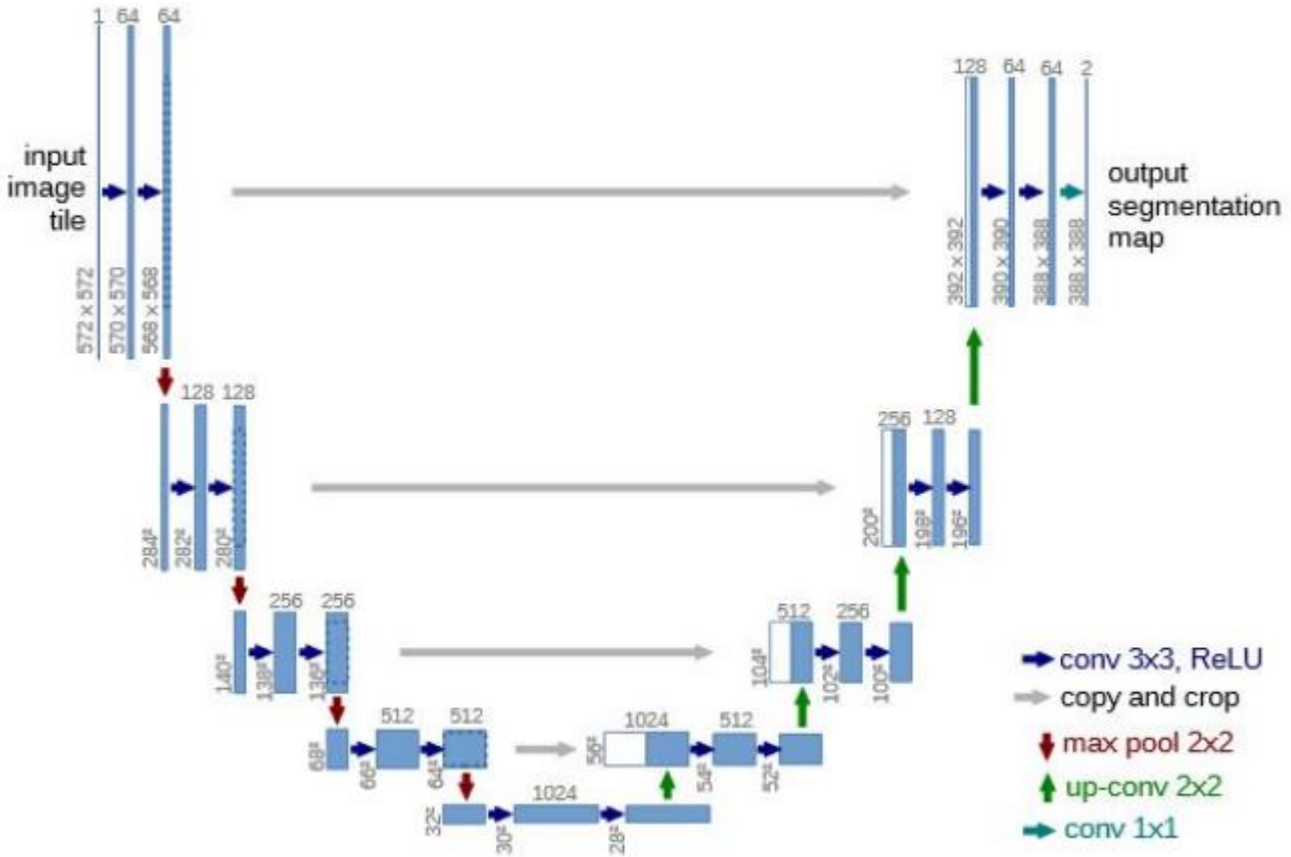


Fig. 2. A slice and it's corresponding annotation from the Mitochondria Segmentation in Electron Microscopy Dataset [16].

Binary Segmentation Task

Two separate volume for training and testing each of dimensions 1065x2048x1536

U-net Architecture



The Original 64-Unet.

We focus on reducing the model size by decreasing the number of input channels.

For notational convenience, A k-Unet is the same architecture with k-channels at the top stage.

Batch Normalization Layer is introduces in BLUE operations

Performance Metrics

- Intersection Over Union Score:
  - Reported only for the final models
- Cross Entropy Loss:
  - Reported both as the minimum and as the minimum of averaged loss over certain number of iterations.

$$\mathcal{H}(\mathbf{x}, \hat{\mathbf{x}}) = -\sum_{(i,j)} \mathbf{x}_{ij} \log(\hat{\mathbf{x}}_{ij}) + (1 - \mathbf{x}_{ij}) \log(1 - \hat{\mathbf{x}}_{ij})$$

$$\mathcal{L}^*_{train} = \min_k \frac{1}{\alpha_t} \sum_{i=k}^{k+\alpha_t} \mathcal{H}(y^t_i, \hat{y^t}_i)$$

$$\mathcal{L}^i_{test} = \frac{1}{n_{test}} \sum_{j=1}^{n_{test}} \mathcal{H}(y_j, \hat{y}_j)$$

$$\mathcal{L}^*_{test} = \min_k \frac{1}{\beta_t} \sum_{j=k}^{k+\beta_t} \mathcal{L}^{jn_t}_{test}$$

