ECCV 2022 Seasons in Drift Challenge

Fact sheet

This is the fact sheet's template for the ECCV 2022 Seasons in Drift Challenge. Please fill out the following sections carefully in a scientific writing style. Then, send the compressed project (in `.zip` format), i.e., the generated PDF, `.tex`, `.bib` and any additional files, following the schedule and instructions ("Wining solutions (post-challenge)", Fact Sheets) provided in the Challenge webpage.

I. Team details

- Team leader name: **Boyong He**
- Username on Codalab: **heboyong**
- Team leader affiliation: **Xiamen University**
- Team leader email: **heboyong@qq.com**
- Name of other team members (and affiliation):
  (1) **Qianwen Ye, Xiamen University**
  (2) **Xianjiang Li, Xiamen University**
  (3) **Weijie Guo, Xiamen University**
- Team website URL (if any): None
- Competition track (mark with X one single option)[1]:
  - ( ) Track 1: Detection at **day** level.
  - (X) Track 2: Detection at **week** level.
  - ( ) Track 3: Detection at **month** level.

II. Contribution details

A. Title of the contribution

In this competition, we use the current SOTA algorithms (Cascade RCNN [1], Swin Transformer [2], CBNetv2 [3]) and targeted data augmentation methods (Large Scale Jitter, Random Crop, MixUp [4], Albu Augmentation [5], CopyPaste [6]) to mitigate the drift problem in the LTD dataset [7] caused by changes in the outdoor environment and significantly improves the detection accuracy of the week level training dataset.

B. Representative image / workflow diagram of the method

As shown in 1

C. Detailed method description

In this competition, we use the MMdetection [8] as the main framework for the verification and development of the algorithms.

The structures of our algorithm include:

- **Cascade RCNN** [1]: We use Cascade RCNN, a two-stage object detection algorithm with high accuracy, as the main architecture for object detection.
- **Swin Transformer** [2]: We use the SOTA transformer structure Swin Transformer (Swin-S) as the backbone. Swin Transformer always achieves better results when comparing with other CNN-based backbone.
- **CBNetv2** [3]: We use CBNetv2 to enhance the Swin Transformer to directly improve the accuracy of the current backbone without retraining the backbone.

Data augmentation, inference and post-processing settings include:

- **Data sampling**: Considering the image continuity and repetition in the dataset, we only randomly sampled 30% of the image data from the original training dataset in order to reduce overfitting.
- **Large Scale Jitter**: We use a training scale of 960 on the longest side and set a training scale range from 0.5 to 2.0.
- **Random Crop**: We also use the Random Crop setting with the size of 1600 to alleviate the GPU memory shortage for efficient training.
- **MixUp** [4]: We use the MixUp method to expand and augment data online.
- **Albu Augmentation** [5]: We use the settings in the Albu library to augment the training data online, including: RandomBrightnessContrast, RGBShift, HSVShift, Blur, Compression, Noise, and ChannelShuffle.
- **CopyPaste** [6]: We use the bounding box label as the segmentation label and use the CopyPaste method for instance-level augmentation online
- **Test Time Augmentation**: In the inference stage, we use Soft-NMS and flip augmentation to further enhance the results, with single model and single scale 960.

The detailed training schedule includes:

- **Optimizer and learning rate schedule**: The default optimizer is AdamW [9],with an initial learning rate of 0.0001 and a weight decay parameter of 0.05, and trained within 12 epochs.
- **SycBN**: We use SycBN as the default batch normalization setting.

D. Challenge results

Fill Table I with your obtained results, shown in the leaderboard of the challenge.

E. Final remarks

Please identify the pros and cons (if any) of the proposed approach.

[1] If you participated in more than one competition track, you need to share with the organizers one fact sheet per track.

