
cytounet

Release 0.2.2

Nelson Gonzabato

Jul 30, 2022

CONTENTS

1 cytounet **1**

 1.1 cytounet package 1

2 cytounet: Deep Learning Based Cell Segmentation **3**

3 Changes to cytounet **9**

4 Contributing to cytounet **13**

5 Indices and tables **15**

Python Module Index **17**

Index **19**

CYTOUNET

1.1 cytounet package

1.1.1 Submodules

1.1.2 cytounet.augmentation module

1.1.3 cytounet.data module

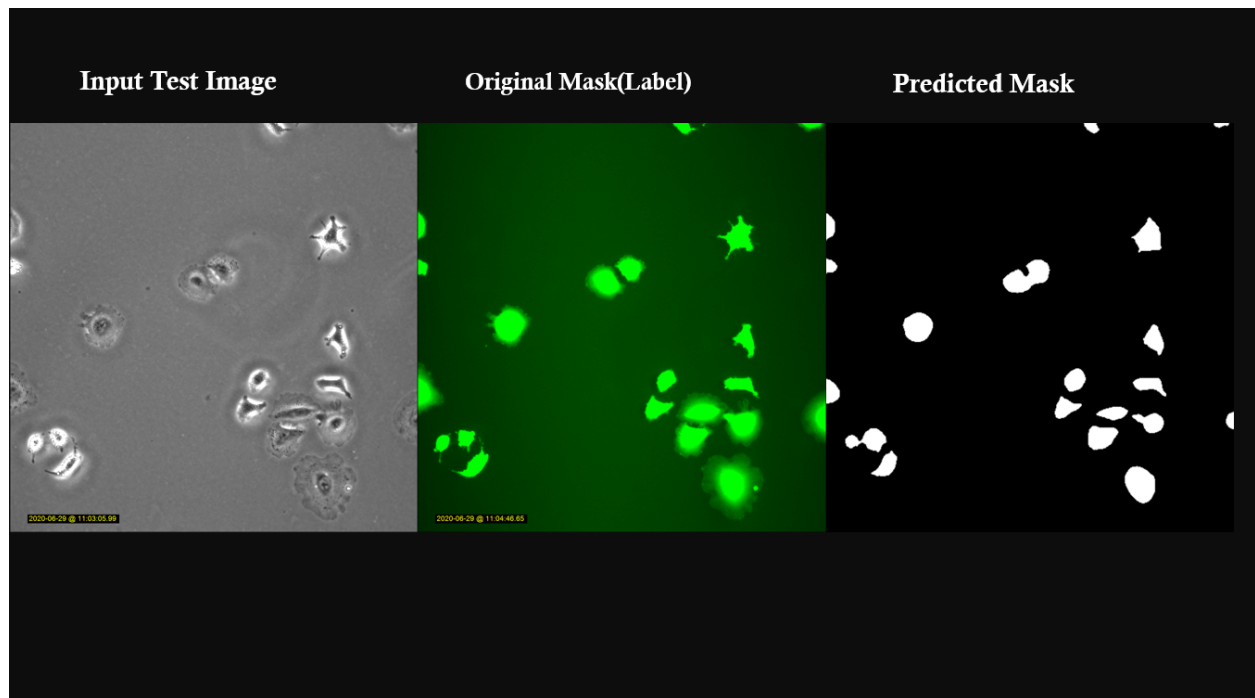
1.1.4 cytounet.model module

1.1.5 Module contents

Deep Learning based Cell Segmentation

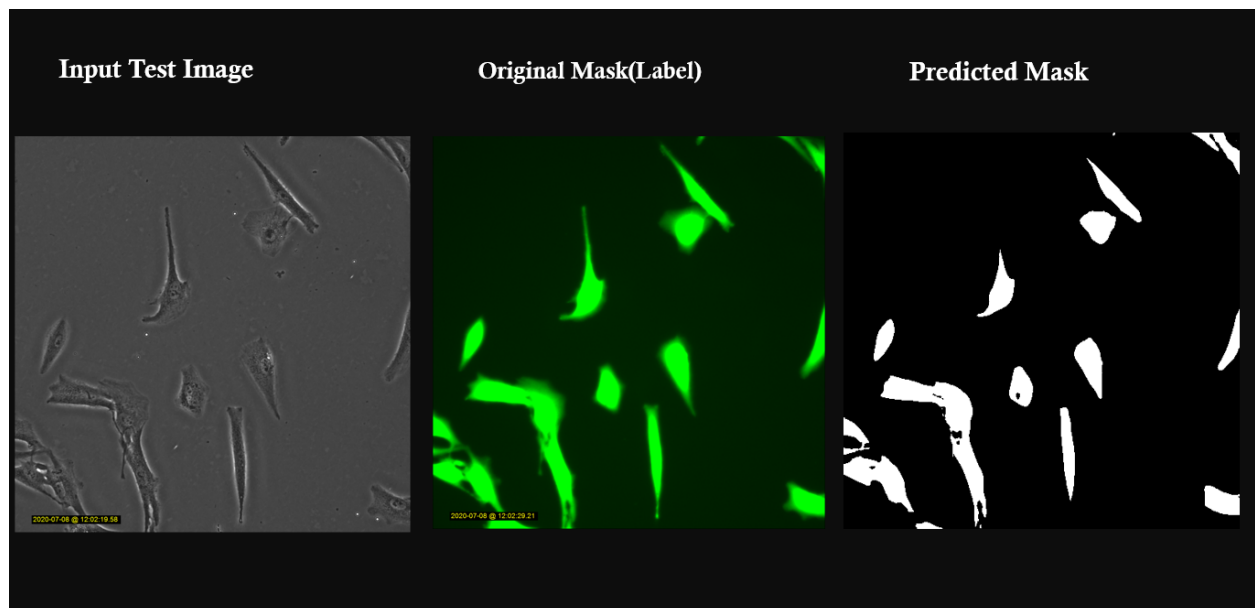
CYTOUNET: DEEP LEARNING BASED CELL SEGMENTATION

