Real-time Detection of Bangladeshi Sign Words from Time Series Data

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

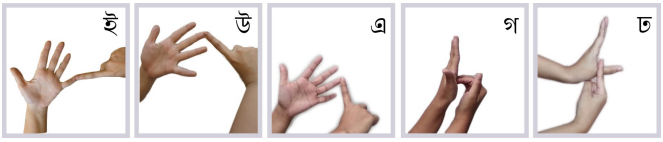
6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

***Abstract*—Bangladeshi Sign Language (BdSL) is a commonly used medium of communication for the hearing-impaired people in Bangladesh. This has a broad social impact and an interesting avenue of research. However, it is a challenging task due to the variation of different subjects (age, gender, etc), backgrounds and the complexity in sign words. BdSL sign letters are a fixed pose of action. On the contrary, sign words are a combination of hand poses. Due to variation in action, sign word recognition gets more difficult. Developing a real-time system to detect these signs from images or videos is a great challenge. Such type of work is rare, especially no work on sign word recognition has been done so far in Bangladesh. We present different methodologies to detect words from real-time videos. For the sign word recognition, CNN followed by Long Short Term Memory (LSTM) network, has been applied to combine the spatio-temporal relationships of the frames for each word. These methods are implemented on our own generated dataset. There were no such datasets available on BdSL and the sign languages are significantly different in different countries and even regions. So, we had to develop a dataset called *BdSLWord* by ourselves to train our systems and our datasets are open for future research. To demonstrate the application on the user level, we developed Graphical User Interfaces (GUIs) for our systems.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Sign language is a non-verbal form of communication, used by people with the inability to speak or hear, through bodily movements especially with hands and arms. Detecting the signs automatically from images or videos is an appealing task in the field of Computer Vision. Understanding what signers are trying to describe always requires reconstructing the different poses of their hands. These poses and gestures differ from region to region, language to language, i.e. for American Sign Language (ASL), Bangladeshi Sign Language (BdSL), etc. A huge number of people in Bangladesh rely on BdSL to communicate in their day to day life, and the need for a communication system—as a digital interpreter between signers and non-signers—is quite obvious. Activity recognition concerning the different body and hand pose from images and videos have advanced significantly, while sign language recognition has received less attention—especially when it comes to Bangladeshi Sign Language. In this work, we tend to simplify the way non-signers communicate with signers through a computer vision system for BdSL by exploiting machine learning tools. However, we investigate the problem of detecting BdSL words—which has particularly been an untouched research domain in Bangladeshi Sign Language detection. BdSL letters consist of single-hand pose (see Fig.1a) whereas sign words consist of sequence of actions (see Fig.1b). Therefore, sign word recognition is much complex than sign letters concerning huge datasets, computational expenses and synchronizing the variations in actions in words.



(a) Different signs of BdSL letters are shown here. Each letter consists of single hand pose.

(b) The sequence of images of a video gesture belonging to class '*Tumi* (You)'. The image sequence of hand poses (from left to right) propagate one word.

Fig. 1: Shows the difference between the BdSL letters and words. BdSL letters can be represented with a single action, but words consist of several actions. In (b) we can see the hand positions are changing with time and hand pose can also differ as well for a particular word. Due to these variations, sign word recognition gets difficult.

For recognizing sign words not only localizing the hand in the image but how they move relative to time is necessary. So, these types of recognition depend on both spatial and temporal data. Most of the previous spatial-temporal computer vision techniques have extracted hand-crafted the spatial features and then used primitive temporal modeling approaches [7]. The advent of modern deep learning methods has removed the need for such hand-crafted representations and enabled systems to entirely learn both the spatial and temporal features. However, there has been no work done on BdSL word recognition, most of the BdSL and other sign language recognition researches are based on isolated sign samples and very few on sequential samples. As mentioned earlier, due to the necessity of a larger dataset, sequence-to-sequence modeling problems, get more difficult to work with.

