ICPR VideoPipe Challenge - Track on Video Defect Classification

Xu Fang
College of Electronics and Information
Engineering, Shenzhen University
Shenzhen, China
Email: fangxu622@126.com

***Abstract*—The abstract goes here.**

I. Method

The Video Defect Classification framework is based on video swin transformer[1], The Swin-Base backbone is adopted. The overall architecture is shown figure 1.

Class balanced focal loss[2] is used to solve the class imbalanced problem. And the class balanced focal loss is show as equation (1), (2)

$$\mathbf{CB}_{\text{focal}}\left(\mathbf{z}, y\right) = -\frac{1-\beta}{1-\beta^{n_y}} \sum_{i=1}^{C} \left(1-p_i^t\right)^{\gamma} \log\left(p_i^t\right) \quad (1)$$

$$p_i^t = \text{sigmoid}\left(z_i^t\right) \quad (2)$$

where β controls the per class loss weight(Default: 0.9999). γ is hyperparameter of the focal loss(Default: 2.0). z_i predicted output from the model. n_y is the number of samples per class.

II. Framework

The implementation of video swin transformer is based on MMAction2 project. The config file of this work could be found at https://github.com/fangxu622/videopipe-icpr-top2. config file shows all detailed settings of the model.

III. Dataset usage

The sample frames of every video was 12. The interval of frame is 5, the data augmentation includes random scale, crop, flip. All the annotation is used to train the model. The weight decay is 0.05.

IV. Train process

The optimizer is AdamW. The learning rate is 0.0002. The learning rate policy is CosineAnnealing. Two tesla T4 GPUs was use to train the model. And fp16 (Half-precision) is used for accelerating the train speed.

References

[1] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," *arXiv preprint arXiv:2106.13230*, 2021.

[2] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9268–9277.

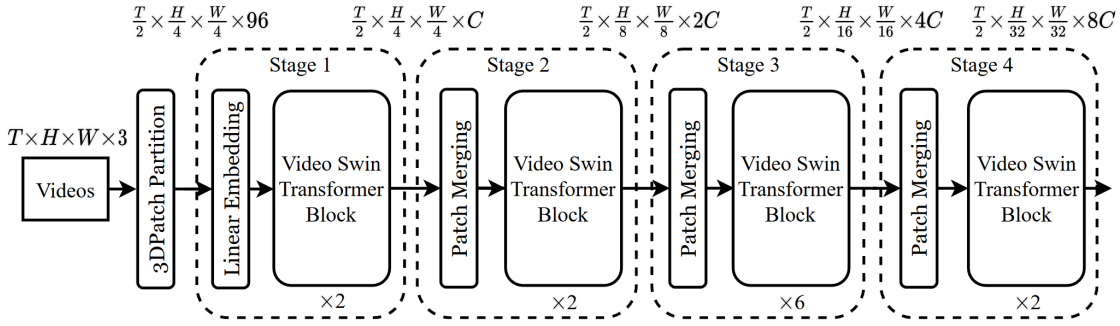

Fig. 1: Overall architecture of Video Swin Transformer

Track on Video Defect Classification

FangXu

1. Method

The method and framework is based on Video Swin Transformer. Swin-Base is adopted.

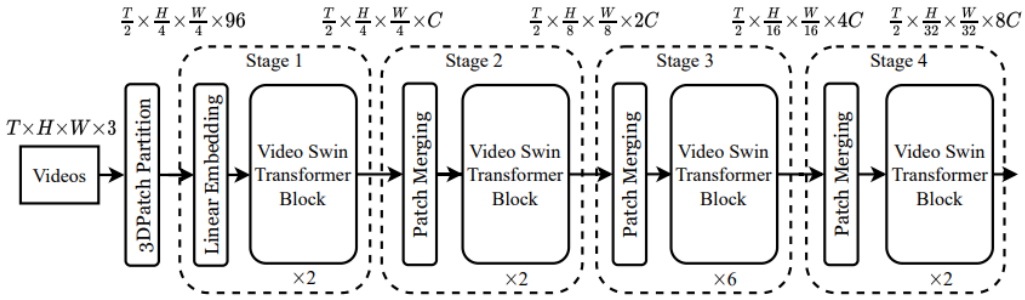

Figure 1: Overall architecture of Video Swin Transformer (tiny version, referred to as Swin-T).

2. Framework

1. The implementation of **video swin transformer** is based on MMAction2 project.

2. **Class balanced Focal loss** is used to solve the class imbalanced problem.

3. Dataset usage

The sample frames of every video was 12. The train data pipeline config part was as follow.

All the annotation is used to train the model.

```
train_pipeline = [
    dict(type='SampleFrames', clip_len=12, frame_interval=5, num_clips=1),
    dict(type='RawFrameDecode'),
    dict(type='RandomRescale', scale_range=(256, 320)),
    dict(type='RandomResizedCrop'),
    dict(type='Resize', scale=(224, 224), keep_ratio=False),
    dict(type='Flip', flip_ratio=0.5),
    dict(
        type='Normalize',
        mean=[123.675, 116.28, 103.53],
        std=[58.395, 57.12, 57.375],
        to_bgr=False),
```

```
    dict(type='FormatShape', input_format='NCTHW'),
    dict(type='Collect', keys=['imgs', 'label'], meta_keys=[]),
    dict(type='ToTensor', keys=['imgs', 'label'])
]
```

4. Train process

Two tesla T4 GPUs was use to train the model. And fp16 (Half-precision) is adopted for accelerating the train speed.

the train process config part is as follow.

```
optimizer = dict(
    type='AdamW',
    lr=0.0002,
    betas=(0.9, 0.999),
    weight_decay=0.05,
    paramwise_cfg=dict(
        custom_keys=dict(
            absolute_pos_embed=dict(decay_mult=0.0),
            relative_position_bias_table=dict(decay_mult=0.0),
            norm=dict(decay_mult=0.0),
            backbone=dict(lr_mult=0.1))))
optimizer_config = dict(grad_clip=dict(max_norm=10.0, norm_type=2))
lr_config = dict(
    policy='CosineAnnealing',
    min_lr=0,
    warmup='linear',
    warmup_by_epoch=True,
    warmup_iters=2.5)
total_epochs = 40
```

5. inference processing

Data processing config part in the inference phase is as follow.

```
    test=dict(
        type='QvPipeDataset',
        ann_file='/home/fangxu/det-prj/qvpipe_test.json',
        data_prefix='/home/fangxu/data/QvPipeData',
        pipeline=[
            dict(
```

```
                type='SampleFrames',
                clip_len=12,
                frame_interval=5,
                num_clips=1,
                test_mode=True),
            dict(type='RawFrameDecode'),
            dict(type='Resize', scale=(-1, 224)),
            dict(type='CenterCrop', crop_size=224),
            dict(type='Flip', flip_ratio=0),
            dict(
                type='Normalize',
                mean=[123.675, 116.28, 103.53],
                std=[58.395, 57.12, 57.375],
                to_bgr=False),
            dict(type='FormatShape', input_format='NCTHW'),
            dict(type='Collect', keys=['imgs', 'label'], meta_keys=[
    ]),
            dict(type='ToTensor', keys=['imgs'])
        ]))
```