ECCV 2022 Sign Spotting Challenge

Fact sheet

This is the fact sheet's template for the ECCV 2022 Sign Spotting 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: Hezhen Hu
- Username on Codalab: th
- Team leader affiliation: University of Science and Technology of China
- Team leader email: alexhu@mail.ustc.edu.cn
- Name of other team members (and affiliation): Landong Liu, Weichao Zhao, Hui Wu, Kepeng Wu, Wengang Zhou and Houqiang Li (University of Science and Technology of China)
- Team website URL (if any): `https://ustc-slr.github.io/`
- Competition track (mark with X one single option)[1]:
  - ( ) Track 1: **MSSL** (multiple shot supervised learning).
  - (X) Track 2: **OSLWL** (one shot learning and weak labels).

II. Contribution details

A. *Title of the contribution*

OSLWL Track is a one-shot learning problem, which aims to spot the exact location of the sign instances given the query isolated sign. In this challenge, we construct a two-stage framework, which consists of feature extraction and sign instance matching. Our simple yet effective framework ranked 1st in the test phase of this track.

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

The overview of our method is illustrated in Fig. 2.

C. *Detailed method description*

Our framework contains two important stages. Firstly, we extract frame-level feature representation from the pose modality. Then we build the similarity graph to perform frame-wise matching between the query isolated sign and long video.

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

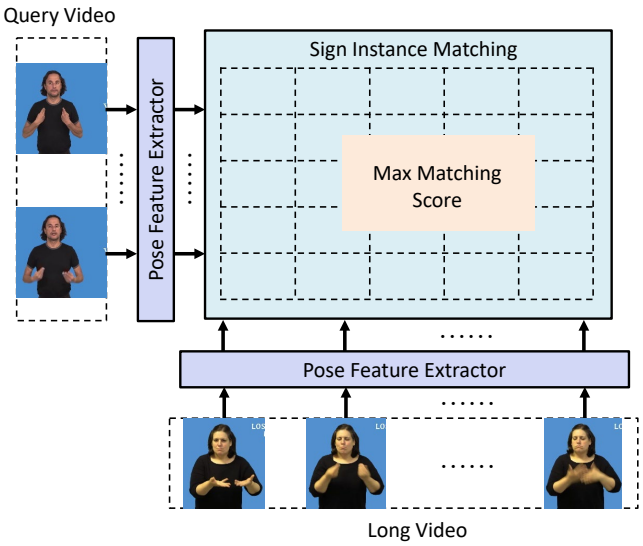

Fig. 1. The workflow diagram of the method.

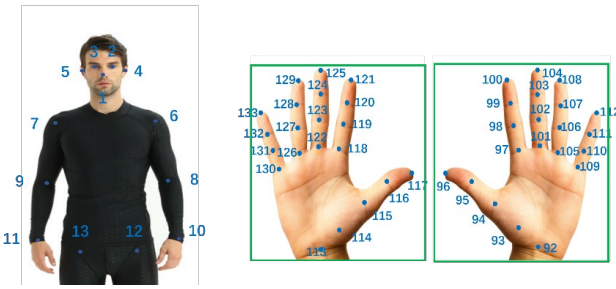

Fig. 2. The view of utilized mmpose skeleton.

Preprocessing. Since the signer only occupies a relatively small size of the original video, we utilize MMDetection [1] to first detect the signer spatial location, which is followed by MMPose [2] to extract the body and hand poses. Then, we only utilize the compact pose to indicate the gesture state of a certain frame. Meanwhile, we also extract the body pose via OpenPose [3] for trimming the effective time span of the query sign.

Feature extraction. Since sign language is mainly expressed via the manual features, we utilize the pre-trained GCN [4], [5] to extract the body and hand representation in a frame-wise manner.

Sign instance matching. Given the extracted feature of

the query and long video, we build a similarity graph to perform sign instance matching. Specifically, since the query sign contains many noisy frames, we first trim it for the effective signing process with the OpenPose detected body pose. Then we calculate the cosine similarity with the power of 3 between the features of each frame of the query video and each frame of the reference video, and get a two-dimensional matrix "sims[:,:]". After that, we create a graph, each point in the graph is a python tuple (x, y), which means that the x frame in the query and the y frame in the reference can be matched. If the two points (x1, y1), (x2, y2) in the map satisfy x1, x2 monotonically does not decrease; y1, y2 does not decrease monotonically and the difference between x2 and x1 does not exceed "max_step=4"; y2 and y1 does not exceed "max_step=4", an edge will be connected between the two points, and the edge weight will be set to "sims[x2,y2]" We introduce the start node (0, 0) and the end node (len(query), len(reference)), and let them also participate in the mapping.

Since the length of the reference video may be very long and there are too many nodes in the graph, for each frame of the query, select the "topk=16" frame with the highest cosine similarity in the reference and save it to the graph, and the nodes corresponding to the remaining frames will be discarded.

We will find the longest path in the graph, remove the start node and end node, and find the minimum and maximum positions of the y frame in the tuple, which is the result of query and reference matching.

Experiment settings. Since this problem is a one-shot learning setting, we collect the output of the sign instance matching as the final results.

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
1	0.595802

E. Final remarks

Our method only relies on the compact body and hand pose to perform matching and achieves a notable gain over other solutions. Incorporating the non-manual cues (e.g. facial expressions) may further boost the performance.

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:
  GCN (Pose).
- **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)?** ( ) Yes, (X) No
  If yes, please detail:
- **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 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?** (X) Yes, ( ) No
  If yes, please detail:
  We utilize the OpenPose to detect the existence of the hand.
- **Did you use any pose estimation method?** (X) Yes, ( ) No
  If yes, please detail:
  We utilize both OpenPose and MMPose to extract the body and hand poses
- **Did you use any spatio-temporal feature extraction strategy?** ( ) Yes, (X) 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)?**
  ( ) Yes, (X) No
  If yes, please detail:
- *Just answer the following question if this fact sheet is associated with an **OSLWL track** submission.*
  Did you use MSSL train/val data and/or OSLWL val data as a means of unsupervised model training?
  ( ) 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://mailustceducn-my.sharepoint.com/:u:/g/personal/alexhu_mail_ustc_edu_cn/Eb3H_CGysyBNiacMbrYHUHUBJF2xHyH9vWru6H9wpsV50w?e=lgmjq3`

REFERENCES

[1] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv*, 2019.

[2] M. Contributors, "Openmmlab pose estimation toolbox and benchmark," https://github.com/open-mmlab/mmpose, 2020.

[3] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "OpenPose: Realtime multi-person 2d pose estimation using part affinity fields," *TPAMIN*, 2019.

[4] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann, "Exploiting spatial-temporal relationships for 3D pose estimation via graph convolutional networks," in *ICCV*, 2019, pp. 2272–2281.

[5] H. Hu, W. Zhao, W. Zhou, Y. Wang, and H. Li, "SignBERT: Pre-training of hand-model-aware representation for sign language recognition," in *ICCV*, 2021, pp. 11 087–11 096.