Searching for a Student Model

Starting Channel Depth	Test Loss	Train Loss	#Iterations
128	0.1170	0.0281	63,000
64	0.1021	0.0256	80,000
32	0.0871	0.0220	125,000
16	0.0822	0.0220	150,000
8	0.0830	0.0221	280,000
4	0.0974	0.0286	300,000
2	0.8227	0.3423	290,000
1	0.8337	0.3438	320,000

Table 1. Performance of different U-net architectures on varying the channel depth in the first layer. The loss parameters used are $\alpha_t = 5000$, $\beta_t = 25$ and $n_t = 500$. The reported number of iterations is where the minimum for the testing loss is achieved.

Surprisingly, the original U-net can be reduced to just 4-Unet without distillation!

We successfully demonstrate over 100x compression of the original model to 2-Unet

64-Unet : Over 31 Million parameters

2 – Unet: ~30,900 parameter (1% of 64-Unet)

The Distillation Procedure

1. Train the teacher network from scratch.
2. Use the teacher's predicted probability distribution (optionally including the original labels) to train the student model at a higher Softmax temperature.

$$\mathrm{P}_{\mathrm{T}}^{\tau} = \mathrm{softmax}\left(\frac{\mathbf{a}_T}{\tau}\right), \quad \mathrm{P}_{\mathrm{S}}^{\tau} = \mathrm{softmax}\left(\frac{\mathbf{a}_S}{\tau}\right).$$

$$\mathcal{L}_{KD}(\mathbf{W_S}) = \mathcal{H}(\mathbf{y_{true}}, \mathrm{P_S}) + \lambda \mathcal{H}(\mathrm{P}_{\mathrm{T}}^{\tau}, \mathrm{P}_{\mathrm{S}}^{\tau}),$$

3. For prediction, set the Softmax temperature back to 1.

Soft Training : 2-Unet

$T_{transfer}$	0.1	0.25	0.5	1	2	4	5	7	10	15	20
Test loss	Div.	Div.	Div.	0.111	0.480	0.104	0.610	0.481	0.480	0.480	0.134
#Iterations	-	-	-	15,000	15,000	50,000	50,000	9000	14,000	14,500	48,800

Table 2. Cross entropy test loss for soft training of 2-Unet. The teacher model is 4-Unet trained for 300,000 iterations with 0.0910 as the test loss. The above reported losses are the minimum across the test losses with no averaging. Div. indicates that the training diverged on several hyper parameter combinations.

Results indicate no obvious pattern between network's performance and softmax temperature used in the distillation process which is odd and warranted further investigation.

Intiailization Matters

$T_{transfer}$	Testing Loss (avg)			Testing loss (min)		
	Trial 1	Trial 2	Trial 3	Trial 1	Trial 2	Trial 3
1	0.481	**0.212**	0.483	0.480	**0.210**	0.480
2	0.488	0.488	0.485	0.480	0.480	0.480
3	0.482	0.482	**0.223**	0.480	0.480	**0.215**
5	**0.123**	0.482	0.489	**0.097**	0.480	0.480
6	**0.151**	0.525	0.482	**0.126**	0.496	0.480
10	**0.120**	0.482	0.482	**0.093**	0.480	0.480
15	**0.209**	0.482	0.491	**0.201**	0.480	0.480

Table 3. Three trials of distillation with 2-Unet as the student model. The parameters for average test loss are $\beta_t = 25$ and $n_t = 500$. The bold values indicate breaking the 0.48 test loss barrier.

The *0.48 loss* that is recurring above physically means that the network's prediction for each pixel to belong to class i is exactly the percentage of class i in the overall dataset and is actually independent of the pixel's value and it's surroundings.

$$\mathcal{P}^*(\mathbf{x}_{ij}) = \begin{cases} 0.947 & \mathbf{x}_{ij} \in \textit{background} \\ 0.053 & \mathbf{x}_{ij} \in \textit{foreground} \end{cases}$$

Some failed experiments I : Soft training 1-Unet

Trial Number	1	2	3	4	5	6	7	8	9	10
2-Unet	0.156	0.142	1.371	0.138	1.376	1.379	1.380	1.381	0.223	1.384
1-Unet	1.37	1.35	1.36	1.33	1.38	1.38	1.36	1.33	1.30	1.31

Table 4. Reruns of the smaller unet architectures with the vanilla training procedure. The reported numbers are the average cross entropy losses calculated on the testing dataset with the parameters $\beta_t = 25$ and $n_t = 500$.