To address the above problem, we propose a real-time system for recognizing sign words using deep neural networks. In this paper, for sign word recognition, we are expecting to detect and recognize signs from a temporal sequence of input in real-time. We base our analysis on the work of Masood et al. [6] and for this purpose, we generated a video dataset under constrained conditions and the only additional equipment necessary for data acquisition is a colored glove for the simplicity of training the classifier. To the best of our knowledge, this is the only temporal dataset available on BdSL words. We divide the task of sign word recognition into: (1) per frame spatial feature extraction by CNN, and (2) using these spatial features as temporal data to feed into an LSTM model. The goal of our work is to develop practical end-to-end systems for sign gesture recognition applications. In sum, our contributions are:

- A real-time system for BdSL sign word using CNNs and LSTM. This is the first BdSL word recognition real-time application in Bangladesh.
- A processed BdSL sign words video dataset, known as *BdSLWord*. This dataset is also available online for further research.

II. Related Works

Hand pose recognition has become a great field of research due to its applications in human-computer interaction, motion control, and activity recognition. However, very few works have been done on Bangladeshi Sign Languages (BdSL) words, particularly when it comes to applying in real-time. Previous works were fully based on old-fashioned image processing methods – i.e. morphological operations, color-based foreground segmentation – yielding a huge obstacle in real-time detection. We first review the works on different sign letter recognition followed by a discussion of work in the field of sequential movement recognition of hand gesture.

A. Letter Based Hand Pose Recognition.

Ahmed and Akhand [1] determine relative fingertip positions from the 2D image and train an artificial neural network using those tip-position vectors. In their approach, the recognition is not in real-time and the authors claim to have an accuracy of 98.99% for BdSL detection. In (Rahaman et al. 2015a) [8], the authors introduced a computer vision-based system that applies contour analysis and Haar-like feature-based cascaded classifier. They trained their classifier and tested the system using 3600 contour templates for 36 Bangladeshi signs separately and achieve 96.46% recognition accuracy. In (M. A. Rahaman 2017), the authors also introduced an approach for detecting BdSL letters and digits which applies a fuzzy logic-based model and grid-pattern analysis in real-time. In (Rahaman et al. 2015b) [10], the authors presented a real-time Bengali and Chinese numeral signs recognition system using contour matching. The system is trained and tested using total of 2000 contour templates separately for both Bengali and Chinese numeral signs from 10 signers and achieved recognition accuracy of 95.80% and 95.90% with the computational cost of 8.023 milliseconds per frame. In (Muhammad Aminur Rahaman 2018) [9], the authors introduced a method of recognizing the Hand-Sign-Spelled Bangla language. The system is divided into two-phase – hand sign classification and automatic recognition of hand-sign-spelled for BdSL using the Bangla Language Modeling Algorithm (BLMA). The system is tested for BLMA using words, composite numerals and sentences in BdSL achieving mean accuracy of $93.50\%, 95.50\%$ and 90.50% respectively. In [5], the authors use the Faster R-CNN model to develop a system that can recognize Bengali sign letters in real-time and they also propose a dataset of 10 classes. They train the system on about 1700 images and were able to successfully recognize 10 signs with an accuracy of 98.2%.

B. Word Based Sequential Hand Pose Recognition

For sequence-to-sequence based recognition one of the popular deep learning methods is two-stream CNNs with 2D convolutional kernels [3], [4], [12], [14], [15]. Feichtenhofer et al. proposed combining two-stream CNNs with ResNets [4]. They showed the architecture of ResNets is effective for action recognition with 2D CNNs. CNN for spatial feature extractions from frames and modeling them by RNN based models have also been popular in sign language recognition [2], [6]. N. C. Camgoz et al. proposed Convolutional Neural Networks (CNNs) take images as inputs and extract spatial features followed by a bidirectional Long Short Term Memory Layers (BLSTM) temporally model the spatial features extracted by the CNNs and a Connectionist Temporal Classification (CTC) Loss Layer [2]. Masood Sarfaraz et al. [6] proposed methodology also consists of CNNs for capturing local spatial patterns in the data and these data are fed into a single layer LSTM consist of 256 LSTM units to recognize the pattern in temporal data.

