ACM multimedia 2020 Seattle

Reproducibility Companion Paper: Instance of Interest Detection

Fan Yu[1,3], Haonan Wang[1], Tongwei Ren[1,3,*], Jinghui Tang[2], Gangshan Wu[1], Jingjing Chen[4], Michael Riegler[5]

[1]State Key Laboratory for Novel Software Technology, Nanjing University

[2]School of Computer Science, Nanjing University of Science and Technology

[3]School of Computer Science, Nanjing University of Science and Technology, Nanjing, China

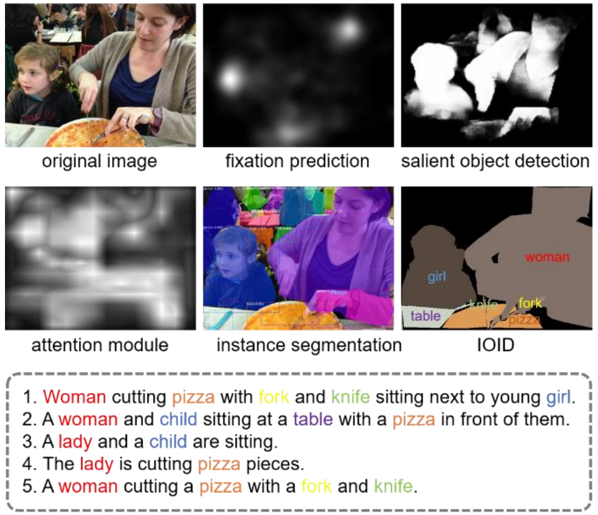
[4]Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai, China

[5]SimulaMet, Norway

MAGUS
MediA recoGnition
and UnderStanding

Instance of Interest Detection

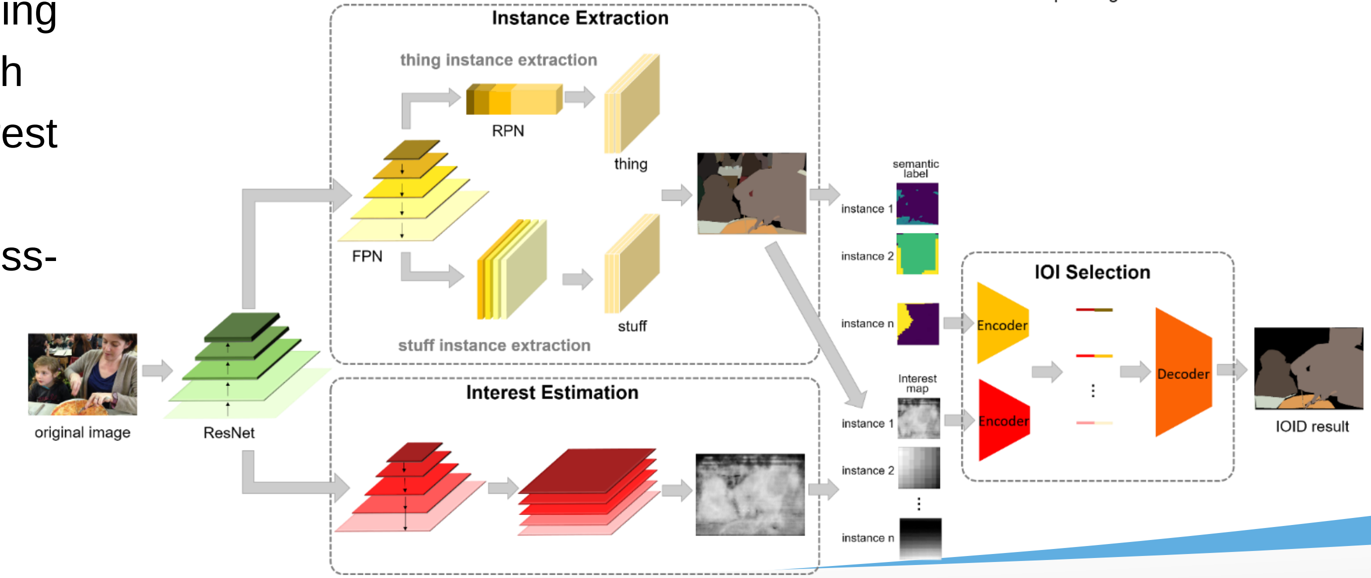
- **Instance of interest detection (IOID)** aims to provide instance-level user interest model for image semantic description
  - **Instance of Interest (IOI)**: the instances which are beneficial to represent image content



1. Woman cutting pizza with fork and knife sitting next to young girl.
2. A woman and child sitting at a table with a pizza in front of them.
3. A lady and a child are sitting.
4. The lady is cutting pizza pieces.
5. A woman cutting a pizza with a fork and knife.

