

▼ 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
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 = 2000
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()

(2000, 256, 256, 3)
(2000, 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

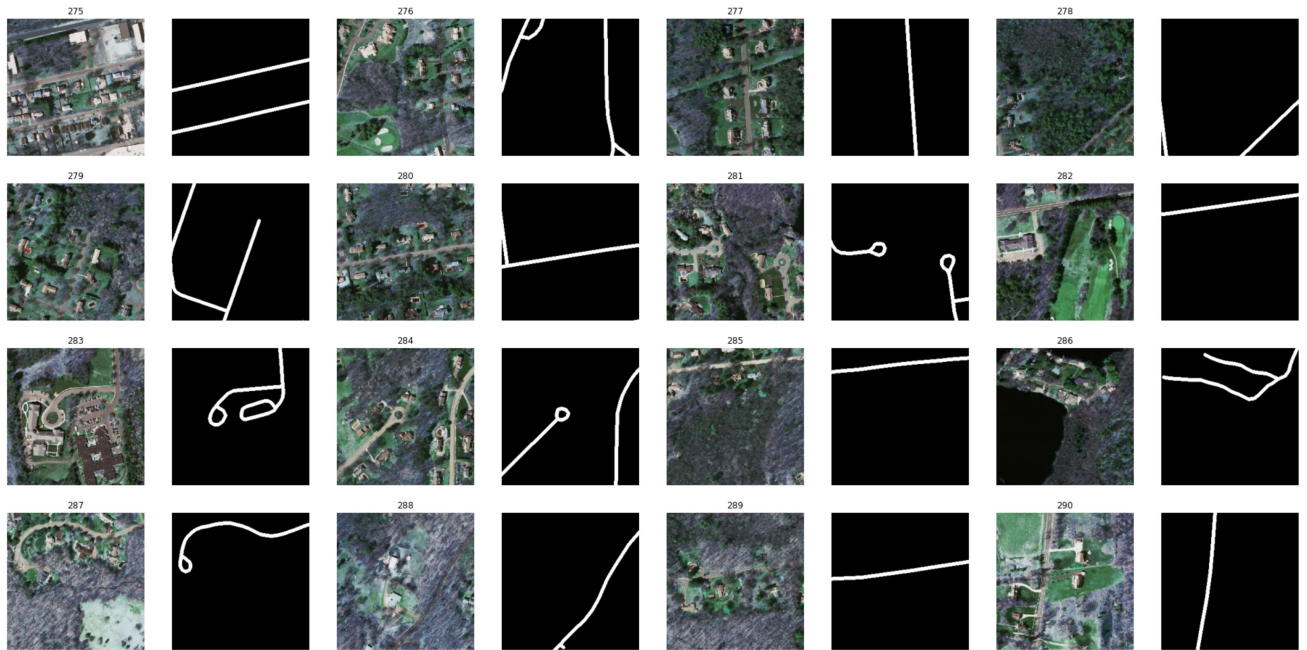
        plt.subplot(y, x*2, i*x*2+j*2+1)
        plt.imshow(train_images[pos])
```

