DAT300 - Compulsory assignment 2

Group 2

Fight Club Goofy Edition

Orion username:

dat300-22-9

Members

- Joel Yacob
- Artush Mkrtchyan

Introduction

The problem that we are going to solve is to differantiate between roads and everything else using pictures of roads as training data. The pictures have three channels; red, green and blue. The different parts in the pictures are already classified, therefore we have to make this a binary problem where roads are categorised as one thing, while everything else falls under another category called 'other'.

We are then going to use U-net and tune the parameters to then train the model. After that we are going to use VGG16 and compare that method with the manual U-net to see which model performed the best.

And last, but not least, we are going to upload these models to Orion and see if there is any difference in accuracy and total time used.

Data handling and visualisation

In [1]:

```
# Import and extraction of data.
import numpy as np
import h5py
import matplotlib.pyplot as plt
import tensorflow as tf
import pandas as pd

from sklearn.model_selection import train_test_split
from tensorflow.keras.layers import Input, Conv2D, BatchNormalization, Activation, MaxPool
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ModelCheckpoint, LearningRateScheduler, EarlyStoppi
from keras import backend as K
from keras.applications.vgg16 import VGG16 as vg, preprocess_input
```

As seen below, we are defining the training and testing data

In [2]:

```
train_data = h5py.File("../input/ca2-test-and-train/train.h5")
test_data = h5py.File("../input/ca2-test-and-train/test.h5")
```

In [3]:

```
print(train_data.keys())
print(test_data.keys())
```

```
<KeysViewHDF5 ['X', 'y']>
<KeysViewHDF5 ['X']>
```

In [4]:

```
X = train_data["X"][:]
y = train_data["y"][:]

X_test_final = test_data["X"][:]
```

In [38]:

```
print("Shape of X: ", X.shape)
print("Shape of y: ", y.shape)
print("Shape of X_test_final: ", X_test_final.shape)
```

```
Shape of X:  (2780, 128, 128, 3)
Shape of y:  (2780, 128, 128, 1)
Shape of X_test_final:  (695, 128, 128, 3)
```

As seen on shape of X, the pictures are all 128 x 128 with 3 channels.

In [6]:

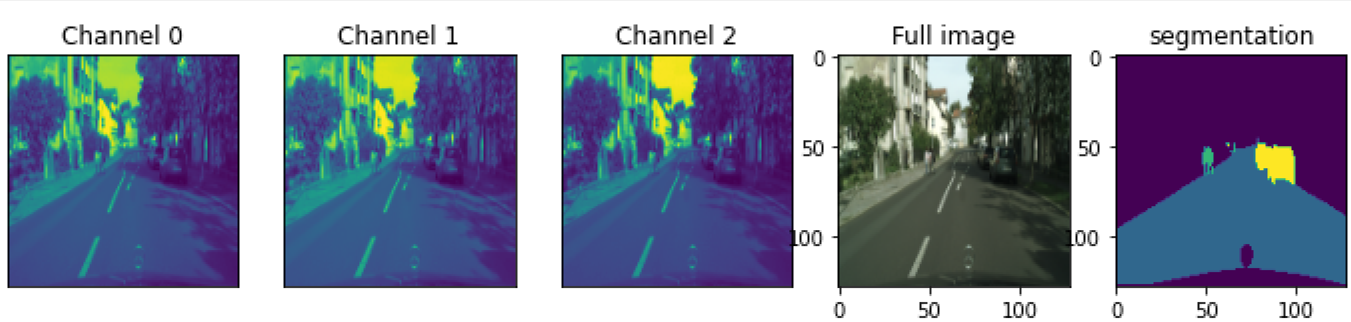
```
# Short exploration and visualisation of dataset (point 1 in Canvas).
print("Height of image: ",X.shape[1])
print("Width of image: ",X.shape[2])
print("Channels of image: ",X.shape[3])
```

```
Height of image:  128
Width of image:  128
Channels of image:  3
```

The code below is to show the three channels, the original picture and how the diffent parts are segmentated.

In [7]:

```
fig, ax = plt.subplots(1,5, figsize=(12,12))

for i , axis in enumerate(ax[:3]):
    axis.imshow(X[0][:,:,i])
    axis.title.set_text(f'Channel {i}')
    axis.set_xticks([])
    axis.set_yticks([])

ax[3].imshow(X[0])
ax[3].title.set_text("Full image")

ax[4].imshow(y[0])
ax[4].title.set_text("segmentation")

plt.show()
```



In channel 0, we see where there are most reds, in channel 1 the same but with greens, and channel 2 with blues. The last picture shows the segmentation of the different parts (cars, roads, people and other things).

Methods

Since the dataset was particularly large, we did not perform gridsearch and such strategies to get better tuned hyperparameters. Other than splitting the dataset for training and testing, we did not use much advanced methods for this task.

We first tried a 2D U-Net, then we combined our U-Net with the pretrained VGG16 for a better result. The last method we used was multiclass segmentation with U-Net, we had to convert our U-Net function to a 3D approach.

Preprocessing

Splitting into train and test using the train data before transforming the train labels into a binary problem.

In [8]:

```
# Code for preprocessing of the data and transformation of labels for the binary problem

y = np.where(y != 0, 1,0) # Transform the train labels into a binary problem

X_train, X_test, y_train, y_test = train_test_split(
    X,y, test_size=0.1, random_state=69)
```

Results

Functions of U-net below:

In [9]:

```
# Code and model training for your best Basic U-Net model (point 3 in Canvas)

def conv2d_block(input_tensor, n_filters, kernel_size = 3, batchnorm = True):
    """Function to add 2 convolutional layers with the parameters passed to it"""
    # first layer
    x = Conv2D(filters = n_filters, kernel_size = (kernel_size, kernel_size),\
              kernel_initializer = 'he_normal', padding = 'same')(input_tensor)
    if batchnorm:
        x = BatchNormalization()(x)
    x = Activation('relu')(x)

    # second layer
    x = Conv2D(filters = n_filters, kernel_size = (kernel_size, kernel_size),\
              kernel_initializer = 'he_normal', padding = 'same')(x)
    if batchnorm:
        x = BatchNormalization()(x)
    x = Activation('relu')(x)

    return x


def get_unet(input_img, n_filters = 16, dropout = 0.1, batchnorm = True, n_classes = 2):

    # Contracting Path
    c1 = conv2d_block(input_img, n_filters * 1, kernel_size = 3, batchnorm = batchnorm)
    p1 = MaxPooling2D((2, 2))(c1)
    p1 = Dropout(dropout)(p1)

    c2 = conv2d_block(p1, n_filters * 2, kernel_size = 3, batchnorm = batchnorm)
    p2 = MaxPooling2D((2, 2))(c2)
    p2 = Dropout(dropout)(p2)

    c3 = conv2d_block(p2, n_filters * 4, kernel_size = 3, batchnorm = batchnorm)
```

```python
    p3 = MaxPooling2D((2, 2))(c3)
    p3 = Dropout(dropout)(p3)

    c4 = conv2d_block(p3, n_filters * 8, kernel_size = 3, batchnorm = batchnorm)
    p4 = MaxPooling2D((2, 2))(c4)
    p4 = Dropout(dropout)(p4)

    c5 = conv2d_block(p4, n_filters = n_filters * 16, kernel_size = 3, batchnorm = batchno

    # Expansive Path
    u6 = Conv2DTranspose(n_filters * 8, (3, 3), strides = (2, 2), padding = 'same')(c5)
    u6 = concatenate([u6, c4])
    u6 = Dropout(dropout)(u6)
    c6 = conv2d_block(u6, n_filters * 8, kernel_size = 3, batchnorm = batchnorm)

    u7 = Conv2DTranspose(n_filters * 4, (3, 3), strides = (2, 2), padding = 'same')(c6)
    u7 = concatenate([u7, c3])
    u7 = Dropout(dropout)(u7)
    c7 = conv2d_block(u7, n_filters * 4, kernel_size = 3, batchnorm = batchnorm)

    u8 = Conv2DTranspose(n_filters * 2, (3, 3), strides = (2, 2), padding = 'same')(c7)
    u8 = concatenate([u8, c2])
    u8 = Dropout(dropout)(u8)
    c8 = conv2d_block(u8, n_filters * 2, kernel_size = 3, batchnorm = batchnorm)

    u9 = Conv2DTranspose(n_filters * 1, (3, 3), strides = (2, 2), padding = 'same')(c8)
    u9 = concatenate([u9, c1])
    u9 = Dropout(dropout)(u9)
    c9 = conv2d_block(u9, n_filters * 1, kernel_size = 3, batchnorm = batchnorm)

    outputs = Conv2D(n_classes, (1, 1), activation='sigmoid')(c9)
    model = Model(inputs=[input_img], outputs=[outputs])
    return model
```

