Eye Vessel Prediction

Pixel-wise Segmentation with U-Net Architectures

Agenda

- Introduction
- The DRIVE Dataset
- The U-Net
- The Fully Convolutional Transformer
- Training & Results
- Summary
- Appendix

Introduction

A Brief History of Deep Learning

- The perceptron was first invented in 1943 and backpropagation in 1986.
- Deep learning gained massive popularity in the 2010s with the advent of big data, big compute, and the success of large computer vision models and large language models.
- There are many different sub-families of neural network architectures, each specializing in a specific task or modality (i.e. image data, sequence/text data, or data generation).
- Convolutional neural nets are still the gold standard in computer vision tasks. However, many recent applications of transformers to these tasks have proven successful.

Deep Learning in Healthcare

- With the recent success of neural networks, many researches have attempted to apply them to various areas of healthcare, with varying levels of success.
- There are many challenges in the healthcare domain which are problematic for deep learning
  - Datasets are usually very small
  - The costs of wrong predictions are very high
  - Model interpretability is highly desired
- Image segmentation is a common task of deep learning in the healthcare domain

*COVID-NET detecting infected areas within a chest radiography image

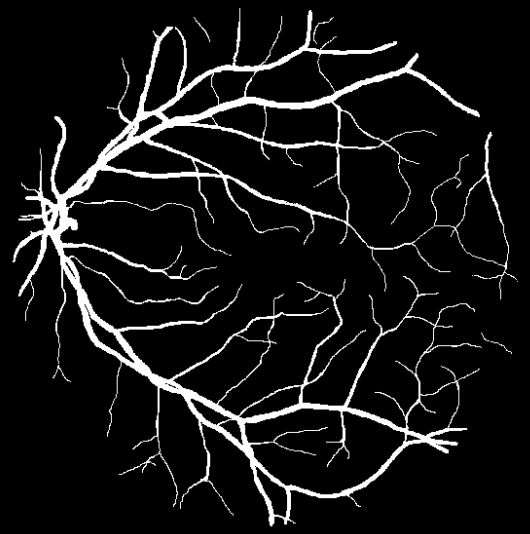
The Task

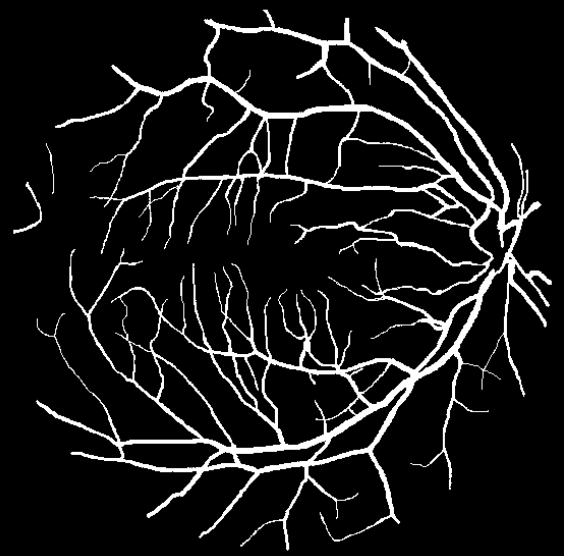
- Identify the blood vessels in the images of retina
- Train a model that takes the RBG retina image as input and produces a binary blood vessel map as output
- In machine learning, this is known as image segmentation
- Measure model performance on a set of held-out images that were not used in training

The DRIVE Dataset

Digital Retina Images for Vessel Extraction (DRIVE)

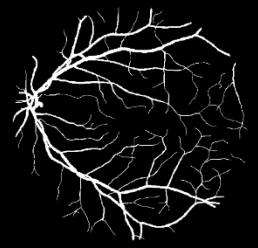
- 40 RGB images of the retina obtained from a diabetic retinopathy screening
- Training and testing sets have been divided into 20 images each
- Each image is paired with a manual segmentation map of the vasculature serving as the ground truth label
  - These human annotations were overseen by medical professionals



