Semantic Segmentation of Glomeruli in Human Kidney Tissue Images

Authors

Sheikh Sakib Ishrak, 160041038

Md. Sakif Khan, 160041039

Ahmad Imam, 160041054

Supervisor

Md. Redwan Karim Sony

Lecturer, Department of Computer Science and Engineering (CSE)

Islamic University of Technology (IUT), OIC

A thesis submitted in partial fulfilment of the requirements

for the degree of B. Sc. Engineering in Computer Science and Engineering

Academic Year: 2019-2020

Department of Computer Science and Engineering (CSE)

Islamic University of Technology (IUT)

A Subsidiary Organ of the Organization of Islamic Cooperation (OIC)

Dhaka, Bangladesh

March, 2021

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by under the supervision of **Md. Redwan Karim Sony**, Lecturer of the Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Sheikh Sakib Ishrak
30.03.2021

Sheikh Sakib Ishrak

Student ID - 160041038

Sakif
30.03.2021

Md. Sakif Khan

Student ID - 160041039

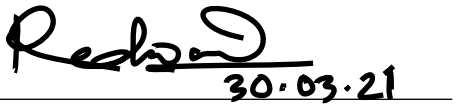
Ahmad Imam
30.3.21

Ahmad Imam

Student ID - 160041054

Approved By:

Supervisor:



Md. Redwan Karim Sony

Lecturer

Department of Computer Science and Engineering (CSE)

Islamic University of Technology (IUT), OIC

Acknowledgement

We would like to express our grateful appreciation for **Mr. Md. Redwan Karim Sony**, Lecturer, Department of Computer Science & Engineering, IUT for being our adviser and mentor. His motivation, suggestions and insights for this research have been invaluable. Without his support and proper guidance this research would never have been possible. His valuable opinion, time and input provided throughout the thesis work, from first phase of thesis topics introduction, subject selection, proposing algorithm, modification till the project implementation and finalization which helped us to do our thesis work in proper way. We are really grateful to him.

Abstract

For reliable disease diagnosis in renal pathology, accurate glomerular microscopic medical image segmentation is important. In this study, we work on a pixel-level labeled glomerular microscopic medical image segmentation dataset, the HuBMAP kidney dataset, and improve a novel pipeline for implementing automatic segmentation of glomerular microscopic medical images. In our thesis work, we have tried using many variations of segmentation models, encoders, feature extractors and explored their potentials for semantic segmentation of glomeruli. In our proposed approach, we used the network architecture which gives the most promising result on the dataset, consisting of LinkNet with EfficientNet as modified encoder block, pretrained on ImageNet. Here the Compound Scaling provides better performance without compromising the efficiency. This pipeline outperformed other models that we have experimented with and allowed better performance than previous non deep learning based methodologies of glomerular identification.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Overview

A tuft of small blood vessels (capillaries) situated at the beginning of a nephron in the kidney is known as a glomerulus (plural glomeruli). It is the kidney's basic filtration unit[10]. The glomerulus' main task is to filter plasma into glomerular filtrate, which then flows down the length of the nephron tubule to form urine. Disease damage to the glomerulus may enable red blood cells, white blood cells, platelets, and blood proteins including albumin and globulin to move through the glomerular filtration barrier. The glomerular filtration barrier, which is responsible for the filtration of blood into urine, is affected by disturbances in the glomerular structure. Proteinuria, a condition marked by the presence of excessive proteins in the urine and a common indicator of a variety of renal diseases, is the product of this disturbance. As a result, when analyzing a renal biopsy, specialists focus on the histological damage within the glomeruli to distinguish kidney diseases[11]. Furthermore, numerous studies have focused on various compartments within the glomeruli, such as the mesangial matrix[12], capillary walls[13], and podocytes[14, 15], in order to better understand the changes that occur within the kidney during various stages of disease. However, in order to detect these compartments, it is important to first accurately define the glomerular boundaries within the whole slide picture (WSI)[14].

1.2 Problem Statement

Due to its complex nature and intense variations in size and shape within the renal section, automated glomerular segmentation remains a challenge today[16]. In vivo, the glomerulus swells during hypertension[17], hypertrophy[18], and diabetes[19], despite being consistent under normal conditions. Furthermore, due to differences in sectioning angles, manual tissue sectioning induces variations in glomerulus sizes. Aside from that, the task is made more difficult by the differences in stain-

ing intensities. Because of the inconsistencies in the geometrical parameters, developing a single robust algorithm capable of detecting and segmenting all of the glomeruli within a tissue section is difficult.

Our goal is to obtain proper identification of glomeruli in human kidney tissue images by semantic segmentation.

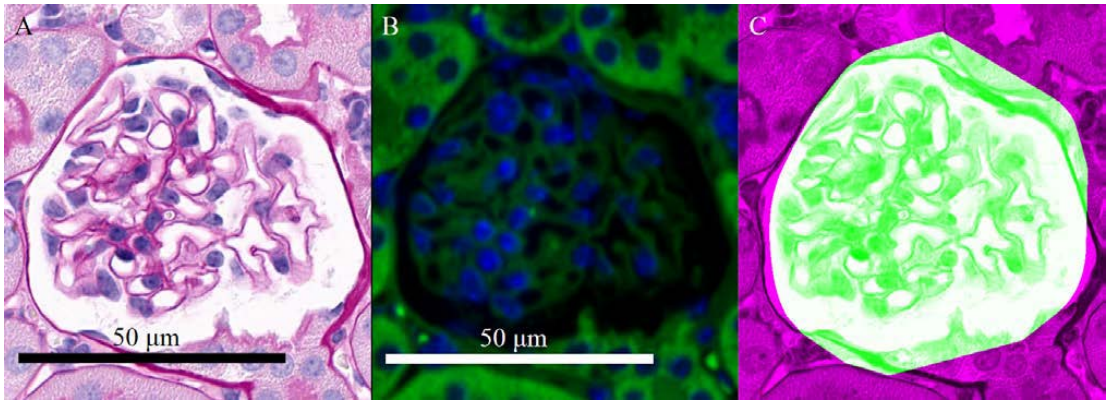


Figure 1: [1](A) Periodic acid-Schiff image patch containing a glomerulus. (B) Corresponding immunofluorescence image patch. (C) Mask generated by the Butterworth band-pass filter

1.3 Motivation

According to our best estimations, the Earth has over 7 billion inhabitants and the Milky Way galaxy has 300 billion stars. The adult human body, on the other hand, comprises 37 trillion cells. It's a mammoth task to figure out what these cells do and how they interact. If we can better understand cellular behavior, it will have an effect on many aspects of human health. Automated reliable methods of glomerular identification will ultimately improve accuracy and speed of kidney research. This process will help us in finding changes within the kidney during different stages of diseases. Complex nature and intense variations in size and shape of glomeruli makes it a very interesting research topic. The current clinical practice for glomerular detection involves the manual observation of histologically stained biopsied tissues under a standard bright-field microscope by

a pathologist[20], which is time consuming, tedious, subjective and requires expertise.Because of these problems there's absence of a highly accurate automatic segmentation process of Glomeruli.

The developed method could thereby aid in renal disease diagnosis and tracking of disease progression and therapeutic response by alleviating the burden of manual detection of glomeruli within the tissue. It would also aid in the development of a tool, capable of rapidly generating glomerular databases by detecting and segmenting them from whole slide renal tissue images, which are crucial for training neural networks. Furthermore, it could also be used to extract various compartments within the glomerulus, such as the mesangium, the podocytes and the capillary walls which are the focus of several stus.

1.4 Research Challenges

The primary research challenge of any medical image is the scarcity of accurately labelled data. Moreover, these data have to be individually double checked and labelled by a specialist of the corresponding sector. The amount of data to research on is very low compared to other fields of studies. Variations of medical data are a huge factor due to the fact that there are people spread out all around the world, and their geographical location, habits, etc. affect the types of diseases they have. One of the most common research challenges is that medical data has to be accurate to the point which amounts to large sized pictures. Even a single slide of image may be of several gigabytes in size. So, these data need a high computation power to process and analyze which isn't available everywhere.

To put it simply :

- Firstly, Scarcity of accurately labeled data by specialist.
- Secondly, Variation of Medical Data based on geographical location and health conditions.
- Thirdly, Amount of properly labeled dataset is relatively low.

- Finally, For microscopic renal data, a single slide scan can consume several gigabytes of storage. Significant amount of computational resources is needed.

1.5 Application

Many diseases affect kidney function by targeting the glomeruli, the tiny units in the kidney that clean the blood. Glomerular diseases encompass a wide range of disorders with a wide range of genetic and environmental causes, but they can be classified into two groups:

- Glomerulonephritis is an inflammation of the kidney's membrane tissue, which acts as a barrier, removing wastes and excess fluid from the blood.
- Glomerulosclerosis is a disorder in which the tiny blood vessels in the kidney scar or harden.

While the causes of glomerulonephritis and glomerulosclerosis vary, both can result in kidney failure.

However, the accurate diagnosis of renal disease solely depends on detecting the glomeruli properly and efficiently. This research will also help in finding changes within the kidney during different stages of diseases.

1.6 Contribution

In our thesis work, we have tried using many variations of segmentation models, encoders, feature extractors and explored their potentials for semantic segmentation of glomeruli.

- In our proposed approach, we used the network architecture which gives the most promising result on the HuBMAP dataset which is a variation of LinkNet, where we replaced the default encoder of LinkNet with EfficientNet which is already pretrained on ImageNet database.

- This model achieved a DICE score of 82.7% when tested on the dataset.
- Here the Compound Scaling provides better performance without compromising the efficiency.
- This pipeline outperformed other models that we have experimented with and allowed better performance than previous non deep learning based methods of glomerular identification.

1.7 Organization of Thesis

The rest of the report is organized as follows:

- Chapter 2 shows our background study regarding our research.
- Chapter 3 gives a literature review discussing different approaches for medical image segmentation used over the years.
- Chapter 4 introduces our proposed methods for efficient and accurate segmentation of glomeruli.
- Chapter 5 describes in detail the different experiments we performed and their result analysis and comparisons.
- Chapter 6 presents an overall conclusion of our thesis and discusses our future plan of work.

2 Background Study

2.1 Semantic Image Segmentation

Image segmentation is a computer vision task in which specific regions of an image are labelled based on what's being seen. In semantic Image Segmentation, also known as dense prediction, each pixel of an image is labelled with a corresponding class of what is being represented.

In computer vision pixel wise dense prediction is the task of predicting a label for each pixel in the image. Convolutional neural networks achieve good performance on this task, while being computationally efficient[21].

Instances of the same class are not don't distinguished in semantic segmentation, we just consider the category of each pixel. In other words, if the input image includes two objects of the same type, the segmentation map will not automatically identify them as separate objects.A particular type of model, known as instance segmentation models, distinguishes between different objects of the same class.

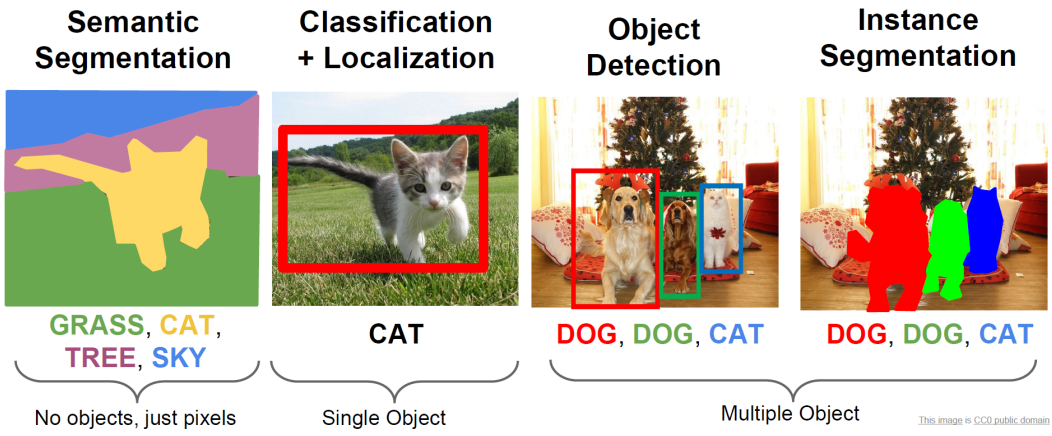


Figure 2: [2]Computer Vision Tasks

The expected output in semantic segmentation are not just labels and bounding box parameters. In semantic segmentation, the predicted output is more than just labels and bounding box parameters. The output is a high-resolution image

(typically the same size as the input) with each pixel categorized into a different class. As a consequence, it's a pixel-by-pixel image classification.

