

▼ 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 /content/gdrive/My Drive/road_finder_data

```
import cv2
import h5py
import random
import numpy as np
import tensorflow as tf
import keras
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_100'
```

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'))[:100]
print(train_images.shape)

train_masks = np.array(hfile.get('masks'))[:100]
print(train_masks.shape)

hfile.close()

(100, 256, 256, 3)
(100, 256, 256)
```

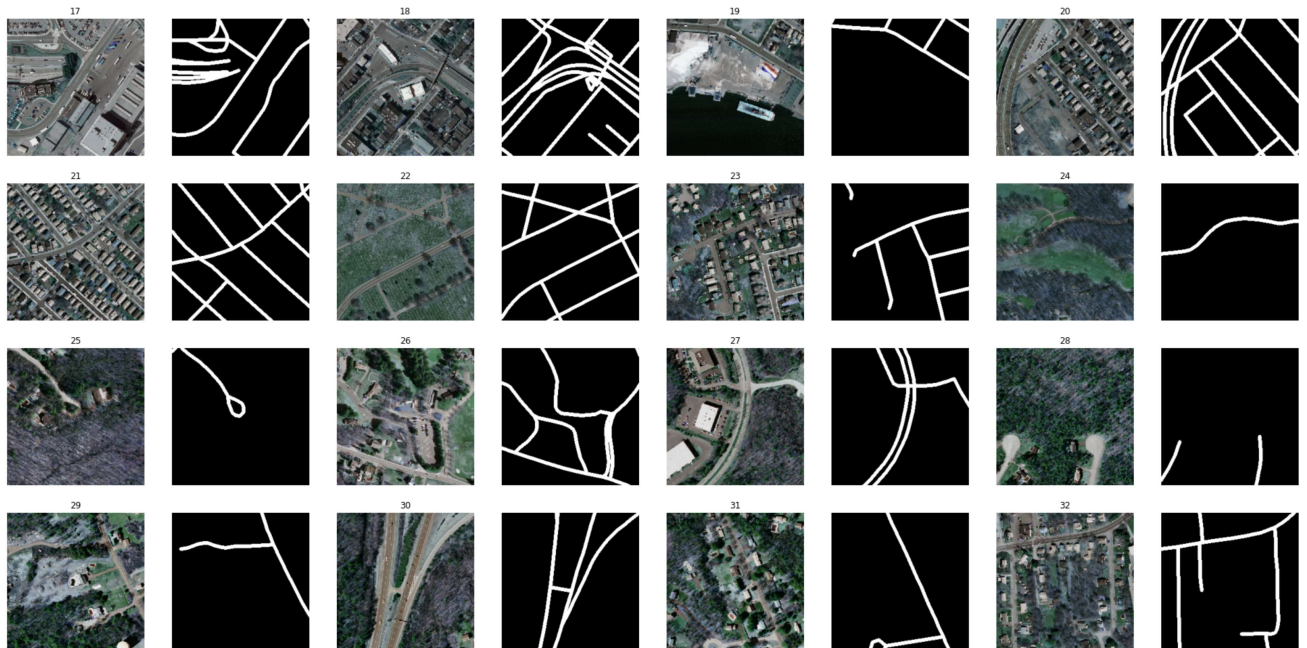
▼ View samples

```
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_4 (Conv2D)	(None, 64, 64, 64)	18496	max_pooling2d_1[0]
conv2d_5 (Conv2D)	(None, 64, 64, 64)	36928	conv2d_4[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 32, 32, 64)	0	conv2d_5[0][0]
conv2d_6 (Conv2D)	(None, 32, 32, 128)	73856	max_pooling2d_2[0]
conv2d_7 (Conv2D)	(None, 32, 32, 128)	147584	conv2d_6[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 16, 16, 128)	0	conv2d_7[0][0]
conv2d_8 (Conv2D)	(None, 16, 16, 256)	295168	max_pooling2d_3[0]
conv2d_9 (Conv2D)	(None, 16, 16, 256)	590080	conv2d_8[0][0]
conv2d_transpose (Conv2DTranspo	(None, 32, 32, 128)	131200	conv2d_9[0][0]
concatenate (Concatenate)	(None, 32, 32, 256)	0	conv2d_transpose[0][0] conv2d_7[0][0]
conv2d_10 (Conv2D)	(None, 32, 32, 128)	295040	concatenate[0][0]
conv2d_11 (Conv2D)	(None, 32, 32, 128)	147584	conv2d_10[0][0]
conv2d_transpose_1 (Conv2DTrans	(None, 64, 64, 64)	32832	conv2d_11[0][0]
concatenate_1 (Concatenate)	(None, 64, 64, 128)	0	conv2d_transpose_1[0][0] conv2d_5[0][0]
conv2d_12 (Conv2D)	(None, 64, 64, 64)	73792	concatenate_1[0][0]