Data Augmentation

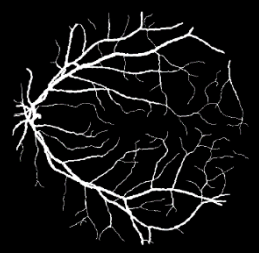
Original Image:



Horizontal Flip:

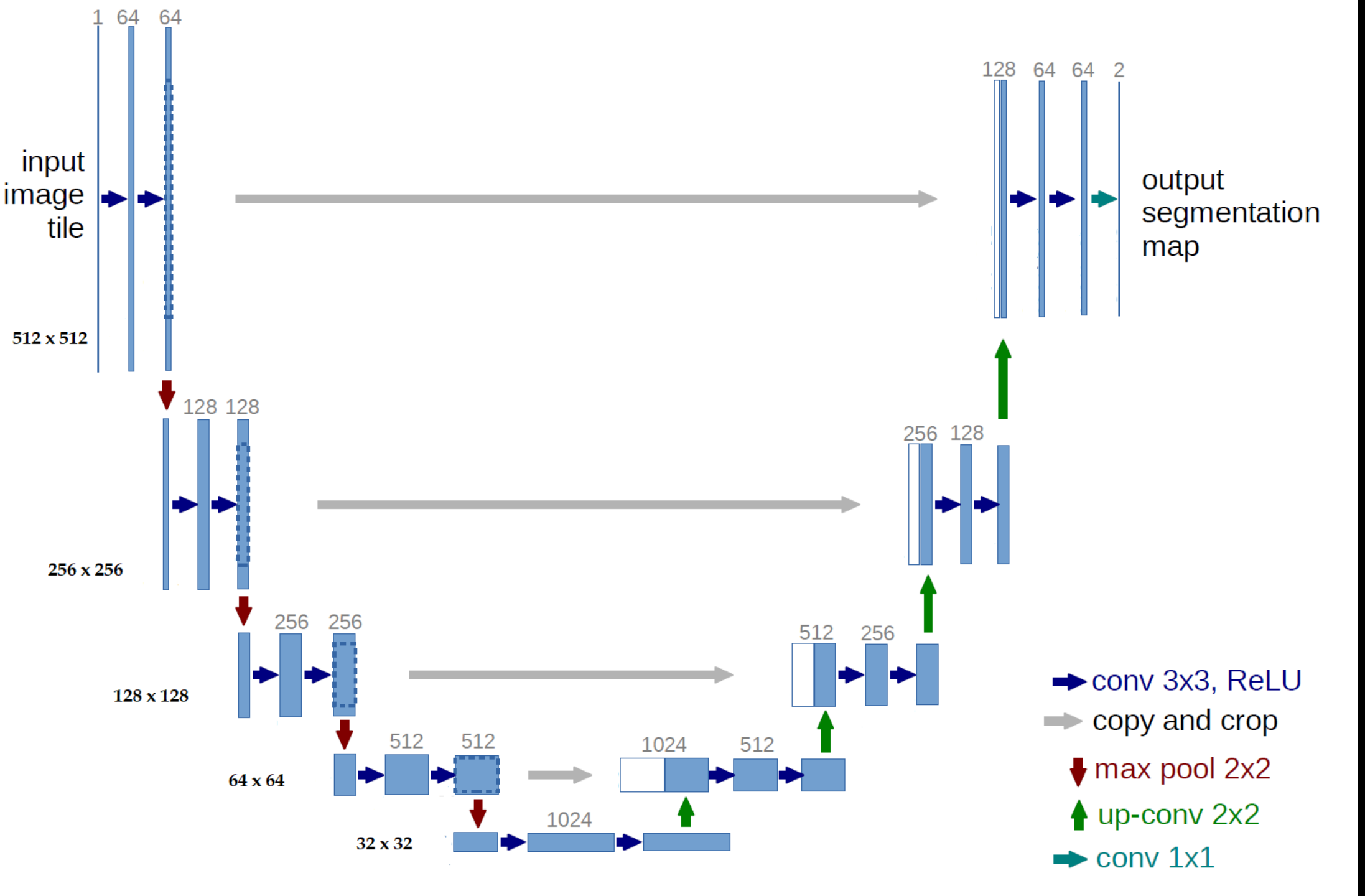
Vertical Flip:

