



This is a repository copy of *Graph neural network based child activity recognition*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/197815/>

Version: Accepted Version

Proceedings Paper:

Mohottala, S., Samarasinghe, P., Kasthurirathna, D. et al. (1 more author) (2023) Graph neural network based child activity recognition. In: 2022 IEEE International Conference on Industrial Technology (ICIT). 2022 IEEE International Conference on Industrial Technology (ICIT), 22-25 Aug 2022, Shanghai, China. IEEE , pp. 1-8. ISBN 9781728119489

<https://doi.org/10.1109/icit48603.2022.10002799>

© 2022, IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works. Reproduced in accordance with the publisher's self-archiving policy.

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

Graph Neural Network based Child Activity Recognition

Sanka Mohottala, Pradeepa Samarasinghe, Dharshana Kasthurirathna

*Faculty of Computing,
Sri Lanka Institute of
Information Technology
Sri Lanka*

sanka.m@slit.lk, pradeepa.s@slit.lk, dharshana.k@slit.lk

Charith Abhayaratne
*Department of Electronic and
Electrical Engineering,
University of Sheffield
United Kingdom*
c.abhayaratne@sheffield.ac.uk

Abstract—This paper presents an implementation on child activity recognition (CAR) with a graph convolution network (GCN) based deep learning model since prior implementations in this domain have been dominated by CNN, LSTM and other methods despite the superior performance of GCN. To the best of our knowledge, we are the first to use a GCN model in child activity recognition domain. In overcoming the challenges of having small size publicly available child action datasets, several learning methods such as feature extraction, fine-tuning and curriculum learning were implemented to improve the model performance. Inspired by the contradicting claims made on the use of transfer learning in CAR, we conducted a detailed implementation and analysis on transfer learning together with a study on negative transfer learning effect on CAR as it hasn't been addressed previously.

As the principal contribution, we were able to develop a ST-GCN based CAR model which, despite the small size of the dataset, obtained around 50% accuracy on vanilla implementations. With feature extraction and fine tuning methods, accuracy was improved by 20%-30% with the highest accuracy being 82.24%. Furthermore, the results provided on activity datasets empirically demonstrate that with careful selection of pre-train model datasets through methods such as curriculum learning could enhance the accuracy levels. Finally, we provide preliminary evidence on possible frame rate effect on the accuracy of CAR models, a direction future research can explore.

Index Terms—Child Action Recognition, Graph Neural Networks, Transfer Learning, Feature Extraction, Fine Tuning, Curriculum Learning

I. INTRODUCTION

Human Action Recognition (HAR) methods based on skeleton data have been widely investigated and received considerable attention due to high accuracy achieved on benchmark datasets such as NTU RGB+D [1] and Kinetics [2]. Recently most of the methods that achieve the state-of-the-art (SOTA) accuracy on benchmark datasets have been based on graph convolutional network (GCN) deep learning methods [3]–[5]. This accuracy increment can be attributed to many factors such as view invariance, robustness to occlusion and segmentation of the skeleton.

Spatial Temporal Graph Convolutional Network (ST-GCN) [3] architecture was the first to utilize graph neural networks for skeleton-based human activity recognition and had achieved the SOTA results outperforming CNN and LSTM

based models [3], [6]. Furthermore, it has been used with small and distinct datasets in different scenarios including fall detection [7], [8], hand sign detection [9], [10] and others [11], [12].

Child activity recognition (CAR) has important applications in video game development [13], early detection of autism [14], [15], safety monitoring [16], object-play behavior assessment [17] and many others. As most of the HAR models are based on adult datasets and as previous studies in motion style transfer [13] and pose estimation [18] have shown that due to differences in size, anatomy, and motion, these models can't be used for CAR, it is required to develop robust and well generalized CAR models.

Early CAR models as well as small dataset based CAR models have utilized signal processing methods along with classical machine learning methods [19], [20]. Deep Learning (DL) methods have been used lately with image based, skeleton based and wearable sensor based approaches through CNN [21], [22], LSTM [23], [24] and other methods [25], [26]. As GCN based models have not been applied in the past on CAR, we direct this research to fill that gap.

Though there are claims made on the availability of public datasets for CAR [27]–[30], few of those were not accessible due to non-availability and few were not capturing the whole body. Most of the existing DL research on CAR have not used the limited public datasets but has demonstrated the results on private datasets. Addressing this research gap, we have considered two standard Kinect camera based child activity datasets named Kinder-gator [31], Child-Whole-Body-Gesture (CWBG) [32] and a standard motion capture system based dataset named kinder-gator 2.0 [33] and we present the first evaluation of CWBG dataset on a SOTA DL model.

Though transfer learning (TL) can be used to overcome the limited dataset challenge in public datasets, [18], [25], [26] suggest that TL from adult to child is a challenging task resulting in undesirable performance. While there is a TL implementation with ST-GCN architecture [7] in HAR, it lacks a detailed study of different TL approaches and TL performance on ST-GCN architecture.

