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
TRAIN_SIZE = 250
```

```
CHECKPOINT_MODEL_PATH = "./Models/road_mapper_2.h5"
FINAL_MODEL_PATH = "./Models/road_mapper_final_relu_"+str(TRAIN_SIZE)
```

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'))[:TRAIN_SIZE]
print(train_images.shape)

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

hfile.close()
```

```
(250, 256, 256, 3)
(250, 256, 256)
```

View samples

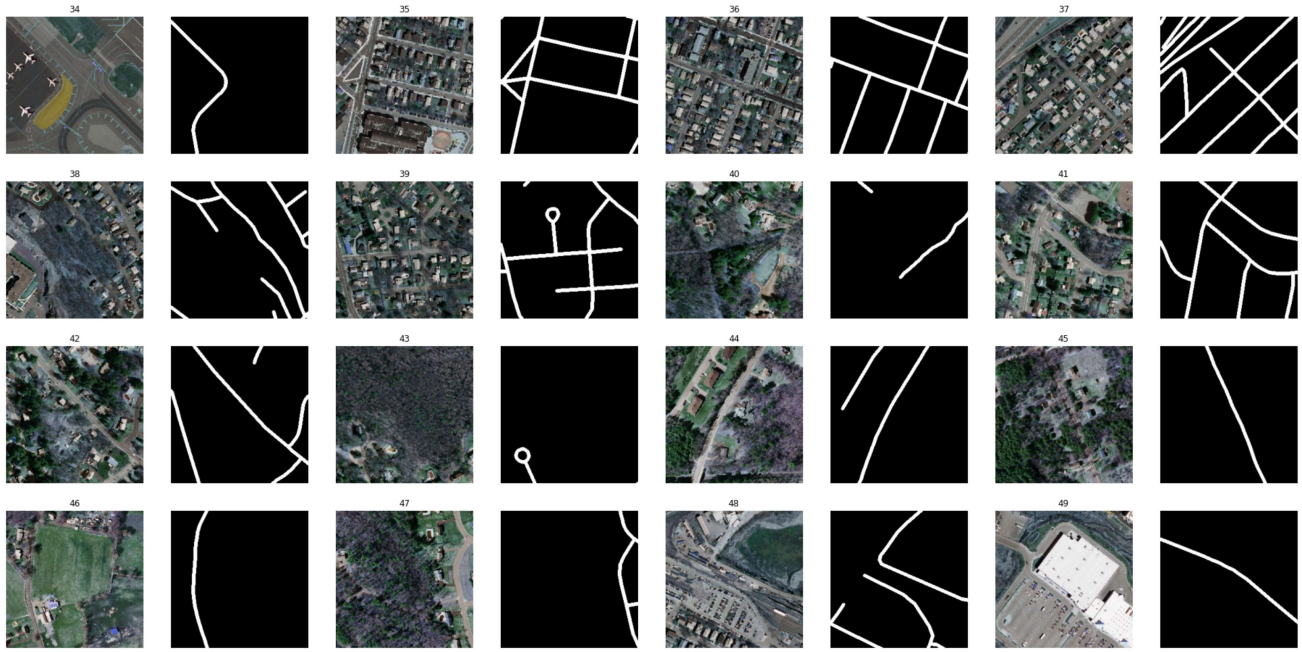
```
random.seed(1)
plt.figure(figsize=(32,16))
x, y = 4, 4
ipos = random.randint(0, len(train_images)-x*y)
for i in range(y):
  for j in range(x):

    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()
```

