Fact sheet: CVPR 2021 ChaLearn Looking at People Large Scale Signer Independent Isolated SLR Challenge

I. Team details

- **Challenge Track**: RGBD
- Team leader name: Songyao Jiang
- Username on Codalab: smilelab2021
- Team leader affiliation:
  Department of Electrical and Computer Engineering, Northeastern University, Boston MA, USA
- Team leader address: 360 Huntington Ave, Boston MA 02115
- Team leader phone number: (734)546-0695
- Team leader email: jiang.so@northeastern.edu
- Name of other team members (and affiliation): Bin Sun, Lichen Wang, Yue Bai, Kunpeng Li and Yun Fu, Department of Electrical and Computer Engineering, Northeastern University
- Team website URL (if any):
  https://web.northeastern.edu/smilelab/

II. Contribution details

A. Sign Language Recognition Based on Whole-body Pose Estimation and Multi-modal Ensemble

Sign language is a visual language that is used by deaf or speech impaired people to communicate with each other. Sign language is always performed by fast transitions of hand gestures and body postures, requiring a great amount of knowledge and training to understand it. Sign language recognition becomes a useful yet challenging task in computer vision. Skeleton-based action recognition is becoming popular that it can be further ensembled with RGB-D based method to achieve state-of-the-art performance. However, skeleton-based recognition can hardly be applied to sign language recognition tasks, majorly because skeleton data contains no indication of hand gestures or facial expressions. Inspired by the recent development of whole-body pose estimation [1], we propose recognizing sign language based on the whole-body key points and features. The recognition results are further ensembled with other modalities of RGB and optical flows to improve the accuracy further. In the challenge about isolated sign language recognition hosted by ChaLearn using a new large-scale multi-modal Turkish Sign Language dataset (AUTSL) [2]. Our method achieved leading accuracy in both the development phase and test phase.

B. Introduction and Motivation

Isolated sign language recognition can be seen as a video classification task. The isolated sign language recognition is also very similar to human action recognition. So in this section, we will introduce the methods of action recognition and discuss their limitation on isolated sign language tasks.

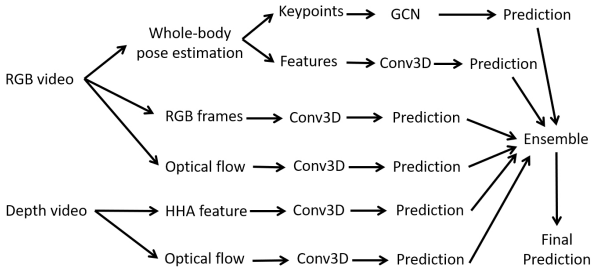

Fig. 1. Workflow of the proposed multi-modal sign language recognition. We use six modalities in RGBD track. We train them separately and ensemble their results after FC layer to obtain our final predictions.

The state-of-the-art methods on action recognition tend to use a 3D convolution architecture to capture spatio-temporal information from the input videos [3], [4], [5], [6]. However, such approaches require a very large-scale dataset due to their huge amount of parameters caused by the 3D convolutional layers. 3D convolution based methods perform better in the recent large dataset such as Kinectics-600 [7]. Smaller datasets often deliver less satisfying results due to overfitting. In isolated sign recognition, the available datasets are always much smaller than action recognition which is not suitable for those large 3D convolutional networks. Another approach of action recognition is using skeleton-based data to recognize human actions.

Spatial temporal Graph Convolutional Network (ST-GCN) was first introduced to handle skeleton-based action recognition in [8]. The authors constructed the skeleton graph to be the natural connection of the human body. They proposed a spatial temporal graph convolutional module to extract both the spatial and temporal information from the input skeleton graph data. There have been efforts to improve ST-GCN in terms of adaptive graph learning and multi-stream ensemble [9], and decoupling the GCN layers to boost the graph modeling capacity [10]. They also find that assembling a skeleton-based model with RGB based model delivers better overall accuracy than both models. Those GCN-based methods always limit themselves to datasets that provide ground-truth keypoints, because pre-trained pose estimation models do not always give a reliable estimation of poses, which introduces a bunch of noises and harms their accuracy on the action recognition task.

