Nuclei Segmentation Using Deep Neural Network Architectures

Final Project Report: Deep Learning Systems

Rashmi Bidanta
MS Computer Science, SICE
Indiana University
Bloomington, Indiana, USA
rbidanta@iu.edu

Syam Sundar Herle
MS Data Science, SICE
Indiana University
Bloomington, Indiana, USA
syampara@iu.edu

Vighnesh Nayak
MS Data Science, SICE
Indiana University
Bloomington, Indiana, USA
vrnayak@iu.edu

ABSTRACT

In this paper we explore the application of Deep Neural Network (DNN) models for spotting nuclei in microscopic biomedical images. Nuclei spotting is the key in detecting the root cause of various diseases. People suffer from various kinds of diseases like Cancer, heart disease, respiratory diseases, chronic obstructive pulmonary disease, Alzheimer, diabetes and the list goes on. Detecting such diseases at an early stage is an important step towards their cure. In this work we evaluate the efficiency of three Deep Neural Network models namely UNET, SEGNET and FCN. In the end we tried to create an ensemble architecture of both UNET and SEGNET we named it U-SEGNET and observed that the nuclei mask predictions were slightly better than both U-NET and SEGNET models.

CCS CONCEPTS

• **Computing methodologies → Neural networks**; *Image segmentation*; *Object detection*; *Supervised learning by regression*; *Batch learning*;

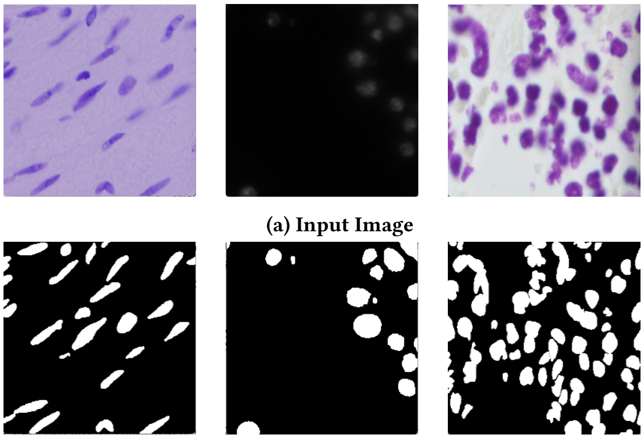
KEYWORDS

Deep Learning, Nuclei Segmentation, Microscopic Images, Neural Network, Convolution Neural Network

1 INTRODUCTION

Spotting nuclei is very important in diagnosis of various critical diseases such as Cancer, Alzheimer's, Parkinson's, Respiratory, Heart and Pulmonary diseases. According to Center for Disease Control and Prevention's (CDC) National Vital Statistics Reports for 2015, close to 2 million deaths[9] were due to diseases that could have been cured if they were diagnosed early at the onset of such diseases. Typically a human body consists of 30 to 40 trillion cells[17] and each cell contains nucleus full of DNA. They all vary in their shape, sizes and posses different physiological properties. This makes them behave differently under disparate conditions. Identifying such nuclei helps researchers and medical practitioners to analyze cells and understand their reaction to various kind of treatments. Traditionally these diagnosis are done by the pathologists who manually analyze the glass slides under electron microscope [15]. This manual analysis is time consuming and is an impediment to the faster disease diagnosis. Automatic nuclei analysis could provide insights to disease mechanism which could not be gleaned from manual qualitative evaluations of cell and molecular specimens.

In response to the above problem statement we decided to leverage the power of Deep Neural Networks[16] to train neural models to spot nuclei in the microscopic images.In this work we evaluate Deep Neural Networks that have been proposed for semantic segmentation of images. For the purpose of this work we have taken the dataset from Data Science Bowl 2018 Kaggle competition [3]. The Figure 1 shows few example training images and corresponding ground truth images.We evaluated this dataset against the neural network models that use Convolution Neural Networks[7] such as U-NET[12], SEGNET[1], FCN (32,16 and 8) [8]. We conclude this work with an ensemble of UNET and SEGNET architectures forming U-SEGNET and trained this ensemble architecture for nucleus prediction, the details of this architecture is explained in section 2.


(a) Input Image

(b) Ground Truth

Figure 1: Sample Training Images and Ground Truth Images (a) Input Image and (b) Corresponding Target Masks.

