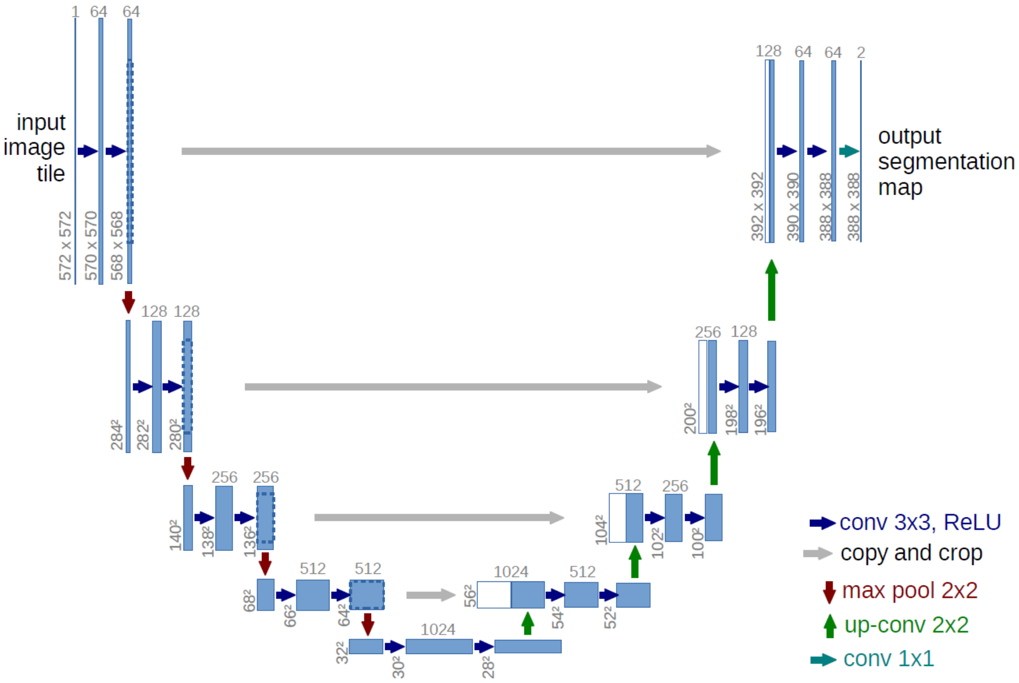
Project Report: Biomedical Segmentaion

Yash Gupta(180050121) Pratyush Agarwal(180050078)

June 24, 2020

1 U-Net: Theory and Results

1.1 Architecture



Architecture: U-Net

Our architecture is identical to the one above except 256 x 256 size patches are used instead of 572 x 572 (hence, other dimensions also change accordingly). Also, the "depth" of our architecture is variable. Changing the depth did not yield significant gains. Finally, depth = 5 was selected.

1.2 Features

1. Encoder and Decoder share a CNN type architecture, hence making use of the compatibility CNNs enjoy with images.

2. The main distinction is the decoder, where max-pooling operators are replaced by up-sampling operators. Hence, these layers increase the resolution of the output. In order to localize, high resolution

features from the contracting path are combined with the up-sampled output. A successive convolution layer can then learn to assemble a more precise output based on this information. Therefore, a good mix of local and global context is ensured.

3. In the up-sampling part, a large number of feature channels are deployed (more or less equal to the encoder), which allow the network to propagate sufficient context information to higher resolution layers.