In sign language recognition, skeleton-based methods often give lower performance simply because the skeleton does

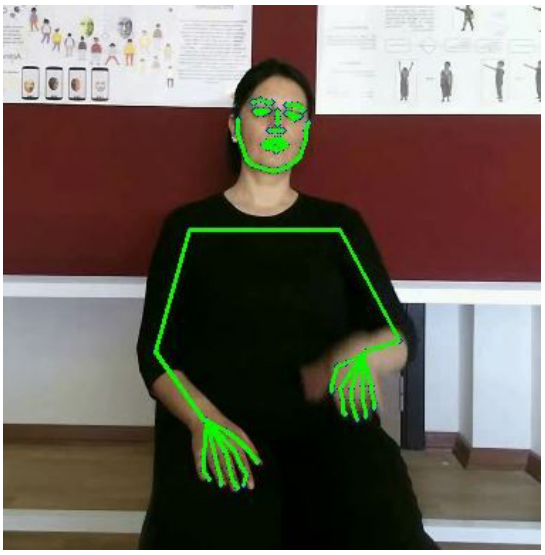

Fig. 2. Example of whole-body pose estimation on AUSL dataset. Note that this video is mirrored. The right hand of the signer is blurred due to its fast motion. Conventional hand detectors will fail in such cases. But whole-body pose estimation algorithm can still provide faithful keypoints estimation because it utilizes global information of the other body parts.

not contain the keypoints of hands gesture. If we want to add hand keypoints to the skeleton using a hand detector and a hand pose estimator, the outcomes always fail us. The hand detector performs poorly as it always lost the targets due to the limitation in resolution or motion blur. The recent development on whole-body pose estimation [1] encourages us to use whole-body pose estimation methods to obtain faithful keypoints of hands, see Fig. 2. We find that the whole-body pose estimation model is able to estimate the location and keypoints of hands based on global information of arms. Naturally, we propose to use whole-body pose keypoints to recognize sign language via a multi-stream GCN model with graph detection and graph dropout. It turns out that our proposed whole-body skeleton-based sign language recognition gives an accurate recognition rate.

Studies on action recognition have revealed that multi-modalities can further improve the performance of recognition. Therefore, in the RGB-D track, we add five more modalities to assemble with the skeleton-based approach introduced above. First, we extract the feature from the pre-trained whole-body pose model frame by frame and use them as the input to a model consisting of spatial and temporal convolutional layers and recognizing sign language. Second, we use RGB frames as another modality and use 3D convolutional models to classify those frames. Third, we extract optical flow from original videos and use the same model as RGB. Fourth, we extract HHA features from depth videos and generate predictions using the same model as RGB frames. The final modality is similar to optical flow. We use the same 3D convolution model to process the flow features extracted from depth videos. In the end, we assemble the results from all modalities and generate our final prediction. The overall workflow is shown in Fig. 1.

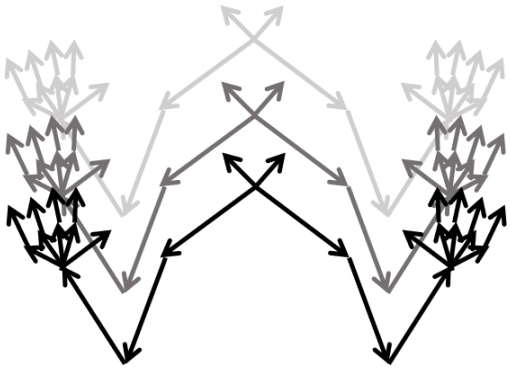

Fig. 3. Whole-body skeleton graph after graph deduction. 133 keypoints are reduced to 27 keypoints containing upper body skeleton (including nose and eyes) and a subset of hands keypoints. In the spatio-temporal graph construction, each keypoint is spatially connected to its adjacent keypoints and also itself in temporal dimension.

Our approach achieves the 1st rank in this challenge in both the development phase and test phase, see Table I.