In [10]:

```python
input_img = Input(shape=(128,128,3))
model = get_unet(input_img, n_filters = 32, dropout = 0.1, batchnorm = True, n_classes = 1
model.summary()
```

```
2022-11-09 12:04:05.515727: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:05.516780: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:05.864303: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:05.865158: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:05.865920: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:05.866666: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:05.868073: I tensorflow/core/platform/cpu_feature_guard.cc:142] This Tens
orFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the fol
lowing CPU instructions in performance-critical operations:  AVX2 AVX512F FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flag
s.
2022-11-09 12:04:06.120224: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
```

```
2022-11-09 12:04:06.121124: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:06.121890: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:06.122654: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:06.123347: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:06.124018: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:10.873350: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:10.874342: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:10.875145: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:10.875884: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:10.876608: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:10.877318: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Creat
ed device /job:localhost/replica:0/task:0/device:GPU:0 with 13789 MB memory:  -> device:
0, name: Tesla T4, pci bus id: 0000:00:04.0, compute capability: 7.5
2022-11-09 12:04:10.882349: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] su
ccessful NUMA node read from SysFS had negative value (-1), but there must be at least one
NUMA node, so returning NUMA node zero
2022-11-09 12:04:10.883113: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Creat
ed device /job:localhost/replica:0/task:0/device:GPU:1 with 13789 MB memory:  -> device:
1, name: Tesla T4, pci bus id: 0000:00:05.0, compute capability: 7.5
Model: "model"
__________________________________________________________________________________________
________
Layer (type)                    Output Shape         Param #     Connected to

==========================================================================================
========
input_1 (InputLayer)            [(None, 128, 128, 3) 0


__________________________________________________________________________________________
________
conv2d (Conv2D)                 (None, 128, 128, 32) 896         input_1[0][0]


__________________________________________________________________________________________
________
batch_normalization (BatchNorma (None, 128, 128, 32) 128         conv2d[0][0]


__________________________________________________________________________________________
________
activation (Activation)         (None, 128, 128, 32) 0           batch_normalization[0][0]


__________________________________________________________________________________________
________
conv2d_1 (Conv2D)               (None, 128, 128, 32) 9248        activation[0][0]


__________________________________________________________________________________________
________
```

```
batch_normalization_1 (BatchNor (None, 128, 128, 32) 128         conv2d_1[0][0]

________________________________________________________________________________________________________________
activation_1 (Activation)       (None, 128, 128, 32) 0           batch_normalization_1[0]
[0]
________________________________________________________________________________________________________________
max_pooling2d (MaxPooling2D)    (None, 64, 64, 32)   0           activation_1[0][0]

________________________________________________________________________________________________________________
dropout (Dropout)               (None, 64, 64, 32)   0           max_pooling2d[0][0]

________________________________________________________________________________________________________________
conv2d_2 (Conv2D)               (None, 64, 64, 64)   18496       dropout[0][0]

________________________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, 64, 64, 64)   256         conv2d_2[0][0]

________________________________________________________________________________________________________________
activation_2 (Activation)       (None, 64, 64, 64)   0           batch_normalization_2[0]
[0]
________________________________________________________________________________________________________________
conv2d_3 (Conv2D)               (None, 64, 64, 64)   36928       activation_2[0][0]

________________________________________________________________________________________________________________
batch_normalization_3 (BatchNor (None, 64, 64, 64)   256         conv2d_3[0][0]

________________________________________________________________________________________________________________
activation_3 (Activation)       (None, 64, 64, 64)   0           batch_normalization_3[0]
[0]
________________________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, 32, 32, 64)   0           activation_3[0][0]

________________________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 32, 32, 64)   0           max_pooling2d_1[0][0]

________________________________________________________________________________________________________________
conv2d_4 (Conv2D)               (None, 32, 32, 128)  73856       dropout_1[0][0]

________________________________________________________________________________________________________________
batch_normalization_4 (BatchNor (None, 32, 32, 128)  512         conv2d_4[0][0]

________________________________________________________________________________________________________________
activation_4 (Activation)       (None, 32, 32, 128)  0           batch_normalization_4[0]
[0]
________________________________________________________________________________________________________________
conv2d_5 (Conv2D)               (None, 32, 32, 128)  147584      activation_4[0][0]

________________________________________________________________________________________________________________
batch_normalization_5 (BatchNor (None, 32, 32, 128)  512         conv2d_5[0][0]
```

```
__________________________________________________________________________________________________
activation_5 (Activation)       (None, 32, 32, 128)  0           batch_normalization_5[0][0]
__________________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D)  (None, 16, 16, 128)  0           activation_5[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 16, 16, 128)  0           max_pooling2d_2[0][0]
__________________________________________________________________________________________________
conv2d_6 (Conv2D)               (None, 16, 16, 256)  295168      dropout_2[0][0]
__________________________________________________________________________________________________
batch_normalization_6 (BatchNor (None, 16, 16, 256)  1024        conv2d_6[0][0]
__________________________________________________________________________________________________
activation_6 (Activation)       (None, 16, 16, 256)  0           batch_normalization_6[0][0]
__________________________________________________________________________________________________
conv2d_7 (Conv2D)               (None, 16, 16, 256)  590080      activation_6[0][0]
__________________________________________________________________________________________________
batch_normalization_7 (BatchNor (None, 16, 16, 256)  1024        conv2d_7[0][0]
__________________________________________________________________________________________________
activation_7 (Activation)       (None, 16, 16, 256)  0           batch_normalization_7[0][0]
__________________________________________________________________________________________________
max_pooling2d_3 (MaxPooling2D)  (None, 8, 8, 256)    0           activation_7[0][0]
__________________________________________________________________________________________________
dropout_3 (Dropout)             (None, 8, 8, 256)    0           max_pooling2d_3[0][0]
__________________________________________________________________________________________________
conv2d_8 (Conv2D)               (None, 8, 8, 512)    1180160     dropout_3[0][0]
__________________________________________________________________________________________________
batch_normalization_8 (BatchNor (None, 8, 8, 512)    2048        conv2d_8[0][0]
__________________________________________________________________________________________________
activation_8 (Activation)       (None, 8, 8, 512)    0           batch_normalization_8[0][0]
__________________________________________________________________________________________________
conv2d_9 (Conv2D)               (None, 8, 8, 512)    2359808     activation_8[0][0]
__________________________________________________________________________________________________
batch_normalization_9 (BatchNor (None, 8, 8, 512)    2048        conv2d_9[0][0]
__________________________________________________________________________________________________
```

```
activation_9 (Activation)        (None, 8, 8, 512)    0           batch_normalization_9[0]
[0]
__________________________________________________________________________________________
________
conv2d_transpose (Conv2DTranspo (None, 16, 16, 256)  1179904     activation_9[0][0]

__________________________________________________________________________________________
________
concatenate (Concatenate)        (None, 16, 16, 512)  0           conv2d_transpose[0][0]

                                                                  activation_7[0][0]

__________________________________________________________________________________________
________
dropout_4 (Dropout)              (None, 16, 16, 512)  0           concatenate[0][0]

__________________________________________________________________________________________
________
conv2d_10 (Conv2D)               (None, 16, 16, 256)  1179904     dropout_4[0][0]

__________________________________________________________________________________________
________
batch_normalization_10 (BatchNo (None, 16, 16, 256)  1024        conv2d_10[0][0]

__________________________________________________________________________________________
________
activation_10 (Activation)       (None, 16, 16, 256)  0           batch_normalization_10[0]
[0]
__________________________________________________________________________________________
________
conv2d_11 (Conv2D)               (None, 16, 16, 256)  590080      activation_10[0][0]

__________________________________________________________________________________________
________
batch_normalization_11 (BatchNo (None, 16, 16, 256)  1024        conv2d_11[0][0]

__________________________________________________________________________________________
________
activation_11 (Activation)       (None, 16, 16, 256)  0           batch_normalization_11[0]
[0]
__________________________________________________________________________________________
________
conv2d_transpose_1 (Conv2DTrans (None, 32, 32, 128)  295040      activation_11[0][0]

__________________________________________________________________________________________
________
concatenate_1 (Concatenate)      (None, 32, 32, 256)  0           conv2d_transpose_1[0][0]

                                                                  activation_5[0][0]

__________________________________________________________________________________________
________
dropout_5 (Dropout)              (None, 32, 32, 256)  0           concatenate_1[0][0]

__________________________________________________________________________________________
________
conv2d_12 (Conv2D)               (None, 32, 32, 128)  295040      dropout_5[0][0]

__________________________________________________________________________________________
________
batch_normalization_12 (BatchNo (None, 32, 32, 128)  512         conv2d_12[0][0]

__________________________________________________________________________________________
________
activation_12 (Activation)       (None, 32, 32, 128)  0           batch_normalization_12[0]
[0]
```

```
__________________________________________________________________________________________________
conv2d_13 (Conv2D)              (None, 32, 32, 128)  147584      activation_12[0][0]
__________________________________________________________________________________________________
batch_normalization_13 (BatchNo (None, 32, 32, 128)  512         conv2d_13[0][0]
__________________________________________________________________________________________________
activation_13 (Activation)      (None, 32, 32, 128)  0           batch_normalization_13[0]
[0]
__________________________________________________________________________________________________
conv2d_transpose_2 (Conv2DTrans (None, 64, 64, 64)   73792       activation_13[0][0]
__________________________________________________________________________________________________
concatenate_2 (Concatenate)     (None, 64, 64, 128)  0           conv2d_transpose_2[0][0]

                                                                 activation_3[0][0]
__________________________________________________________________________________________________
dropout_6 (Dropout)             (None, 64, 64, 128)  0           concatenate_2[0][0]
__________________________________________________________________________________________________
conv2d_14 (Conv2D)              (None, 64, 64, 64)   73792       dropout_6[0][0]
__________________________________________________________________________________________________
batch_normalization_14 (BatchNo (None, 64, 64, 64)   256         conv2d_14[0][0]
__________________________________________________________________________________________________
activation_14 (Activation)      (None, 64, 64, 64)   0           batch_normalization_14[0]
[0]
__________________________________________________________________________________________________
conv2d_15 (Conv2D)              (None, 64, 64, 64)   36928       activation_14[0][0]
__________________________________________________________________________________________________
batch_normalization_15 (BatchNo (None, 64, 64, 64)   256         conv2d_15[0][0]
__________________________________________________________________________________________________
activation_15 (Activation)      (None, 64, 64, 64)   0           batch_normalization_15[0]
[0]
__________________________________________________________________________________________________
conv2d_transpose_3 (Conv2DTrans (None, 128, 128, 32) 18464       activation_15[0][0]
__________________________________________________________________________________________________
concatenate_3 (Concatenate)     (None, 128, 128, 64) 0           conv2d_transpose_3[0][0]

                                                                 activation_1[0][0]
__________________________________________________________________________________________________
dropout_7 (Dropout)             (None, 128, 128, 64) 0           concatenate_3[0][0]
__________________________________________________________________________________________________
```

```
conv2d_16 (Conv2D)              (None, 128, 128, 32) 18464       dropout_7[0][0]

__________________________________________________________________________________________
batch_normalization_16 (BatchNo (None, 128, 128, 32) 128         conv2d_16[0][0]

__________________________________________________________________________________________
activation_16 (Activation)      (None, 128, 128, 32) 0           batch_normalization_16[0]
[0]
__________________________________________________________________________________________
conv2d_17 (Conv2D)              (None, 128, 128, 32) 9248        activation_16[0][0]

__________________________________________________________________________________________
batch_normalization_17 (BatchNo (None, 128, 128, 32) 128         conv2d_17[0][0]

__________________________________________________________________________________________
activation_17 (Activation)      (None, 128, 128, 32) 0           batch_normalization_17[0]
[0]
__________________________________________________________________________________________
conv2d_18 (Conv2D)              (None, 128, 128, 1)  33          activation_17[0][0]

==========================================================================================
Total params: 8,642,273
Trainable params: 8,636,385
Non-trainable params: 5,888
__________________________________________________________________________________________
```

