Homework 02

ELE 2346 - DEEP LEARNING

Pedro Henrique Cardoso Paulo
pedrorjpaulo.phcp@gmail.com

Professor: Raul Queiroz Feitosa

Departamento de Engenharia Mecânica
PUC-RJ Pontifícia Universidade Católica do Rio de Janeiro
April, 2023

Homework 02

ELE 2346 - DEEP LEARNING

Pedro Henrique Cardoso Paulo

April, 2023

1 Introduction

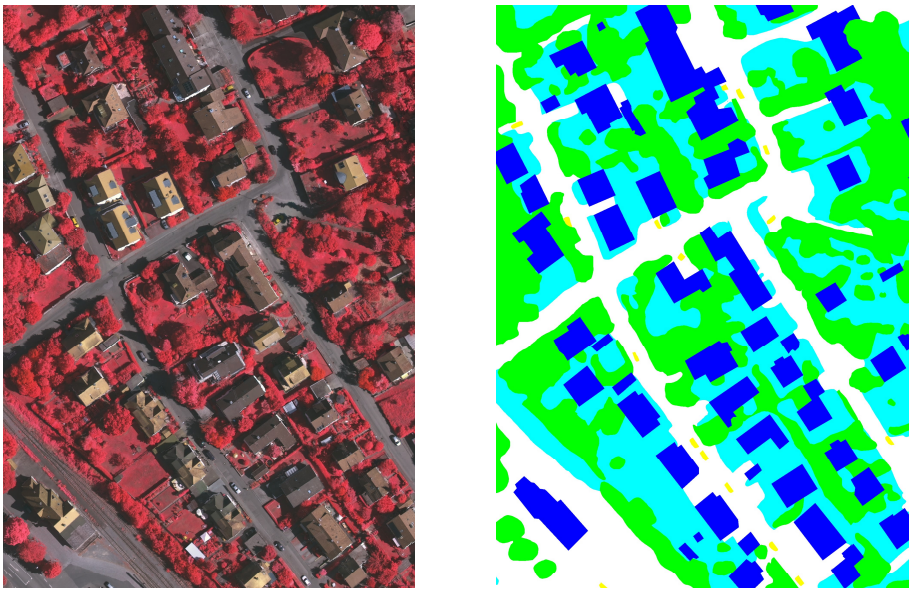
1.1 Objectives

The main objectives of this exercice is to provide the students some experience with:

- The PyTorch library
- Segmentation models
- The patch strategy for dealing with big images

1.2 Dataset

For this exercice, the dataset will be comprised of two images that will be splitted into train, validation and test sets by creating patches of the image. Figure 1 shows the images



(a) Train set **(b)** Train mask

Figure 1: Train image to be used in the exercice

1.3 Exercice

The main objextive of this notebook is to implement a DeepLabV3+ Network for semantic segmentation using Segmentation Models of PyTorch. The following steps should be followed:

1. Split the train image for training (80%) and validation (20%)
2. Generate patches from the training image (128x128, 64x64, 32x32)

3. Train a DeepLabV3+ network using the Segmentation Models of pytorch using the ecoder `'mobilenet_v2'` from scratch and with pretrained weights of `'imagenet'`, `encoder_depth = 4` and `decoder_channels = [256, 128, 64, 32]`
4. For training, use the `\weighted_categorical_crossentropy"` as a loss function. To compute the weights you must count the number of pixels of each class
5. Evaluate the model on the test image using patches and make the mosaic to visualize the complete image
6. Compute metrics for each class
7. Compare and analize the results

Use the following mean and standard deviation values according to the adopted weights initialization:

```
#Normalization for ImageNet
mean_ = [0.485, 0.456, 0.406]
std_  = [0.229, 0.224, 0.225]

#Normalization for trainning from scratch
mean_ = 0.0
std_  = 1.0
```

1.4 Cases of study

From the proposed exercice, the following tests will be performed and compared regardign general error metrics and confusion matrices:

1. Using imagenet weights as starting values
   a Creating patches of 32 x 32 pixels
   b Creating patches of 64 x 64 pixels
   c Creating patches of 128 x 128 pixels
2. Using randomly initiated weights
   a Creating patches of 32 x 32 pixels
   b Creating patches of 64 x 64 pixels
   c Creating patches of 128 x 128 pixels

2 Results and discussions

2.1 Colab Notebook

All the code, examples and tests made are documented on the following Colab Notebook.