Based on the literature study and the gaps identified, in this paper we provide the following main contributions,

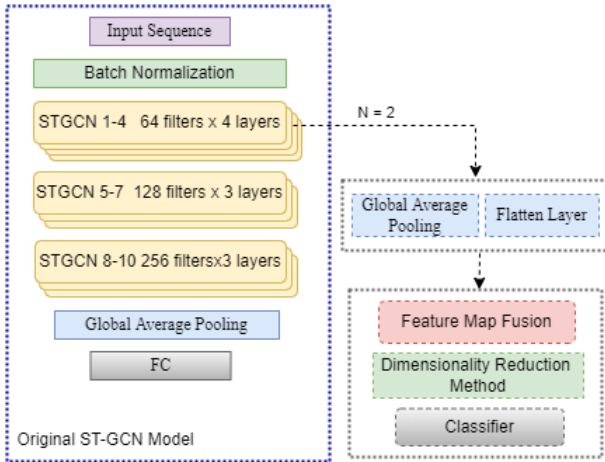


Fig. 1: ST-GCN model and feature extraction pipeline

- To the best of our knowledge, this is the first implementation of a GCN based model for child activity recognition.
- We provide the first benchmark results on a publicly available child activity dataset using ST-GCN model.
- Our research shows that comparable results can be achieved through transfer learning with ST-GCN model and demonstrate performance and comparative analysis with different learning approaches.
- A pre-train dataset selection method is introduced in this research to improve transfer learning and reduce negative transfer learning based on curriculum learning concept.

The rest of the paper is organized as follows. The models and methods used in the ST-GCN based implementations are discussed in Section II. Pre-processing process used with each dataset and the experimental setups used in each case are discussed in Section III. The performance of learning methods and the best results achieved by each method on different protocols are discussed in Section IV while Section V concludes with future research directions.

II. METHODOLOGY

Based on the ST-GCN original paper [3] and on the official ST-GCN PyTorch implementation [34], a TensorFlow based model was implemented and further experiments were carried out based on this model with NTU RGB+D dataset used for the pre-training models.

Two approaches were applied to quantitatively evaluate the ST-GCN model on the available child activity datasets:

- Standard Deep Learning: Train the ST-GCN model on the child activity datasets directly and do the hyperparameter tuning to attain a suitable model,
- Transfer Learning: Use a pre-trained ST-GCN model to leverage the learnt feature representations.

A. Learning Methods

1) *Standard Deep Learning*: Under this approach, ST-GCN model was directly trained with child activity datasets. Due to the small size of these datasets, the original ST-GCN model

with 256 channels in each of 10 ST-GCN layers could appear to contain excessive capacity. Since excessive capacity result in over-fitting [35], model tuning was also attempted with different number of filters as detailed in Section IV-A.

2) *Transfer Learning*: Lack of acceptable scale datasets generally results in poor performance of deep learning models. As all the openly available child activity datasets are small in size, developing improved learning methods was essential. Several TL approaches such as fine tuning [36]–[38], and feature extraction [39]–[41] have been used in the subsequent implementations to improve the performance of the model.

a) *Fine Tuning Method*: Fine tuning of the pre-trained ST-GCN model was done using several methods as detailed in TL literature [42], [43].

- 1) Frozen layer approach - Fine tuning n top ST-GCN layers, where $1 \leq n < 10$.
- 2) Hybrid approach - In the original ST-GCN model in Fig. 1, n top ST-GCN layers were randomly initialized where $1 \leq n \leq 10$.
 - Hybrid-Frozen : Combines feature extraction and standard deep learning together. Bottom $10 - n$ ST-GCN layers were kept frozen.
 - Hybrid-FineTuned : Combines fine tuning and standard deep learning together. Bottom $10 - n$ ST-GCN layers were fine tuned.

3) Propagation approach - Fine tuning all ST-GCN layers.

Since ST-GCN contains only a single layer classifier, initial experiments were done with a single randomly initialized dense layer. Experiments were later done with the FC classifiers with up to 4 dense layers. A dropout layer was used as a regularization method when overfitting occurred. Hyperparameter tuning was also applied to get the best accuracy.

b) *Feature Extraction Method*: Inspired by [44], [45], the feature representations were extracted from feature maps of the ST-GCN model (Fig. 1). The original ST-GCN model was then supplemented with either a flattening layer or a global average pooling (GAP) layer as the intermediate layer between the ST-GCN model and the classifier. Fusion of feature maps was done with consecutive maps from two and three layers of ST-GCN. Enhancing this approach further, dimensionality reduction techniques such as principle component analysis (PCA), truncatedSVD were employed. Experiments were done with support vector machine (SVM), logistic regression as well as feed forward neural network (FFNN) as the classifier.

3) *Curriculum Learning*: To improve the efficiency of both Standard Deep Learning and TL, Curriculum Learning (CL) [46] was applied, where the model was trained starting from easy samples and gradually exposing to more challenging samples. While there are several variants in implementing CL, our implementation contained Scoring and Pacing functions as in [47]:

- Scoring Function: gives the probability of level of difficulty of each sample, calculated by training the model with 10% of epochs.

- Pacing Function: decides the number of samples used in each epoch enabling us to add the samples based on the descending order of the values generated by the Scoring Function. This was implemented as a step function with varying number of steps.

B. Activity Datasets

A preliminary research was done on the existing child activity datasets and found that all three publicly available datasets, Kinder-Gator [31], Kinder-Gator 2.0 [48] and CWBG dataset [32] were depth sensor based and were available only in skeleton modality. In order to overcome the challenge in limited data sets, we used existing adult skeleton mode activity data sets, Kinect v2 based NTU-120 and NTU-60 for building the pre-trained models. NTU-120 was taken as the main dataset, which contains 850 videos with an average number of frames ranging from 76 to 300. Actions are performed by 106 participants above 10 years of age.

To improve the TL and to reduce the negative TL effect [49], several subsets of NTU were identified.

1) Large Scale Datasets:

- NTU-120: The full NTU RGB+D-120 dataset was used with a different dataset splitting method as detailed in Section III-A than the one proposed in [50]. With this approach, we were able to remove any potential bias resulting from an unbalanced data distribution.
- NTU-60: We used the full NTU RGB+D dataset along with 11 interaction classes.
- NTU-51: Out of the datasets of NTU-60, in order to minimize ambiguities in activity identification, 9 interaction classes were removed resulting in 49 single action classes and 2 interaction classes.

