Inception U-Net: Ultrasound Nerve Segmentation

Peter James Bernante [1]
Maharishi University of Management
pjbernante@mum.edu

Abstract

We develop an algorithm that can detect a collection of nerve structures called Brachial Plexus (BP) (Brachial Plexus, 2017). Our algorithm, Inception U-Net, is a deep convolutional neural network trained on ultrasound images from (Kaggle, 2016). The images were manually annotated by humans who were trained by experts. We used a modified U-Net (Ronneberger et al., 2015) architecture, and replaced its VGG-like (Simonyan & Zisserman, 2014) layers with Inception Modules (Szegedy et al., 2015). We find that the Inception Modules made the model deeper and significantly reduced the number of parameters. We also found out that, although the number of parameters is only a fraction of the standard U-Net, it had better performance in terms of defined metrics. Our algorithm was also able to annotate the images where the human annotators missed out.

1. Introduction

Surgery oftentimes involves post-surgical pain. Managing pain involves the use of narcotics which have several unwanted side effects; under or overdosing respiratory side effects and sedation.

One way to manage pain with less dependency on narcotics is through the use of indwelling catheters that deliver anesthetic. Pain management catheters block or mitigate the pain at the source. These catheters are inserted in the area around the nerves that carries sensation from the surgical site. It is therefore imperative to accurately identify nerve structures in order to effectively insert the catheters.

The goal of this project is to identify a collection of nerve structures called the Brachial Plexus (BP) . Given an ultrasound image, highlight or annotate the area in the image where the BP is located.

We will apply deep learning in computer vision to recognize the BPs in ultrasound images. In doing so, the following tasks will be involved:

1. Identify a collection of nerve structures called Brachial Plexus.
2. Given an ultrasound image, annotate the area in the image where BP is present
3. Design a deep neural network to accomplish the task.

The generated trained model can be useful in several ways. For example, it can be integrated with ultrasound apparatus to automatically show the area of interest.

2. Problem Formulation

The nerve identification task is a classification problem, where the input is an ultrasound image and the output is a binary label $t \in \{0, 1\}$. We want to classify each pixel in an ultrasound image as to whether it belongs to the BP (positive class) or not (negative class).

For this binary classification project, it is important to classify all positive classes, and it is also important that all positively identified classes are correct. F_1 score , where both precision and recall are considered, would be appropriate as the metric of choice to measure the performance of our model. The general formula for an $F - measure$ in terms of Type I and Type II errors is given as:

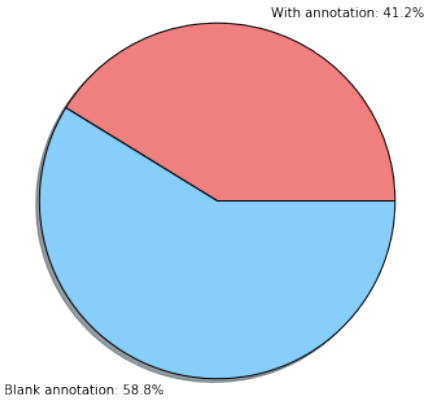
$$F_\beta = \frac{(1 + \beta^2) \cdot P_{true}}{(1 + \beta^2) \cdot P_{true} + \beta^2 \cdot N_{false} + P_{false}}$$

[1]Computer Science Department, Maharishi University of Management, Iowa, USA. Correspondence to: Peter James Bernante <pjbernante@mum.edu>.

where:

$$P_{true} = True\ Positive$$
$$N_{false} = False\ Negative$$
$$P_{false} = False\ Positive$$
$$\beta = 1$$

(a) Classes are severely imbalanced



(b) Majority of the images has blank annotations (i.e. no annotated brachial plexus)

Figure 1. Distribution of classes

3. Data Exploration

There are 5,635 images for training and 5,508 images for testing. The images are gray scale with dimensions 580 x 420 pixels and are noisy. Training images have masks to indicate where the BP is present, while there is none for testing images. In the training images, only 2,323 images have positively identified the BP. Groups of images are taken from the same patient. Images that come from the same patient are highly correlated (see Figure 2).