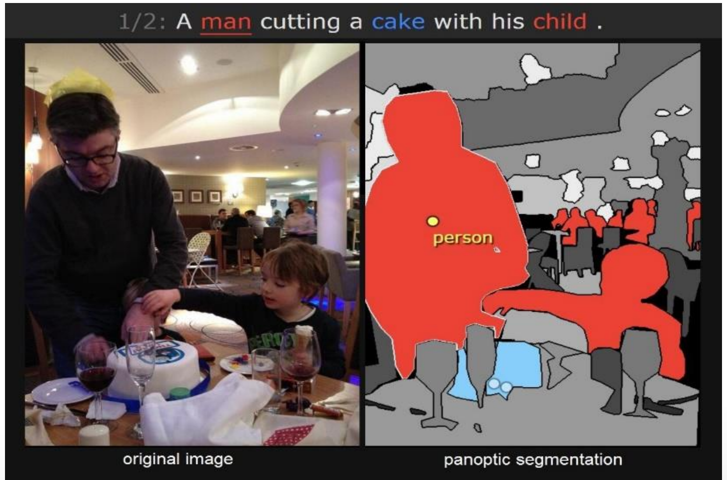
captioning

- Our solution: **Cross-Influential Network (CIN)**
  - **Instance Extraction:** containing a thing extraction branch and a stuff extraction branch
  - **Interest Estimation:** estimate pixel-interest according to feature maps
  - **IOI Selection:** select IOIs with a Cross-influential Encoder-decoder Network



Dataset

- IOID
  - Based on MSCOCO
  - Training set
    - 36,000 images
    - 165094 IOIs
  - Test set
    - 9000 images
    - 40617 IOIs



1/2: A man cutting a cake with his child .

```
{
    <image_id>:{
        “image_id”: int,
        “height”: int,
        “width”: int,
        “image_name”: string,
        “instances”: {
            <instance_id>: {
                “id”: int,
                “category_id”: int,
                “box”: [y1, x1, y2, x2],
                “labeled”: boolean,
                “iscrowd”: int
            }
        }
    }
}
```

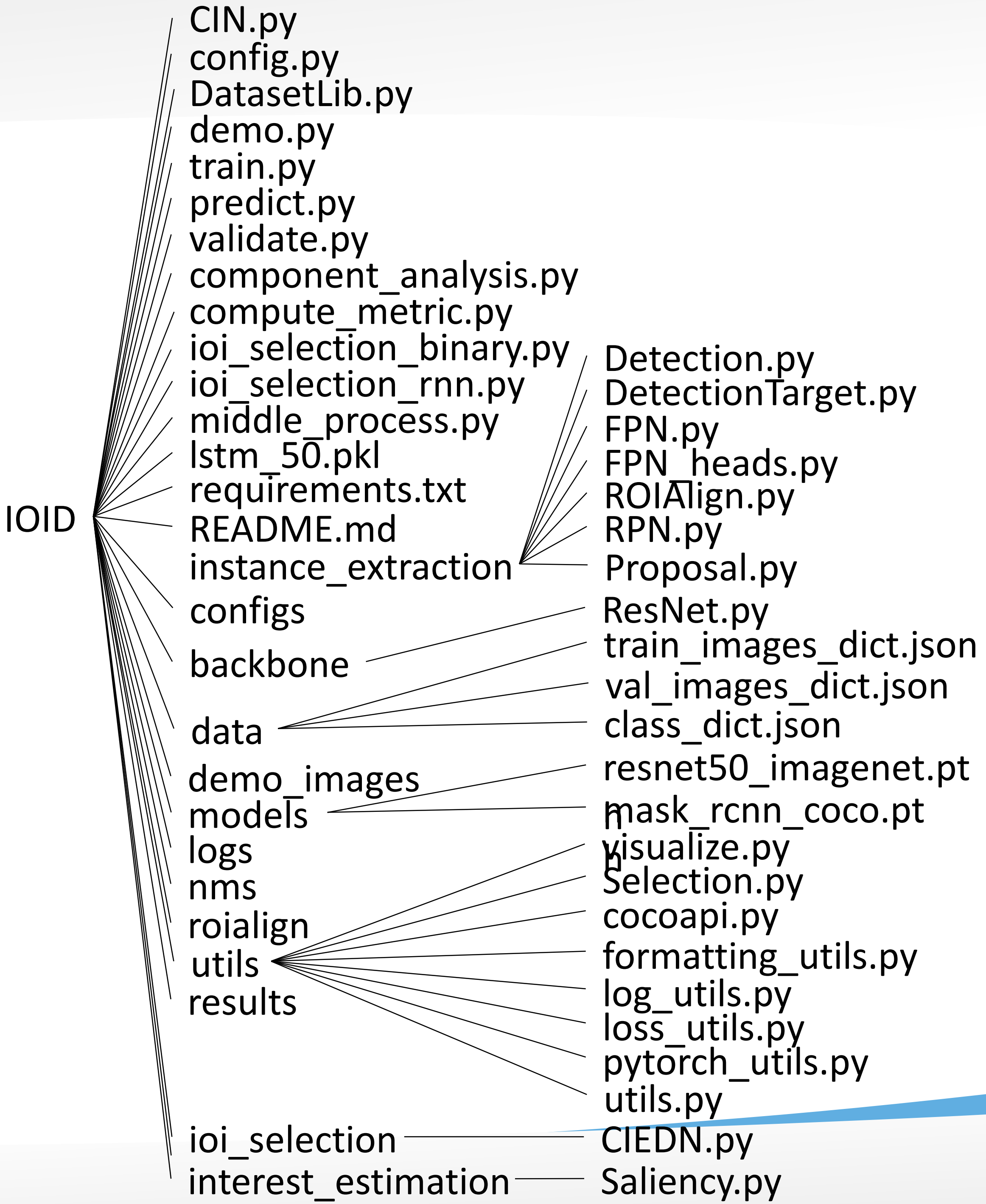
*_images_dict.json

```
{
    <class_id>: {
            “class_id”: int,
            “category_id”: int,
            “isthing”: int,
            “name”: string,
            “supercategory”: string,
            “color”: [int, int, int]
    }
}
```