4. Due to the scarcity of accurate data in our use-case, extensive augmentation has been used.

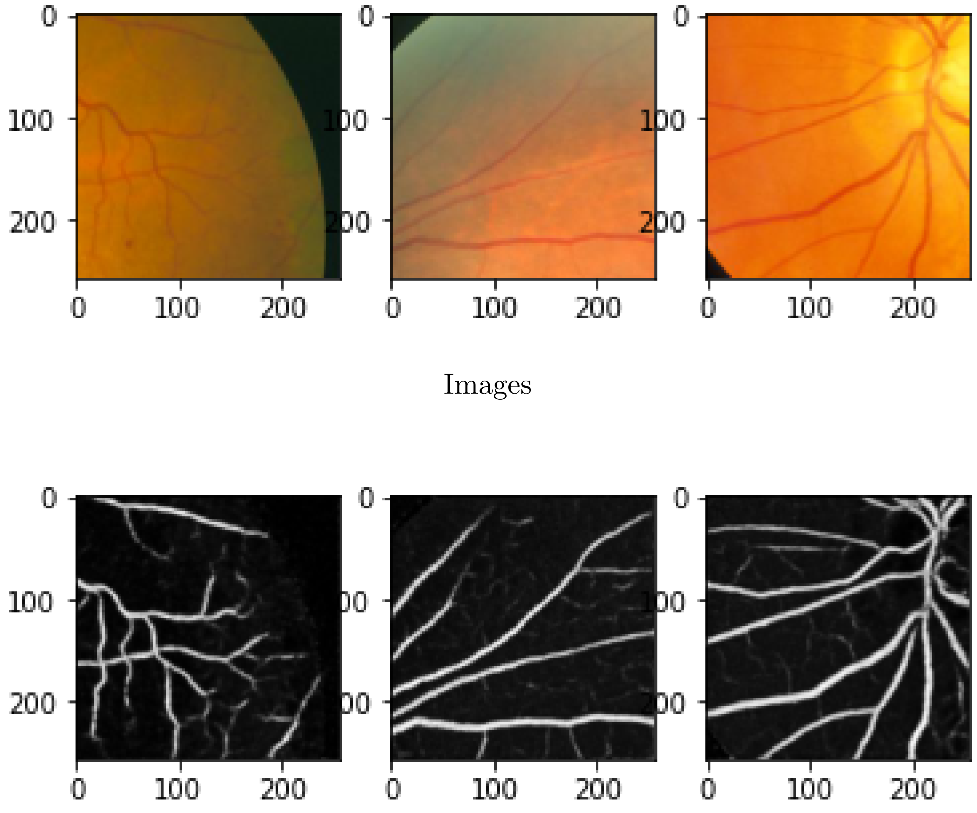
1.3 Datasets used

Blood vessel segmentations from retina images - DRIVE, STARE datasets

1.4 Results

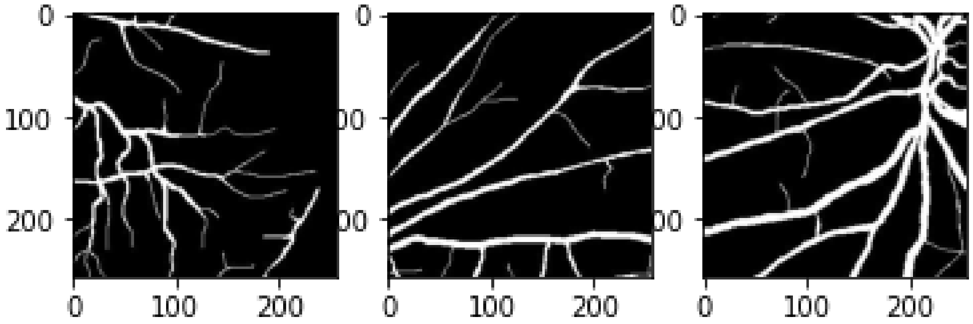
1.4.1 Performance metrics

Accuracy: 0.928
Sensitivity: 0.531
Specificity 0.991



Images

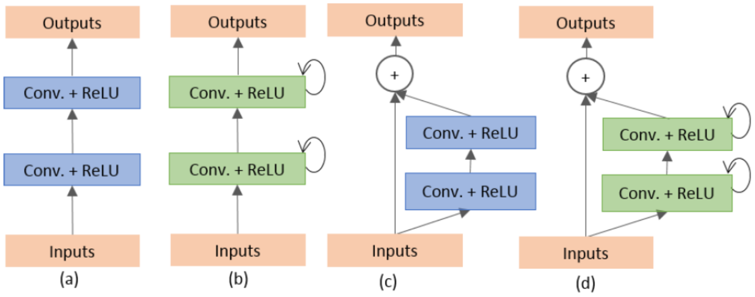
Figure 1: Predicted Masks



True Masks

2 R2U-Net: Theory and Results

2.1 Architecture



Architecture: R2U-Net

(a) - standard convulution block in U-Net.
(b) - recurrence introduced in (a).
(c) - residual units introduced in (a) for efficient learning.
(d) - recurrence units and residual units introduced simultaneously.this forms the basic block of R2U-Net. Everything else is similar to U-Net.

