An Action Recognition Approach with Context and Multiscale Motion Awareness

Danilo Barros Cardoso, Luiza C.B. Campos, and Erickson R. Nascimento
Computing Science Department, Universidade Federal de Minas Gerais (UFMG), Brazil
{danilo.cardoso, luiza.chagas, erickson}@dcc.ufmg.br

***Abstract*—Despite the substantial progress made by computer vision approaches in solving image classification, object detection, and pose estimation, to name a few, activity recognition remains one of the key challenges in computer vision and pattern recognition. This paper proposes a new learning framework based on multiscale spatiotemporal graph convolution layers and a transformer architecture. Even though several approaches present high accuracy in more traditional datasets like NTU, their performance significantly drops when tested in datasets with a high level of ambiguity among activities and an unbalanced number of samples for each class. We evaluated our architecture in the challenging BABEL dataset, where we achieved state of the art in terms of accuracy (65.4%) in action classification when considering both ambiguity and class unbalance. The source code and trained models are publicly available at https://github.com/verlab/AnActionRecognitionApproach_SIBGRAPI_2022.**

I. Introduction

Humans are very good at making sense of what is going on with movements and assigning labels to the observed action. We have an innate ability to understand human behavior by analyzing a small set of human poses. Although in the past decade we have witnessed the performance gap between humans and computers, in many tasks such as image classification [1], scene recognition [2], and object detection [3], the same does not hold for action recognition - especially when recent and large-scale video datasets such as BABEL [4] are concerned. Although high accuracy values have been achieved in datasets with simple actions (*e.g.*, NTU [5], [6], MSR Action3D Dataset [7]–[9], and Kinetics [10]), the proposal of a new dataset with more natural and complex human motions has led several methods to a significant drop in performance.

Human action recognition techniques face several challenges. Foremost is the high level of motion ambiguity when classifying natural and routine human movements. For instance, more general actions such as interacting with or using an object are easily misclassified with specific actions with finer and low-scale motions like taking or picking something up. Run and jog actions, for their turn, have very similar motions for the joints and links in a human skeleton, albeit they are easily separable when considering the spatiotemporal relation of joints. Recent advances on human pose estimation [11], [12], self-attention layers [13], and graph architectures [14]–[16] have shed new light on action recognition approaches. However, modeling the multiscale nature and spatiotemporal relation of the skeleton joints remains a key challenge in the action recognition field.

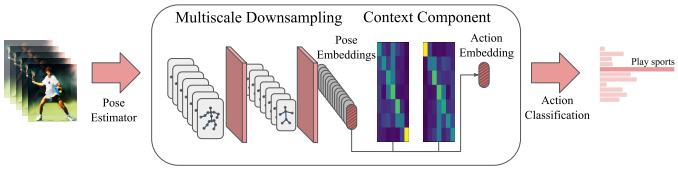


Fig. 1. Learning to recognize actions with ambiguous motions. Our architecture is based on a set of downsampling operations that are applied to the skeleton graphs until they are reduced to a series of pose embeddings. The sequence of feature vectors and a summary token feed the Transformers layers, which give context to the poses. Finally, the summary token is given as input to a classifier that infers the action class.

In this paper, we present a new action recognition framework based on a Transformer [13] architecture and multiscale spatiotemporal graph convolution layers. Our learning approach applies a stage-wise strategy to train our architecture, where we first pre-train a multiscale encoder-decoder model with self-supervision in an auto-regressive task to create rich representations for a sequence of poses. Then, the feature vector extracted from the encoder is provided as input to a classifier. Our encoder uses a transformer architecture, which enriches the pose representation with context information. The experimental results show that our architecture is superior to the state-of-the-art methods in key metrics when using the challenging BABEL dataset. In summary, our contributions are as follows:

- We investigate the use of a stage-wise strategy in an action recognition problem;
- A multiscale component to support capturing fine-grained movements;
- We present an architecture that uses the multiscale component and Transformers layers to mix multiscale motion and context into an action embedding.

II. Related Work

A. Skeleton Based Action Recognition

The usage of skeleton and joint trajectories to classify an action has advantages over other methods thanks to its robustness to illumination change and scene variation. Yan *et al*. [14] proposed the Spatial-Temporal Graph Convolutional Network (ST-GCN) for skeleton-based action recognition. Their technique innovates by modeling poses into a graph with temporal connections and applying a series of convolutions in space (single pose) and time dimensions.