Before the advent of deep learning, classical machine learning techniques like SVM, Random Forest, K-means Clustering were used to solve the problem of image segmentation. But as with most of the image related problem statements deep learning has worked comprehensively better than the existing techniques and has become a norm now when dealing with Semantic Segmentation.

2.2 Convolutional Neural Network

Convolutional neural networks (CNN) are widely used in image recognition applications of machine learning. Convolutional neural networks provide an advantage over feed-forward networks because they are capable of considering locality of features. CNN image classifications take an input image, process it and classify it under certain categories. Computers see an input image as an array of pixels and it depends on the image resolution.

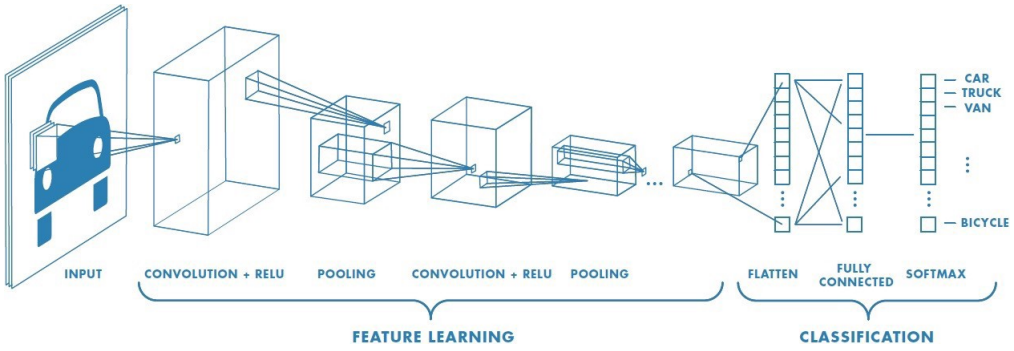


Figure 3: [3]Convolulational Newural Network

Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernals), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The figure is a complete flow of CNN to process an input image and classifies the objects based on values.

2.2.1 Convolution Layers

A convolution layer provides a window through which we can inspect a subset of the image and then scans the entire image when looking through it. Since it generates an output image that focuses exclusively on the regions of the image that exhibit the function it was looking for, this window is also known as a filter/kernel. A feature map is a representation of the output of a convolution.

By learning image features with small squares of input data, convolution maintains the relationship between pixels. It's a mathematical operation with two inputs: an image matrix and a filter or kernel. Convolution of an image with various filters may be used to perform operations such as edge detection, blurring, and sharpening.

There are two key advantages of performing convolutions on images rather than connecting each pixel to the neural network units:

1. Reduces the number of parameters we need to learn. We just need to learn the weights of the filter, rather than the weights connecting each input pixel (which usually is a lot smaller than the input image).

2. Locality is maintained. The image matrix does not need to be flattened into a vector, so the relative locations of the image pixels are retained. If we represent the picture as a long string of numbers, we lose the insights.

2.2.2 Depth

Depth corresponds to the number of filters we use for the convolution operation. In the network shown in Figure, convolution of the original image is performed using three distinct filters, thus producing three different feature maps.

2.2.3 Padding

Filters do not always exactly match the input image. There are two possibilities:

1. To suit the image, padding it with zeros (zero-padding) is required.

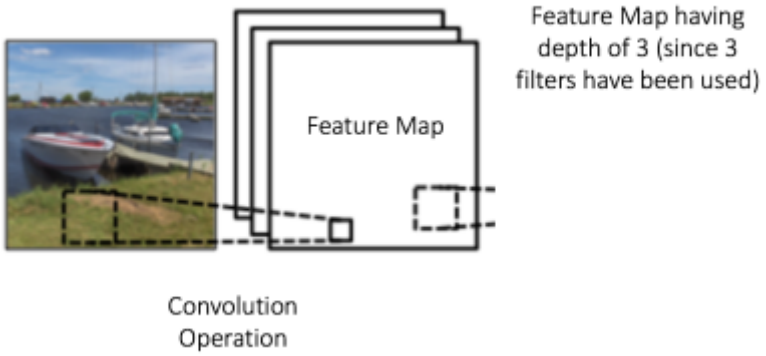


Figure 4: [4]A simple ConvNet.

2. Delete the portion of the image where the filter didn't operate. This is known as true padding, and it just holds the image's valid pieces.

The image's dimension is diminished when it is convolutioned. The image size shrinks and gradually becomes too small to be useful if the input is passed through several convolution layers without padding. Same-padding is the method of applying zero padding to an image such that the output has the same width and height as the input.

2.2.4 Strides

The number of pixels the filter will pass over the input matrix each time is referred to as the stride. We shift the filter to n pixels at a time when the stride is n.

2.2.5 Pooling Layer

Pooling reduces the number of parameters that the network must learn by performing nonlinear downsampling on the output. Spatial Pooling (also known as downsampling or subsampling) reduces the dimensionality of each function map while preserving the most relevant details.

Different forms of spatial pooling exist: maximum, average, sum, and so on. In case of Max Pooling, a spatial neighborhood is defined and the largest element from the rectified feature map within that window is taken. Instead of taking the

largest element also take the average (Average Pooling) or sum of all elements in that window.

In practice, Max Pooling has been shown to work better. When Max Pooling is used, a spatial neighborhood is specified, and the largest element from the rectified feature map within that window is selected. Take the average (Average Pooling) or total of all elements in that window instead of the largest element. Max Pooling has been shown to perform better in practice.

2.2.6 Non Linearity (ReLU)

ReLU stands for Rectified Linear Unit for a non-linear operation. The output is f(x) = max(0,x).

The aim of ReLU is to add non-linearity to our ConvNet. Since the data we want our ConvNet to learn in the real world is non-negative linear values. In addition to ReLU, other nonlinear functions such as tanh and sigmoid can be used. ReLU is mostly used because it outperforms the other two in terms of results.

2.2.7 Fully Connected Layer

This layer takes an input volume (whatever the output of the conv, ReLU, or pool layer before it is) and outputs an N-dimensional vector, where N is the number of classes from which the program must choose. The fully connected layer looks at the performance of the previous layer (which should reflect the activation maps of high level features) and decides which features are most associated with a specific class. Basically, an FC layer looks at the high-level features that are most closely correlated with a particular class and assigns weights to them such that when the products of the weights and the previous layer are calculated, the correct probabilities for the various classes are obtained.

2.3 Fully Convolutional Networks for Semantic Segmentation

A CNN's basic architecture consists of a few convolutional and pooling layers, followed by a few fully connected layers. According to the paper published in 2014[5], the final fully connected layer can be thought of as performing a 1x1 convolution that covers the entire region.

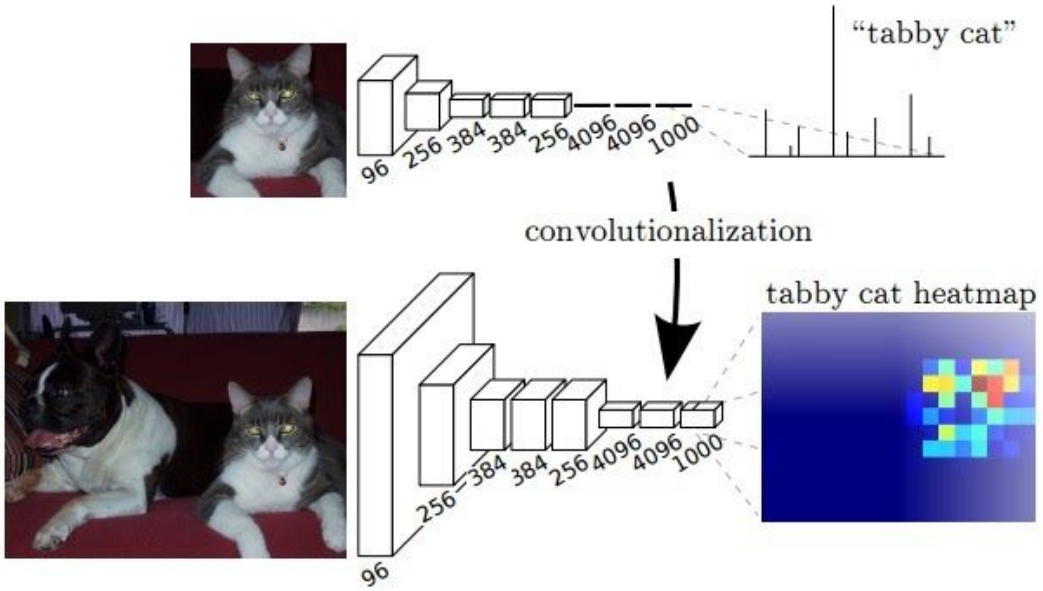


Figure 5: [5]Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss produces an efficient machine for end-to-end dense learning.

As a result, the final dense layers can be replaced with a convolution layer and the result will be the same. However, the benefit of doing so now is that the size of the input does not have to be set. When dealing with dense layers, the size of the input is small, so if a different size input is needed, it must be resized. This restriction is eliminated when a dense layer is replaced with convolution.

Also, when a larger image is used as an input, the result is a feature map rather than a class output, as is the case for a smaller image. The final feature map's observed behavior also reflects the appropriate class's heatmap, with the location

of the object highlighted in the feature map. The feature map's output is a heatmap of the required object, which is useful information for segmentation use-case.

Since the feature map obtained at the output layer is down sampled as a consequence of the convolutions performed, it must be up-sampled using an interpolation technique. While bilinear up sampling is efficient, the paper recommends learning up sampling with deconvolution, which can also learn non-linear up sampling.

The network's encoder is responsible for down sampling, while the decoder is responsible for up sampling. This is a common trend in many architectures, with the encoder reducing the size and the decoder raising the sampling rate. In an ideal world, we would not use pooling to down sample and keep the sample size constant throughout, but this would result in a large number of parameters and would be computationally inefficient.

Despite the fact that the output results were satisfactory, the output observed was rough and not smooth. The explanation for this is that downsampling by 32 times using convolution layers causes information loss at the final feature layer. It is now extremely difficult for the network to perform 32x upsampling with this little data. FCN-32 is the name of this architecture.

The paper proposed two additional architectures to solve this problem: FCN-16 and FCN-8. The knowledge from the previous pooling layer is merged with the final feature map in FCN-16, and the network's job is now to learn 16x up sampling, which is better than FCN-32. FCN-8 attempts to boost it even further by adding data from a previous pooling layer.

2.4 Mask R-CNN

There are two stages to the faster R-CNN. A Region Proposal Network (RPN) is the first stage, which proposes candidate object bounding boxes. The second

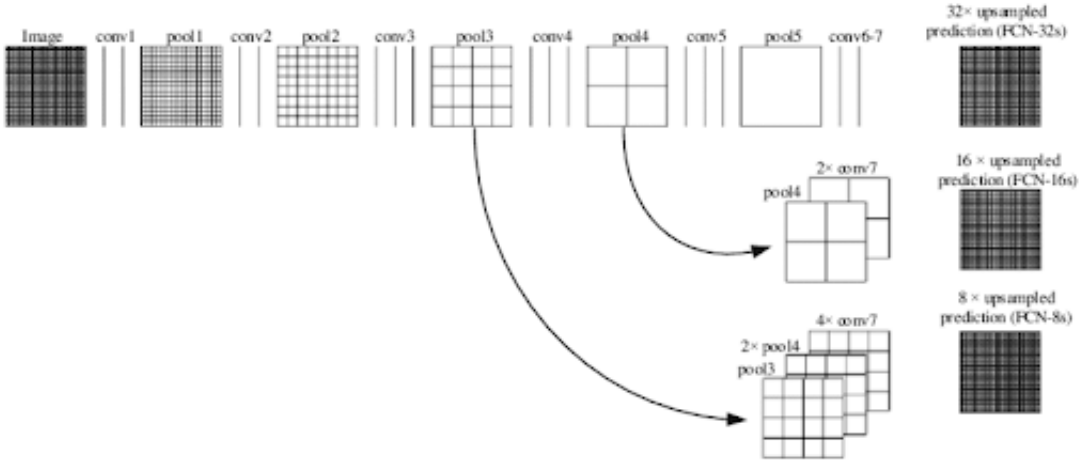


Figure 6: [5]fcn32, fcn16, fcn8 overall diagram.

stage, known as Fast R-CNN, extracts features from each candidate box using RoIPool and then performs classification and bounding-box regression. For faster inference, the features used by both stages can be shared.