- Link to the Colab Notebook

2.2 Code Highlights

```
class PatchDataset(Dataset):

    def __init__(self, img_path, mask_path, mean, std, transform=None, n_patches=32, stride=1)
                                                                    :
        self.img_path = img_path
        self.mask_path = mask_path
        self.transform = transform
        self.n_patches = n_patches
        self.stride = stride
        self.mean = mean
        self.std = std

        #Reading the image
        img = cv2.imread(self.img_path)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        mask = cv2.imread(self.mask_path, cv2.IMREAD_GRAYSCALE)
        self.img = img
```

```
        self.mask = mask
        orig_gray_lbls = np.sort(np.unique(mask))
        self.label_dict = {}
        self.classes_weights = []
        class_i = 0
        for orig_lbl in orig_gray_lbls:
          mask[mask == orig_lbl] = class_i
          self.label_dict[class_i] = orig_lbl
          print(f'Total number of elements in class {class_i}: {np.sum(mask == class_i)} ({np.
                                                    sum(mask == class_i)/mask.size})')
          self.classes_weights.append(mask.size/np.sum(mask == class_i))
          class_i += 1

        img = torch.from_numpy(img).long()
        mask = torch.from_numpy(mask).long()

        self.img_patches = img.unfold(0,self.n_patches,self.stride).unfold(1,self.n_patches,
                                                    self.stride)
        self.mask_patches = mask.unfold(0,self.n_patches,self.stride).unfold(1,self.n_patches,
                                                    self.stride)
        #self.img_patches = F.unfold(img,kernel_size=self.n_patches,stride=self.stride)#.
                                                    unfold(1,self.n_patches,self.stride)
        #self.mask_patches = F.unfold(mask,kernel_size=self.n_patches,stride=self.stride)#.
                                                    unfold(1,self.n_patches,self.stride)
        self.img_patches_shape = self.img_patches.shape

    def __len__(self):
        self.img_patches_shape = self.img_patches.shape
        return self.img_patches_shape[0]*self.img_patches_shape[1]

    def __getitem__(self, idx):

        self.img_patches_shape = self.img_patches.shape

        idx_x = idx % self.img_patches_shape[0]
        idx_y = idx // self.img_patches_shape[0]

        #print(idx_x, idx_y)
        img = self.img_patches[idx_x, idx_y, :, :, :]#.numpy()#.astype('int32')
        img = img.transpose(0,2).transpose(0,1).numpy().astype(np.uint8)
        #img2 = np.zeros((self.n_patches, self.n_patches, 3), dtype=np.uint8)
        #img2[:, :, 0] = img[0, :, :]
        #img2[:, :, 1] = img[1, :, :]
        #img2[:, :, 2] = img[2, :, :]

        #img = img2.copy()
        #del img2

        mask = self.mask_patches[idx_x, idx_y, :,  :].numpy()

        if self.transform is not None:
            aug = self.transform(image=img, mask=mask)
            img = Image.fromarray(aug['image'])
            mask = aug['mask']

        if self.transform is None:
            img = Image.fromarray(img)

        t = T.Compose([T.ToTensor(), T.Normalize(self.mean, self.std)])
        img = t(img)
        mask = torch.from_numpy(mask).long()

        return img, mask
```

