PULL REQUESTS SEMINAR

Sirine Younsi, 1st year Master’s Student, Kaneko Lab

CONTENTS

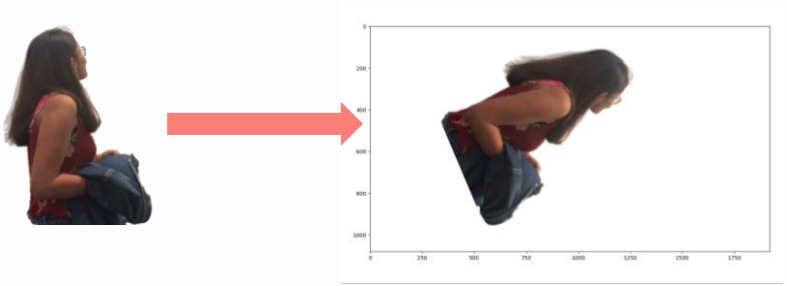
CONTENTS

2022-BackgroundMatting

In Dataset_utils/augmentation.py:

Added Class Random rotation in

=> Apply random rotation with angle between 180 and -180 to the picture



```
+ import cv2
+
+ class random_rotation():
+     def __call__(self, image):
+         rows_x, cols_x, chl_x = image.shape
+         rand_num = np.random.randint(-180, 180)
+         M1 = cv2.getRotationMatrix2D((cols_x / 2, rows_x / 2), rand_num, 1)
+         image = cv2.warpAffine(image, M1, (cols_x, rows_x))
+         return (image)
```

CONTENTS

2022-U-Net

1. Create BCDU-net.py

Add deep learning network BCDU-net

=> An extended version of U-net

=> Used for medical images segmentation (cancer detection)

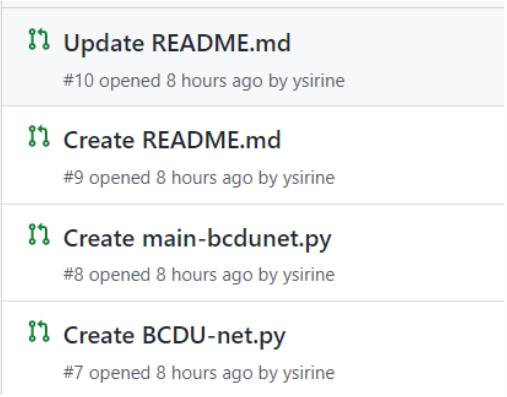


Figure 1. BCDU-Net for lesion segmentation

2022-U-Net

1. **Create BCDU-net.py**

Replace the skip connection with ConvLSTM

Figure 2. BCDU-Net with bi-directional ConvLSTM Reza et al. (2019)

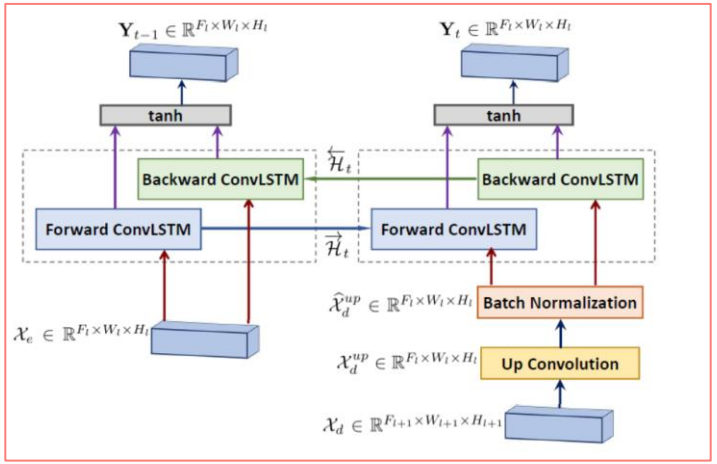


Figure 3. bi-directional ConvLSTM Reza et al. (2019)

2022-U-Net

2. Add main-bcdunet file for training and testing

=> Trained model then tested it and saved predicted output in folder output

Figure 4. BCDU-net prediction segmentation examples

2022-U-Net

3. Add necessary files to save outputs and checkpoints for weights (Checkpoint / Output)

4. Couldn’t add saved weights because the file size was too large

=> The interest for this pull request is more for exploration than improving something

