3 Methods

The goal of this project is to train a model which as input takes the two-photon microscopy images of mice brains and outputs the locations of vescicles puncta. To train the model an annotated dataset is needed, where each puncta has been marked by hand by an expert in the field. The CNN is then trained by supervised learning to predict the location of the puncta. This section describes the methods used to achieve this.

3.1 Preprocessing

As the two-photon microscopy images comes in many different resolution depending on the region of interest imaged, preprocessing is needed to make all images the same resolution. This has been achieved by finding the image with the highest resolution and pad all other images with zeros until they achieve the same resolution. For the images in this project this resolution is 1024 by 1088. To ensure the masks still correspond to the images, the same padding have been applied to the masks. The images only have one colour channel with intensities between 0 and 4095. To make this comply with our modified U-Net model, the intensities are normalized to be between 0 and 1 by first clamping all pixel intensities to be between 0 and 4095 and then dividing each pixel in the images with 4095. As the masks are binary to mark the location of puncta, this preprocessing step has not been applied to the masks. The, in total, 722 images are now split into a training set of 506 images, a validation set of 104 images and a test set of 112 images.

Unsure if this is trivial or fine to mention?

Most models have been trained on images with a resolution of 512 by 544, that is the images have been halved. A bicubic interpolation has been used for the images and a nearest neighbour interpolation has been used for the masks in all datasets. Regardless of image sizes, in order to make the images comply with the modified U-Net model, three colour channels are needed. As our images only have one colour channel, this channel has simply been replicated twice.

3.2 Models

For training, four models were developed. The first being a modified U-Net model where the encoder part has been replaced by a ResNet-18 model, transposed convolutions have been used for the upsampling layers to learn how the images are upsampled optimally, a sigmoid function have been used as activation function for the final output layer, Adam have been used for optimization, DICE have been used for loss function and its weights have been initialized with the ImageNet weights. Very similarly, the second model uses the same settings except it is not initialized with ImageNet weights. The last two models are LinkNet models using the same settings as the previous two models.

Can it be called settings?

3.3 Testing

To measure the models unbiased performance on unknown data, the test set is used. As previously described most models have been trained on halved images. For these models the predictions and masks are resized back to original sizes using the same interpolation rules. To remove false positives from the predicted masks we threshold them. This is done by thresholding all predicted masks from the training set and then computing the average DICE. The threshold that achieves the highest average DICE is then chosen, and all predicted masks from the test set are then thresholded by this value, and its unbiased performance can then be measured.

3.4 Augmented datasets

In this subsection a description of how the original images have been augmented and saved as a new dataset will be given while how these datasets have been used will be described in the following subsection. The first augmentation used is large random rotations. Here each of the original images have been rotated between -45 and 45 degrees. Another dataset of small random rotations between -30 and 30 degrees have also been created. The next augmentations flips each image horizontally and vertically giving us two more datasets. Two datasets have also been created for small and large random shearing in both the x and y direction. A *mix* dataset has also been created where the original images were first sheared then rotated and finally flipped horizontally. A dataset was also created where the original images were 'bend' around the middle to form an arc shape. The last dataset created used the python package *imgaug* to randomly apply one or more of the following augmentations to each image: horizontal flipping, vertical flipping, $80-120\%$ scaling, small translations, -45 to 45 degrees rotations, -16 to 16 degrees shearing, randomly drop $1-10\%$ of the pixels in the image, locally move some pixels around and lastly, perform local distortions.

3.5 Construction of augmented training sets

With the augmented datasets, several training sets were constructed and used to train models. Each training set consists of 506 training images, and is constructed by taking an even amount of images from each involved dataset. That is, if we construct a training set from two datasets, a list with length 506 is constructed and filled randomly with ones and zeros. Then iterating through this list, each time a 0 is encountered the image with index corresponding to the index in the list is chosen from the first dataset, and every time a 1 is encountered the image with index corresponding to the index in the list is chosen from the second dataset. This ensures we get different images in the final training set and never the same image just augmented differently. This method extends to several datasets. The training sets in the results section comes from combining datasets like this .

Very much in doubt whether to keep the description like this or list the training sets. But the listing becomes 10+ datasets, and listing the names seems quite irrelevant.