2) *Curriculum Learning Inspired Datasets*: While there are other methods [51], [52], we used a simplified CL inspired approach to select better classes. Best classes are chosen by analysing the confusion matrices of NTU-60/120 based STGCN models.

- NTU-44: Out of those sorted, 44 classes were selected. To reduce ambiguities introduced from spatial and temporal symmetrical classes, we kept only one such class in this subset.
- NTU-22: Enhancing the approach taken for NTU-44 further, a more discriminative dataset of 22 classes was introduced. Classes were selected by analysing the NTU-44 confusion matrix.

3) *NTU Frame Rate Adjusted (NTU-FRA)*: Since the CWBG dataset is recorded with an approximate 10FPS frame rate, we introduced three down-sampled data sets.

- NTU-44-FRA : Selected each of the 3rd frame in every sequence in NTU-44 subset.
- NTU-60-FRA : Selected each of the 3rd frame in every sequence in NTU-60 subset.
- NTU-120-FRA : Selected each of the 3rd frame in every sequence in NTU-120 subset.

C. Skeleton Structure

As the NTU dataset was created with Kinect v2 and CWBG dataset was created using Kinect v1, output skeleton structure generated in NTU and CWBG are different to each other.

Transfer learning when applied to convolution based models usually takes a input as a grid structure similar to images. Thus, when the target domain data samples differ in size from source domain, resizing with standard interpolation techniques can be utilized. But with graph structure, structural change has to be taken into consideration and knowledge transferability in GCN based models is still not fully developed [53]. Thus we use 20 shared joints with removal of 5 joints from NTU source dataset.

D. Child Dataset Protocols

The CWBG dataset is released for the study of child gesture elicitation and contains 1312 sequences from 30 children between the ages of 3 - 6 with an equal gender distribution.

- CWBG-Full: Contains 15 classes and is used to test the performance of entire dataset.
- CWBG-Similar: Contains 10 classes, removing most dissimilar classes. This group contains challenging classes even for a human (Table I - column Challenging).
- CWBG-Dissimilar: Contains 10 classes including most discriminative (Table I - column Discriminative).

III. EXPERIMENTS

This section discusses the stages followed in conducting the quantitative analysis including: pre-processing of the datasets, and the experimental settings.

A. Data Pre-Processing

In the first stage of data pre-processing, noisy data such as empty sequences and pseudo-skeleton sequences were removed. If a class belongs to a human-human interaction, then the action of the person with the most activity movements was selected in both NTU-60 and NTU-120 datasets whereas these ambiguous actions were removed in NTU-51, NTU-44, and NTU-22 datasets.

In the next stage, following process was followed for all the data sets.

- Fixed the frame size to 300, increasing feature visibility.
- First order information was extracted by translating each skeleton such that the spine joint position in each frame is the origin of coordinate system (i.e., [0,0,0]).
- Rotated the skeleton around spine joint such that person is looking towards positive x-axis and spine is parallel to z-axis.

In the final stage, dataset was split into train and validation subsets according to the cross subject method [1] for NTU-60 and CWBG based datasets. For additional classes other than NTU-60 in NTU-120 based activity datasets, training and validation was split randomly 70% to 30%.

TABLE I: Challenging and Discriminative Child Whole-Body Gesture (CWBG) Dataset classes

Challenging classes	Discriminative classes
Draw a circle	Hands up
Draw a square	Crouch
Draw a flower	Jump
Angry like a bear	Applaud

B. Implementation Details

1) *Standard Deep Learning*: Standard deep learning approach was implemented with original ST-GCN model without any changes to the architecture and other hyper-parameters were also kept same except for the learning rate scheduler. Further implementations were done to change the model capacity by adjusting the number of filters in each layer but the ratio of filter number between layers was kept at the same value as in original ST-GCN model. New layer filter number was changed by R , $R = \bar{F}_n/F_n$, where n refers to ST-GCN layer number ($1 \leq n \leq 10$) while \bar{F} refers to new model filters and F to the original model filters.

2) Transfer Learning:

a) *Pre-trained model*: The approach used for implementing ST-GCN in this research differs with the previous implementations in a number of ways.

- Since use of dropout and batch normalization together result in degrading results [54], dropout layers were not used as outlined in [3].
- The implemented TensorFlow based ST-GCN uses an equal weighting system for the graph rather than a trainable weight mask as in [3], since the mask weights are intrinsic to the dataset used for training. Hence those weights won't be adaptable in TL.

Training was done for 30 epochs, with a batch size of 4, using stochastic gradient descent (SGD) optimizer. The learning rate was initialized to 0.1 and was dropped to 0.01 and further to 0.001 using the piece-wise constant decay scheduler.

b) *Fine Tuning*: To evaluate fine-tuning approach, experiments were done using the NTU-44 dataset based pre-trained model. In frozen layer approach, FC layer as in Fig. 1 was replaced with a randomly initialized dense layer.

For the comparison of fine tuning approach on three main CWBG dataset protocols, implementations were done on all activity dataset based pre-trained models. Hyper-parameter tuning was done on each to achieve the best accuracy keeping other parameters constant.

c) *Feature Extraction*: All feature extraction approach evaluations were done using the NTU-44 dataset with same hyper-parameters as in fine-tuning evaluation. Performance evaluations were done based on the output layer, use of GAP layer, use of dimension reduction methods and the fusion of feature maps.