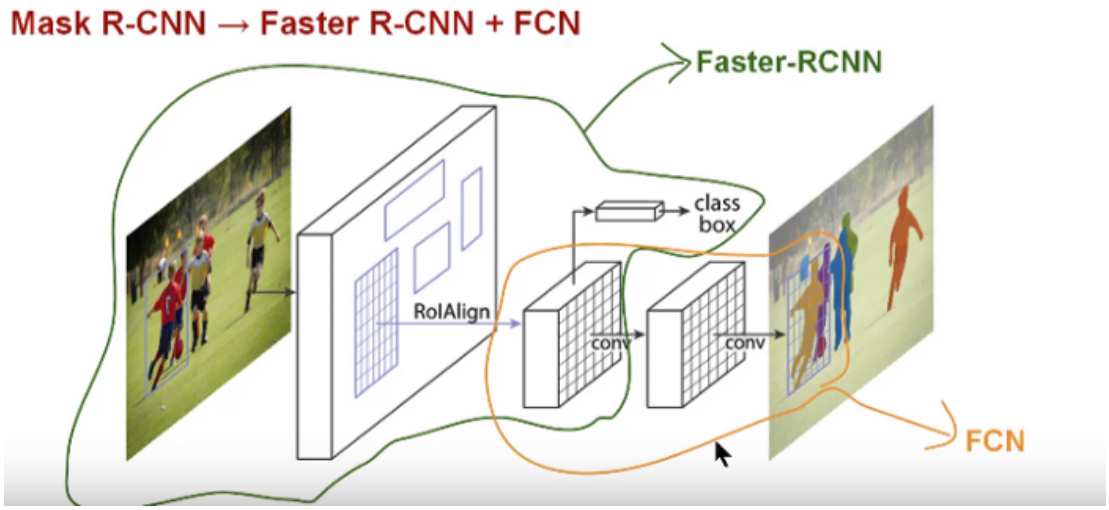


Figure 7: [6]The Mask R-CNN framework for instance segmentation.

The concept of Mask R-CNN is simple: Faster R-CNN outputs a class label and a bounding-box offset for each candidate object; we add a third branch that outputs the object mask, which is a binary mask that shows the pixels where the object is in the bounding box. The additional mask output, on the other hand, is distinct from the class and box outputs, necessitating the extraction of a much finer spatial

layout of an object. To achieve this, RCNN employs the Fully Convolutional Network (FCN).

Mask R-CNN, then, is a hybrid architecture that combines the two networks — Faster RCNN and FCN. The total loss in classification, generating the bounding box, and generating the mask is the loss function for the model. Mask RCNN has a set of enhancements over FCN that make it much more accurate[6].

2.5 U-Net for Semantic Segmentation

U-Net builds on top of the fully convolutional network. It was built for medical purposes to find tumours in lungs or the brain. It also consists of an encoder which down-samples the input image to a feature map and the decoder which up samples the feature map to input image size using learned deconvolution layers[22].

The main contribution of the U-Net architecture is the shortcut connections. In FCN, since we down-sample an image as part of the encoder we lost a lot of information which can't be easily recovered in the encoder part. FCN tries to address this by taking information from pooling layers before the final feature layer.

U-Net proposes a new approach to solve this information loss problem. It proposes to send information to every up sampling layer in decoder from the corresponding down sampling layer in the encoder as can be seen in the figure above thus capturing finer information whilst also keeping the computation low. Since the layers at the beginning of the encoder would have more information they would bolster the up sampling operation of decoder by providing fine details corresponding to the input images thus improving the results a lot.

2.6 Data Augmentation for Semantic Segmentation

Data augmentation is a technique for increasing the amount of training data available by changing or transforming existing data in a realistic way. Data augmentation is a technique for increasing the number of samples in a training dataset.

Overfitting can be minimized with data augmentation. The augmented images would introduce variance to the model, making it more robust. The following are some of the most widely used data augmentation strategies in semantic segmentation tasks:

- **Random Crop:** To retain the shapes of objects, Random Crop randomly selects a region and crops it out to generate a new data sample. The cropped region should have the same width/height ratio as the original image.
- **CenterCrop:** CenterCrop is used to crop the central part of the size H x W, from both image and the mask.
- **RandomRotate90:** Randomly rotate both the image and mask by 90 degrees. After RandomRotate90, GridDistortion may be used to transform both image and mask.
- **Horizontal Flip:** Horizontally flip both image and mask.
- **Vertical Flip:** Vertically flip the image and mask.
- **Cutout:** Cutout involves randomly masking out square regions of image during training.

2.7 Encoders

The encoder basically compresses the input and produces the code. Different kind of encoders used in different kind of deep learning tasks.

2.7.1 Residual Networks (ResNet)

ResNet is a Convolutional Neural Network (CNN) architecture composed of residual blocks (ResBlocks) with skip connections, which distinguishes ResNets from other CNNs[7].

ResNet won the ImageNet competition that year by a wide margin because it solved the vanishing gradient problem, while as more layers are introduced, training slows and accuracy stagnates or worsens. This is achieved by networks missing

links. Shorter connections between layers closer to the input and those closer to the output can make convolutional networks significantly deeper, more accurate, and more efficient to train.

The loss surface (the search space for the varying loss of the model's prediction) tends to be a collection of hills and valleys when visualized. The lowest point is also the lowest loss. Even if it is an exact part of a larger network, a smaller optimal network may be overlooked. This is attributable to the difficulty of navigating the loss surface. This means that adding a lot of deep layers to a model will make it worse at predicting.

Adding cross connections between layers of the network has proved to be a very successful solution, allowing large parts of the network to be skipped if necessary. It is also easier to train the model with optimal weights in order to reduce the loss.

Figure 8: [7]Bottom 34 Layer CNN, top 34 Layer ResNet CNN.

Each ResBlock has two connections from its input, one of which goes through a series of convolutions, batch normalization, and linear functions, while the other bypasses those steps. An identity, cross, or skip connection is what these are called. Both connections' tensor outputs are added together. A ResNet can be used for the encoder/down sampling section of the U-Net (the left half of the U).

Pretrained Encoder: If a pretrained model is used to train an image generation/prediction model, it significantly reduces training time. The model now has a basic understanding of the types of features that must be detected and enhanced. It's popular to use a model and weights that have been pre-trained on ImageNet.

2.7.2 MobileNet Network

MobileNet is an efficient network architecture and a set of two hyper-parameters in order to build very small, low latency models that can be easily matched to the design requirements for mobile and embedded vision applications[23].

MobileNet is built on the core layers which are depthwise separable filters. We then describe the MobileNet network structure and conclude with descriptions of the two model shrinking hyperparameters width multiplier and resolution multiplier.

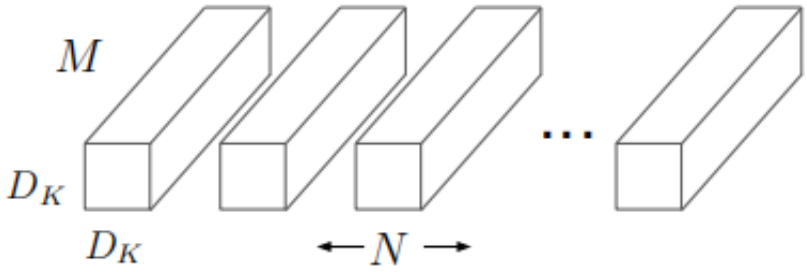


Figure 9: Standard Convolution Filters.

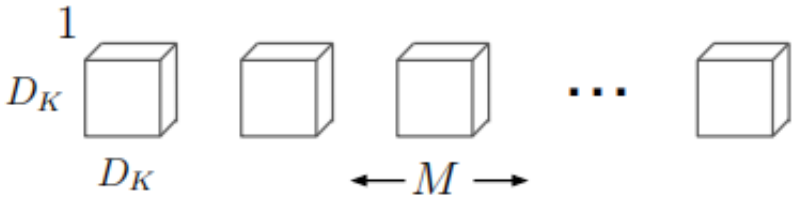


Figure 10: Depthwise Convolutional Filters.

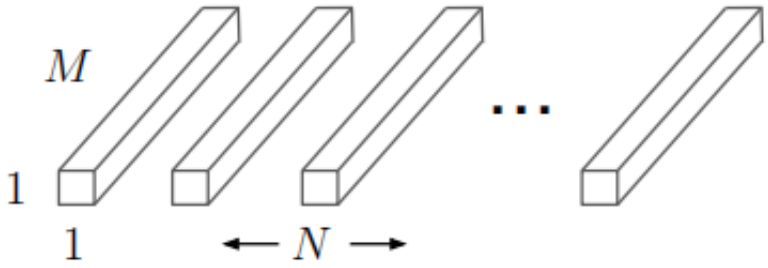


Figure 11: 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution.

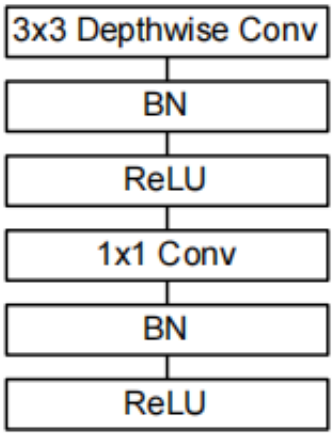


Figure 12: Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batchnorm and ReLU.

The MobileNet model is based on depth wise separable convolutions which is a form of factorized convolutions which factorize a standard convolution into a depthwise convolution and a 1×1 convolution called a pointwise convolution. For MobileNets the depthwise convolution applies a single filter to each input channel. The pointwise convolution then applies a 1×1 convolution to combine the outputs with the depthwise convolution. A standard convolution both fifilters and combines inputs into a new set of outputs in one step. The depthwise separable convolution splits this into two layers, a separate layer for filtering and a separate layer for combining.

2.7.3 MobileNetV2

MobileNetV2, that improves the state-of-the-art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes[24]. Our main contribution is a novel layer module: the inverted residual with linear bottleneck. This module takes as an input a low-dimensional compressed representation which is first expanded to high dimension and filtered with a lightweight depthwise convolution. Features are subsequently projected back to a low-dimensional representation with a linear convolution.

The bottleneck blocks appear similar to residual blocks where each block contains

an input followed by several bottlenecks followed by expansion . However, inspired by the intuition that the bottlenecks actually contain all the necessary information, while an expansion layer acts merely as an implementation detail that accompanies a non-linear transformation of the tensor, we use shortcuts directly between the bottlenecks.

