REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

NUCLEI SEGMENTATION WITH GENERATIVE ADVERSARIAL NETWORKS

Onur Can KOYUN

MASTER OF SCIENCE THESIS
Department of Electronics and Communication Engineering
Electronics Engineering Program

Advisor
Prof. Dr. Tülay YILDIRIM

May, 2019

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

NUCLEI SEGMENTATION WITH GENERATIVE ADVERSARIAL NETWORKS

A thesis submitted by Onur Can KOYUN in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 09.05.2019 in Department of Electronics and Communication Engineering, Electronics Engineering Program.

Prof. Dr. Tülay YILDIRIM
Yildiz Technical University
Advisor

Approved By the Examining Committee

Prof. Dr. Tülay YILDIRIM, Advisor
Yildiz Technical University _____

Doç. Dr. Burcu Erkmen, Member
Yildiz Technical University _____

Prof. Dr. Zümray Dokur, Member
İstanbul Technical University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Nuclei Segmentation With Generative Adversarial Networks supervised by my supervisor, Prof. Dr. Tülay YILDIRIM. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Onur Can KOYUN

Signature

Dedicated to my family
and those who seek knowledge

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Prof. Dr. Tülay Yıldırım for the continuous support of my MSc study and related research, for her patience, motivation and knowledge. Her guidance helped me in all the time of research and writing of this thesis.

I would like to thank to my parents and to my sisters for supporting me spiritually throughout writing this thesis and my life in general.

Onur Can KOYUN

TABLE OF CONTENTS

LIST OF SYMBOLS vii

LIST OF ABBREVIATIONS viii

LIST OF FIGURES x

LIST OF TABLES xii

ABSTRACT xiii

ÖZET xv

1 Introduction 1
- 1.1 Literature Review . . . 2
- 1.2 Objective of the Thesis . . . 7
- 1.3 Hypothesis . . . 7

2 Deep Learning Models 8
- 2.1 Neural Networks . . . 9
- 2.2 Deep Models . . . 12
  - 2.2.1 Convolutional Neural Networks . . . 13
  - 2.2.2 Activation Functions, Normalization and Pooling Operations . . . 14
- 2.3 Generative Models . . . 21
  - 2.3.1 Generative Adversarial Networks . . . 21
  - 2.3.2 Wasserstein-GAN . . . 25
- 2.4 Cycle-Consistent Generative Adversarial Networks . . . 26

3 Common Deep Learning methods and Metrics for Nuclei Segmentation Problem 29
- 3.1 Patch Classification Approach . . . 29
- 3.2 Fully Convolutional Network Approach . . . 29
- 3.3 Performance Metrics for Nuclei Segmentation . . . 31
  - 3.3.1 Accuracy . . . 31
  - 3.3.2 Precision-Recall . . . 32

3.3.3 F1-Score 32
3.3.4 Intersection Over Union 33

4 Unsupervised Nuclei Segmentation using Generative Modeling 34
4.1 Domain Transfer for Unlabeled Nuclei Segmentation 35
4.1.1 Results 39
4.2 Unsupervised Nuclei Segmentation
Using Random Annotations 41
4.2.1 Random Nuclei Label Generation 43
4.2.2 Post Processing 44
4.2.3 Experiments and Results 45

5 Conclusion and Discussion 47

A Detailed Architectures 49

References 51

Publications From the Thesis 59

LIST OF SYMBOLS

θ	Weight Parameters Of Network
$J(\theta)$	Loss Function Over Current Weights
$\nabla J(\theta)$	Back-propagation Error Over Loss Function
$P_x(x)$	Probability Distribution of x
z	Latent Variables
$\mathbb{E}$	Expected Value
ϵ	Learning Rate
$\mathscr{L}$	Loss Function

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Network
DNN	Deep Neural Network
CNN	Convolutional Neural Network
FCN	Fully Convolutional Network
VAE	Variational Autoencoder
AE	Autoencoder
BM	Boltzmann Machine
GAN	Generative Adversarial Network
CycleGAN	Cycle Consistent Generative Adversarial Network
WGAN	Wasserstein Generative Adversarial Network
WGAN-GP	Wasserstein-GAN with gradient penalty
ELBO	Evidence Lower Bound
ADAM	Adaptive Moment Estimation
ReLU	Rectified Linear Unit
PReLU	Parametric Rectified Linear Unit
SELU	Scaled Exponential Linear Unit
ICS	Internal Covariate Shift
JS	Jensen-Shannon
JSD	Jensen-Shannon Divergence
KL	Kullback-Leibler
KLD	Kullback-Leibler Divergence
TV	Total Variation Distance

H&E	Hematoxylin and Eosin
IHC	Immunohistochemical
Gcuts	Graph Cuts
ACM	Active Contour Model
DoG	Difference of Gaussian
TP	True Positive
FP	False Positive
FN	False Negative
TN	True Negative
TPR	True Positive Rate
PPV	Positive Predictive Value
IOU	Intersection Over Union
mIOU	Mean Intersection Over Union

LIST OF FIGURES

Figure 2.1 A feed forward neural net. 10
Figure 2.2 Linear separability of logic operations 11
Figure 2.3 Sigmoid and Tangent hyperbolic activation functions 11
Figure 2.4 Convolution kernels of early layer [72] 13
Figure 2.5 An interpretation of Convolutional Neural Network 14
Figure 2.6 Convolution operation 15
Figure 2.7 ReLU Activation function and its variants 16
Figure 2.8 Inception module 17
Figure 2.9 VggNet Layer 18
Figure 2.10 A: VGG Layer, B: Residual Layer, C: Residual Layer with dimension reduction 19
Figure 2.11 Fully convolutional neural networks [26] 19
Figure 2.12 Detailed U-Net architecture [6] 20
Figure 2.13 Generative Adversarial Networks 22
Figure 2.14 Training process of Generative Adversarial Networks, A: Update of generator, B: Update of discriminator 23
Figure 2.15 KL Divergence between two Gaussian distributions. 24
Figure 2.16 Interpolating in the latent space z of GAN's generator network . . . 25
Figure 2.17 Paired image-to-image translation problem solved with Pix-to-pix [89] 27
Figure 2.18 Image to Image translation with CycleGAN [90] 27
Figure 3.1 Basic Patch Classifier 30
Figure 3.2 Basic Unet architecture 30
Figure 4.1 Domain transfer results on histology images 37
Figure 4.2 Segmentation results with generated data 39
Figure 4.3 Divergent samples and generated images 40
Figure 4.4 Generated random nuclei annotations 41
Figure 4.5 Diagram of CycleGAN for unsupervised segmentation 42
Figure 4.6 Diagram of Post Processing 44
Figure 4.7 Experiment results 44
Figure 4.8 Segmentation structure 45

Figure 4.9 Dilated Convolution 46

LIST OF TABLES

Table 2.1 Generator architecture 28
Table 2.2 Discirminator architecture 28
Table 4.1 Generator architecture 38
Table 4.2 Discriminator architecture 38
Table 4.3 Results 45
Table A.1 Generator Network: F 49
Table A.2 Generator Network: G 50
Table A.3 Discriminator Network 50
Table A.4 Residual Layer 50

ABSTRACT

Nuclei Segmentation With Generative Adversarial Networks

Onur Can KOYUN

Department of Electronics and Communication Engineering
Master of Science Thesis

Advisor: Prof. Dr. Tülay YILDIRIM

Since the discovery of the microscope, pathologists analyze stained tissue samples taken from patients which indicates various diseases and properties that helps to understand and cure the diseases. However, process of pathological analysis is very expensive and time consuming.Additionally, to interpret the information underlying histopathology images, pathologists need a great understanding of cell and tissue physiology. Also, there is observer variations. Different pathologists might reach different conclusions consequently their experience levels.

Fast, accurate and observer independent automated methods in digital pathology are advancing rapidly. Problems like segmentation, classification and detection in digital pathology, are solved with machine learning and image processing techniques. State-of-the-art methods in nuclei segmentation problem include supervised deep learning methods. However, there are drawbacks. Labeled data is often not obtainable and models that is trained to generalize a certain set of data, are not usable in dataset from different domains.

In this work, generative models are considered for unsupervised nuclei segmentation. Two different approach is proposed. A domain adaptation model is proposed to solve domain dependency of deep models and Nuclei-to-annotation translation model is proposed to perform unsupervised segmentation on histopathology images.

Keywords: Cycle, GAN, adversarial, segmentation, nuclei, unsupervised, image-to-image, histopathology

ÖZET

Çekişmeli Üretici Ağlar ile Hücre Çekirdeği bölütlenmesi

Onur Can KOYUN

Elektronik ve Haberleşme Mühendisliği
Yüksek Lisans Tezi

Danışman: Prof. Dr. Tülay YILDIRIM

Mikroskobun keşfedilmesinden bu güne, doku örnekleri analiz edilerek hastalıkların tanı ve ilerliyiş analizinin yapılması çok önemli bir hale gelmiştir. Patologlar hücreleri, hücre çekirdeklerini veya kolajeni vurgulamak amacı ile kimyasallar ile boyanmış doku örneklerini analiz eder. Hücrelerin sahip olduğu morfolojik özellikler, hastalığı anlamaya ve tedavi yolları bulmaya yardımcı olur. Ancak, patoloji analizi çok pahalı ve zaman alıcı bir süreçtir. Ek olarak, histopatoloji görüntülerinin altında yatan bilgileri yorumlamak için patologların bir hücre ve doku fizyolojisini derinlemesine bilmesi gerekir. Ayrıca, farklı patologlar farklı yorumlar yapabilirler.

Dijital patolojide hızlı, güvenilir ve gözlemciden bağımsız otomatik yöntemler hızla gelişmektedir. Bölütleme, histopatoloji görüntülerinin sınıflandırılması ve hücre tespiti gibi problemler makine öğrenmesi ve görüntü işleme teknikleri ile çözülmektedir. Çekirdek bölütleme problemindeki en başarılı yöntemler, eğiticili derin öğrenme yöntemlerini içerir. Ancak, bu yöntemlerin bazı dezavantajları vardır. Etiketli veriler genellikle elde edilmesi zordur ve belirli bir veri kümesini genelleştirmek için eğitilmiş modeller, farklı yollarla elde edilmiş veya farklı dokulardan alınmış veri kümelerinde kullanılamayabilir.

Bu çalışmada, çekirdek bölütleme problemi için çekişmeli üretici modeller ele alınmıştır. İki farklı yaklaşım önerilmiştir. Derin modellerin veriye bağımlılığını çözmek için domen-adaptasyon modeli önerilmiştir. Diğer yöntem, histopatoloji görüntüleri üzerinde denetimsiz bir bölümleme yapmak için çekirdekten-etikete-geçiş modeli önerilmiştir.

Anahtar Kelimeler: GAN, çekişmeli, üretici, derin, öğrenme, hücre, bölütleme, eğiticisiz, histopatoloji

1
Introduction

Since the discovery of the microscope, diagnosis and prognosis of diseases through analyzing tissue samples have become crucial. Pathologists analyze tissue samples which are stained with agents to highlight cells, nuclei or collagen. These samples contain morphological properties of cells which are informative on many aspects of diseases. However, pathological examinations are expensive and time consuming processes. In addition, to interpret the information underlying pathology images, pathologists need a great understanding of cell and tissue physiology. Thus, according to their knowledge, different observers might reach different conclusions.

With the advancements on machine learning and image processing methods, automated pathological analysis in digital pathology has been emerged. There are different strategies such as detecting and extracting meaningful elements from pathology images or extracting features that represent the problem according to clinical data. The most common indicators of many diseases are related to nuclei [1].

Morphology of nuclei is used to determine mitotic index and degree of nuclear pleomorphism [2]. Mitotic index functions as a critical prognostic tool for estimating the response of cancer cells to chemotherapy and the nuclear pleomorphism is used to diagnose malignant lesions. Hence, segmentation and detection of nuclei is a crucial problem in automated digital pathology.

Classical methods for nuclei segmentation includes simple image thresholding, morphology based methods, region growing and watershed algorithm [3]. These methods work well enough with the right parameters if the morphology of the nuclei in data does not variate too much. Statistical models such as Gaussian mixture models and Bayesian models [4], [5] are performing well in certain cases however they are mostly depended on initialization. In addition, statistics of histopathology image variate with data and this causes statistical models to under perform. As a result, classic methods with predefined parameters cannot be the general solution for nuclei

segmentation problem.

Deep learning methods are often used for nuclei segmentation in histology images. Encoder-decoder based CNNs such as U-Net [6] variants with pixel-wise loss are state-of-the-art. Problem with supervised techniques in deep learning is that the network is trained to learn distribution of certain amount of data. This yields some problems. When a network trained with a certain dataset, its performance in other dataset for same problem is poor. For nuclei segmentation problem, data variations can be caused by diffrent image retrieving methods, usage of different microscopes, different magnification rates, different staining processes or any other minor changes in data distribution. This means methods are domain specific. Solution to this problem is an active research area. Generalized solution might be far-fetched idea but expanding domain dependent narrow solutions to other domains is possible.

1.1 Literature Review

Considering nuclei segmentation problem, literature is rich and active. Image processing methods such as image thresholding, morphology based methods, region growing, active contour models, watershed algorithm [3]. Statistical models like Gaussian mixture models and Bayesian models [4], [5] are performing well in certain cases. Recently, deep learning models drew a lot of attention for nuclei segmentation problem. In this section, past and current research for nuclei segmentation are presented.

Thresholding is used for binarization of target image. After selecting a global threshold, pixel intensities that is bigger then threshold value becomes 1 and other intensities becomes zero. This way, image I can be converted to binary image I'. This method performs better on a distinct, preferably homogeneous background. The main parameter to be set is the threshold level. The difference between nuclei and background regions may not be obvious, making it difficult to find an accurate threshold level. Although this methodology is generally only defined in gray-level images, it can be applied to color images using multidimensional kernels. Thresholding method performs poorly in the segmentation of clustered or overlapping nuclei [3]. A threshold can selected using methods like Otsu's method which calculates an optimum threshold according to minimum intraclass variance [7]. Another technique is that choosing an adaptive threshold [8].

Thresholding methods are often used to solve nuclei segmentation problem. Phansalkar et al. [9] proposed an adaptive local thresholding method for non-uniform staining of nuclei. Plissiti et al. [10] proposed a spatially adaptive thresholding

method to separate overlapping nuclei. They used an energy based threshold selection method and selected threshold applied spatially to segment overlapping nuclei separately. Li et al. [11] used gradient flow algorithm and adaptive thresholding on segmentation of 3-dimensional histopathology images.

Another method is using morphological operations. In morphology, images are processed as sets [12]. A filter which has predefined structural elements such as shape, are applied to an image and drawing conclusions from correlation between filter and image. Basic morphological operations are opening, closing, erosion and dilation. In the literature, morphology based approaches are usually used with additional algorithms. Thiran et al. [13] proposed a segmentation method which geodesy (mathematical morphology) is used to remove background noise from images and operate a segmentation of the nuclei. Plissiti et al. [14] proposed a method that uses morphological reconstruction to detect nuclei. Mathematical morphology based watershed algorithm [15] is also used in many study. Watershed depends on geodesy of images. Wahlby et al. [16] applied watershed algorithm to gradient magnitude of image. Jung et al. [17] proposed a watershed based algorithm for segmentation of cervical an breast cell images. They first apply a distance transform [18] and elliptic shape of nuclei are used as a prior to choosing h-value in the watershed based algorithm. Cloppet et al. [19] used watershed algorithm for segmentation overlapped/aggregating nuclei.

Active contour models (ACM) couple shape and image features such as gradient and distribution of pixel values. Performance of segmentation is depended on initial points of seeds. Cosatto et al. [20] proposed an ACM method to measure size of nuclei and grade pleomorphism of nuclei. Method consist of difference of Gaussian (DoG) filtering, Hough transform and ACM to extract boundary of nuclei. They claim 90% accuracy. Huang et al. [21] combined watershed algorithm and ACM to operate segmentation on hepatocellular carcinoma histopathology images. First, noise is removed using morphological operations. Then, watershed transform is applied to detect edges and ACM is applied to fit contours on nuclei.

Dalle et al. [22] proposed a novel method which benefits from gradient of image in polar coordinates. Initially, detection of nuclei are made with thresholding and morphology methods. Then, each patch of detected nucleus centers transformed into polar coordinates. Subsequently, using a bi-quadratic filter, gradients are obtained. This method yields 92,16% accuracy according to authors.

Graph-based models express conditional dependence between variables. Graph-based models can be used with different approaches such as discrete modeling of

images at different pixel intensities or different morphological relationships. Ta et al. [23] proposed a method based on graph regularization. Using Voronoi diagrams, a segmentation scheme is proposed for breast cancer histopathology images. Comparison between proposed method and k-means along with Bayesian classification is done by authors, and they claim 93,6% and 96.4% accuracy respectively.

Kohafi et al. [24] proposed graph-cuts(Gcuts) based nuclei segmentation method. Using Gcuts based binarization approach foreground of histopathology images are extracted. Poisson distribution based energy function is minimized to catch spatial relationship with neighboring pixels. After binarization process, seed point of each nucleus are detected with a combination of multiscale Laplacian of Gaussian filtering and distance map based scale selection. These points are used as basis for segmentation process. Authors claim 86% accuracy on 7400 nuclei within 25 histopathology images. Undersegmentation is reported by the authors in the case of highly clustered nuclei.