2.3 Hyperparameter selection

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis

nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2.4 Item 1a

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

(a) Loss evolution **(b)** Score evolution **(c)** Accuracy evolution

Figure 2: Metrics evolution during training for item 1a

2.5 Item 1b

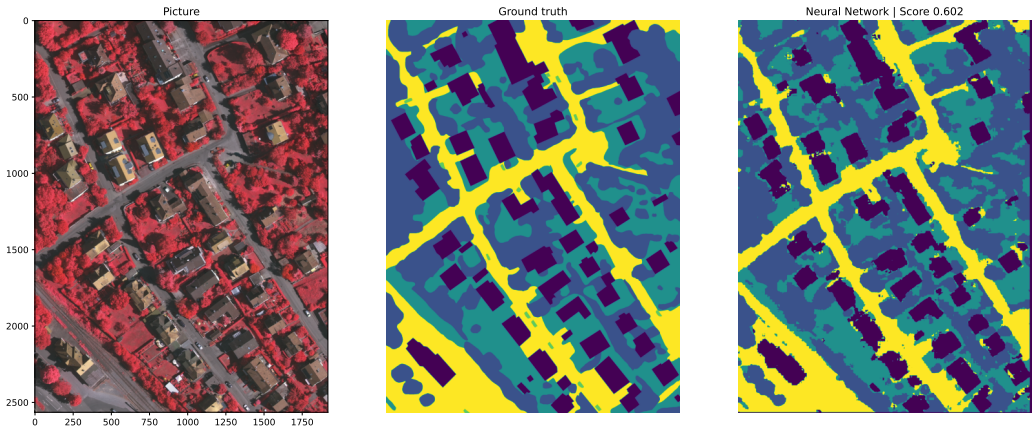
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2.6 Item 1c

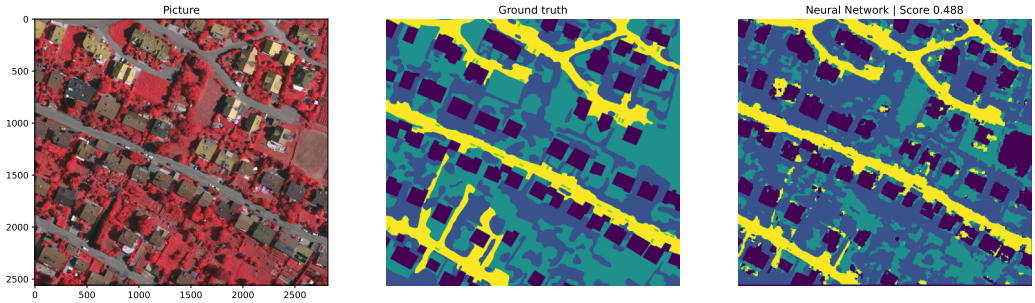
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2.7 Item 2a

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



(a) Train results

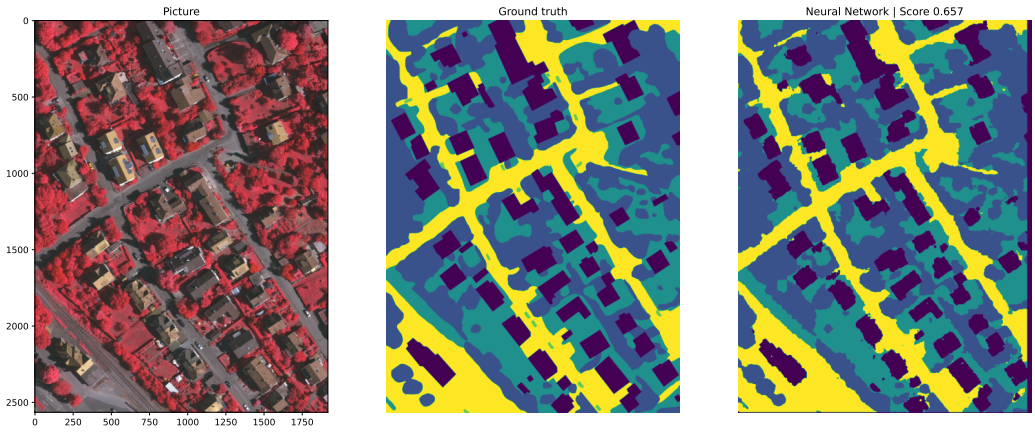


(b) test results

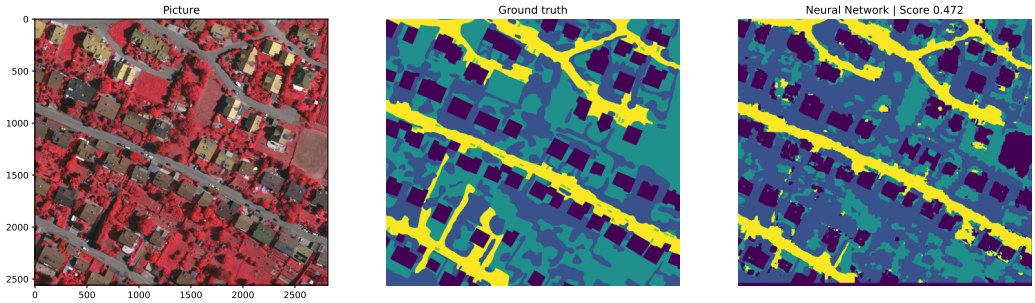
Figure 3: Results of predictions for item 1a