Li *et al*. [17] introduced two structures: actional links that learn action-specific dependencies and structural links, which extend the existing skeleton graphs connections with higher-order dependencies. Their proposed layer, the Actional-Structural Graph Convolutional Network (AS-GCN), is a combination of the two structures that, when stacked, can be used for action recognition. Another extension was proposed by Shi *et al*. [18] where structural information, *i.e*., the adjacency matrix, is complemented by a set of learned weights capable of establishing higher-order topology connections.

Cheng *et al*. [15] adapted the shift convolution operation from CNN architectures to graphs, improving the results while using fewer parameters. More recently, Chen *et al*. [16] proposed the Multi-Scale Spatial Graph Convolution Module (MST-GCN). It increases the receptive field of the model in spatial and temporal dimensions by partitioning the feature space and applying convolutions in cascade, each time adding a new partition and increasing the receptive field.

B. Transformers in Vision Data

The Transformer architecture, proposed by Vaswani *et al*. [13], is currently the basic build block for Natural Language Processing (NLP) models due to its power of linking context between words in a sentence, even in situations in which two related words are separated by many others. It is based on an encoder-decoder structure where its components are self-attention blocks. Since its proposal, significant advances have been achieved in the NLP area. Devlin *et al*. [19] used the Transformer power in a language representation model called BERT, designed to pre-train deep bidirectional representations from unlabeled text, and then fine-tune the model to a specific task with just one additional output layer.

The Transformer's success inspired many researchers to apply its model components in computer vision tasks. Lu *et al*. [20] proposed ViBERT connecting NLP and computer vision in task-agnostic self-supervised pre-trained models that produce rich representations which can be fine-tuned later. Dosovitskiy *et al*. [21] explored the usage of Transformers blocks in computer vision tasks with only minor changes to adapt the input, an image, to a series of tokens. Wu *et al*. [22] proposed The Visual Transformer, a model to tokenize the image feature map into semantic groups through self-attention. For human action recognition, Mazzia *et al*. [23] proposed a method entirely based on the Transformer. The skeleton's keypoints are mapped to the Transformer input space by a simple linear transformation and a class token is added to the beginning of the sequence to aggregate information through the layers. The class token final representation serves as input for the classifier. Plizzari *et al*. [24] used the self-attention blocks to replace ST-GCN architecture spatial convolution (GCN) and temporal convolution (TCN) components by spatial-transformer and temporal Transformer, respectively.

In our work, we aim to exploit the benefits of graph convolutional networks and the Transform architecture to model multiscale features and the context in motion to improve the robustness of action recognition in the presence of ambiguous activities and an unbalanced dataset.

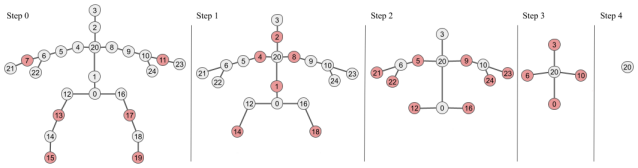


Fig. 2. Downsampling operation. The skeleton graph is reduced by each layer k accordingly the steps above and condensed into a single feature vector. The eliminated nodes in each step are highlighted in red.

III. Methodology

Our architecture is based on an encoder-decoder structure, where the encoder is composed of a multiscale downsampling and a context component. While the first is responsible for transforming a sequence of poses into a sequence of feature vectors, the latter is based on a Transformer encoder that enriches the pose representations with context information. The decoder, for its turn, has three components: the multiscale downsampling, a Transformer decoder, and an upsampling layer. The downsampling weights in both encoder and decoder branches are shared, since they have the same objective. The decoder receives the action embedding from the encoder branch as memory, and the upsampling layer predicts the next pose considering the sequence of poses received as input.

A. Multiscale downsampling

Let $\mathbf{X_0} \in R^{T\times V\times C}$ be an input tensor, where T is the number of poses, V is the number of vertices of the skeleton graph, and C is the number of spatial dimensions. In our case $C = 6$, *i.e*., the Euclidean coordinates of the 3D position of the node and its 3D velocity between two frames.