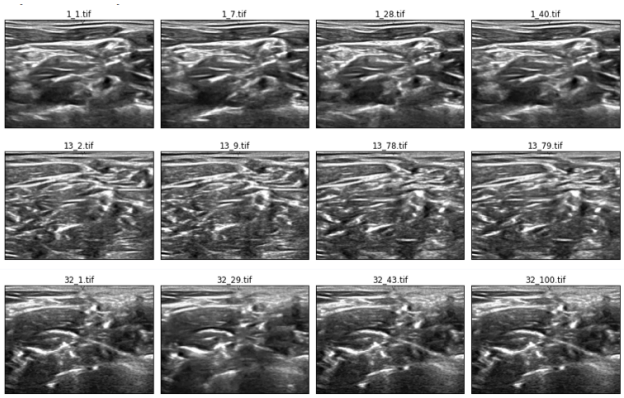


Figure 2. High correlation of images. The file names have the format <patient_id_xxx.tif>. Images coming from the same patient ID are highly correlated. (NOTE: The images are not exactly the same.)

The classes are severely imbalanced; consisting of 1.2% positive class and 98.8% negative class (see Figure 1. This class imbalance is mitigated by using F_1 score.

There are inaccuracies in the annotations of the BP. There is no prefect ground truth or gold standard (see Figure 3). There are very similar images but with conflicting annotations; one image has positively identified the BP, however the other has none (see Figure 4). There are also very similar images that have positive annotations, but the area where they are annotated differ in shape/area, although the annotations are located approximately in the same region (see Figure 5).

Where the BP is present in an image, the BP annotations has the following characteristics:

Minimum size	2,684 pixels
Maximum size	17,439 pixels
Average size	7,125.74 pixels

4. Data Preprocessing

4.1. Filtering

As shown in Figure 4, several images are very similar but have conflicting masks. This will greatly negatively affect the learning process. To mitigate the negative impact, images without masks, but with very similar images that have masks, are filtered out (images without masks and have no similar images are retained). The reason for dropping these images is, they most likely have similar annotation with their annotated counterpart, and were overlooked during manual annotations (it is more likely to miss to annotate an

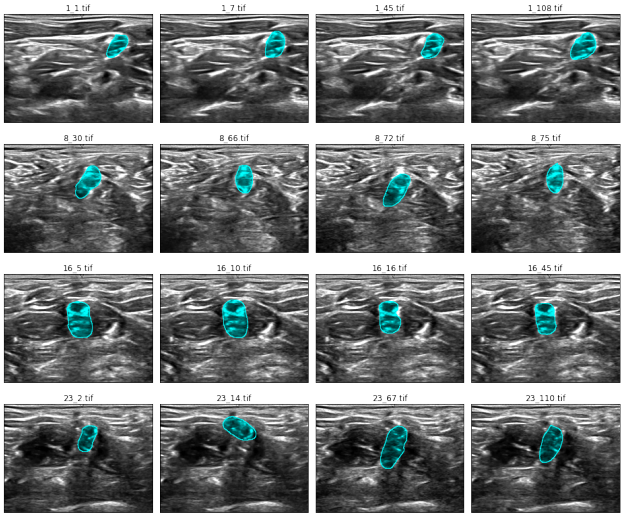


Figure 3. Similar images with varying annotations. Human-annotated training images have inaccurate annotations.

