



Status report on **DNNROI sigproc**

Hokyeong Nam
Chung-Ang University

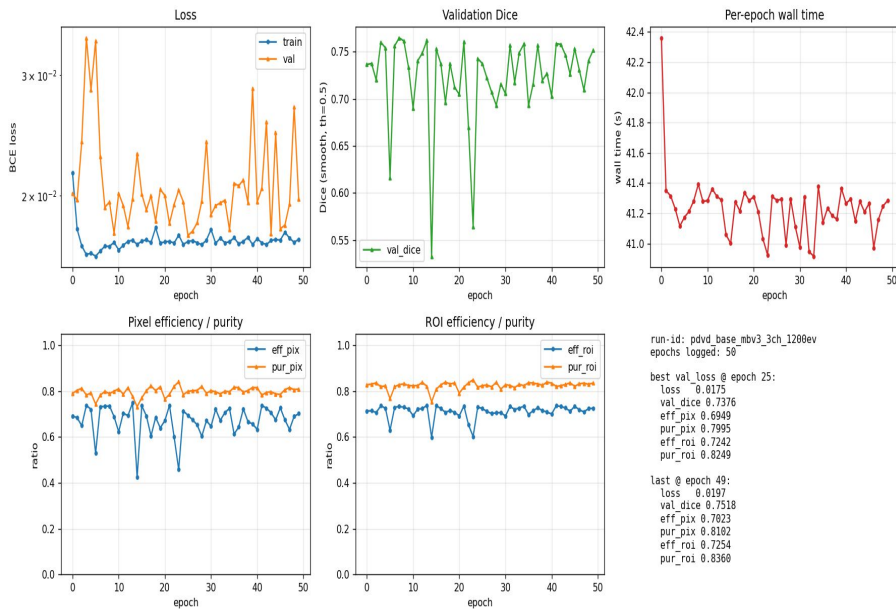
Baseline Model Training

- Training dataset:
/nfs/data/1/xqian/toolkit-dev/DNN_ROI_SP/data/pdvd/processed/pdvd_trainval_1200ev_6ch.h5
- 6 models are trained on PD-VD training samples (TDE+BDE, 1200 events)
 - 3ch inputs: MobileU-Net, Transformer, NestedUNet
 - 6ch inputs: MobileU-Net, Transformer, NestedUNet
- Best checkpoints will be used as teacher for the next KD training
- My goal is to reproduce Xin's result 
 - M1: TDE samples (200 events)
 - M2: BDE samples (200 events)
 - M3: TDE + BDE samples (200 events)

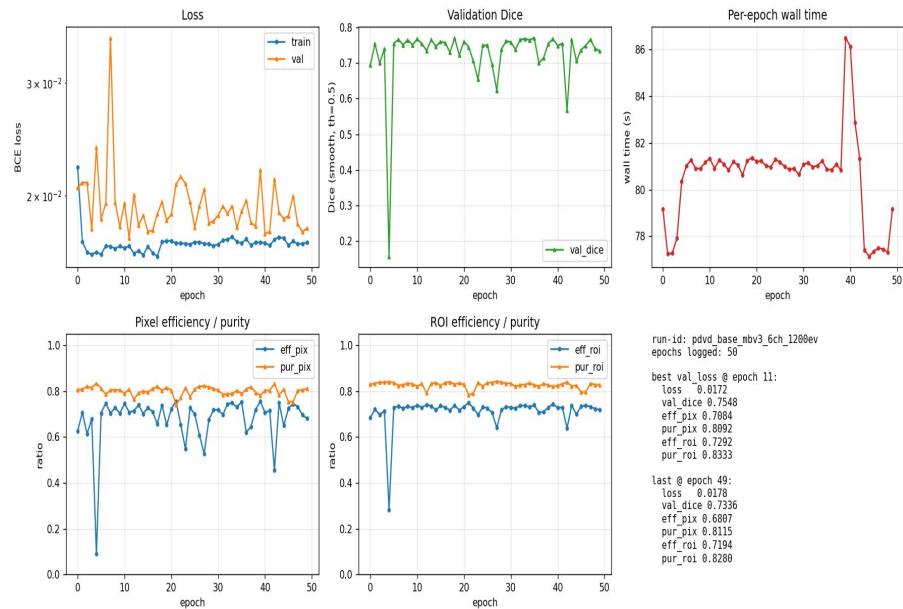
model	test set	n	eff_pix	pur_pix	eff_roi	pur_roi	dice
M1	top	120	0.7631	0.8618	0.7543	0.8967	0.8089
M1	bot	120	0.7281	0.8033	0.7395	0.8291	0.7636
M2	top	120	0.7141	0.8629	0.7372	0.8911	0.7793
M2	bot	120	0.6453	0.8137	0.7034	0.8285	0.7116
M3	top	120	0.7638	0.8579	0.7607	0.8831	0.8077
M3	bot	120	0.7199	0.8047	0.7397	0.8221	0.7596