class_dict.json

Source code structure

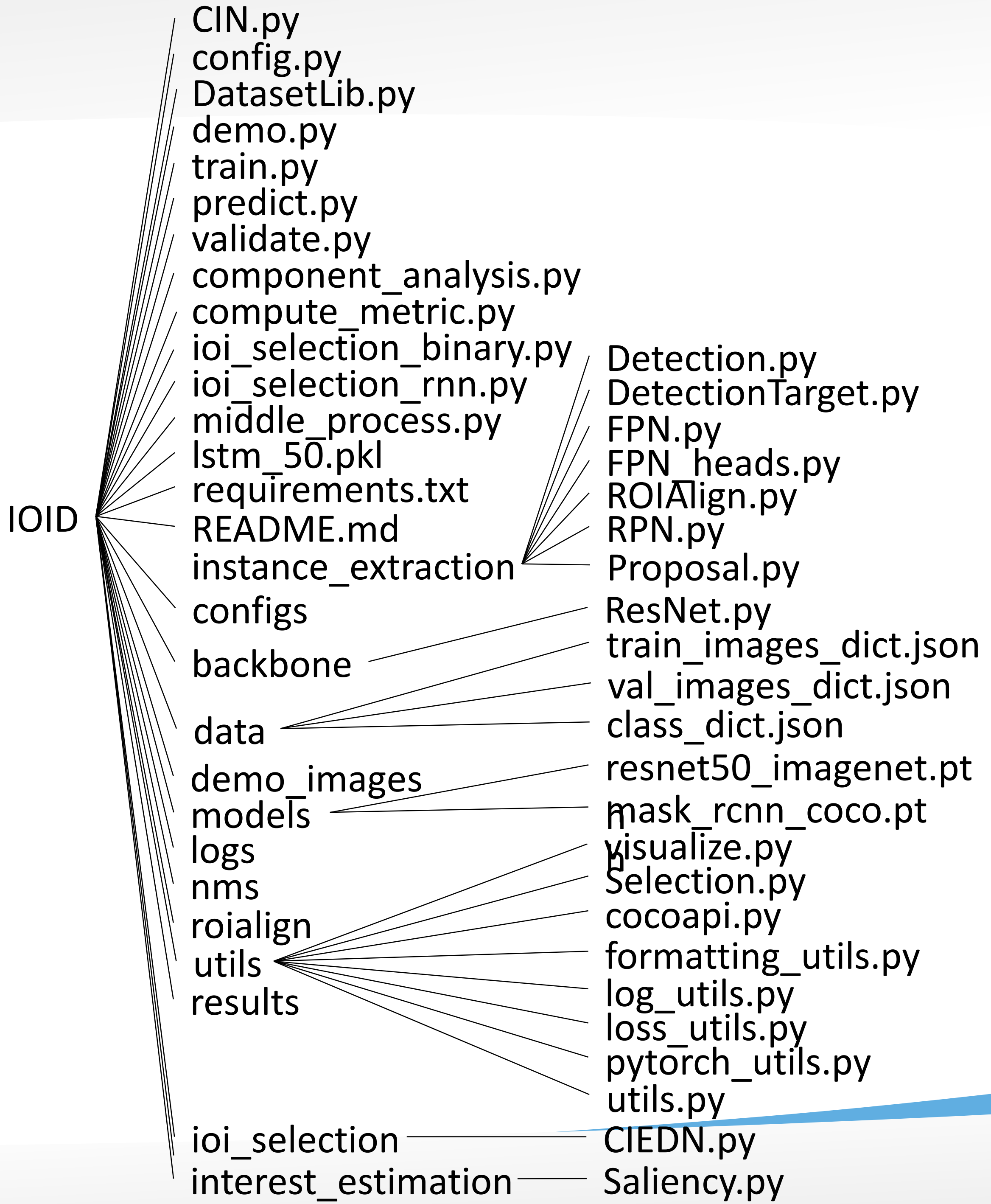
- **data**: containing the json files of the IOID dataset.
- **demo images**: containing some images for testing and visualization.
- **models**: containing some pretrained models.
- **logs**: saving model parameters during training.
- **results**: saving result files generated during testing.
- **utils**: containing python files for assistance.
- **nms**: containing files for non-maximum suppression.
- **roialign**: containing files for region of interest alignment.