45 deg Rotation:



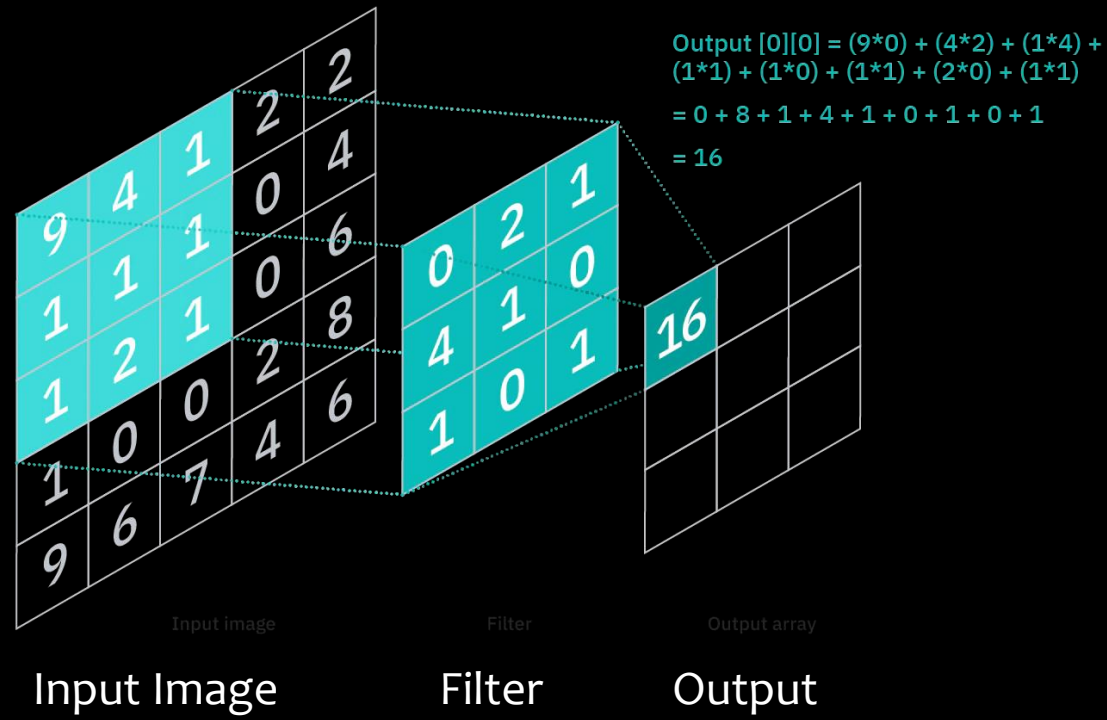
- Increases training set size by applying neutral transformations to images
- Applied a horizontal flip, vertical flip, and a 45-degree rotation.
- Training set size boosted from 20 images to 80 images

The U-Net


1
64
64
input image tile
512 x 512
128
64
64
2
output segmentation map
128
128
256 x 256
256
128
256
256
128 x 128
512
256
512
512
64 x 64
1024
512
1024
32 x 32
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