conv2d_13 (Conv2D)	(None, 64, 64, 64)	36928	conv2d_12[0][0]
conv2d_transpose_2 (Conv2DTrans	(None, 128, 128, 32)	8224	conv2d_13[0][0]
concatenate_2 (Concatenate)	(None, 128, 128, 64)	0	conv2d_transpose_2[0][0]
conv2d_14 (Conv2D)	(None, 128, 128, 32)	18464	concatenate_2[0][0]
conv2d_15 (Conv2D)	(None, 128, 128, 32)	9248	conv2d_14[0][0]
conv2d_transpose_3 (Conv2DTrans	(None, 256, 256, 16)	2064	conv2d_15[0][0]
concatenate_3 (Concatenate)	(None, 256, 256, 32)	0	conv2d_transpose_3[0][0]
conv2d_16 (Conv2D)	(None, 256, 256, 16)	4624	concatenate_3[0][0]
conv2d_17 (Conv2D)	(None, 256, 256, 16)	2320	conv2d_16[0][0]
conv2d_18 (Conv2D)	(None, 256, 256, 1)	17	conv2d_17[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
```

```
(100, 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')
```

```
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")])
```

```

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 00058: val_loss did not improve from 0.61657

Epoch 00058: ReduceLROnPlateau reducing learning rate to 9.999999747378752e-06.

Epoch 59/100

6/6 [=====] - 1s 124ms/step - loss: 0.5889 - iou_coef: 0.

Epoch 00059: val_loss improved from 0.61657 to 0.60568, saving model to ./Models/r

Epoch 60/100

6/6 [=====] - 1s 122ms/step - loss: 0.5724 - iou_coef: 0.

Epoch 00060: val_loss improved from 0.60568 to 0.60398, saving model to ./Models/r

Epoch 61/100

6/6 [=====] - 1s 122ms/step - loss: 0.5703 - iou_coef: 0.

Epoch 00061: val_loss did not improve from 0.60398

Epoch 62/100

6/6 [=====] - 1s 123ms/step - loss: 0.5721 - iou_coef: 0.

Epoch 00062: val_loss did not improve from 0.60398

Epoch 63/100

6/6 [=====] - 1s 125ms/step - loss: 0.5585 - iou_coef: 0.

Epoch 00063: val_loss improved from 0.60398 to 0.60114, saving model to ./Models/r

Epoch 64/100

6/6 [=====] - 1s 123ms/step - loss: 0.5723 - iou_coef: 0.

Epoch 00064: val_loss did not improve from 0.60114

Epoch 65/100

6/6 [=====] - 1s 125ms/step - loss: 0.5803 - iou_coef: 0.

Epoch 00065: val_loss did not improve from 0.60114

Epoch 66/100

6/6 [=====] - 1s 124ms/step - loss: 0.5771 - iou_coef: 0.

Epoch 00066: val_loss improved from 0.60114 to 0.59993, saving model to ./Models/r

Epoch 67/100

6/6 [=====] - 1s 123ms/step - loss: 0.5588 - iou_coef: 0.

Epoch 00067: val_loss did not improve from 0.59993

Epoch 68/100

6/6 [=====] - 1s 126ms/step - loss: 0.5632 - iou_coef: 0.

Epoch 00068: val_loss did not improve from 0.59993

Epoch 69/100

```
Epoch 69/100
6/6 [=====] - 1s 124ms/step - loss: 0.5699 - iou_coef: 0.

Epoch 00069: val_loss did not improve from 0.59993
Epoch 70/100
6/6 [=====] - 1s 125ms/step - loss: 0.5459 - iou_coef: 0.

Epoch 00070: val_loss did not improve from 0.59993

Epoch 00070: ReduceLROnPlateau reducing learning rate to 9.99999747378752e-07.
Epoch 71/100
6/6 [=====] - 1s 125ms/step - loss: 0.5595 - iou_coef: 0.

Epoch 00071: val_loss did not improve from 0.59993
```

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

```
INFO:tensorflow:Assets written to: ./Models/road_mapper_final_relu_100/assets
```

✓ 3 сек. выполнено в 18:48