In section 2 we explain briefly about the features of network models that we evaluated as part of this study[1, 8, 12]. In section 3 we explain about the experiments we performed using the various hyper parameters and tweaks to the network models, we also explain our approach of data processing and augmentation before training the respective models. The section also talks about the modifications to these networks to understand the resulting response to these changes. In section 4 we discuss the results of our experiments, we found that U-NET model seemed to be doing slightly better than SEGNET achieving a mean Intersection Over Union (IoU) of around 84.07%, for validation data set, the SEGNET model converged to 82.21% mean IoU on this nuclei dataset. The ensemble model U-SEGNET did better than both SEGNET and U-NET model with mean IoU of 84.45% at learning rate $1e-3$. As far as FCN model is concerned we were not able to get a convergence.

However, these models did seem to learn the prediction masks that captured the nucleus location but the spatial representation seemed out of place. In section 5 we conclude this work and set goals for the future work on the application of DNNs for semantic segmentation of biomedical images.

2 DEEP NEURAL NETWORKS ARCHITECTURES

In this section we elaborate on the various models we evaluated. We discuss our observation on the models and then we go on to discuss the ensemble model that we came up by merging U-NET and SEGNET which gave slight overall improvement in the prediction of nuclei.

2.1 U-NET

U-NET[12] model was proposed in the year 2015 specifically for the biomedical images. It is a CNN based model with a contracting path that learns the dense features of the image and these dense features are then up-sampled using transposed convolution which increased the resolution and helped in localizing the intricate features. The model was originally trained on a very low training dataset of 30 images which the authors had augmented using elastic deformation[11]. We however, did not perform any elastic deformation on our dataset. The paper does not mention about the hyper parameters they used such as the number of epochs, learning rate, if they used any batch normalization[6] or not, did they use any regularization technique such as dropout[14] or not. We used standard batch normalization with momentum of 10% and also added dropout of 20% before applying ReLu[10] activation in each convolution layer. Applying dropout to the network improved the results of the existing model. Another observation was that, the original model did not maintain the size of input image in its output segmentation map. We applied padding to maintain the size of the the output segmentation maps similar to the input images.

2.2 SEGNET

SEGNET model was proposed in the year 2016, a deep fully convolution neural network architecture for pixel-wise semantic segmentation. This network consists of an encoder branch and a decoder branch followed by softmax layer, which helps in pixel wise classification. The SEGNET consists of 13 layers of encoder layers, identical to VGG16, followed by the same number of decoder layers which map the low level encoder feature maps to full input resolution feature maps for pixel-wise classification. The decoder layer uses pooling indices from max-pooling step of corresponding encoder to up-sample the feature maps, then the sparse up-sampled maps are convolved with trainable filters to produce dense feature maps. As our problem dealt with two classes, nuclei or non-nuclei we used sigmoid layer instead of softmax layer. After applying filter in encoder, batch normalization and ReLu[10] was applied over the feature maps, but no information about number of epochs or learning rate were given in the paper. For our problem, we did not use any pre-trained model and we trained SEGNET architecture with different parameters and selected the one which gave promising result for our problem.