In [11]:

```
def recall_m(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
    recall = true_positives / (possible_positives + K.epsilon())
    return recall

def precision_m(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
    precision = true_positives / (predicted_positives + K.epsilon())
    return precision

def f1_m(y_true, y_pred):
    precision = precision_m(y_true, y_pred)
    recall = recall_m(y_true, y_pred)
    return 2*((precision*recall)/(precision+recall+K.epsilon()))
```

In [12]:

```
model.compile(optimizer = "Adam", loss = "binary_crossentropy", metrics = ['acc',f1_m]) #
callback = EarlyStopping(monitor="loss", patience=5)
```

In [13]:

```
var = model.fit(X_train, y_train, epochs=50, callbacks = callback, validation_data=(X_test
```

```
2022-11-09 12:04:11.891639: W tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocati
on of 491913216 exceeds 10% of free system memory.
2022-11-09 12:04:12.424395: W tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocati
on of 327942144 exceeds 10% of free system memory.
2022-11-09 12:04:12.827627: W tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocati
on of 491913216 exceeds 10% of free system memory.
```

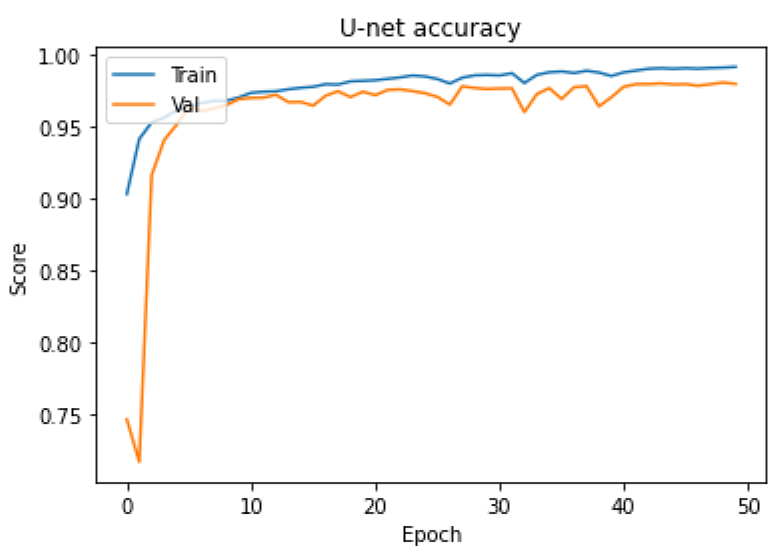
```
2022-11-09 12:04:13.219286: W tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocation of 327942144 exceeds 10% of free system memory.
2022-11-09 12:04:13.522752: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)
Epoch 1/50
2022-11-09 12:04:17.909853: I tensorflow/stream_executor/cuda/cuda_dnn.cc:369] Loaded cuDNN version 8005
79/79 [==============================] - 37s 234ms/step - loss: 0.2396 - acc: 0.9032 - f1_m: 0.9277 - val_loss: 0.9053 - val_acc: 0.7464 - val_f1_m: 0.8364
Epoch 2/50
79/79 [==============================] - 15s 194ms/step - loss: 0.1500 - acc: 0.9415 - f1_m: 0.9568 - val_loss: 1.2142 - val_acc: 0.7168 - val_f1_m: 0.8260
Epoch 3/50
79/79 [==============================] - 15s 195ms/step - loss: 0.1215 - acc: 0.9531 - f1_m: 0.9655 - val_loss: 0.2202 - val_acc: 0.9163 - val_f1_m: 0.9392
Epoch 4/50
79/79 [==============================] - 16s 198ms/step - loss: 0.1140 - acc: 0.9563 - f1_m: 0.9678 - val_loss: 0.1820 - val_acc: 0.9407 - val_f1_m: 0.9567
Epoch 5/50
79/79 [==============================] - 16s 200ms/step - loss: 0.0984 - acc: 0.9615 - f1_m: 0.9717 - val_loss: 0.1435 - val_acc: 0.9513 - val_f1_m: 0.9640
Epoch 6/50
79/79 [==============================] - 16s 203ms/step - loss: 0.0925 - acc: 0.9642 - f1_m: 0.9733 - val_loss: 0.1041 - val_acc: 0.9640 - val_f1_m: 0.9730
Epoch 7/50
79/79 [==============================] - 16s 206ms/step - loss: 0.0866 - acc: 0.9665 - f1_m: 0.9752 - val_loss: 0.1088 - val_acc: 0.9611 - val_f1_m: 0.9710
Epoch 8/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0830 - acc: 0.9681 - f1_m: 0.9764 - val_loss: 0.0980 - val_acc: 0.9628 - val_f1_m: 0.9730
Epoch 9/50
79/79 [==============================] - 16s 207ms/step - loss: 0.0821 - acc: 0.9681 - f1_m: 0.9763 - val_loss: 0.1128 - val_acc: 0.9650 - val_f1_m: 0.9737
Epoch 10/50
79/79 [==============================] - 16s 206ms/step - loss: 0.0762 - acc: 0.9704 - f1_m: 0.9783 - val_loss: 0.0842 - val_acc: 0.9692 - val_f1_m: 0.9768
Epoch 11/50
79/79 [==============================] - 16s 206ms/step - loss: 0.0689 - acc: 0.9737 - f1_m: 0.9807 - val_loss: 0.0819 - val_acc: 0.9701 - val_f1_m: 0.9774
Epoch 12/50
79/79 [==============================] - 16s 207ms/step - loss: 0.0671 - acc: 0.9744 - f1_m: 0.9812 - val_loss: 0.0845 - val_acc: 0.9703 - val_f1_m: 0.9780
Epoch 13/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0654 - acc: 0.9747 - f1_m: 0.9812 - val_loss: 0.0808 - val_acc: 0.9722 - val_f1_m: 0.9793
Epoch 14/50
79/79 [==============================] - 16s 207ms/step - loss: 0.0621 - acc: 0.9761 - f1_m: 0.9824 - val_loss: 0.0965 - val_acc: 0.9671 - val_f1_m: 0.9753
Epoch 15/50
79/79 [==============================] - 16s 207ms/step - loss: 0.0591 - acc: 0.9771 - f1_m: 0.9831 - val_loss: 0.1061 - val_acc: 0.9673 - val_f1_m: 0.9753
Epoch 16/50
79/79 [==============================] - 16s 206ms/step - loss: 0.0577 - acc: 0.9778 - f1_m: 0.9837 - val_loss: 0.0833 - val_acc: 0.9647 - val_f1_m: 0.9743
Epoch 17/50
79/79 [==============================] - 16s 207ms/step - loss: 0.0534 - acc: 0.9797 - f1_m: 0.9850 - val_loss: 0.0803 - val_acc: 0.9716 - val_f1_m: 0.9789
Epoch 18/50
79/79 [==============================] - 16s 206ms/step - loss: 0.0541 - acc: 0.9794 - f1_m: 0.9849 - val_loss: 0.0676 - val_acc: 0.9748 - val_f1_m: 0.9814
Epoch 19/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0482 - acc: 0.9817 - f1_m: 0.9866 - val_loss: 0.0782 - val_acc: 0.9707 - val_f1_m: 0.9783
Epoch 20/50
79/79 [==============================] - 16s 207ms/step - loss: 0.0472 - acc: 0.9820 - f1_