IOID
- CIN.py
- config.py
- DatasetLib.py
- demo.py
- train.py
- predict.py
- validate.py
- component_analysis.py
- compute_metric.py
- ioi_selection_binary.py
- ioi_selection_rnn.py
- middle_process.py
- lstm_50.pkl
- requirements.txt
- README.md
- instance_extraction
  - Detection.py
  - DetectionTarget.py
  - FPN.py
  - FPN_heads.py
  - ROIAlign.py
  - RPN.py
  - Proposal.py
- configs
- backbone
  - ResNet.py
- data
  - train_images_dict.json
  - val_images_dict.json
  - class_dict.json
- demo_images
- models
  - resnet50_imagenet.pt
  - mask_rcnn_coco.pt
- logs
- nms
- roialign
- utils
  - visualize.py
  - Selection.py
  - cocoapi.py
  - formatting_utils.py
  - log_utils.py
  - loss_utils.py
  - pytorch_utils.py
  - utils.py
- results
- ioi_selection
  - CIEDN.py
- interest_estimation
  - Saliency.py

Source code structure

- **configs**: containing the configuration files.
- **backbone**: containing the backbone of the CIN model.
- **ioi selection**: containing the python file used in the IOI selection module.
- **interest estimation**: containing the python file used in the interest estimation module.
- **instance extraction**: containing the python files used in the instance extraction module.
- **CIN.py**: working as the main file for the CIN model.
- **config.py**: working as the configuration file with default values.



IOID

CIN.py
config.py
DatasetLib.py
demo.py
train.py
predict.py
validate.py
component_analysis.py
compute_metric.py
ioi_selection_binary.py
ioi_selection_rnn.py
middle_process.py
lstm_50.pkl
requirements.txt
README.md
instance_extraction
configs
backbone
data
demo_images
models
logs
nms
roialign
utils
results
ioi_selection
interest_estimation

Detection.py
DetectionTarget.py
FPN.py
FPN_heads.py
ROIAlign.py
RPN.py
Proposal.py

ResNet.py

train_images_dict.json
val_images_dict.json
class_dict.json

resnet50_imagenet.pth
mask_rcnn_coco.pth

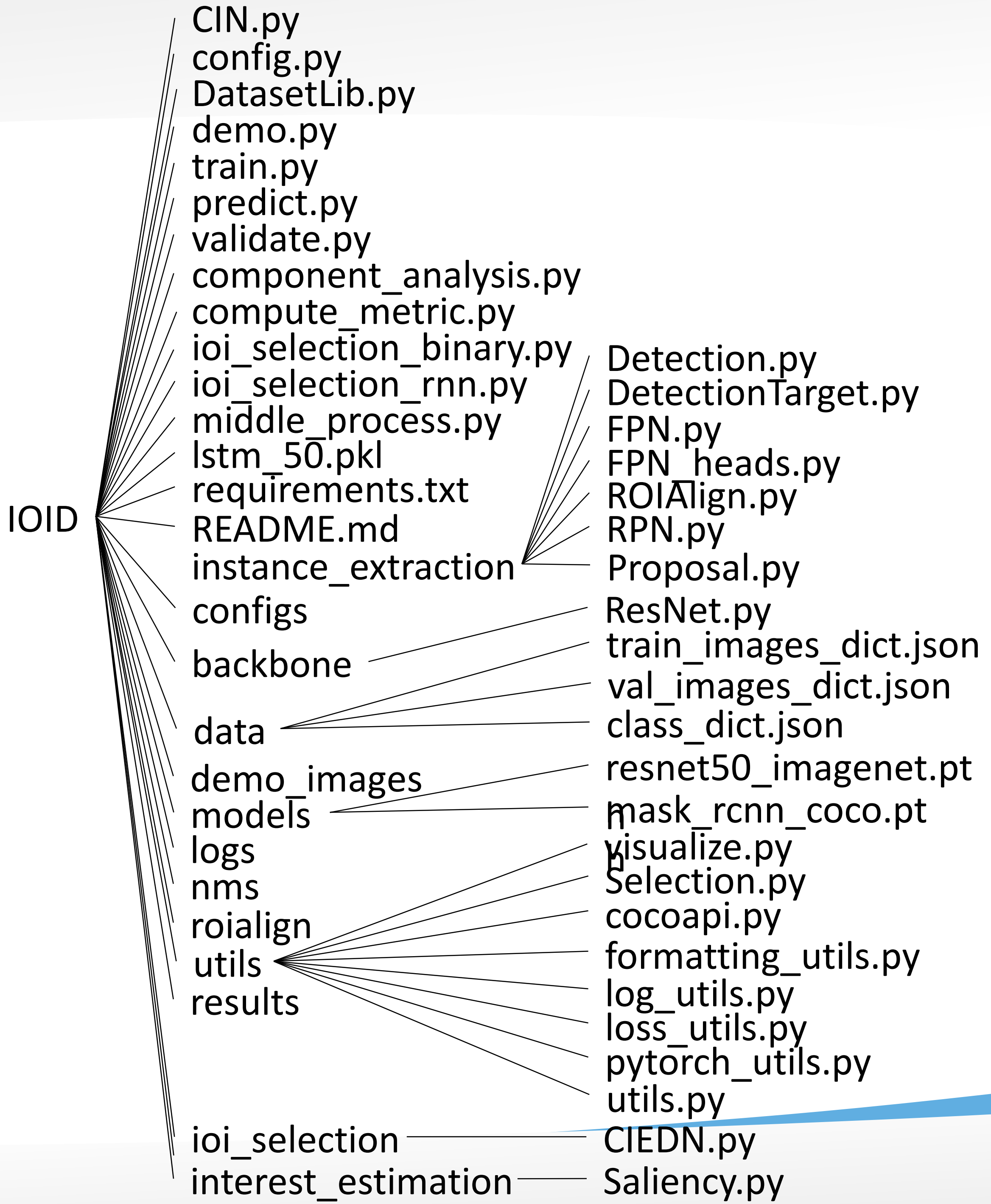
visualize.py
Selection.py
cocoapi.py
formatting_utils.py
log_utils.py
loss_utils.py
pytorch_utils.py
utils.py