C. Detailed method description

In RGBD track, we use six modalities including whole-body pose keypoints, whole-body pose features, RGB frames, optical flows, HHA and depth flows to obtain our final predictions. In this section, we introduce each modality one by one and present the ensemble method we used to obtain the final prediction.

1) Skeleton keypoints: The whole-body pose network estimate 133 keypoints from the detected person, including facial landmarks, body skeleton, hands, and feet keypoints. We construct a spatio-temporal graph by connecting the spatially adjacent keypoints according to the natural connections of human body, and keypoints to themselves temporally. The large number of keypoints introduces a lot of noise to the model. Simply feeding such a whole-body graph containing all the estimated keypoints gives very low accuracy. Therefore, according to the observations on the data and the visualization of GCN activations, we do a graph detection and trim down the 133 keypoints to 27 keypoints only. The remaining graph is shown in Fig. 3. Our experiments show that those 27 keypoints contain all the information we need to do sign language recognition. Such graph deduction gives faster model convergence and higher accuracy.

Inspired by [9], we use four streams in our GCN model, which are joint, bone, joint motion, and bone motion. Bone data are generated by representing joint data in a vector form using the natural connection of the human body. Motion data are generated by subtracting a frame by its previous frame. We train one model for each stream separately and combine their predicted results before the softmax layer by simply adding them together with weights. Each stream uses data augmentation, which randomly flips, rotates, scales, and shifts the input keypoints location. We use a video length of 150. If a video has lesser frames than 150, we repeat that video until we get 150 frames. The coordinates of keypoints

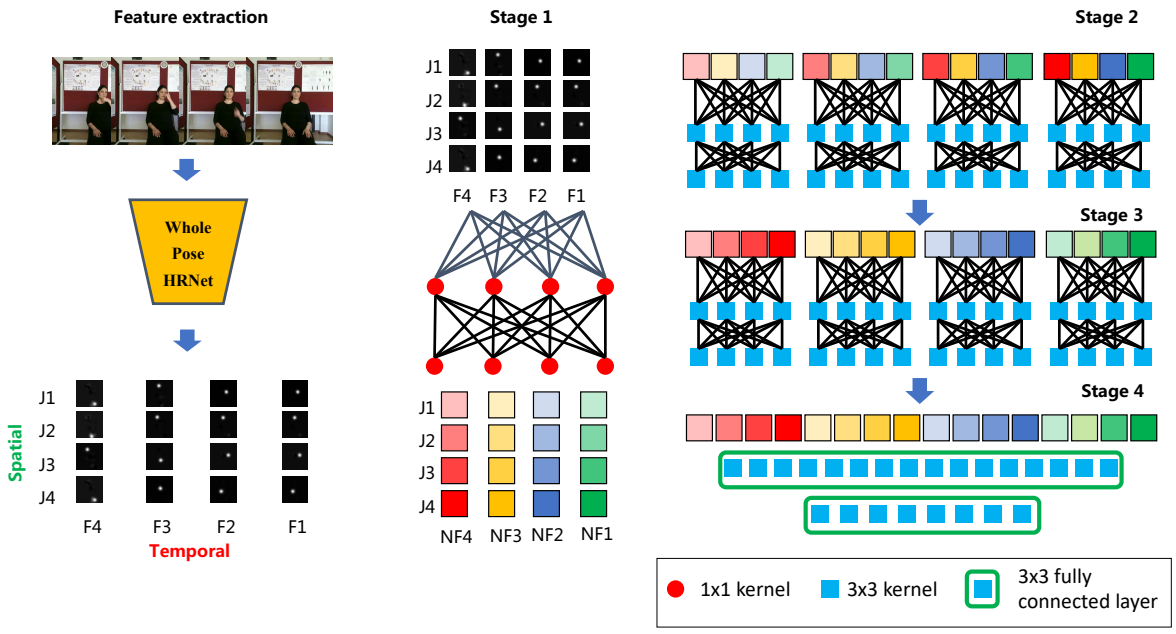

Fig. 4. The pipeline of Sign Language Recognition based on skeleton features.

are normalized to [-1,1]. Based on ST-GCN, we implement a STC (spatial, temporal, and channel) attention module described in [9] to improve the predicted accuracy. We adopt the Decoupled-GCN and DropGraph module in [10] to boost GCN modeling capacity with no extra computation cost and avoid overfitting.