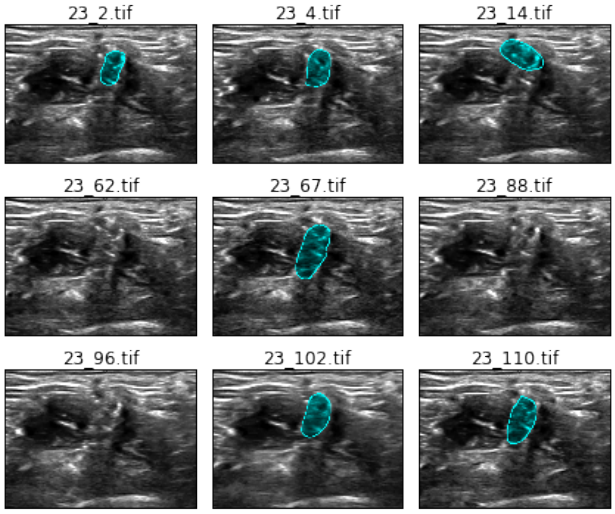


Figure 4. Conflicting annotations. Very similar images have conflicting annotations. One image has BP while the other has none. These are human errors during manual annotation of the dataset.

image than accidentally annotating random location of the image).

For similar images with varying annotations (in terms of shapes and sizes), they mostly cover the same general area of the image, and have common intersections. These images are retained for training to force the model to figure out what is common between the annotations.

The similarity of the images are measured using normalized cross-correlation coefficient (Edwards, 1976) with a threshold of 0.8.

4.2. Resizing Images

The ultrasound images are large files and have a lot of noise. To reduce the noise, the images are downsized to 96 x 128 pixels using inter-area interpolation (Kyriakidis, 2004). Downsizing also significantly reduces the dimensionality of the dataset, which will also improve the training time.

4.3. Splitting

The dataset is split to 80% train and 20% validation sets. The split is stratified based on the presence of mask.

4.4. Normalization

To have better training performance, the dataset is centered at zero mean and normalized to unit variance. This was done by subtracting the mean and then dividing by the standard deviation of the dataset.

5. Model Architecture

In this semantic image segmentation problem, a deep neural network will be used to predict the labels for each pixel in the image. Convolutional networks (Lecun et al., 1998), from which most state of the art image recognition techniques are derived, will used as the building block of our architecture.

In our dataset, the pixels are roughly similar all throughout the image. There is no clear groupings or sharp boundaries in which an untrained eye can clearly identify the brachial plexus.

The U-Net architecture (Ronneberger et al., 2015) uses VGG-like layers (Simonyan & Zisserman, 2014), where a series of convolutional layers are place on top of each other, and gradually decreasing the grid size while increasing the depth. The convolutions are then gradually reversed until the layer is back to its original size. This forms a U-like structure, from which the architecture got its name.

Marko Jocic (Jocić, 2016) has demonstrated the use of U-Net on the same dataset. Our architecture improves U-Net by replacing the VGG-like layers with Inception Modules (Szegedy et al., 2015) (See Figure 6). All layers in the middle uses Inception V3 and Grid Reduction modules instead of VGG and Max-Pooling. Inception V4 is used in the coarsest layer

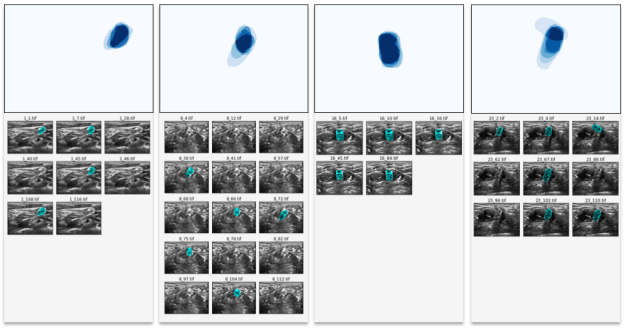


Figure 5. Average annotations of similar images. Similar images have similar annotations, if present. The annotations do not exactly have the same shape, but they cover the same general area of the image and have common intersections.

at the bottom. Transpose Convolution (Dumoulin & Visin, 2016) is used to upsample the layer to the same size as the input. The Inception modules made the model so deep and results to vanishing gradients. To solve this, SELU (Klambauer et al., 2017) activations are used which make it impossible to have vanishing or exploding gradients. Sigmoid was used at the last layer for binary classification (see Table 1).