CIEDN.py

Saliency.py

Source code structure

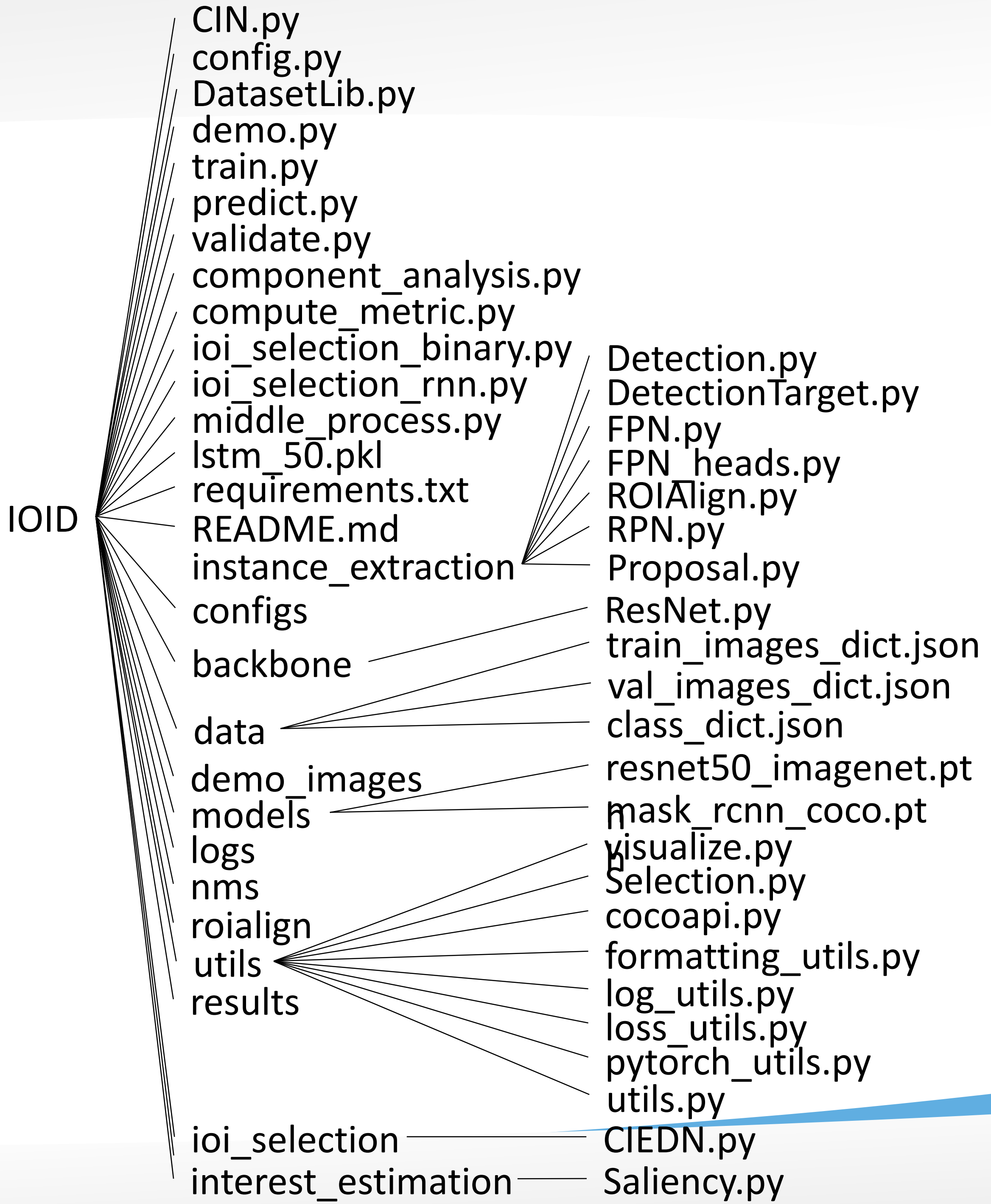
- **datasetLib.py**: loading data for training and validation.
- **demo.py**: working as the main file for testing and visualization.
- **train.py**: working as the main file for training the CIN model.
- **predict.py**: working as the main file for predicting the final or intermediate results based on the CIN model.
- **compute_metric.py**: computing metrics.
- **requirements.txt**: listing the python dependencies of the code.