On implementing the SVM classifier, linear kernel function with L1 regularization constant set to 1 was applied. For the logistic regression implementation, multi-nominal logistic

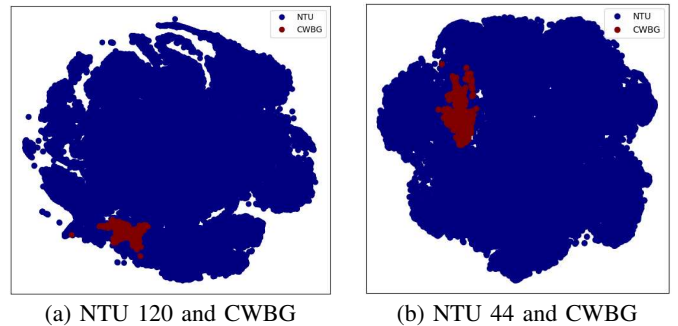


Fig. 2: NTU and CWBG dataset visualization

regression classifier was used with L2 regularization constant kept at 1. For the FFNN based classifier, experiments were done with different combinations of layers and nodes and for the evaluation results single layer classifier was used.

While the performance analysis was carried out with all three classifiers, based on the best performance, only FFNN classifier was used for the comparative analysis.

3) *Curriculum Learning*: Curriculum learning was implemented with feature extraction and fine tuning methods on the NTU-FRA dataset based pre-trained models as well as with standard deep learning approach. For pacing function, a non-uniform step function was used such that latter steps last longer (i.e., more epochs) than the previous steps. Same batch size and optimizer was used but learning rate was changed to account for the training behavior resulted from step function used as the pacing function.

Frame rate effect on the performance was evaluated by dropping the frames of NTU-44/60/120 datasets to match the CWBG dataset's frame rate. Experiments were done by taking the moving average of joint position vectors with sliding window of 5 time periods and 3 time periods in order to see if the noisy data of NTU had affected the performance of the model.

IV. RESULTS AND DISCUSSION

On the the ST-GCN model implementation, accuracy of 78.7% was achieved on the NTU RGB+D dataset. While the original authors of [3] achieves 81.5% accuracy on the cross-subject protocol, the change of accuracy(-2.8%) could be explained through the changes done on the ST-GCN architecture as detailed in Section III-B2a.

Visualization of NTU-120 and CWBG datasets as well as NTU-44 and CWBG datasets through t-distributed Stochastic Neighbourhood Embedding (t-SNE) (Fig. 2) shows overlapping distributions, confirming the potential use of TL to increase the accuracy as shown in [51].

A. Standard Deep Learning

In the vanilla ST-GCN model implementation with CWBG datasets, when the training accuracy converges to an optimal value, test datasets reach an accuracy in the range of 40%-55% (Table II). Such low accuracy is expected on a limited dataset such as CWBG, yet ST-GCN over performed Random

TABLE II: Standard Deep Learning Top-1 Accuracy

Child Datasets Method	CWDG-Full	CWDG-Similar	CWDG-Dissimilar
Vanilla ST-GCN	41.71	47.84	53.18
Vanilla Curriculum Learning	40.65	44.36	51.46
ST-GCN $R = 1/32$	39.64	42.29	47.19
ST-GCN $R = 1/2$	39.12	50.16	47.57
ST-GCN $R = 2$	27.46	44.71	57.68
Random	6.67	10	10

TABLE III: Curriculum Learning based feature extraction results on activity datasets

Accuracy	NTU-44-FRA	NTU-60-FRA	NTU-120-FRA
CWDG-Full	<u>58.62</u>	53.74	25.65
CWDG-Similar	<u>63.53</u>	62.14	33.56
CWDG-Dissimilar	<u>79.03</u>	76.78	44.57

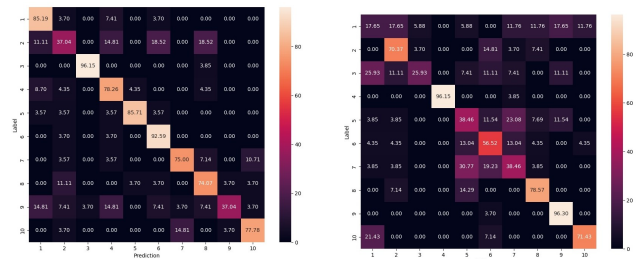
accuracy in multitudes. Given that model capacity is extremely high ($\approx 3M$ parameters) compared to dataset size, over-fitting should happen according to classical machine learning theory [35]. But the results from smaller capacity model implementations (for $R=1/32$, $\approx 5k$ parameters and for $R=1/2$, $\approx 700k$ parameters) in table II shows that rather than increasing the accuracy, smaller capacity slightly decreases the accuracy. This suggests that the low accuracy in the vanilla ST-GCN model is not due to over-fitting. These results are consistent with the existing research done on over-parameterized neural networks and the resultant high generalization accuracy [55], [56]. Another implementation done on a higher capacity model (for $R=2$, $\approx 12M$ parameters) shows this indeed is the case. Hence we can conclude vanilla ST-GCN results are the optimal results on all CWBG protocols.

B. Feature Extraction Results

Gradual increase in accuracy as the the output ST-GCN layer number reaches 10 in Fig. 4b is analogous to the conventional CNN model behaviour in feature extraction [57], thus confirming the existence of a hierarchical feature representation in ST-GCN architecture which is the fundamental requirement for TL applicability. Furthermore, similar increase of accuracy is present in the flatten layer implementation (Fig. 4a) as well as in feature map fusion implementation (Fig. 4c). Compared to the SVM and LogReg classifier feature extraction accuracy (around 65%-75%), when an untrained ST-GCN model is used, same classifiers give a lower accuracy (around 25%-30%) (Fig. 4b), thus empirically proving that the resultant feature accuracy is indeed due to the pre-trained model’s feature representations rather than the classifier.