Background



This project was largely done as a summer 2020 intern in Dr. Mikael Bjorklund's lab, whose guidance, resources, and time I am grateful for.

The aim was to automate a549 and rpe cancer cell segmentation and size determination.



Sample data(10 random images each of train, validate, test sets) is provided in [original_data](#).

A complete a549 cancer cell [segmentation notebook](#) is also provided.

Finally, pre-trained [weights](#) are provided that can be used for transfer learning. These were trained on considerably more data and for more epochs. For more pre-trained weights and/or data, please [contact](#) the author.

Note

- To generate masks(labels) provided here a549 cancer cells were stained and imaged with fluorescence mi-

croscopy. These fluorescent images were then thresholded with `threshold_images` with a threshold value of 83. The images were then saved with `save_images`. The original fluorescent images are not provided here mainly due to the already huge size of the project.

- This project is not limited to cancer cells. The model can be trained on almost any object, living and non-living. More examples are given below.

Installation

From PyPI:

```
pip install cytounet
```

From source or to use without installing locally:

```
git clone https://github.com/Nelson-Gon/cytounet.git
# proceed with usual source build procedure
```

Or:

```
pip install git+https://github.com/Nelson-Gon/cytounet.git
```

Import

```
from cytounet.model import *
from cytounet.data import *
from cytounet.augmentation import *
from cytounet.post_model import *
```

Detailed Sample Usage

Script mode

```
python -m cytounet -t "examples/original_data/a549/train" -i "images" -m "masks" -v
"examples/original_data/a549/validation" -l "1e-8" -s 512 -ep 5 -se 250 -b 8 -tt
↪ "examples/original_data/a549/test/"
-w "models/a549_test/test_model" -o "Adam" -mt "dice_coef" -ls "dice_coef_loss" -sd 2 -f_
↪ 0 -p 0
```

To get help:

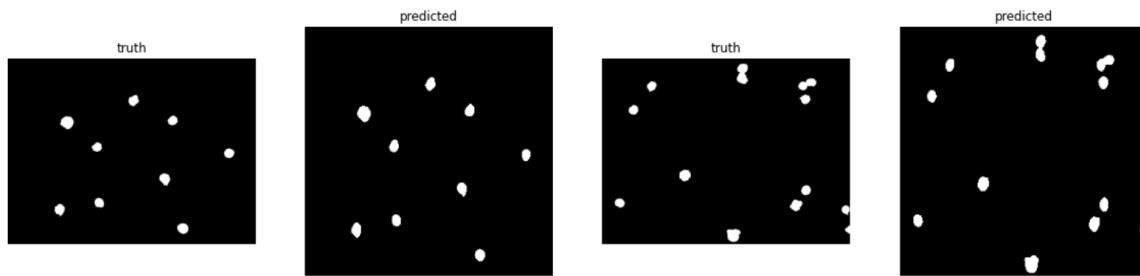
```
python -m cytounet -h
```

Notebooks

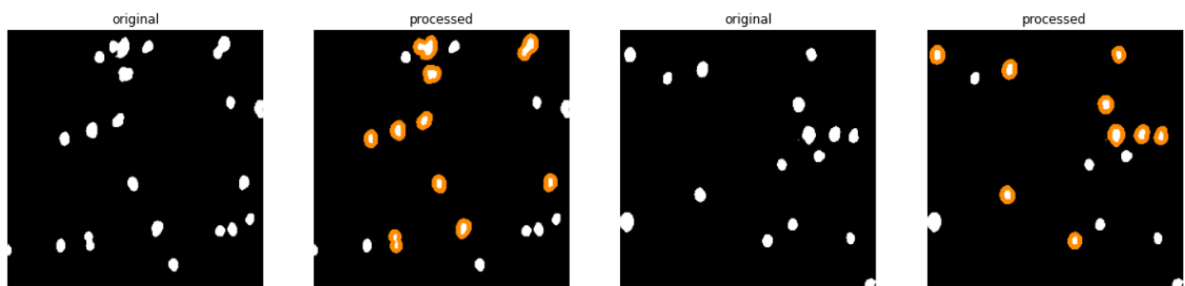
Please see the following examples:

