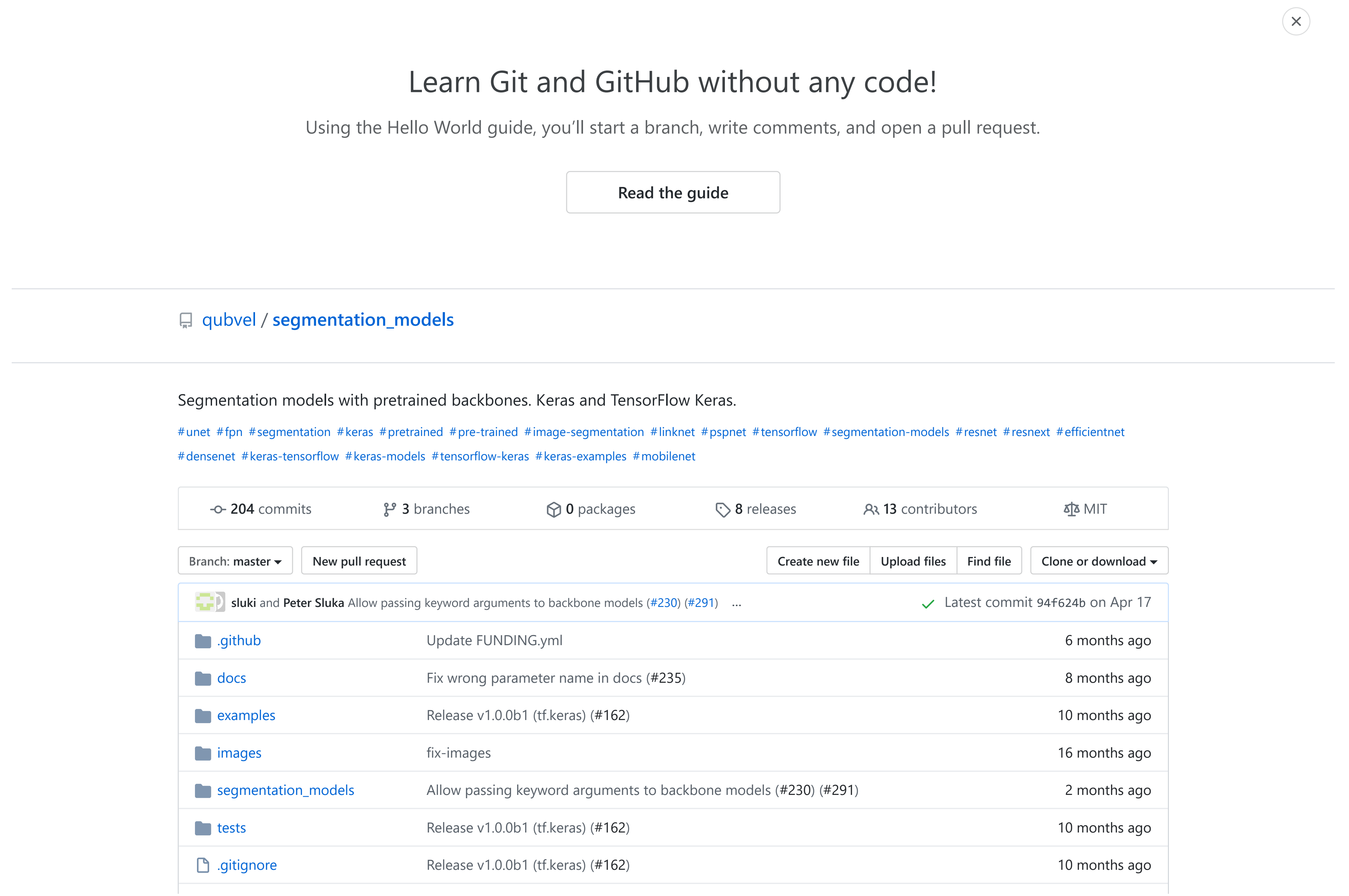


Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

qubvel / segmentation_models

Segmentation models with pretrained backbones. Keras and TensorFlow Keras.