Chang et al. proposed a multi reference Gcuts method for segmentation of glioblastoma histopathology images. Their main objective was to solve problems caused by biological variations of geodesic features. They claim 85% TPR and 75% PPV on TCGA dataset [25].

Convolutional neural network (CNN) based methods yield thriving results in most of the medical image segmentation problems. There are different approaches of CNNs for segmentation problems. First approach is that extracting patches from image and classifying each patch with a CNN classifier. This method yields good results but performance is poor because of overlapping patches which means unnecessary computation. Another approach is fully convolutional network approach (FCN) [26]. Basically, FCN uses same idea with batch-classification approach. Extracting patches with a sliding window approach can be written as convolution operation. Instead of fully connected layers at the end of CNN, convolution layers are used in FCN. Output has same size with input and same depth as number of classes. This way, classification of each pixel can be made sparsely.

Ronneberger et al. proposed an encoder-decoder CNN named U-Net [6] with pixel-wise loss for medical image segmentation. In this architecture, similar to FCNs, fully convolutional network approach is used. The main difference is that using downsample-upsample methodology, low level and high level features are used together. Similar approach is applied to 3D data by Çiçek et al. [27]. They used Dice loss instead of cross-entropy loss in their work.

In AMIDA-13 challenge of mitosis detection, Cireşan et al.[28] outperformed other

participants with a CNN approach. Same authors participated to EM 2012 challenge of neural process segmentation and they proposed a CNN with post-processing [29] such as smoothing and thresholding of output probability map.

Studies related to nuclei segmentation using deep learning methods are mostly supervised. Cruz-Roa et al.[30] used a convolutional autoencoder approach for segmentation of H&E stained basal cell carcinoma histopathology images. Malon and Cosatto [31] proposed a method that combines shape related features with CNN to perform mitosis detection. Wang et al. [32] utilized ensemble of CNNs and handcrafted features for mitosis detection problem. Ferrari et al. [33] used a CNN based patch classifier to count bacterial colony from culture plate. Shkolyar et al. [34] used patch classifier to detect mitosis. Song et al. [35] proposed multiscale CNN and graph-partitioning method for segmentation of cytoplasm and nuclei. Xie et al. [36] adopted a CNN model that learns voting offset vectors and confidence to perform nuclei detection. Akram et al. [37] used FCN for cell bounding box proposal and CNN for segmentation. Albarqouni et al. [38] utilized a crowd searching layer for CNNs and perform nuclei detection on H&E stained data. Bauer et al. [39] used CNN based patch classifier on IHC stained nuclei data. Chen et al. [40] wield deep regression network for mitosis detection. Janowczyk et al. [41] introduced resolution adaptive deep hierarchical learning scheme to perform segmentation on H&E stained nuclei data. Kashif et al. [42] utilized a combination of CNN and hand-crafted features for detection of nuclei. Mao and Yin [43] proposed hierarchical CNNs for patch sequence classification to perform mitosis detection. Mishra et al. [44] used CNN based patch classifier for mitochondria classification. Phan et al. [45] used transfer learning for classification of Hep2-cells. Sirinukunwattana et al. [46] introduced locally sensitive CNN using spatial constraints to detect and classify nuclei. Veta et al. [47] utilized a CNN based method to directly measure nuclei area without segmentation. Wang et al. used a combination of two CNNs for joint cell detection of classification. Xie et al. [48] used a fully convolutional regression network to detect and count nuclei. Xing et al. [49] proposed selection based shape model combined with CNN to perform segmentation on H&E and IHC stained histopathology data. Xu et al. [50] used stacked sparse autoencoders (SSAE) for nuclei detection. Yang et al. [51] proposed FCN with an iterative k-terminal cut algorithm.

Recently, generative deep learning models is gotten a lot of attention. Architectures like generative adversarial networks (GAN) [52], variational autoencoders (VAE) [53] and their variants are used to solve many problems in computer vision. One of the problems that is solved with generative models, is called image-to-image translation. The problem is defined as transforming an image from its original domain to some other domain (changing its style, adding or deleting visual features etc.)

automatically, while preserving the original structure or semantics.

There are different generative modeling approaches. Hu et al. [54] used generative adversarial networks to learn cell-level visual representation of histopathology images. They showed that classification, segmentation and detection can be made in an unsupervised manner with generative models. Han et al. [55] utilized a nuclei counting method that uses GAN for 3D image volumes under bad conditions such as crowding and touching of nuclei, overlapping of nuclei, and shape and size variances of the nuclei. Ihle et al. [56] proposed a GAN based segmentation network that uses histogram of images to generate segmentation maps. Proposed method compares histograms of generated and real images instead of discriminating generated and real samples. Hou et al. [57] used GAN for generating background of histopathology images. They used random dilation to generate nuclei and applied nuclei to generated background. Xu et al. [58] introduced CycleGAN to transform H&E stained histopathology images to IHC stained ones. Shaban et al. [59] proposed a method that uses CycleGAN to normalize color variations caused by staining process.

1.2 Objective of the Thesis

Nuclei segmentation problem is a crucial step for automated digital pathology. Current supervised and unsupervised methods either requires additional inputs such as annotations or depend on initial parameters and are not scalable. In this study, a scalable, domain independent nuclei segmentation scheme is introduced. Problem of nuclei segmentation is treated as domain adaptation problem and using adversarial training, an unsupervised segmentation method for histopathology images is proposed. The representation of nuclei in a trained DNN is intended to be used for performing nuclei segmentation independent of domain.

1.3 Hypothesis

Current methods for nuclei segmentation are problem specific. On the contrary, human perception is quite flexible. If the general morphology of nuclei are known, individuals can apply this knowledge to detect nuclei regardless of image domain. Hence, a deep neural network which is trained with appropriate objective, can be used to perform nuclei segmentation to other image domains. Domains can be changed with parameters such as stain method, type of tissue, type of microscopy, amount of magnification etc. To perform nuclei segmentation, basic knowledge of nuclei and its background must be learned by agent and spreading this knowledge to other domains is depended on differences in features between the domains. Thus, there must be two objective, first one is learning of segmentation process, the second one is learning of different domain representations. This yields two different approach for nuclei segmentation. First one is that using a nuclei segmentation dataset from domain-A for training and performing segmentation on domain-B, the second approach is generating random nuclei annotations and injecting domain characteristic to this model. In this case, segmentation problem becomes a domain adaptation problem.

2
Deep Learning Models

Can machines think? It is a very old question that is asked by Alan Turing and many other scientists. The machines that can think, in other words machines with intelligence, could be interpreted as machines that mimic, develop, demonstrate human cognition or behavior. Despite recent developments in the artificial intelligence, we are still in the era of weak-AI.

The definition of weak artificial intelligence(Weak-AI or Narrow AI) can be made as artificial intelligence that is focused on a specific task. On the contrary, strong artificial intelligence (Strong-AI or Artificial General Intelligence) is able to apply intelligence to solve any kind of problem. All currently existing artificial intelligence systems are weak-AI at most.

Despite complexity of neurons, a single neuron cell can be modeled and could be used to create artificial neural networks that is able to learn a specific task. In 40's, McCulloch and Pitt modeled a single neuron as weighted sum of inputs with a non-linear activation function [60]. It's the fundamental model which is core of many ideas.

After McCulloch-Pitt neuron model, Donald Hebb wrote The Organization of Behavior in 1949, a study that points to the fact that neural connections are reinforced each time they are used, a concept that is similar to human learning process. He argued that if two neurons fired at the same time, the connection between them was strengthened [61].

In 1959, Bernard Widrow and Marcian Hoff proposed "ADALINE" and "MADALINE" [62]. ADAptive LINear Elements was developed to recognize binary patterns to predict next stream of bits on a phone line. Multiple ADAptive LINear Elements was the first neural network which was used to solve a real world problem. It was designed as adaptive filter that cancels echoes on phone lines.

In 1962, Widrow & Hoff proposed a learning algorithm which minimizes the error for

all weights [63]. The idea is that while one neuron may have a erroneous output, the weight values can be updated to distribute the error across other neurons, or at least to neighboring neurons. Applying this rule would result an increase in total error, however network will eventually converge to minimal error surface.

Despite the later success of the neural network, traditional von Neumann architecture took over the computing scene, and neural research was left behind. It was the first AI winter.

In 1982, interest in the field was renewed. John Hopfield published a paper about his new approach which was to create more useful networks by using bidirectional connections between neurons. Previously, the connections was only one way [64].

In 1986, the problem was how to extend the Widrow-Hoff rule to multiple layers. David Rumelhart, a former member of Stanford's psychology department, came up with the idea of distributing the error throughout the network similar to Widrow & Hoff learning algorithm [65]. In 1989 Hinton and Rumelhart [66] proposed the back-propagation networks which was able to use many layers contrary to widrow & Hoff learning.

In 1989, Yann LeCun developed the convolutional neural network [67] which is specialized kind of neural network for processing data that has a known grid-like topology such as time-series data, which can be thought of as a 1-D grid taking samples at regular time intervals or image data, which can be thought of as a 2-D grid of pixels. Convolutional networks have been extremely successful in practical applications.[68]

In 90's the second AI winter was emerged. Expectations had not been met and governments and research centers made huge budget cut-offs for AI research.

In late 2000's, with the advanced computers and easy access to data, deep neural networks have been emerged. Using GPU's training process of deep neural networks have got faster and easier.

Today deep learning is state-of-the-art solution for many problems from many different domains.

2.1 Neural Networks

Artificial Neural networks (ANN), also called feed-forward neural networks or Multi-Layer perceptron, are the basic models to approximate a function $y = f^*(x)$. From mathematical point of view, ANNs is able to approximate any kind of function

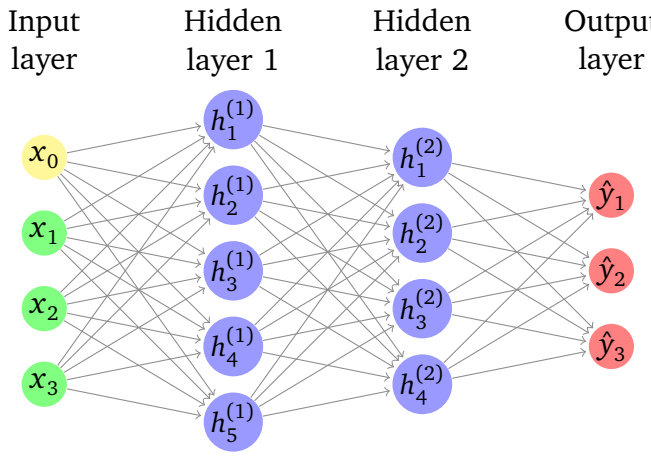


Figure 2.1 A feed forward neural net.

with enough capacity and true objective. This networks called feed-forward nets. because of the structure of calculations. Using information x, y is calculated within the network and calculations defines function $f^*(x)$. Function depends on hidden layer weights θ_w and function definition can be interpreted as mapping input x to output y $y = f^*(x, \theta_w)$. Each consecutive layer can be modeled as function f^i. In equation (2.1), NN in figure 2.1 is written as a mathematical function of x and θ_w.

$$y = f^*(x, \theta_w) = f^2(f^1(x_i, \theta_{w^h(1)}), \theta_{w^h(2)}) \tag{2.1}$$

One way to understand NNs is begin with understanding linear models and their limitations. Linear models can be used for modeling linearly separable data distributions. For example, logical OR & AND problems can be solved with a linear model. However XOR problem cannot be solved with linear models. On the other hand, NNs are non-linear models because of the non-linear activation functions that are used at the end of each hidden layer.

$$f^1(x, \theta_{h^{(1)}}) = h_1^{(1)} = x_1 * w_{11} + x_2 * w_{12} + x_3 * w_{13} + x_0 \tag{2.2}$$

In general, activation functions are chosen as non-linear, fully or partially differentiable functions such as sigmoid or tangent hyperbolic functions. Gradient based learning algorithms are mostly used to optimize NNs. Non-linearity of activation functions causes a non-convex optimization process when stochastic gradient descent is applied over a loss function. Optimization of weight parameters in NNs are highly depended to initialization of weights and biases. Weight values must be chosen small values and bias values can be chosen as zero or a small positive value. Still, stochastic

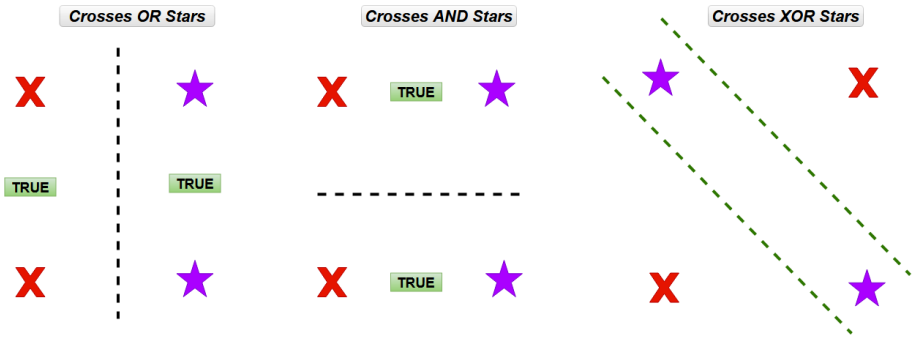


Figure 2.2 Linear separability of logic operations

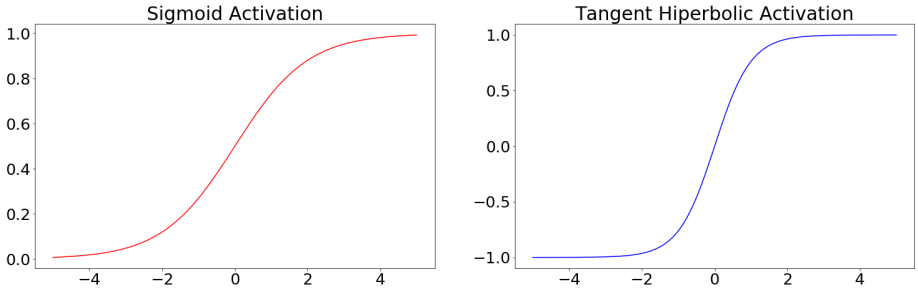


Figure 2.3 Sigmoid and Tangent hyperbolic activation functions

gradient descent algorithm does not guarantee convergence of network.

Optimizing a feed forward network with gradient descent using a cost function depends on chain rule. Beginning from input i, forward propagation through neural network yields a cost $J(\theta)$. According to this cost and chain rule, error of each weight in each layer can be calculated from its gradient and weights are updated to lower the cost. This process is named as back-propagation of error in neural networks.

$$\begin{aligned} y &= f(g(k(x))) \\ z &= k(x) q = g(z) y = f(q) \\ \frac{dy}{dx} &= \frac{dy}{dq} \cdot \frac{dq}{dz} \cdot \frac{dz}{dx} \end{aligned} \tag{2.3}$$

Generally loss (Cost, objective etc.) function over a dataset can be written as $J(\theta) = \mathbb{E}_{(x,y)\sim\hat{p}_{data}} \mathfrak{L}(f(x;\theta), y)$ where the $\mathfrak{L}$ is per-instance loss. Then, finding optimum θ over gradient descent becomes maximum likelihood approach.

$$\theta_{ML} = \underset{\theta}{argmax} \sum_{i=1}^{m} log p_{model}(x^{(i)}, y^{(i)}; \theta)$$
$$\nabla J(\theta) = \mathbb{E}_{x,y \sim \hat{(p)}_{data}} \nabla_{\theta} log p_{model}(x, y; \theta) \tag{2.4}$$

•Learning rate schedule $\epsilon_1, \epsilon_2, ...$

•Set initial θ parameters.

for *number of iterations* **do**

Sample minibatch of k samples $\{x^1, ..., x^k\}$ from training set with corresponding labels y^k

Update network's weights by descending its stocastic gradient:

$\theta^{new} = \theta - \epsilon \nabla_{\theta_g} \frac{1}{k} \sum_{l=1}^{k} \mathfrak{L}(f(x^{(l)}; \theta), y^l)$

end

Algorithm 1: Stochastic Gradient Descent algorithm to optimize weights.

Neural network optimization may face many difficulties. One of them is local minima problem. Depending on initialization of weights, NNs may converge to local minima and yield lower accuracy.

Another problem is vanishing/exploding gradient problem. If the NN is too deep gradients will saturate at some point (there are points with zero slope in activation function as seen in 2.3) and NN will not get better with updates. This is called vanishing gradient. On the contrary, when magnitude of gradients are too big, network becomes unstable which is named as exploding gradient problem.

NNs with fully connected approach is not an optimal solution. When network goes deeper, number of parameters grow dramatically and optimization of parameters gets harder.

As a result of these problems, neural nets with shared weights are proposed. Convolutional neural networks (CNN) are used with shared weight approach and yielded successful results. With advancements on hardware like GPUs and multicore CPUs, Deeper CNNs with many layers became the most common approach for neural network research.