The multiscale downsampling is a series of k layers that gradually reduces the number of nodes V of the input tensor $\mathbf{X_k}$, simplifying the skeleton graph, but keeping semantic information necessary to describe the pose. In our case the pose skeletons are reduced from $V_0 = 25$ in the input tensor, $V_1 = 19$ after layer $k = 0$, $V_2 = 13$ after layer $k = 1$, $V_3 = 5$ after $k = 2$ and $V_4 = 1$ after $k = 3$. Figure 2 shows the downsampling scheme adopted in our implementation.

Each layer k is composed of two modules. First, the data passes through an aggregation step implemented by a graph convolution network that is responsible for extracting features regarding skeleton pose and motion. The aggregation module can be any spatiotemporal convolution block such as those proposed by Yan *et al*. [14], Li *et al*. [17], Shi *et al*. [18], and Chen *et al*. [16]. We use the AGCN convolution block proposed by Shi *et al*. [18] and defined as

$$h_k = W_{hk} X_k (A_k + B_k + C_k), \tag{1}$$

where $\mathbf{h_k}$ is the tensor after the aggregation operation, $\mathbf{W_{hk}}$ are weights learned by the network, $\mathbf{A_k}$ is the adjacency matrix of the graph that models the skeleton in layer k, and $\mathbf{B_k}$ is a

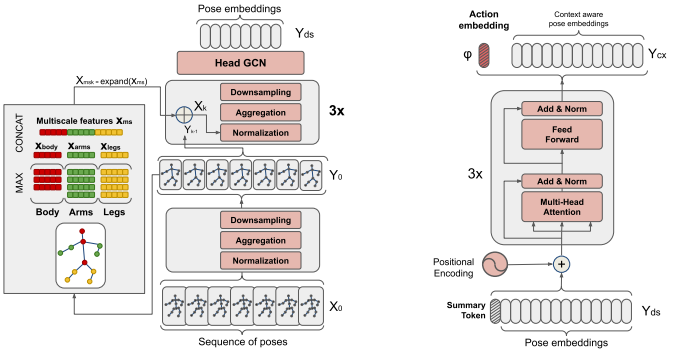


Fig. 3. Multiscale Downsampling and Context components. The downsampling component (left) receives a sequence of poses that are downsampled to a single vector. After the downsampling step, features from the body, arms, and legs are extracted and max-pooled. The features are concatenated to form the multiscale features that are injected into all nodes in all subsequent layers and producing pose embeddings. The pose embeddings feed a stack of transformer layers together with a summary token (right). The transformer encoder enriches each token with context information and the summary token collects context information in each layer to produce the action token.

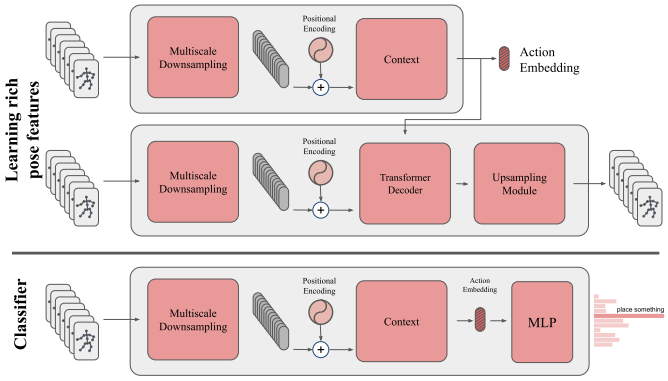


Fig. 4. Learning rich pose features. The model architecture during pre-training is composed of an auto-regressive encoder-decoder pair whose objective is to reconstruct a sequence of poses. The encoder is connected to the decoder by an action token which is trained to encode the whole action into a single vector. In the classification step, the decoder branch is dropped, and an MLP is added after the encoder. The classifier uses the action token as input and outputs score values for each class.

matrix of weights learned by the model to allow the network establish connections between nodes that are not linked by $\mathbf{A_k}$. $\mathbf{C_k}$ matrix measures similarities between two node feature vectors and acts like an attention mechanism, considering that nodes with similar features should communicate with each other. The matrix $\mathbf{C_k}$ is given by

$$C_k = softmax(X_k^T W_{\theta k}^T W_{\phi k} X_k), \tag{2}$$

which applies a linear transformation of the input vector by the parameters $\mathbf{W}_{\theta\mathbf{k}}$ and the weights $\mathbf{W}_{\phi\mathbf{k}}$ learned in the training.