2.2 Features

1. Recurrent Units - The basic idea is to add recurrent connections within every convolutions layer of the feed-forward CNN, which increased the depth of the original CNN while kept the number of parameters constant by weight sharing between layers. (????)This structure enabled the units to be modulated by other units in the same layer, which enhanced the capability of the CNN to capture statistical regularities in the context of the object.(????)

2. Residual Units - helps when training the architecture. Training very deep models is difficult due to the vanishing gradient problem, which is resolved by an identity mapping to facilitate the training process.

3. Cropping and Copying unit from the basic U-Net model may be removed resulting in a much-sophisticated architecture that results in better performance. (not done in this project)

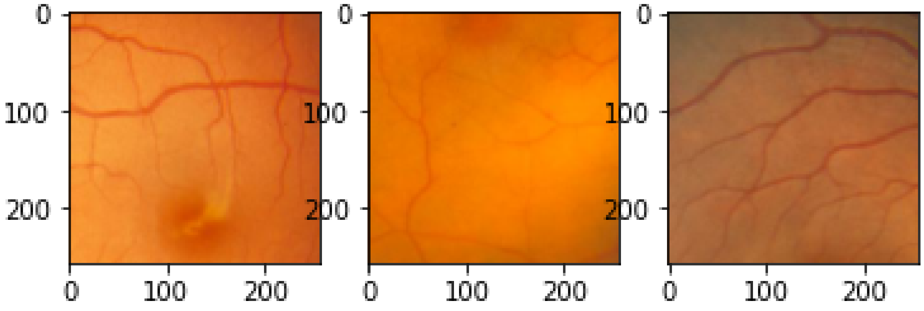
2.3 Datasets used

Blood vessel segmentations from retina images - DRIVE, STARE datasets

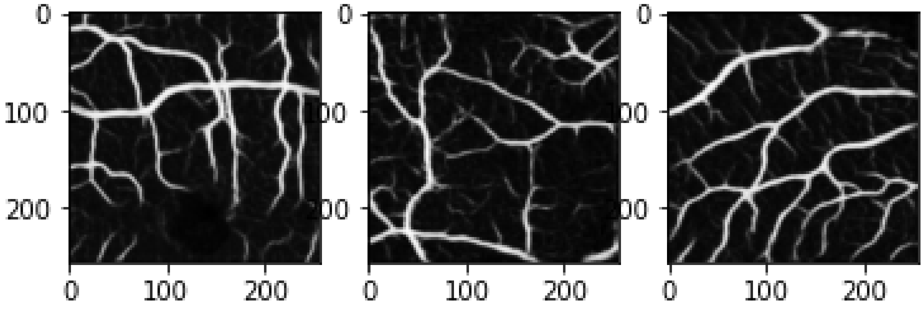
2.4 Results

2.4.1 Performance metrics

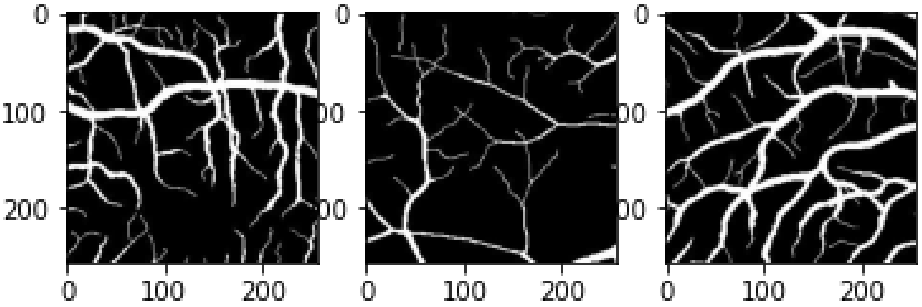
Accuracy: 0.933
Sensitivity: 0.562
Specificity 0.990



Images



Predicted Masks



True Masks

3 Variational Autoencoders : Theory and Results

3.1 Architecture and Features

1. Composed of both an encoder and a decoder and is trained to minimise the reconstruction error between the encoded-decoded data and the initial data.