SVM and logistic regression results can be considered as the upper bound of feature extraction accuracy. While 1-layer FFNN performs on par with SVM and LogReg on top ST-GCN layers, its poor performance on bottom ST-GCN layers suggests that feature maps are more linearly separable as layer number increases and this is to be expected in a model that learns hierarchical feature representation [57].

As detailed in Fig 4d and Figs 4a, 4c, even though dimensionality reduction and the use of flattening layers do not



(a) CWBG-Dissimilar protocol (b) CWBG-Similar protocol

Fig. 3: Feature Extraction Performance on NTU-44 based Model

TABLE IV: Feature extraction results on child datasets

Accuracy	NTU 60	NTU 120	NTU 51	NTU 44	NTU 22	NTU 44-FRA	NTU 60-FRA	NTU 120-FRA
CWDG-Full	47.98	<u>52.69</u>	49.48	51.26	47.41	57.77	52.85	<u>59.33</u>
CWDG-Similar	56.26	56.07	54.9	61.17	52.94	67.31	61.96	63.14
CWDG-Dissimilar	70.04	73.03	73.03	<u>74.16</u>	67.79	79.10	77.15	82.24

increase accuracy, feature map fusion contributes to accuracy improvement slightly.

C. Fine Tuning Results

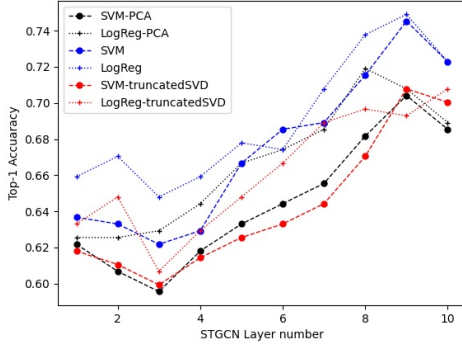
In evaluating performance between the two Hybrid and the Frozen approaches II-A2a, either of the Hybrid approaches always outperformed the fine tuning approach irrespective of the number of layers in the ST-GCN model (Table V). In order to compare the standard fine tuning approach and the Hybrid approach on the child datasets, Hybrid-Frozen and Vanilla Fine-Tuned models were chosen as the models in Table VI.

D. Curriculum Learning Results

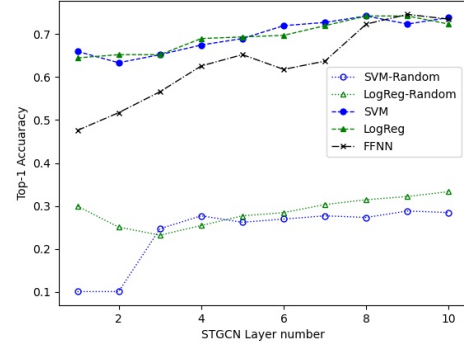
Vanilla curriculum learning in Standard Deep Learning approach did not result in a significant accuracy improvement (Table II). When CL is applied to feature extraction method (Table III), implementations on NTU-44-FRA and NTU-60-FRA results in similar accuracies as in original feature extraction results, yet there is either little improvement or no improvement. With NTU-120-FRA, the results are significantly worse than the original feature extraction results. Based on the model loss, there is significant overfitting and with the pacing function we used, it can’t be avoided. These results suggest that even though with CL some improvement can be achieved, it is difficult to tune the model to achieve those results and with more challenging datasets like NTU-120, it is even extremely difficult.

TABLE V: Layer-wise Fine Tuning Implementation

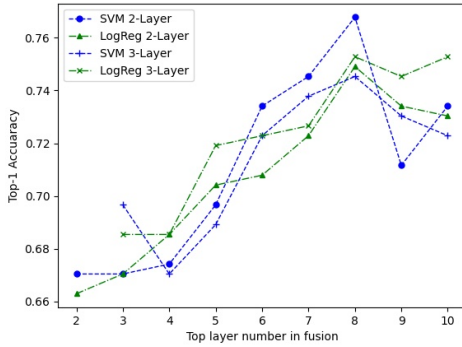
Fine-Tuning Approaches	ST-GCN layer number					
	4	5	6	7	8	9
Hybrid-Fine-Tuned	74.91	76.78	76.4	77.53	76.03	74.91
Hybrid-Frozen	76.78	77.9	75.66	74.53	76.03	75.28
Vanilla Fine-Tuned	74.96	75.28	73.41	74.53	71.54	72.28



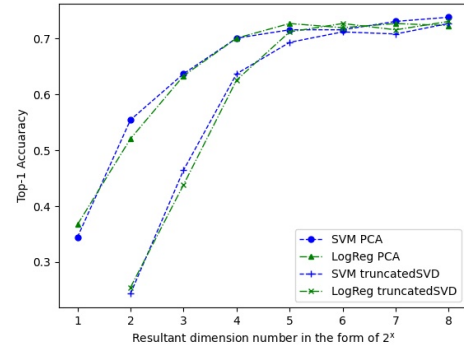
(a) Flattening layer approach



(b) Results on ST-GCN layer output



(c) Feature map fusion approach



(d) Dimensionality reduction approach

Fig. 4: Feature extraction performance with different configurations

TABLE VI: Fine-tuning comparison on child datasets

Accuracy		NTU 120	NTU 60	NTU 51	NTU 44	NTU 22	NTU 44-FRA	NTU 60-FRA	NTU 120-FRA
Vanilla	CWDG-Full	58.03	52.33	54.66	56.74	52.07	57.25	55.44	<u>57.51</u>
	CWDG-Similar	<u>66.27</u>	63.92	58.43	63.14	58.43	62.35	66.67	60.78
	CWDG-Dissimilar	<u>76.78</u>	69.66	68.54	75.28	67.04	74.91	76.4	<u>76.78</u>
Hybrid	CWDG-Full	58.55	57.25	55.3	55.7	51.55	54.14	55.44	<u>58.29</u>
	CWDG-Similar	65.88	63.92	61.38	63.92	57.65	63.14	65.49	65.1
	CWDG-Dissimilar	<u>77.53</u>	<u>78.28</u>	76.4	76.4	70.79	76.03	78.65	78.28