	U-Net	Inception U-Net
Parameters	7,759,521[1]	1,846,353[1]
File size	93 MB[1]	22.7 MB[1]
Layer	VGG-like	Inception Module
Pooling	Max pooling	Reduction Module
Activations	ReLU	SELU

Table 1. Comparison of U-Net and Inception U-Net.

6. Training

6.1. Loss Function

We want our output to produce similar mask as the labels. This entails a measure of similarity. Dice Coefficient (Sørensen, 1948) is a static used for comparing the similarity of two samples, and is given with the formula:

$$QS = \frac{2|X \bigcap Y|}{|X| + |Y|}$$

It was originally intended to be applied to presence/absence of data. QS is the quotient of similarity and ranges between 0 and 1, where 1.0 means the two samples are exactly the same. It can be viewed as a similarity measure over sets.

The loss function minimizes, so we take the negative Dice Coefficient as our loss function.

6.2. Training Configuration

Training the model was run with the following configuration:

Optimizer	Adam
Learning rate	0.0001
Learning rate decay	0.001
Loss Function	- Dice Coefficient
Batch size	32
Epochs	120

The weights of the network were randomly initialized and trained end-to-end using Adam optimizer (Kingma & Ba, 2014) default values for hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate was set to 0.0001 with a decay of 0.001 for every iteration.

With limited training dataset, data augmentation was applied. Artificial images were generated on-the-fly which consists of transformation of the images including horizontal and vertical shifts, rotation between ± 30 degrees, up to 10% zoom, and horizontal and vertical flips.

We use Keras (Chollet et al., 2015) to run the training with TesnsorFlow (Abadi et al., 2015) backend.

6.3. Evaluation and Validation

The learning rate that produced reasonable result is around 0.0001. Higher learning rate converged to higher loss value while with lower learning the training was not able to converge after several epochs. Exponential learning rate decay of 0.001, decayed at every iteration, helped stabilized the loss convergence at the later stage of training.

he training and validation loss were very close to each either which indicates that the model is neither overfitting nor has high variance (See Figure 7).

The F_1 score also shows a result that is in agreement with the loss (See Figure 8).

The final model was chosen using early stopping. The checkpoint with highest validation score was chosen as the final model. The validation set uses images that were not included during training. It generally has lower score compared to training, which reflects more accurate performance metric of the model.