```

```
m: 0.9867 - val_loss: 0.0723 - val_acc: 0.9744 - val_f1_m: 0.9811
Epoch 21/50
79/79 [==============================] - 16s 207ms/step - loss: 0.0456 - acc: 0.9824 - f1_
m: 0.9870 - val_loss: 0.0810 - val_acc: 0.9720 - val_f1_m: 0.9790
Epoch 22/50
79/79 [==============================] - 16s 207ms/step - loss: 0.0431 - acc: 0.9834 - f1_
m: 0.9878 - val_loss: 0.0649 - val_acc: 0.9757 - val_f1_m: 0.9820
Epoch 23/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0408 - acc: 0.9844 - f1_
m: 0.9885 - val_loss: 0.0651 - val_acc: 0.9761 - val_f1_m: 0.9823
Epoch 24/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0371 - acc: 0.9857 - f1_
m: 0.9895 - val_loss: 0.0796 - val_acc: 0.9748 - val_f1_m: 0.9812
Epoch 25/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0387 - acc: 0.9851 - f1_
m: 0.9890 - val_loss: 0.0773 - val_acc: 0.9734 - val_f1_m: 0.9804
Epoch 26/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0433 - acc: 0.9833 - f1_
m: 0.9876 - val_loss: 0.0975 - val_acc: 0.9708 - val_f1_m: 0.9780
Epoch 27/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0515 - acc: 0.9801 - f1_
m: 0.9853 - val_loss: 0.1519 - val_acc: 0.9654 - val_f1_m: 0.9739
Epoch 28/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0412 - acc: 0.9842 - f1_
m: 0.9884 - val_loss: 0.0671 - val_acc: 0.9782 - val_f1_m: 0.9837
Epoch 29/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0367 - acc: 0.9858 - f1_
m: 0.9896 - val_loss: 0.0670 - val_acc: 0.9770 - val_f1_m: 0.9829
Epoch 30/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0360 - acc: 0.9861 - f1_
m: 0.9897 - val_loss: 0.0691 - val_acc: 0.9763 - val_f1_m: 0.9823
Epoch 31/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0365 - acc: 0.9857 - f1_
m: 0.9895 - val_loss: 0.0673 - val_acc: 0.9766 - val_f1_m: 0.9827
Epoch 32/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0328 - acc: 0.9873 - f1_
m: 0.9907 - val_loss: 0.0683 - val_acc: 0.9768 - val_f1_m: 0.9828
Epoch 33/50
79/79 [==============================] - 16s 207ms/step - loss: 0.0522 - acc: 0.9805 - f1_
m: 0.9856 - val_loss: 0.1832 - val_acc: 0.9602 - val_f1_m: 0.9688
Epoch 34/50
79/79 [==============================] - 16s 207ms/step - loss: 0.0359 - acc: 0.9861 - f1_
m: 0.9897 - val_loss: 0.0884 - val_acc: 0.9728 - val_f1_m: 0.9800
Epoch 35/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0312 - acc: 0.9880 - f1_
m: 0.9912 - val_loss: 0.0663 - val_acc: 0.9769 - val_f1_m: 0.9828
Epoch 36/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0294 - acc: 0.9885 - f1_
m: 0.9915 - val_loss: 0.0876 - val_acc: 0.9694 - val_f1_m: 0.9771
Epoch 37/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0324 - acc: 0.9874 - f1_
m: 0.9908 - val_loss: 0.0645 - val_acc: 0.9775 - val_f1_m: 0.9833
Epoch 38/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0282 - acc: 0.9890 - f1_
m: 0.9918 - val_loss: 0.0646 - val_acc: 0.9783 - val_f1_m: 0.9838
Epoch 39/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0311 - acc: 0.9878 - f1_
m: 0.9910 - val_loss: 0.1280 - val_acc: 0.9642 - val_f1_m: 0.9739
Epoch 40/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0381 - acc: 0.9854 - f1_
m: 0.9892 - val_loss: 0.1251 - val_acc: 0.9705 - val_f1_m: 0.9778
Epoch 41/50
79/79 [==============================] - 16s 209ms/step - loss: 0.0310 - acc: 0.9879 - f1_
m: 0.9910 - val_loss: 0.0700 - val_acc: 0.9779 - val_f1_m: 0.9835
Epoch 42/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0278 - acc: 0.9891 - f1_
```

```
m: 0.9920 - val_loss: 0.0667 - val_acc: 0.9797 - val_f1_m: 0.9847
Epoch 43/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0243 - acc: 0.9904 - f1_
m: 0.9929 - val_loss: 0.0655 - val_acc: 0.9797 - val_f1_m: 0.9848
Epoch 44/50
79/79 [==============================] - 16s 207ms/step - loss: 0.0235 - acc: 0.9908 - f1_
m: 0.9932 - val_loss: 0.0659 - val_acc: 0.9802 - val_f1_m: 0.9852
Epoch 45/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0244 - acc: 0.9904 - f1_
m: 0.9929 - val_loss: 0.0701 - val_acc: 0.9795 - val_f1_m: 0.9847
Epoch 46/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0233 - acc: 0.9908 - f1_
m: 0.9931 - val_loss: 0.0676 - val_acc: 0.9797 - val_f1_m: 0.9848
Epoch 47/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0240 - acc: 0.9905 - f1_
m: 0.9930 - val_loss: 0.0863 - val_acc: 0.9786 - val_f1_m: 0.9839
Epoch 48/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0228 - acc: 0.9910 - f1_
m: 0.9933 - val_loss: 0.0698 - val_acc: 0.9797 - val_f1_m: 0.9849
Epoch 49/50
79/79 [==============================] - 16s 207ms/step - loss: 0.0222 - acc: 0.9912 - f1_
m: 0.9935 - val_loss: 0.0669 - val_acc: 0.9808 - val_f1_m: 0.9857
Epoch 50/50
79/79 [==============================] - 16s 208ms/step - loss: 0.0209 - acc: 0.9917 - f1_
m: 0.9939 - val_loss: 0.0715 - val_acc: 0.9799 - val_f1_m: 0.9849
```