CONTENTS

2022-Flooding

In mlp_mnist.py:

Added following metrics for Keras model

Precision: TP/(TP+FP)*100

Recall: TP/(TP+FN)*100

F1 Score: 2*Precision * Recall /

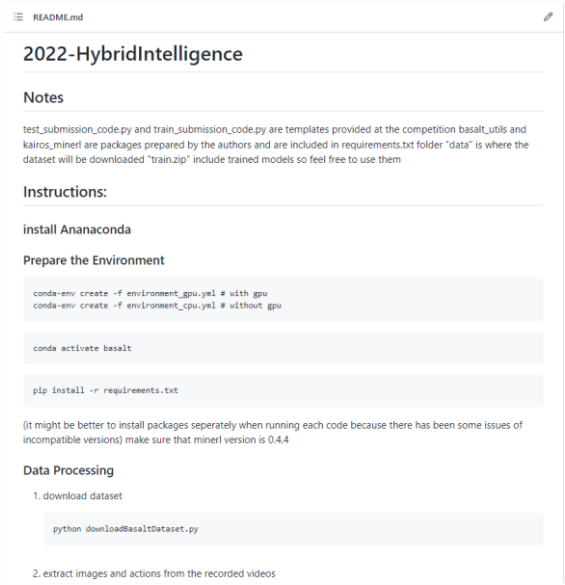
(Precision + Recall)*100

```
+ from keras import backend as K
+
+ def recall_m(y_true, y_pred):
+     true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
+     possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
+     recall = true_positives / (possible_positives + K.epsilon())
+     return recall
+
+ def precision_m(y_true, y_pred):
+     true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
+     predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
+     precision = true_positives / (predicted_positives + K.epsilon())
+     return precision
+
+ def f1_m(y_true, y_pred):
+     precision = precision_m(y_true, y_pred)
+     recall = recall_m(y_true, y_pred)
+     return 2*((precision*recall)/(precision+recall+K.epsilon()))
+
```

CONTENTS

2022-hybridIntelligence

1. Optimize README.md



README.md

2022-HybridIntelligence

Notes

test_submission_code.py and train_submission_code.py are templates provided at the competition basalt_utils and kairos_minerl are packages prepared by the authors and are included in requirements.txt folder "data" is where the dataset will be downloaded "train.zip" include trained models so feel free to use them

Instructions:

install Ananaconda

Prepare the Environment

```
conda-env create -f environment_gpu.yml # with gpu
conda-env create -f environment_cpu.yml # without gpu
```

```
conda activate basalt
```

```
pip install -r requirements.txt
```

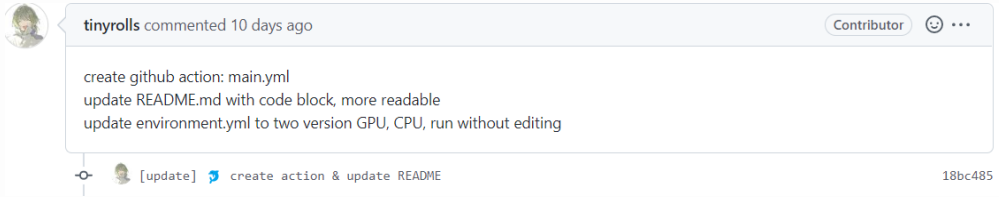
(it might be better to install packages seperately when running each code because there has been some issues of incompatible versions) make sure that minerl version is 0.4.4

Data Processing

1. download dataset

```
python downloadBasaltDataset.py
```

2. extract images and actions from the recorded videos



tinyrolls commented 10 days ago — Contributor

create github action: main.yml
update README.md with code block, more readable
update environment.yml to two version GPU, CPU, run without editing

[update] create action & update README 18bc485

2022-hybridIntelligence

1. Anaconda environment for GPU and CPU

11 lines (11 sloc) | 228 Bytes

```yaml
name: hybridintelligence
channels:
  - conda-forge
  - defaults
dependencies:
  - python=3.7
  - nomkl
  - pip
  - py-opencv
  # - cudatoolkit=10.0 # Remove this (put behind the comment line) if not on a GPU machine
  - chardet
```