2.2 Deep Models

Deep neural networks(DNN) can be interpreted as neural networks with many layers. However there are more to explain. Classic NNs do not have capacity to represent high

dimensional data such as image, text and speech. In the past, dimensionality reduction and feature extraction algorithms were used before NNs to solve this problems. In contrary, DNNs have more representation power and does not need any other feature extraction process in many cases.

2.2.1 Convolutional Neural Networks

The history of convolutional neural networks begins with neuroscientific experiments long before the relevant computational models were developed. Neurophysiologists Hubel and Wiesel collaborated for several years to determine many of the most basic facts about how the mammalian vision system works [69],[70],[71]. Their accomplishments were eventually awarded with a Nobel prize. Their findings that have had the greatest inuence on deep learning models were based on recording the activity of individual neurons in cats. They observed that neurons in the cat's visual cortex responded to images projected in precise locations on a screen in front of the cat. Their greatest discovery was that the neurons in the early vision system could respond most strongly to very specific light patterns, such as edges and curves of objects, but almost never responded to other patterns. Their work helped to characterize many aspects of process of visual information in brain.

In 1989, Yann LeCun proposed convolutional networks which was inspired by this work. In the network, 2-dimensional weights or more traditionally kernels are used to detect low level features. With a data driven approach, kernels trained using gradient descent algorithm. It appeared that trained kernels were focused to the same structures with the cat's brain which verified Hubel and Wiesel's work.

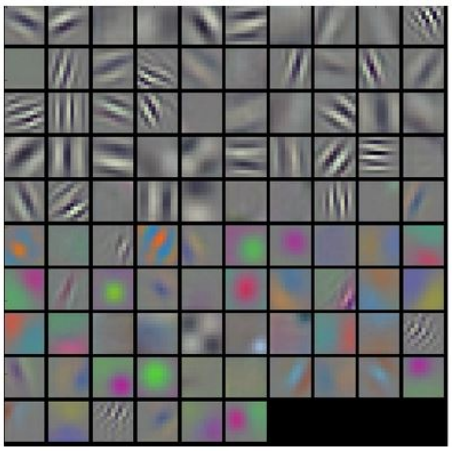


Figure 2.4 Convolution kernels of early layer [72]

Kernel sizes is related to cortical neurons' receptive fields. In Hubel's work, it is shown that simple earlier neurons in visual cortex have smaller receptive fields and more complex neurons have larger receptive fields. In a similar way, while convolutional neural networks' first layers have smaller receptive fields, deeper layers have larger receptive fields. Earlier layers detects edges, corners etc. and deeper layers detect more definitive features.

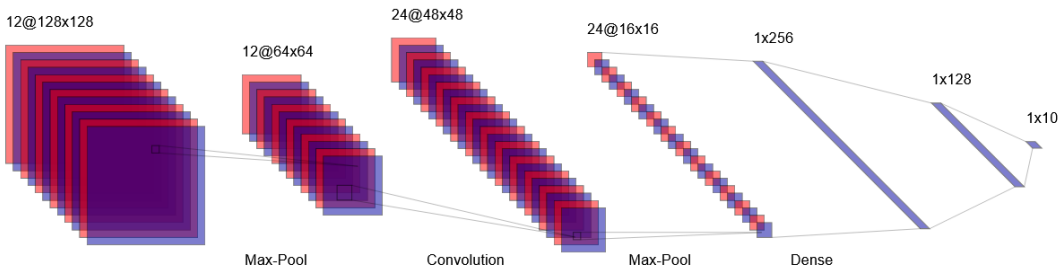


Figure 2.5 An interpretation of Convolutional Neural Network

2.2.1.1 Convolution Operation

In mathematics, convolution operation is an operation on two functions to produce a third function which is an altered form of one of the previous functions. In the convolutional network terminology, w is the kernel which carries the piece of information that represents the data and f is an instance of the data.

$$(f * w) \overset{def}{=} \int_{-\infty}^{\infty} f(\tau) \cdot w(t-\tau) d\tau \tag{2.5}$$

Discrete form can be defined as;

$$(f * w) \overset{def}{=} \sum_{a=-\infty}^{\infty} f(a) \cdot w(t-a) \tag{2.6}$$

and 2D form for images can be defined as;

$$(I * K)(i,j) \overset{def}{=} \sum_{m} \sum_{n} I(m,n) \cdot K(i-m, j-n) \tag{2.7}$$

In equation 3, image I and kernel K convolved over row and column pixels.

2.2.2 Activation Functions, Normalization and Pooling Operations

Despite the fact that convolution is the main operation, without non-linearity of activation functions, down-sampling and normalization layers, CNNs are not scalable.

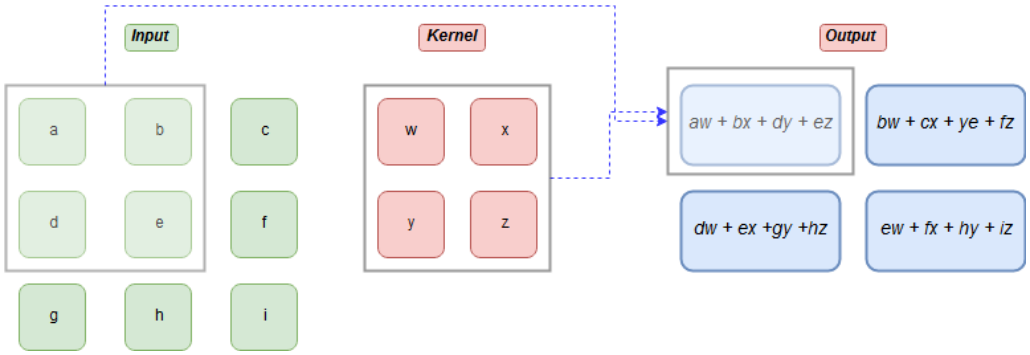


Figure 2.6 Convolution operation

Using activation functions after the convolution layer causes non-linearity, pooling operations provides down-sampling and normalization layers provides robustness to learning process.

Almost every CNN architecture uses a variant of Rectified Linear Unit (ReLU) as activation function after convolution layers. ReLU is a non-linear function which rectifies the input vector similar to a half-wave rectifier. Negative parts of input vector becomes zero and positive part does not change.

Activation functions like sigmoid or hyperbolic tangent create vanishing gradient problems in deeper networks. Because, these functions saturate in both directions. On the contrary, ReLU only saturates in one direction. This approach somewhat solve the vanishing gradient problem. However, with large learning rates, dying ReLU problem occurs. Dying ReLU problem is a variant of vanishing gradient problem that neurons are pushed into states in which they become inactive for all inputs. There is no gradient flow from the inactive neurons thus their states cannot change with further iterations.

Maas et al. [73] proposed Leaky ReLU to solve dying ReLU problem. This activation is similar to ReLU activation, yet the rectifier leaks negative values to output. In Equation (2.8), Leaky ReLU activation function is defined.

$$y = \begin{cases} x_i, & \text{if } x_i > 0. \\ x_i \cdot a, & \text{if } x_i < 0. \end{cases} \tag{2.8}$$

After leaky ReLU, He et al.[74] proposed parametric rectified linear unit (PReLU) [74] which is similar to both ReLU and leaky ReLU. This function adaptively learns the leak parameter a of rectifier. It is shown that using PReLU solves gradient vanishing

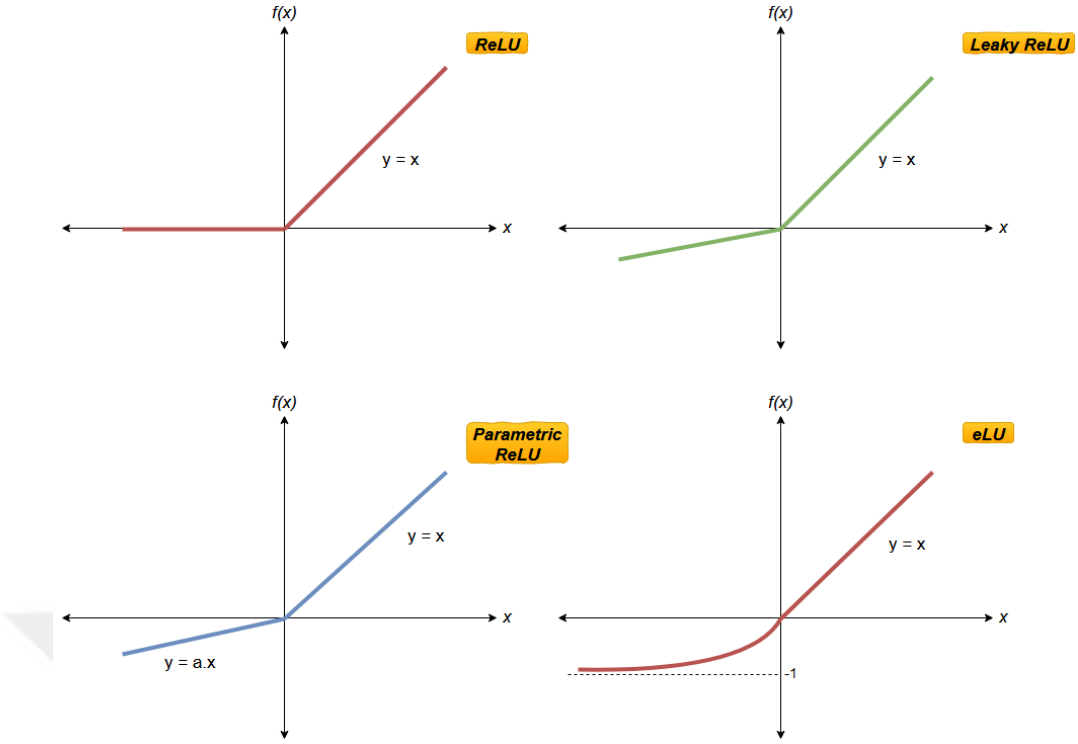


Figure 2.7 ReLU Activation function and its variants

problem in very deep networks. He et al. achieved human level performance on ImageNet classification task using PReLU (DNN with PReLU : 4.96% error [74], Human performance: 5.1% error [75]).

There are other activation functions such as exponential linear unit (ELU)[76], scaled exponential linear units (SELU)[77] etc. These functions have good results in different tasks, however their performance over PReLU or Leaky ReLU are not noteworthy.

After activation function, usually a down-sampling method is applied. CNN invented in 90's and hardware which was used to train CNNs were very slow. This was the main reason to add some kind of down-sampling method to CNNs. Second reason is that down-sampling provides non-linearity which improves the performance of networks. Trendy down-sampling methods in literature are max pooling, average pooling, stochastic pooling and strided convolution.

Even though pooling methods works perfectly well, there are fundamental problems that is caused by pooling. Using pooling layers causes losing spatial relations. For example, if the CNN classifier is trained to classify faces and if features like eyes, nose, mouth are detected, classification will be made as there is a face. Geometric locations of these feature are not taken into account.

Another problem is losing important features. Every layer in which pooling is used, selectively sends signals to further layers. This selection process is not always beneficial to network. Despite these problems, pooling methods are still in use.

Another important layer is normalization layer. Because of the mini-batch training procedure, each mini-batch of data show different statistical properties. This causes oscillation on loss surface and convergence of network might get slower. Also, there is internal co-variate shift (ICS) problem between layers. Internal co-variate shift is defined as change of distribution of each layer with each gradient update. Update in a particular layer is done as if other layers won't be updated. This yields ICS problem and convergence gets harder with deeper networks. Normalization layers are especially important for generative models.

Batch normalization [78], instance normalization [79], spectral normalization [80], group normalization [81] and local response normalization [82] are most preferred normalization techniques in deep learning literature.

2.2.2.1 Convolutional Neural Network Approaches

In this section important CNN architectures and their important contributions are highlighted.

2.2.2.1.1 AlexNet AlexNet, the winner of Imagenet classification competition in 2012 (ILSVRC12), is a basic architecture for classification problems. It consists of convolutional layers with ReLU activation function, pooling layers, normalization layers and fully connected layers with dropout to prevent overfitting.

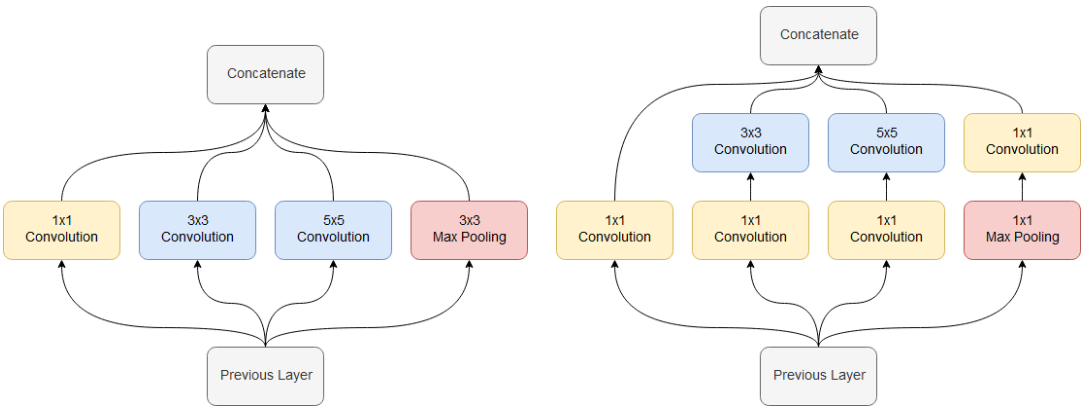


Figure 2.8 Inception module

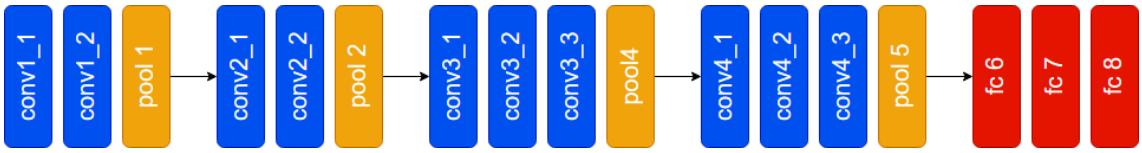


Figure 2.9 VggNet Layer

2.2.2.1.2 GoogLeNet Szegedy et al. [82] proposed a deep convolutional neural network architecture that consists of Inception modules which was achieved the state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 [75] (ILSVRC14). The inception modules were the main contribution to literature.

Inception module is called network in a network because of its structure. The operation is like basic convolution layers but the difference is convolutions with different receptive fields (kernel sizes). After the convolutions, feature maps is concatenated before next layer. This way, different sized features can be detected in each layer.

2.2.2.1.3 VGGNet Simonyan et al. [83] developed a model that only uses 3x3 kernels in convolution layers. This approach showed that stacking two convolutional layers with 3x3 sized kernels is same with using 7x7 kernel in one layer.Stacking two convolution layers back to back creates non-linearity in the layer.

2.2.2.1.4 Residual Networks In 2015, He et al. [84] proposed a deeper network with 156 layers and won a lot of classification competitions including ILSVRC'15 and COCO'15. The problem of deeper networks was the vanishing gradients and in this work residual blocks is proposed to solve this phenomenon.

Basic structure of residual block can be defined as, $L(x) + x$ where function L is the convolutional layer and x is the signal.

2.2.2.1.5 Fully Convolutional Neural Networks Fully Convolutional Networks (FCN) is proposed by Long et al. [26] to solve semantic segmentation problems. Instead of using fully connected layers, they only use convolutional layers. Pixel-wise loss function are used sparsely for pixel-wise prediction of semantic segmentation map. Instead of train a segmentation network from scratch, they fine-tune classification models such as AlexNet, VGG16 and GoogLeNet.