- IOID
  - CIN.py
  - config.py
  - DatasetLib.py
  - demo.py
  - train.py
  - predict.py
  - validate.py
  - component_analysis.py
  - compute_metric.py
  - ioi_selection_binary.py
  - ioi_selection_rnn.py
  - middle_process.py
  - lstm_50.pkl
  - requirements.txt
  - README.md
  - instance_extraction
    - Detection.py
    - DetectionTarget.py
    - FPN.py
    - FPN_heads.py
    - ROIAlign.py
    - RPN.py
    - Proposal.py
  - configs
  - backbone
    - ResNet.py
  - data
    - train_images_dict.json
    - val_images_dict.json
    - class_dict.json
  - demo_images
  - models
    - resnet50_imagenet.pt
    - mask_rcnn_coco.pt
  - logs
  - nms
  - roialign
  - utils
    - visualize.py
    - Selection.py
    - cocoapi.py
    - formatting_utils.py
    - log_utils.py
    - loss_utils.py
    - pytorch_utils.py
    - utils.py
  - results
  - ioi_selection
    - CIEDN.py
  - interest_estimation
    - Saliency.py

Source code structure

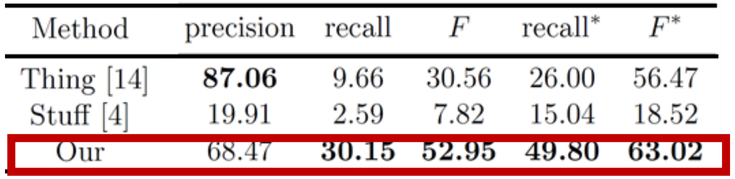
- **validate.py**: working as the main file for evaluating the performance of the CIN model.
- **component_analysis.py**: working as the main file for component analysis.
- **ioi_selection_binary.py**: implementing a simple model as a variant of the IOI selection module in the CIN model.
- **ioi_selection_rnn.py**: implementing an rnn model as a variant of the IOI selection module in the CIN model.
- **middle_process.py**: implementing data processing for component analysis.
- **lstm_50.pkl**: saving the parameters of the pretrained model for the “ioi selection rnn.py”.



IOID
- CIN.py
- config.py
- DatasetLib.py
- demo.py
- train.py
- predict.py
- validate.py
- component_analysis.py
- compute_metric.py
- ioi_selection_binary.py
- ioi_selection_rnn.py
- middle_process.py
- lstm_50.pkl
- requirements.txt
- README.md
- instance_extraction
  - Detection.py
  - DetectionTarget.py
  - FPN.py
  - FPN_heads.py
  - ROIAlign.py
  - RPN.py
  - Proposal.py
- configs
- backbone
  - ResNet.py
- data
  - train_images_dict.json
  - val_images_dict.json
  - class_dict.json
- demo_images
- models
  - resnet50_imagenet.pt
  - mask_rcnn_coco.pt
- logs
- nms
- roialign
- utils
  - visualize.py
  - Selection.py
  - cocoapi.py
  - formatting_utils.py
  - log_utils.py
  - loss_utils.py
  - pytorch_utils.py
  - utils.py
- results
- ioi_selection
  - CIEDN.py
- interest_estimation
  - Saliency.py

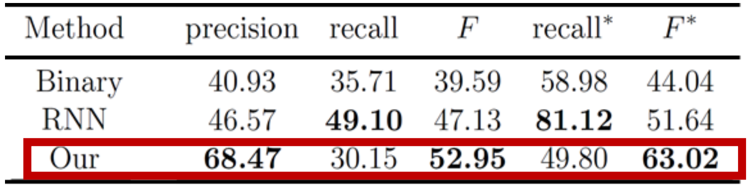
Experiments

- Evaluation criteria: precision, recall, *F*, recall*, *F**



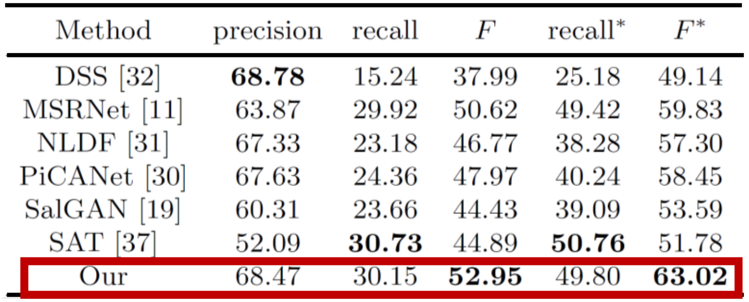
Method	precision	recall	F	recall*	F^*
Thing [14]	**87.06**	9.66	30.56	26.00	56.47
Stuff [4]	19.91	2.59	7.82	15.04	18.52
Our	68.47	**30.15**	**52.95**	**49.80**	**63.02**