Our study on several vision-based techniques for BdSL detection exhibits that no research attempt had been taken yet to exploit deep learning in this particular field of BdSL word recognition. There are few on BdSL letters, words with combining the letters but no work on BdSL word recognition has been done yet. Hence, in our work, we take this advantage and develop techniques to detect BdSL in real-time.

III. Dataset

As mentioned earlier, there were no available datasets to train deep learning models on BdSL. We contribute a dataset called *BdSLWords* which has been employed in our systems. This dataset is used for CNN & LSTM based word recognition classifier. For the video dataset, we have considered hands-only till wrist. Sample of our datasets is available here: `https://github.com/anno23/BdSLWord`.

A. Video Dataset

As no previous research is done on BdSL word recognition system using deep learning, there are no available datasets on BdSL words. *Dataset_2* follows the dataset Masood Sarfaraz et al. [6]. Currently, this consists of 4 different classes. In this dataset, there is a total of 200 videos with 8000 frames, each of the class having 50 videos and 2000 frames of 5 different subjects. The subjects wore a red hand glove on the right hand and their clothes and background were kept different than the gloves. Videos are collected in the same environment for the simplicity of hand segmentation while fully retaining the hand pose. For hand segmentation from each frame, we used RGB thresholding and then the frames were converted to gray–scale images. The threshold values used for our dataset which is based on the color of the gloves, are minimum and maximum values of $[10, 10, 150] - [90, 90, 255]$ where the values are for $[B, G, R]$. Each video was cropped in a way so that the frames only contained hand gesture of an individual word. Each video of gesture are of 1 second and 40 frames are collected from each of them. Figure 2 shows some example from our *BdSLWord*.

IV. Proposed Methodology

In this section, we describe our approaches for Sign Word Recognition System. Sign words contain a sequence of actions. Only predicting on a single frame is not enough, it requires prediction on multiple frames and to combine the predictions to identify the class. For this purpose, we implemented Masood et al. work on our *Dataset_2*. The network pipeline consists of the CNN layer for spatial feature identification followed by an LSTM layer with 256 hidden units to learn the temporal features. For extracting the spatial feature from an individual frame, a pre-trained Inceptionv3 [13] model from the TensorFlow library is used which is trained for the ImageNet Dataset. We only trained the final layer of the network and added a new final layer having the size of the number of total classes. We have extracted features from CNN from two different layers and trained the LSTM with each of them to compare the outputs.

A. Feature Extraction and Background Removal.

Each video is converted into a sequence of frames. Each of these frames goes through background removal and hand segmentation process. For hand segmentation from each frame, we used RGB thresholding and then the frames are converted to gray-scale images. These final frames only consist of hand pose to avoid the high computational expenses and easier learning of features for a CNN. As our goal is to identify the temporal features related to the spatial features Long Short Term Memory (LSTM) architectures are best suited for it. LSTM is a recurrent neural network (RNN) architecture that can remember the values over arbitrary intervals. Where there is a time series of unknown duration, LSTM is well-suited to classify, process and predict [11].

B. LSTM With Prediction Layer of CNN.

In this approach, we extracted features from the output layer. This layer consists of a sequence of predictions made by CNN and these predictions for each frame fed into the LSTM as input. The size of the input of LSTM depends on the number of classes.