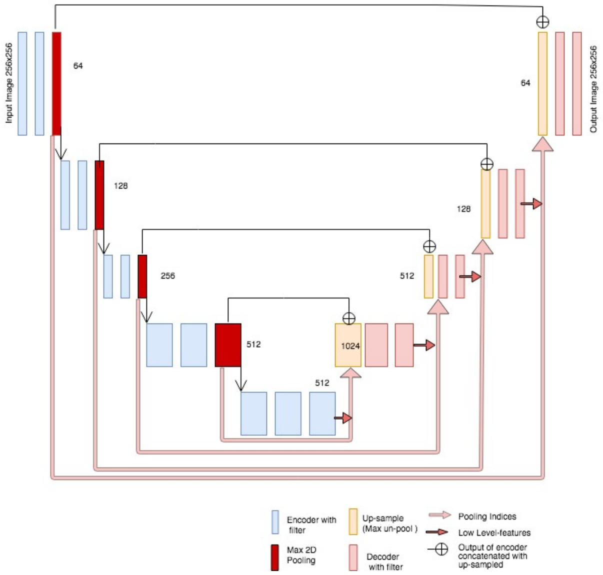


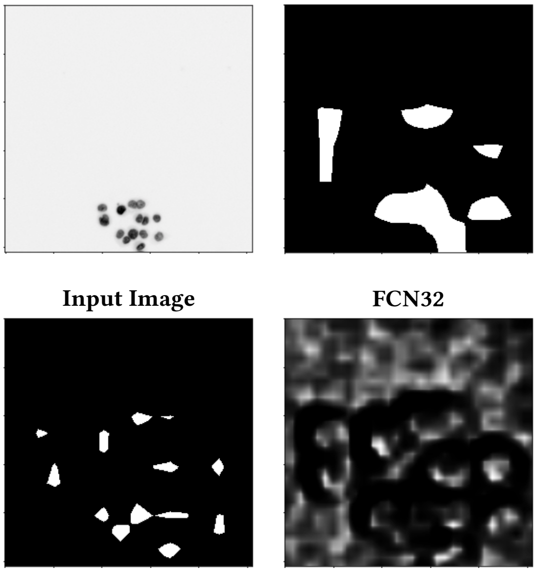
Figure 2: U-SEGNET Architecture: Ensemble of U-NET and SEGNET.

2.3 U-SEGNET

U-SEGNET (Term coined by us for fun) is an ensemble model that we came up by combining the U-NET and SEGNET architectures. The figure 2 shows the architecture of U-SEGNET. As the overview of both the models in terms of encoder and decoder branch were similar except for up-sampling and deconvolution technique, we tried to mix up both the models. We used 13 layers of encoder similar to VGG-16 and 13 layers of decoder, batch normalization and ReLu activations were applied on features after applying filters in encoder branch. Some of the changes we tried to come up when combining both models were like adding three layers of encoder with batch normalization and ReLu for each encoder, in between the encoder and decoder branch. In U-SEGNET the decoder layer uses pooling indices from corresponding max-pool step of encoder layer to up-sample the low level feature maps, then the sparse feature maps are concatenated with the input feature maps of the corresponding pooling step of encoder layer. The final concatenated features are convolved with trainable filters to produce dense feature maps. For this architecture, we have used batch normalization momentum of 0.99 for the each encoder layer.

2.4 FCN32, FCN16, FCN8

FCN[8] is a pioneering paper published in 2014 and was a breakthrough for semantic segmentation of images. This model was trained on PASCAL VOC 2011 and 2012 [2] which is a very different kind of dataset where images are generally things we encounter in our day to day life. The model used pre-trained VGG16[13] for transfer learning[18]. Image localization was achieved through single up-sampling at the output convolution layer for FCN32. In-case of FCN16 the up-sampled features maps from fourth layer of VGG16 net was added to the output features and then up-sampled to get the masks. FCN8 did the same for the feature maps learned in the third layer of the VGG16 network. Essentially, for all these versions of FCN there is only one up-sampling layer for output maps. We tried to put FCN to work on Data Science Bowl 2018 dataset and the results were not satisfying. The network seemed to learn the dense features but was not able to localize the nuclei positions. The figure 3 shows the input images and their corresponding masks learned. Training FCN model was very difficult with the hyper parameters that the original paper had mentioned. Initially we tried to train the model with the pre-trained VGG16 model but the model was not training well. Then we decided to use vanilla VGG16 model and tried the default hyper parameters, but the model did not converge. The default hyper parameters were modified to get the loss to converge, but the learned masks were not desirable. Because of this behavior we were not able to perform any significant evaluation for the FCN architecture.



Input Image **FCN32**

FCN16 **FCN8**

Figure 3: Masks obtained from FCN Architecture

3 EXPERIMENTS

3.1 Data Processing

We collected the data from Booz Allen Hamilton Data Science Bowl 2018. The dataset consisted of 670 training images and 65 testing 3 channel images with varying dimensions. The lowest resolution image was 256 × 256 and the largest resolution of images was 1024 × 1024. We converted all the images to gray scale and reduced the dimension of the images to the lowest available resolution which was 256 × 256. Majority of the images were dark background and light nuclei in the foreground, however there are few images with lighter background and dark nuclei in the foreground, which created non uniformity in the dataset. In order to overcome this we calculated the ratio of light pixel in the images. A pixel with intensity more than 127 was considered as light pixel. We came up with a thresholding procedure to solve this problem. The threshold T for an image of dimension $H \times W$ and P_{rc} as the pixel value for row r and column c was calculated using the following expression,

$$T = \frac{1}{H \times W} \sum_{r=0}^{H-1} \sum_{c=0}^{W-1} f(P_{rc})$$

$$Where, f(P_{rc}) = \begin{cases} 0 & P_{rc} < 127 \\ 1 & P_{rc} \geq 127 \end{cases}$$

