New keypoint based approach for recognising British Sign Language (BSL) from sequences

Oishi Deb, Prajwal Kondajji Renukananda, **Andrew Zisserman**

UNIVERSITY OF OXFORD

AIMS EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines & Systems

VGG University of Oxford

NEURAL INFORMATION PROCESSING SYSTEMS Workshop

Abstract

In this paper, we present a novel keypoint-based classification model designed for the recognition of British Sign Language (BSL) words within continuous signing sequences. Our model's performance is assessed using the BOBSL dataset, revealing that the keypoint-based approach surpasses its RGB-based counterpart in computational efficiency and memory usage. Furthermore, it offers expedited training times and demands fewer computational resources. To the best of our knowledge, this is the inaugural application of a keypoint-based model for BSL word classification, rendering direct comparisons with existing works unavailable.

Dataset

1. BOBSL dataset has been used which contains a total of **1,467 hours** of BBC episodes and 8162 sign words, hence making it a **8162**-class classification problem.
2. Each episode are 30 to 60 minutes long, and the videos in the dataset have a resolution of 444 × 444 pixels and a frame rate of 25 fps.

Keypoint Extraction and Representations

1. We retrieved keypoints using the Mediapipe library, specifically designed for human pose estimation. It has capability to extract keypoints in real time, even without the need for GPUs.
2. We extracted 33 pose keypoints, 21 keypoints for left and right fingers each, and 468 face keypoints.
3. Extracted 543 keypoints from over **132 Millions frames** in total i.e from 1,467 hours of videos.

Model Architecture and Hyper-parameters

1. We used keypoint vectors as a direct input to the Transformer model.
2. Our Transformer model is consists of 6 encoder layers, each equipped with 8 attention heads.
3. Our model uses 512- dimensional embeddings. To train our network, we utilize the Adam optimizer with a learning rate set to 1e-4.
4. We used Batch size of 128, and the training process is halted when the validation loss does not improve for 3 consecutive epochs.

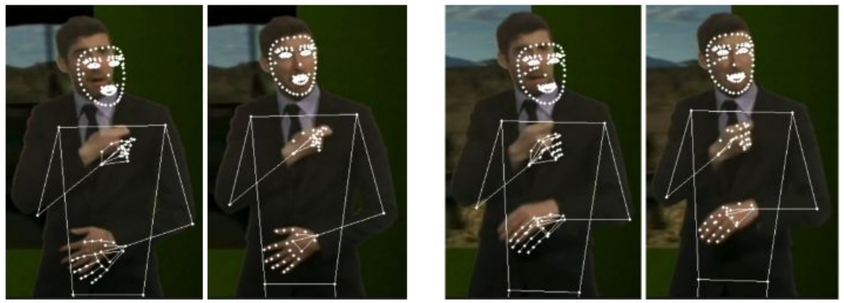
Experiments

1. 203-keypoints Model Frame-wise or Trajectory-wise without Augmentation; and with Augmentation.
2. 543-keypoints Model Frame-wise or Trajectory-wise without Augmentation; and with Augmentation.

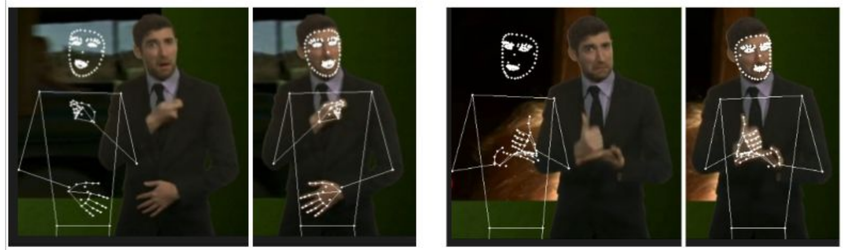
Results

1. **Accuracy:** We got over 60% Top5% accuracy whereas RGB model got 75% but given the fact that we are only using keypoints and no RGB frames, this is a commendable result.
2. **Computational Comparison:** Our model has 23.9 million parameters, compared to the 34.5 million parameters for the RGB model, hence we surpasses its RGB-based counterpart in computational efficiency and memory usage.

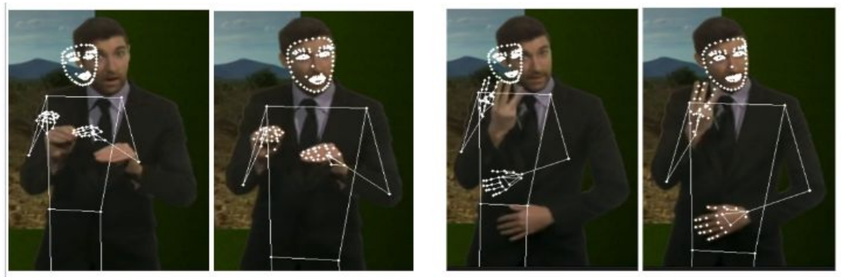
Data Augmentation



Shift augmentation is exemplified on two distinct frames, including the original keypoint representation prior to augmentation. For visualization clarity, the shift is depicted as 15 coordinates on both the x and y-axis. However, within the code, shifts range between -2 and 2. It's pertinent to note that while RGB images are overlaid for visualization, they are not utilized in model training.



Horizontally Flipped augmentation example on two different frames, it also shows the original keypoint representation before the augmentation.



Scale augmentation is illustrated using two distinct frames, with the original keypoint representation also displayed before augmentation. For visualization, the scale is depicted as 75 percent. However, in the actual code, scaling varies between 90 to 110 percent.

Rotated augmentation example on two different frames with the original keypoint representation before the augmentation. For visualization, it is rotated by 30 deg, however, in the code it varies by 5 deg.

Model Architecture