C. LSTM With Pool Layer of CNN.

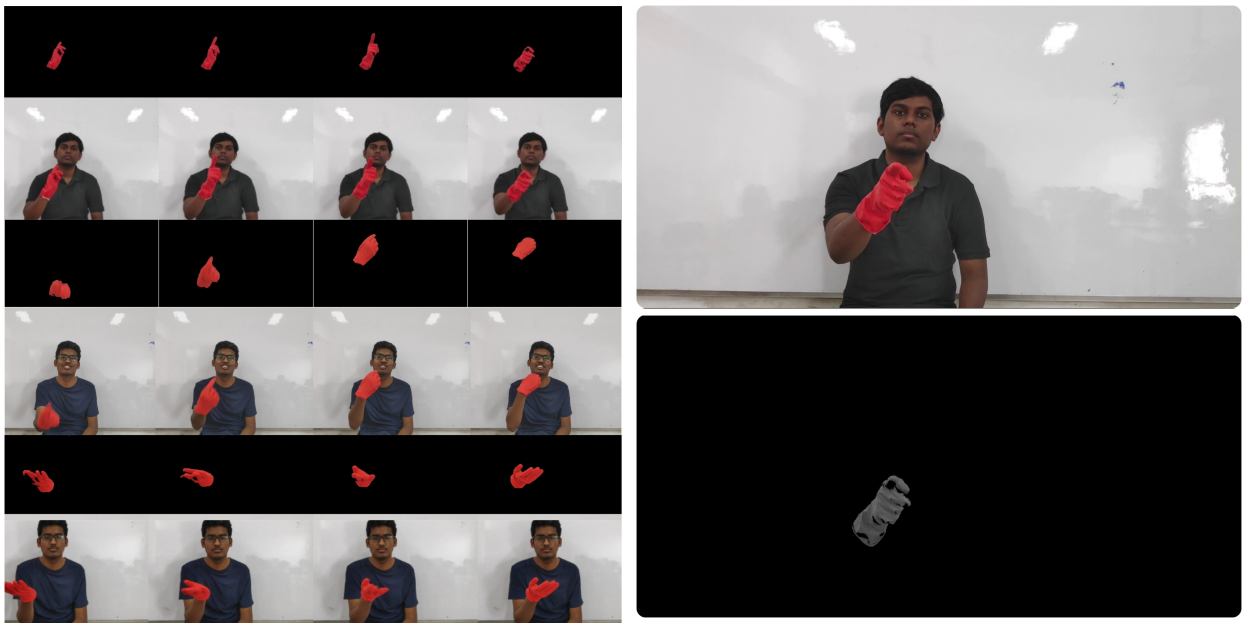
In this approach, we have used the output of the pool layer rather than the prediction layer to train the LSTM. This layer gives us a 2048 dimensional vector that consists of the convoluted features of the image, not the class prediction. In our case, the number of prediction is for four classes.

In Figure 3b a visual demonstration of sign word recognition technique has been presented.

V. Experiments and Results

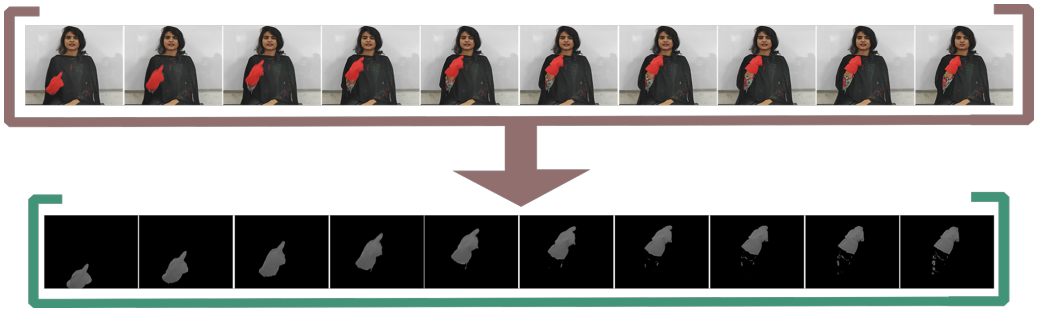
This section represents the experimental result of our systems implemented on our dataset. Our techniques are implemented in Tensorflow-GPU V1.5 and cuda V9.0. The experiment has been conducted on a machine having CPU CoreTM i7-7500U of 2.7 GHz, GPU Nvidia 1050GTX with 4.00GB and with 8.00GB memory on a Windows 10 operating system.