E. Comparative Results

Out of all the comparative results, the highest accuracy is shown with CWBG-Dissimilar protocol (Tables II, III, IV, and VI). This result is further corroborated through the generated confusion matrices and two such resultant confusion matrices for feature extraction approach are given in Fig. 3. Strong correlation is evident in the classes with labels 5-7 (Fig. 3 - (b)) representing the first three classes in Table I (column Challenging). Action visualization demonstrates label 1 class corresponding to "angry like a bear" doesn't result in considerable skeleton movements, explaining the misclassification. Better accuracies are observed for the more distinct classes of Fig. 3 - (a), endorsing that ST-GCN model achieves an acceptable accuracy in child action recognition.

Detailed feature extraction implementations were done on final layer output without dimensionality reduction and with

GAP layer (Table IV). Best accuracies are achieved by NTU-120-FRA based model for CWBG-dissimilar protocol and CWBG Full protocol while for the CWBG similar, NTU-60-FRA achieves the best accuracy. When 30FPS based activity datasets are compared, best accuracy for CWBG similar and CWBG dissimilar protocols are achieved by NTU-44, demonstrating our curriculum inspired dataset based model is suitable for TL just as large dataset based models. But the results on NTU-22 shows that when this approach is taken to an extreme with small number of classes, the resultant model's performance degrades.

Comparison of fine-tuning models was done with $n = 1$ as detailed in II-A2a. Results on vanilla fine-tuned models in Table VI suggest that unlike in feature extraction method, largest activity dataset (i.e., NTU-120) based model outperform all the other models including the NTU-44. Even when 30FPS-activity datasets and 10FPS-activity datasets are considered separately, this finding still holds. Comparison results on Hybrid-Frozen models in Table VI also confirms this interpretation. CWBG-Full protocol and CWBG-Dissimilar protocol perform better on the Hybrid-Frozen approach while CWBG-Similar protocols performs better with the Vanilla Fine-Tuned approach. Contrary to the other implementations [42], we achieved marginal improvement of Hybrid-Frozen approach over the vanilla Fine-Tuning approach.

TABLE VII: Benchmark results of CWBG protocols

Top-1 Accuracy	CWBG-Full	CWBG-Similar	CWBG-Dissimilar
All activity datasets	59.33	67.31	82.24
30FPS-activity datasets	58.55	65.88	78.28

To provide the benchmark results, performance of CWBG protocols across the learning methods was compared based on two criterion. One considering all the activity dataset based models and the other considering only the 30FPS-activity dataset based models (Table VII). While all the 30FPS-activity dataset results were derived from fine-tuning approach, the all activity dataset results were derived from the feature extraction approach. Furthermore, when each NTU-FRA result in Table IV is compared with its corresponding 30FPS NTU result, there is a significant increase in accuracy but such increase is not present in Table VI. These observations provides clear indication of frame rate effect in feature representation and further experiments are needed to evaluate its implications.

V. CONCLUSION AND FUTURE WORKS

This paper presents a CAR model based on GCN architecture and a detailed analysis with different learning methods. With the ST-GCN implementation, despite the small size of CWBG dataset, acceptable accuracies of 47.84% and 57.68% were achieved on CWBG-Similar and CWBG-Dissimilar, providing the first GCN based CAR model results and demonstrating the applicability of GCN in CAR.

Applying TL approaches such as feature extraction and fine-tuning, we were able to improve these accuracies to 65.88% and 78.28% on 30FPS-activity datasets and to 67.31% and 82.24% on all activity datasets. These results corroborate TL approaches like feature extraction and fine-tuning can improve accuracy in CAR despite contrary claims made by previous research.

Superior performance of NTU-44 based model over other 30FPS based models in feature extraction gives strong evidence of CL inspired dataset selection process use in TL but NTU-22 performance suggest there needs to be a balance between dataset size and dataset classes.

These results suggest several future research directions such as analysis of frame rate effect in TL, detailed study of CL parameter selection and effective methods to achieve a trade off between parameters to avoid negative TL. Moreover, to achieve scalability in practical implementations, a robust approach for TL with different skeleton structures is also essential.

ACKNOWLEDGMENT

This research was supported by the Accelerating Higher Education Expansion and Development (AHEAD) Operation of the Ministry of Higher Education of Sri Lanka funded by the World Bank (<https://ahead.lk/result-area-3/>).

