Road Finder

Initial Config

```
from google.colab import drive
drive.mount('/content/gdrive')
base_path = 'gdrive/My\ Drive/road_finder_data/'
%cd gdrive/My\ Drive/road_finder_data/
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive
[Errno 2] No such file or directory: 'gdrive/My Drive/road_finder_data/'
/content/gdrive/My Drive/road_finder_data
```

```
import cv2
import h5py
import random
import numpy as np
import tensorflow as tf
import keras
import time
from matplotlib import pyplot as plt
from keras import backend as K
from keras.models import Model, load_model
from keras.metrics import MeanIoU
from keras.layers import Input
from keras.layers.core import Lambda
from keras.layers.convolutional import Conv2D, Conv2DTranspose
from keras.layers.pooling import MaxPooling2D
from keras.layers.merge import concatenate
from keras import optimizers
from keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau
```

Pathes to datasets

```
train_file = './big/train.hdf5'
```

Constants

```
ACT_FUNCTION = 'relu'
KERNEL_INIT = 'he_normal'
PADDING_TYPE = 'same'
EPOCHS = 100
LEARNING_RATE = 0.0001
BATCH_SIZE = 16
CHECKPOINT_MODEL_PATH = "./Models/road_mapper_2.h5"
```

```
FINAL_MODEL_PATH = "./Models/road_mapper_final_relu_1000"
```

Loss function

```
def soft_dice_loss(y_true, y_pred, smooth = 1):
  y_true_f = K.flatten(y_true)
  y_pred_f = K.flatten(y_pred)
  intersection = K.sum(y_true_f * y_pred_f)
  return 1 - (2. * intersection + smooth) / (K.sum(y_true_f) + K.sum(y_pred_f) + smooth)
```

Intersection over Union

```
def iou_coef(y_true, y_pred, smooth = 1):
  I = K.sum(K.abs(y_true * y_pred), axis=[1,2,3])
  U = K.sum(y_true,[1,2,3]) + K.sum(y_pred, [1,2,3]) - I
  iou = K.mean((I + smooth) / (U + smooth), axis=0)
  return iou
```

Load Train images

```
hfile = h5py.File(train_file, 'r')

train_images = np.array(hfile.get('images'))[:1000]
print(train_images.shape)

train_masks = np.array(hfile.get('masks'))[:1000]
print(train_masks.shape)

hfile.close()
```

```
(1000, 256, 256, 3)
(1000, 256, 256)
```

View samples

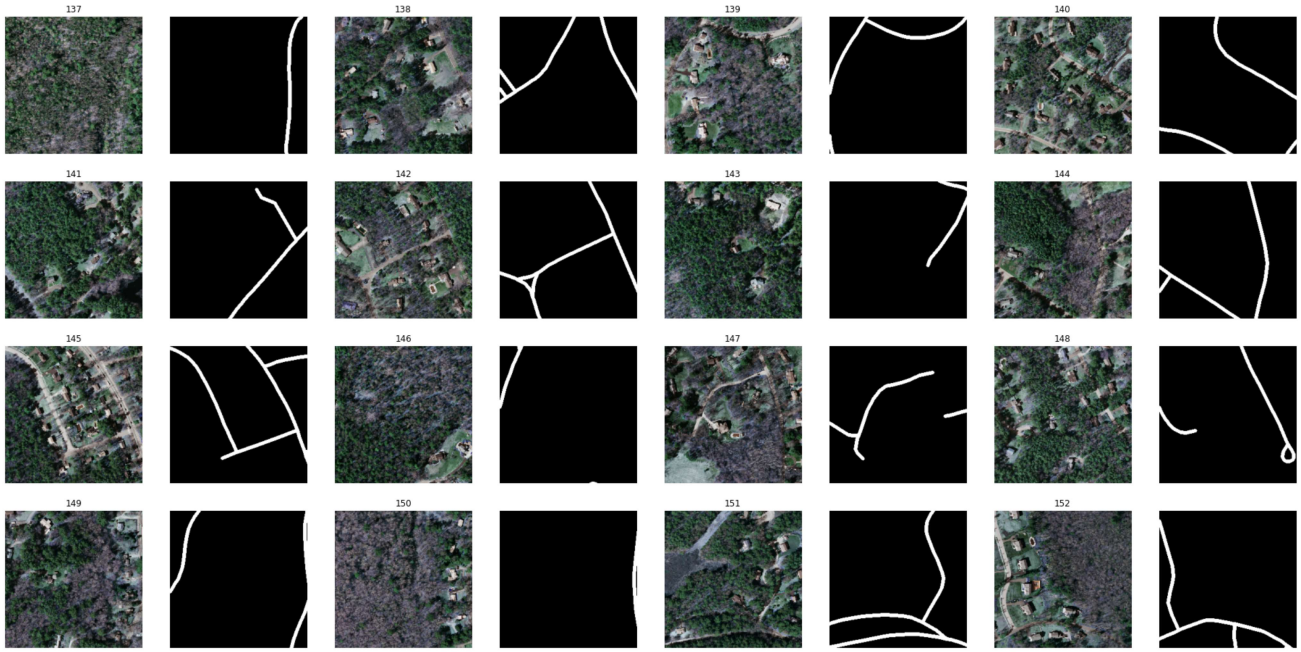
```
random.seed(1)
plt.figure(figsize=(32,16))
x, y = 4, 4
ipos = random.randint(0, len(train_images)-x*y)
for i in range(y):
  for j in range(x):

    pos = ipos + i*x + j

    plt.subplot(y, x*2, i*x*2+j*2+1)
    plt.imshow(train_images[pos])
```

```
        plt.title(pos)
        plt.axis('off')

        plt.subplot(y, x*2, i*x*2+(j*2)+2)
        plt.imshow(train_masks[pos], cmap='gray', vmin=0, vmax=255)
        plt.axis('off')
    plt.show()
```