Our sign word recognition system trained Tensorflow based CNN InceptionV3 model on each of the frames to extract spatial features from the image as a 4-way classifier for 4 expressions: ‘ami (I)’, ‘tumi (You)’, ‘kemon_acho (how are you)’, ‘good (fine)’ – these signs are based on Bangladeshi sign language. We randomly split the frames into an $8 : 2$ ratio for training and testing set and the training set was further split to $8 : 2$ ratio for training and validation set. During training, frames were augmented for better learning. After 6k iterations, the network was tested with ground truth labels on test datasets. This same CNN trained on the same dataset twice to extract different layer features from the trained CNN. Firstly, the class prediction layer was the output layer. And then the 2048-dimensional pool layer is performed as the output layer and the input for the next layer. These values from CNN were fed into an LSTM model. For LSTM, we used Adam (Adaptive Moment Estimation) which is a stochastic optimizer and categorical cross-entropy as the loss function. The video dataset was also randomly split into train and test set with the ratio of $8 : 2$ to train the LSTM. It is to be noted that, all the videos were shot in a similar condition

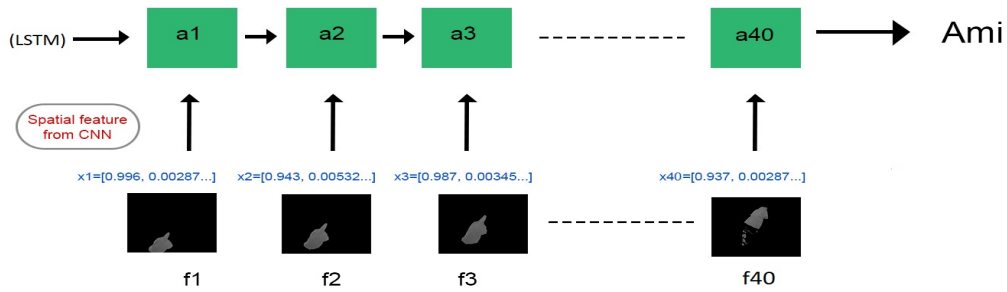


(a) (b)

Fig. 2: In (a), samples of video frames with before and after background removal of class ‘Kemon (How are you?)’, ‘Tumi (You)’ and ‘Ami (I)’ from bottom to top from our *Dataset_2* for videos. Hand segmentation has been done using RGB thresholding on each frame. In (b), the actual input frame converted to a gray–scale has been shown.



(a) Pre-processing on video dataset. Backgrounds are removed and each frame is converted to gray-scale image.



(b) This figure represents a visualization of how extracted frames spatial values from CNN model are passed in LSTM model to train.

Fig. 3: For training word recognition system, after hand segmentation, all frames are converted into gray-scale and then fed into a CNN. CNN extracts the spatial feature of each frame and pass it to LSTM as input and learn the temporal sequences of each word.

ID	Gesture	Video	Time Needed To Detect		Correctly Classified		Accuracy (%)	
			Approach1	Approach2	Approach1	Approach2	Approach1	Approach2
1	Ami	20			20	20	100	100
2	Tumi	20	About	About	20	20	100	100
3	Kemon_Acho	20	1.30	30	20	20	100	100
4	Valo_Achi	20	minutes	seconds	19	20	95	100

TABLE I: Comparisons and results of two approaches applied in sign word recognition system.

for simpler computational purposes due to the limitation of hardware specifications.

A. Class-Prediction Layer Output.

CNN's last regression layer was fed into the LSTM to learn the temporal features of the frame. Each video of words consists of 41 frames and each frame has 4 class prediction/spatial values from the trained CNN which were the input to the LSTM in this case. For this approach, we achieved 97.5% accuracy on our test set. But in real-time testing, where lighting conditions and subjects were different than the actual dataset, we find that the results are less accurate, and it took about 1.30 minutes to compute the result. This is because this approach only considering the class prediction of each frame rather than the actual features of the hand pose. Moreover, the length of the probabilistic prediction by CNN in the sequences of predictions depends on the total number of classes.

B. Pool Layer Output.

CNN's pool layer values were fed into the LSTM to learn the temporal features of the frame. Each video of words consists of 40 frames and each frame has a shape of 2048-dimensional spatial values which were the input to the LSTM in this case. For this approach, we achieved 100% accuracy on our test set. However, in real-time testing, where lighting conditions and subjects were different than the actual dataset, we find that the results more accurate with proper hand segmentation, and it took about 30 seconds to compute the result. This approach considers the pool layer values of each frame and surprisingly it generalizes better than the first approach. Initially, we were considering 40 frames for each gesture where it is assumed that the action of each word takes one second. And the system took about 30 seconds for recognizing each gesture. To reduce the recognition time, we considered 20 frames randomly and sequentially out of these 40 frames. And the recognition time was reduced to 15 seconds and the results were accurate too.

