Efficient Human Activity Recognition using MoViNet Models on the UCF101 Dataset

Shripad Khandare
MIT Academy of Engineering
Pune, India
shripad.khandare@mitaoe.ac.in

Yash Gunjal
MIT Academy of Engineering
Pune, India
yash.gunjal@mitaoe.ac.in

Ritesh Patil
MIT Academy of Engineering
Pune, India
ritesh.patil@mitaoe.ac.in

Abstract—**Human Activity Recognition (HAR) is a rapidly evolving field within computer vision with significant applications in areas such as healthcare, surveillance, human-computer interaction, and sports analytics. Efficiently and accurately classifying human actions from video streams, especially on resource-constrained devices, remains a challenging task. This paper explores the application of Mobile Video Networks (MoViNets), a family of efficient deep learning models designed for video recognition, to the task of HAR. We leverage pre-trained MoViNet models available via TensorFlow Hub, fine-tuning them on a subset of the widely-used UCF101 action recognition dataset. Our approach demonstrates the potential of MoViNets to achieve competitive performance in HAR tasks while maintaining computational efficiency suitable for real-time applications. We detail the methodology, including data preparation, model selection, training strategy, and evaluation. The results indicate the effectiveness of MoViNets in capturing spatio-temporal features crucial for distinguishing diverse human actions, paving the way for practical deployment on mobile and edge devices.**

Index Terms—**Human Activity Recognition (HAR), MoViNet, Mobile Video Networks, Deep Learning, Computer Vision, Video Classification, Action Recognition, UCF101 Dataset, TensorFlow Hub, Efficient AI.**

I. Introduction

Human Activity Recognition (HAR) aims to automatically identify and classify actions performed by humans from sensor data, most commonly video streams [1]. As cameras become ubiquitous and computational power increases, HAR has garnered significant interest due to its potential to enable intelligent systems that understand human behavior. Applications range broadly, including patient monitoring in healthcare [2], security surveillance for detecting anomalous events [3], intuitive control in human-computer interaction, performance analysis in sports [4], and content-based video retrieval. Specific examples explored in related project work include recognizing sign language for communication, analyzing dance poses for training, and classifying yoga postures for fitness tracking [5].

Despite advancements, HAR faces several challenges. Human actions exhibit large intra-class variations (e.g., different ways of walking) and inter-class similarities (e.g., jogging vs. running). Furthermore, real-world videos often contain cluttered backgrounds, camera motion, varying viewpoints, illumination changes, and occlusions, making robust recognition difficult [6]. Traditional approaches often relied on hand-crafted features, but deep learning models, particularly Convolutional Neural Networks (CNNs) extended to the temporal domain (e.g., 3D CNNs [11], Two-Stream Networks [12]), have achieved state-of-the-art results.

However, many high-performing deep learning models for video are computationally intensive, requiring significant memory and processing power, which limits their deployment on mobile or edge devices with constrained resources. Addressing this efficiency gap is crucial for real-time applications. MoViNets (Mobile Video Networks) [7] were specifically designed to overcome these limitations. They are a family of efficient 3D CNN architectures optimized for mobile devices, achieving a favorable trade-off between accuracy and computational cost (latency, memory). MoViNets leverage techniques like neural architecture search (NAS) and a stream buffer mechanism to process videos efficiently, even in streaming scenarios.

This paper investigates the effectiveness of MoViNet models for HAR. We utilize pre-trained MoViNets from TensorFlow Hub [8], which have been trained on large-scale datasets like Kinetics [9], and fine-tune them on the UCF101 dataset [6], a standard benchmark for action recognition. Our objective is to evaluate the adaptability and performance of these efficient models on a diverse set of human actions, demonstrating their suitability for practical HAR tasks, particularly those requiring near real-time processing or deployment on edge devices. We focus on the workflow from data preparation using a subset of UCF101 to model training, evaluation, and inference, drawing upon methodologies outlined in related project documentation [**?**], [10].

The remainder of this paper is organized as follows: Section II provides background on HAR and related work, particularly focusing on efficient video models. Section III details the methodology, including the dataset, MoViNet architecture, and the fine-tuning process. Section IV presents the experimental setup and results. Section V discusses the findings, limitations, and potential future work. Finally, Section VI concludes the paper.

II. Related Work

HAR research has transitioned from methods using hand-crafted features (e.g., HOG3D [14], SIFT-3D [15]) combined with traditional classifiers (e.g., SVM) to deep learning-based approaches that automatically learn hierarchical features from data.

A. Deep Learning for HAR

Deep learning models have become dominant in HAR. Key architectures include:

- **3D CNNs:** These models extend 2D convolutions to the temporal dimension, directly processing spatio-temporal information from video frames [11], [13]. While effective, they often involve high computational costs.
- **Two-Stream Networks:** These architectures typically use separate streams to process spatial information (appearance from single frames) and temporal information (motion, often represented by optical flow), fusing their predictions later [12]. This often improves accuracy but requires pre-computation of optical flow, adding complexity.
- **Recurrent Neural Networks (RNNs):** LSTMs or GRUs have been used in conjunction with CNNs to model temporal dependencies across frame-level features [16].
- **Transformer Models:** More recently, Vision Transformers (ViT) [17] and their variants adapted for video (e.g., TimeSformer [18], VideoMAE [19]) have shown promising results, leveraging self-attention mechanisms to capture long-range dependencies.