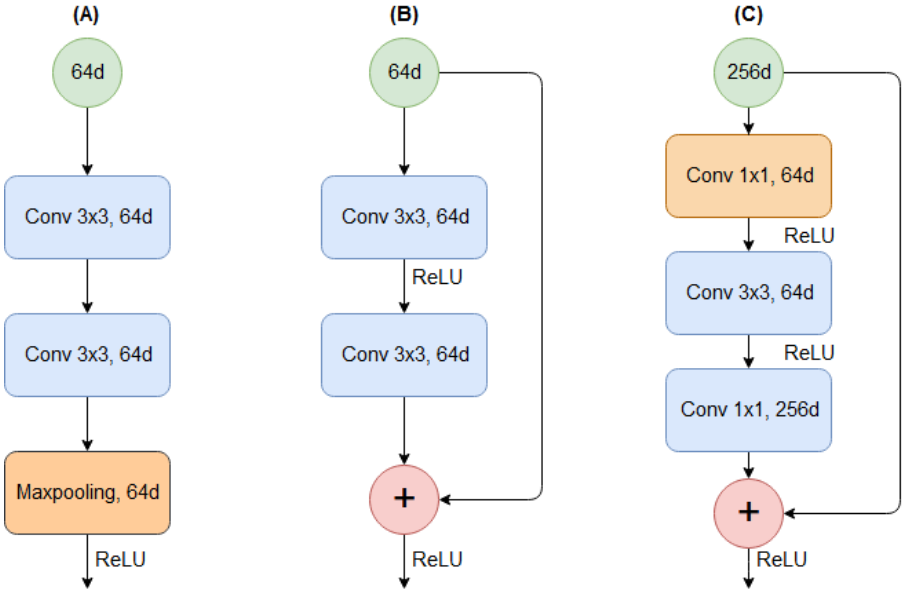


Figure 2.10 A: VGG Layer, B: Residual Layer, C: Residual Layer with dimension reduction

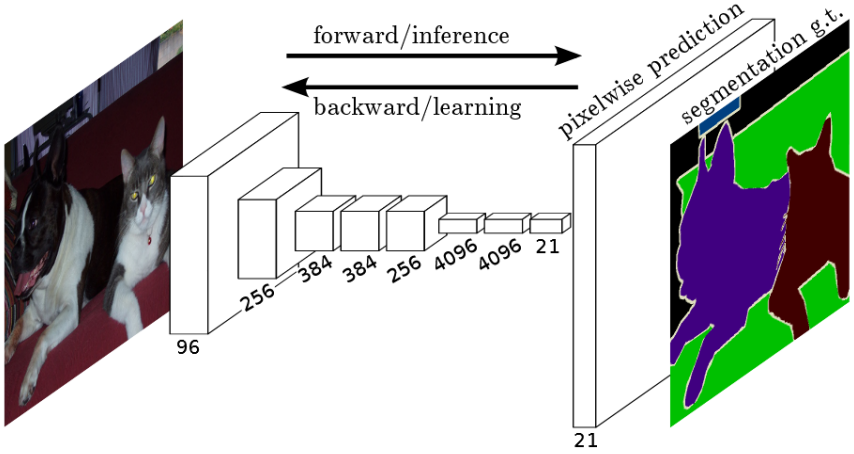


Figure 2.11 Fully convolutional neural networks [26]

2.2.2.1.6 U-Net Medical image segmentation, similar to semantic segmentation, is one of the challenging research topics. From microscopy images to medical images obtained with any kind of imaging method, medical image segmentation on many tasks carries great importance. Ronneberger et al. [6] proposed a CNN based architecture for medical image segmentation problems. In proposed architecture consist of convolution layers, down sampling and up-sampling layers and normalization layers. They use the features in early layers and features in later layers together to generate sparse segmentation map. Because of the U shape, they named the architecture as U-Net.

U-Net is a fully convolutional network approach. The difference is that both down-sampling and up-sampling methods are used together. While size of grid signal decreases number of activation maps increases, contrary while size of signal increases with up-sampling, size of activation map decreases. Each down-sampling layer concatenated to corresponding up-sampling layer (shapes are needed to be same for this operation). This approach yields state-of-the-art results for segmentation problems.

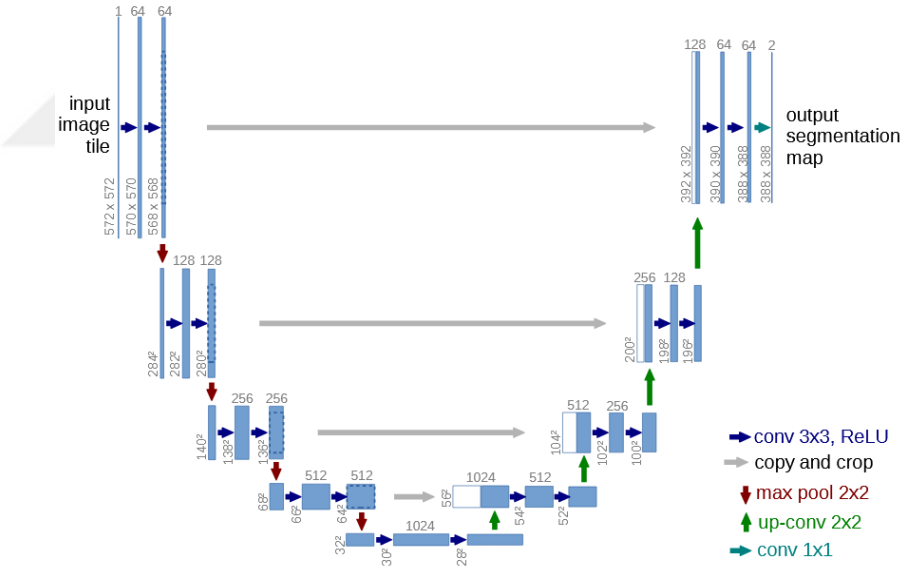


Figure 2.12 Detailed U-Net architecture [6]

2.3 Generative Models

Deep learning promises to represent probability distribution of the data that AI applications encounter. Images, speech, text or any structural data that carry patterns or any kind of information, can be represented with deep neural networks. With the success of discriminative models, representation power of deep neural networks have been proven. On the other side, generative models that use deep neural networks were a bit late because of difficulties in estimating gradients in maximum likelihood approaches. Excluding Boltzman machines (probabilistic) and autoenconders, deep generative models were not considered until proposition of famous Generative Adversarial Networks [52] and Autoencoding Variational Bayes papers [53].

A generative model can be defined as a model that is able to generate data instances from a $P(x)$ data distribution. Generated instances might be same instances as in dataset or new instances within the same distribution. There are different approaches for representing data distributions.

Very small amount of the representation power that deep models do normally have, are used on classification problems and mapping data to labels through deep neural networks cannot solve every practical problem. On the other hand, generative models can be used for lots of applications including classification, segmentation, generation of data instances, style transfer etc. Basically, with appropriate loss function, generative models can be trained to solve any kind of problem. Even though flexibility of generative models, training processes are often unstable.

2.3.1 Generative Adversarial Networks

Generative adversarial networks (GAN) proposed by Goodfellow et al. [52] is a deep generative model that uses a game theoretic approach. Most of the generative models are suffered by lack of a supervision signal and statistical inference models or maximum likelihood approximation are used to provide a supervision signal for training. In GAN's the supervision signal is provided by a zero-sum game between two neural networks (Or between two learnable agent). One of the neural networks is trained to map data distribution to latent variable z. Other NN is trained to discriminate the samples whether coming from the real data distribution or generated by the generator with input as latent variable z.

Relationship between generator and discriminator is similar to relationship between a counterfeiter that mints fake coins and the police who decide whether the coins are real or fake. Every time, counterfeiter mints more realistic coins and tries to deceive

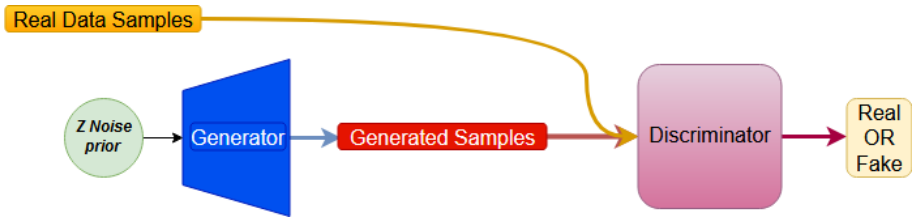


Figure 2.13 Generative Adversarial Networks

the policeman. As the policeman gain experience, he/she learns to separate real coin from counterfeit one.

The objective function that are used to train both generator and discriminator is a minimax game as seen in equation (2.9). In theory, if both of the NN's capacities are enough, at some point of training, generator and discriminator networks will not get better at their objectives with further training. They reach to Nash equilibrium[52].

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[log D(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))] \tag{2.9}$$

Minimizing $1-\log(1-D(G(z)))$ may not provide enough gradient for G to learn. In the begining of the training G generates bad samples and D rejects with high confidence because generated samples are very different from the original data. To avoid this, instead of minimizing $\log(1-D(G(z)))$, $\log D(G(z))$ can be maximized. This function provides better gradients in the beginning of training process.

Theoretically, minimax loss assure that Jensen-Shannon (JS) between distribution of generated samples and data distribution are approximately zero. JS divergence shows how much a particular probability distribution differs from another distribution. KL divergence at zero means distributions are identical. JS divergence (2.11) is a smoothed and symmetrized version of Kullback–Leibler (KL) divergence (2.10).

$$D_{KL}(P||Q) = \sum_{x\in X} P(x) log(\frac{P(x)}{Q(x)}) \tag{2.10}$$

$$JSD(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M) \quad where \quad M = \frac{1}{2}(P+Q) \tag{2.11}$$

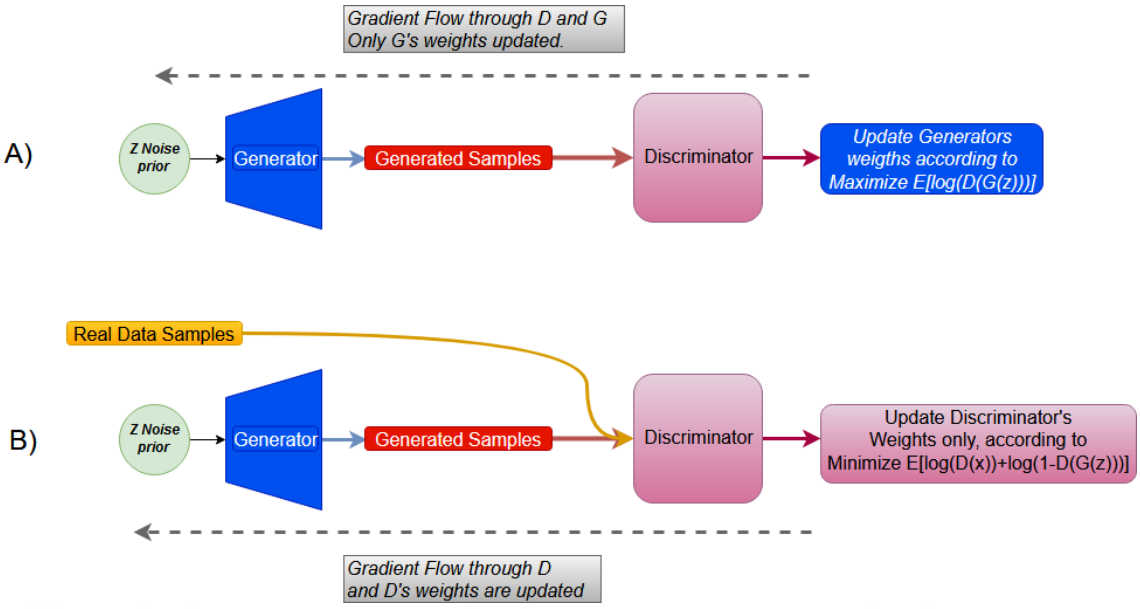


Figure 2.14 Training process of Generative Adversarial Networks, A: Update of generator, B: Update of discriminator

for *number of iterations* **do**

  for *i steps* **do**

    Sample minibatch of k samples $\{z^1, ..., z^k\}$ from noise prior $p_g(z)$.

    Sample minibatch of k examples $\{x^1, ..., x^k\}$ from data distribution $p_x(x)$.

    Update discriminator network's weights by ascending its stochastic gradient: $\nabla_{\theta_d} \frac{1}{k} \sum_{l=1}^{k} [logD(x^l) + log(1 - D(G(z^l)))]$

  end

  Sample minibatch of k samples $\{z^1, ..., z^k\}$ from noise prior $p_g(z)$.

  Update generator network's weights by descending its stochastic gradient: $\nabla_{\theta_g} \frac{1}{k} \sum_{l=1}^{k} log(1 - D(G(z^l)))$

end

Algorithm 2: GAN training algorithm. The number of steps to update discriminator network i is a hyper parameter.

In practise, training process of GAN's is often unstable. Instead of reaching to Nash equilibrium, networks might oscillate on the loss surface. Goodfellow et. al. mentioned this as mode collapse [52]. One of the failure modes is that generator network generates similar samples which are accepted by discriminator. Once generator finds this non-optimal point in space, discriminator will not provide enough gradient and generator will not escape from this non-optimal point. If data provide similar directions of gradients this problem might occur which means generating same

or similar samples only. This problem can be solved with mini-batch discrimination [85]. The concept of Mini-batch discrimination can be explained as discriminator that is trained with more than one example in combination, rather than training with isolated samples, can potentially help prevent the mode collapse of the generator.

Even though the system reaches to Nash equilibrium, the quality of generated samples might not be visually satisfying. This is caused by the vanishing gradient problem. More technical explanation is that distance metric between data distribution and distribution of generated samples does not provide sufficient gradients to improve quality of generation at some point.

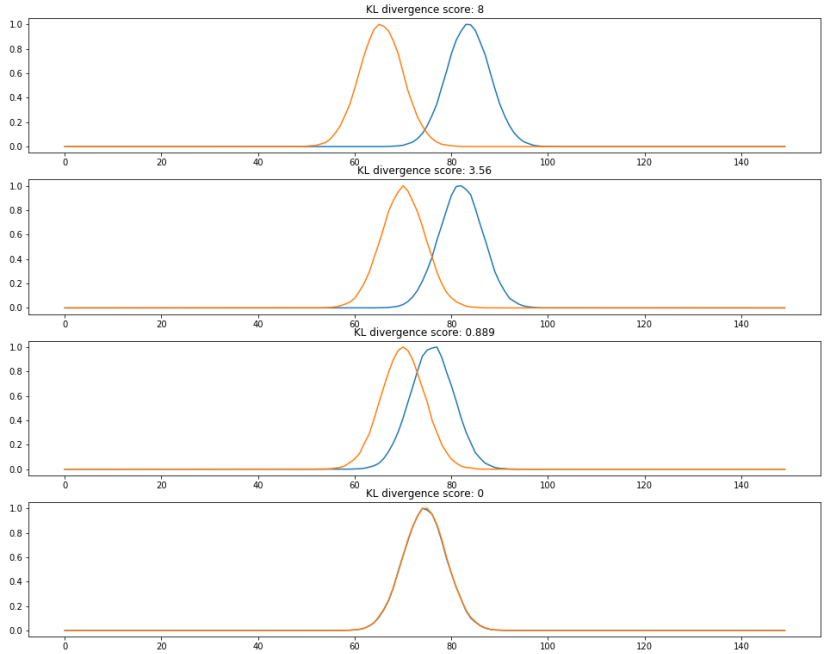


Figure 2.15 KL Divergence between two Gaussian distributions.

In original GAN paper[52], Jensen-Shannon (JS) divergence (2.11) is used instead of KL divergence which was the mostly used distance metric over probability distributions for generative models. There are distance measures other than KLD and JSD such as total variation distance (TV), Wasserstein (Earth Mover) distance and other f-divergences [86].

Arjovsky et. al. [87] is proposed Wasserstein-GAN (WGAN) to generate high quality and realistic samples. In WGAN, Wasserstein distance (2.12) is used instead of JS

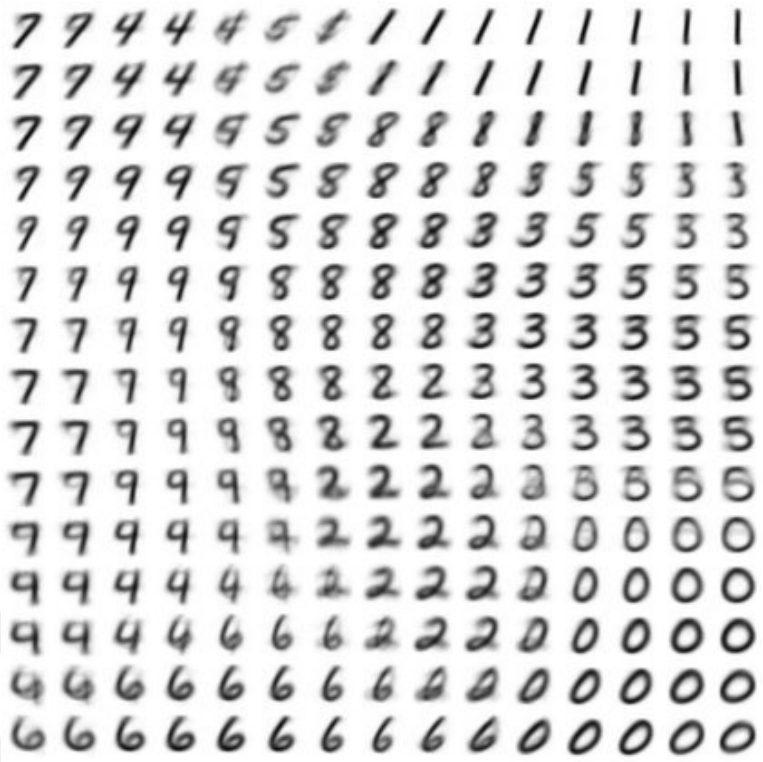


Figure 2.16 Interpolating in the latent space z of GAN's generator network

divergence and this provides more linear gradients for generator.

$$W(P_r||Q_g) = \inf_{\gamma \in \Pi(P_r,P_g)} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||] \tag{2.12}$$

2.3.2 Wasserstein-GAN

Wasserstein distance is more robust than JS divergence according to authors of WGAN paper [87]. However the infimum in equation (2.12) is intractable. To solve this problem, they benefit from Kantorovich-Rubinstein duality which claims for all the 1-Lipschitz functions the supremum can be attained by maximizing the objective function in a truncated variable space.

$$W(P_r, P_\theta) = \sup_{||f||_L \leq 1} \mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{x\sim P_\theta}[f(x)] \tag{2.13}$$

$$\max_{w\in W} \mathbb{E}_{x\sim P_r}[f_w(x)] - \mathbb{E}_{z\sim P_z}[f_w(g_\theta(z))] \tag{2.14}$$

Using wasserstein distance, supremum must be present as seen in equation (2.13). However, without Lipschitz constraint on weights, supremum of this function is

untractable. When Lipschitz condition is met, loss function becomes a maximisation problem (2.14). Choosing a constraint on weights as $W = [-0.01, 0.01]$ provides this condition [87]. Nevertheless, weight clipping is a terrible way to implement Lipschitz constraint. If the clipping is very small, vanishing gradient problem will arise, if the clipping parameter is large, training process of critic (discriminator) gets harder.

