ECCV 2022 Sign Spotting Challenge

Fact sheet

I. Team details

- Team leader name: Xilin Chen
- Team name: VIPL NHCI
- Username on Codalab: Random guess
- Team leader affiliation: Institute of Computing Technology, Chinese Academy of Sciences
- Team leader address: 6 Kexueyuan Nanlu Zhongguancun, Haidian District, Beijing, China
- Team leader phone number:
- Team leader email: xlchen@ict.ac.cn
- Name of other team members (and affiliation): Yuecong Min, Peiqi Jiao and Aiming Hao (Institute of Computing Technology, Chinese Academy of Sciences)
- Team website URL: `http://vipl.ict.ac.cn/en/rese/sign/`
- Competition track (mark with X one single option) [1]:
  - (X) Track 1: **MSSL** (multiple shot supervised learning).
  - ( ) Track 2: **OSLWL** (one shot learning and weak labels).

II. Contribution details

A. *Title of the contribution*

Sign Spotting Using Representations from Domain Experts

Recent works have shown that effectively extracting features is the key challenge of sign language recognition. Due to the costly annotation of sign language, sign spotting is a valuable tool for the automatic annotation process of sign language. In this submission, we propose a two-stage pipeline for sign spotting: the first stage aims to condense sign language relevant information from multiple domain experts into a compact sign representation for sign spotting, and the goal of the second stage is to spot signs from longer video with a more powerful temporal module. Experimental results show that the proposed pipeline can perform better with limited training data.

B. *Representative image / workflow diagram of the method*

The workflow diagram is presented in Fig. 2. During the inference, the processed video clips are fed into four trained expert modules. The extracted features of each clip are concatenated into a vector to represent sign information for spotting. The temporal module takes vectors as input and extracts contextual information, and predicts class and corresponding offsets to the start and the end for each vector through two collaborative heads.

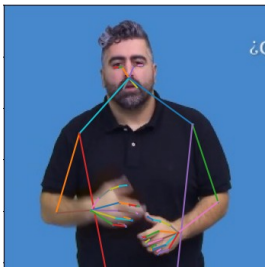

Fig. 1. An example of pose estimation

C. *Detailed method description*

Data Preprocessing. The original spatial resolution of the video is 1280x720, which is too large for the feature extractor. As shown in Fig. 1, we firstly adopt mediapipe [1] to estimate the 3D body and hand poses, and then crop the video frames to 720x720 based on the skeleton center and further resize them to 256x256. Due to the limited samples of the dataset, the spotting model is easily affected by the background and lighting. We further keep the hand region (64x64 for each hand) and mask other regions based on the center of hands estimated by mediapipe to extract hand-oriented spatio-temporal features as in previous work does [2]. Besides the video and skeleton, we also estimate TV-L1 optical flow to capture more motion information. In summary, we generate four kinds of processed data (cropped video, masked video, optical flow, and 3D skeleton, as shown in Fig. 2) from the original video to capture sign information more accurately.

The original dataset provides the start and the end of each sign, however, sign spotting needs fine-grained spatio-temporal information. Therefore, we segment the long video into a sequence of clips (each clip contains 8 frames) and adopt the majority class as the clip-wise label.

Backbone and Header. The total workflow is presented in Fig 2. For clipped video and masked video data, we adopt I3D model [3] pretrained on Kinetics-400 [2] dataset as the backbone. For optical flow data, we adopt p3d model [4] pretrained on Kinetics-400 [3] as the backbone. For skeleton data, we adopt ST-GCN model [5] [4] as the backbone with-

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

[2] `https://github.com/Tushar-N/pytorch-resnet3d`
[3] `https://github.com/qijiezhao/pseudo-3d-pytorch`
[4] `https://github.com/yysijie/st-gcn`