```

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()

```

Layer (type)	Output Shape	Param #	Connected to
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]

conv2d_7[0][0]

conv2d_10 (Conv2D)	(None, 32, 32, 128)	295040	conv2d_10[0][0]
conv2d_11 (Conv2D)	(None, 32, 32, 128)	147584	conv2d_11[0][0]
conv2d_transpose_1 (Conv2DTrans	(None, 64, 64, 64)	32832	conv2d_transpose_1[0][0]
concatenate_1 (Concatenate)	(None, 64, 64, 128)	0	concatenate_1[0][0]
conv2d_12 (Conv2D)	(None, 64, 64, 64)	73792	conv2d_12[0][0]
conv2d_13 (Conv2D)	(None, 64, 64, 64)	36928	conv2d_13[0][0]
conv2d_transpose_2 (Conv2DTrans	(None, 128, 128, 32)	8224	conv2d_transpose_2[0][0]
concatenate_2 (Concatenate)	(None, 128, 128, 64)	0	concatenate_2[0][0]
conv2d_14 (Conv2D)	(None, 128, 128, 32)	18464	conv2d_14[0][0]
conv2d_15 (Conv2D)	(None, 128, 128, 32)	9248	conv2d_15[0][0]
conv2d_transpose_3 (Conv2DTrans	(None, 256, 256, 16)	2064	conv2d_transpose_3[0][0]
concatenate_3 (Concatenate)	(None, 256, 256, 32)	0	concatenate_3[0][0]
conv2d_16 (Conv2D)	(None, 256, 256, 16)	4624	conv2d_16[0][0]
conv2d_17 (Conv2D)	(None, 256, 256, 16)	2320	conv2d_17[0][0]
conv2d_18 (Conv2D)	(None, 256, 256, 1)	17	conv2d_18[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

(2000, 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 18/100
113/113 [=====] - 32s 288ms/step - loss: 0.3002 - iou_coef: 0.3463
Epoch 00018: val_loss improved from 0.34634 to 0.33973, saving model to ./Models/road_finder_00018.h5
Epoch 19/100
113/113 [=====] - 32s 288ms/step - loss: 0.3037 - iou_coef: 0.3397
Epoch 00019: val_loss improved from 0.33973 to 0.33579, saving model to ./Models/road_finder_00019.h5
Epoch 20/100
113/113 [=====] - 32s 287ms/step - loss: 0.2897 - iou_coef: 0.3357
Epoch 00020: val_loss improved from 0.33579 to 0.33541, saving model to ./Models/road_finder_00020.h5
Epoch 21/100
113/113 [=====] - 32s 287ms/step - loss: 0.2801 - iou_coef: 0.3354
Epoch 00021: val_loss improved from 0.33541 to 0.33538, saving model to ./Models/road_finder_00021.h5
Epoch 22/100
113/113 [=====] - 32s 287ms/step - loss: 0.2730 - iou_coef: 0.3353
Epoch 00022: val_loss did not improve from 0.33538
Epoch 23/100
113/113 [=====] - 33s 288ms/step - loss: 0.2754 - iou_coef: 0.3353
Epoch 00023: val_loss did not improve from 0.33538
Epoch 24/100
113/113 [=====] - 32s 287ms/step - loss: 0.2659 - iou_coef: 0.3353
Epoch 00024: val_loss improved from 0.33538 to 0.33374, saving model to ./Models/road_finder_00024.h5
Epoch 25/100
113/113 [=====] - 33s 288ms/step - loss: 0.2608 - iou_coef: 0.3337

```

```

Epoch 00025: val_loss did not improve from 0.33374
Epoch 26/100
113/113 [=====] - 32s 287ms/step - loss: 0.2571 - iou_coef: 0.4000

Epoch 00026: val_loss improved from 0.33374 to 0.33368, saving model to ./Models/road_mapper_final_relu_2000/assets
Epoch 27/100
113/113 [=====] - 32s 286ms/step - loss: 0.2435 - iou_coef: 0.4000

Epoch 00027: val_loss did not improve from 0.33368
Epoch 28/100
113/113 [=====] - 32s 286ms/step - loss: 0.2435 - iou_coef: 0.4000

Epoch 00028: val_loss did not improve from 0.33368

Epoch 00028: ReduceLROnPlateau reducing learning rate to 9.999999747378752e-06.
Epoch 29/100
113/113 [=====] - 32s 287ms/step - loss: 0.2323 - iou_coef: 0.4000

Epoch 00029: val_loss did not improve from 0.33368
Epoch 30/100
113/113 [=====] - 32s 286ms/step - loss: 0.2276 - iou_coef: 0.4000

Epoch 00030: val_loss did not improve from 0.33368
Epoch 31/100
113/113 [=====] - 32s 286ms/step - loss: 0.2203 - iou_coef: 0.4000

Epoch 00031: val_loss did not improve from 0.33368

```

```

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

```

```
Total training time: 1060.3363628387451s
```

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

```
INFO:tensorflow:Assets written to: ./Models/road_mapper_final_relu_2000/assets
```

✓ 4 сек. выполнено в 21:22