Baseline Model Training

Training curves — pdvd_base_mbv3_3ch_1200ev



Training curves — pdvd_base_mbv3_6ch_1200ev



Baseline Model Training

Model	Input Channels	Test Set	Loss	Eff_pix	Pur_pix	Eff_roi	Pur_roi	Dice
MobileNetV3	3	Top	0.0213	0.6314	0.7430	0.6634	0.7681	0.6759
MobileNetV3	3	Bottom	0.0128	0.7326	0.8085	0.7443	0.8237	0.7657
MobileNetV3	6	Top	0.0212	0.6533	0.7505	0.6717	0.7772	0.6976
MobileNetV3	6	Bottom	0.0123	0.7225	0.8222	0.7450	0.8315	0.7686
Nested U-Net	3	Top	0.0194	0.6769	0.7501	0.6787	0.7695	0.7084
Nested U-Net	3	Bottom	0.0107	0.7771	0.8145	0.7646	0.8277	0.7937
Transformer	3	Top	0.0198	0.6092	0.7653	0.6436	0.7760	0.6625
Transformer	3	Bottom	0.0107	0.7345	0.8282	0.7453	0.8339	0.7736

model	test set	n	eff_pix	pur_pix	eff_roi	pur_roi	dice
M3	top	120	0.7638	0.8579	0.7607	0.8831	0.8077
M3	bot	120	0.7199	0.8047	0.7397	0.8221	0.7596

- Top (CRP 3, 4), Bottom (CRP 1, 2)
- The performance was found to be degraded

- Since I do not have the M3 dataset, I plan to train the model using the M1 and M2 files, check whether the numbers are consistent there

TorchScript Conversion

- `torch.jit.script`
 - Compiles model code, including control flow
 - More flexible for different input image sizes
 - Requires TorchScript-compatible model code
- `torch.jit.trace`
 - Records operations from one example input
 - Easier to export when script fails
 - Can become input-shape specific
- Current issues
 - The updated MobileU-Net model file violates `torch.jit.script`'s restrictions
 - The traced model may not generalize safely to different image sizes
 - Transformer model is only compatible with `torch.jit.script` for now

```
def _collect_encoder_features(self, x_rgb):
    """
    Run the encoder and collect outputs at the pre-recorded indices (self._feat_indices).
    Returns a list [f1 (shallow), f2, f3, f4, f5 (deep)] of as many features as exist (typical
    Stops once the deepest required feature is collected – trailing
    layers (indices > max(_feat_indices)) produce no decoder input, so
    running them is wasted compute and leaves their parameters with
    zero gradient (which trips DDP's reducer).
    """
    feats = []
    last_idx = self._feat_indices[-1] if self._feat_indices else -1
    for i, m in enumerate(self.encoder):
        x_rgb = m(x_rgb)
        if i in self._feat_indices:
            feats.append(x_rgb)
        if i >= last_idx:
            break
    return feats
```

Back Up

Following up Xin's progress

- Lots of progress made by Xin for DNN ROI studies including new techniques and bug fix
- All the scripts and documents are in private git repo: https://github.com/lastgeorge/DNN_ROI_SP
- Key changes
 - Create pre-processed training data separately from DNN training
 - Flexible DNN inputs: [loose_lf, mp2_roi, mp3_roi] + [**tight_lf, gauss, decon_charge**]
 - New techniques: Knowledge Distillation (KD), Quantization-Aware Training (QAT), INT8
 - Field response file for PD-VD is updated: protodunevd_FR_imbalance3p_260501.json.bz2
 - For PD-VD, training samples now include TDE data
 - DNN ROI threshold is changed from 0.5 to 0.2
 - Wire-Cell-Toolkit now supports DNNROIFindingMultiPlane
- Notes
 - Training time decreased (pre-processed training sample + upload full sample to CPU memory)
 - 6ch input + new techniques showed better performance than baseline (3ch + MobileU-Net V3) in terms of ML metrics
 - PD-VD model trained on BDE + TDE showed better performance than baseline (BDE only)