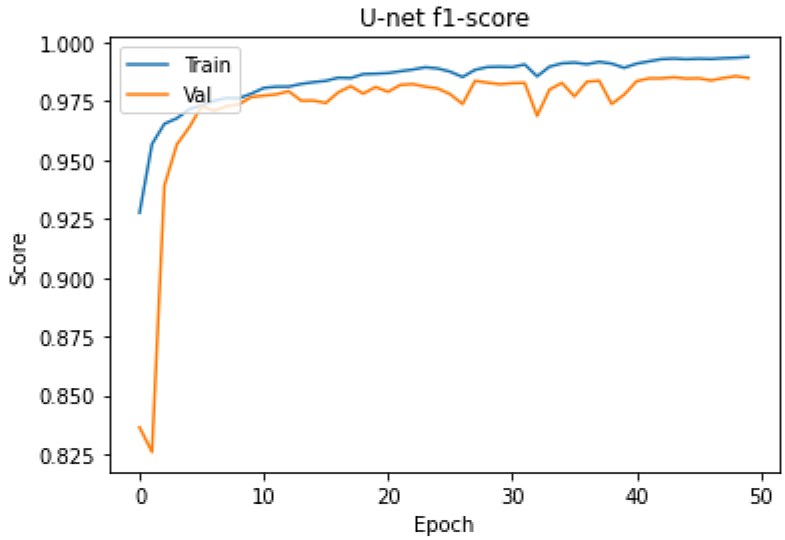
In [14]:

```
# Plot of the accuracy of the U-net model over each epoch

plt.plot(var.history['acc'])
plt.plot(var.history['val_acc'])
plt.title('U-net accuracy')
plt.ylabel('Score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper left')
plt.show()
```



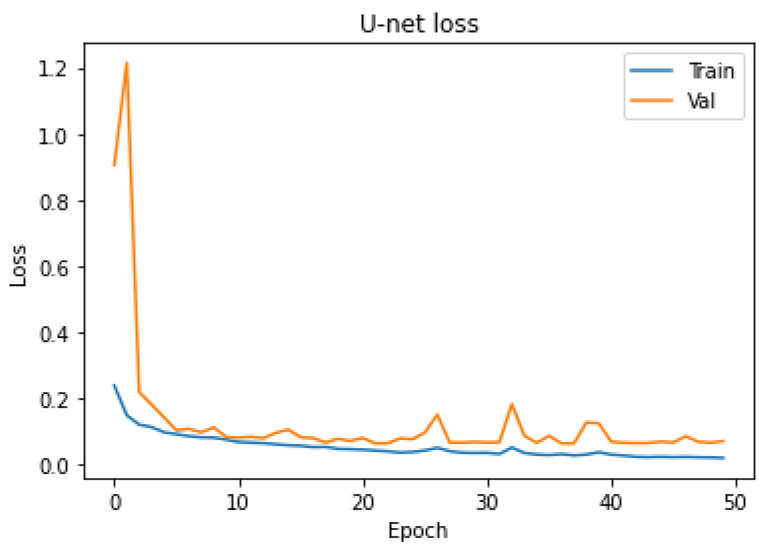
In [15]:

```
# Plot of the f1-score of the U-net model over each epoch

plt.plot(var.history['f1_m'])
plt.plot(var.history['val_f1_m'])
plt.title('U-net f1-score')
plt.ylabel('Score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper left')
plt.show()
```



In [16]:

```
#Plot of the loss of the U-net model over each epoch

plt.plot(var.history['loss'])
plt.plot(var.history['val_loss'])
plt.title('U-net loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper right')
plt.show()
```



One can see that the training accuracy and f1-score is a small bit higher than the validation score. This means that the manual U-net model overfits a tiny bit. Furthermore, one can see that the loss gets smaller for each epoch.

In [17]:

```
# Code and model training for your best transfer learning model (point 4 in Canvas)

def get_unet_vg16(input_img, n_filters = 16, dropout = 0.1, batchnorm = True, n_classes =

    encode_model = vg(input_tensor=input_img, include_top = False, weights="imagenet")

    for layer in encode_model.layers:
        layer.trainable = False

    encoder_output = encode_model.get_layer("block5_conv3").output

    # Expansive Path
    u6 = Conv2DTranspose(n_filters * 8, (3, 3), strides = (2, 2), padding = 'same')(encode
```

```
    u6 = concatenate([u6, encode_model.get_layer("block4_conv3").output])
    u6 = Dropout(dropout)(u6)
    c6 = conv2d_block(u6, n_filters * 8, kernel_size = 3, batchnorm = batchnorm)

    u7 = Conv2DTranspose(n_filters * 4, (3, 3), strides = (2, 2), padding = 'same')(c6)
    u7 = concatenate([u7, encode_model.get_layer("block3_conv3").output])
    u7 = Dropout(dropout)(u7)
    c7 = conv2d_block(u7, n_filters * 4, kernel_size = 3, batchnorm = batchnorm)

    u8 = Conv2DTranspose(n_filters * 2, (3, 3), strides = (2, 2), padding = 'same')(c7)
    u8 = concatenate([u8, encode_model.get_layer("block2_conv2").output])
    u8 = Dropout(dropout)(u8)
    c8 = conv2d_block(u8, n_filters * 2, kernel_size = 3, batchnorm = batchnorm)

    u9 = Conv2DTranspose(n_filters * 1, (3, 3), strides = (2, 2), padding = 'same')(c8)
    u9 = concatenate([u9, encode_model.get_layer("block1_conv2").output])
    u9 = Dropout(dropout)(u9)
    c9 = conv2d_block(u9, n_filters * 1, kernel_size = 3, batchnorm = batchnorm)

    outputs_vg = Conv2D(n_classes, (1, 1), padding = 'same', activation='sigmoid')(c9)
    vg_model = Model(inputs=[encode_model.input], outputs=[outputs_vg])
    return vg_model
```

