

# Automatic Segmentation of COVID-19 CT Images Based on MIScnn

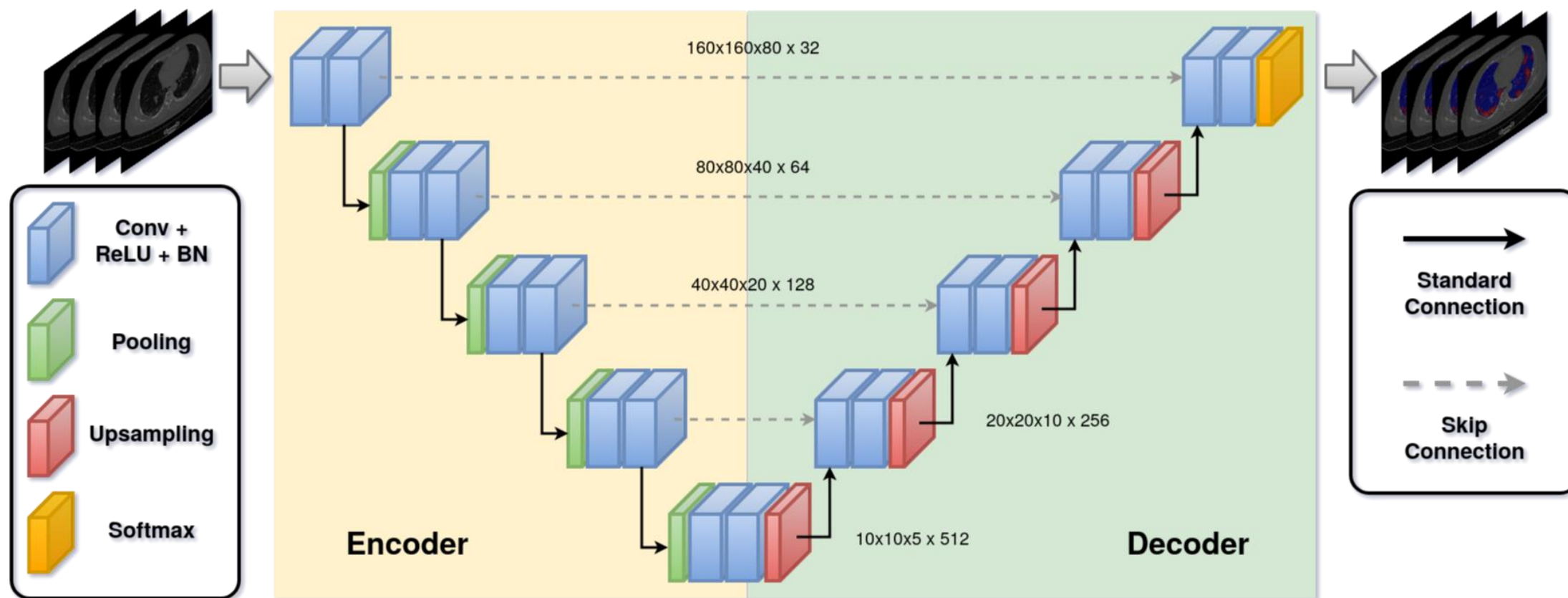
By Chen ZHU, Heng GU



# 1. Introduction

- Our segmentation approach is based on 3D U-Net and MlScnn.
- The method focused on the on-the-fly generation of unique and random image patches for training by exploiting heavy preprocessing and extensive data augmentation.
- Thus, it is possible to handle limited dataset size which acts as a variant database.
- We adopt the standard 3D U-Net.