Another way to enforce Lipschitz constraint is clipping gradients with a gradient penalty [88]. Wasserstein GAN with Gradient Penalty (WGAN-GP) is proposed by Gulrajani et. al. to improve Wasserstein GAN's. Instead of forcing weights to a box, WGAN-GP uses an additional loss term which is a gradient penalty function.

$$\mathfrak{L} = \underset{w \in W}{max} \mathbb{E}_{x \sim P_r}[f_w(x)] - \mathbb{E}_{z \sim P_z}[f_w(g_\theta(z))] + \lambda \mathbb{E}_{\hat{x} \sim P_{\hat{x}}}[(|| \bigtriangledown_{\hat{x}} D(\hat{x}))||_2 - 1)^2] \quad (2.15)$$

Gradient penalty enforces the gradient's norm to 1 to ensure the Lipschitz constraint as seen in (2.15). A function is 1-Lipschitz if and only if norm of its gradients is one at most.

Wasserstein GAN with gradient penalty is one of the good ways to generate realistic, high quality data instances.

2.4 Cycle-Consistent Generative Adversarial Networks

Image-to-image translation is a computer vision problem, defined as transforming an image from its original domain to some other domain (changing its style, adding or deleting visual features etc.) automatically, while preserving the original structure or semantics. Problem is introduced by Isola et al [89]. They proposed GAN based pix-to-pix architecture to generate realistic and unique samples from structured images as seen in 2.17. However, this approach is limited to transforming structured, label-like images to original domain. In pix-to-pix, authors used a conditional GAN to generate images from labels. Cycle-Consistent Generative adversarial [90] networks (CycleGAN) is proposed to solve unpaired image to image translation problems. Instead of using a conditional GAN, two GANs used together for translation process. First step is the conversion from domain-A to domain-B, second step is the conversion from domain-B to domain-A, additionally there is a reconstruction over generated (translated) images. This way, structural properties of images are preserved but features of translated domain injected to image.

CycleGAN, is a method that is able to learn special features of a specific domain and



Figure 2.17 Paired image-to-image translation problem solved with Pix-to-pix [89]

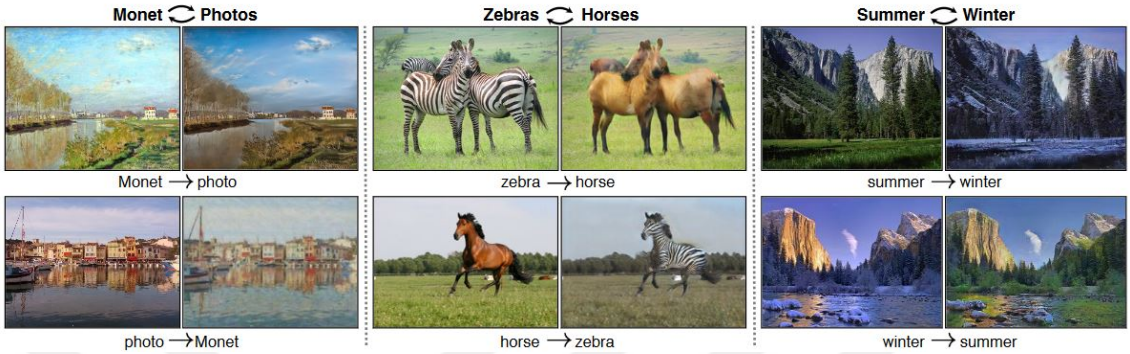


Figure 2.18 Image to Image translation with CycleGAN [90]

cast it to images from different domain. Despite the fact that there is no supervision, CycleGAN uses sets from different domains as supervision signal. In the case of paired data, structural properties of domains are same as seen in 2.17. However, in the unpaired case, images might be very different from each other. To solve this, CycleGAN uses a reconstruction loss between two generator network in addition to adversarial training loss.

Objective of generative parts are $G : A \longrightarrow B', \quad B' \sim P_B(B)$, $F : B \longrightarrow B', \quad B' \sim P_A(A)$ respectively. Here, A and B is two different domains and using adversarial loss translation process is done. To preserve structure of images L1 or L2 loss is applied over $F(G(X))$ and $G(F(Y))$.

$$\begin{aligned} \mathscr{L}_{GAN}(G, D_y, X, Y) = & E_{y \sim p_{data}(y)}[log D_y(y)] \\ & + E_{x \sim p_{data}(x)}[log(1 - D_y(G(x)))] \end{aligned} \tag{2.16}$$

$$\begin{aligned} \mathscr{L}_{cyc}(G, F) = & \mathbb{E}_{x \sim p_{data}(x)}[\|F(G(x)) - x\|] \\ & + \mathbb{E}_{y \sim p_{data}(y)}[\|G(F(y)) - y\|] \end{aligned} \tag{2.17}$$

As a result, the total loss function is shown in 4.9. The coefficient is selected as $\lambda = 10$

according to original paper [90].

$$\mathscr{L}(G,F,D_x,D_y) = \mathscr{L}_{GAN}(F,D_x,Y,X) + \mathscr{L}_{GAN}(G,D_y,X,Y) + \lambda\mathscr{L}_{cyc}(G,F) \tag{2.18}$$

Architectures that are used for F, G, D_x, D_y networks are CNN variants. For G and F, and encoder-decoder based architecture was used. Beginning with 7x7x64 convolution kernels, in each layer number of filters doubled until 256 filters. After that 9 Residual layers were used for translation and deconvolution (Transposed convolution) layers for decoding. After each layer, instance normalization and ReLU activation was applied. Residual layers consist of two 3x3x256 convolutional layer.

Table 2.1 Generator architecture

Generator	Stride	Padding	Normalization	Activation
Conv 7x7x64	1	Reflection	Instance norm.	ReLU
Conv 3x3x128	2	Reflection	Instance norm.	ReLU
Conv 3x3x256	2	Reflection	Instance norm.	ReLU
9 X Residual layer	1	Reflection	Instance norm.	ReLU
DeConv 3x3x128	1/2	Reflection	Instance norm.	ReLU
DeConv 3x3x64	1/2	Reflection	Instance norm.	ReLU
Conv 7x7x3	1	Reflection	Instance norm.	ReLU

Discriminator's network architecture is a basic CNN architecture with Leaky ReLU activation function. Each layer consist of 4x4 Convolution-Instance normalization-Leaky ReLU layers with stride 2. Slope of Leaky ReLU is 0.2. Instead of full input, random 70x70 patches from input is fed into network [89].

Table 2.2 Discirminator architecture

Discriminator	Stride	Padding	Normalization	Activation
Conv 4x4x64	1	Reflection	-	LeakyReLU
Conv 4x4x128	2	Reflection	Instance norm.	LeakyReLU
Conv 4x4x256	2	Reflection	Instance norm.	LeakyReLU
Conv 4x4x512	2	Reflection	Instance norm.	LeakyReLU
Conv 4x4x1	1	Reflection	Instance norm.	LeakyReLU

Cycle-consistent generative adversarial network is an appropriate architecture for unpaired image-to-image translation problem. However, its capabilities are not restricted to solve this kind of problems only. CycleGAN can be thought as an universal adapter. If structure of datasets is similar but distributions diverge from each other, CycleGAN can be used.

3
Common Deep Learning methods and Metrics for Nuclei Segmentation Problem

There are different approaches with deep learning for nuclei segmentation problem. This approaches differ with loss function, architecture and layers. Most of the deep learning approaches are supervised which means nuclei annotations are needed. In this chapter, different approaches are discussed.

3.1 Patch Classification Approach

Initial approaches for nuclei segmentation problem with CNNs were classification-like frameworks. Overlapping patches are extracted for classification process. Classification is performed to determine each consecutive patch is part of nuclei or not. This way, a probability map is generated. Segmentation process is done with a simple threshold to probabilities of each pixel.In general, cross-entropy loss is used for classification of patches. Ciresan et al. [29] trained a network in the patch classification setup to estimate the class label of each pixel.

There are some disadvantages with this method. First of all, overlapping patches causes unnecessary operations. Same points are convolved with same filters over and over again. Another disadvantage is that selecting smaller patches causes loss of spatial relationships and some nuclei could not be detected.

3.2 Fully Convolutional Network Approach

The general use of convolutional networks is classification of an image with corresponding labels. However, with many computer vision problem, especially with the biomedical vision problems, desired output should include localization which means each pixel needs to be assigned to a class label. Furthermore, thousands of training images are often inaccessible in biomedical studies. Fully convolutional

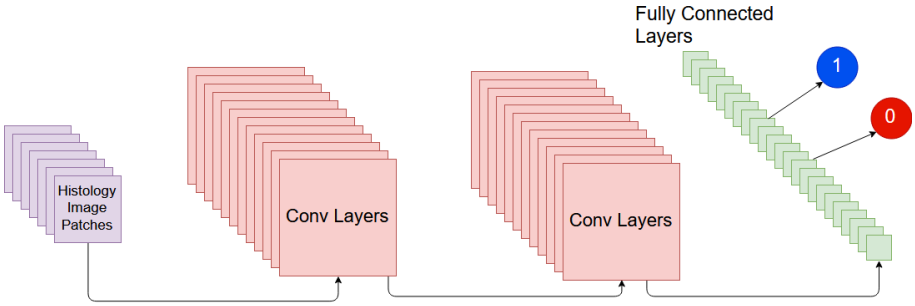


Figure 3.1 Basic Patch Classifier

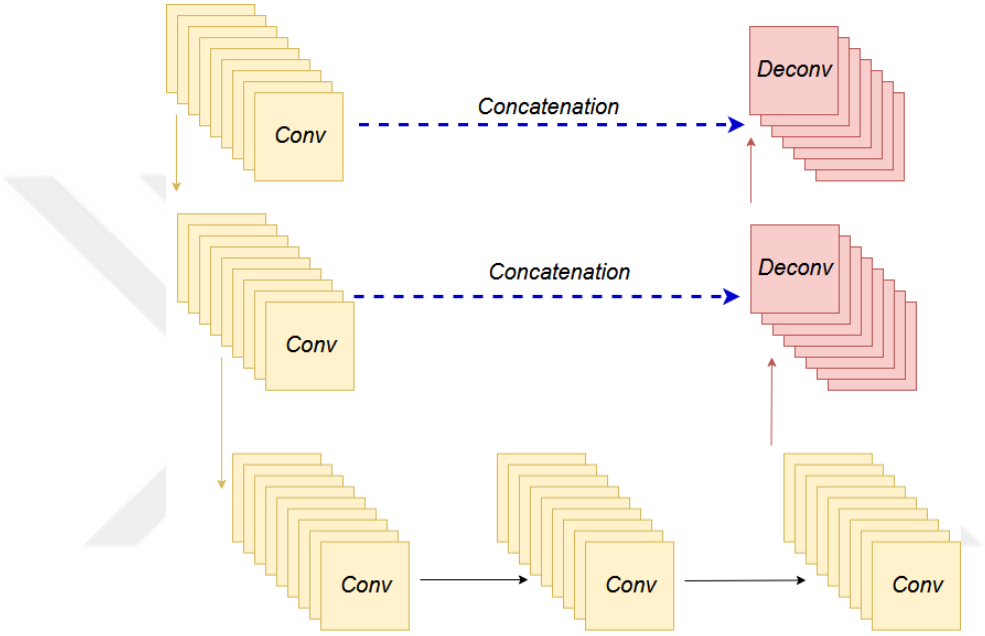


Figure 3.2 Basic Unet architecture

networks (FCN) [26] are proposed by Long et al., uses only convolutional layers, this way network generates a sparse segmentation map which means multi-class segmentation can be done. In this architecture, instead of classification of patches, each pixel is classified sparsely. Full image is fed to a fully convolutional network (FCN) and using pixel-wise cross-entropy loss function segmentation maps are generated.

This approach yielded good results and complexity of segmentation process is decreased. Ronneberger et al. [6] proposed Unet after FCN. Authors claim that using up-sampling layers and concatenation of down-sampling and up-sampling layers allows network to capture context and provide localization at the same time. The energy function is selected as pixel-wise softmax over feature map combined with cross-entropy loss by the authors.

The separation of objects for medical images are done with a morphological weighting process. If a pixel is between a two close object, then weight of this pixel increases as seen in (3.1). Here $d_1(x)$ is the distance of pixel to first object border, $d_2(x)$ is the distance of pixel to second object border. This allows more distinctive borders in segmentation map.

$$w(x) = w_c(x) + w_0.exp(-\frac{(d_1(x) + d_2(x))^2}{2\sigma^2}) \tag{3.1}$$

Another approach on loss function, instead of using pixel-wise loss, applying overlap aware loss function. Milletari et al. [91] proposed dice loss for segmentation of volumetric data. Dice loss is inspired by Dice metric which is not differentiable. Let X is the softmax output and Y is the label of corresponding pixel then dice coefficient is expressed as in equation (3.2). Dice loss must be differentiable to apply gradient descent. Therefore, equation is written as in (3.3).

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \tag{3.2}$$

$$D = \frac{2\sum_i^N X_i Y_i}{\sum_i^N X_i + \sum_i^N Y_i} \tag{3.3}$$

With dice loss function, there is no need of a weighting process to establish the balance between background and foreground pixels.

3.3 Performance Metrics for Nuclei Segmentation

Researchers have used different metrics to measure performance of proposed methods for nuclei segmentation. Some of the primitive methods that are used, are not enough measurement power to analyse results. More advanced methods yield different statistical information and analysis of proposed method becomes more meaningful. Most used performance metrics for nuclei segmentation includes accuracy, F1-Score, precision-recall and intersection over union.

3.3.1 Accuracy

Accuracy is the most primitive technique to measure performance of segmentation. It is the rate between correctly predicted pixels and ground truth. It measures how many pixels from ground truth are predicted true. Some authors include negative background pixels too. However, this is a false way to measure. Falsely measured

higher overall performance are caused by this method because, in some images there are lesser nuclei to perform segmentation which means overall performance seems to increase because of the truly predicted negative background pixels.

$$Accuracy = \frac{True \quad Prediction \quad of \quad Nuclei}{Ground \quad Truth} \tag{3.4}$$

3.3.2 Precision-Recall

In pattern recognition and classification, precision (also known as positive predictive value) is informative on how many of predicted instances are true predictions, while recall (also known as sensitivity) is informative on how many of relevant instances are predicted correctly. Both of these metrics evaluate relevancy of the model.

For nuclei segmentation problem which can be think as sparse binary classification, precision and recall yield better understanding of model performance. Precision measures, how much of predicted area is correctly overlapped with nuclei and recall measures how many of nuclei in total are predicted correctly (pixel wise). Precision and Recall is calculated as seen in (3.5) where TP is true positive, FP is false positive and FN is false negative.

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \end{aligned} \tag{3.5}$$

3.3.3 F1-Score

F1-Score is related to precision and recall metrics. It can be interpreted as more sensitive prediction process. It combines precision and recall scores instead of only using true positives for measuring overall performance [92]. F-measure do not use true negatives. This way, true negatives does not provide an increase to performance. F1-score is calculated as in (3.6).

$$F1-Score = 2\frac{Precision \cdot Recall}{Precision + Recall} \tag{3.6}$$

3.3.4 Intersection Over Union

Intersection over union (IOU) also called Jaccard Index, is mostly used to measure similarity and diversity of some set of instances. For computer vision problems, it is mostly used in object detection. It measures how predicted bounding-box overlaps to ground-truth bounding box. Using this metric for performance analysis of nuclei segmentation problem is not common.

$$IOU(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cup B| - |A \cap B|}{|A| + |B| - |A \cap B|} \tag{3.7}$$

Common performance metrics for nuclei segmentation are highlighted in this section. There are other metrics that are useful to measure segmentation performance. Metric that will be used can be selected according to problem structure and evaluation criteria.

4
Unsupervised Nuclei Segmentation using Generative Modeling

Most of deep nuclei segmentation approaches use supervised learning. With corresponding labels and sparse cross entropy loss function, gradient signals flow through deep neural network and weights of network updated until optimal loss surface is obtained. Using deep learning with an unsupervised approach is often difficult. Because of the need for some kind of supervision signal, there must be a valid objective for optimization process. Classification approach is not a valid way for unsupervised learning, thus objective is usually selected differently.

In computer vision, unsupervised deep learning models are usually generative models. Generative modeling can be interpreted as drawing samples from a data distribution. These samples might be a copy of a data instance or a unique instance that is not included in original dataset but carries all the properties of data distribution.