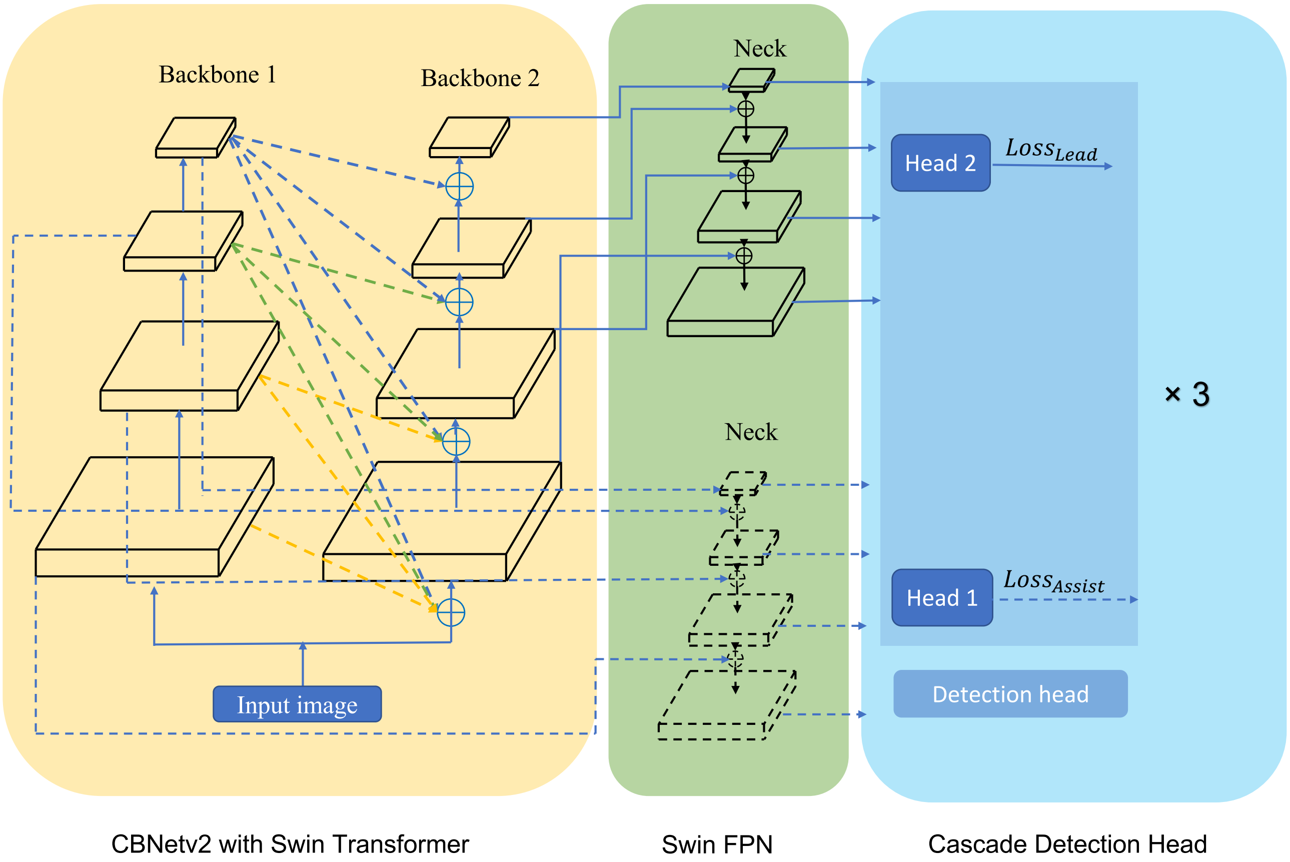

Fig. 1. The structures of our algorithm.

TABLE I
RESULTS FROM LEADERBOARD (TEST PHASE) OBTAINED BY THE PROPOSED APPROACH.