(a) Loss evolution **(b)** Score evolution **(c)** Accuracy evolution

Figure 4: Metrics evolution during training for item 1b



(a) Train results

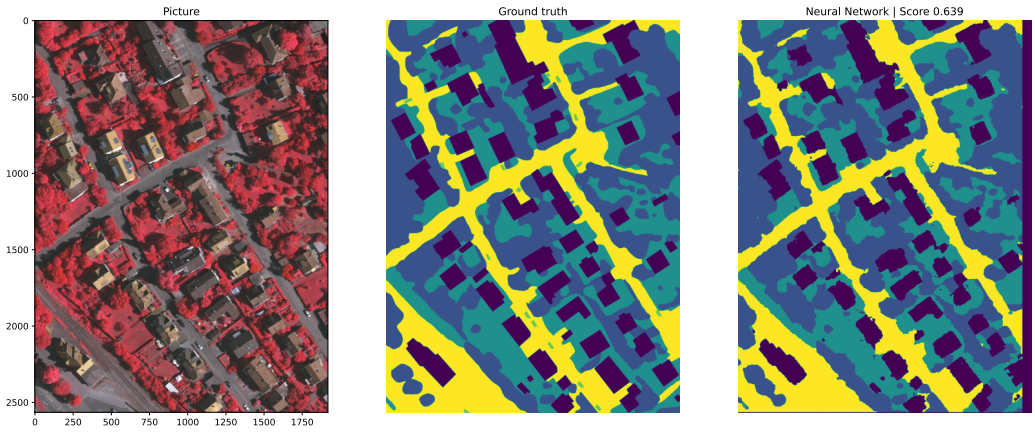


(b) test results

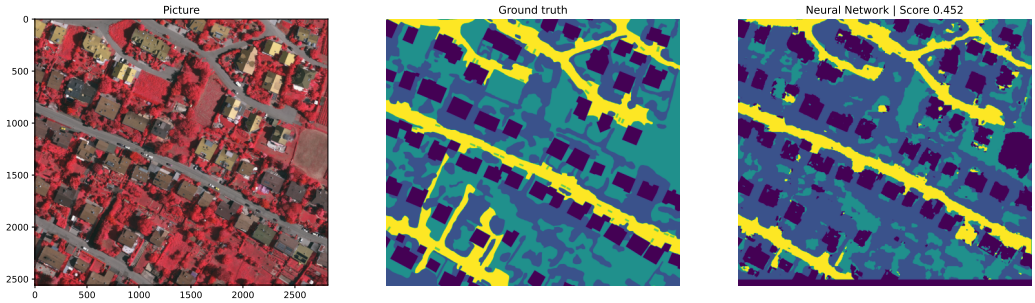
Figure 5: Results of predictions for item 1b