The second step is the downsampling module that is responsible for simplifying the skeleton. It decreases the number of nodes from V_k to V_{k+1} by applying the operation

$$Y_k = W_{sk} A_k h_k, \tag{3}$$

where $\mathbf{W_{sk}}$ is a weight matrix of dimension $V_{k+1} \times V_k$ and $\mathbf{Y_k}$ is the output tensor of layer k.

The multiscale features are extracted from the output tensor $\mathbf{Y_0}$, and injected into the upstream layers. Let V_a, V_l and V_b be the set of node indexes i in dimension V_1, representing arm nodes, leg nodes, and body nodes. The multiscale features are constructed through the following rule:

$$\begin{cases} x_{arms} = \max(y_{0i}) & \forall i \in V_a, \\ x_{legs} = \max(y_{0i}) & \forall i \in V_l, \\ x_{body} = \max(y_{0i}) & \forall i \in V_b. \end{cases} \tag{4}$$

Then each component is concatenated to form the final multiscale vector x_{ms}, as

$$x_{ms} = x_{arms} \oplus x_{legs} \oplus x_{body}, \tag{5}$$

where $\oplus$ is the operation of concatenation.

The vector x_{ms} is stored internally and, in each subsequent layer, it is expanded into the matrix $\mathbf{X_{ms_k}}$ to meet the number of nodes in each pose inputted into that layer. Finally, next layer input is given by

$$X_k = X_{ms_k} \oplus Y_{k-1}. \tag{6}$$

where the expanded vector is concatenated to the previous layer output. The multi-scale downsampling process is illustrated in Figure 3.

B. Context Component

The context component is a transformer's encoder-decoder pair as proposed by Vaswani *et al*. [13]. In our method, a stack of three identical layers composes the encoder. Each layer is composed of two sub-layers: a multi-head self-attention mechanism and a fully connected feed-forward network. The decoder is also composed of three identical layers. Besides the sub-layers already described for the encoder layer, it adds a third sub-layer, which performs multi-head attention over the output of the encoder stack.

The layer's input is the sequence of pose embeddings $\mathbf{Y_{ds}}$ produced by the downsampling component. A summary token is concatenated to the sequence of pose embeddings, whose values are parameters learned by the network. The token is used to collect and summarize context data from transformers layers and therefore, describe the complete action. Finally, a positional encoding vector is added to the embeddings, as proposed by Vaswani *et al*. [13]. It allows the network to know the relative position of each token in the sequence.

The encoder is present both in the pre-training and in the classification phase. The encoder outputs a sequence of arrays, where the first vector is the summary token after passing through the transformer's layers and collecting context data. We call this token Action Embedding Token. The remaining vectors are pose embeddings enriched by context information.

The decoder is present only during the pre-training phase. It receives a series of pose embedding produced by the downsampling component as input. In the decoder branch,

the poses are shifted back one time step, and a subsequent mask is applied to the self-attention block to model an auto-regressive task. The encoder information is provided by the action embedding token, which is inputted to the encoder-decoder attention block of each decoder layer. The output of the decoder is a sequence of arrays, one for each pose, with features capable of describing the pose with sufficient information to reconstruct the pose.

C. Upsampling layer

The upsampling block is used in the pre-training phase. It generates a pose skeleton in three-dimensional space from a pose embedding outputted by the decoder. The upsampling component is similar to the downsampling component but the upsampling operation replaces the downsampling operation. This operation is responsible for increasing the number of nodes in the graph following the inverse order of the down-sampling. The upsampling operation is defined by

$$Y_{uq} = W_{uq} A_q h_q, \tag{7}$$

where W_{uq} is a weight matrix of dimension $V_{q+1} \times V_q$ and Y_{uq} is the output tensor of upsampling q-th layer.

D. Training and classification

We adopted a stage-wise training strategy. First, we train the model in a self-supervised and auto-regressive manner to create rich representations for a sequence of poses stored in the tensor $\mathbf{X_0}$. Then, we extract the encoder branch and add a final layer responsible for the activity recognition. Figure 4 depicts an overview of these two steps in our training phase.