In [18]:

```
input_img = Input(shape=(128,128,3))
model_vg = get_unet_vg16(input_img, n_filters = 64, dropout = 0.2, batchnorm = True, n_cla
model_vg.summary()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/v
gg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58892288/58889256 [==============================] - 0s 0us/step
58900480/58889256 [==============================] - 0s 0us/step
Model: "model_1"
__________________________________________________________________________________________
________
Layer (type)                    Output Shape         Param #     Connected to

==========================================================================================
========
input_2 (InputLayer)            [(None, 128, 128, 3) 0


__________________________________________________________________________________________
________
block1_conv1 (Conv2D)           (None, 128, 128, 64) 1792        input_2[0][0]


__________________________________________________________________________________________
________
block1_conv2 (Conv2D)           (None, 128, 128, 64) 36928       block1_conv1[0][0]


__________________________________________________________________________________________
________
block1_pool (MaxPooling2D)      (None, 64, 64, 64)   0           block1_conv2[0][0]


__________________________________________________________________________________________
________
block2_conv1 (Conv2D)           (None, 64, 64, 128)  73856       block1_pool[0][0]


__________________________________________________________________________________________
________
block2_conv2 (Conv2D)           (None, 64, 64, 128)  147584      block2_conv1[0][0]


__________________________________________________________________________________________
________
block2_pool (MaxPooling2D)      (None, 32, 32, 128)  0           block2_conv2[0][0]
```

```
__________________________________________________________________________________________________
block3_conv1 (Conv2D)           (None, 32, 32, 256)  295168      block2_pool[0][0]
__________________________________________________________________________________________________
block3_conv2 (Conv2D)           (None, 32, 32, 256)  590080      block3_conv1[0][0]
__________________________________________________________________________________________________
block3_conv3 (Conv2D)           (None, 32, 32, 256)  590080      block3_conv2[0][0]
__________________________________________________________________________________________________
block3_pool (MaxPooling2D)      (None, 16, 16, 256)  0           block3_conv3[0][0]
__________________________________________________________________________________________________
block4_conv1 (Conv2D)           (None, 16, 16, 512)  1180160     block3_pool[0][0]
__________________________________________________________________________________________________
block4_conv2 (Conv2D)           (None, 16, 16, 512)  2359808     block4_conv1[0][0]
__________________________________________________________________________________________________
block4_conv3 (Conv2D)           (None, 16, 16, 512)  2359808     block4_conv2[0][0]
__________________________________________________________________________________________________
block4_pool (MaxPooling2D)      (None, 8, 8, 512)    0           block4_conv3[0][0]
__________________________________________________________________________________________________
block5_conv1 (Conv2D)           (None, 8, 8, 512)    2359808     block4_pool[0][0]
__________________________________________________________________________________________________
block5_conv2 (Conv2D)           (None, 8, 8, 512)    2359808     block5_conv1[0][0]
__________________________________________________________________________________________________
block5_conv3 (Conv2D)           (None, 8, 8, 512)    2359808     block5_conv2[0][0]
__________________________________________________________________________________________________
conv2d_transpose_4 (Conv2DTrans (None, 16, 16, 512)  2359808     block5_conv3[0][0]
__________________________________________________________________________________________________
concatenate_4 (Concatenate)     (None, 16, 16, 1024) 0           conv2d_transpose_4[0][0]

                                                                 block4_conv3[0][0]
__________________________________________________________________________________________________
dropout_8 (Dropout)             (None, 16, 16, 1024) 0           concatenate_4[0][0]
__________________________________________________________________________________________________
conv2d_19 (Conv2D)              (None, 16, 16, 512)  4719104     dropout_8[0][0]
__________________________________________________________________________________________________
batch_normalization_18 (BatchNo (None, 16, 16, 512)  2048        conv2d_19[0][0]
```

```
__________________________________________________________________________________________________
________
activation_18 (Activation)      (None, 16, 16, 512)  0           batch_normalization_18[0]
[0]
__________________________________________________________________________________________________
________
conv2d_20 (Conv2D)              (None, 16, 16, 512)  2359808     activation_18[0][0]

__________________________________________________________________________________________________
________
batch_normalization_19 (BatchNo (None, 16, 16, 512)  2048        conv2d_20[0][0]

__________________________________________________________________________________________________
________
activation_19 (Activation)      (None, 16, 16, 512)  0           batch_normalization_19[0]
[0]
__________________________________________________________________________________________________
________
conv2d_transpose_5 (Conv2DTrans (None, 32, 32, 256)  1179904     activation_19[0][0]

__________________________________________________________________________________________________
________
concatenate_5 (Concatenate)     (None, 32, 32, 512)  0           conv2d_transpose_5[0][0]

                                                                 block3_conv3[0][0]

__________________________________________________________________________________________________
________
dropout_9 (Dropout)             (None, 32, 32, 512)  0           concatenate_5[0][0]

__________________________________________________________________________________________________
________
conv2d_21 (Conv2D)              (None, 32, 32, 256)  1179904     dropout_9[0][0]

__________________________________________________________________________________________________
________
batch_normalization_20 (BatchNo (None, 32, 32, 256)  1024        conv2d_21[0][0]

__________________________________________________________________________________________________
________
activation_20 (Activation)      (None, 32, 32, 256)  0           batch_normalization_20[0]
[0]
__________________________________________________________________________________________________
________
conv2d_22 (Conv2D)              (None, 32, 32, 256)  590080      activation_20[0][0]

__________________________________________________________________________________________________
________
batch_normalization_21 (BatchNo (None, 32, 32, 256)  1024        conv2d_22[0][0]

__________________________________________________________________________________________________
________
activation_21 (Activation)      (None, 32, 32, 256)  0           batch_normalization_21[0]
[0]
__________________________________________________________________________________________________
________
conv2d_transpose_6 (Conv2DTrans (None, 64, 64, 128)  295040      activation_21[0][0]

__________________________________________________________________________________________________
________
concatenate_6 (Concatenate)     (None, 64, 64, 256)  0           conv2d_transpose_6[0][0]

                                                                 block2_conv2[0][0]

__________________________________________________________________________________________________
________
```

```
dropout_10 (Dropout)            (None, 64, 64, 256)  0           concatenate_6[0][0]
__________________________________________________________________________________________________
conv2d_23 (Conv2D)              (None, 64, 64, 128)  295040      dropout_10[0][0]
__________________________________________________________________________________________________
batch_normalization_22 (BatchNo (None, 64, 64, 128)  512         conv2d_23[0][0]
__________________________________________________________________________________________________
activation_22 (Activation)      (None, 64, 64, 128)  0           batch_normalization_22[0]
[0]
__________________________________________________________________________________________________
conv2d_24 (Conv2D)              (None, 64, 64, 128)  147584      activation_22[0][0]
__________________________________________________________________________________________________
batch_normalization_23 (BatchNo (None, 64, 64, 128)  512         conv2d_24[0][0]
__________________________________________________________________________________________________
activation_23 (Activation)      (None, 64, 64, 128)  0           batch_normalization_23[0]
[0]
__________________________________________________________________________________________________
conv2d_transpose_7 (Conv2DTrans (None, 128, 128, 64) 73792       activation_23[0][0]
__________________________________________________________________________________________________
concatenate_7 (Concatenate)     (None, 128, 128, 128 0           conv2d_transpose_7[0][0]

                                                                 block1_conv2[0][0]
__________________________________________________________________________________________________
dropout_11 (Dropout)            (None, 128, 128, 128 0           concatenate_7[0][0]
__________________________________________________________________________________________________
conv2d_25 (Conv2D)              (None, 128, 128, 64) 73792       dropout_11[0][0]
__________________________________________________________________________________________________
batch_normalization_24 (BatchNo (None, 128, 128, 64) 256         conv2d_25[0][0]
__________________________________________________________________________________________________
activation_24 (Activation)      (None, 128, 128, 64) 0           batch_normalization_24[0]
[0]
__________________________________________________________________________________________________
conv2d_26 (Conv2D)              (None, 128, 128, 64) 36928       activation_24[0][0]
__________________________________________________________________________________________________
batch_normalization_25 (BatchNo (None, 128, 128, 64) 256         conv2d_26[0][0]
__________________________________________________________________________________________________
activation_25 (Activation)      (None, 128, 128, 64) 0           batch_normalization_25[0]
[0]
__________________________________________________________________________________________________
```

```
conv2d_27 (Conv2D)              (None, 128, 128, 1)  65          activation_25[0][0]

==========================================================================================
Total params: 28,033,217
Trainable params: 13,314,689
Non-trainable params: 14,718,528
__________________________________________________________________________________________
```