11 lines (11 sloc) | 226 Bytes

```yaml
name: hybridintelligence
channels:
  - conda-forge
  - defaults
dependencies:
  - python=3.7
  - nomkl
  - pip
  - py-opencv
  - cudatoolkit=10.0 # Remove this (put behind the comment line) if not on a GPU machine
  - chardet
```

2022-hybridIntelligence

3. Github Actions

```
+ name: Test Actions
+
+ on:
+   push:
+     branches: [ main ]
+
+ jobs:
+   build:
+     runs-on: ubuntu-latest
+
+     steps:
+       - uses: actions/checkout@v3
+       - uses: actions/setup-python@v3
+         with:
+           python-version: '3.7'
+
+       - name: Install dependencies
+         run: |
+         pip install opencv-python chardet
+         pip install -r requirements.txt
+
+       - name: Dataset
+         run: |
+           python downloadBasaltDataset.py
+           python dataProcessing.py
+           python compileLabels.py
+
+       - name: Training
+         run: |
+           python train.py
+
+       - name: Test
+         run: |
+           python test.py
```

```
steps:
  - uses: actions/checkout@v3
  - uses: actions/setup-python@v3
    with:
      python-version: '3.7'
      update-conda: true
      conda-channels: anaconda, conda-forge
  - uses: actions/setup-java@v3
    with:
     distribution: 'temurin' # See 'Supported distributions' for available options
     java-version: '8'


  - run: conda --version

  - name: create Anaconda environment
    run: |

     $CONDA/bin/conda env create --file environment_cpu.yml
     $CONDA/bin/activate hybridintelligence
```

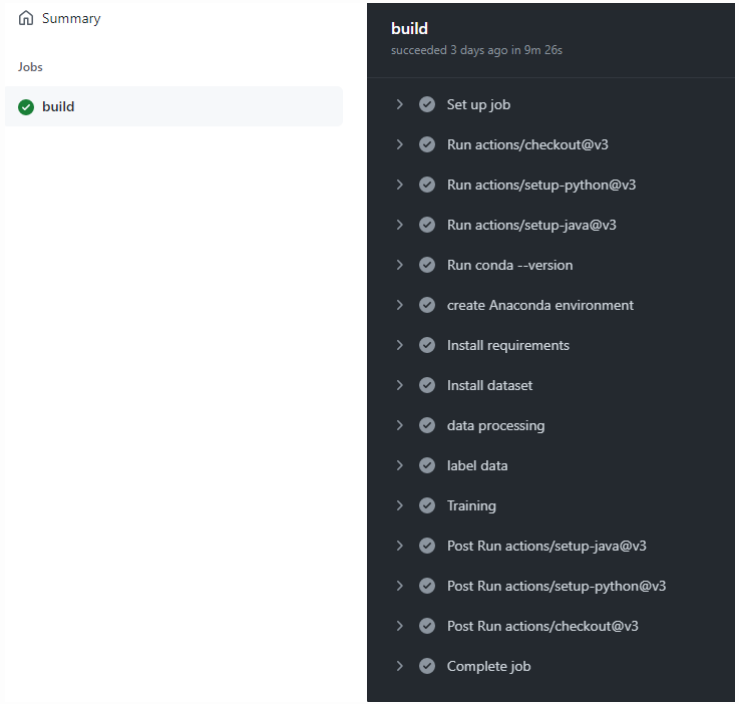
2022-hybridIntelligence

3. GitHub Actions

```
- name: Install requirements
  run: |

    pip install -r requirements.txt

- name: Install dataset
  run: |
     pip install minerl
     python downloadBasaltDataset.py

- name: data processing
  run: |

    pip install opencv-python numpy
    python dataProcessing.py

- name: label data
  run: |

    pip install scikit-multilearn
    python compileLabels.py
```

2022-hybridIntelligence

3. GitHub Actions



THANK YOU FOR YOUR ATTENTION

Sirine Younsi