C. Real-Time System for Users.

Our system records the video from users and the user must specify the duration of gestures by giving input of start and end time of which portion of the video contains sign, for a system to predict from that real-time video. Our system can predict multiple words at a given time duration. Our system is tested for both of our two methodologies. We tested our system on various backgrounds by different users. Figure 4 demonstrates the results of real-time recognition of our system. Table I shows summarized comparisons and results of two approaches applied in the sign word recognition system. A demonstration of our real-time result is available here: `https://youtu.be/my5NY5QVQK0`.

VI. Conclusion and Future Work

In this paper, we have developed a system to recognize BdSL-Words in real-time. We have created our own dataset — *BdSLWord*— where we collected about 50 videos for each gesture where 5 subjects have participated. Each video was then represented by a sequence of predictions made by the CNN model for individual frames and this sequence of predictions was given as input to the RNN. The result is satisfactory for both of the models used but the second approach was suitable for real-time applications. The number of classes in our video dataset, made for video recognition is small, only for four signs. We need to improve our system on word recognition for detecting the words at a much shorter time as well.
For a feasible system for users, the system has to cope with multiple varying backgrounds and recognize the words without any added gloves. In the future, we have the plan to evaluate our model by genuine users to sort out its limitations and improve the system. This will also help us see how the system reacts to real-life situations and how clearly it can recognize the pattern and interpret it effectively.

References

[1] S. T. Ahmed and M. A. H. Akhand. Bangladeshi sign language recognition using fingertip position. pages 1–5, Dec 2016.

[2] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden. Subunets: End-to-end hand shape and continuous sign language recognition. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3075–3084, Oct 2017.

[3] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Spatiotemporal residual networks for video action recognition. *CoRR*, abs/1611.02155, 2016.

[4] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1933–1941, 2016.

[5] O. B. Hoque, M. I. Jubair, M. S. Islam, A. Akash, and A. S. Paulson. Real time bangladeshi sign language detection using faster r-cnn. In *2018 International Conference on Innovation in Engineering and Technology (ICIET)*, pages 1–6, 2018.

[6] S. Masood, A. Srivastava, H. Thuwal, and M. Ahmad. Real-time sign language gesture (word) recognition from video sequences using cnn and rnn. pages 623–632, 01 2018.

[7] R. Poppe. A survey on vision-based human action recognition. *Image Vision Comput.*, 28(6):976–990, June 2010.

[8] M. A. Rahaman, M. Jasim, M. H. Ali, and M. Hasanuzzaman. Computer vision based bengali sign words recognition using contour analysis. pages 335–340, Dec 2015.

Fig. 4: Some results of real-time detection of words in different backgrounds by different subjects. The user recorded video by pressing the snapshot button and then provide input of the start (t_1) and end (t_2) time from the recorded video, containing sign in that portion and then click the predict button to see the results. In (c) start time is 0 second and the end time is 2 second. These inputs are given by the user. In this output, the system has recognized two gestures at a time: 'Tumi' and 'Kemon_acho' (*How are you?* in English) which are displayed in the box by the system.

[9] M. A. Rahaman, M. Jasim, M. H. Ali, and M. Hasanuzzaman. Bangla language modeling algorithm for automatic recognition of hand-sign-spelled bangla sign language. *Frontiers of Computer Science*, page 0, 2018.

[10] M. A. Rahaman, M. Jasim, M. H. Ali, T. Zhang, and M. Hasanuzzaman. A real-time hand-signs segmentation and classification system using fuzzy rule based rgb model and grid-pattern analysis. *Frontiers of Computer Science*, page 0, 2017.

[11] A. Sherstinsky. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *CoRR*, abs/1808.03314, 2018.

[12] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pages 568–576, Cambridge, MA, USA, 2014. MIT Press.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, June 2016.

[14] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. *CoRR*, abs/1505.04868, 2015.

[15] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream convnets. *CoRR*, abs/1507.02159, 2015.