- [Typical Usage](#)
- [Predicting Embryonic DIC Image Labels with Keras](#)

Visually:



Sample Object Area Visualization(see the typical usage notebook above for detailed usage)



For more examples or to add your own, please see the examples [folder](#).

Experiments/Benchmarks

This section shows some experimental results based on publicly available data.

- Comparison of low vs high quality masks on the model's output

This notebook shows the effects of “filled holes”(outlines whose area is filled with some colour e.g. white) on the model's quality. The results in general show that filled masks which are also better seen by the human eye result in better quality output.

The [notebook](#) can be accessed via this [link](#).

TODO List

A checked box indicates support. You can either add more feature requests here or tackle unchecked boxes and make a pull request to add such support.

- ☒ Single class segmentation
- ☒ Grayscale images
- ☒ Model Validation
- ☒ Determining areas of objects in an image
- ☐ Custom Number of Layers
- ☐ Multi-class segmentation
- ☐ Colored image input
- ☐ COCO Datasets

- [] CSV Based Annotations
- [] XML Based Annotations

Frequently Asked Questions

Please read our Wiki [Pages](#)

To raise an issue or question

If the [wiki](#) does not answer your question, please raise a new [issue](#). You can also open an issue if you have any discussion, criticism, or bug reports.

Thank you very much.

References

- Mouse Embryos [Dataset](#) from [Broad Bioimage Benchmark](#).
- Red Blood Cell [Images](#) provided by Anne Carpenter and Roger Wiegand, available [here](#).
- Chinese Hamster Ovary [Cells](#) provided by Koos et al.(2016)

Credits

- This repository started out as a clone of [zhixuhao](#)'s original unet [implementation](#).

This repository has considerably diverged from the original implementation hence the need to distribute it separately.

This decision was taken in part due to the relative inactivity of the original implementation which would have made it harder to collaborate. Please take a look at the list of [changes](#) from the original implementation.

- The Unet algorithm was introduced by Ronneberger et al. in their [paper](#).
-

If you would like to cite this work, please use:

Nelson Gonzabato(2020) cytounet: Deep Learning Based Cell Segmentation, <https://github.com/Nelson-Gon/cytounet>

BibTex

```
@misc{Gonzabato2021,  
  author = {Gonzabato, N},  
  title = {cytounet: Deep Learning Based Cell Segmentation},  
  year = {2021},  
  publisher = {GitHub},  
  journal = {GitHub repository},  
  howpublished = {\url{https://github.com/Nelson-Gon/cytounet}},  
  commit = {58bd951ef4417fc8542f8f3e277071e6cd6980ea}  
}
```


CHANGES TO CYTOUNET

Version 0.2.2

- Fixed a bug in setting test paths when using script mode.
- Versioning is now automated, as is linking to the GitHub release. Please ensure you release in the form `v#versionnumberhere`.
- Fixed issues with script mode
- Using `tensorflow.Keras` instead of `Keras`.
- Added sanity checks to ensure paths actually exist.

Version 0.2.1

- Extended script to handle fine-tuning and from scratch-training
- Added a script only mode.
- Added support for docs.
- Added original a549 sample data, notebook, and pre-trained weights.
- Added experimental results to the README.
- Fixed issues with original images being overwritten. It is now possible to return a copy of non overwritten images.
- Made `draw_contours` more flexible. Specifically, it is now possible to turn off text display as this makes the image crowded.
- Added `find_contours` and `draw_contours`, useful methods for area determination.
- Added `read_image_spec` for use only for post modeling processing. This fixes issues with incorrect shapes when using `read_images`

Version 0.2.0

- Kernel regularization can now be turned off via a boolean argument(`use_regularizer`)
- Added a new data set from BBBC.
- `finetune` is a new function dedicated to the finetuning workflow.
- Regularization is now supported. It is currently limited to L1 and L2.
- `pretrained_weights` was dropped as an argument to `unet`. Use a `callback` instead. A future version will include a fine tuning function.
- `save_as` was removed from `train`. Use `ModelCheckpoint` instead and provide it as a callback.
- `show_images` now shows titles. These functions will be removed later and imported from `pyautocv` instead.