(a) Loss evolution **(b)** Score evolution **(c)** Accuracy evolution

Figure 6: Metrics evolution during training for item 1c



(a) Train results

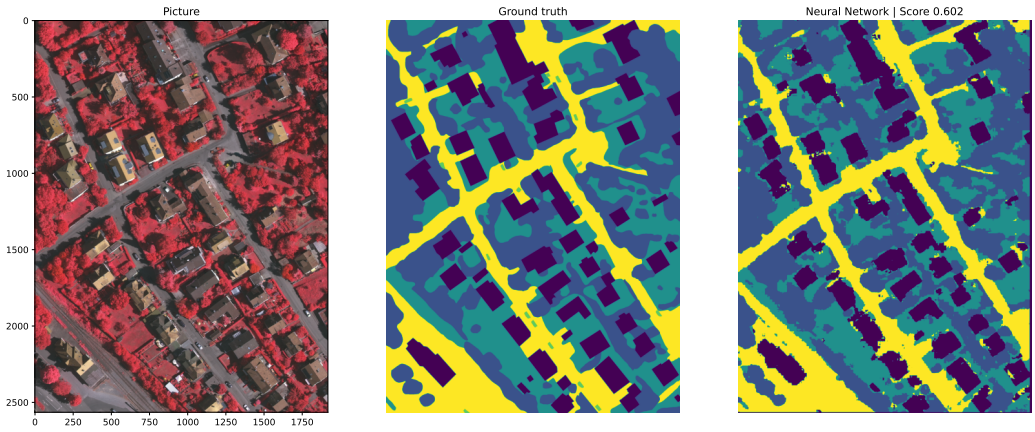


(b) test results

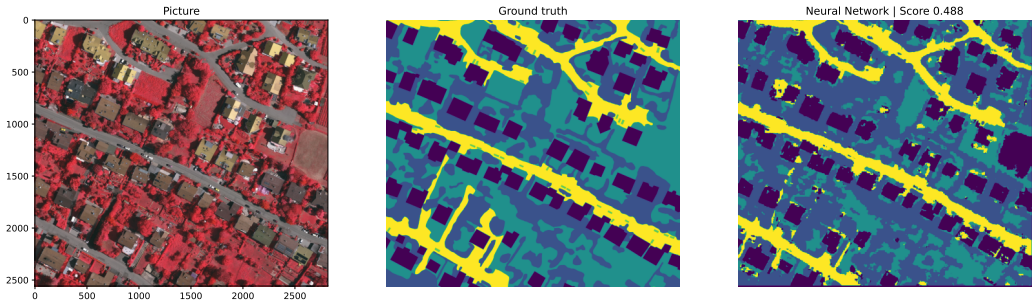
Figure 7: Results of predictions for item 1c