#unet #fpn #segmentation #keras #pretrained #pre-trained #image-segmentation #linknet #pspnet #tensorflow #segmentation-models #resnet #resnext #efficientnet #densenet #keras-tensorflow #keras-models #tensorflow-keras #keras-examples #mobilenet

204 commits | 3 branches | 0 packages | 8 releases | 13 contributors | MIT

sluki and Peter Sluka Allow passing keyword arguments to backbone models (#230) (#291) … ✓ Latest commit `94f624b` on Apr 17

.github	Update FUNDING.yml	6 months ago
docs	Fix wrong parameter name in docs (#235)	8 months ago
examples	Release v1.0.0b1 (tf.keras) (#162)	10 months ago
images	fix-images	16 months ago
segmentation_models	Allow passing keyword arguments to backbone models (#230) (#291)	2 months ago
tests	Release v1.0.0b1 (tf.keras) (#162)	10 months ago
.gitignore	Release v1.0.0b1 (tf.keras) (#162)	10 months ago

.gitmodules	submodule removed	2 years ago
.travis.yml	Release v1.0.0 (#228)	8 months ago
CHANGELOG.md	Release v1.0.0 (#228)	8 months ago
LICENSE	corrected	2 years ago
MANIFEST.in	Release v1.0.0 (#228)	8 months ago
README.rst	Update README.rst (#275)	6 months ago
__init__.py	Update __init__.py	2 years ago
requirements.txt	Release v1.0.0 (#228)	8 months ago
setup.py	Release v1.0.0b1 (tf.keras) (#162)	10 months ago

README.rst

Python library with Neural Networks for Image Segmentation based on Keras and TensorFlow.

pypi package 1.0.1 docs passing build passing

The main features of this library are:

- High level API (just two lines of code to create model for segmentation)
- **4** models architectures for binary and multi-class image segmentation (including legendary **Unet**)
- **25** available backbones for each architecture
- All backbones have **pre-trained** weights for faster and better convergence
- Helpful segmentation losses (Jaccard, Dice, Focal) and metrics (IoU, F-score)

Important note

> Some models of version `1.*` are not compatible with previously trained models, if you have such models and want to load them - roll back with:
>
> $ pip install -U segmentation-models==0.2.1

Table of Contents

Quick start

Library is build to work together with Keras and TensorFlow Keras frameworks

```
import segmentation_models as sm
# Segmentation Models: using `keras` framework.
```

By default it tries to import `keras`, if it is not installed, it will try to start with `tensorflow.keras` framework. There are several ways to choose framework:

- Provide environment variable `SM_FRAMEWORK=keras` / `SM_FRAMEWORK=tf.keras` before import `segmentation_models`
- Change framework `sm.set_framework('keras')` / `sm.set_framework('tf.keras')`

You can also specify what kind of `image_data_format` to use, segmentation-models works with both: `channels_last` and `channels_first`. This can be useful for further model conversion to Nvidia TensorRT format or optimizing model for cpu/gpu computations.

```
import keras
# or from tensorflow import keras

keras.backend.set_image_data_format('channels_last')
# or keras.backend.set_image_data_format('channels_first')
```

Created segmentation model is just an instance of Keras Model, which can be build as easy as:

```
model = sm.Unet()
```

Depending on the task, you can change the network architecture by choosing backbones with fewer or more parameters and use pretrainded weights to initialize it:

```
model = sm.Unet('resnet34', encoder_weights='imagenet')
```

Change number of output classes in the model (choose your case):

```
# binary segmentation (this parameters are default when you call Unet('resnet34')
model = sm.Unet('resnet34', classes=1, activation='sigmoid')
```

```
# multiclass segmentation with non overlapping class masks (your classes + background)
model = sm.Unet('resnet34', classes=3, activation='softmax')
```

```
# multiclass segmentation with independent overlapping/non-overlapping class masks
model = sm.Unet('resnet34', classes=3, activation='sigmoid')
```

Change input shape of the model:

```
# if you set input channels not equal to 3, you have to set encoder_weights=None
# how to handle such case with encoder_weights='imagenet' described in docs
model = Unet('resnet34', input_shape=(None, None, 6), encoder_weights=None)
```

Simple training pipeline

```
import segmentation_models as sm

BACKBONE = 'resnet34'
preprocess_input = sm.get_preprocessing(BACKBONE)

# load your data
x_train, y_train, x_val, y_val = load_data(...)

# preprocess input
x_train = preprocess_input(x_train)
x_val = preprocess_input(x_val)

# define model
model = sm.Unet(BACKBONE, encoder_weights='imagenet')
model.compile(
    'Adam',
    loss=sm.losses.bce_jaccard_loss,
    metrics=[sm.metrics.iou_score],
)

# fit model
# if you use data generator use model.fit_generator(...) instead of model.fit(...)
# more about `fit_generator` here: https://keras.io/models/sequential/#fit_generator
model.fit(
   x=x_train,
   y=y_train,
   batch_size=16,
   epochs=100,
   validation_data=(x_val, y_val),
)
```

Same manipulations can be done with `Linknet`, `PSPNet` and `FPN`. For more detailed information about models API and use cases Read the Docs.