- Fixed issues with reading mixed jpg and png images.
- Added `reshape_images` and `resize_images`. These are helper functions that may be useful when plotting or restoring original image size.
- `show_images` and `read_images` are now imported from `pyautocv >= 0.2.2`
- Fixed issues with inconsistent image order in `show_images` when reading from a directory.
- Added filename printing to data generators to make it easier to show what order the files are being read in. This can be disabled by setting `show_names` to `False`.
- Changes to prediction generation were made. We now use `ImageDataGenerator` for test time data generation.
- Fixed a bug related to `load_augmentations` that led to image flipping.
- Changed outputs to `sigmoid` instead of `ReLU`
- Updated to latest API ie `predict` vs `predict_generator`
- Added `train` to simplify model fitting.
- Added `predict` to reduce code repetition and make predicting easier.
- `unet` was rewritten to increase complexity and solve issues with blank predictions. It now also uses `Conv2DTranspose` instead of `UpSampling2D`.
- Initial support for a simpler model to optimise the bias-variance trade off for small(er) datasets.
- Removed `Dropout` since this is known to have no improvement over `Batch Normalisation`.
- Initial support for `SGD` as the default optimiser
- Moved from `camelCase` to `snake_case`, now using more descriptive function names.
- Fixed issues with list input to `show_images`

-
- Release 0.1.0
-

- Renamed repository to `cytounet` to reflect the heavy focus on biological images.
 - Initiated support for validation via `validGenerator`.
 - Fixed issues with `show_images` failing to load `numpy ndarray` images.
-

- Initiated ability to install with `pip` and `setup.py`.
-

• `show_augmented` was renamed to `show_images` and refactored as a more general method not limited to just augmented images. A `cmap` argument was also added for more flexibility. This replaces `labelVisualize` which has now been dropped.

- Introduced a separate save method for images and predictions. Use `saveImages` and `savePredictions` respectively.
-

- Fixed issues with information loss following saving of predictions.
 - `geneTrainNpY` was refactored and renamed `LoadAugmented`
-

- Added `thresholdImages` to threshold masks(mostly). Please see [pyautocv](#) for a more general and flexible way to manipulate images.
 - Added `saveImages`, a helper to save images as(by default) `.tif`. This is because biological images are normally tiff in nature.
 - Removed `savePredictions`. Use `saveImages` instead.
-

- Updated module documentation
- `adjustData` was removed since it had known issues. It may be restored in the future.
- Fixed issues that resulted in blank predictions
- Added `show_augmented` to show results of data augmentation
- Added BatchNormalisation steps
- Training made more flexible by allowing usage of different metrics and loss functions without editing source code(i.e change on the fly)
- Saving and image reading functions made more flexible to read/save any image file format.
- Made most functions compatible with Keras ≥ 2.0
- Added dice loss and dice coefficient.

CONTRIBUTING TO CYTOUNET

This document provides guidelines for contributions to `cytounet`.

Kinds of contribution

- Typo fixes
- Documentation enhancements
- Pull requests

Fixing typos and enhancing documentation

To fix typos and/or grammatical errors, please edit the corresponding `.py` or `.md` file that generates the documentation.

Please also update documentation by running `bash scripts/mkdocs.sh` as described also later.

Pull Requests

- Please raise an issue for discussion and reproducibility checks at <https://github.com/Nelson-Gon/cytounet/issues>
- Once the bug/enhancement is approved, please create a Git branch for the pull request.
- Make changes and ensure that builds are passing the necessary checks on Travis.
- Update `changelog.md` to reflect the changes made.
- Do the following:

```
bash scripts/mkdocs.sh
```

See also for a guide on Sphinx documentation.

Commit messages

Please write commit messages in the format “Extends functionality” instead of “Extended functionality”.

Maintainers only notice

Releasing

This should ideally not be run since we are auto-releasing via a GitHub action. If for some reason you would like to release a new version manually, then:

```
bash scripts/release.sh
```

The above does the following:

- Makes dist with `python setup.py sdist` at the very minimum. Ensure everything necessary is included in `Manifest.in`.

- Uploads `dist` to `test.pypi.org` with `twine upload --repository-url https://test.pypi.org/legacy/ dist/*`
- If everything looks good, asks you to upload to `pypi.org` with `twine upload dist/*`

Please note that the ‘cytounet’ project is released with a [Contributor Code of Conduct](#). By contributing to this project, you agree to abide by its terms.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

C

- `cytounet`, [1](#)
- `cytounet.augmentation`, [1](#)
- `cytounet.data`, [1](#)
- `cytounet.model`, [1](#)

INDEX

C

- cytounet
 - module, 1
- cytounet.augmentation
 - module, 1
- cytounet.data
 - module, 1
- cytounet.model
 - module, 1

M

- module
 - cytounet, 1
 - cytounet.augmentation, 1
 - cytounet.data, 1
 - cytounet.model, 1