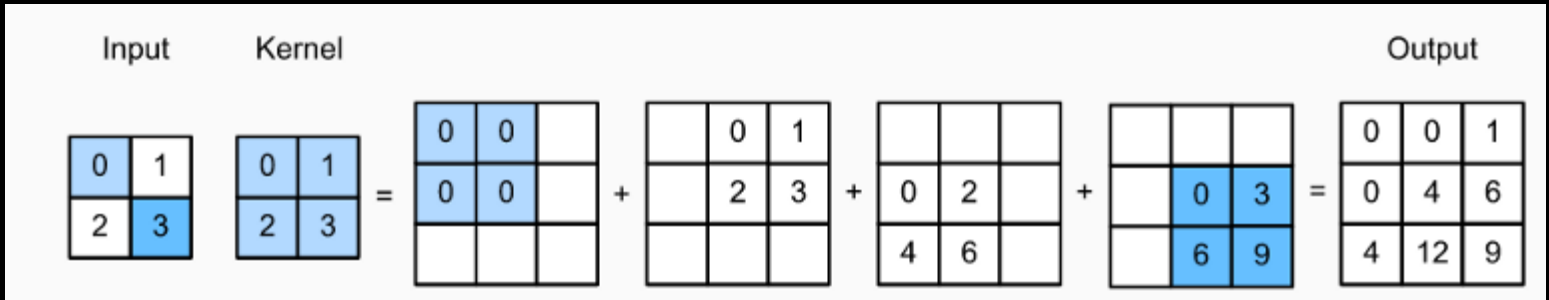
The Convolutional Layer

- An image is effectively a matrix of integers between 0-255.
- A ‘kernel’ or ‘filter’ is a smaller matrix that slides over the input image (i.e. convolves), and the dot product of is taken at each step, which forms a smaller output matrix.
- Many such filters are often applied at each layer.
- The weights of the kernel matrix are learned during training to extract salient features from the images.