For the current data set, if converted to gray scale, we found that an image whose T threshold was more than 0.25 is in fact an image with light background with dark nuclei. With this assumption in place we converted all such images to their negative scale both for training as well as for testing images. For the purpose of this study we split the training images into 600 training and 70 validation images. The data was augmented lazily during the training process by randomly applying horizontal, vertical and mirror transformation to the image and its corresponding ground truth mask simultaneously. We then trained the models discussed in previous section with this dataset.

3.2 Model Training

The processed input images and their corresponding nuclei masks were used for training the networks. We chose Stochastic Gradient Descent as optimization for all the networks. The code was written using PyTorch. We used PyTorch's binary cross entropy loss(BCE) as our energy function. The BCE loss is defined as,

$$L(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} Y_{ij} log(P_{ij})$$

Where, i indexes image samples/observations and j indexes classes in our case it is 2, and y is the sample label and $P_{ij} \in (0, 1)\colon \sum_j P_{ij} = 1 \quad \forall i, j$ is the prediction for a sample.

As the problem at hand was a two class problem we used sigmoid of the output from the model and then used that as input to the loss function along with the target mask. The original papers took a softmax on the output but we had to use sigmoid since each pixel would either be a nuclei or non-nuclei.

Mean Intersection over Union (IoU) other wise also known as Jaccard index as the evaluation criteria because most of the previous work on semantic segmentation[1, 4, 5, 12] use mean IoU as the primary evaluation criteria. The IoU is the ratio of intersection of nuclei pixels to the union of nuclei pixels for predicted mask and the ground truth masks. Mathematically IoU for a predicted mask P and ground truth mask G for an input image is represented as:

$$IoU = \frac{P \cap G}{P \cup G}$$

We tried various hyper parameter combinations to train the models like learning rate, momentum, batch normalization and dropouts. Initially we tried with a learning rate of 1e−2 which did not give any convergence on any of the model. We then tried learning rate of 1e−3 and 1e−4 the we were able to get convergence on all the three models. The figure 4 shows the comparison of learning

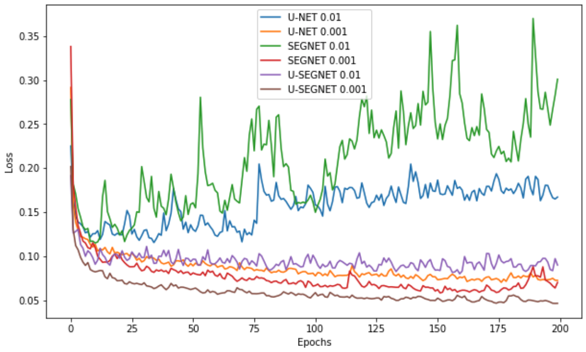


Figure 4: Loss convergence comparison for the models at various learning rates.

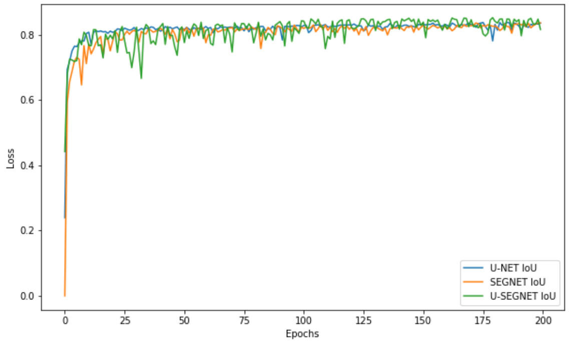


Figure 5: Convergence of the Mean IoU on Validation dataset.

rate[1] impact on loss convergence. Some other hyper parameters we used were adaptive learning rate with weight decay of 2.5e−4. We applied a momentum of 0.99 for the stochastic gradient descent which gave us faster convergence. At first we did not use any batch to train our model essentially we were doing 600 backward propagation per epoch which was taking close to three hours on Tesla V100-SXM2 to train the model for 200[2] epochs. We then tried to train the model on batch sizes of two and four images and found that the results were better while the model was trained on four images per batch. Subsequently we decided to use a batch size of four images per batch which reduced the training time considerably to about one hour for same number of epochs. One interesting fact we learned was that mini-batching could potentially help in getting better results. The figure 5 shows the convergence on accuracy for the validation dataset. The ensemble model seems to be getting slightly more accurate results than other two models.