[1]Based on 96 x 128 input shape

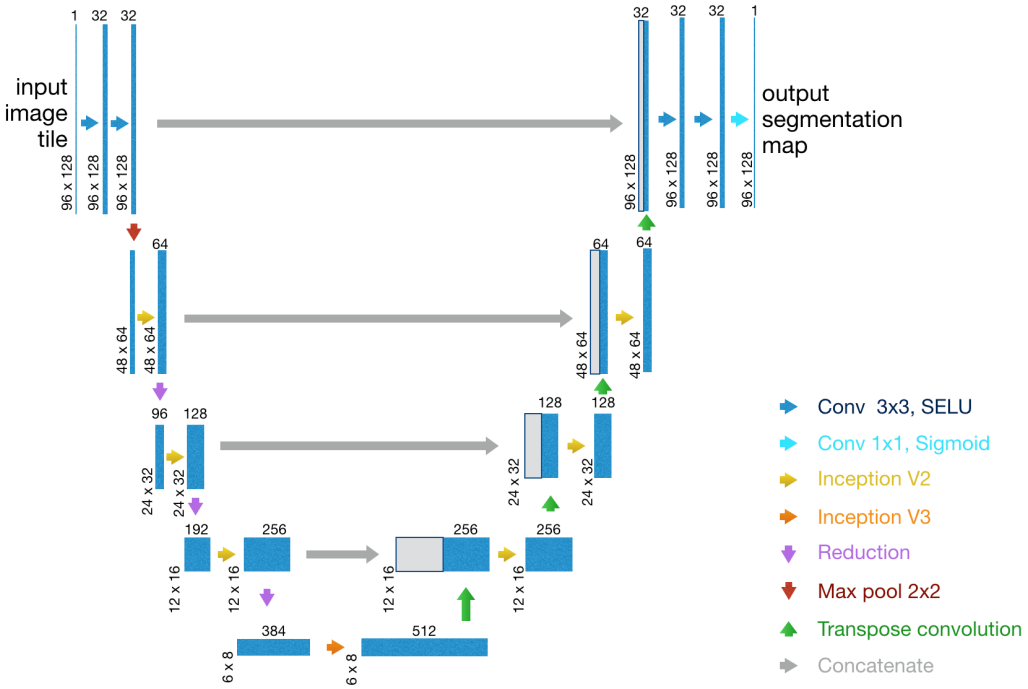


Figure 6. Inception U-Net Architecture. All layers in the middle uses Inception V3 and Grid Reduction modules instead of VGG and MaxPooling. Inception V4 is used in the coarsest layer at the bottom. Transpose Convolution is used to upsample the layer to the same size as the input. All Layers used SELU activation, with the exception of the last layer where Sigmoid was used for binary classification.

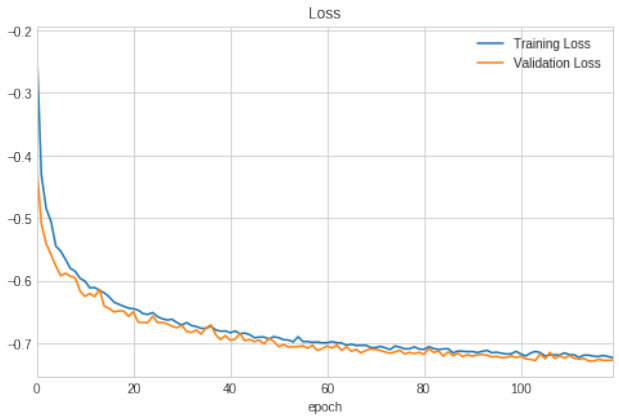


Figure 7. Training and Validation Loss

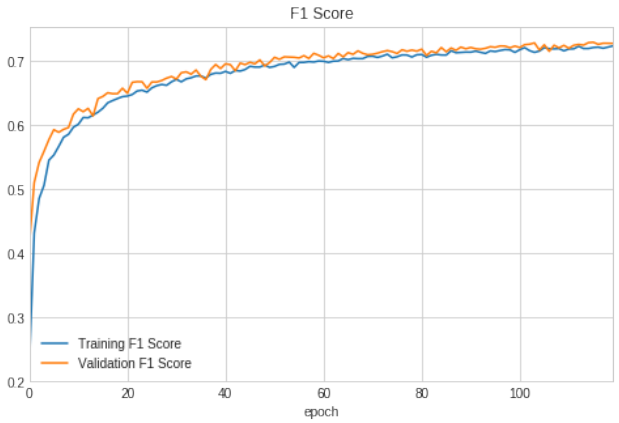
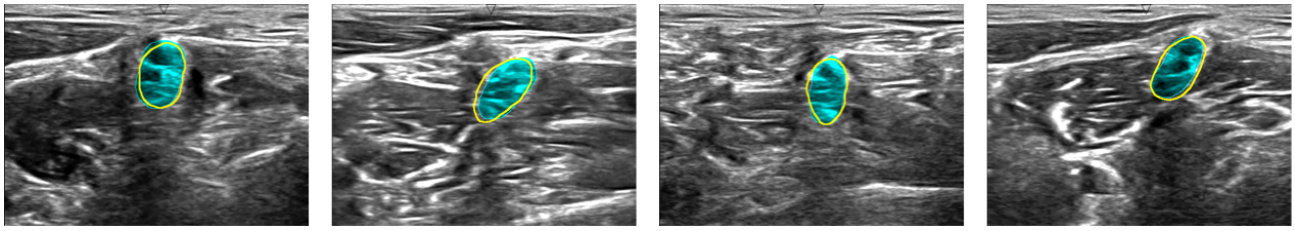
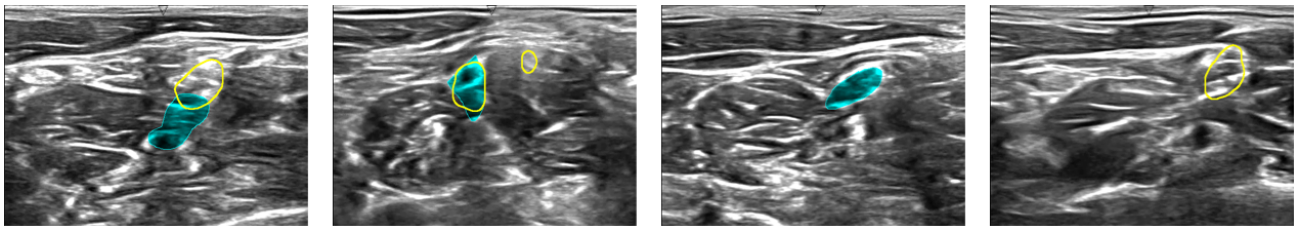