conv2d_23 (Conv2D)	(None, 64, 64, 64)	18496	max_pooling2d_5[0
conv2d_24 (Conv2D)	(None, 64, 64, 64)	36928	conv2d_23[0][0]
max_pooling2d_6 (MaxPooling2D)	(None, 32, 32, 64)	0	conv2d_24[0][0]
conv2d_25 (Conv2D)	(None, 32, 32, 128)	73856	max_pooling2d_6[0
conv2d_26 (Conv2D)	(None, 32, 32, 128)	147584	conv2d_25[0][0]
max_pooling2d_7 (MaxPooling2D)	(None, 16, 16, 128)	0	conv2d_26[0][0]
conv2d_27 (Conv2D)	(None, 16, 16, 256)	295168	max_pooling2d_7[0
conv2d_28 (Conv2D)	(None, 16, 16, 256)	590080	conv2d_27[0][0]
conv2d_transpose_4 (Conv2DTrans	(None, 32, 32, 128)	131200	conv2d_28[0][0]
concatenate_4 (Concatenate)	(None, 32, 32, 256)	0	conv2d_transpose_ conv2d_26[0][0]

```
conv2d_29 (Conv2D)              (None, 32, 32, 128)  295040      concatenate_4[0][(

conv2d_30 (Conv2D)              (None, 32, 32, 128)  147584      conv2d_29[0][0]

conv2d_transpose_5 (Conv2DTrans (None, 64, 64, 64)   32832       conv2d_30[0][0]

concatenate_5 (Concatenate)     (None, 64, 64, 128)  0           conv2d_transpose_
                                                                 conv2d_24[0][0]

conv2d_31 (Conv2D)              (None, 64, 64, 64)   73792       concatenate_5[0][(

conv2d_32 (Conv2D)              (None, 64, 64, 64)   36928       conv2d_31[0][0]

conv2d_transpose_6 (Conv2DTrans (None, 128, 128, 32) 8224        conv2d_32[0][0]

concatenate_6 (Concatenate)     (None, 128, 128, 64) 0           conv2d_transpose_
                                                                 conv2d_22[0][0]

conv2d_33 (Conv2D)              (None, 128, 128, 32) 18464       concatenate_6[0][(

conv2d_34 (Conv2D)              (None, 128, 128, 32) 9248        conv2d_33[0][0]

conv2d_transpose_7 (Conv2DTrans (None, 256, 256, 16) 2064        conv2d_34[0][0]

concatenate_7 (Concatenate)     (None, 256, 256, 32) 0           conv2d_transpose_
                                                                 conv2d_20[0][0]

conv2d_35 (Conv2D)              (None, 256, 256, 16) 4624        concatenate_7[0][(

conv2d_36 (Conv2D)              (None, 256, 256, 16) 2320        conv2d_35[0][0]

conv2d_37 (Conv2D)              (None, 256, 256, 1)  17          conv2d_36[0][0]
==================================================================================
Total params: 1,941,105
Trainable params: 1,941,105
Non-trainable params: 0
```

Compile model

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

```
(250, 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)
                                ]
                     )
```

```
15/15 [==============================] - 2s 127ms/step - loss: 0.4679 - iou_coef: 0

Epoch 00025: val_loss improved from 0.42119 to 0.41337, saving model to ./Models/r
Epoch 26/100
15/15 [==============================] - 2s 126ms/step - loss: 0.4731 - iou_coef: 0

Epoch 00026: val_loss did not improve from 0.41337
Epoch 27/100
15/15 [==============================] - 2s 126ms/step - loss: 0.4808 - iou_coef: 0

Epoch 00027: val_loss did not improve from 0.41337
Epoch 28/100
15/15 [==============================] - 2s 126ms/step - loss: 0.4568 - iou_coef: 0

Epoch 00028: val_loss did not improve from 0.41337
Epoch 29/100
15/15 [==============================] - 2s 127ms/step - loss: 0.4755 - iou_coef: 0

Epoch 00029: val_loss did not improve from 0.41337

Epoch 00029: ReduceLROnPlateau reducing learning rate to 9.999999747378752e-06.
Epoch 30/100
15/15 [==============================] - 2s 126ms/step - loss: 0.4278 - iou_coef: 0

Epoch 00030: val_loss improved from 0.41337 to 0.41039, saving model to ./Models/r
Epoch 31/100
15/15 [==============================] - 2s 126ms/step - loss: 0.4186 - iou_coef: 0

Epoch 00031: val_loss improved from 0.41039 to 0.40611, saving model to ./Models/r
Epoch 32/100
15/15 [==============================] - 2s 127ms/step - loss: 0.4287 - iou_coef: 0
```

```
Epoch 00032: val_loss improved from 0.40611 to 0.40369, saving model to ./Models/r
Epoch 33/100
15/15 [==============================] - 2s 127ms/step - loss: 0.4199 - iou_coef: 

Epoch 00033: val_loss improved from 0.40369 to 0.40276, saving model to ./Models/r
Epoch 34/100
15/15 [==============================] - 2s 127ms/step - loss: 0.4289 - iou_coef: 

Epoch 00034: val_loss did not improve from 0.40276
Epoch 35/100
15/15 [==============================] - 2s 127ms/step - loss: 0.4226 - iou_coef: 

Epoch 00035: val_loss did not improve from 0.40276
Epoch 36/100
15/15 [==============================] - 2s 127ms/step - loss: 0.4126 - iou_coef: 

Epoch 00036: val_loss did not improve from 0.40276
Epoch 37/100
15/15 [==============================] - 2s 126ms/step - loss: 0.4299 - iou_coef: 

Epoch 00037: val_loss did not improve from 0.40276

Epoch 00037: ReduceLROnPlateau reducing learning rate to 9.999999747378752e-07.
Epoch 38/100
15/15 [==============================] - 2s 127ms/step - loss: 0.4047 - iou_coef: 

Epoch 00038: val_loss did not improve from 0.40276
```

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

```
Total training time: 83.20842623710632s
```

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

```
INFO:tensorflow:Assets written to: ./Models/road_mapper_final_relu_250/assets
```

✓ 3 сек. выполнено в 20:13