2. In order to introduce some regularisation of the latent space, a slight modification of the general encoding-decoding process is used : instead of encoding an input as a single point, we encode it as a distribution over the latent space

3. The distributions returned by the encoder are enforced to be close to a standard normal distribution

3.2 Training Process

1. The input is encoded as distribution over the latent space

2. A point from the latent space is sampled from that distribution

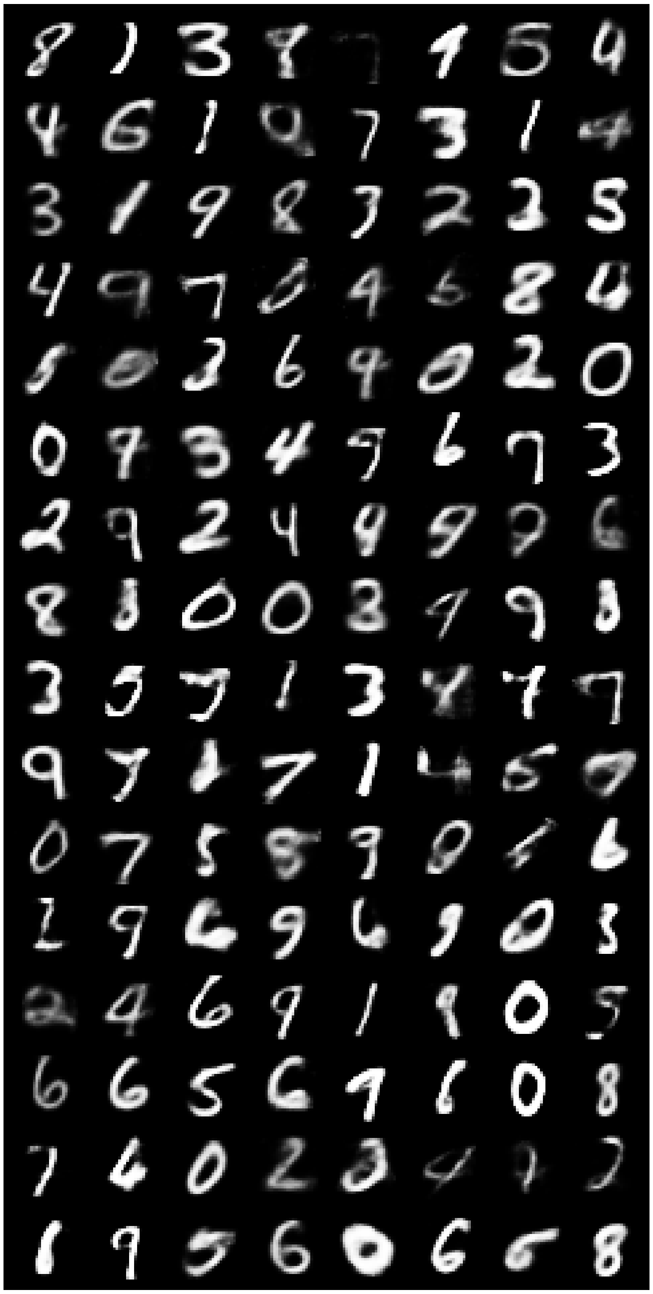
3. The sampled point is decoded and the reconstruction error can be computed