Figure 8. Training and Validation F_1 Scores

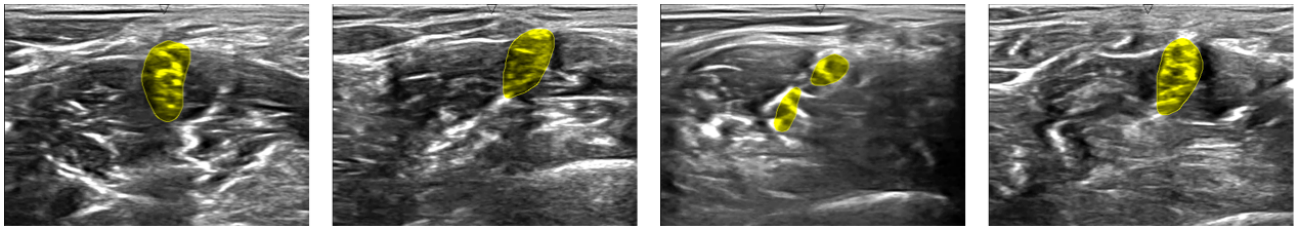
To verify the robustness of the model, inference was run using the test dataset. This dataset does not have accompanying ground truth that can be compared with to produce a quantitative measure. It is rather difficult for untrained eyes to make a judgment as to whether or not the predicted annotations are accurate. As a non-medical professional, we can look over and again the training set and pick up general pattern of nerve structures. However, this would still be not very reliable and consulting with a trained medical professional would be best to verify the result. Some samples predictions using the test dataset are shown in Figure 9c for qualitative assessment.



(a) Good performance of the model. The area highlighted in cyan is the ground truth while the area outlined in yellow is the prediction of the model. The model predictions are almost identical to the ground truth.



(b) Missed out predictions. From left to right: (a) the model prediction mostly missed the ground truth, (b) the model hit the ground truth but also annotated an extra area, (c) the model was not able to detect brachial plexus but there actually exists, and (d) the model was able to find brachial plexus where human annotator missed out.



(c) Inference on test dataset. The yellow highlights are the prediction of the model. The test dataset has no accompanying ground truth to compare the results with. Consultation from trained ultrasound professional is needed.

Figure 9. Final Model Output

7. Result

Some images from validation and test set that were not used in training were evaluated. It can be seen that the model performs well on most images. However, there are also cases where it misses. The area highlighted in cyan is the ground truth while the area outlined in yellow is the prediction produced by the model (See Figure 9).