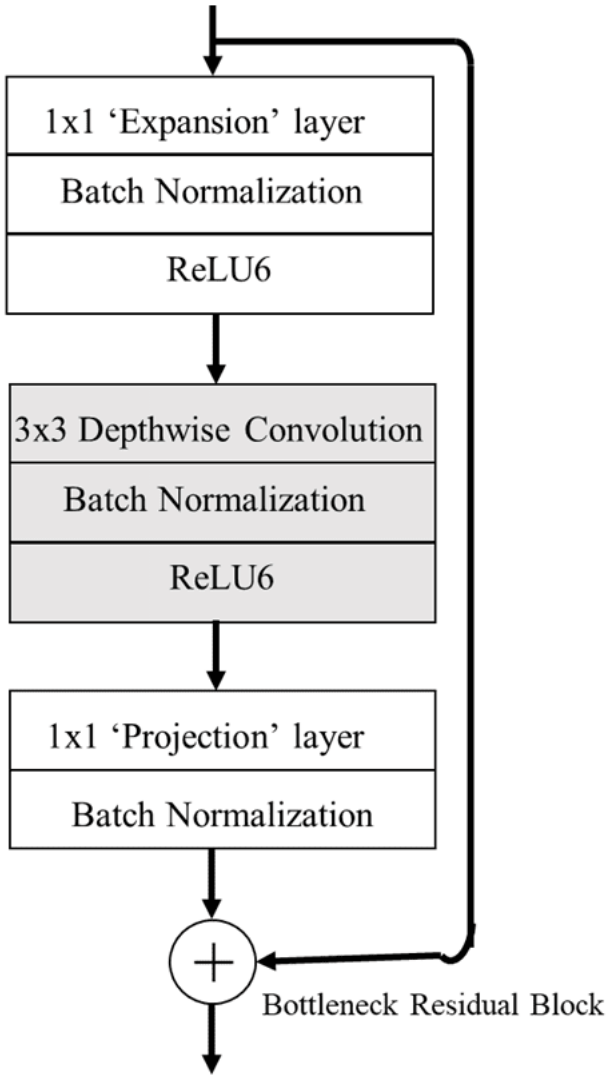


Figure 13: Basic building block of MobileNet V2.

3 Related Work

3.1 Non-Deep Learning Methods

3.1.1 Glomerular detection and segmentation using a Butterworth band-pass filter

Previously, a rapid, high throughput, scalable, and robust computational pipeline, capable of detecting and segmenting multiple glomeruli within the field-of-view was developed, using minimal computational complexity, by integrating the two different microscopic imaging modalities of immunofluorescence and histology[1].

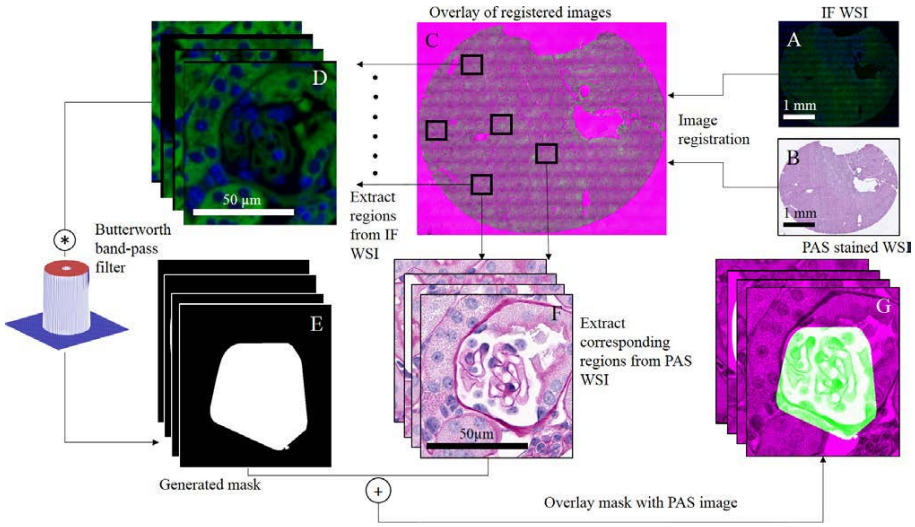


Figure 14: [1] Schematic diagram of the computational pipeline used to extract accurate glomerular boundaries.

In the above figure-

(A) Whole-slide image (WSI) of renal tissue section stained via immunofluorescence markers.

(B) WSI of the same renal section post-stained with Periodic acid-Schiff (PAS).

(C) Result of image registration by matching speeded up robust features.

(D) Extracted image patches containing glomeruli, from registered immunofluorescence WSI. The cell nuclei were stained with DAPI (blue). The green signal depicts tissue autofluorescence.

(E) Mask generated upon band-pass filtering of the image in Fig. D. (F) Image patches from PAS WSI corresponding to the ones shown in Fig. D.

(F) Overlay image of the masks shown in Fig. E and the PAS image patch shown in Fig. F.

This pipeline utilizes the robust yet simple Butterworth band-pass filter to exploit previously unexplored innate features of fluorescence photo physical properties of DAPI generated and tissue autofluorescence signals, thereby reducing the computational cost and complexity when compared to other techniques[25], while generating comparable performance.

Here, the standard Butterworth band-pass filter[26] with an order of n = 1 was used. The transfer functions of the low pass and high pass filter used to design the Butterworth band-pass filter are:

$$H_{LP}(u,v) = \frac{1}{1 + \frac{D(u,v)}{D_L^{2n}}}$$

$$H_{HP}(u,v) = 1 - \frac{1}{1 + \frac{D(u,v)}{D_H^{2n}}}$$

$$H_{BP}(u,v) = H_{LP}(u,v) * H_{HP}(u,v)$$

where, DL and DH indicate the upper and lower cut off frequencies and D(u, v) indicates the distance of each pixel from the origin.

3.1.2 Automatic glomerular identification using image analysis and machine learning

Previously, glomerular identification required expert pathologists as identification of glomeruli in pathology samples is difficult for both computers and untrained individuals.. Earlier methods of scoring histological kidney samples (glomeruli)

did not allow for collection of quantitative data in a high-throughput and consistent manner. Manual analysis and identification can be time consuming and often inefficient. Therefore, usage of machine learning approaches was initiated to develop high-throughput methods which will automatically identify and collect quantitative data from glomeruli with minimal human interaction between steps and provide quantifiable data independent of user bias. Previous works which didn't use machine learning methods were limited in recognizing glomeruli with varying characteristics (size, disease state, race).

Here, the goal was to achieve an automated reliable method of glomerular tuft classification, with an expandable workflow for phenotype quantification within glomeruli. Ilastic, a software which uses various machine learning techniques for image segmentation and ImageJ for scientific image analysis and enhancement were used[27].

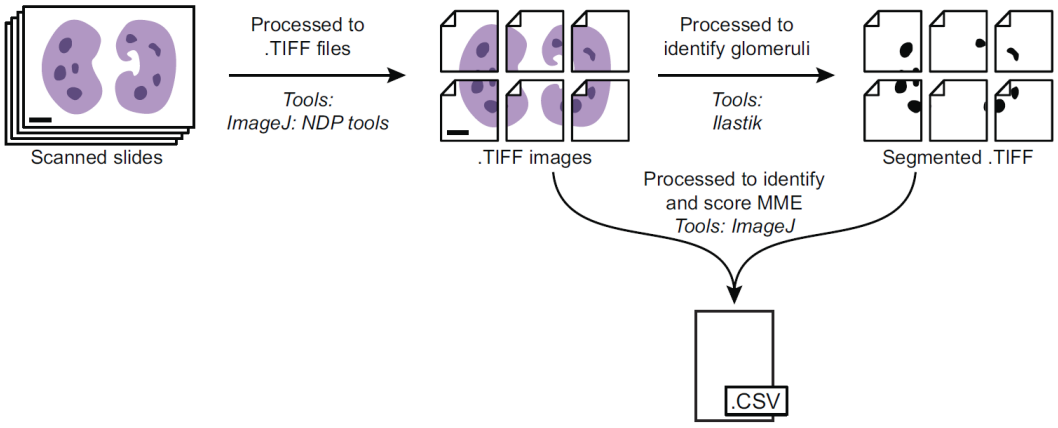


Figure 15: [8] Overview of the workflow and tools used throughout the acquisition, segmentation, and quantification of images.

3.2 Deep Learning Methods

3.2.1 Segmentation of Glomeruli Within Trichrome Images Using Deep Learning

Shruti Kannan et al. used a deep learning approach to correctly detect glomeruli and used a segmentation pipeline to segment the glomeruli.

In one of the previous implementations[28], there were three classes of data:

1. No glomerulus.
2. Normal or partially sclerosed (NPS) glomerulus.
3. Globally sclerosed (GS) glomerulus.

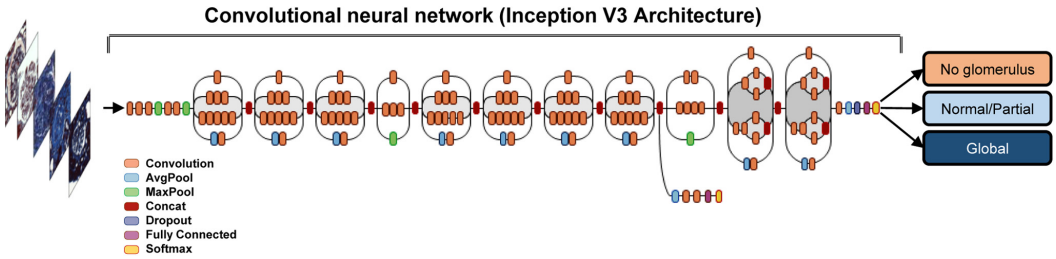


Figure 16: Schematic of the deep neural network. The classification technique is based on leveraging a pretrained convolutional neural network, which was fine-tuned on this dataset

At first Google's Inception v3 architecture was used in the images to correctly detect classes of images within given data. Then it was split into a 7:3 train test ratio. Some data augmentation techniques were used like random whitening, cropping which helped accurately identifying glomeruli within the border of the images. These images were put into a segmentation pipeline. A heatmap was generated which found how confident the model was in terms of detecting the presence of a GS glomerulus in that area, Then it was binarized and a distance transform was calculated. Finally the watershed segmentation provided with a segmentation of glomeruli within given data.

3.2.2 U-Net: Convolutional Networks for Biomedical Image Segmentation

This paper[22] illustrates an elegant architecture using the fully convolutional network[5]. This architecture was modified such that it works with very few training images and yields more precise segmentations. The main idea in[5] is

to supplement a usual contracting network by successive layers, where pooling operators are replaced by upsampling operators. As a result, these layers increase the resolution of the output. In order to localize, high resolution features from the contracting path are combined with the upsampled output. A successive convolution layer can then learn to assemble a more precise output based on this information

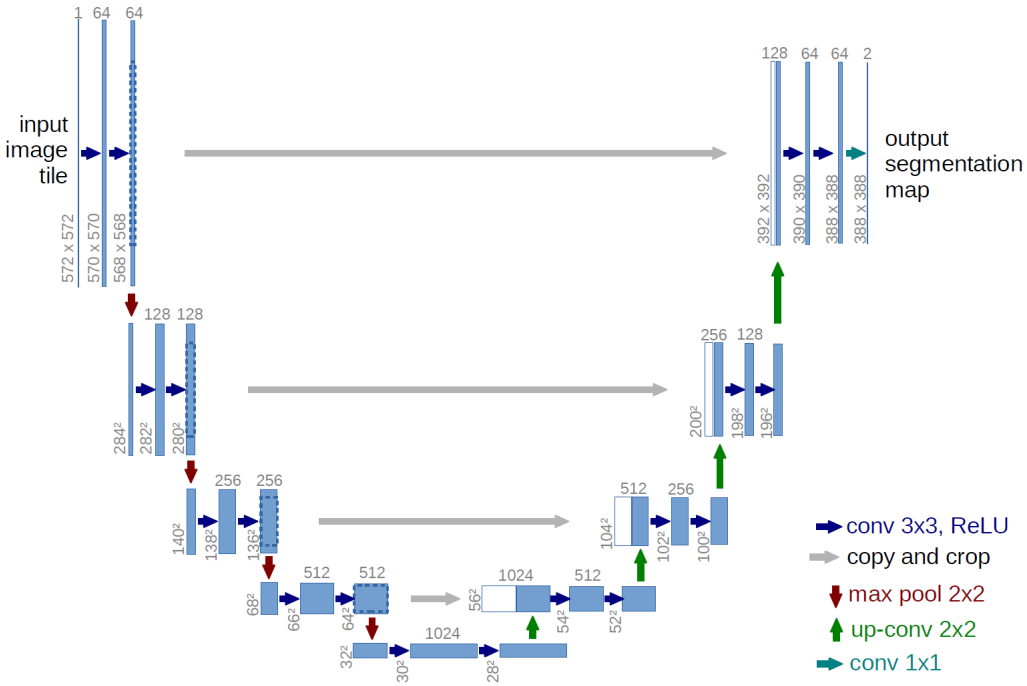


Figure 17: U-net architecture (example for 32x32 pixels in the lowest resolution).