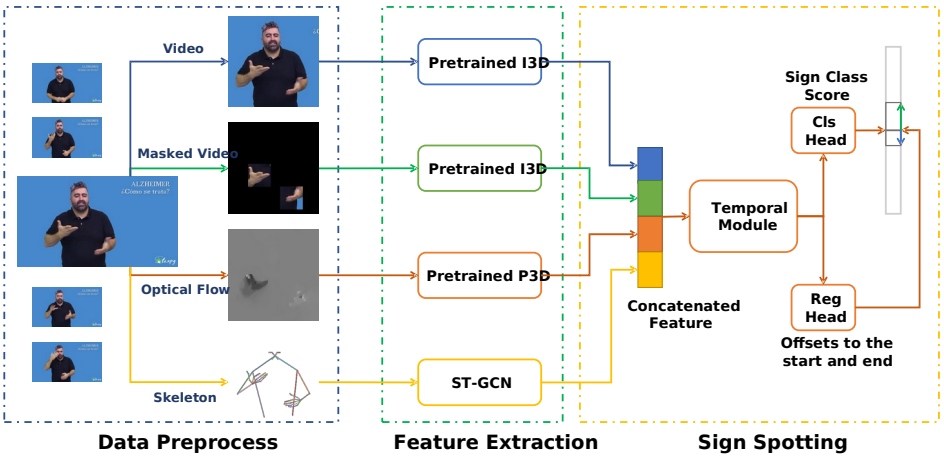

Fig. 2. Workflow diagram of the proposed sign spotting method

out pretrained weight. After feature extraction, the features from different data are concatenated into a 6400-dim vector (2048*3+256) and are fed into the temporal module. The temporal module is composed of a two-layer 1D convolutional and a two-layer BiLSTM for local and contextual temporal information extraction, respectively. Their outputs are integrated through a convolutional layer. The integrated features are fed into the classification head and the regression head, separately, to predict the corresponding class and the offsets to its start and end. Each head is composed of four convolutional layers.

Loss Function. For feature extraction and classification head, we adopt the vanilla cross-entropy loss to provide clip-wise supervision:

$$L_{cls} = -\sum_{i=1}^{C} p_i \log q_i \tag{1}$$

where p_i and q_i are ground-truth and prediction probabilities of the class i, and C is the total number of classes.

For the regression head, we adopt a differentiable Intersection over Union (IoU) loss [6], which considers both the IoU and the distance between central points:

$$L_{reg} = 1 - IoU + \frac{\rho^2(c, c^{gt})}{d^2}, \tag{2}$$

where c and c^{gt} are the central points of predicted and ground-truth signs, $\rho(\cdot)$ is the Euclidean distance, and d is the smallest enclosing interval containing both signs.

Training Details. Due to the high imbalance ratio between query signs and other signs, we adopt a two-round training scheme in the first stage (feature extraction training). Both rounds train the backbone like the training process of the isolate sign language recognition, and the L_{cls} is adopted to provide supervision. In the first round, a subset of clips that contains clips of query signs is built to increase the discriminative ability of the backbone. In the second round, all clips are used for training.

After the two-round training of the backbone, we adopt the trained backbone to extract the clip-wise feature sequence for each video. Both L_{cls} and L_{reg} are adopted to provide accurate supervision for sign spotting in the second stage (sign spotting training).

Algorithm 1 Decode Method

Input: the classification prediction $q \in \mathbb{R}^{T \times C}$ and the regression prediction $s \in \mathbb{R}^{T}$ and $e \in \mathbb{R}^{T}$, where s and e correspond to the offsets to the start and the end of the current sign.
Output: the decoded results r_t.

```
Init the vote matric v ∈ R^{T×C} ← 0
Init decoded result r ∈ R^T ← 0
for t = 1, 2, . . . , T do
    k = arg max_{c∈[1,C]} q_{t,c}
    for τ = t − s_t, . . . , t + e_t do
        v_{τ,c} ← v_{τ,c} + 1
for t = 1, 2, . . . , T do
    r_t = arg max_{c∈[1,C]} v_{t,c}
return r_t
```

Decode Method. As shown in Fig. 2, the proposed sign spotting model predicts class and its offsets. We propose a vote-based decode method to provide better localization results. Details can be found in Algorithm 1.