The output of the encoder branch can be interpreted as an action embedding that summarizes the whole action. Therefore, we feed an MLP classifier with the action embedding. We used two fully connected layers as classifier, where the last layer is responsible for producing logits that classify the action. The complete classification architecture is represented in the bottom row of Figure 4.

IV. Experiments

In this section, we first present the dataset, testing setup, and implementation used in the experiments. Then, we present and discuss the results, comparing our method with the state of the art. Finally, we present the ablation study to analyze the effect of each component of our architecture.

A. Dataset

In our experiments, we used the BABEL [4] dataset for action recognition. BABEL is a large dataset with language labels describing actions captured by mocap sequences. Its objective is to support research focused on understanding human movement semantics. The authors proposed a pre-processed version of the dataset as a new benchmark for action recognition, on account that current state-of-the-art models reach above 95% of accuracy in the most used benchmark datasets - NTU RGB+D 60 [5] and NTU RGB+D 120 [6] - leaving short space to measure progress. The pre-processed

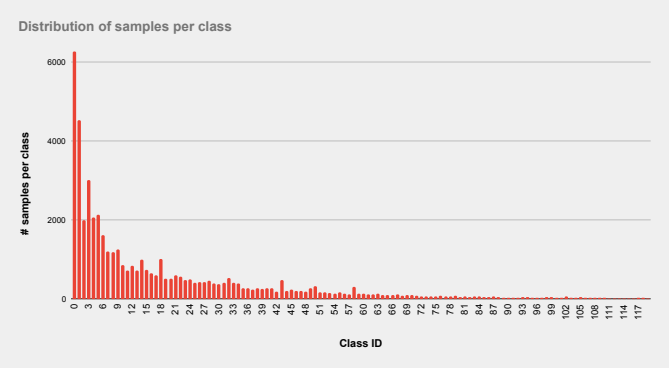


Fig. 5. Distribution of samples per class, showing that BABEL is a very unbalanced dataset. The first ten classes amount to over 50% of all samples.

version of the dataset has the characteristics desired for this work, *i.e.*, a high level of ambiguity and an unbalanced distribution of samples, as shown in Figure 5.

The BABEL dataset for action recognition is organized into two subsets. BABEL 60 with $45{,}473$ samples representing the 60 most common action labels of the original dataset and BABEL 120 with 120 classes and $48{,}978$ samples that comprises same samples from BABEL 60, plus $3{,}505$ additional samples of the following 60 most common classes of the original dataset.

B. Data pre-processing

Each pose skeleton was normalized by transforming the coordinates such that the middle spine is fixed in origin, shoulder blades are parallel to the X-axis, and the spine to the Y-axis, following methods proposed by Shahroudy *et al*. [5] and used by Punnakkal *et al*. [4]. All pose sequences are sampled at 30 fps and truncated to five seconds. After normalization, velocity features are calculated and concatenated to the input tensor. The velocity is calculated by subtracting node spatial features of two adjacent temporal frames.

C. Implementation details

For the pre-training phase, we used the L1 loss to measure the distance of the predicted position of a given pose key-point in 3D space to ground-truth coordinates. For the fine-tuning, we experiment using focal loss [25]. Focal loss compensates for class imbalance by up-weighting the cross-entropy loss for inaccurate predictions. The experiment could be early stopped to prevent overfitting. In both phases, we use the AdamW optimizer [26] with a learning rate of 0.00005, batch size of 64 samples, and a maximum of 300 epochs.

D. Results

To analyze the method, we used the same metrics of Punnakkal *et al*. [4]. Top-1 measures the accuracy of the most activated class. Top-5 measures if the correct class ranks among the top five highest-scoring predictions. Top-1-norm is the mean Top-1 accuracy across classes.

Since Top-5 accuracy accounts for labeling noise and ambiguity, it is considered the best indicator to evaluate the model

TABLE I
RESULTS - FOCAL LOSS.