Create Model

```
inputs = Input((256, 256, 3))
s = Lambda(lambda x: x / 255) (inputs)

conv1 = Conv2D(16, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PAD
conv1 = Conv2D(16, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PAD
pooling1 = MaxPooling2D() (conv1)

conv2 = Conv2D(32, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PAD
conv2 = Conv2D(32, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PAD
pooling2 = MaxPooling2D() (conv2)

conv3 = Conv2D(64, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PAD
conv3 = Conv2D(64, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PAD
pooling3 = MaxPooling2D() (conv3)

conv4 = Conv2D(128, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PA
conv4 = Conv2D(128, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PA
pooling4 = MaxPooling2D() (conv4)
```

```
conv5 = Conv2D(256, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PA
conv5 = Conv2D(256, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PA

upsample6 = Conv2DTranspose(128, 2, strides=(2,2), padding=PADDING_TYPE) (conv5)
upsample6 = concatenate([upsample6, conv4])
conv6 = Conv2D(128, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PA
conv6 = Conv2D(128, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PA

upsample7 = Conv2DTranspose(64, 2, strides=(2, 2), padding=PADDING_TYPE) (conv6)
upsample7 = concatenate([upsample7, conv3])
conv7 = Conv2D(64, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PAD
conv7 = Conv2D(64, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PAD

upsample8 = Conv2DTranspose(32, 2, strides=(2, 2), padding=PADDING_TYPE) (conv7)
upsample8 = concatenate([upsample8, conv2])
conv8 = Conv2D(32, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PAD
conv8 = Conv2D(32, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PAD

upsample9 = Conv2DTranspose(16, 2, strides=(2, 2), padding=PADDING_TYPE) (conv8)
upsample9 = concatenate([upsample9, conv1], axis=3)
conv9 = Conv2D(16, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PAD
conv9 = Conv2D(16, 3, activation=ACT_FUNCTION, kernel_initializer=KERNEL_INIT, padding=PAD

outputs = Conv2D(1, 1, activation='sigmoid') (conv9)

model = Model(inputs=[inputs], outputs=[outputs])
model.summary()
```

```
Model: "model_2"
_________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================
input_3 (InputLayer)         [(None, 256, 256, 3) 0
_________________________________________________________________________________
conv2d_38 (Conv2D)           (None, 256, 256, 16) 448         input_3[0][0]
_________________________________________________________________________________
conv2d_39 (Conv2D)           (None, 256, 256, 16) 2320        conv2d_38[0][0]
_________________________________________________________________________________
max_pooling2d_8 (MaxPooling2D)  (None, 128, 128, 16) 0        conv2d_39[0][0]
_________________________________________________________________________________
conv2d_40 (Conv2D)           (None, 128, 128, 32) 4640        max_pooling2d_8[0
_________________________________________________________________________________
conv2d_41 (Conv2D)           (None, 128, 128, 32) 9248        conv2d_40[0][0]
_________________________________________________________________________________
max_pooling2d_9 (MaxPooling2D)  (None, 64, 64, 32)   0        conv2d_41[0][0]
_________________________________________________________________________________
conv2d_42 (Conv2D)           (None, 64, 64, 64)   18496       max_pooling2d_9[0
_________________________________________________________________________________
conv2d_43 (Conv2D)           (None, 64, 64, 64)   36928       conv2d_42[0][0]
_________________________________________________________________________________
max_pooling2d_10 (MaxPooling2D) (None, 32, 32, 64)   0        conv2d_43[0][0]
_________________________________________________________________________________
conv2d_44 (Conv2D)           (None, 32, 32, 128)  73856       max_pooling2d_10[
_________________________________________________________________________________
conv2d_45 (Conv2D)           (None, 32, 32, 128)  147584      conv2d_44[0][0]
_________________________________________________________________________________
max_pooling2d_11 (MaxPooling2D) (None, 16, 16, 128)  0        conv2d_45[0][0]
_________________________________________________________________________________
```

```
conv2d_46 (Conv2D)              (None, 16, 16, 256)  295168      max_pooling2d_11[(
conv2d_47 (Conv2D)              (None, 16, 16, 256)  590080      conv2d_46[0][0]
conv2d_transpose_8 (Conv2DTrans (None, 32, 32, 128)  131200      conv2d_47[0][0]
concatenate_8 (Concatenate)     (None, 32, 32, 256)  0           conv2d_transpose_
                                                                 conv2d_45[0][0]
conv2d_48 (Conv2D)              (None, 32, 32, 128)  295040      concatenate_8[0][(
conv2d_49 (Conv2D)              (None, 32, 32, 128)  147584      conv2d_48[0][0]
conv2d_transpose_9 (Conv2DTrans (None, 64, 64, 64)   32832       conv2d_49[0][0]
concatenate_9 (Concatenate)     (None, 64, 64, 128)  0           conv2d_transpose_
                                                                 conv2d_43[0][0]
conv2d_50 (Conv2D)              (None, 64, 64, 64)   73792       concatenate_9[0][(
conv2d_51 (Conv2D)              (None, 64, 64, 64)   36928       conv2d_50[0][0]
conv2d_transpose_10 (Conv2DTran (None, 128, 128, 32) 8224        conv2d_51[0][0]
concatenate_10 (Concatenate)    (None, 128, 128, 64) 0           conv2d_transpose_
                                                                 conv2d_41[0][0]
conv2d_52 (Conv2D)              (None, 128, 128, 32) 18464       concatenate_10[0]
```