Deep generative models include Boltzman machines (BM) [93], Autoencoders (AE)[94], Variational autoencoders (VAE) [53] and Generative adversarial networks (GAN) [52]. Autoencoders are deterministic models thus, generated samples are copies of original data instances. Boltzmann machines and Variational autoencoders uses statistical inference models. BMs are difficult to train because of the intractable partition function in its loss term. VAEs uses ELBO loss to generate data instances and outputs are often blurry. GANs generate realistic data instances from data distribution. However training process of GAN is unstable because of the need for finding equilibrium according to the objective function based on game theory.

Nuclei segmentation can be thought of a domain translation problem which can be done with generative models. In this chapter two different generative modeling approach is proposed. In the first approach, nuclei segmentation problem solved with domain transfer. A labeled set of data translated to target domain to perform segmentation on a dataset from that domain. In second approach, nuclei segmentation

problem is thought as if it is an image-to-image translation problem. Histology images translated into label domain using generative adversarial networks.

4.1 Domain Transfer for Unlabeled Nuclei Segmentation

The purpose of artificial intelligence can be interpreted as learning a data or process, adapting to environment and solving problems. However, for deep learning agents, solving an unknown problem is very difficult. Even though some problems do not change, domain of data might be different such as nuclei segmentation problem. If an AI agent have true understanding of problem, then there must be a way to apply this knowledge to other domains.

Interestingly, problem solving skills of individuals depend on domain adaptation. When we see an unknown problem, we just adapt our knowledge base to problem. Using this analogy nuclei segmentation problem can be solved. A labeled set of data from a certain domain (type of stain, type of microscope, amount of magnification, tissue or organ type etc.) can be used to perform segmentation on a dataset from another domain. This can be done using generative models such as GANs and its variants.

Generative adversarial networks (GAN) proposed by Goodfellow et al. [52] is a generative model that learns a data distribution to generate realistic samples. In this structure, generative and discriminative networks play a zerosum game. Generative network tries to fool discriminative network and discriminative network tries to discriminate generated fake samples and real samples from the data distribution. In the end, system reaches to Nash equilibrium.

Relationship between generator and discriminator is similar to relationship between a counterfeiter that mints fake coins and the police who decide whether the coins are real or fake. Progressively, counterfeiter mints more realistic coins and tries to deceive the police. As the police gain experience, he/she learns to separate real coin from counterfeit one.

$$G : z \longrightarrow X', \quad X' \sim P_x \tag{4.1}$$

In Equation 4.1, the generative network (G) searches for the most optimal parameters that convert the uniformly distributed noise z to X' with P_x distribution. P_x refers

to data distribution. By this way, morphological features of samples may not be preserved, considering segmentation problem, they must be preserved.

Vanilla GAN does not capable of solving segmentation problem. However, using adversarial loss with other objectives is possible. Cycle-consistent generative adversarial networks is proposed for image-to-image translation problem. Adversarial loss (minimax loss) is used along with a reconstruction loss to change the style of an image and preserve its morphology.

$$\begin{aligned} G : X \longrightarrow Y', \quad Y' \sim P_y \\ F : Y \longrightarrow X', \quad X' \sim P_x \end{aligned} \tag{4.2}$$

In Cycle-GAN, a two-step conversion takes place from two different data distributions, as indicated by Zhu et al. [90] in equation 4.2. Let P_x be the distribution of histopathology images from domain X and let P_y be the distribution of histopathology images from domain Y. The first conversion is $X \xrightarrow{G} Y'$, where the histopathology image X is converted to a P_y distributed image and the morphological features are preserved, the second conversion is $Y' \xrightarrow{F} X'$ searches for the optimum parameters that are needed to convert P_y distributed Y' into a P_x distributed instance and morphological features are preserved.

$$\begin{aligned} \mathscr{L}_{GAN}(G, D_y, X, Y) = E_{y \sim p_{data}(y)}[log D_y(y)] \\ + E_{x \sim p_{data}(x)}[log(1 - D_y(G(x)))] \\ \mathscr{L}_{GAN}(F, D_x, Y, X) = E_{x \sim p_{data}(x)}[log D_y(x)] \\ + E_{y \sim p_{data}(y)}[log(1 - D_x(F(y)))] \end{aligned} \tag{4.3}$$

In equation (4.3) adversarial loss is applied over generator G and F. This process transforms the target image to desired domain. Equation (4.4) shows the reconstruction loss. It is applied to preserve morphology of transformed image.

$$\mathscr{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)}[\|F(G(x)) - x\|] + \mathbb{E}_{y \sim p_{data}(y)}[\|G(F(y)) - y\|] \tag{4.4}$$

As a result, the total loss function is shown in 4.9. The coefficient is $\lambda = 10$ as in original paper.

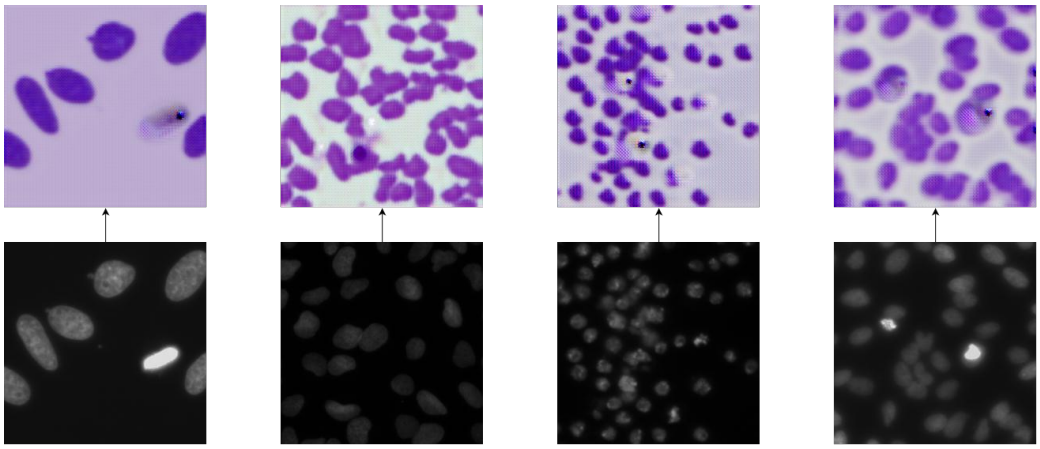


Figure 4.1 Domain transfer results on histology images

$$\begin{aligned}\mathscr{L}(G,F,D_x,D_y) = &\mathscr{L}_{GAN}(F,D_x,Y,X)\\ &+\mathscr{L}_{GAN}(G,D_y,X,Y)\\ &+\lambda\mathscr{L}_{cyc}(G,F)\end{aligned} \tag{4.5}$$

This specific loss function provides a transformation process. Let A be the domain which we have the nuclei annotations alongside with histopathology images and let B be the domain in which we want to perform segmentation and we do not have nuclei annotations. Goal is to translate the data from the domain-A to the domain-B to perform nuclei segmentation. Translated images are shown in figure **??**.

After translation, a new artificial set of data is obtained in target domain.A segmentation network is trained using artificial dataset. Unet architecture with pixel-wise cross entropy loss is used for this purpose.

Five DNNs are used in total. Four of them is used in CycleGAN to perform image-to-image translation and one is used for nuclei segmentation with generated dataset.

4.1.0.1 Generator's Unet Architecture

An architecture similar to autoencoder architecture was used in G and F generative networks. The only difference is that between encoder and decoder nine ResNet layers [84] are used. Leaky ReLU activation and instance normalization[95] are used. The slope of Leaky ReLU is chosen as 0.2.

Table 4.1 Generator architecture

Generator	Shape	Stride	Padding	Normalization	Activation
Input	256x256x3	-	-	-	-
Conv	7x7x64	1	Reflection	Instance norm.	ReLU
Conv	3x3x128	2	Reflection	Instance norm.	ReLU
Conv	3x3x256	2	Reflection	Instance norm.	ReLU
9 X Residual layer	3x3x256	1	Reflection	Instance norm.	ReLU
DeConv	3x3x128	1/2	Reflection	Instance norm.	ReLU
DeConv	3x3x64	1/2	Reflection	Instance norm.	ReLU
Conv	7x7x3	1	Reflection	Instance norm.	ReLU
Output	256x256x3	-	-	-	-

4.1.0.2 Discriminator's Architecture

For discriminators D_x and D_y, a four layer CNN is used. Beginning with 32, number of filters doubles in each layer. Leaky ReLU activation function is used for each layer. Filter size is chosen as 4x4. Stride parameter is chosen as 2.

Instead of using full image, we used 50x50 sized, randomly cropped image patches for discriminators[89]. This method yields less parameters and causes a stochastic process in discriminator. We have used 4x4 sized 32,64,128 and 256 filters in each layer respectively. Stride parameter is selected 2 for first three layers and Leaky-ReLU's slope is selected as $0,2$.

Table 4.2 Discriminator architecture

Patch Discriminator	Shape	Stride	Padding	Normalization	Activation
Input	50x50x3	-	-	-	-
Conv 4x4x64	2	Reflection	-	LeakyReLU	
Conv 4x4x128	2	Reflection	-	LeakyReLU	
Conv 4x4x256	2	Reflection	-	LeakyReLU	
Conv 4x4x512	1	Reflection	-	LeakyReLU	
Conv 4x4x1	1	Reflection	-	LeakyReLU	

4.1.0.3 Segmentation Network architecture

U-net architecture[6] is used for segmentation. U-net architecture is an autoencoder-like architecture that works by reducing the U-shaped size and increasing it again. Low and high-level features are used together with links between the down sampling and up-sampling layers.

In the Unet architecture that is used, 3x3 sized convolution kernels with one stride, are used. Each down-sampling layer in the network merges with the up-sampling layer of

the same shape. This combination is in the form of concatenation of activation maps. ReLU activation function in the layers and sigmoid activation function in the last layer were used. Cross entropy loss function with Adam optimizer is used. Learning rate selected as $1e-3$.

4.1.0.4 Training Process

The training was carried out over 100 epochs, with the learning rate starting at 1e-3 and decreasing exponentially in each epoch. However early stopping yields better results. The network was trained using one sample at each step. The samples from the artificially generated dataset were used when the network was being trained. Both grey-level dataset and RGB dataset was taken from the 2018 Data Science Bowl competition in Kaggle [96].

4.1.1 Results

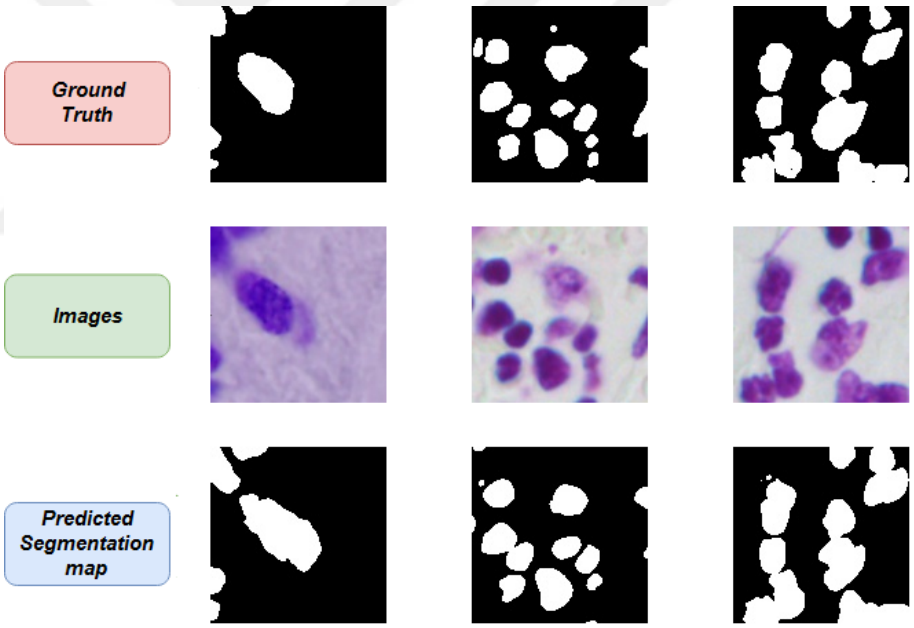


Figure 4.2 Segmentation results with generated data

The results with a single dataset in deep learning problems do not necessarily mean that the same performance will be achieved for the same problem. The distribution of the dataset needs to cover every situation and this is almost impossible. In the case of segmentation, the network that is trained with the dataset available may not display the same performance in another set of data, and it is time consuming to label each dataset separately.

In the cell segmentation problem, the morphological characteristics of the images are similar in different datasets from different domains. By maintaining this similarity, a dataset was obtained by translating a labeled dataset to target domain. The network, which was trained with artificially generated data, achieved a mIOU score of 72.6% for the segmentation problem. When the network is trained with the real dataset, the result is mIOU score of 82.3%. In the absence of the segmentation label, the proposed method has a useful success even if the performance achieved with the actual dataset is not exceeded.

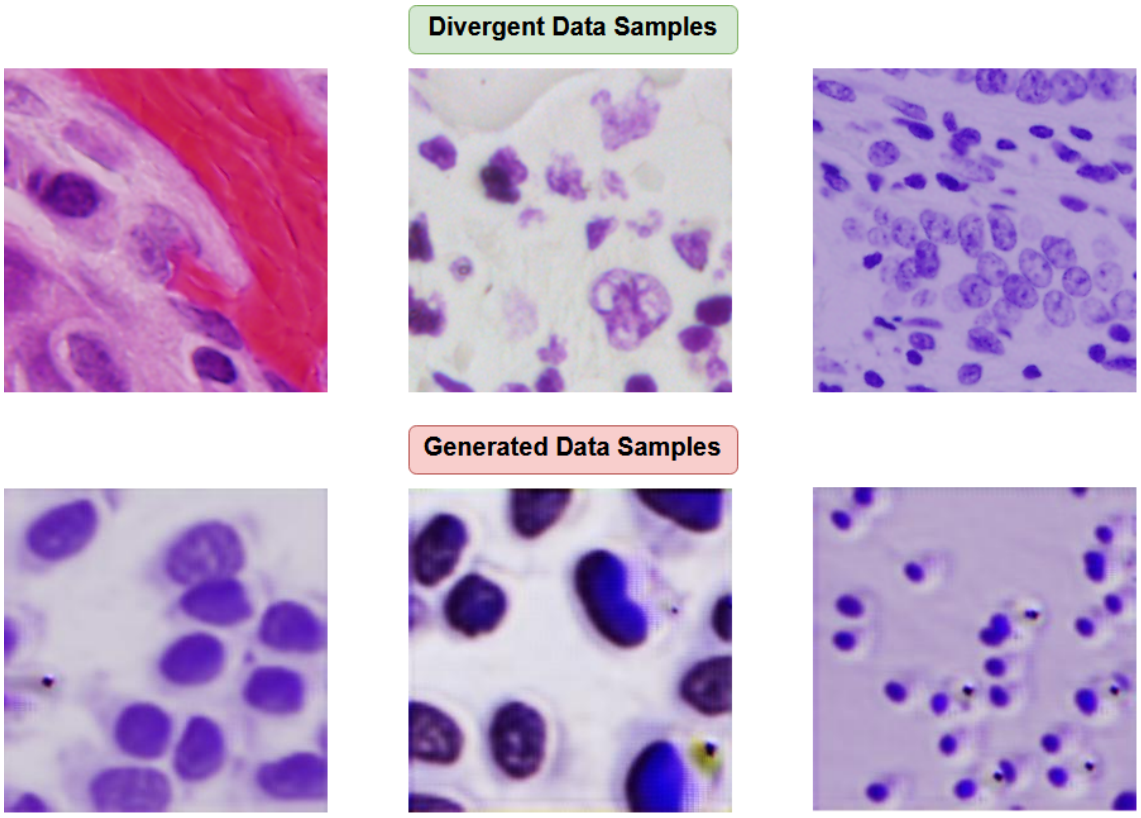


Figure 4.3 Divergent samples and generated images

There are some problems that lower the performance. Even though generated data are similar to real dataset, variations in generated data is very small which means generated data points does not span the entire space that the real data is distributed. This yields some problems with segmentation as some data points which are divergent samples, cannot be detected as nuclei. There is also problems with the generated data and labels. Some labels do not fully fit to data after transformation. Thus, performance of segmentation decreases significantly.

4.2 Unsupervised Nuclei Segmentation Using Random Annotations

In section 4.1 domain translation is discussed for segmentation and a dataset translated to target domain to perform segmentation. Using the same idea, domain of segmentation maps can be thought as another domain. Thus, a set of nuclei annotations are needed for domain translation.

Morphology of nuclei does not change with staining process, type of microscope or amount of magnification. This property provide a ground knowledge for unsupervised segmentation. A probabilistic model can be made for nuclei annotations. Nuclei have circular shape with parameters like biggest radius, smallest radius and angle. Furthermore, there is a relation between nuclei locations and shape. In the same slide, nuclei assumed to be Gaussian distributed. Location of each nucleus is uniformly distributed with a condition to decrease ratio of overlap. Using this priors, a nuclei annotation generator is modeled.