REFERENCES

- [1] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1010–1019.
- [2] 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.
- [3] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-second AAAI conference on artificial intelligence*, 2018.
- [4] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 026–12 035.
- [5] L. Shi, Y. Zhang, H. Lu, and J. Cheng, "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 9532–9545, 2020.
- [6] W. Peng, J. Shi, T. Varanka, and G. Zhao, "Rethinking the ST-GCNs for 3D skeleton-based human action recognition," *Neurocomputing*, vol. 454, pp. 45–53, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231221007153>
- [7] O. Keskes and R. Noumeir, "Vision-based fall detection using st-gcn," *IEEE Access*, vol. 9, pp. 28 224–28 236, 2021.
- [8] Y. Zheng, D. Zhang, L. Yang, and Z. Zhou, "Fall detection and recognition based on gcnn and 2d pose," in *2019 6th International Conference on Systems and Informatics (ICSAI)*. IEEE, 2019, pp. 558–562.
- [9] C. C. de Amorim, D. Macêdo, and C. Zanchettin, "Spatial-temporal graph convolutional networks for sign language recognition," *Lecture Notes in Computer Science*, p. 646–657, 2019. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-30493-5_59
- [10] Y. Li, Z. He, X. Ye, Z. He, and K. Han, "Spatial temporal graph convolutional networks for skeleton-based dynamic hand gesture recognition," *EURASIP Journal on Image and Video Processing*, vol. 2019, no. 1, p. 78, Sep. 2019. [Online]. Available: <https://doi.org/10.1186/s13640-019-0476-x>
- [11] X. Cao, W. Kudo, C. Ito, M. Shuzo, and E. Maeda, "Activity recognition using ST-GCN with 3d motion data," in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, ser. UbiComp/ISWC '19 Adjunct. New York, NY, USA: Association for Computing Machinery, 2019, p. 689–692. [Online]. Available: <https://doi.org/10.1145/3341162.3345581>
- [12] M.-F. Tsai and C.-H. Chen, "Spatial temporal variation graph convolutional networks (stv-gcn) for skeleton-based emotional action recognition," *IEEE Access*, vol. 9, pp. 13 870–13 877, 2021.
- [13] Y. Dong, A. Aristidou, A. Shamir, M. Mahler, and E. Jain, "Adult2child: Motion style transfer using cyclegans," in *Motion, Interaction and Games*, ser. MIG '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: <https://doi.org/10.1145/3424636.3426909>
- [14] A. Zunino, P. Morerio, A. Cavallo, C. Ansuini, J. Podda, F. Battaglia, E. Veneselli, C. Becchio, and V. Murino, "Video gesture analysis for autism spectrum disorder detection," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 3421–3426.
- [15] Y. Zhang, Y. Tian, P. Wu, and D. Chen, "Application of Skeleton Data and Long Short-Term Memory in Action Recognition of Children with Autism Spectrum Disorder," *Sensors*, vol. 21, no. 2, 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/2/411>
- [16] J. Goto, T. Kidokoro, T. Ogura, and S. Suzuki, "Activity recognition system for watching over infant children," in *2013 IEEE RO-MAN*, 2013, pp. 473–477.
- [17] T. L. Westeyn, G. D. Abowd, T. E. Starner, J. M. Johnson, P. W. Presti, and K. A. Weaver, "Monitoring children's developmental progress using augmented toys and activity recognition," *Personal Ubiquitous Comput.*, vol. 16, no. 2, p. 169–191, feb 2012. [Online]. Available: <https://doi.org/10.1007/s00779-011-0386-0>
- [18] G. Sciortino, G. M. Farinella, S. Battiato, M. Leo, and C. Distanto, "On the Estimation of Children's Poses," in *Image Analysis and Processing - ICIAP 2017*, S. Battiato, G. Gallo, R. Schettini, and F. Stanco, Eds. Cham: Springer International Publishing, 2017, pp. 410–421.
- [19] J. Rehg, G. Abowd, A. Rozga, M. Romero, M. Clements, S. Sclaroff, I. Essa, O. Ousley, Y. Li, C. Kim *et al.*, "Decoding children's social behavior," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3414–3421.
- [20] A. Tsiami, P. Koutras, N. Efthymiou, P. P. Filintisis, G. Potamianos, and P. Maragos, "Multi3: Multi-sensory perception system for multi-modal