Compile model

```
train_masks = np.expand_dims(train_masks, -1)
train_masks.shape
```

```
(1000, 256, 256, 1)
```

```
opt = keras.optimizers.Adam(LEARNING_RATE)
model.compile(
      optimizer=opt,
      loss=soft_dice_loss,
      metrics=[iou_coef]) #MeanIoU(num_classes=2)
```

```
tf.test.gpu_device_name()
```

```
'/device:GPU:0'
```

```
start_time = time.time()
```

```
history = model.fit(train_images,
                    train_masks/255,
                    validation_split = 0.1,
                    epochs=EPOCHS,
```

```
                    batch_size = BATCH_SIZE,
                    callbacks = [
                                ModelCheckpoint(CHECKPOINT_MODEL_PATH,
                                                monitor="val_loss",
                                                mode="min",
                                                save_best_only = True,
                                                verbose=1),
                                EarlyStopping(monitor = 'val_loss',
                                              min_delta = 0,
                                              patience = 5,
                                              verbose = 1,
                                              restore_best_weights = True),
                                ReduceLROnPlateau(monitor='val_loss',
                                                  factor=0.1,
                                                  patience=4,
                                                  verbose=1,
                                                  min_delta=1e-4)
                                ]
                       )
```

```
Epoch 1/100
57/57 [==============================] - 10s 154ms/step - loss: 0.9566 - iou_coef: 0

Epoch 00001: val_loss improved from inf to 0.89923, saving model to ./Models/road_map
Epoch 2/100
57/57 [==============================] - 7s 121ms/step - loss: 0.8837 - iou_coef: 0.6

Epoch 00002: val_loss improved from 0.89923 to 0.89732, saving model to ./Models/road
Epoch 3/100
57/57 [==============================] - 7s 121ms/step - loss: 0.8828 - iou_coef: 0.6

Epoch 00003: val_loss improved from 0.89732 to 0.89729, saving model to ./Models/road
Epoch 4/100
57/57 [==============================] - 7s 121ms/step - loss: 0.8832 - iou_coef: 0.6

Epoch 00004: val_loss did not improve from 0.89729
Epoch 5/100
57/57 [==============================] - 7s 122ms/step - loss: 0.8807 - iou_coef: 0.6

Epoch 00005: val_loss did not improve from 0.89729
Epoch 6/100
57/57 [==============================] - 7s 122ms/step - loss: 0.8832 - iou_coef: 0.6

Epoch 00006: val_loss did not improve from 0.89729

Epoch 00006: ReduceLROnPlateau reducing learning rate to 9.999999747378752e-06.
Epoch 7/100
57/57 [==============================] - 7s 123ms/step - loss: 0.8846 - iou_coef: 0.6

Epoch 00007: val_loss did not improve from 0.89729
Epoch 8/100
57/57 [==============================] - 7s 123ms/step - loss: 0.8814 - iou_coef: 0.6

Epoch 00008: val_loss did not improve from 0.89729
Restoring model weights from the end of the best epoch.
Epoch 00008: early stopping
```

```
end_time = time.time()
total_time = end_time - start_time
print("Total training time: {}s".format(total_time))
```

```
Total training time: 61.07701826095581s
```

```
model.save(FINAL_MODEL_PATH)
```

```
INFO:tensorflow:Assets written to: ./Models/road_mapper_final_relu_1000/assets
```

✓ 3 сек. выполнено в 19:50