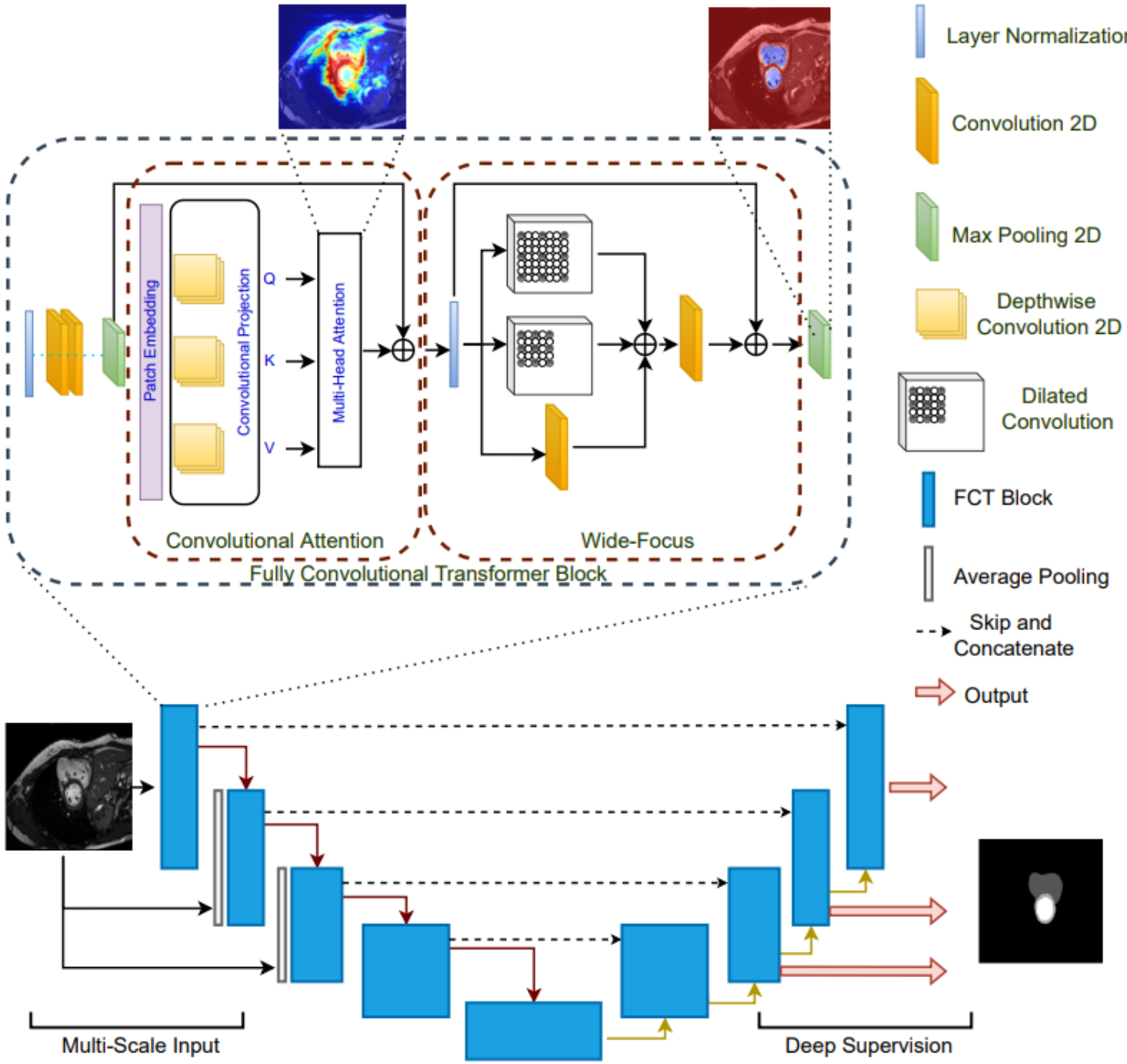


The Transposed Convolution

- How does the decoder up-sample the convoluted outputs?
- A kernel is learned such that each entry in the input matrix is multiplied by the entire kernel.
- The resulting matrices are then summed element-wise to return an output matrix with a larger size.



The Fully Convolutional Transformer



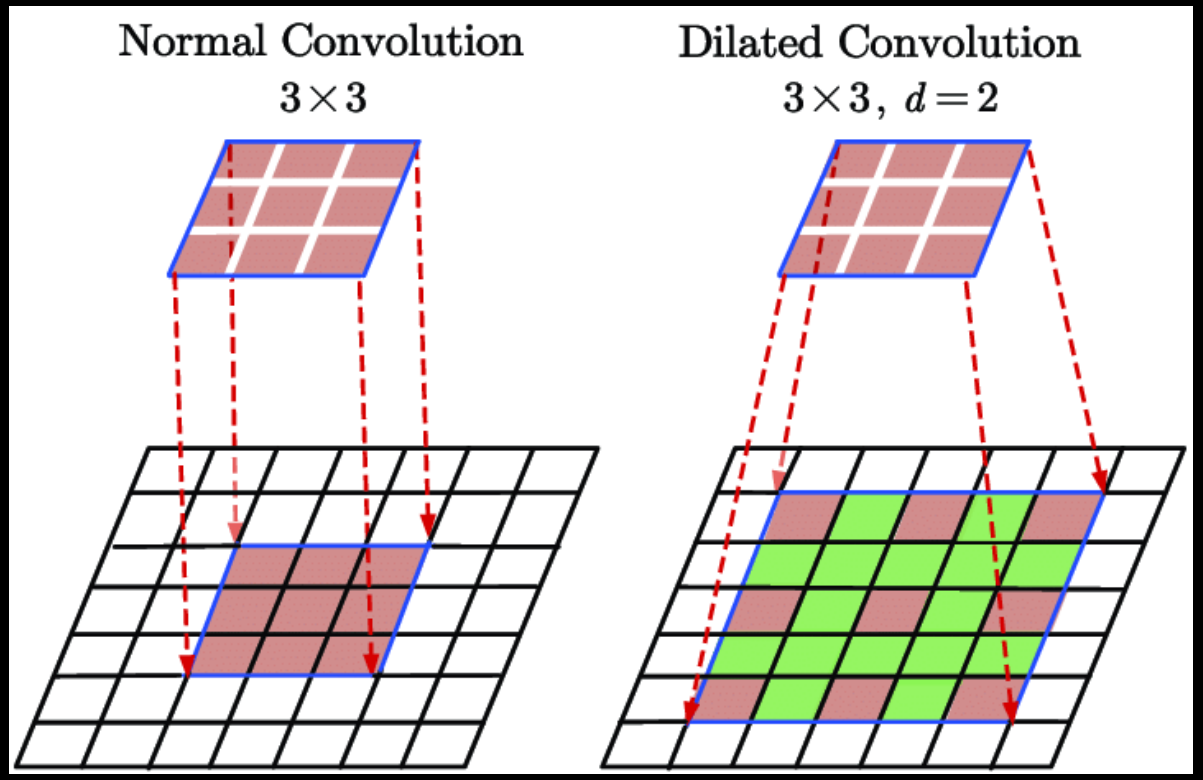
- Same basic U-Net shape
- A novel FCT block has replaced each convolutional block from U-Net
- Each FCT block is composed of a "Convolutional Attention" block and a "Wide-Focus" block
- "Patch Encoding" will convert each input image into a series of flattened token vectors to be passed through a multi-head self-attention layer
- "Wide-Focus" applies basic convolution and two layers of dilated convolution independently
- Up-sampling the decoder uses the more basic nearest neighbors