It consists of a contracting path (left side) and an expansive path (right side). The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3x3 convolutions (unpadded convolutions), each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step the number of feature channels was doubled. Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed

by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution is used to map each 64- component feature vector to the desired number of classes. In total the network has 23 convolutional layers.

3.2.3 Double U-Net

Figure 18 shows an overview of the proposed architecture. As seen from the figure, DoubleU-Net starts with a VGG-19 as encoder sub-network, which is followed by decoder subnetwork. What distinguishes DoubleU-Net from U-Net in the first network (NETWORK 1) is the use of VGG-19 marked in yellow, ASPP marked in blue, and decoder block marked in light green. The squeeze-and-excite block[29] is used in the encoder of NETWORK 1 and decoder blocks of NETWORK 1 and NETWORK 2. An element-wise multiplication is performed between the output of NETWORK 1 with the input of the same network. The difference between DoubleU-Net and U-Net in the second network (NETWORK 2) is only the use of ASPP and squeeze-and-excite block. All other components remain the same.

In the NETWORK 1, the input image is fed to the modified U-Net, which generates a predicted mask (Output1). We then multiply the input image and the produced mask (Output1), which acts as an input for the second modified U-Net that produces another mask (Output2). Finally, we concatenate both the masks (Output1 and Output2) to see the qualitative difference between the intermediate mask (Output1) and final predicted mask (Output2).

We assume that the produced output feature map from NETWORK 1 can still be improved by fetching the input image and its corresponding mask again, and concatenating with Output2 will produce a better segmentation mask than the previous one. This is the main motivation behind using two U-Net architectures in the proposed architecture. The squeeze-and-excite block in the proposed networks reduces the redundant information and passes the most relevant information. ASPP has been a popular choice for modern segmentation architecture because it helps to extract high-resolution feature maps that lead to superior performance[30].

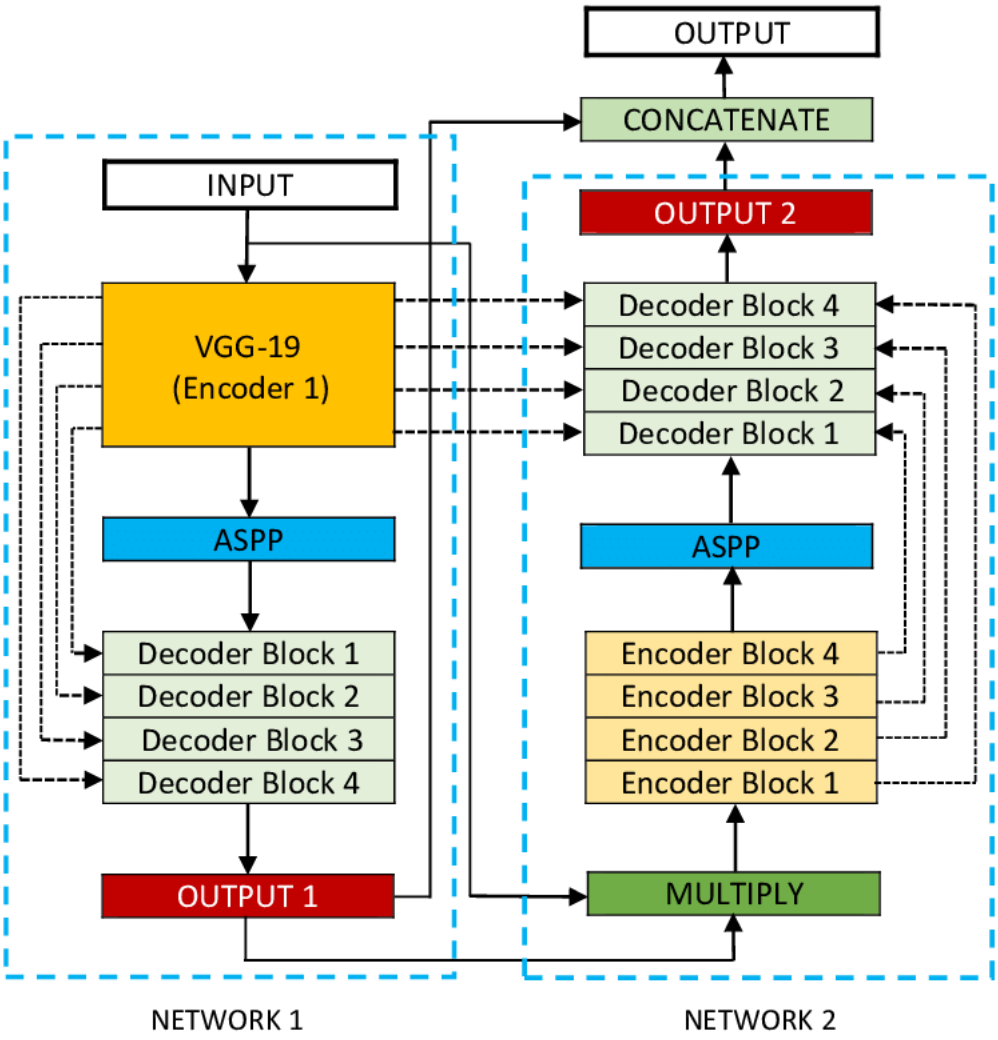


Figure 18: The DoubleU-net architecture

Encoder Explanation The first encoder in DoubleU-Net (encoder1) uses pre-trained VGG-19, whereas the second encoder (encoder2), is built from scratch. Each encoder tries to encode the information contained in the input image. Each encoder block in the encoder2 performs two 3×3 convolution operations, each followed by a batch normalization. The batch normalization reduces the internal co-variant shift and also regularizes the model. A Rectified Linear Unit (ReLU) activation function is applied, which introduces non-linearity into the model. This is followed by a squeeze-and- excitation block, which enhances the quality of the feature maps. After that, max-pooling is performed with a 2×2 window and stride 2 to reduce the spatial dimension of the feature maps.

Decoder Explanation As shown in Figure 18, we use two decoders in the entire network, with small modifications on the decoder as compared with that of the original U-Net. Each block in the decoder performs a 2×2 bi-linear up-sampling on the input feature, which doubles the dimension of the input feature maps. Now, we concatenate the appropriate skip connections feature maps from the encoder to the output feature maps. In the first decoder, we only use skip connection from the first encoder, but in the second decoder, we use skip connection from both the encoders, which maintains the spatial resolution and enhances the quality of the output feature maps. After concatenation, we again perform two 3×3 convolution operations, each of which is followed by batch normalization and then by a ReLU activation function. After that, we use a squeeze and excitation block. At last, we apply a convolution layer with a sigmoid activation function, which is used to generate the mask for the corresponding modified U-Net.

3.2.4 Glomerular Microscopic Image Segmentation Based on Convolutional Neural Network

Xueweu Han et al. proposed The Improved Masked R-CNN algorithm.

In this paper, the deep learning method is applied to glomerular microscopic image segmentation to replace the traditional glomerular segmentation method. They constructed the glomerular microscopic image segmentation dataset, and for the first time, the Improved Mask R-CNN algorithm based on convolutional neural network[6] is applied to the glomerular segmentation of medical microscopic images. The Improved Mask R-CNN algorithm achieved a simple, flexible, and accurate glomerular microscopic medical image segmentation by changing the length of square anchor side in pixels and increasing the head mask branch deconvolution layers. For comparison, we apply the Improved Mask R-RNN algorithm and other existing classic methods on the dataset we built. The experimental results show that our improved algorithm achieves superior performance in the segmentation and detection of the glomerular microscopic medical image dataset.

The Improved Mask R-CNN algorithm consists of a two-stage procedure. In the

first stage, region proposal network (RPN) proposes multiple candidate object bounding boxes by sliding a 3x3 spatial window over convolutional feature maps. Since the glomerulus is small in the microscopic image, the number is one or more and the size is not uniform, for the glomerular medical image segmentation, their improved algorithm reduces the length of square anchor side in pixels to improve the accuracy of model training. In the second stage, the network head predicts softmax probability of the class, bounding-box regression, and outputs a binary mask for each region of interest (Rol). Medical segmentation datasets tend to have a small number of images and concentration of categories, thus they increased the number of deconvolution layers on the head mask branch to improve segmentation accuracy. The Improved Mask R-CNN network framework is presented in the following figure.

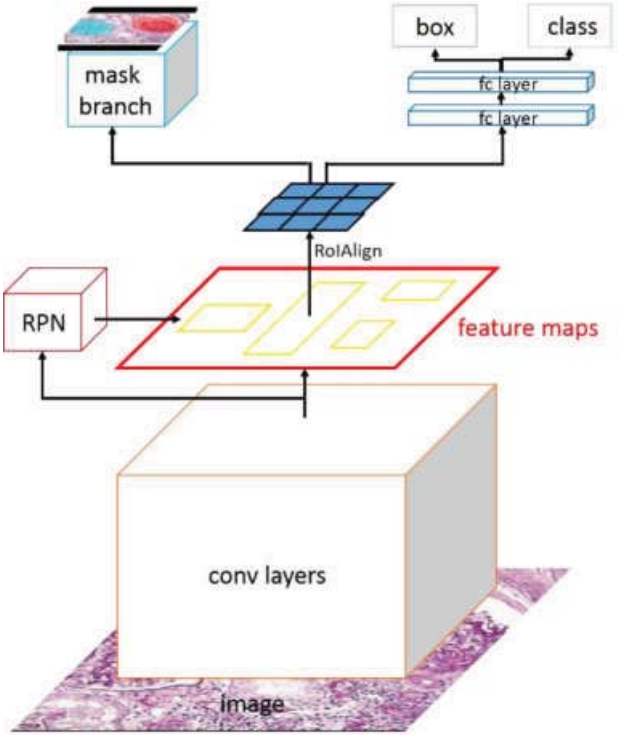


Figure 19: Improved Masked R-CNN algorithm network structure

4 Proposed Approach

4.1 Efficient LinkNet

In order to be useful in real-time applications, pixel-wise semantic segmentation for visual scene comprehension must be accurate as well as effective. Current algorithms, while accurate, do not concentrate on effectively using the parameters of neural networks. As a consequence, they are large in terms of parameters and operations, and therefore redundant.