Other Hyper-parameters. For the training process of the backbone in the first stages, we train each model for 80 epochs with a mini-batch of 16 on a single 3090. We adopt the AdamW with a learning rate of 1e-4 for I3D and P3D models and Adam with a learning rate of 5e-4 for ST-GCN. The learning rate is divided by 5 at epochs 40 and 60.

For the training of the sign spotting model in the second stage, we train each model for 80 epochs with a mini-batch of 2 on a single 3090. The LayerNorm and ReLU are adopted after each convolutional layer. We adopt the AdamW optimizer with a learning rate of 1e-4. The learning rate is divided by 5 at epochs 20, 40, and 60. We augment the video training set with random crop and random horizontal flip (50%). For skeleton training data, we normalize coordinates and adopt a random horizontal flip (50%).

D. Challenge results

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

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

Rank position	avg F1
3	0.564260

E. Final remarks

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

Pros:

- Proposing a simple but efficient pipeline for sign spotting with the limited dataset. Four kinds of data are generated from the original video and capture sign information more accurately.
- Proposing a sign spotting approach that can provide more accurate results with extracted features. With the extracted features, the proposed approach can achieve a high F1 score without late fusion.

Cons:

- Adopting a two-round training scheme to learn more robust features, which can be improved with more efficient supervision, such as the Focal loss.
- From our experience, the feature extraction stage plays an important role in sign spotting. But we do not evaluate too many sota approaches and pretrained weights.

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.

- **Did you use pre-trained models?** (X) Yes, ( ) No
  If yes, please detail:
  As discussed in Sect. II-C, we use pretrained i3d [3] and p3d [4] models for RGB and optical flow, respectively.
- **Did you use external data?** ( ) Yes, (X) No
  If yes, please detail:
- **Did you use any kind of depth information (e.g., 3D pose estimation trained on RGBD data)?** (X) Yes, ( ) No
  If yes, please detail:
  We adopt mediapipe [1] to estimate 3D pose information and adopt them as the input of skeleton model.
- **At the final phase, did you use the provided validation set as part of your training set?** (X) Yes, ( ) No
  If yes, please detail:
  Yes, we retraining the model on both training and validation sets, and submit the prediction of the last epoch.
- **Did you use other 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 face / hand / body detection, alignment or segmentation strategy?** ( ) Yes, (X) No
  If yes, please detail:
- **Did you use any pose estimation method?** (X) Yes, ( ) No
  If yes, please detail:
  We adopt mediapipe [1] to estimate 3D pose information and adopt them as the input of skeleton model.
- **Did you use any spatio-temporal feature extraction strategy?** (X) Yes, ( ) No
  If yes, please detail:
- **Did you explicitly classify any attribute (e.g., gender/handedness)?** ( ) Yes, (X) No
  If yes, please detail:
- **Did you use any bias mitigation technique (e.g. rebalancing training data)?**
  (X) Yes, ( ) No
  If yes, please detail:
  Due to the high imbalance ratio between query signs and other signs, we adopt a two-round training scheme in the first stage (feature extraction training). Both rounds train the backbone like the training process of the isolate sign language recognition, and the L_{cls} is adopted to provide supervision. In the first round, a subset of clips that contains clips of query signs is built to increase the discriminative ability of the backbone. In the second round, all clips are used for training.

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/ycmin95/Chalearn_2022_Sign_Spotting_MSSL_track`

REFERENCES

[1] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee *et al.*, "Mediapipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172*, 2019.

[2] Z. Liu, X. Chai, Z. Liu, and X. Chen, "Continuous gesture recognition with hand-oriented spatiotemporal feature," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 3056–3064.

[3] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[4] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5533–5541.
[5] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the AAAI conference on artificial intelligence*, 2018.
[6] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 12 993–13 000.