	Variation	Top-5	Top-1	Top-1-Norm
BABEL 60	Dataset Benchmark [4]	69.0	34.0	30.0
	ST-GCN [14]	44.2	24.2	14.4
	2s-AGCN [18]	67.8	33.8	30.4
	MST-GCN [16]	70.3	36.3	**35.4**
	Ours	**70.4**	**36.4**	30.3
BABEL 120	Dataset Benchmark [4]	59.0	29.0	23.0
	ST-GCN [14]	28.6	20.5	5.5
	2s-AGCN [18]	58.0	27.9	26.2
	MST-GCN [16]	60.1	29.9	**29.8**
	Ours	**65.4**	**31.4**	28.4

TABLE II
ABLATION STUDY.

		Top-5	Top-1	Top-1-Norm
BABEL 60	No pretrain	69.7	34.5	30.3
	Transformer (1 layer)	68.2	34.2	**31.4**
	Transformer (2 layer)	69.2	33.8	31.2
	No multiscale mechanism	66.2	33.2	29.4
	Complete model	**70.4**	**36.4**	30.3
BABEL 120	No pretrain	63.5	29.8	25.3
	Transformer (1 layer)	61.9	29.9	26.8
	Transformer (2 layer)	64.8	30.3	**28.6**
	No multiscale mechanism	61.2	29.0	24.7
	Complete model	**65.4**	**31.4**	28.4

for BABEL. The results for Top-5 using focal loss can be seen in Table I. Our methods could outperform competitors by 0.1 p.p. for the BABEL 60 and 5.3 p.p. for the BABEL 120.

Our method also presented significant results for the Top-1 metric when optimizing with focal loss. Our method outperformed all competitors for both BABEL 60 and BABEL 120 (0.1 p.p. and 1.5 p.p., respectively). The results for the Top-1 metric with focal loss optimization are summarized in Table I.

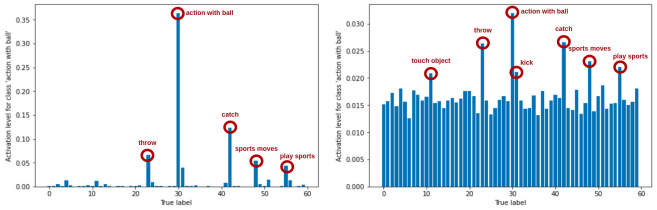
For the Top-1 metric with accuracy normalized by the number of samples we see a good performance, but not enough to outperform all competitors. Our method can surpass or equal the ST-GCN and 2s-AGCN methods in both versions of the dataset. However, we perform below MST-GCN by 4.1 p.p and 2.0 p.p for BABEL-120 and BABEL 60, respectively. It is worth mentioning that although this metric considers dataset imbalance, it cannot measure performance between ambiguous classes. The results for this metric are compiled in Table I.

E. Ablation Study

In the ablation study, we examine several parts of our architecture in order to analyze the impact of each component in the results. First, we execute experiments directly for action recognition to evaluate the contribution of the pre-training for the final result. Then, we analyzed the effect of the transformer, changing the size of the stack of transformer blocks. Finally, we remove the multiscale mechanism to verify its effect. The results are reported in Table II.

The results show that, especially in the metrics taken as a reference, *i.e.*, the Top-5 and Top-1 accuracy, all components contribute to the result. The element with the highest impact on main metrics was the use of the multiscale component. This mechanism alone contributed to an increase of 4.2 p.p. in Top-5 accuracy, both for the BABEL-60 and BABEL-120 datasets, when trained with focal loss. In the case of the Top-1 accuracy, the result is a gain of 3.2 p.p. and 2.4 p.p. for BABEL-60 and BABEL-120, respectively.

The use of pre-training shows a relevant effect on the final result, since the results was consistent in both versions of the dataset. For the BABEL-60 dataset, one can see an increase of 0.7 p.p. in the Top-5 and 1.9 p.p. accuracy. In the case of the Top-1 accuracy, the use of pre-training had no observable effect on the normalized Top-1 metric. For BABEL-120, pre-training was responsible for an increase of 1.9 p.p. in performance measured by the Top-5 metric and 1.6 p.p in the performance measured by the Top-1 metric.



(a) Cross-entropy (b) Focal loss

Fig. 6. Comparison between the activation averages of the 'Actions with ball' class samples. a) Model trained using cross-entropy loss. Classes with similar semantics have unbalanced activation levels, indicating difficulty in dealing with ambiguity. b) Model trained with focal loss. Actions with similar semantics have similar activation, indicating better ambiguity handling.