In [19]:

```python
inputs = Input((128,128,3))
model_vg.compile(optimizer = Adam(learning_rate = 0.001), loss = "binary_crossentropy", me
# Fit data to model

history = model_vg.fit(X_train, y_train,
                        epochs=50,
                        batch_size=42,
                       # shuffle=True,
                        validation_data=(X_test, y_test),
                        callbacks= callback
             )
```

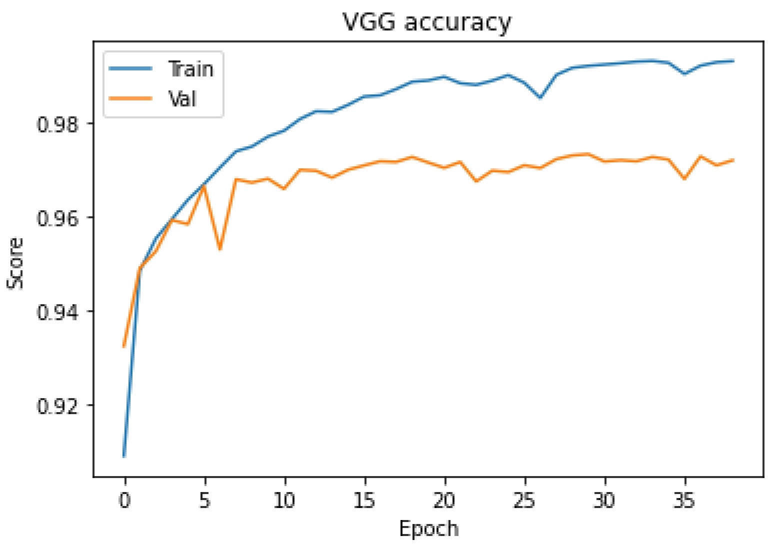
```
2022-11-09 12:18:12.148071: W tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocati
on of 491913216 exceeds 10% of free system memory.
Epoch 1/50
60/60 [==============================] - 57s 732ms/step - loss: 0.2168 - acc: 0.9089 - f1_
m: 0.9308 - val_loss: 1.1796 - val_acc: 0.9323 - val_f1_m: 0.9485
Epoch 2/50
60/60 [==============================] - 32s 528ms/step - loss: 0.1299 - acc: 0.9483 - f1_
m: 0.9617 - val_loss: 0.2013 - val_acc: 0.9489 - val_f1_m: 0.9615
Epoch 3/50
60/60 [==============================] - 32s 530ms/step - loss: 0.1127 - acc: 0.9552 - f1_
m: 0.9669 - val_loss: 0.1433 - val_acc: 0.9525 - val_f1_m: 0.9641
Epoch 4/50
60/60 [==============================] - 32s 531ms/step - loss: 0.1032 - acc: 0.9593 - f1_
m: 0.9700 - val_loss: 0.1274 - val_acc: 0.9591 - val_f1_m: 0.9695
Epoch 5/50
60/60 [==============================] - 32s 530ms/step - loss: 0.0925 - acc: 0.9634 - f1_
m: 0.9729 - val_loss: 0.1229 - val_acc: 0.9582 - val_f1_m: 0.9692
Epoch 6/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0836 - acc: 0.9667 - f1_
m: 0.9755 - val_loss: 0.0883 - val_acc: 0.9664 - val_f1_m: 0.9751
Epoch 7/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0746 - acc: 0.9703 - f1_
m: 0.9781 - val_loss: 0.1229 - val_acc: 0.9529 - val_f1_m: 0.9656
Epoch 8/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0666 - acc: 0.9737 - f1_
m: 0.9806 - val_loss: 0.0846 - val_acc: 0.9678 - val_f1_m: 0.9761
Epoch 9/50
60/60 [==============================] - 32s 528ms/step - loss: 0.0643 - acc: 0.9748 - f1_
m: 0.9814 - val_loss: 0.0906 - val_acc: 0.9671 - val_f1_m: 0.9757
Epoch 10/50
60/60 [==============================] - 32s 528ms/step - loss: 0.0585 - acc: 0.9769 - f1_
m: 0.9829 - val_loss: 0.0961 - val_acc: 0.9679 - val_f1_m: 0.9760
Epoch 11/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0553 - acc: 0.9781 - f1_
m: 0.9839 - val_loss: 0.1050 - val_acc: 0.9657 - val_f1_m: 0.9741
Epoch 12/50
60/60 [==============================] - 32s 530ms/step - loss: 0.0498 - acc: 0.9806 - f1_
m: 0.9857 - val_loss: 0.0908 - val_acc: 0.9698 - val_f1_m: 0.9774
Epoch 13/50
60/60 [==============================] - 32s 532ms/step - loss: 0.0456 - acc: 0.9823 - f1_
m: 0.9869 - val_loss: 0.0857 - val_acc: 0.9696 - val_f1_m: 0.9774
Epoch 14/50
60/60 [==============================] - 32s 530ms/step - loss: 0.0459 - acc: 0.9821 - f1_
```

```
m: 0.9868 - val_loss: 0.0990 - val_acc: 0.9681 - val_f1_m: 0.9763
Epoch 15/50
60/60 [==============================] - 32s 531ms/step - loss: 0.0419 - acc: 0.9836 - f1_
m: 0.9879 - val_loss: 0.0892 - val_acc: 0.9698 - val_f1_m: 0.9775
Epoch 16/50
60/60 [==============================] - 32s 531ms/step - loss: 0.0374 - acc: 0.9854 - f1_
m: 0.9892 - val_loss: 0.0842 - val_acc: 0.9707 - val_f1_m: 0.9784
Epoch 17/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0367 - acc: 0.9857 - f1_
m: 0.9894 - val_loss: 0.0863 - val_acc: 0.9716 - val_f1_m: 0.9790
Epoch 18/50
60/60 [==============================] - 32s 530ms/step - loss: 0.0332 - acc: 0.9870 - f1_
m: 0.9904 - val_loss: 0.0877 - val_acc: 0.9715 - val_f1_m: 0.9787
Epoch 19/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0291 - acc: 0.9885 - f1_
m: 0.9915 - val_loss: 0.0815 - val_acc: 0.9725 - val_f1_m: 0.9796
Epoch 20/50
60/60 [==============================] - 32s 528ms/step - loss: 0.0285 - acc: 0.9888 - f1_
m: 0.9918 - val_loss: 0.0968 - val_acc: 0.9714 - val_f1_m: 0.9787
Epoch 21/50
60/60 [==============================] - 32s 528ms/step - loss: 0.0264 - acc: 0.9896 - f1_
m: 0.9923 - val_loss: 0.1042 - val_acc: 0.9702 - val_f1_m: 0.9779
Epoch 22/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0300 - acc: 0.9882 - f1_
m: 0.9913 - val_loss: 0.1100 - val_acc: 0.9715 - val_f1_m: 0.9788
Epoch 23/50
60/60 [==============================] - 32s 528ms/step - loss: 0.0306 - acc: 0.9879 - f1_
m: 0.9911 - val_loss: 0.1187 - val_acc: 0.9673 - val_f1_m: 0.9757
Epoch 24/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0284 - acc: 0.9888 - f1_
m: 0.9917 - val_loss: 0.1055 - val_acc: 0.9696 - val_f1_m: 0.9774
Epoch 25/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0253 - acc: 0.9900 - f1_
m: 0.9926 - val_loss: 0.1012 - val_acc: 0.9693 - val_f1_m: 0.9773
Epoch 26/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0298 - acc: 0.9883 - f1_
m: 0.9914 - val_loss: 0.1057 - val_acc: 0.9708 - val_f1_m: 0.9783
Epoch 27/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0376 - acc: 0.9851 - f1_
m: 0.9890 - val_loss: 0.1142 - val_acc: 0.9702 - val_f1_m: 0.9777
Epoch 28/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0251 - acc: 0.9900 - f1_
m: 0.9926 - val_loss: 0.0974 - val_acc: 0.9721 - val_f1_m: 0.9792
Epoch 29/50
60/60 [==============================] - 32s 528ms/step - loss: 0.0213 - acc: 0.9915 - f1_
m: 0.9937 - val_loss: 0.0930 - val_acc: 0.9729 - val_f1_m: 0.9798
Epoch 30/50
60/60 [==============================] - 32s 528ms/step - loss: 0.0201 - acc: 0.9919 - f1_
m: 0.9940 - val_loss: 0.0983 - val_acc: 0.9731 - val_f1_m: 0.9800
Epoch 31/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0193 - acc: 0.9922 - f1_
m: 0.9942 - val_loss: 0.1072 - val_acc: 0.9716 - val_f1_m: 0.9789
Epoch 32/50
60/60 [==============================] - 32s 530ms/step - loss: 0.0186 - acc: 0.9925 - f1_
m: 0.9945 - val_loss: 0.1109 - val_acc: 0.9719 - val_f1_m: 0.9790
Epoch 33/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0177 - acc: 0.9928 - f1_
m: 0.9947 - val_loss: 0.1053 - val_acc: 0.9716 - val_f1_m: 0.9790
Epoch 34/50
60/60 [==============================] - 32s 531ms/step - loss: 0.0173 - acc: 0.9930 - f1_
m: 0.9948 - val_loss: 0.1088 - val_acc: 0.9725 - val_f1_m: 0.9796
Epoch 35/50
60/60 [==============================] - 32s 530ms/step - loss: 0.0184 - acc: 0.9926 - f1_
m: 0.9945 - val_loss: 0.1357 - val_acc: 0.9720 - val_f1_m: 0.9794
Epoch 36/50
60/60 [==============================] - 32s 532ms/step - loss: 0.0247 - acc: 0.9902 - f1_
```

```
m: 0.9928 - val_loss: 0.1557 - val_acc: 0.9679 - val_f1_m: 0.9763
Epoch 37/50
60/60 [==============================] - 32s 530ms/step - loss: 0.0199 - acc: 0.9920 - f1_
m: 0.9941 - val_loss: 0.1132 - val_acc: 0.9727 - val_f1_m: 0.9797
Epoch 38/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0181 - acc: 0.9927 - f1_
m: 0.9946 - val_loss: 0.1253 - val_acc: 0.9708 - val_f1_m: 0.9785
Epoch 39/50
60/60 [==============================] - 32s 529ms/step - loss: 0.0175 - acc: 0.9929 - f1_
m: 0.9948 - val_loss: 0.1184 - val_acc: 0.9718 - val_f1_m: 0.9791
```