(a) Loss evolution **(b)** Score evolution **(c)** Accuracy evolution

Figure 8: Metrics evolution during training for item 2a



(a) Train results



(b) test results

Figure 9: Results of predictions for item 2a

2.8 Item 2b

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

(a) Loss evolution **(b)** Score evolution **(c)** Accuracy evolution

Figure 10: Metrics evolution during training for item 2b

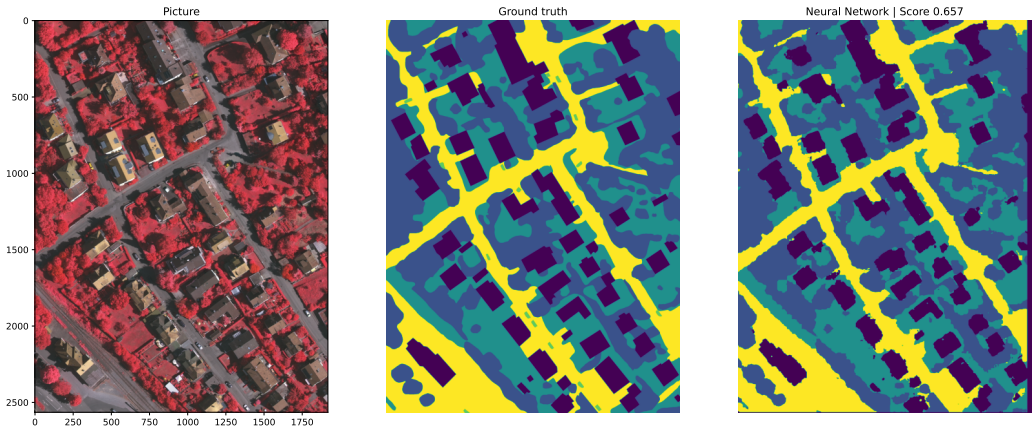
2.9 Item 2c

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

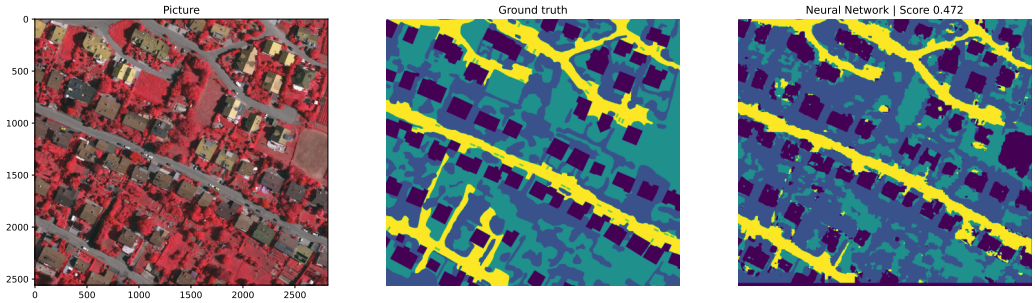
2.10 Results Summary and Conclusions

Exp	train acc	train loss	val acc	val loss	test acc	test loss	training time
1a	94.00%	0.159551	94.74%	0.174357	92.19%	0.229864	14 min
1b	87.43%	0.329751	92.48%	0.233386	85.94%	0.390936	06 min
1c	98.26%	0.055598	98.50%	0.056195	96.09%	0.112185	03 min
2a	95.84%	0.121377	96.24%	0.082859	91.41%	0.176876	04 min
2b	95.84%	0.116910	98.50%	0.076800	94.53%	0.131344	11 min
2c	95.84%	0.116910	98.50%	0.076800	94.53%	0.131344	11 min

Table 1: Results summary



(a) Train results

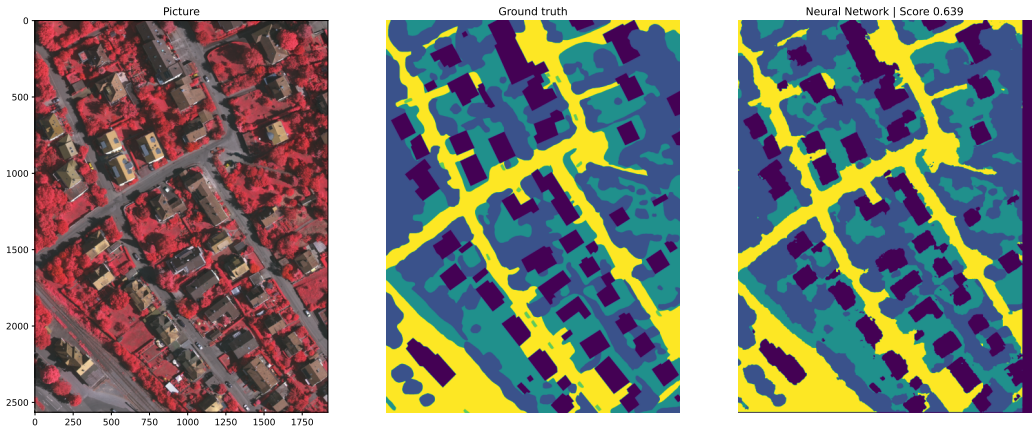


(b) test results

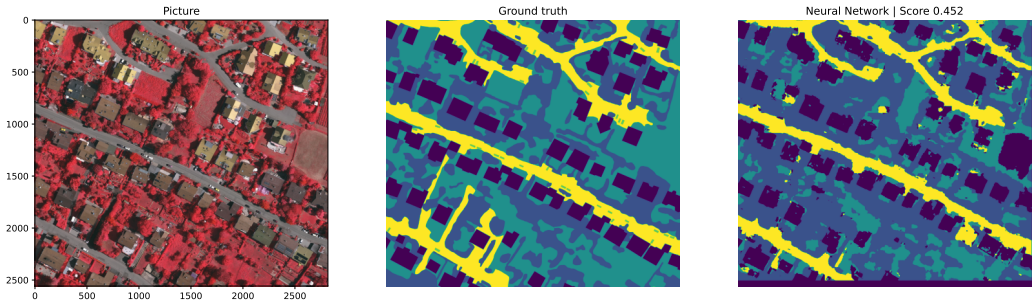
Figure 11: Results of predictions for item 2b

(a) Loss evolution **(b)** Score evolution **(c)** Accuracy evolution

Figure 12: Metrics evolution during training for item 2c



(a) Train results



(b) test results

Figure 13: Results of predictions for item 2c