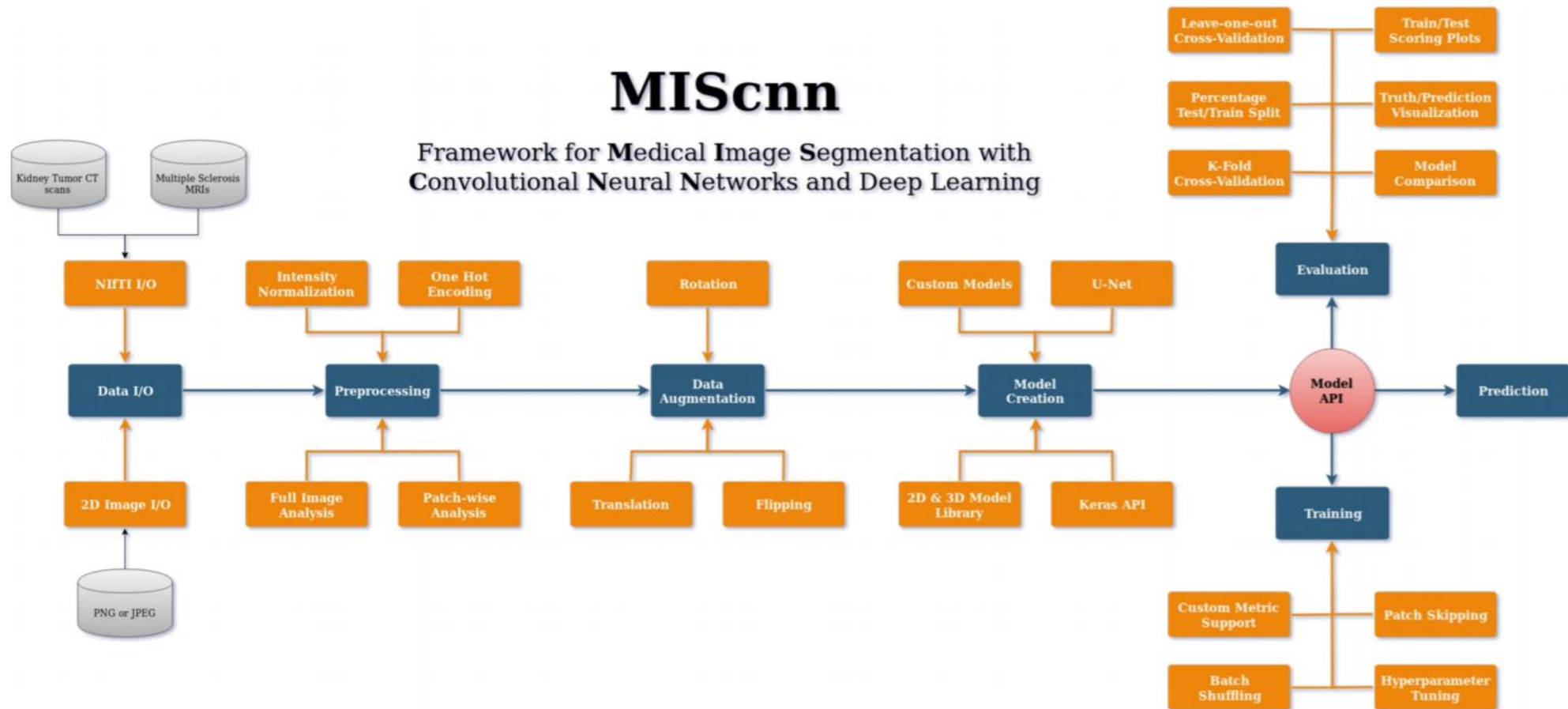
## 2. Method



## 2. Method

- The pipeline is based on MIScnn, which is an in-house developed open-source framework to setup complete medical image segmentation pipelines with convolutional neural networks and deep learning models on top of TensorFlow/Keras.
- The framework supports extensive preprocessing, data augmentation, the latest learning models, and diverse evaluation techniques.

### 3. MIScnn



# 3. MIScnn



- MIScnn provides several core features:
  - 2D/3D medical image segmentation for binary and multi-class problems
  - Data I/O, preprocessing and data augmentation for biomedical images
  - Patch-wise and full image analysis
  - State-of-the-art deep learning model and metric library
  - Intuitive and fast model utilization (training, prediction)
  - Multiple automatic evaluation techniques (e.g. cross-validation)
  - Custom model, data I/O, pre-/postprocessing and metric support
  - Based on Keras with Tensorflow as backend

## 4. Data Preprocessing

- Download Data
- Download CT volumes and segmentations

```
path_data = "data"
# Links to the data set
url_vol = "https://zenodo.org/record/3757476/files/COVID-19-CT-Seg_20cases.zip?download=1"
url_seg = "https://zenodo.org/record/3757476/files/Lung_and_Infection_Mask.zip?download=1"
```

- Extract the sample list from the ZIP file

```
print("INFO:", "Obtain sample list from the volumes ZIP file")
with zipfile.ZipFile(path_vol_zip, "r") as zip_vol:
    sample_list = zip_vol.namelist()
```

- Extract each sample from the ZIP files

```
with zipfile.ZipFile(path_vol_zip, "r") as zip_vol:
    zip_vol.extract(sample, path_sample)
```

# 5. Data Exploration

- Obtain the volume shape and the minimum and maximum volume intensity
- Store the voxel spacing, identify and category distribution
- The mean and median of the shape, the average category frequency