2) Skeleton features: Besides using key point coordinates generated from the whole-body pose network, we also propose a model to recognize the sign language with whole-body features. We extract 33 landmarks from 60 frames of each video as the input of our model, which contain 1 landmark on the nose, 4 landmarks on the mouth, 2 landmarks on shoulders, 2 landmarks on elbows, 2 landmarks on wrists, and 22 landmarks on hands. All the features are down-sampled to 24×24 using max pooling. Instead of using 3D convolution, we process the input features with a 2D convolution layer separably, which is easier to converge. The pipeline is shown in Figure 4. There are four stages in total. In the first stage, we reshape the features from $60 \times 33 \times 24 \times 24$ to $60 \times 792 \times 24$, and feed them to 1×1 convolution layers, which means we only process temporal information in this stage. Then we shuffle the features and divide them into 60 groups, and utilize grouped 3×3 convolution to extract temporal and spatial information among the same key point features from different frames. In this stage, temporal information and part of spatial information are processed. In the third stage, the features are shuffled again and divided into 33 groups. We still use grouped 3×3 convolution, but only spatial information in each frame is extracted. Finally, a couple of 3×3 fully connected layers are implemented to generate prediction features. In the first 3 stages, all the output is added by a residual. Moreover, a dropout layer is deployed in each module to avoid over-fitting.

3) RGB and optical flow: As mentioned in Section II-B, studies on action recognition have revealed that multi-modal ensembles can further boost each modality's performance. So we also implement traditional modalities of RGB frames and optical flows using 3D convolutions. As mentioned in Section II-B, most 3D convolutional architectures suffer from overfitting, especially on smaller datasets due to the large number of parameters in 3D convolutional layers. In our experiment, we have tried large capacity 3D convolutional nets like Resnet3D [3], [11], SlowFast [4], SlowFast with Bert [5], which are commonly used in action recognition. All the above models are hard to be trained on this relatively smaller-scale sign language recognition dataset. Loading pre-trained models on large-scale action recognition datasets such as Kinectics dataset [7] do improve the accuracy a little, but the performance of those 3D convolution networks is still quite low.

In our study, we find out that Resnet2+1D [6], which decouples spatial and temporal convolution and does them one after another, provides the best results among the above 3D convolutional architectures. We find that increasing the architecture's depth does not improve the performance and is easier to overfit. So in our experiments, we choose Resnet2+1D-18 with weights pre-trained on Kinectics dataset as our backbone network for both RGB and optical flow modalities. For RGB data, we pre-train the model on the Chinese Sign Language (CSL) dataset [12] as well to improve the model convergence. We find that pre-training on CSL can reduce the training time and improve the final accuracy by around 1%.

Studied in [13], using one-hot labels with cross-entropy loss results in overfitting in some cases. So we adopt the label smoothing technique to alleviate such effect. Mathematically,

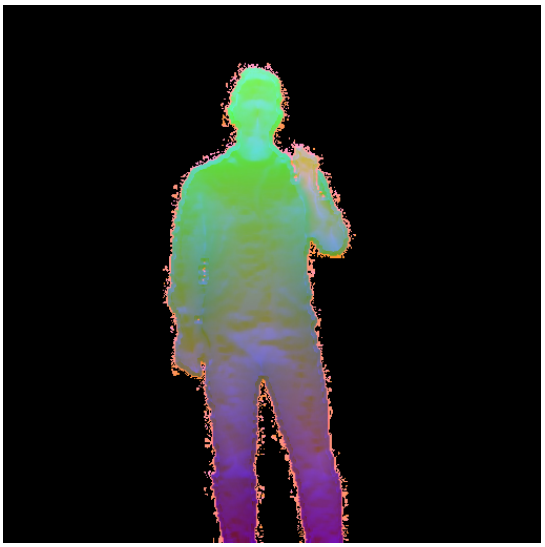

Fig. 5. An example of extracted HHA feature map. The non-person region are masked out and filled with zeros.