8. Inception U-Net vs. Other Architectures

Following the previous work on Ultra Sound Nerve Segmentation (Jocić, 2016), we run our model against the test dataset. We compare our results from running single model, without any ensemble. Table 2 illustrates the dice coefficients of various architectures on the test dataset.

Architecture	Dice Coefficient
U-Net	0.57
Image Pyramid	0.58
Inception U-Net	0.62

Table 2. Inception U-Net outperforms the standard U-Net architecture

9. Conclusion

There is no doubt the role of deep learning in advancing the technologies used in the modern medical industry is getting more significant over the years. Ultrasound imaging is only one of the many medical applications. This project is only a small step, but it shows a great potential to what it can accomplish.

Given that the training data has a good portion of inaccuracies, the predictions of the model is surprisingly good that it can point the location of nerve structures, or the absence thereof. The predicted area of the im-

age shows the general location of the brachial plexus, very much like how the human annotators did.

It is interesting to note that the machine learning model was able to identify nerve structures after training within hours, while an untrained person would not be able to figure out the pattern from the same set of images.

Significant improvement on inference time would be very useful to make the model run on a real time application in video input, or run natively in less powerful devices such as on mobile platform.

10. Acknowledgements

We would like to acknowledge Emdad Khan, Phd. for is support and for sharing his technical expertise that allowed us to complete this project.

References

Abadi, Martín, Agarwal, Ashish, Barham, Paul, Brevdo, Eugene, Chen, Zhifeng, Citro, Craig, Corrado, Greg S., Davis, Andy, Dean, Jeffrey, Devin, Matthieu, Ghemawat, Sanjay, Goodfellow, Ian, Harp, Andrew, Irving, Geoffrey, Isard, Michael, Jia, Yangqing, Jozefowicz, Rafal, Kaiser, Lukasz, Kudlur, Manjunath, Levenberg, Josh, Mané, Dan, Monga, Rajat, Moore, Sherry, Murray, Derek, Olah, Chris, Schuster, Mike, Shlens, Jonathon, Steiner, Benoit, Sutskever, Ilya, Talwar, Kunal, Tucker, Paul, Vanhoucke, Vincent, Vasudevan, Vijay, Viégas, Fernanda, Vinyals, Oriol, Warden, Pete, Wattenberg, Martin, Wicke, Martin, Yu, Yuan, and Zheng, Xiaoqiang. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Brachial Plexus. Brachial plexus — Wikipedia, the free encyclopedia, 2017. URL https://en.wikipedia.org/wiki/Brachial_plexus.

Chollet, François et al. Keras. https://github.com/fchollet/keras, 2015.

Dumoulin, V. and Visin, F. A guide to convolution arithmetic for deep learning. *ArXiv e-prints*, March 2016.

Edwards, A. L. *An Introduction to Linear Regression and Correlation*, chapter 4, pp. 33–46. W. H. Freeman, San Francisco, CA, 1976.

Jocić, Marko. Deep learning tutorial for kaggle ultrasound nerve segmentation competition, using keras. https://github.com/jocicmarko/ultrasound-nerve-segmentation, 2016.

Kaggle. Ultrasound nerve segmentation, 2016. data retrieved from World Development Indicators, https://www.kaggle.com/c/ultrasound-nerve-segmentation.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. *ArXiv e-prints*, December 2014.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. Self-Normalizing Neural Networks. *ArXiv e-prints*, June 2017.

Kyriakidis, Phaedon C. A geostatistical framework for area-to-point spatial interpolation. *Geographical Analysis*, 36(3):259–289, 2004. ISSN 1538-4632. doi: 10.1111/j.1538-4632.2004.tb01135.x. URL http://dx.doi.org/10.1111/j.1538-4632.2004.tb01135.x.

Lecun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pp. 2278–2324, 1998.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *ArXiv e-prints*, May 2015.

Simonyan, K. and Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv e-prints*, September 2014.

Sørensen, T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.*, 5:1–34, 1948.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the Inception Architecture for Computer Vision. *ArXiv e-prints*, December 2015.