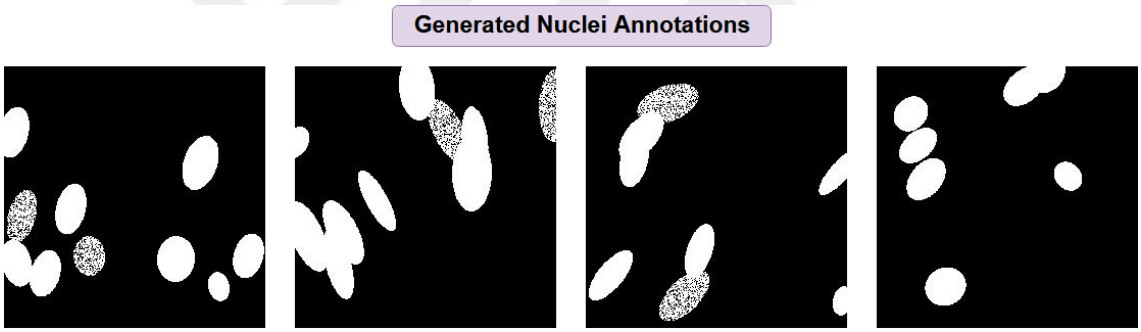


Figure 4.4 Generated random nuclei annotations

Domain-A represents the distribution of histopathology data and Domain-B represents output of random nuclei annotation model. Using İmage-to-image translation methodology, histopathology images can be translated into annotation domain. It is the similar to nuclei segmentation process, however nuclei annotations are not needed. Instead of real annotations, a probabilistic annotation model is used. The training process is same as presented in section 4.1.

Let P_A be the distribution of histopathology data (Domain-A) and let P_B be the distribution of segmentation maps (Domain-B). The first conversion is $X \xrightarrow{G} Y'$,

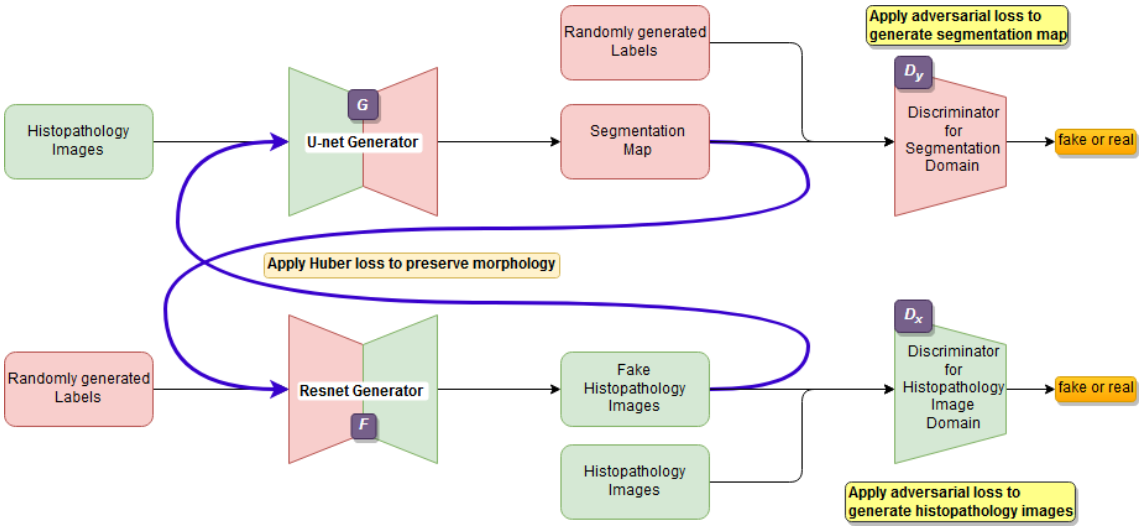


Figure 4.5 Diagram of CycleGAN for unsupervised segmentation

where the P_A distributed histopathology image X is converted to a P_B distributed segmentation map and the morphological features are preserved, the second conversion is $Y' \xrightarrow{F} X'$ searches for the optimum parameters that are needed to convert a randomly generated nuclei label Y' into a P_A distributed histopathology instance and morphological properties are preserved. This conversions are done cyclically and reconstruction loss is applied over each cycle.

Loss function used for conversion is the minimax game between a discriminative network and a generative network. Here, G and F functions are generative networks. Discriminator D_x and D_y networks decide whether the data comes from the P_A and P_B distributions, respectively.

$$\begin{aligned}\mathcal{L}_{GAN}(G, D_y, X, Y) = E_{y\sim p_B(y)}[logD_y(y)] \\ +E_{x\sim p_A(x)}[log(1-D_y(G(x)))]\end{aligned} \tag{4.6}$$

$$\begin{aligned}\mathcal{L}_{GAN}(F, D_x, Y, X) = E_{x\sim p_A(x)}[logD_x(x)] \\ +E_{y\sim p_B(y)}[log(1-D_x(F(y)))]\end{aligned} \tag{4.7}$$

The loss function (4.7) that is used in the GANs is also used for G and F networks. In addition, the Huber loss function [97] shown in equation (4.8) is also applied to provide cyclic consistency over $X' = F(G(X))$ and $Y' = G(F(Y))$ instead of L1 or L2 loss. Experiments showed that Huber loss provide more robust training process and more accurate reconstruction.

$$\begin{aligned}\mathscr{L}_{cyc}(G,F) =& \mathbb{E}_{x\sim p_A(x)}[\sqrt{1+(F(G(x))-x)^2}] \\ &+ \mathbb{E}_{y\sim p_B(y)}[\sqrt{1+(G(F(y))-y)^2}]\end{aligned} \tag{4.8}$$

As a result, the total loss function is shown in 4.9. The coefficient λ is selected 10 as in the original CycleGAN paper [90].

$$\mathscr{L}(G,F,D_x,D_y) = \mathscr{L}_{GAN}(F,D_x,Y,X) + \mathscr{L}_{GAN}(G,D_y,X,Y) + \lambda\mathscr{L}_{cyc}(G,F) \tag{4.9}$$

4.2.1 Random Nuclei Label Generation

We used a probabilistic model to generate nuclei annotations. We modeled nuclei with circular shapes using height, width, angle parameters. Let h ,w and Θ be the height, width and angle parameters respectively. We selected Gaussian distributed parameters. With each label generation, we redefined μ and σ parameters according to equation (4.10).

$$\begin{aligned} \mu_h \sim U(5,20) \quad \sigma_x \sim U(1,3) \\ \mu_w \sim U(5,25) \quad \sigma_y \sim U(1,4) \\ \mu_\Theta \sim U(0,180) \quad \sigma_\Theta \sim U(20,25) \\ h \sim N(\mu_h,\sigma_h) \quad w \sim N(\mu_w,\sigma_w) \quad \Theta \sim N(\mu_\Theta,\sigma_\Theta) \end{aligned} \tag{4.10}$$

In equation (4.10), height defined as x, width defined as y and angle of nuclei defined as θ.

Finally, we placed randomly generated nuclei annotations to image with a condition to ensure lower overlap ratio between each nucleus.

Random annotation model is very important for performance of segmentation process. If statistical properties of generated annotations is not close enough to statistics of nuclei, performance decreases significantly.

4.2.2 Post Processing

After proposed segmentation scheme, post processing required for more convenient results. Because of the unsupervised training, results are just the blind estimate of real annotations. Thus, for post processing, we estimated an average background using segmentation map and subtract it from original image as shown in figure **??**. Finally we applied Otsu's thresholding [7] to processed image.

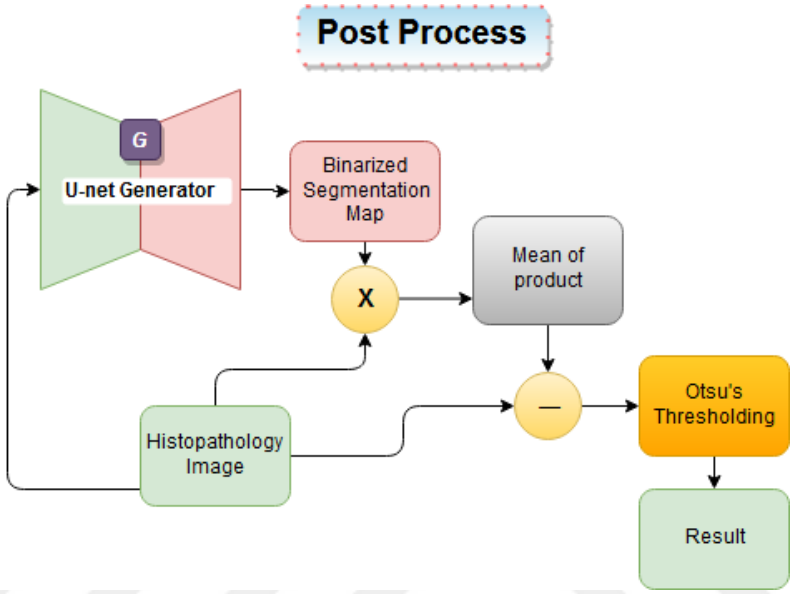


Figure 4.6 Diagram of Post Processing

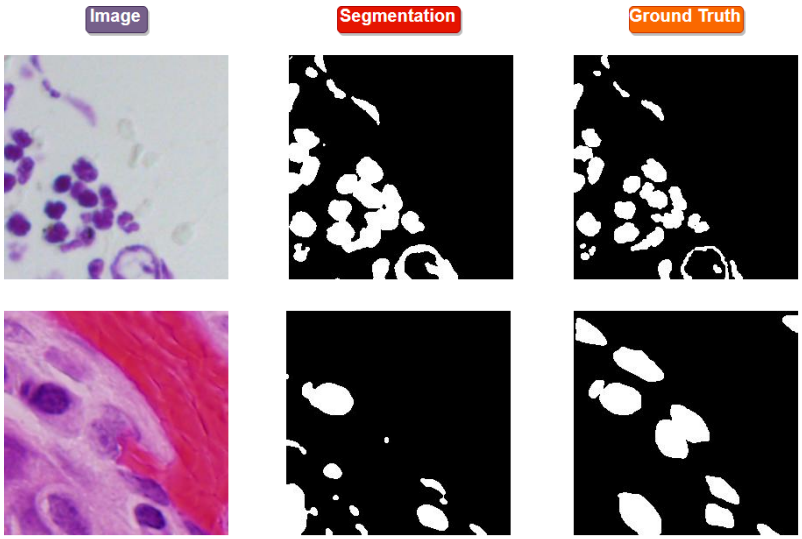


Figure 4.7 Experiment results

4.2.3 Experiments and Results

In this section, we conduct experiments on the 2018 Data Science Bowl competition dataset from Kaggle [96]. Dataset includes diverse histopathology images from different domains. We only used H&E stained RGB images. We cropped overlapping 256x256 sized patches from images and separated them equally for training and testing.

For each configuration, network is trained for 20 epochs with exponentially decaying learning rate starting with 0.0002.

In the first configuration, we used instance normalization [79] in convolution layers. This yields 0.716 F1 score. With this configuration, nuclei that are less brighter, cannot be found. We changed normalization layers to cosine normalization [98] which provides vectoral similarity approach. This improves recall score but lowers the precision score.

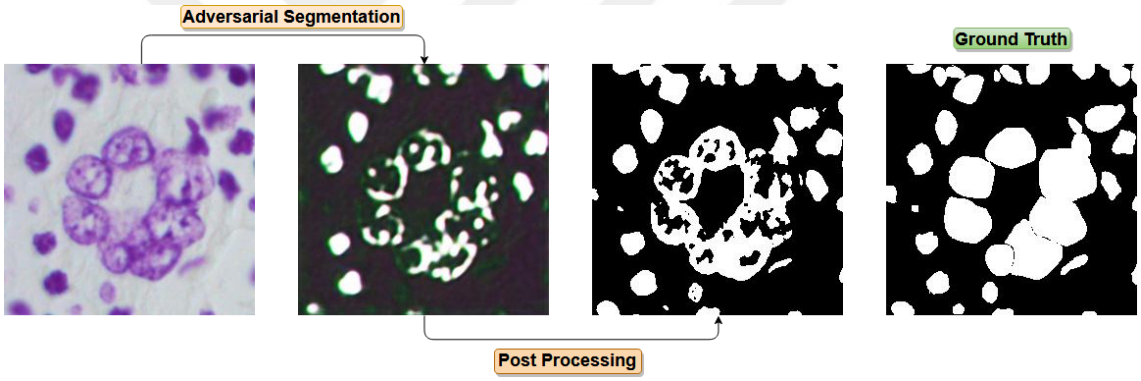


Figure 4.8 Segmentation structure

In the third configuration, dilated (atrous) convolutions are used in last three layers of segmentation network 'G'. Dilated convolutions are used for segmentation of small objects in literature. With this approach, smaller nuclei are detected and F1 score of segmentation improved to 0.733.

Table 4.3 Results

Normalization	Dilation	F1-Score	Precision	Recall
Instance norm.	False	0.716	0.708	0.724
Instance norm.	Dilation=3	0.733	0.728	0.738
Cosine norm.	False	0.719	0.699	0.741
Cosine norm.	Dilation=3	0.730	0.701	0.761
Post Processing	False	0.811	0.786	0.837

Last configuration is applying post processing to segmentation map. We applied post processing to all three configurations and performance with post processing increased dramatically. F1 score is approximately 0.81 for all configurations after post processing. Post processing method saturates for this set of data.

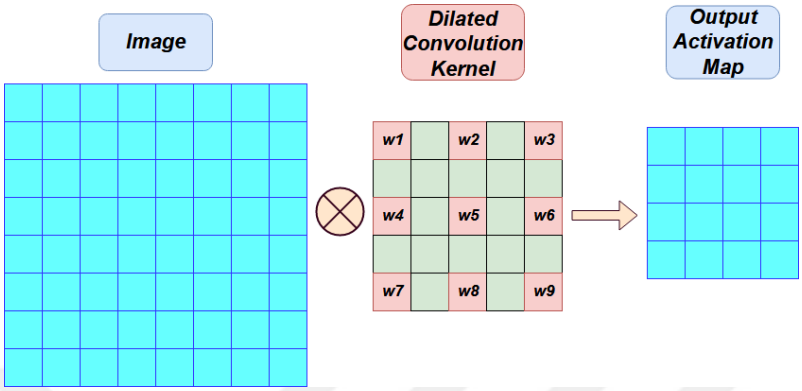


Figure 4.9 Dilated Convolution

In this chapter, proposed methods for nuclei segmentation are presented. Performance evaluations are done with metrics which are mentioned in Chapter 3. Detailed architectures can be found in Appendix-A.

5
Conclusion and Discussion

In this work, we examined generative modeling, especially the usage of cycle-consistent generative adversarial networks on nuclei segmentation problem. The purpose was to perform nuclei segmentation on histology images in an unsupervised manner. Proposed methods were based on the idea of domain adaptation. In first approach, annotated set of data from a certain domain was translated into target domain to train a segmentation network. In the second approach, a random nuclei annotation generator was modeled. Using this model, an artificial label domain was made and histology images translated to this domain. This process equal to perform unsupervised nuclei segmentation. After translation a post processing method was applied to segmentation maps.

Performance of proposed methods does not exceed supervised deep learning methods, however results are promising. In the first approach, networks were trained to represent the differences in domains of datasets which can be interpreted as understanding the underlying information of histology images. If network is able to learn this information, then solving nuclei segmentation problem in different domains becomes an easy task because the problem does not change with domain.

In practise, there is some problems with first approach. Because of nature of adversarial training, generative networks converges to a point that generated nuclei samples are very similar. This yields some problems with segmentation such as not detecting divergent nuclei. This can be solved using noise injection to generative networks. With this method, translation process becomes stochastic and generated samples variates. Nonetheless, some of the nuclei are very similar to the background. Some kind of supervision signal is needed to successfully detect these nuclei. Another problem is overlap of annotations and nuclei after translation. Even though the reconstruction loss is applied, nuclei do not fully overlap with corresponding annotations. This yields decrease in performance of segmentation.

In the second approach, performance extremely depend on random annotation

generation model. Statistics of generated annotations affect segmentation. If shapes of generated annotations is variate from original nuclei, network still detects the location of nuclei but segmentation is poorly made. This might seems as a problem but network might be able to separate different shaped nuclei, if appropriate statistical model is used.

Cyclic consistency loss (Reconstruction loss) for segmentation problem yields some problems. Signals coming from less brighter nuclei is suppressed with each gradient update because reconstruction of input image is accepted by discriminator without reconstructing less brighter nuclei. Even with different normalization layers, performance does not exceed a certain level. Yet, without reconstruction loss, segmentation map will not overlap to nuclei in histology images.

Using generative models for nuclei segmentation problem is a promising approach. Our method shows that generative networks can be used effectively for this problem. However there are space for improvements. Alongside of adversarial loss, different loss functions can be applied for reconstruction. Different distances can be used in adversarial loss term such as Wasserstein distance, this yields more realistic images and overall performance might increase.