Examples

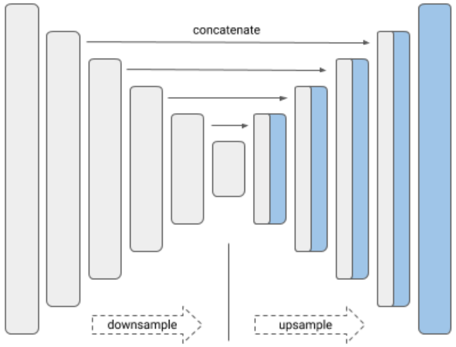
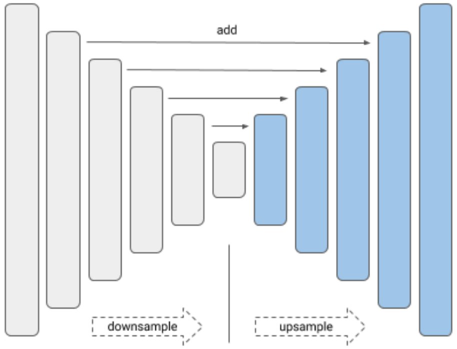
Models training examples:

- [Jupyter Notebook] Binary segmentation (cars) on CamVid dataset here.
- [Jupyter Notebook] Multi-class segmentation (cars, pedestrians) on CamVid dataset here.

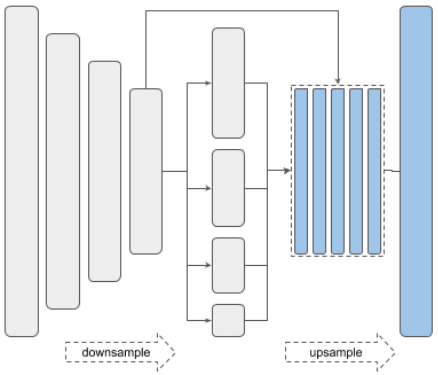
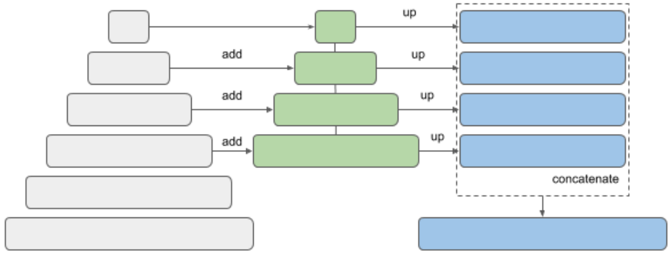
Models and Backbones

Models

- Unet
- FPN
- Linknet
- PSPNet

Unet	Linknet
	

PSPNet	FPN

PSPNet	FPN
	

Backbones

Type	Names
VGG	`'vgg16' 'vgg19'`
ResNet	`'resnet18' 'resnet34' 'resnet50' 'resnet101' 'resnet152'`
SE-ResNet	`'seresnet18' 'seresnet34' 'seresnet50' 'seresnet101' 'seresnet152'`
ResNeXt	`'resnext50' 'resnext101'`
SE-ResNeXt	`'seresnext50' 'seresnext101'`
SENet154	`'senet154'`
DenseNet	`'densenet121' 'densenet169' 'densenet201'`
Inception	`'inceptionv3' 'inceptionresnetv2'`
MobileNet	`'mobilenet' 'mobilenetv2'`
EfficientNet	`'efficientnetb0' 'efficientnetb1' 'efficientnetb2' 'efficientnetb3' 'efficientnetb4' 'efficientnetb5' efficientnetb6' efficientnetb7'`

> All backbones have weights trained on 2012 ILSVRC ImageNet dataset (`encoder_weights='imagenet'`).

Installation

Requirements

1. python 3
2. keras >= 2.2.0 or tensorflow >= 1.13
3. keras-applications >= 1.0.7, <=1.0.8
4. image-classifiers == 1.0.*
5. efficientnet == 1.0.*

PyPI stable package

```
$ pip install -U segmentation-models
```

PyPI latest package

```
$ pip install -U --pre segmentation-models
```

Source latest version

```
$ pip install git+https://github.com/qubvel/segmentation_models
```

Documentation

Latest **documentation** is avaliable on Read the Docs

Change Log

To see important changes between versions look at CHANGELOG.md

Citing

```
@misc{Yakubovskiy:2019,
  Author = {Pavel Yakubovskiy},
  Title = {Segmentation Models},
  Year = {2019},
  Publisher = {GitHub},
  Journal = {GitHub repository},
  Howpublished = {\url{https://github.com/qubvel/segmentation_models}}
}
```

License