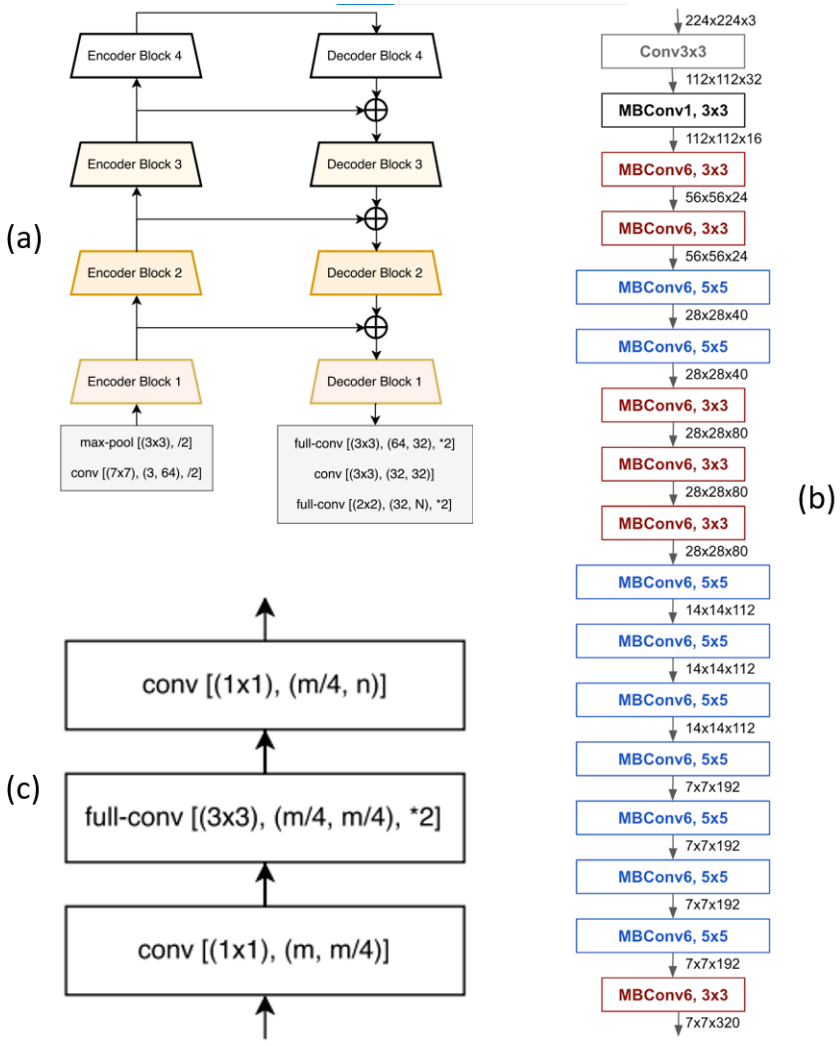


Figure 20: (a) LinkNet architecture; (b) Encoder block; (c) Decoder Block

Many current semantic segmentation techniques have an encoder-decoder pair at

the heart of their network architecture. The encoder encodes data into feature space, and the decoder encodes that data into spatial categorization in order to perform segmentation.

The pooling indices or complete convolution are used to recover spatial information lost in the encoder due to pooling or strided convolution. Bypassing spatial information and moving straight from the encoder to the corresponding decoder, accuracy is increased while processing time is reduced significantly. In this way, information that would have otherwise been lost at the encoder stage is saved, and no additional parameters or operations are wasted in relearning this lost information.

Semantic segmentation involves labeling each and every pixel of an image and therefore, retaining spatial information becomes utmost important. Despite the fact that semantic segmentation is targeted at applications that need real-time activity, most existing deep networks have an extremely long processing time.

On existing embedded hardware, the majority of these networks were unable to perform real-time segmentation. Apart from that, recurrent neural networks (RNNs) have recently been used to obtain contextual knowledge[31], but RNN use is computationally costly. There was also some work done in the field of developing efficient networks, with DCNN being designed for a faster forward processing time but a decline in prediction accuracy.

4.2 Network

The architecture of LinkNet[32] is presented in Fig. 20 (a). Here, conv means convolution and full-conv means full convolution[5]. Furthermore, /2 denotes downsampling by a factor of 2 which is achieved by performing strided convolution, and *2 means upsampling by a factor of 2. We use batch normalization between each convolutional layer and which is followed by ReLU non-linearity[33], [34]. Left half of the network shown in Fig. 20 (a) is the encoder while the one on the right is the decoder. The encoder starts with an initial block which performs convolution

on the input image with a kernel of size 7 × 7 and a stride of 2. This block also performs spatial max-pooling in an area of 3 × 3 with a stride of 2.

The later portion of encoder consists of encoder-block. Layers within these encoder-blocks are shown in detail in Fig. 20 (b). Similarly, layer details for decoder-blocks are provided in Fig. 20 (c).

For the first block of encoder and decoder is

$$m = n = 64$$

, while for rest of the blocks

$$m_{encoder} = n_{decoder} = 64 * 2^{i-1}$$

and

$$n_{encoder} = m_{decoder} = 64 * 2^{i}$$

.

Contemporary segmentation algorithms use networks such as VGG16 (138 million parameters), ResNet101 (45 million parameters) as their encoder which are huge in terms of parameters and GFLOPs. We used EfficientNet as an encoder which is fairly lighter and outperforms other encoders which we will discuss later.

LinkNet uses the technique of full-convolution in our decoder as proposed earlier by[5]. Every conv(k × k)(im, om) and full-conv(k × k)(im, om) operations have three parameters. Here, (k × k) represent (kernel size) and (im, om) represent (inputmap, outputmap) respectively.

Unlike other neural network architectures for segmentation, the network links each encoder to the decoder. Any spatial information is lost when the encoder conducts several downsampling operations. Using only the encoder's downsampled output, it's difficult to recover this missing information. Each encoder layer's input is also bypassed to the output of its corresponding decoder in this paper. Missing spatial information is expected to be recovered so that it can be used by the decoder and upsampling operations by doing so.

Furthermore, since the decoder shares the encoder's information at each layer, the decoder may use fewer parameters.

Model	ClassIoU	ClassiIoU
SegNet*	56.1	34.2
ENet*	58.3	34.4
Dilation10	68.7	-
Deep-Lab CRF (VGG16)	65.9	-
Deep-Lab CRF (ResNet101)	71.4	42.6
LinkNet Without bypass	72.6	51.4
LinkNet	76.4	58.6

Table 1: Cityscape[9] test set result

Despite the fact that the network's primary objective was to operate on handheld devices, we discovered that it is also very effective on high-end GPUs such as the NVIDIA Titan X.

This may be useful in data-center applications where large numbers of high-resolution images must be processed. Our network enables large-scale computations to be done much faster and more effectively, potentially saving a lot of money.

4.3 Encoder

ConvNets are commonly scaled up to increase accuracy. ResNet[7] can be scaled up from ResNet-18 to ResNet-200 by adding layers; GPipe[35] recently achieved 84.3 percent ImageNet top-1 accuracy by scaling up a baseline model four times larger. The method of scaling up ConvNets, on the other hand, has never been well known, and there are currently various approaches. The most popular approach is to increase the depth[7] or width[36] of ConvNets. Another less common, but growingly popular approach is to scale up models based on their image resolution[35].

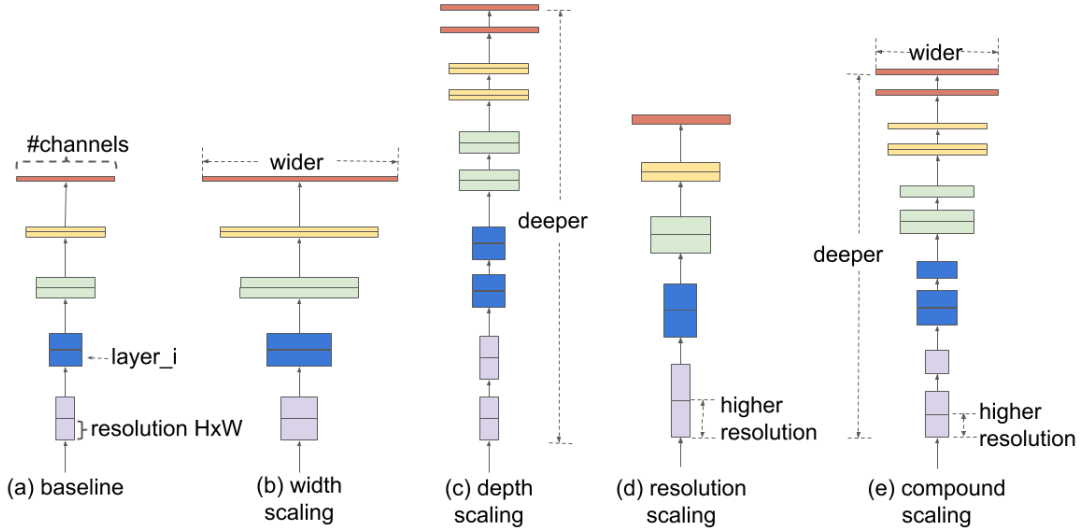


Figure 21: Compound Scaling

An empirical study shows that balancing all dimensions of network width, depth or resolution is important, and that this balance can be accomplished surprisingly easily by simply scaling each of them with a constant ratio. A simple but successful compound scaling method is used on this observation.

The compound scaling method makes sense because if the input image is bigger, then the network needs more layers to increase the receptive field and more channels to capture more fine-grained patterns on the bigger image.

The compound scaling method, which use a compound coefficient ϕ to uniformly scales network width, depth, and resolution in a principled way

$$depth : d = \alpha^{\phi}$$

$$width : w = \beta^{\phi}$$

$$resolution : r = \gamma^{\phi}$$

$$\text{s.t. } \alpha \times \beta^2 \times \gamma^2 \approx 2\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

where α, β, γ are constants that can be determined by a small grid search. Intuitively, ϕ is a user-specified coefficient that controls how many more resources are available for model scaling, while α, β, γ specify how to assign these extra resources to network width, depth, and resolution respectively

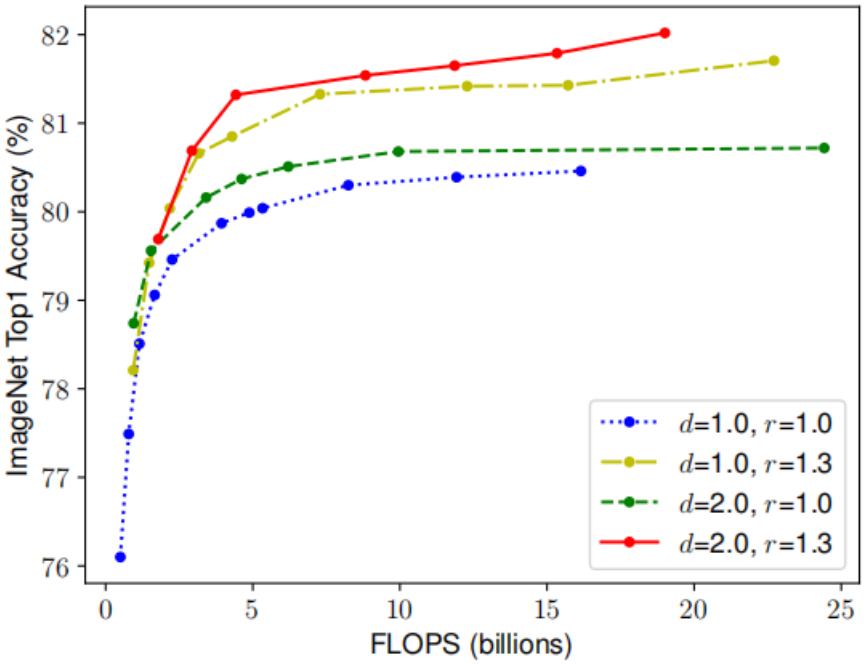


Figure 22: Scaling Network Width for Different Baseline Networks

Each dot in a line denotes a model with different width coefficient (w). The first baseline network (d=1.0, r=1.0) has 18 convolutional layers with resolution 224x224, while the last baseline (d=2.0, r=1.3) has 36 layers with resolution 299x299.

Since model scaling does not change layer operators in the baseline network, having a good baseline network is also critical. The scaling method is evaluated using existing ConvNets, but in order to better demonstrate the effectiveness of the scaling method, a new mobile-size baseline, called EfficientNet was also developed.

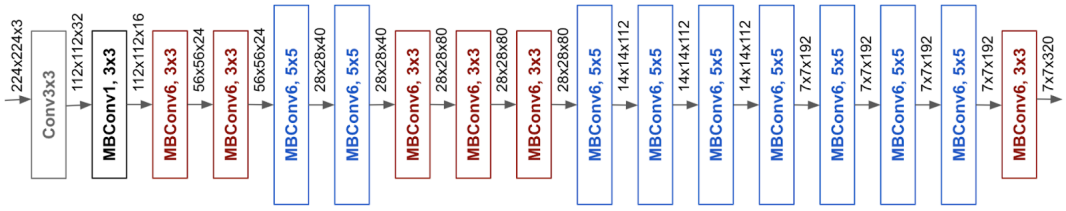


Figure 23: EfficientNet-B0

The baseline network is leveraged by a multi-objective neural architecture search that optimizes both accuracy and FLOPS. Specifically, The same search space

is used as[37], and use ACC(m)×[F LOP S(m)/T]w as the optimization goal, where ACC(m) and FLOPS(m) denote the accuracy and FLOPS of model m, T is the target FLOPS and w=-0.07 is a hyperparameter for controlling the trade-off between accuracy and FLOPS. Unlike[37], here optimization is done on FLOPS rather than latency since there are no target specific hardware devices. The search produces an efficient network, which is named EfficientNet-B0. It is slightly bigger due to the larger FLOPS target (our FLOPS target is 400M). Its main building block is mobile inverted bottleneck MBConv[24], to which squeeze-and-excitation optimization is added.