1-Unet has *less than 0.5%* of the original parameters of 64-Unet

Degradation in **Expected** Performance

>=4-Unet : Always work, ~0.1 Loss
2-Unet : Sometimes works (0.1 Loss), mostly doesn't (~1.4 Loss)
1-Unet : Never works

Some failed experiments II : Sequential Distillation

Starting Model			Test Loss
Loss	#Iterations	$T_{transfer}$	
0.48	10,000	5	0.883
0.48	20,000	5	0.595
0.550	20,000	10	0.497
0.480	10,000	20	0.590
0.480	20,000	20	0.498

Table 6. Cross entropy loss for 1-Unet trained through sequential distillation. The averaging parameter for the test loss are $\beta_t = 25$ and $n_t = 1000$. The reported loss for the starting model is the test loss calculated using $\beta_t = 25$ and $n_t = 500$.

Remarkable is the rate at which the test loss blows up to much higher values than 0.480 when starting hard training starts. Within a few tens of iterations, the error shoots up to >1.0 loss!

Mixed Distillation: Both Soft & Hard labels

$T_{transfer}$	Test (avg.)	Test (min)	Training Loss (Hard)	Training Loss (Soft)
2	0.490	0.480	0.268	0.167
5	0.505	0.481	0.267	0.239
10	0.506	0.480	0.268	0.305
15	0.507	0.479	0.270	0.326
20	0.507	0.480	0.271	0.334

Table 5. Network performance of the 2-Unet guided by both soft and hard labels. The teacher model is 4-Unet trained for 300,000 iterations. The averaged test loss has the averaging parameters $\beta_t = 25$ and $n_t = 500$. The training soft losses are reported after multiplying by $T^2_{transfer}$.

Generalization Error!

Solution: Increase regularization with Batch Normalization layer and Class weights

Working with Batch Normalization layer

Sanity Check

Batch Normalization Paper : Feb 2015
U-net paper : May 2015

AHA!

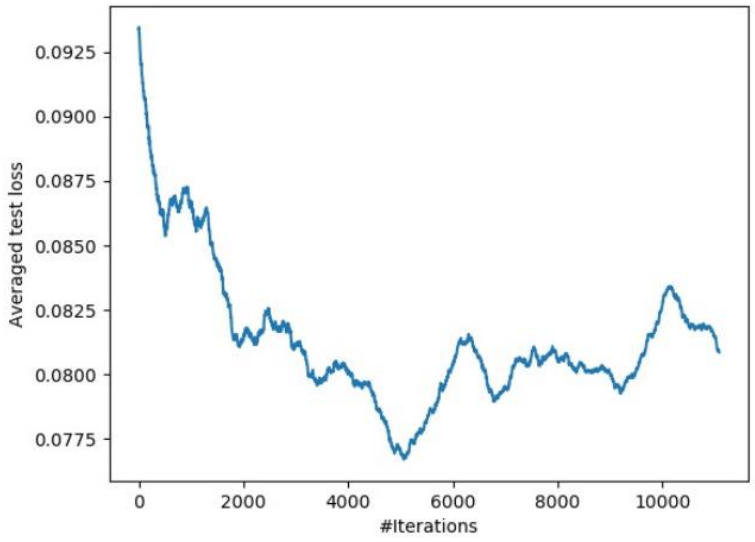


Fig. 4. The test loss curve for training the 4-Unet with the BN layer for 3.6 million iterations. The averaging setting for the reported curve is $\beta_t = 500$ and $n_t = 300$ to allow a smoother graph. The lowest test loss achieved is 0.07660 near 1,600,000 training iterations and takes around 2.2 days to train on a single Titan X GPU.

Vanilla Training 4-Unet from scratch

Takes **5x longer training time**
300,000 iteration vs.1.5 million iterations

Improves cross entropy performance
by **over 22%** .
0.0930 vs. 0.0727

Making It Work: Batch Normalization & Class weights

Teacher model : 64-Unet (31,042,434 parameters) [without BN]
Student model : 2-Unet (30,902 parameters) [with BN]

Soft Training
Cross entropy loss : 0.134
Intersection Over Union score: 0.752
100,00 iterations

Mixed Distillation
Cross entropy loss: 0.135
Intersection Over Union score: 0.759
155,000 iterations

Overall similar performance with or without hard labels

100x Compression!

Conclusion

Apart from demonstrating the distillation procedure to produce over 100x compression, we Also studies some interesting properties that can improve the architecture in different way:

- The U-net model can be significantly compressed without any extra effort!
- The degradation in **"expected"** performance Unet.
- Faster convergence rate with teacher guided training
- The dynamics of sequential and parallel guidance of soft and hard labels
- Performance boost in vanilla training with BN layer


Thank You
For Your Attention