Following up Xin's progress

- Additional bug fix required for sp-filters.jsonnet in cfg/experiment/protodunevd/

```
97      hf('Wiener_tight_U_b', { sigma: 0.148788 * wc.megahertz, power: 3.76194 }),
98      hf('Wiener_tight_U_t', { sigma: 0.148788 * wc.megahertz, power: 3.76194 }),
99      hf('Wiener_tight_V_b', { sigma: 0.1596568 * wc.megahertz, power: 4.36125 }),
100     hf('Wiener_tight_V_t', { sigma: 0.1596568 * wc.megahertz, power: 4.36125 }),
101     hf('Wiener_tight_W_b', { sigma: 0.13623 * wc.megahertz, power: 3.35324 }),
102     hf('Wiener_tight_W_t', { sigma: 0.13623 * wc.megahertz, power: 3.35324 }),
```

- Parameters are from PD-SP
- Optimized filter parameters for PD-VD BDE
- Which one is used for recent training samples?



```
hf('Wiener_tight_U', {
  sigma: 0.286735 * wc.megahertz,
  power: 8.07074,
}),
hf("Wiener_tight_V", {
  sigma: 0.289041 * wc.megahertz,
  power: 9.94129 }),
hf('Wiener_tight_W', {
  sigma: 0.207127 * wc.megahertz,
  power: 4.13956,
}),
```

Following up Xin's progress

- Trained baseline models with different inputs 3ch/6ch and converted into TorchScript files
- Training data: /nfs/data/1/xqian/toolkit-dev/DNN_ROI_SP/data/pdvd/processed/pdvd_trainval_1200ev_6ch.h5
- Updated to_torchscript.py to support input-channels (e.g. 3 or 4 or 5 or 6)

```
[ckpt] checkpoints/pdvd_base_mbv3_3ch_1200ev/CP25.pth (epoch 25)
[data] test slice [960, 1200) x=(240, 3, 952, 1600) y=(240, 952, 1600)
[model] mobilenetv3 input_channels=3
[eval] all n= 240 loss=0.0175 dice=0.7376 eff_pix=0.6949 pur_pix=0.7995 eff_roi=0.7242 pur_roi=0.8249
[eval] A (crp1,crp2) n= 60 loss=0.0128 dice=0.7657 eff_pix=0.7326 pur_pix=0.8085 eff_roi=0.7443 pur_roi=0.8237
[eval] B (crp3,crp4) n= 60 loss=0.0213 dice=0.6759 eff_pix=0.6314 pur_pix=0.7430 eff_roi=0.6634 pur_roi=0.7681
```

metric	all	A (crp1,crp2)	B (crp3,crp4)
loss	0.0175	0.0128	0.0213
dice	0.7376	0.7657	0.6759
eff_pix	0.6949	0.7326	0.6314
pur_pix	0.7995	0.8085	0.7430
eff_roi	0.7242	0.7443	0.6634
pur_roi	0.8249	0.8237	0.7681
n	240	60	60

```
[ckpt] checkpoints/pdvd_base_mbv3_6ch_1200ev/CP11.pth (epoch 11)
[data] test slice [960, 1200) x=(240, 6, 952, 1600) y=(240, 952, 1600)
[model] mobilenetv3 input_channels=6
[eval] all n= 240 loss=0.0172 dice=0.7548 eff_pix=0.7084 pur_pix=0.8092 eff_roi=0.7292 pur_roi=0.8333
[eval] A (crp1,crp2) n= 60 loss=0.0123 dice=0.7686 eff_pix=0.7225 pur_pix=0.8222 eff_roi=0.7450 pur_roi=0.8315
[eval] B (crp3,crp4) n= 60 loss=0.0212 dice=0.6976 eff_pix=0.6533 pur_pix=0.7505 eff_roi=0.6717 pur_roi=0.7772
```

metric	all	A (crp1,crp2)	B (crp3,crp4)
loss	0.0172	0.0123	0.0212
dice	0.7548	0.7686	0.6976
eff_pix	0.7084	0.7225	0.6533
pur_pix	0.8092	0.8222	0.7505
eff_roi	0.7292	0.7450	0.6717
pur_roi	0.8333	0.8315	0.7772
n	240	60	60