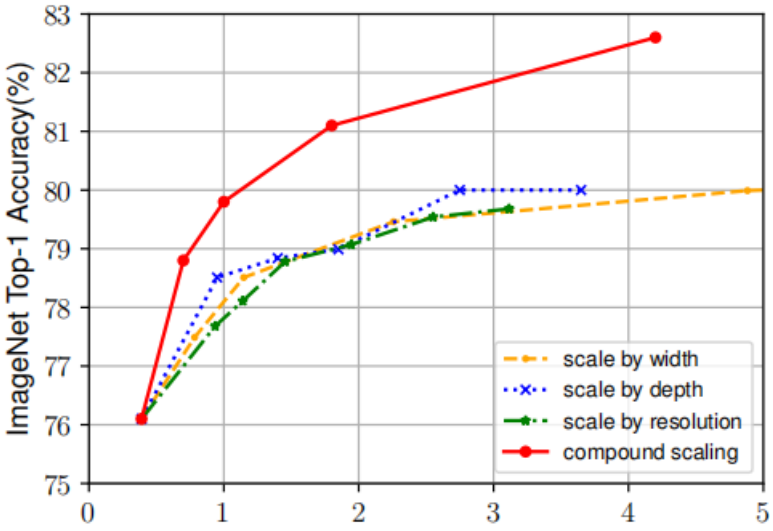


Figure 24: Scaling Up EfficientNet-B0 with Different Methods.

5 Experimental Analysis

5.1 Dataset

The Glomeruli FTU Segmentation Dataset provided by HuBMAP includes histological images of the kidney and annotation information representing the glomerular segmentation. The anatomical structure segmentation information and additional information (including anonymized patient data) about each image.

The HuBMAP data includes 11 fresh frozen and 9 Formalin Fixed Paraffin Em-

bedded (FFPE) PAS kidney images. Glomeruli FTU annotations exist for all 20 tissue samples.

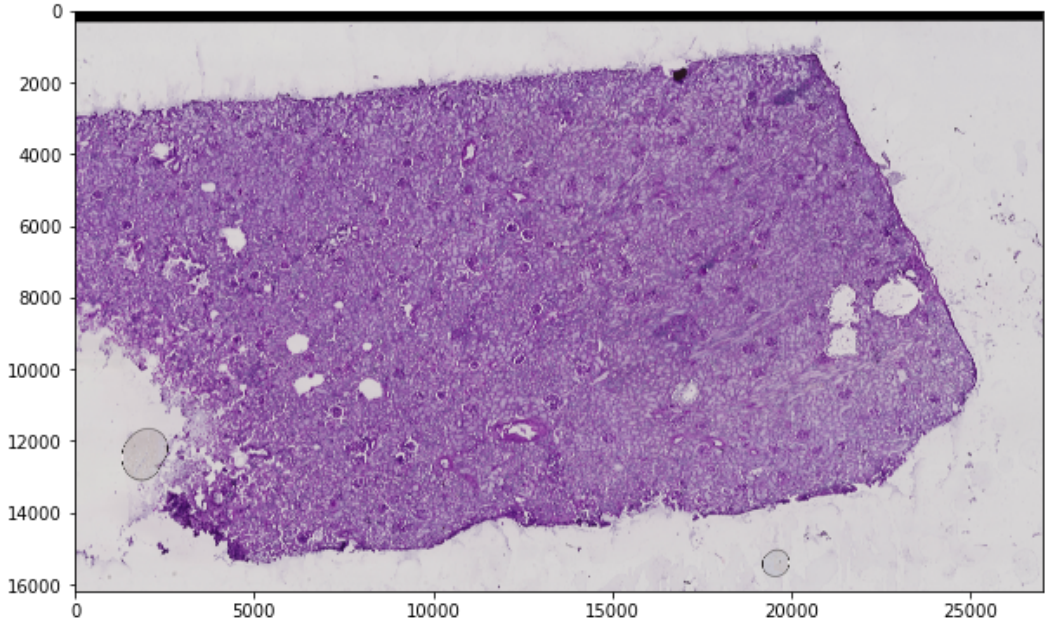


Figure 25: Sample image from the dataset.

The dataset consists of very large (>500MB - 5GB) TIFF files. There are 8 training set. This csv includes ids corresponding to data in the train directory. Also it has mask data in *encoding* column. This data is encoded with RLE encoding. The public test set has 5. All the histological images of the kidney are in tiff format.

We also have a dataset information file where the training images can be sectioned into sex, ethnicity, race, weight, bmi for further analysis.

We can decode masks from the encoding column of train.csv file.

The dataset also includes two kinds of annotation files. The annotations denote segmentations of glomeruli.Both the training and public test sets also include anatomical structure segmentations.

1. **Glomerulus segmentation file:** According to the description of the dataset, the same information as the rle-encoded mask is stored.

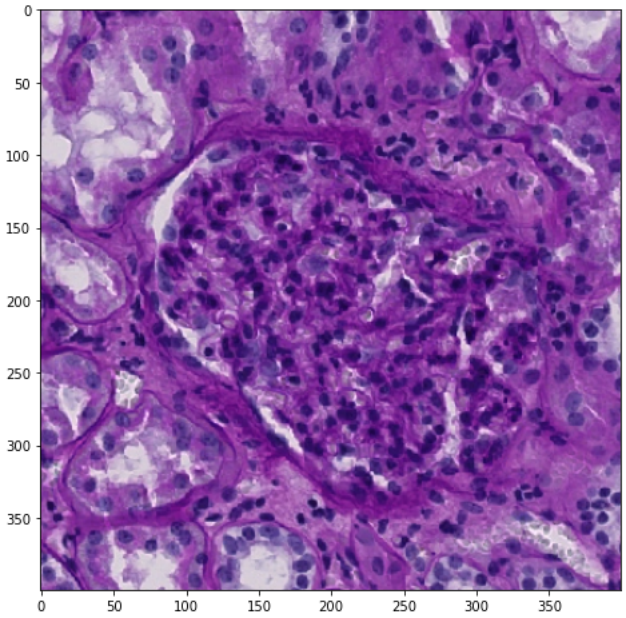


Figure 26: Enhanced version of the sample image containing glomeruli.

2. **Anatomical structure file:** In the same way with a glomerulus segmentation file, anatomical structure segmentations are shown. This file contains anatomical structure segmentations. It's used to identify the various parts of the tissue

It is also notable that there is a **private** hidden dataset similar to the public dataset but larger. It's used in testing in order to ensure the unbiased performance measure of the model.

5.2 Preprocessing

Each of the images in the dataset has 50k pixel size and is saved as a high-resolution tiff image. To make such large images to be suitable for training of a neural network, they must be cut into tiles. Based on the size of the detected features, the appropriate tile size for this data should be 1024x1024. But, it would be an overshoot for a starter code and the initial model development. Therefore, tiles of 4 times lower resolution are used - 256x256. After this we got 256x256 sized trained images cut into tiles and their corresponding masks.

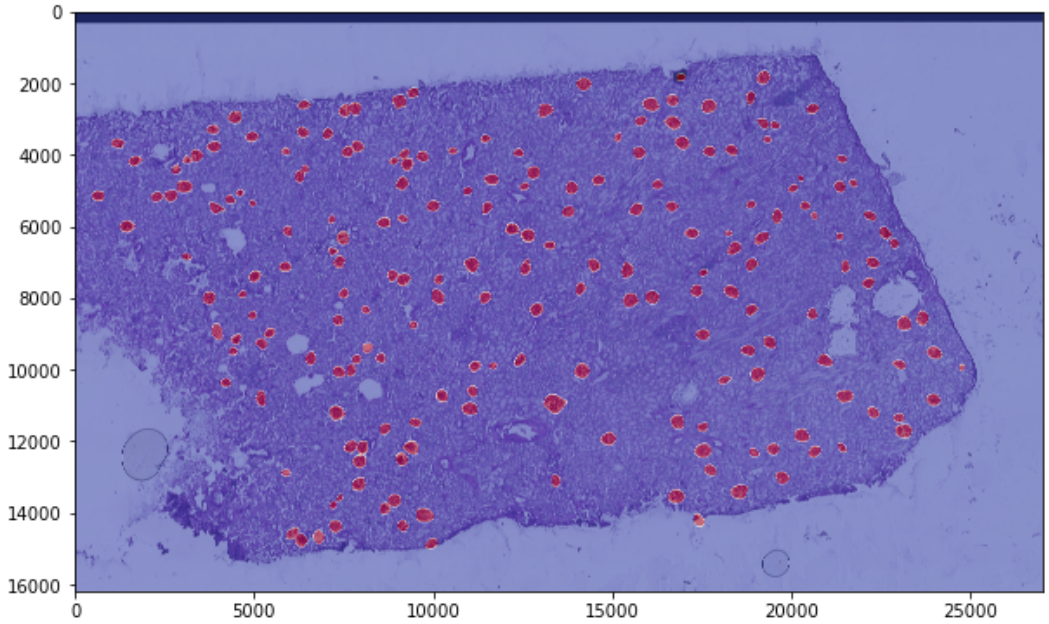


Figure 27: Sample image from training dataset with glomeruli mask applied.

5.3 Data Augmentation

We used many combinations of data augmentation techniques. We used - Compose which composes several transforms together. At First Horizontal Flip, Vertical Flip, RandomRotate90, ShiftScaleRotate is used in one Compose. Then one of OpticalDistortion, GridDistortion and PiecewiseAffine is used. Finally one of HueSaturationValue ,CLAHE ,Random Brightness Contrast is used.

5.4 Methodology

Among many variations of segmentation models, encoders, feature extractors, the model which gave the most promising result is: LinkNet with EfficientNet-B5 as modified encoder block, pretrained on ImageNet. Our models and overall procedure used PyTorch and required libraries. When using LinkNet with EfficientNet encoder, pre-trained EfficientNet worked as the backbone and feature extractor of our model. EfficientNet had pretrained encoder weights from ImageNet dataset.