Convolutional Attention

- Tokens are created by convolving over the input with a 3x3 kernel, with ‘same’ padding to preserve the output shape. After layer normalization, the output is flattened into a token vector
- The tokens are then passed through a multi-head self-attention (MHSA) layer that uses depth-wise convolution rather than linear projections, which removes the need for positional encoding
- MHSA forms the foundation for the innovation of the Transformer architecture. The details are out-of-scope for this presentation but suffice it to say it should allow the network to learn long range spatial context from the images.

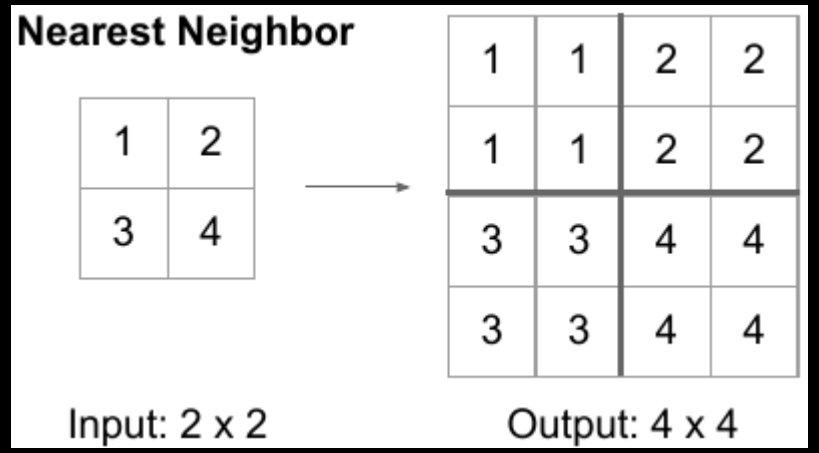
Dilated Convolution

- A generalization of the convolution operation
- Standard convolution is dilated convolution with d=1
- Expands the kernel outward, skipping over entries in the input matrix when performing the dot product
- Potentially learns salient features over a larger resolution



Nearest Neighbor Up-sampling

- The up-sampling method in FCT is simpler than the transposed convolution seen in U-Net
- Each entry in the input matrix is directly projected without any learnable weights or filters



Training & Results

Training Schemas

	U-Net	FCT
Python Library	PyTorch	PyTorch
GPU	NVIDIA 3090 (24 GB)	NVIDIA 3090 (24 GB)
Epochs*	63	46
Learning Rate	0.001	0.0001
Optimizer	Adam	Adam
Loss Function	Dice + BCE	BCE
Batch Size	2	1
Image Scale	100%	50%