Different instance extraction



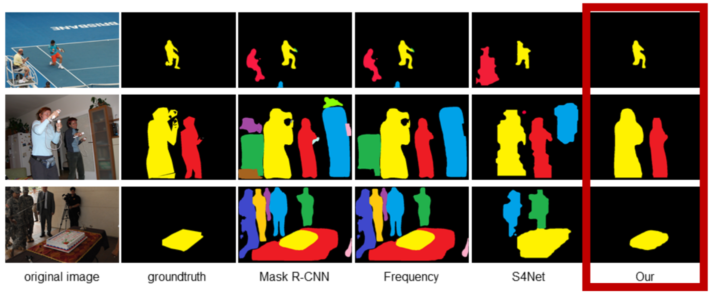
Method	precision	recall	F	recall*	F^*
Binary	40.93	35.71	39.59	58.98	44.04
RNN	46.57	**49.10**	47.13	**81.12**	51.64
Our	**68.47**	30.15	**52.95**	49.80	**63.02**

Different interest estimation

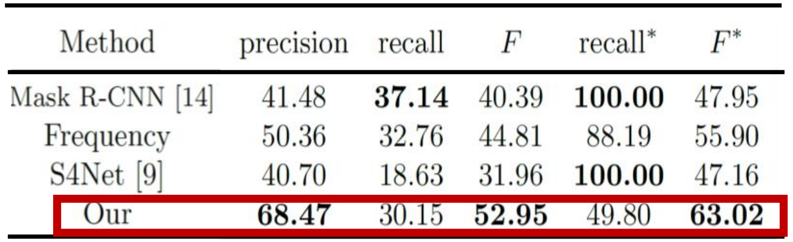


Method	precision	recall	F	recall*	F^*
DSS [32]	**68.78**	15.24	37.99	25.18	49.14
MSRNet [11]	63.87	29.92	50.62	49.42	59.83
NLDF [31]	67.33	23.18	46.77	38.28	57.30
PiCANet [30]	67.63	24.36	47.97	40.24	58.45
SalGAN [19]	60.31	23.66	44.43	39.09	53.59
SAT [37]	52.09	**30.73**	44.89	**50.76**	51.78
Our	68.47	30.15	**52.95**	49.80	**63.02**

Different IOI selection



Qualitative examples

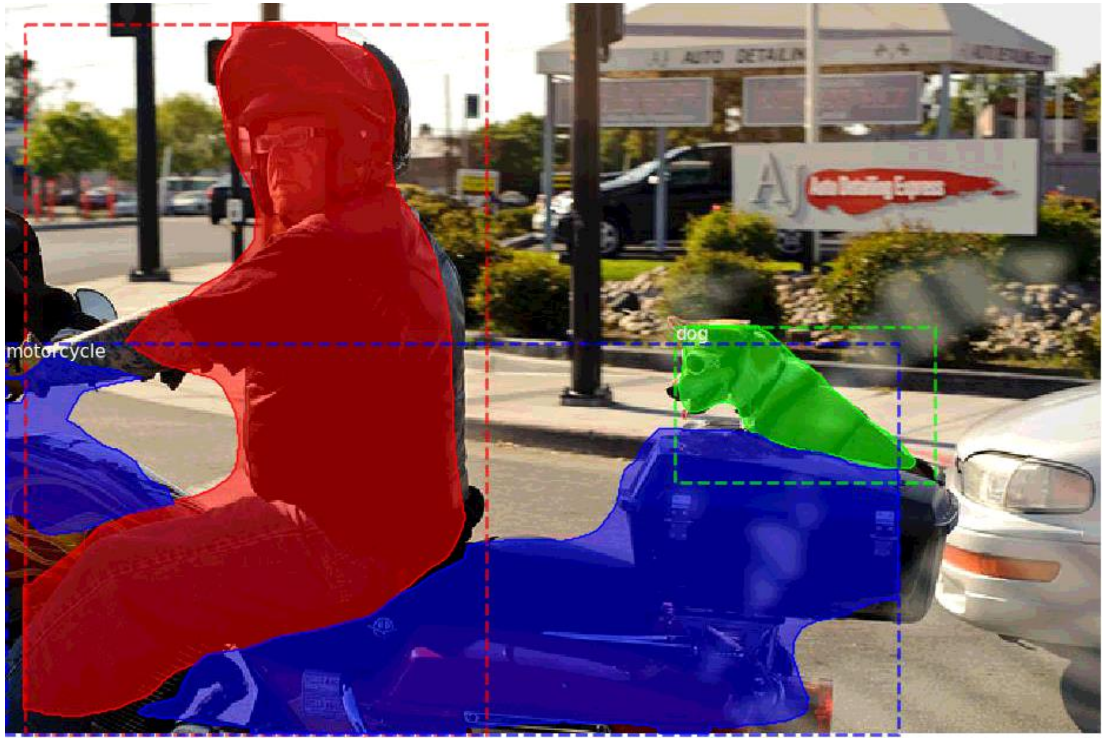


Method	precision	recall	F	recall*	F^*
Mask R-CNN [14]	41.48	**37.14**	40.39	**100.00**	47.95
Frequency	50.36	32.76	44.81	88.19	55.90
S4Net [9]	40.70	18.63	31.96	**100.00**	47.16
Our	**68.47**	30.15	**52.95**	49.80	**63.02**

Overall

Experiments

- Environment
  - Operating system Ubuntu 16.04 LTS with CPU i7-8086K, GPU TITAN V, 64GB memory and 1TB free space.
  - CUDA 9.0 and cuDNN 7.0.
  - Python 3.5.6 with opencv python==3.4.3.18, numpy==1.16.2, scikit image==0.14.2, torchvision==0.2.1, torch==0.4.1, scipy==1.1.0, matplotlib==3.0.0, Pillow==7.0.0, skimage==0.0, tensorboardX==2.0, PyYAML==3.13 and cffi==1.12.2.