4 RESULTS

Different network architecture outlined in section 2 were trained with different choices of learning rate for 200 epochs, the results of the experiment can be seen in 1, the table represents the mean Intersection over Union (IoU) comparison between the models. The pixel values for the prediction output map were all in the range of (0, 1) so we had to choose a threshold convert the pixel values to binary values of either 0 or 1. This was achieved by keeping a threshold of 0.5 so any output map which was less than 0.5 was converted to 0 and rest were converted to 1. These values were then multiplied by 255 to restore the actual pixel color (0 for black and 255 for white) .The table 2 shows the running time of each model per epoch. As we can see, SEGNET runs faster compared to other models since SEGNET uses only max pool indices to up-sample the low-level features.

Results		
Network	L.R = 0.001	L.R = 0.0001
U-NET	84.07%	82.17%
SEGNET	82.21%	82.36%
U-SEGNET	84.54%	84.48%

Table 1: Mean IoU for different learning rate

Network	Running time(lr=0.001)
U-NET	18 seconds
SEGNET	17.1 seconds
U-SEGNET	18.5 seconds

Table 2: Running time of models per epoch in seconds

The figure 6 shows the results of how well each of the above model predicts the nuclei for the data science bowl 2018. All the three models seems to be doing fairly well on the input images that are very clear and easy to decipher. However the predictions vary for the complex images with nuclei blobs having very difficult boundaries and overlapping areas. It is evident from the images that U-SEGNET does well in predicting the nuclei in images that are very hard and complex.

5 CONCLUSION AND FUTURE WORK

We evaluated various state of art deep convolutional neural network architecture for pixel wise semantic segmentation of medical images. We came up with an ensemble model from top performing models namely U-NET and SEGNET and named it U-SEGNET. We observed that there are a very few DNN models that were purposefully created for medical images. While models like SEGNET and FCN were primarily used for road and indoor scene understanding we wanted to put these to test on medical images and found that encoder and decoder based models perform strikingly better than the just a series of Convolution Neural Network as in case of FCN. There were a few other models we wanted to try as part of this study which includes MASK-RCNN[4] which was again trained on non-medical images. Another very interesting paper we came across was use of sparse autoencoder for unsupervised nucleus detection[5]. The author of this paper used Sparse

[1] The loss convergence graph for learning rate $1e-3$ and $1e-4$ were comparable hence the graph only shows the comparison of $1e-2$ and $1e-3$

[2] We saved the parameters of the model after every 10 epoch, so that if the final model ends up over-fitting we could recover the models that do not over-fit.