A
Detailed Architectures

Table A.1 Generator Network: F

Layer	Input	Output	Kernel	Filters	Padding	Stride	Activation	Dilation	Normalization
Convolution	256x256x3	256x256x32	7x7	32	Same (Reflection)	1	ReLu	1	Instance Norm.
Convolution	256x256x32	128x128x64	3x3	64	Same (Reflection)	2	ReLu	1	Instance Norm.
Convolution	128x128x64	64x64x128	3x3	128	Same (Reflection)	2	ReLu	1	Instance Norm.
9 x Residual Layer	64x64x128	64x64x128	3x3	128	Same (Reflection)	1	ReLu	1	Instance Norm.
Deconvolution	64x64x128	128x128x64	3x3	64	Same (Reflection)	1/2	ReLu	1	Instance Norm.
Deconvolution	128x128x64	256x256x32	3x3	32	Same (Reflection)	1/2	ReLu	1	Instance Norm.
Convolution	256x256x32	256x256x3	7x7	3	Same (Reflection)	1	Tanh	1	-

Table A.2 Generator Network: G

Layer	Input	Output	Kernel	Filters	Padding	Stride	Activation	Dilation	Normalization
Convolution 1	256x256x3	256x256x32	7x7	32	Same (Reflection)	1	ReLu	1	Instance Norm.
Convolution 2	256x256x32	128x128x64	3x3	64	Same (Reflection)	2	ReLu	1	Instance Norm.
Convolution 3	128x128x64	64x64x128	3x3	128	Same (Reflection)	2	ReLu	1	Instance Norm.
9 x Residual Layer	64x64x128	64x64x128	3x3	128	Same (Reflection)	1	ReLu	1	Instance Norm.
Deconvolution 1	64x64x128	128x128x64	3x3	64	Same (Reflection)	1/2	ReLu	1	Instance Norm.
Concat. (Conv2&DeConv1)	128x128x64	128x128x128	-	-	-	-	-	-	-
Deconvolution 2	128x128x128	256x256x32	3x3	32	Same (Reflection)	1/2	ReLu	1	Instance Norm.
Concat. (Conv1 & DeConv2)	256x256x32	256x256x64	-	-	-	-	-	-	-
Convolution 3	256x256x64	256x256x3	7x7	3	Same (Reflection)	1	Tanh	1	-

Table A.3 Discriminator Network

Layer	Input	Output	Kernel	Filters	Padding	Stride	Activation	Dilation	Normalization
Convolution	50x50x3	25x25x64	4x4	64	Same	2	Leaky ReLU	-	-
Convolution	25x25x64	12x12x128	4x4	128	Same	2	Leaky ReLU	-	-
Convolution	12x12x128	6x6x256	4x4	256	Same	2	Leaky ReLU	-	-
Convolution	6x6x256	6x6x512	4x4	512	Same	1	Leaky ReLU	-	-
Convolution	6x6x512	6x6x1	4x4	1	Same	2	Leaky ReLU	-	-

Table A.4 Residual Layer

Layer	Input	Output	Kernel	Filters	Padding	Stride	Activation	Dilation	Normalization
Convolution 1	64x64x128	64x64x128	3x3	128	Same (Reflection)	1	ReLU	-	Instance Norm.
Convolution 2	64x64x128	64x64x128	3x3	128	Same (Reflection)	1	-	-	Instance Norm.
Add (Conv1 + conv2)	64x64x128	64x64x128	-	-	-	-	ReLU	-	-

References

[1] K.-H. Chow, R. E. Factor, and K. S. Ullman, "The nuclear envelope environment and its cancer connections," *Nature Reviews Cancer*, vol. 12, no. 3, p. 196, 2012.

[2] C. W. Elston and I. O. Ellis, "Pathological prognostic factors in breast cancer. i. the value of histological grade in breast cancer: Experience from a large study with long-term follow-up," *Histopathology*, vol. 19, no. 5, pp. 403–410, 1991.

[3] H. Irshad, A. Veillard, L. Roux, and D. Racoceanu, "Methods for nuclei detection, segmentation, and classification in digital histopathology: A review—current status and future potential," *IEEE reviews in biomedical engineering*, vol. 7, pp. 97–114, 2014.

[4] C. Jung, C. Kim, S. W. Chae, and S. Oh, "Unsupervised segmentation of overlapped nuclei using bayesian classification," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 12, pp. 2825–2832, 2010.

[5] A. M. Khan, H. El-Daly, and N. M. Rajpoot, "A gamma-gaussian mixture model for detection of mitotic cells in breast cancer histopathology images," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, IEEE, 2012, pp. 149–152.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[7] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[8] P. D. Wellner, "Adaptive thresholding for the digitaldesk," *Xerox, EPC1993-110*, pp. 1–19, 1993.

[9] N. Phansalkar, S. More, A. Sabale, and M. Joshi, "Adaptive local thresholding for detection of nuclei in diversity stained cytology images," in *2011 International Conference on Communications and Signal Processing*, IEEE, 2011, pp. 218–220.

[10] M. E. Plissiti and C. Nikou, "Overlapping cell nuclei segmentation using a spatially adaptive active physical model," *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4568–4580, 2012.

[11] G. Li, T. Liu, A. Tarokh, J. Nie, L. Guo, A. Mara, S. Holley, and S. T. Wong, "3d cell nuclei segmentation based on gradient flow tracking," *BMC cell biology*, vol. 8, no. 1, p. 40, 2007.

[12] J. Serra, *Image analysis and mathematical morphology*. Academic Press, Inc., 1983.

[13] J.-P. Thiran and B. Macq, "Morphological feature extraction for the classification of digital images of cancerous tissues," *IEEE Transactions on biomedical engineering*, vol. 43, no. 10, pp. 1011–1020, 1996.

[14] M. E. Plissiti, C. Nikou, and A. Charchanti, "Automated detection of cell nuclei in pap smear images using morphological reconstruction and clustering," *IEEE Transactions on information technology in biomedicine*, vol. 15, no. 2, pp. 233–241, 2011.

[15] L. Najman and M. Schmitt, "Watershed of a continuous function," *Signal Processing*, vol. 38, no. 1, pp. 99–112, 1994.

[16] C. Wählby, I.-M. Sintorn, F. Erlandsson, G. Borgefors, and E. Bengtsson, "Combining intensity, edge and shape information for 2d and 3d segmentation of cell nuclei in tissue sections," *Journal of microscopy*, vol. 215, no. 1, pp. 67–76, 2004.

[17] C. Jung and C. Kim, "Segmenting clustered nuclei using h-minima transform-based marker extraction and contour parameterization," *IEEE transactions on biomedical engineering*, vol. 57, no. 10, pp. 2600–2604, 2010.

[18] R. Kimmel, N. Kiryati, and A. M. Bruckstein, "Sub-pixel distance maps and weighted distance transforms," *Journal of Mathematical Imaging and Vision*, vol. 6, no. 2-3, pp. 223–233, 1996.

[19] F. Cloppet and A. Boucher, "Segmentation of overlapping/aggregating nuclei cells in biological images," in *2008 19th International Conference on Pattern Recognition*, IEEE, 2008, pp. 1–4.

[20] E. Cosatto, M. Miller, H. P. Graf, and J. S. Meyer, "Grading nuclear pleomorphism on histological micrographs," in *2008 19th International Conference on Pattern Recognition*, IEEE, 2008, pp. 1–4.

[21] P.-W. Huang and Y.-H. Lai, "Effective segmentation and classification for hcc biopsy images," *Pattern Recognition*, vol. 43, no. 4, pp. 1550–1563, 2010.

[22] J.-R. Dalle, H. Li, C.-H. Huang, W. K. Leow, D. Racoceanu, and T. C. Putti, "Nuclear pleomorphism scoring by selective cell nuclei detection.," in *WACV*, 2009.

[23] V.-T. Ta, O. Lézoray, A. Elmoataz, and S. Schüpp, "Graph-based tools for microscopic cellular image segmentation," *Pattern Recognition*, vol. 42, no. 6, pp. 1113–1125, 2009.

[24] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam, "Improved automatic detection and segmentation of cell nuclei in histopathology images," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 4, pp. 841–852, 2010.

[25] N. C. I. Std. (2008). Glioblastoma tcga dataset, [Online]. Available: `https://tcga-data.nci.nih.gov/docs/publications/tcga/` (visited on 03/31/2019).

[26] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[27] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2016, pp. 424–432.

[28] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer, 2013, pp. 411–418.

[29] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in neural information processing systems*, 2012, pp. 2843–2851.

[30] A. A. Cruz-Roa, J. E. A. Ovalle, A. Madabhushi, and F. A. G. Osorio, "A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2013, pp. 403–410.

[31] C. D. Malon and E. Cosatto, "Classification of mitotic figures with convolutional neural networks and seeded blob features," *Journal of pathology informatics*, vol. 4, 2013.

[32] H. Wang, A. C. Roa, A. N. Basavanhally, H. L. Gilmore, N. Shih, M. Feldman, J. Tomaszewski, F. Gonzalez, and A. Madabhushi, "Mitosis detection in breast cancer pathology images by combining handcrafted and convolutional neural network features," *Journal of Medical Imaging*, vol. 1, no. 3, p. 034 003, 2014.

[33] A. Ferrari, S. Lombardi, and A. Signoroni, "Bacterial colony counting by convolutional neural networks," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2015, pp. 7458–7461.

[34] A. Shkolyar, A. Gefen, D. Benayahu, and H. Greenspan, "Automatic detection of cell divisions (mitosis) in live-imaging microscopy images using convolutional neural networks," in *2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, IEEE, 2015, pp. 743–746.

[35] Y. Song, L. Zhang, S. Chen, D. Ni, B. Lei, and T. Wang, "Accurate segmentation of cervical cytoplasm and nuclei based on multiscale convolutional network and graph partitioning," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 10, pp. 2421–2433, 2015.

[36] Y. Xie, X. Kong, F. Xing, F. Liu, H. Su, and L. Yang, "Deep voting: A robust approach toward nucleus localization in microscopy images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2015, pp. 374–382.

[37] S. U. Akram, J. Kannala, L. Eklund, and J. Heikkilä, "Cell proposal network for microscopy image analysis," in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2016, pp. 3199–3203.

[38] S. Albarqouni, C. Baur, F. Achilles, V. Belagiannis, S. Demirci, and N. Navab, "Aggnet: Deep learning from crowds for mitosis detection in breast cancer histology images," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1313–1321, 2016.

[39] S. Bauer, N. Carion, P. Schüffler, T. Fuchs, P. Wild, and J. M. Buhmann, "Multi-organ cancer classification and survival analysis," *arXiv preprint arXiv:1606.00897*, 2016.

[40] H. Chen, X. Wang, and P. A. Heng, "Automated mitosis detection with deep regression networks," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, IEEE, 2016, pp. 1204–1207.

[41] A. Janowczyk, S. Doyle, H. Gilmore, and A. Madabhushi, "A resolution adaptive deep hierarchical (radhical) learning scheme applied to nuclear segmentation of digital pathology images," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 6, no. 3, pp. 270–276, 2018.

[42] M. N. Kashif, S. E. A. Raza, K. Sirinukunwattana, M. Arif, and N. Rajpoot, "Handcrafted features with convolutional neural networks for detection of tumor cells in histology images," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, IEEE, 2016, pp. 1029–1032.

[43] Y. Mao and Z. Yin, "A hierarchical convolutional neural network for mitosis detection in phase-contrast microscopy images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2016, pp. 685–692.

[44] M. Mishra, S. Schmitt, L. Wang, M. K. Strasser, C. Marr, N. Navab, H. Zischka, and T. Peng, "Structure-based assessment of cancerous mitochondria using deep networks," in *2016 IEEE 13Th International Symposium on Biomedical Imaging (ISBI)*, IEEE, 2016, pp. 545–548.

[45] H. T. H. Phan, A. Kumar, J. Kim, and D. Feng, "Transfer learning of a convolutional neural network for hep-2 cell image classification," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, IEEE, 2016, pp. 1208–1211.

[46] K. Sirinukunwattana, S. E. A. Raza, Y.-W. Tsang, D. R. Snead, I. A. Cree, and N. M. Rajpoot, "Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1196–1206, 2016.

[47] M. Veta, P. J. Van Diest, and J. P. Pluim, "Cutting out the middleman: Measuring nuclear area in histopathology slides without segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2016, pp. 632–639.

[48] W. Xie, J. A. Noble, and A. Zisserman, "Microscopy cell counting and detection with fully convolutional regression networks," *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization*, vol. 6, no. 3, pp. 283–292, 2018.

[49] F. Xing, Y. Xie, and L. Yang, "An automatic learning-based framework for robust nucleus segmentation," *IEEE transactions on medical imaging*, vol. 35, no. 2, pp. 550–566, 2016.

[50] J. Xu, L. Xiang, Q. Liu, H. Gilmore, J. Wu, J. Tang, and A. Madabhushi, "Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images," *IEEE transactions on medical imaging*, vol. 35, no. 1, pp. 119–130, 2016.

[51] L. Yang, Y. Zhang, I. H. Guldner, S. Zhang, and D. Z. Chen, "3d segmentation of glial cells using fully convolutional networks and k-terminal cut," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2016, pp. 658–666.

[52] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[53] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[54] B. Hu, Y. Tang, I. Eric, C. Chang, Y. Fan, M. Lai, and Y. Xu, "Unsupervised learning for cell-level visual representation with generative adversarial networks," *IEEE journal of biomedical and health informatics*, 2018.

[55] S. Hana, S. Leea, C. Fua, P. Salamab, K. W. Dunnc, and E. J. Delpa, "Nuclei counting in microscopy images with three dimensional generative adversarial networks,"

[56] I. J. Stephan, A. M. Reichmuth, S. Girardin, H. Han, F. Stauffer, A. Bonnin, M. F. M. Stampanoni, J. Voros, and C. Forro, "Udct: Unsupervised data to content transformation with histogram-matching cycle-consistent generative adversarial networks," *bioRxiv*, p. 563 734, 2019.

[57] L. Hou, A. Agarwal, D. Samaras, T. M. Kurc, R. R. Gupta, and J. H. Saltz, "Unsupervised histopathology image synthesis," *arXiv preprint arXiv:1712.05021*, 2017.

[58] Z. Xu, C. F. Moro, B. Bozóky, and Q. Zhang, "Gan-based virtual re-staining: A promising solution for whole slide image analysis," *arXiv preprint arXiv:1901.04059*, 2019.

[59] M. T. Shaban, C. Baur, N. Navab, and S. Albarqouni, "Staingan: Stain style transfer for digital histological images," *arXiv preprint arXiv:1804.01601*, 2018.

[60] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[61] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[62] B. Widrow and M. E. Hoff, "Adaptive switching circuits," STANFORD UNIV CA STANFORD ELECTRONICS LABS, Tech. Rep., 1960.

[63] B. Widrow and F. W. Smith, "Pattern-recognizing control systems," *Computer and Information Sciences (COINS) Proceedings*, 1964.

[64] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[65] J. L. McClelland, D. E. Rumelhart, P. R. Group, *et al.*, *Parallel distributed processing*. MIT press Cambridge, MA: 1987, vol. 2.

[66] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *et al.*, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

[67] Y. Le Cun, L. Jackel, B. Boser, J. Denker, H. Graf, I. Guyon, D. Henderson, R. Howard, and W. Hubbard, "Handwritten digit recognition: Applications of neural network chips and automatic learning," *IEEE Communications Magazine*, vol. 27, no. 11, pp. 41–46, 1989.

[68] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[69] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[70] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat," *Journal of neurophysiology*, vol. 28, no. 2, pp. 229–289, 1965.

[71] D. H. Hubel and T. N. Wiesel, "Ferrier lecture-functional architecture of macaque monkey visual cortex," *Proc. R. Soc. Lond. B*, vol. 198, no. 1130, pp. 1–59, 1977.

[72] S. University. (2019). Cs231n: Convolutional neural networks for visual recognition, [Online]. Available: `http://cs231n.github.io/understanding-cnn` (visited on 03/31/2019).

[73] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, 2013, p. 3.

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[75] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[76] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[77] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 972–981.

[78] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[79] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[80] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[81] Y. Wu and K. He, "Group normalization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[82] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[83] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[84] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[85] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in neural information processing systems*, 2016, pp. 2234–2242.

[86] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

[87] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[88] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.

[89] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint*, 2017.

[90] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint*, 2017.

[91] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 Fourth International Conference on 3D Vision (3DV)*, IEEE, 2016, pp. 565–571.

[92] D. M. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[93] R. Salakhutdinov and G. Hinton, "Deep boltzmann machines," in *Artificial intelligence and statistics*, 2009, pp. 448–455.

[94] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[95] X. Huang and S. J. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization.," in *ICCV*, 2017, pp. 1510–1519.

[96] Kaggle. (2018). 2018 data science bowl, [Online]. Available: `https://www.kaggle.com/c/data-science-bowl-2018/overview` (visited on 03/31/2019).

[97] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*, Springer, 1992, pp. 492–518.

[98] C. Luo, J. Zhan, X. Xue, L. Wang, R. Ren, and Q. Yang, “Cosine normalization: Using cosine similarity instead of dot product in neural networks,” in *International Conference on Artificial Neural Networks*, Springer, 2018, pp. 382–391.

Publications From the Thesis

Contact Information: onurck@yildiz.edu.tr

Conference Papers

1. Onur C. Koyun, T. Yıldırım, "Etiketsiz Verilerde Hücre Bölütleme", Elektrik-Elektronik ve Biyomedikal Mühendisliği Konferansı (ELECO), 2018, 538-542