Morpheus Documentation

Ryan Hausen, Brant Robertson

Jun 15, 2020

Contents:

Morpheus is a neural network model used to generate pixel-level morphological classifications for astronomical sources. This model can be used to generate segmentation maps or to inform other photometric measurements with granular morphological information.

CHAPTER 1

Installation

Morpheus is implemented using TensorFlow. TensorFlow is **not** listed in the dependencies for the package. So you need to install TensorFlow before you install Morpheus. It has to be done this way to support the GPU accelerated version of TensorFlow, which has a different package name. For more information on installing TensorFlow visit the TensorFlow website.

```
pip install morpheus-astro
```

CHAPTER 2

Docker

Morpheus has two main flavors of Docker Image: `gpu` for the GPU enabled version of TensorFlow and `cpu` for the standard CPU implementation of TensorFlow. Visit the Docker Hub page for relevant tags.

For GPU support:

```
docker run --runtime=nvidia -it morpheusastro/morpheus:lastest-gpu
```

For CPU only:

```
docker run -it morpheusastro/morpheus:latest-cpu
```

CHAPTER 3

Usage

There are two ways to use morpheus on images: the python API and the command line interface

3.1 Python API

The `morpheus.classifier.Classifier` class is the interface to the various functionalities of Morpheus.

3.1.1 Morphological classification

To perform a pixel-level morphological classification, the image needs to be provided in the H, J, Z, and V bands. See `classify()` for more information.

```
from morpheus.classifier import Classifier
from morpheus.data import example

h, j, v, z = example.get_sample()
classified = Classifier.classify(h=h, j=j, v=v, z=z)
```

The classify function returns a dictionary where the keys indicate the output for example `spheroid`, and the value is the corresponding numpy ndarray.

Using the output from `classify()` you can:

- Make a segmap
- Make a morphgological catalog
- Make colorized version of the morphological classifications

3.1.2 Segmentation Map

To create a segmentation map using Morpheus, you need to provide the output from the `classify()` function and a single flux band. In the below example we use H. For more information see `segmap_from_classified()`

```
from morpheus.classifier import Classifier
from morpheus.data import example

h, j, v, z = example.get_sample()
classified = Classifier.classify(h=h, j=j, v=v, z=z)
segmap = Classifier.segmap_from_classified(classified, h)
```

3.1.3 Catalog

To create a catalog using Morpheus, you need to provide the output from the `classify()` function, the flux in a single band (we use H), and a segmentation map. The segmentation map doesn't have to be generated by Morpheus, but it must be similar in form. It should assign pixels values greater than 0 for all pixels that are associated with a source. Each source should be assigned a unique ID. Background should be set to 0 and excluded regions should be assigned -1. The catalog returned is a JSON compatible list of morphological classifications for each source in the segmap. For more information, see `catalog_from_classified()`.

```
from morpheus.classifier import Classifier
from morpheus.data import example

h, j, v, z = example.get_sample()
classified = Classifier.classify(h=h, j=j, v=v, z=z)
segmap = Classifier.segmap_from_classified(classified, h)
catalog = Classifier.catalog_from_classified(classified, h, segmap)
```

3.1.4 Colorized Classifications

A colorized classification is a way of making a single image to interpret the pixel level morphological classifications. For more information see `colorize_classified()`.

```
from morpheus.classifier import Classifier
from morpheus.data import example

h, j, v, z = example.get_sample()
classified = Classifier.classify(h=h, j=j, v=v, z=z)
color_rgb = Classifier.colorize_classified(classified)
```

3.1.5 Parallelization

Morpheus supports simple parallelization by breaking an image into equally sized pieces along the y axis, classifying them in seperate processes, and stitching them back into a single image. Parallelization can be split into CPU jobs or GPU jobs. Importantly, you cannot specify both at the same time.

GPUS

The `gpus` argument should be a list of integers that are the ids assigned to the GPUS to be used. These ids can be found by using `nvidia-smi`.

```
from morpheus.classifier import Classifier
from morpheus.data import example

h, j, v, z = example.get_sample()

classified = Classifier.classify(h=h, j=j, v=v, z=z, gpus=[0,1])
```

CPUS

The `cpus` argument should be an integer indicating how many processes to spin off.

```
from morpheus.classifier import Classifier
from morpheus.data import example

h, j, v, z = example.get_sample()

classified = Classifier.classify(h=h, j=j, v=v, z=z, cpus=2)
```

3.2 Command Line Interface

Morpheus can be used from the terminal using the `morpheus` command. To classify an image, it needs to be available in the H, J, V, and Z bands. From the terminal the following actions can be performed:

- Per pixel morphological classification
- Make segmentation map
- Make a catalog of morphological classifications
- Make a colorized version of the morphological classifications

3.2.1 Morphological classification

```
morpheus h.fits j.fits v.fits z.fits
```

Order is important when calling the Morpheus from the terminal. The files should be in the order H, J, V, and Z, as displayed in the above example. The output classification will be saved in the current working directory unless otherwise indicated by the `--out_dir` optional argument.

3.2.2 Segmentation Map

```
morpheus h.fits j.fits v.fits z.fits --action segmap
```

To create a segmap, append the optional `--action` flag with the argument `segmap`. This will save both the classifications and the segmap to the same directory.

3.2.3 Catalog

```
morpheus h.fits j.fits v.fits z.fits --action catalog
```

This will create a catalog by classifying the input images, creating a segmap, and using both of those to generate a morphological catalog. The morphological classifications, segmap, and catalog are all saved to the same place.

3.2.4 Colorized Classifications

```
morpheus h.fits j.fits v.fits z.fits --action colorize
```

Using `--action colorize` will classify the image and then generate a colorized version of that classification and save the classification and colorized version to the same place.

3.2.5 Parallelization

Morpheus supports simple parallelization by breaking an image into equally sized pieces along the y axis, classifying them in separate processes, and stitching them back into a single image. Parallelization can be split into CPU jobs or GPU jobs. Importantly, you cannot specify both at the same time.

GPUS

The `gpus` optional flag should be a comma-separated list of ids for the GPUS to be used. These ids can be found by using `nvidia-smi`.

```
morpheus h.fits j.fits v.fits z.fits --gpus 0,1
```

CPUS

The `cpus` optional flag should be an integer indicating how many processes to spin off.

```
morpheus h.fits j.fits v.fits z.fits --cpus 2
```

CHAPTER 4

Python Demo

Try it out on Google Colab!

CHAPTER 5

Documentation

https://morpheus-astro.readthedocs.io/

CHAPTER 6

Indices and tables

- genindex
- modindex
- search