label smoothing can be represented as

$$q'(k|x) = (1-\epsilon)\delta_{k,y} + \epsilon u(k), \tag{1}$$

where $q'(k|x)$ is a new form of predicted distribution, ϵ is a hyper-parameter between 0 and 1, $u()$ is a uniform distribution and k is the number of classes. The cross-entropy loss can then be replaced as

$$H(q', p) = -\sum_{k=1}^{K} \log p(k) q'(k) = (1-\epsilon)H(q, p) + \epsilon H(u, p), \tag{2}$$

where such representation can be regarded as a combination of penalties to the difference between predicted distribution with real distribution and prior distribution (uniform distribution). In our experiment, we set ϵ to be 0.1 and observe 0.5% to 1.0% gain in predicted accuracy for both RGB and optical flow modalities.

4) Depth HHA and depth flow: We extract HHA features from depth video as another modality. HHA features encode depth information and generate using a RGB-like 3-channel output, where HHA stand for

H – Horizontal Disparity
H – Height above the ground
A – Angle Normal makes with

Using HHA instead of using gray-scale depth video directly enables better understanding of the scene yet improves the recognition performance. We observe that the provided depth videos come with a mask that the non-person region is filled with zeros. So when generating HHA features, we mask out those regions as well and fill the corresponding regions in the final HHA outputs with zeros. An example of our extracted HHA with mask can be found in Fig. 5. We use HHA modality as same as the RGB modality (with Resnet2+1D as backbone) . The only difference is that we do not pretrain the HHA model on CSL dataset.

Besides, we find that we can extract optical flow from the depth video as well. The depth flow is cleaner compared with the RGB flow and delivers better accuracy. So we use the depth flow as one of our modalities as well. Both the HHA prediction and depth flow prediction are enssembled in addition to the modalities used in RGB track to make our final prediction.

5) Multi-modal ensemble: We use a simple ensemble method to ensemble all four modalities above. Specifically, we save the output of the last fully-connected layers of each modality before softmax layer. Those outputs have the size n_c where n_c is the number of classes. We assign weights to the every modality according to their accuracy on validation set and sum them up with weights as our final predicted score

$$score = \alpha_1 q_{skel} + \alpha_2 q_{RGB} + \alpha_3 q_{flow} + \alpha_4 q_{feat} + \alpha_5 q_{HHA} + \alpha_6 q_{depthflow},$$

where q represents the result of each modality, $\alpha_{1,2,3,4,5,6}$ are hyper-parameters to be tuned based on ensemble accuracy on validation set. We find the maximum score as our final predicted classes. In our experiments, we use [1.0,1.4,0.5,0.4,0.5,0.4] for RGBD track. We have tried other ensemble methods such as early fusion or training fully-connected layers to ensemble the final prediction. Despite that, we find that the simplest method we presented above gives us the best accuracy. We are going to submit a workshop paper which includes more details of our method.

D. Challenge results and final remarks

Our team (smilelab2021) ranked 1st in both development phase and test phase in both RGB and RGBD tracks. Our rankings and accuracy can be found in Table I and are also shown in the leaderboard of the challenge track [1] [2]).

TABLE I
LEADERBOARD: RESULTS OBTAINED BY THE PROPOSED APPROACH.

Phase	Track	Rank position	Rec. Rate
Development	RGBD	1st	97.03
Test	RGBD	1st	98.53

III. ADDITIONAL METHOD DETAILS

Please reply if your challenge entry considered (or not) the following strategies and provide a brief explanation.

- **Did you use any kind of depth information (directly, such as RGBD data, or indirectly such as 3D pose estimation trained on RGBD data), either if during training or testing stage?** (X) Yes, ( ) No
  We extract HHA features and optical flow from the depth videos and use them separately to predict the sign language classes. The results are later ensembled with

[1]RGB: https://competitions.codalab.org/competitions/27901#results
[2]RGB+D: https://competitions.codalab.org/competitions/27902#results

the other modalities to obtain our final predictions in the RGBD track.