The loss that works the best for semantic segmentation in most of the cases is symmetricLovasz-Softmax loss, a differentiable surrogate of IoU. However, ReLU

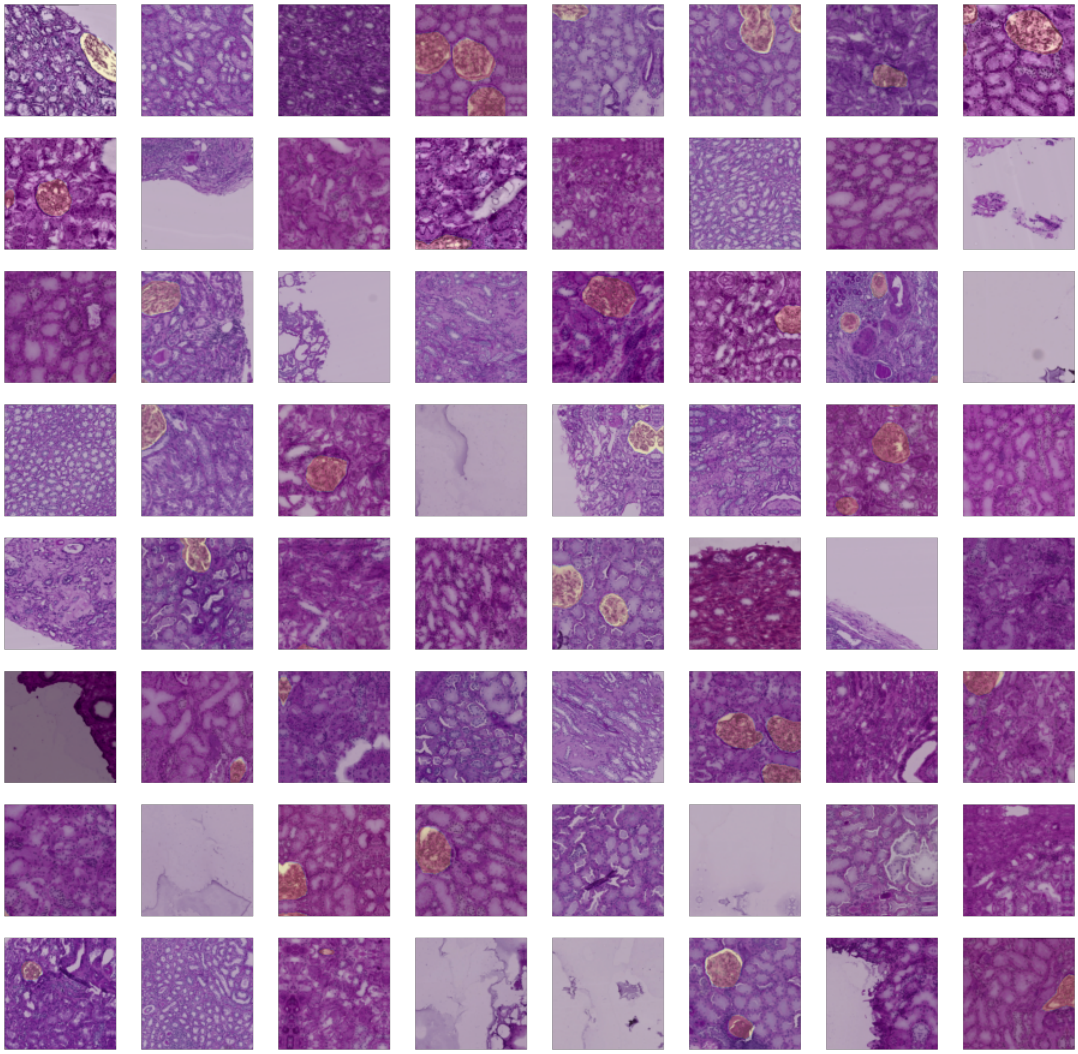


Figure 28: Sample training image cut into tiles as 256x256 images.

in it must be replaced by (ELU + 1), Because sometimes regular ReLU in lovasz predicts a lot of noise that must be eliminated by careful selection of the threshold. With ELU+1 usually the prediction is close to zero where it should be zero, but with ReLU we have quite many predictions with 0.1-0.3 probability even if they are quite easy to identify as negative examples.

$def symmetric\_lovasz(outputs, targets)$:

$return 0.5*(lovasz\_hinge(outputs, targets)+lovasz\_hinge(-outputs, 1.0-targets)$

We have tried with batch sizes of 64, 32 and 16. Number of workers is taken as 4. We've used k-fold cross validation with 4 folds. In each fold first only the head

was trained in 6 epochs, freezing other layers. Then after unfreezing the layers, the whole model was trained in 32 epochs.

5.5 Result Analysis

We used Dice Coefficient as our evaluation metric since it performs well in case of our semantic segmentation problem.

$$DiceCoefficient = \frac{2 \times AreaofOverlap}{TotalNumberofPixelsinbothImages}$$

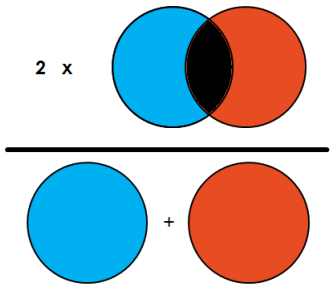


Figure 29: Dice Coefficient.

In our case we specifically used dice with automatic threshold selection. After numerous experimentation our pipeline finally produces a 0.827 score. The comparison table is given below :

Model	Dice
DoubleU-Net	0.736
U-Net + MobileNetV2	0.770
U-Net + MobileNetV2 (Pretrained on ImageNet)	0.801
LinkNet + MobileNetV2 (Pretrained on ImageNet)	0.806
LinkNet + EfficientNet-B5 (Pretrained on ImageNet)	0.827

Table 2: Result Comparison

From the above table we can see that our proposed method produces overall the best result. UNet which is the state of art model for medical image segmentation produces 0.801 which is 2.6% less than our final result. Even using pre-trained enocoders like mobilenetv2 we get less optimized results. Later we tried with more complex architecture like DoubleUnet but the performance of that network was very poor due to our tiles' low resolution images.

Model	Dice
LinkNet + EfficientNet-B2 (Pretrained on ImageNet)	0.820
LinkNet + EfficientNet-B3 (Pretrained on ImageNet)	0.825
LinkNet + EfficientNet-B5 (Pretrained on ImageNet)	0.827

Table 3: Result Comparison among variants of Efficient LinkNet

So we needed more lighter models like EfficinetNet but since our dataset is very small and comprises very high resolution images we needed to use a feature extractor which performs better than the network in compound scaling method but in the meantime can handle a small number of training data. Due to our hardware resource limitations we could implement upto EfficinetNet-B5 which finally gives us overall the best performance till now.

6 Conclusion

6.1 Summary

Segmentation of Glomeruli in Kidney Tissue Images may have a wide range of health impacts. Glomerular identification methods that are automated and accurate would ultimately increase the accuracy and pace of kidney study. Manually identifying functional tissue units is time and cost inefficient due to the dynamic nature of glomeruli and their intense variability in size and shape. While glomerular identification is a difficult task, deep learning semantic segmentation models have shown promising results in this area.

In our thesis work, we have tried using many variations of segmentation models, encoders, feature extractors and explored their potentials for semantic segmentation of glomeruli. In our proposed approach, we used the network architecture which gives the most promising result on the dataset, consisting of LinkNet with EfficientNet as modified encoder block, pretrained on ImageNet. This model achieved a DICE score of x% when tested on our dataset. Here the Compound Scaling provides better performance without compromising the efficiency. This pipeline outperformed other models that we have experimented with and allowed better performance than previous non deep learning based methodologies of glomerular identification.

6.2 Future Work

Our main problem was we were not able to train with large dataset due to our limited computational resources. As a result we did not get better results as expected in our test set. In future our main focus is to work with larger data. We also aim to work on developing a more complex network to handle the problem efficiently.

We plan to work in an online competition on Identifying glomeruli in human kidney tissue images and we hope that as time passes we will be able to grow our model in order to improve performance without sacrificing accuracy.

References

[1] Darshana Govind, Brandon Ginley, Brendon Lutnick, John E Tomaszewski, and Pinaki Sarder. Glomerular detection and segmentation from multimodal microscopy images using a butterworth band-pass filter. In *Medical Imaging 2018: Digital Pathology*, volume 10581, page 1058114. International Society for Optics and Photonics, 2018.

[2] Serena Yeung Fei-Fei Li, Justin Johnson. Detection and segmentation. *Fei-Fei Li, Justin Johnson, Serena Yeung*, 2017.

[3] https://www.mathworks.com/discovery/convolutional-neural-network matlab.html. Convolutional neural network-3 things you need to know. *https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html*, 2015.

[4] https://ujjwalkarn.me/2016/08/11/intuitive-explanation convnets/. An intuitive explanation of convolutional neural networks. *https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/*, 2016.

[5] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Susan M Sheehan and Ron Korstanje. Automatic glomerular identification and quantification of histological phenotypes using image analysis and machine learning. *American Journal of Physiology-Renal Physiology*, 315(6):F1644–F1651, 2018.

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[10] Homer William Smith. *The kidney: structure and function in health and disease*. Oxford University Press, USA, 1951.

[11] Pinaki Sarder, Brandon Ginley, and John E Tomaszewski. Automated renal histopathology: Digital extraction and quantification of renal pathology. In *Medical Imaging 2016: Digital Pathology*, volume 9791, page 97910F. International Society for Optics and Photonics, 2016.

[12] Agnes B Fogo. Mesangial matrix modulation and glomerulosclerosis. *Nephron Experimental Nephrology*, 7(2):147–159, 1999.

[13] Gunter Wolf, Sheldon Chen, and Fuad N Ziyadeh. From the periphery of the glomerular capillary wall toward the center of disease: podocyte injury comes of age in diabetic nephropathy. *Diabetes*, 54(6):1626–1634, 2005.

[14] Brandon Ginley, John E Tomaszewski, Rabi Yacoub, Feng Chen, and Pinaki Sarder. Unsupervised labeling of glomerular boundaries using gabor filters and statistical testing in renal histology. *Journal of Medical Imaging*, 4(2):021102, 2017.

[15] Wilhelm Kriz, Norbert Gretz, and Kevin V Lemley. Progression of glomerular diseases: is the podocyte the culprit? *Kidney international*, 54(3):687–697, 1998.

[16] JR Nyengaard and TF Bendtsen. Glomerular number and size in relation to age, kidney weight, and body surface in normal man. *The Anatomical Record*, 232(2):194–201, 1992.

[17] Michael D Hughson, Victor G Puelles, Wendy E Hoy, Rebecca N Douglas-Denton, Susan A Mott, and John F Bertram. Hypertension, glomerular hypertrophy and nephrosclerosis: the effect of race. *Nephrology Dialysis Transplantation*, 29(7):1399–1409, 2014.

[18] Otto Saphir. The state of the glomerulus in experimental hypertrophy of the kidneys of rabbits. *The American journal of pathology*, 3(4):329, 1927.

[19] Ruth Rasch, Finn Lauszus, Jesper Skovhus Thomsen, and Allan Flyvbjerg. Glomerular structural changes in pregnant, diabetic, and pregnant-diabetic rats. *Apmis*, 113(7-8):465–472, 2005.

[20] SK Agarwal, S Sethi, and AK Dinda. Basics of kidney biopsy: A nephrologist's perspective. *Indian journal of nephrology*, 23(4):243, 2013.

[21] Tom Sercu and Vaibhava Goel. Dense prediction on sequences with time-dilated convolutions for speech recognition. *arXiv preprint arXiv:1611.09288*, 2016.

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[23] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[24] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[25] Anibal Pedraza, Jaime Gallego, Samuel Lopez, Lucia Gonzalez, Arvydas Laurinavicius, and Gloria Bueno. Glomerulus classification with convolutional neural networks. In *Annual conference on medical image understanding and analysis*, pages 839–849. Springer, 2017.

[26] Rafael C Gonzalez, Richard E Woods, et al. Digital image processing, 2002.

[27] Christoph Sommer, Christoph Straehle, Ullrich Koethe, and Fred A Hamprecht. Ilastik: Interactive learning and segmentation toolkit. In *2011 IEEE international symposium on biomedical imaging: From nano to macro*, pages 230–233. IEEE, 2011.

[28] Shruti Kannan, Laura A Morgan, Benjamin Liang, McKenzie G Cheung, Christopher Q Lin, Dan Mun, Ralph G Nader, Mostafa E Belghasem, Joel M

Henderson, Jean M Francis, et al. Segmentation of glomeruli within trichrome images using deep learning. *Kidney international reports*, 4(7):955–962, 2019.

[29] Ke Zhang, Yurong Guo, Xinsheng Wang, Jinsha Yuan, Zhanyu Ma, and Zhenbing Zhao. Channel-wise and feature-points reweights densenet for image classification. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 410–414. IEEE, 2019.

[30] Debesh Jha, Pia H Smedsrud, Michael A Riegler, Dag Johansen, Thomas De Lange, Pål Halvorsen, and Håvard D Johansen. Resunet++: An advanced architecture for medical image segmentation. In *2019 IEEE International Symposium on Multimedia (ISM)*, pages 225–2255. IEEE, 2019.

[31] Francesco Visin, Marco Ciccone, Adriana Romero, Kyle Kastner, Kyunghyun Cho, Yoshua Bengio, Matteo Matteucci, and Aaron Courville. Reseg: A recurrent neural network-based model for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 41–48, 2016.

[32] Abhishek Chaurasia and Eugenio Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2017.

[33] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[34] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[35] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Mia Xu Chen, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *arXiv preprint arXiv:1811.06965*, 2018.

[36] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[37] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.