4. The reconstruction error is backpropagated through the network

3.3 Datasets used

MNIST

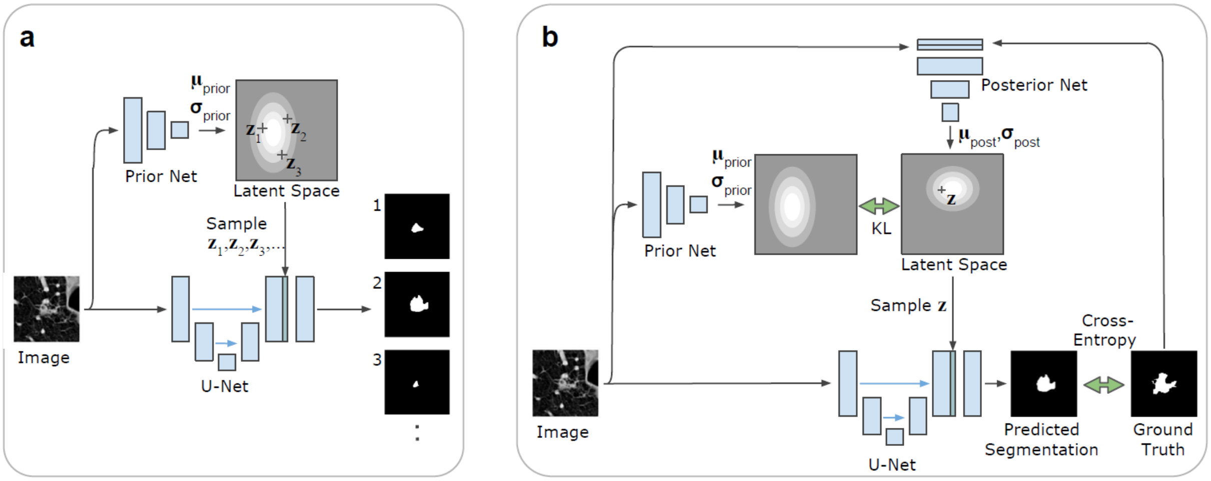
3.4 Results : Some generated images



4 Probabilistic U-Net: Theory and Results

4.1 Architecture

(a) Sampling process on the trained model. Image is sent to both UNet and Prior Net. Latent space (has low dimensions i.e. Prior Net has encoder architecture) is sampled and is concatenated (after broadcasting) with the UNet output for one last convulution. For each sample in the latent space a slightly different segmentation mask is achieved.
(b) Training process. Green arrows are the two terms of the loss function



The Probabilistic U-Net

4.2 Features

4.2.1 Sampling

1. Prior Net has learned to output a mean and variance of an axis-aligned Gaussian latent space of low dimension, where each position encodes a segmentation variant.

2. In order to obtain multiple segmentation we just sample the latent space as many times and apply the last convolution. Rest of the computation need not be repeated.

4.2.2 Training

1. U-Net is simply trained by a cross entropy term in the loss function.

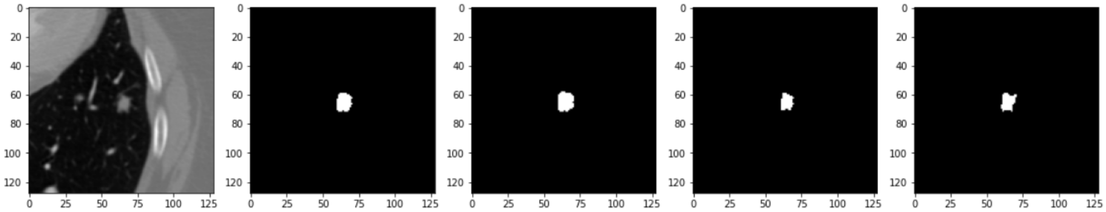
2. Prior Net is trained by forcing it to be close to a Posterior Net (but only the image as input not the segmentation) by a KL divergence term in the loss. Posterior Net learns to recognize a segmentation variant and to map this to a position with some uncertainty in the latent space. A sample from this distribution combined with the activation map of the U-Net must result in a predicted segmentation identical to the ground truth segmentation provided in the training example.

4.3 Datasets used

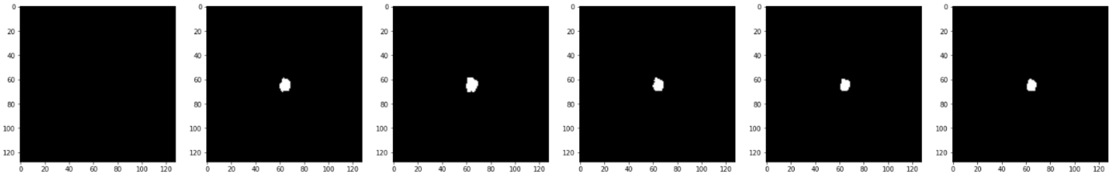
Lung Lesions

4.4 Results

Because the experts can disagree whether the lesion is abnormal tissue, some masks per image can be empty.



Input and True Masks



Predicted Masks

5 Formulae

True positive (TP) = the number of pixels correctly identified as white
False positive (FP) = the number of pixels incorrectly identified as white
True negative (TN) = the number of pixels correctly identified as black
False negative (FN) = the number of pixels incorrectly identified as black

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$