Rank position	mAP_w	mAP	Jan	Mar	Abr	May	Jun	Jul	Aug	Sep
2	0.322642	0.3301	0.3691	0.3548	0.3279	0.2827	0.2856	0.2435	0.3112	0.4662

We have not addressed well the long-tail problem in our dataset caused by the extreme sparsity of the bicycle and motorcycle categories. In our experiments, we found extremely low precision for these two categories.

III. ADDITIONAL METHOD DETAILS

Please, reply if your challenge entry considered (or not) the following strategies and provide a brief explanation. For each question, mark with X one single option.

- **For the competition track associated with this fact sheet, you confirm that you have trained your model on the predefined and single:** ( ) Day, (X) Week, ( ) Month - as instructed in the challenge webpage.

- **Did you use any pre-trained model:** (X) Yes, ( ) No.
  If yes, please detail (e.g., model architecture, training strategy, train data, etc.).
  The pretrained model can be found in this URL: Cascade Mask RCNN with CBNetv2 and Swin Transformer. Detailed information can be found in this URL: CBNetv2.

- **Did you use external data?** ( ) Yes, (X) No
  If yes, please detail:

- **Did you perform any data augmentation?**
  (X) Yes, ( ) No
  If yes, please detail:
  Data augmentation methods:
  - **Data sampling**: Considering the image continuity and repetition in the dataset, we only randomly sampled 30% of the image data from the original training dataset in order to reduce overfitting.
  - **Large Scale Jitter**: We use a training scale of 960 on the longest side and set a training scale range from 0.5 to 2.0.
  - **Random Crop**: We also use the Random Crop setting with the size of 1600 to alleviate the GPU memory shortage for efficient training.
  - **MixUp** [4]: We use the MixUp method to expand and augment data online.

  - **Albu Augmentation** [5]: We use the settings in the Albu library to augment the training data online, including: RandomBrightnessContrast, RGBShift, HSVShift, Blur, Compression, Noise, and ChannelShuffle.
  - **CopyPaste** [6]: We use the bounding box label as the segmentation label and use the CopyPaste method for instance-level augmentation online

- **At the final phase, did you use the provided validation set as part of your training set?** ( ) Yes, (X) No
  If yes, please detail:
- **Did you use any regularization strategies/terms?** ( ) Yes, (X) No
  If yes, please detail:
- **Did you use handcrafted features?** ( ) Yes, (X) No
  If yes, please detail:
- **Did you use any spatio-temporal feature extraction strategy?** ( ) Yes, (X) No
  If yes, please detail:
- **Did you perform object tracking?**
  ( ) Yes, (X) No
  If yes, please detail:
- **Did you leverage timestamp information?**
  ( ) Yes, (X) No
  If yes, please detail:
- **Did you use empty frames present in the dataset?**
  ( ) Yes, (X) No
  If yes, please detail:
- **Did you construct any type of prior to condition for visual variety?**
  ( ) Yes, (X) No
  If yes, please detail:

IV. Code repository

Link to a code repository with complete and detailed instructions so that the results obtained on Codalab can be reproduced locally. This includes a list of requirements, pre-trained models, and so on. Note, training code with instructions is also required. This is recommended for all participants and mandatory for winners to claim their prize. **Organizers strongly encourage the use of docker to facilitate reproducibility**.

Code repository: `https://github.com/heboyong/week-submit`

References

[1] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[2] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[3] T. Liang, X. Chu, Y. Liu, Y. Wang, Z. Tang, W. Chu, J. Chen, and H. Ling, "Cbnetv2: A composite backbone network architecture for object detection," *arXiv preprint arXiv:2107.00420*, 2021.

[4] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[5] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, 2020.

[6] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2918–2928.

[7] I. A. Nikolov, M. P. Philipsen, J. Liu, J. V. Dueholm, A. S. Johansen, K. Nasrollahi, and T. B. Moeslund, "Seasons in drift: A long-term thermal imaging dataset for studying concept drift," in *Thirty-fifth Conference on Neural Information Processing Systems*, 2021.

[8] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[9] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.