F. Dealing with ambiguity

A relevant aspect of our work is how to tackle ambiguity. In this section, we present a qualitative analysis seeking a better understanding of the results and why using focal loss is fundamental when dealing with ambiguity, and what this means in practical terms. For that, we observed the accumulated activation profile of selected class samples when presented to models trained with cross-entropy compared to models trained with focal loss. The activation profile is built from all samples of a specific class inputted in a trained model. The activation levels outputted by each sample accumulate, resulting in a histogram.

Figure 6 shows the result of this analysis for the class 'action with ball'. It is noteworthy that for the model trained using cross-entropy, a peak stands out from 'action with ball' class bin. Classes with similar meanings have significantly less intense activation. We argue that this difference in semantic terms is not justified, *i.e.*, the level of certainty presented by the network does not match the real meaning of the available options. On the contrary, when we observe the activation profile for a model trained using focal loss, classes with close semantics also show relevant activation. Furthermore, we

observe that the profile represents better the movement itself, with the inclusion of action classes like 'kick' in the set of highly activated.

V. Conclusion

In this paper, we propose a new learning framework based on multiscale features and the context of motions. The experimental results showed that the use of pre-training and the proposed multiscale mechanism contributed to improving the overall performance in the BABEL dataset, particularly when analyzing the results of Top-5 trained with focal loss. Since this metric is the only one that considers, at the same time, the imbalance and ambiguity of the dataset, the fact that we outperformed the other competitors by a significant margin shows the effectiveness of the proposed approach.

Moreover, the use of focal loss was fundamental in achieving the objective of dealing with ambiguous motions, as it allowed the model to similarly represent classes with similar semantics, as demonstrated in the activation histograms.

Acknowledgments

The authors would like to thank CAPES, CNPq, and FAPEMIG for funding different parts of this work. We also thank NVIDIA for the donation of a Titan XP GPU.

References

[1] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[2] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu, "CoCa: Contrastive Captioners are Image-Text Foundation Models," 2022.

[3] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection," *arXiv preprint arXiv:2203.03605*, 2022.

[4] A. R. Punnakkal, A. Chandrasekaran, N. Athanasiou, A. Quiros-Ramirez, and M. J. Black, "BABEL: Bodies, Action and Behavior with English Labels," in *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Jun. 2021, pp. 722–731.

[5] 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.

[6] 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, pp. 2684–2701, 2020.

[7] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010, pp. 9–14.

[8] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, and M. F. M. Campos, "STOP: Space-time occupancy patterns for 3d action recognition from depth map sequences," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 252–259.

[9] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, and M. F. Campos, "On the improvement of human action recognition from depth map sequences using space–time occupancy patterns," *Pattern Recognition Letters*, vol. 36, pp. 221–227, 2014.

[10] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The Kinetics Human Action Video Dataset," *CoRR*, vol. abs/1705.06950, 2017.

[11] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," in *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.

[12] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation," *arXiv preprint arXiv:2204.12484*, 2022.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[14] 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*, ser. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.

[15] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-Based Action Recognition With Shift Graph Convolutional Network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 183–192.

[16] Z. Chen, S. Li, B. Yang, Q. Li, and H. Liu, "Multi-Scale Spatial Temporal Graph Convolutional Network for Skeleton-Based Action Recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1113–1122.

[17] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-Structural Graph Convolutional Networks for Skeleton-based Action Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3595–3603.

[18] 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 Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 026–12 035.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[20] J. Lu, D. Batra, D. Parikh, and S. Lee, "ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations*, 2021.

[22] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, and P. Vajda, "Visual Transformers: Token-based Image Representation and Processing for Computer Vision," *arXiv preprint arXiv:2006.03677*, 2020.

[23] V. Mazzia, S. Angarano, F. Salvetti, F. Angelini, and M. Chiaberge, "Action Transformer: A Self-Attention Model for Short-Time Pose-Based Human Action Recognition," *Pattern Recognition*, vol. 124, p. 108487, 2022.

[24] C. Plizzari, M. Cannici, and M. Matteucci, "Spatial Temporal Transformer Network for Skeleton-based Action Recognition," in *ICPR International Workshops and Challenges: Virtual Event*. Springer, 2021, pp. 694–701.

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[26] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *ICLR*, 2018.