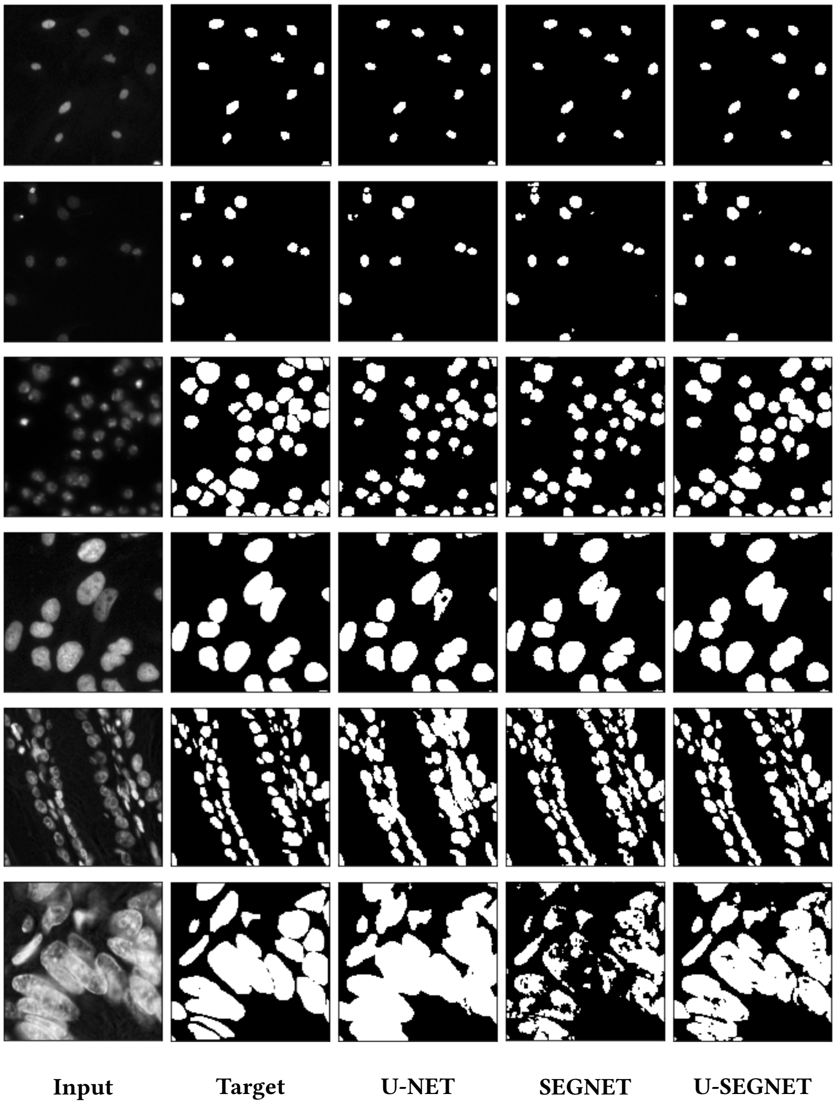


Figure 6: Comparison of Prediction Masks obtained from U-NET, SEGNET and U-SEGNET. The first column is the Input image, second column is the target mask and rest columns are the prediction masks for each model.

Convolution Autoencoder for nuclei detection we plan to use this model for nucleus segmentation which would be an extension to the original paper. We would also like to make a submission to this Kaggle competition The code for the work we performed on U-NET, SEGNET, U-SEGNET and FCN can be found in the GitHub link https://github.iu.edu/ENGR533/DSBowl2018.git

REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2015. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *CoRR* abs/1511.00561 (2015).

[2] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2015. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision* 111, 1 (Jan. 2015), 98–136.

[3] Booz Allen Hamilton. 2018 (accessed April 20, 2018). *Booz Allen Data Science Bowl 2018.* https://www.kaggle.com/c/data-science-bowl-2018

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. *CoRR* abs/1703.06870 (2017).

[5] Le Hou, Vu Nguyen, Dimitris Samaras, Tahsin M. Kurç, Yi Gao, Tianhao Zhao, and Joel H. Saltz. 2017. Sparse Autoencoder for Unsupervised Nucleus Detection and Representation in Histopathology Images. *CoRR* abs/1704.00406 (2017).

[6] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *CoRR* abs/1502.03167 (2015).

[7] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. 1999. *Object Recognition with Gradient-Based Learning.* Springer Berlin Heidelberg, Berlin, Heidelberg, 319–345. https://doi.org/10.1007/3-540-46805-6_19

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2014. Fully Convolutional Networks for Semantic Segmentation. *CoRR* abs/1411.4038 (2014).

[9] Sherry L. Murphy, Jiaquan Xu, Kenneth D. Kochanek, Sally C. Curtin, and Elizabeth Arias. 2017. National Vital Statistics Report, Deaths: Final Data for 2015. *CDC Report* abs/66.6 (2017).

[10] Vinod Nair and Geoffrey E. Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML'10).* Omnipress, USA,

807–814. http://dl.acm.org/citation.cfm?id=3104322.3104425

[11] Vadim Ratner and Yehoshua Zeevi. 2009. The dynamics of image processing viewed as damped elastic deformation. (01 2009).

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR* abs/1505.04597 (2015).

[13] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html

[15] Wikipedia. 2010. Electron microscope — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Electron_microscope [Online; accessed 20-April-2018].

[16] LeCun Y, Bengio Y, and Hinton G3. 2015. Deep Learning. *Nature* (2015). https://doi.org/doi:10.1038/nature14539

[17] Yella Hewings-Martin PhD. 2017. How many cells are in the human body? https://www.medicalnewstoday.com/articles/318342.php [Online; accessed 20-April-2018].

[18] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? *CoRR* abs/1411.1792 (2014).