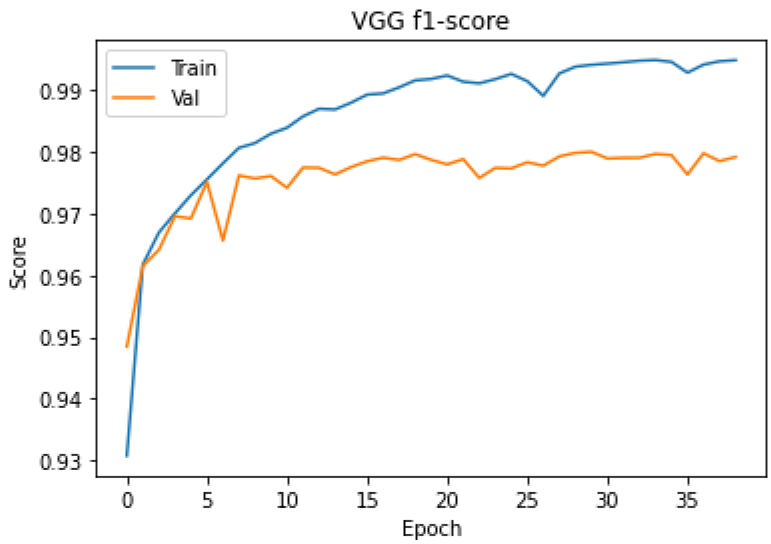
In [20]:

```python
# Plot of the accuracy of the VGG model over each epoch

plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('VGG accuracy')
plt.ylabel('Score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper left')
plt.show()
```

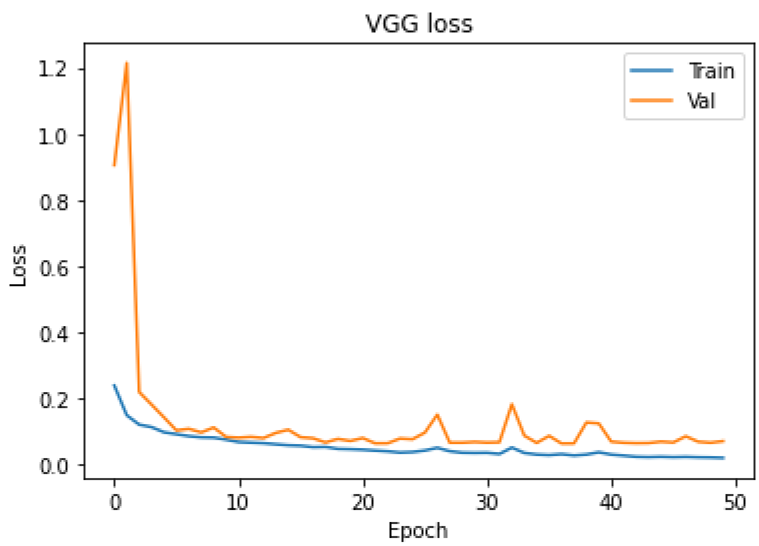


In [21]:

```python
# Plot of the f1-score of the VGG model over each epoch

plt.plot(history.history['f1_m'])
plt.plot(history.history['val_f1_m'])
plt.title('VGG f1-score')
plt.ylabel('Score')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper left')
plt.show()
```



In [22]:
```
#Plot of the loss of the VGG model over each epoch

plt.plot(var.history['loss'])
plt.plot(var.history['val_loss'])
plt.title('VGG loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper right')
plt.show()
```



One can see that the VGG model overfits much more than the manual U-net model

Orion related code, i.e. slurm-script, username, code to access data on Orion and time usage for your modelling (point 5)

Slurm script: singularity exec --nv --bind /mnt/courses/DAT300-22:/mnt/courses/DAT300-22 $SIFFILE python Unet-CA2-Orion.py

From what we see, Orion was a little bit faster at training the models and predicting

In [23]:
```
# Optional: Code and model training for multiclass segmentation with U-net (Point 5 in Ca
```

Describe the results you are observing.

In [24]:
```
unet_predictions = model.predict(X_test_final) # U-net predicting
```

```
pred_1 = unet_predictions.flatten()
pred_1

vg_predictions = model_vg.predict(X_test_final) # VGG16 predicting
pred_2 = vg_predictions.flatten()
pred_2
```

Out[24]:
```
array([0.9989022 , 0.9969715 , 0.9989262 , ..., 0.9999335 , 0.99993956,
       0.9996289 ], dtype=float32)
```

In [ ]:
```
submission_df = pd.DataFrame(data=list(range(len(pred))),
                             columns=["Id"])

submission_df["Predicted"] = pred
submission_df = submission_df.round(0).astype("int")

submission_df['Predicted'] = np.where(submission_df['Predicted'] == 0,
                                      False, True)

submission_df.to_csv("CA2_goofy_submission.csv", index=False)
submission_df
```

Discussion / conclusion

When we started modelling, we didn't change the output activation from softmax to sigmoid, which gave us an accuracy of around 60%. After changing this to sigmoid, our accuracy got much better on around 80%, but didn't really excel around 98% until we changed our preprocessing method. By using sklearn's train-test-split, we managed to get above beat me. We also had some problems with Orion, because it did not have sklearn and had a steep learning curve, because of the linux commands.

Our best model was the manual U-net, which was a little bit better than VGG, since it didn't overfit as much. Anyway, when running the models in Orion, each epoch for both of the models took less time to run, but the overall accuracy stayed the same. Given that Orion did not work for us for quite a long time, this increase in speed did not help us much.

In other words, we did not have a pleasant experience with Orion. Anyway, given more time we would commit to complete the optional task of multiclass segmentation. We did manage to make the function work, but we did not understand how to fit that to the model. We have therefore concluded with not including it in this compulsary assignment.