- child interaction with multiple robots,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4585–4592.
- [21] C.-D. Huang, C.-Y. Wang, and J.-C. Wang, “Human action recognition system for elderly and children care using three stream convnet,” in *2015 International Conference on Orange Technologies (ICOT)*. IEEE, 2015, pp. 5–9.
- [22] V. Silva, F. Soares, C. P. Leão, J. S. Esteves, and G. Vercelli, “Skeleton driven action recognition using an image-based spatial-temporal representation and convolution neural network,” *Sensors*, vol. 21, no. 13, p. 4342, 2021.
- [23] A. Dechemi, V. Bhakri, I. Sahin, A. Modi, J. Mestas, P. Peiris, D. E. Barundia, E. Kokkoni, and K. Karydis, “Babynet: A lightweight network for infant reaching action recognition in unconstrained environments to support future pediatric rehabilitation applications,” in *2021 30th IEEE International Conference on Robot Human Interactive Communication (RO-MAN)*, 2021, pp. 461–467.
- [24] J. Hadfield, G. Chalvatzaki, P. Koutras, M. Khamassi, C. S. Tzafestas, and P. Maragos, “A deep learning approach for multi-view engagement estimation of children in a child-robot joint attention task,” *CoRR*, vol. abs/1812.00253, 2018. [Online]. Available: <http://arxiv.org/abs/1812.00253>
- [25] S. Suzuki, Y. Mitsukura, H. Igarashi, H. Kobayashi, and F. Harashima, “Activity recognition for children using self-organizing map,” in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, 2012, pp. 653–658.
- [26] N. Efthymiou, P. Koutras, P. P. Filntisis, G. Potamianos, and P. Maragos, “Multi-view fusion for action recognition in child-robot interaction,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 455–459.
- [27] S. Lemaignan, C. E. R. Edmunds, E. Senft, and T. Belpaeme, “The PInSoRo dataset: Supporting the data-driven study of child-child and child-robot social dynamics,” *PLOS ONE*, vol. 13, no. 10, pp. 1–19, Oct. 2018, publisher: Public Library of Science. [Online]. Available: <https://doi.org/10.1371/journal.pone.0205999>
- [28] K. Sun, D. Yu, J. Chen, D. Yu, Y. Choi, and C. Cardie, “DREAM: A challenge dataset and models for dialogue-based reading comprehension,” *CoRR*, vol. abs/1902.00164, 2019. [Online]. Available: <http://arxiv.org/abs/1902.00164>
- [29] S. S. Rajagopalan, A. Dhall, and R. Goecke, “Self-stimulatory behaviours in the wild for autism diagnosis,” in *2013 IEEE International Conference on Computer Vision Workshops*, 2013, pp. 755–761.
- [30] E. Marinou, M. Zanfir, V. Olaru, and C. Sminchisescu, “3d human sensing, action and emotion recognition in robot assisted therapy of children with autism,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2158–2167.
- [31] A. Aloba, G. Flores, J. Woodward, A. Shaw, A. Castonguay, I. Cuba, Y. Dong, E. Jain, and L. Anthony, “Kinder-Gator: The UF Kinect Database of Child and Adult Motion,” in *EG 2018 - Short Papers*, O. Diamanti and A. Vaxman, Eds. The Eurographics Association, 2018.
- [32] R.-D. Vatavu, “The dissimilarity-consensus approach to agreement analysis in gesture elicitation studies,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–13. [Online]. Available: <https://doi.org/10.1145/3290605.3300454>
- [33] Y. Dong, A. Aristidou, A. Shamir, M. Mahler, and E. Jain, “Kinder-Gator 2.0, Optical motion capture, Dataset, MIG2020,” Oct. 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.4079507>
- [34] J. W. D. L. Sijie Yan, Yuanjun Xiong, “Mmskeleton,” <https://github.com/open-mmlab/mmskeleton>, 2019.
- [35] H. Wang, Z. Lei, X. Zhang, B. Zhou, and J. Peng, “Machine learning basics,” *Deep learning*, pp. 98–164, 2016.
- [36] M. Wang and W. Deng, “Deep visual domain adaptation: A survey,” *Neurocomputing*, vol. 312, p. 135–153, Oct 2018. [Online]. Available: <http://dx.doi.org/10.1016/j.neucom.2018.05.083>
- [37] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [38] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” in *International conference on artificial neural networks*. Springer, 2018, pp. 270–279.
- [39] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, “A comprehensive survey on transfer learning,” *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.
- [40] J. Zhang, W. Li, P. Ogunbona, and D. Xu, “Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.
- [41] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf: an astounding baseline for recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.
- [42] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” *Advances in neural information processing systems*, vol. 27, 2014.
- [43] J. Zhang, W. Li, P. Ogunbona, and D. Xu, “Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.
- [44] M. Awais, X. Long, B. Yin, C. Chen, S. Akbarzadeh, S. F. Abbasi, M. Irfan, C. Lu, X. Wang, L. Wang, and W. Chen, “Can pre-trained convolutional neural networks be directly used as a feature extractor for video-based neonatal sleep and wake classification?” *BMC Research Notes*, vol. 13, no. 1, p. 507, Nov. 2020. [Online]. Available: <https://doi.org/10.1186/s13104-020-05343-4>
- [45] S. Yu, Y. Cheng, S. Su, G. Cai, and S. Li, “Stratified pooling based deep convolutional neural networks for human action recognition,” *Multimedia Tools and Applications*, vol. 76, no. 11, pp. 13 367–13 382, Jun. 2017. [Online]. Available: <https://doi.org/10.1007/s11042-016-3768-5>
- [46] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.
- [47] G. Hachohen and D. Weinshall, “On the power of curriculum learning in training deep networks,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 2535–2544.
- [48] Y. Dong, A. Aristidou, A. Shamir, M. Mahler, and E. Jain, “Kinder-Gator 2.0, Optical motion capture, Dataset, MIG2020,” Oct. 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.4079507>
- [49] Z. Wang, Z. Dai, B. Póczos, and J. Carbonell, “Characterizing and avoiding negative transfer,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 293–11 302.
- [50] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, “NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, p. 2684–2701, Oct 2020. [Online]. Available: <http://dx.doi.org/10.1109/TPAMI.2019.2916873>
- [51] B. Wang, M. Qiu, X. Wang, Y. Li, Y. Gong, X. Zeng, J. Huang, B. Zheng, D. Cai, and J. Zhou, “A minimax game for instance based selective transfer learning.”
- [52] M. A. Ponti, L. S. F. Ribeiro, T. S. Nazare, T. Bui, and J. Collomosse, “Everything you wanted to know about deep learning for computer vision but were afraid to ask,” in *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, 2017, pp. 17–41.
- [53] Q. Zhu, C. Yang, Y. Xu, H. Wang, C. Zhang, and J. Han, “Transfer learning of graph neural networks with ego-graph information maximization,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.
- [54] C. Garbin, X. Zhu, and O. Marques, “Dropout vs. batch normalization: an empirical study of their impact to deep learning,” *Multimedia Tools and Applications*, vol. 79, no. 19, pp. 12 777–12 815, May 2020. [Online]. Available: <https://doi.org/10.1007/s11042-019-08453-9>
- [55] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning (still) requires rethinking generalization,” *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.
- [56] Z. Allen-Zhu, Y. Li, and Y. Liang, “Learning and generalization in overparameterized neural networks, going beyond two layers,” *CoRR*, vol. abs/1811.04918, 2018. [Online]. Available: <http://arxiv.org/abs/1811.04918>
- [57] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” in *International conference on machine learning*. PMLR, 2014, pp. 647–655.