- **Did you use pre-trained models?** (X) Yes, ( ) No
  We used Resnet2+1d pretrained on Kinectics dataset [7].
- **Did you use external data?** (X) Yes, ( ) No
  For RGB modality, we pretrained our models on Chinese Sign Language dataset [12] before training on the challenge dataset. The pretrained model is provided in our google drive. We didn't use external data for the other modalities such as keypoints, features, optical flow or HHA models.
- **Did you use other regularization strategies/terms?** (X) Yes, ( ) No
  We used label smoothing and weight decay as regularization in training our models.
- **Did you use handcrafted features?** (X) Yes, () No
  We use two kinds of handcraftded features. The first type is HHA feature obtained from depth videos. The second one is the optical flow extracted from both RGB and depth videos.
- **Did you use any face / hand / body detection, alignment or segmentation strategy?** ( ) Yes, (X) No

- **Did you use any pose estimation method?** (X) Yes, ( ) No
  We use wholebody pose estimation algorithm to extract 133-point whole body pose from the input images. These keypoints include face, hand, body and foot keypoints. These keypoints is used in our GCN network as skeleton modality. The features extracted from pretrained wholebody pose estimation are used as another modality. The keypoints are also used to crop frames in other modalities (RGB and optical flow).
- **Did you use any fusion strategy of modalities?** (X) Yes, ( ) No
  Since we have multiple modalities (skeleton keypoints, skeleton features, RGB, optical flow, HHA and depth flow), we adopt a late fusion techniques that we save the output of the last fully-connected layers, before softmax, associate weights to them and sum them up with weights as our final predicted score. Those weights serve as hyper-parameters and we tune those parameters based on the accuracy on validation set.
- **Did you use ensemble models?** ( ) Yes, (X) No

- **Did you use any spatio-temporal feature extraction strategy?** (X) Yes, ( ) No
  All of our models such as spatio-temporal GCN and Resnet2+1D extract spatio-temporal features from the input data sequence before feeding to the last fully-connected layer for classification.
- **Did you explicitly classify any attribute (e.g. gender)?** ( ) Yes, (X) No

- **Did you use any bias mitigation technique (e.g. rebalancing training data)?**
  ( ) Yes, (X) No

IV. Code repository

We have made our code public to reproduce our results and facilitate the research on sign language recognition. We upload pretrained models and preprocessed test data as well. We also provide an Nvidia docker image for fast deployment of our environment. Code, pretrained models, and detailed instruction can be found in our repository.

Code repository:
`https://github.com/jackyjsy/CVPR21Chal-SLR`

References

[1] S. Jin, L. Xu, J. Xu, C. Wang, W. Liu, C. Qian, W. Ouyang, and P. Luo, "Whole-body human pose estimation in the wild," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[2] O. M. Sincan and H. Y. Keles, "Autsl: A large scale multi-modal turkish sign language dataset and baseline methods," *IEEE Access*, vol. 8, pp. 181 340–181 355, 2020.

[3] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6546–6555.

[4] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6202–6211.

[5] M. E. Kalfaoglu, S. Kalkan, and A. A. Alatan, "Late temporal modeling in 3d cnn architectures with bert for action recognition," in *European Conference on Computer Vision*. Springer, 2020, pp. 731–747.

[6] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[7] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, "A short note about kinetics-600," *arXiv preprint arXiv:1808.01340*, 2018.

[8] 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*, vol. 32, no. 1, 2018.

[9] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 9532–9545, 2020.

[10] K. Cheng, Y. Zhang, C. Cao, L. Shi, J. Cheng, and H. Lu, "Decoupling gcn with dropgraph module for skeleton-based action recognition."

[11] H. Kataoka, T. Wakamiya, K. Hara, and Y. Satoh, "Would mega-scale datasets further enhance spatiotemporal 3d cnns?" *arXiv preprint arXiv:2004.04968*, 2020.

[12] J. Zhang, W. Zhou, C. Xie, J. Pu, and H. Li, "Chinese sign language recognition with adaptive hmm," in *2016 IEEE international conference on multimedia and expo (ICME)*. IEEE, 2016, pp. 1–6.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.