	vol_shape	vol_minimum	vol_maximum	\
coronacases_001	(512, 512, 301, 1)	-1021	2996	
coronacases_002	(512, 512, 200, 1)	-1023	9567	
coronacases_003	(512, 512, 200, 1)	-1023	8931	
coronacases_004	(512, 512, 270, 1)	-1021	2020	
coronacases_005	(512, 512, 290, 1)	-1021	5528	
coronacases_006	(512, 512, 213, 1)	-1023	2217	
coronacases_007	(512, 512, 249, 1)	-1023	2515	
coronacases_008	(512, 512, 301, 1)	-1021	8575	
coronacases_009	(512, 512, 256, 1)	-1021	1845	
coronacases_010	(512, 512, 301, 1)	-1021	1920	
radiopaedia_10_85902_1	(630, 630, 39, 1)	0	255	
radiopaedia_10_85902_3	(630, 630, 418, 1)	0	255	
radiopaedia_14_85914_0	(630, 401, 110, 1)	0	255	
radiopaedia_27_86410_0	(630, 630, 66, 1)	4	255	
radiopaedia_29_86490_1	(630, 630, 42, 1)	0	255	
radiopaedia_29_86491_1	(630, 630, 42, 1)	0	255	
radiopaedia_36_86526_0	(630, 630, 45, 1)	0	255	
radiopaedia_40_86625_0	(630, 630, 93, 1)	12	255	
radiopaedia_4_85506_1	(630, 630, 39, 1)	0	255	
radiopaedia_7_85703_0	(630, 630, 45, 1)	0	255	

```

X-Axes Mean: 571.0
X-Axes Median: 571.0
Y-Axes Mean: 559.55
Y-Axes Median: 512.0
Z-Axes Mean: 176.0
Z-Axes Median: 200.0
background      0.896091
lung_L          0.043423
lung_R          0.049364
infection       0.011122
dtype: float64

```



## 6. Sample Processing

- Divide the sample into four categories: background, left lung, right lung, and COVID-19

```
interface = NIFTI_interface(channels=1, classes=4)|  
data_io = Data_IO(interface, input_path="data", delete_batchDir=False)
```

- Divide the sample into five folds

```
sample_list = data_io.get_indiceslist()  
sample_list.sort()  
split_folds(sample_list, k_fold=5)
```

## 7. Neural Network Training

- Use 4 classifications to train the neural network
- Set the epoch of the neural network to 15
- Set the number of iterations to 150

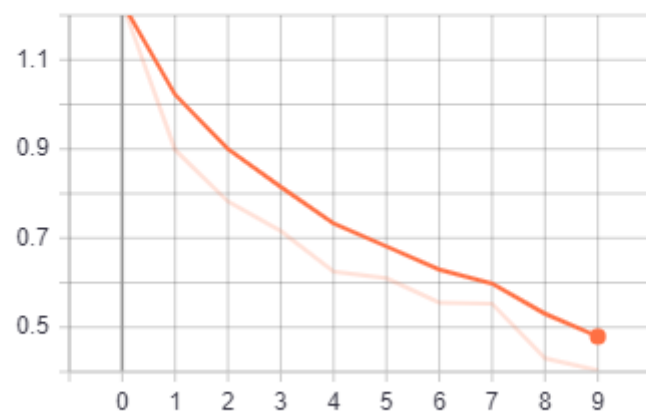
```
run_fold(fold, model, epochs=10, iterations=150, evaluation_path="evaluation",  
        draw_figures=True, callbacks=[cb_lr, cb_es, cb_tb, cb_cl],  
        save_models=False)
```

```
model = Neural_Network(preprocessor=pp, architecture=unet_standard,  
                        loss=tversky_crossentropy,  
                        metrics=[tversky_loss, dice_soft, dice_crossentropy],  
                        batch_queue_size=3, workers=1, learning_rate=0.0001)
```

# 8. Result

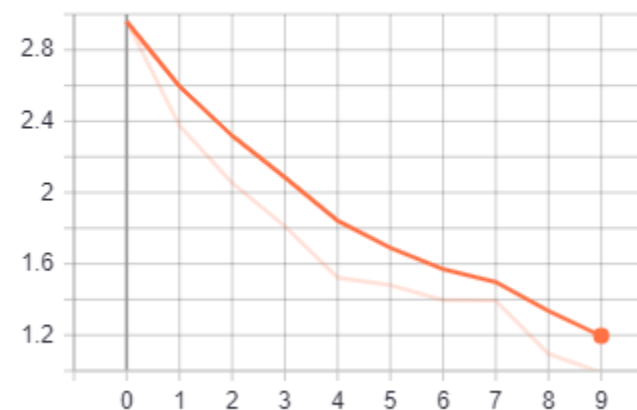
epoch\_dice\_crossentropy

epoch\_dice\_crossentropy



epoch\_loss

epoch\_loss



# References

- [1] Dominik Müller and Frank Kramer. (2019). MIScnn: A Framework for Medical Image Segmentation with Convolutional Neural Networks and Deep Learning.
- [2] Jun Ma, Cheng Ge, Yixin Wang, Xingle An, Jiantao Gao, Ziqi Yu, Mingqing Zhang, Xin Liu, Xueyuan Deng, Shucheng Cao, Hao Wei, Sen Mei, Xiaoyu Yang, Ziwei Nie, Chen Li, Lu Tian, Yuntao Zhu, Qiongjie Zhu, Guoqiang Dong, Jian He. (2020). COVID-19 CT Lung and Infection Segmentation Dataset (Version 1.0) [Data set]. Zenodo.