*100 epochs, but with early stopping patience of 5

Results

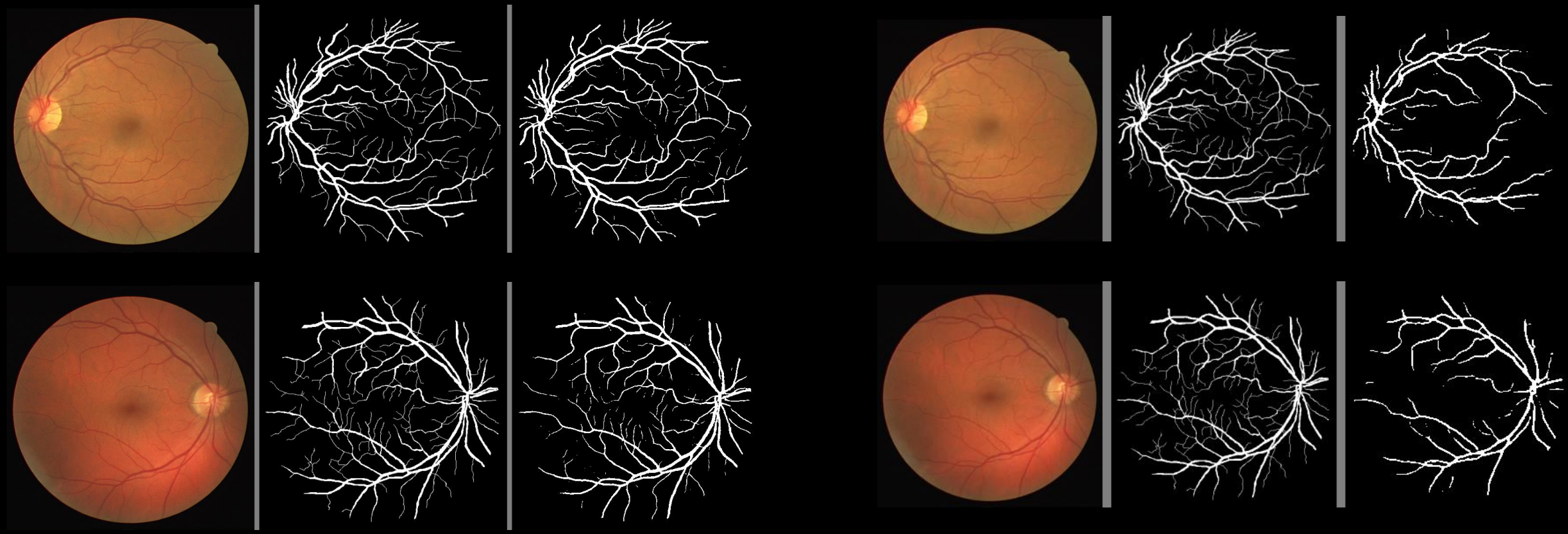
- Neither of my implementations achieve SOTA
- Likely due to a combination of more complex data augmentation and algorithms (the top SOTA model was a GAN)
- U-Net performed better than FCT, which I didn't expect
- Likely due to the rescaled images necessary for training

Metric	U-Net	FCT	SOTA
Jaccard	0.6662	0.5581	N/A
F-1 Score	0.7994	0.7154	0.8690
Precision	0.8163	0.8180	N/A
Recall	0.7878	0.6404	0.7927
Accuracy	0.9657	0.9581	0.9790
AUC ROC	0.8854	0.8137	0.9887

Test Set Prediction Examples

U-Net

FCT



Summary

Capstone Summary

- Implemented two deep learning architectures designed for image segmentation, one classic and one new: U-Net and the Fully Convolutional Transformer
- Modeled the DRIVE dataset, a small set of retina blood vessel images
- Data augmentation was employed to expand the size of the training set from 20 to 80 images
- Neither model achieved SOTA performance, but both did well
- U-Net surprisingly performed better than FCT, but likely an artifact do to training limitations
- Models of this nature offer innovative technologies to the healthcare profession

Next Steps: If I had more time

- Analyze misclassified pixels. Any patterns?
- Additional data augmentation to increase training set
- Experiment with tuning hyperparameters
- If I had a better workstation, train FCT with out rescaling

Appendix

GitHub repository containing all python code
(www.github.com/ddixonAI/SLU_Capstone)

Performance Metric Definitions:

- Jaccard Score
- F1-Score
- Precision
- Recall
- Accuracy
- AUC ROC

Loss Functions:

- BCE - Dice Loss
- BCE Loss (Binary Cross Entropy)