- Demo
  - python demo.py −−img <image path> −−config <configuration file path>

Experiments

- Train

  -
    ```
    python train.py −−setting <setting sequence> −−config <configuration file path>
    ```
- Predict

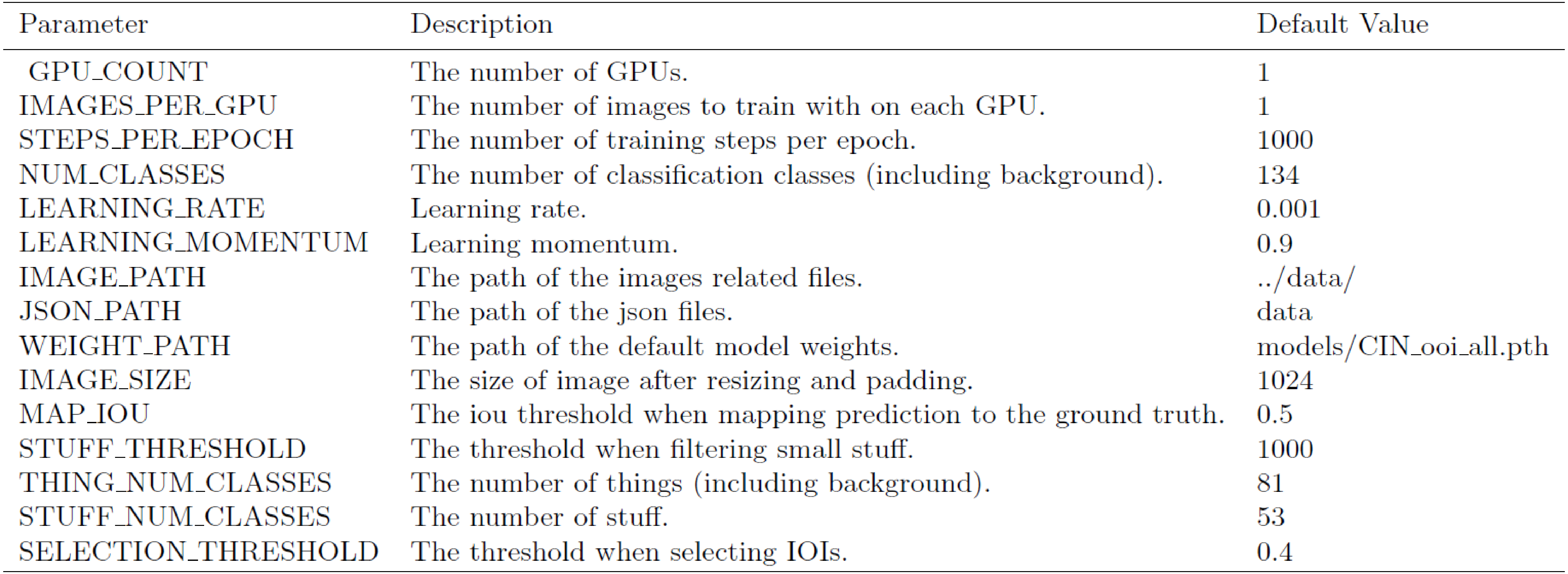
  -
    ```
    python predict.py −−mode <mode> −−subset <performing on which dataset> −−config <configuration file path>
    ```
- Evaluate

  -
    ```
    python validate.py −−config <configuration file path>
    ```
- Component analysis

  -
    ```
    python component analysis.py −−ins_ext <panoptic segmentation path> −−sem_ext <semantic segmentat ion path>
    −−p_intr_ext <interestestimation path> −−sel_ext <IOI selection method> −−config <configuration file path>
    ```

Experiments

- Important parameters that can be customized



Parameter	Description	Default Value
GPU_COUNT	The number of GPUs.	1
IMAGES_PER_GPU	The number of images to train with on each GPU.	1
STEPS_PER_EPOCH	The number of training steps per epoch.	1000
NUM_CLASSES	The number of classification classes (including background).	134
LEARNING_RATE	Learning rate.	0.001
LEARNING_MOMENTUM	Learning momentum.	0.9
IMAGE_PATH	The path of the images related files.	../data/
JSON_PATH	The path of the json files.	data
WEIGHT_PATH	The path of the default model weights.	models/CIN_ooi_all.pth
IMAGE_SIZE	The size of image after resizing and padding.	1024
MAP_IOU	The iou threshold when mapping prediction to the ground truth.	0.5
STUFF_THRESHOLD	The threshold when filtering small stuff.	1000
THING_NUM_CLASSES	The number of things (including background).	81
STUFF_NUM_CLASSES	The number of stuff.	53
SELECTION_THRESHOLD	The threshold when selecting IOIs.	0.4

THANK YOU

MAGUS

MediA recoGnition
and UnderStanding