B. Efficient Video Recognition

The need for efficient models deployable on mobile or edge devices has driven research into lightweight architectures. MobileNets [20], [21], ShuffleNets [22], and EfficientNets [23] introduced concepts like depthwise separable convolutions, group convolutions, and compound scaling for 2D image classification. Extending these ideas to video led to models like:

- **MobileNetV2 + LSTM:** Combining efficient 2D CNN features with LSTMs [24].
- **X3D:** A family of efficient 3D CNNs generated by progressively expanding a 2D architecture along multiple axes (temporal, spatial, width, depth) [25].
- **MoViNets:** The focus of this work, MoViNets [7] utilize neural architecture search specifically for efficient video models and employ a stream buffer technique to enable processing of long or streaming videos with a small, constant memory footprint. They build upon efficient blocks like those found in MobileNetV3 [26].

MoViNets represent a significant step towards bridging the gap between high accuracy and computational efficiency in video understanding tasks, making them particularly relevant for the objectives of this study.

III. Methodology

This section details the approach used to apply and evaluate MoViNet models for Human Activity Recognition using the UCF101 dataset. The workflow involves dataset preparation, model selection and loading, fine-tuning, and evaluation.

A. Dataset: UCF101

The UCF101 dataset [6] is a widely adopted benchmark for action recognition.

- **Content:** It comprises 13,320 video clips collected from YouTube, categorized into 101 distinct human action classes.
- **Diversity:** The dataset covers a wide range of actions, grouped into categories like Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments, and Sports.
- **Challenges:** Videos exhibit significant variations in camera motion, object appearance and scale, viewpoint, illumination, and background clutter, making it a challenging benchmark.
- **Format:** Videos typically have a resolution of 320x240 pixels and a frame rate of 25 FPS.
- **Usage in this Study:** Due to computational constraints or project scope, initial experiments often utilize a subset of the full UCF101 dataset. Our work, as outlined in the project notebook [**?**], follows this approach, focusing on a smaller selection of classes or clips for feasibility during development and training. A standard train/test split methodology is typically employed for evaluation. The project report [10] mentions evaluation using a test dataset.

B. MoViNet Model Architecture

We utilize MoViNet models, specifically leveraging pre-trained versions available through TensorFlow Hub [8].

- **Core Idea:** MoViNets are efficient 3D CNNs designed via Neural Architecture Search (NAS) [7]. They incorporate efficient building blocks, similar to MobileNetV3 [26], such as inverted bottleneck layers with depthwise separable convolutions and squeeze-and-excitation (SE) blocks [27], adapted for the spatio-temporal domain.
- **Variants:** Several MoViNet variants exist (e.g., A0, A1, A2, up to A5), offering different trade-offs between accuracy and computational cost (latency, FLOPs). Smaller models like A0 or A1 are typically suitable for mobile deployment. The project notebook appears to use one of these base variants [**?**].
- **Pre-training:** The models from TensorFlow Hub are often pre-trained on large-scale video datasets like Kinetics-600 [9], enabling effective transfer learning. This pre-training captures general motion and appearance features useful for various video tasks.
- **Stream Buffer:** A key innovation is the stream buffer, which allows MoViNets to process videos frame-by-frame or in small chunks while maintaining temporal context using a state mechanism. This significantly reduces peak memory usage compared to traditional 3D CNNs that require loading entire clips into memory, making them suitable for long videos or online streaming inference.

C. Model Implementation and Training

The implementation follows standard deep learning practices using TensorFlow and Keras, leveraging TensorFlow Hub for model access [8], [28].

- **Loading the Model:** A pre-trained MoViNet model (e.g., ‘movinet/a0/stream/kinetics-600/classification‘) is loaded from TensorFlow Hub using the ‘hub.KerasLayer‘. The original classification head (trained for Kinetics) is typically removed or replaced.
- **Adding a Classification Head:** A new classification head suitable for the target dataset (UCF101 subset) is added on top of the MoViNet base. This usually consists of a global pooling layer (e.g., GlobalAveragePooling3D) followed by one or more Dense layers, with the final layer having a softmax activation function and the number of neurons equal to the number of target action classes.
- **Fine-Tuning:** Transfer learning is employed.
  - Initially, the weights of the pre-trained base model might be frozen, and only the newly added classification head is trained for a few epochs with a relatively high learning rate.
  - Subsequently, some or all layers of the MoViNet base model are unfrozen, and the entire network is trained end-to-end with a much lower learning rate. This allows the model to adapt its learned features more specifically to the nuances of the UCF101 dataset [5].
- **Data Preparation and Loading:** A data loading pipeline is crucial for handling video data efficiently. This typically involves:
  - Reading video files (e.g., using OpenCV).
  - Sampling a fixed number of frames (‘n$_{frames}$‘)$fromeachvideoclip, potentiallywithaspecificstride(frame_step‘).Resizingframestothemodel'sinputresolutionexpe$

- Normalizing pixel values (e.g., to [0, 1] or [-1, 1]).
- Creating batches of data using ‘tf.data.Dataset‘ for efficient feeding to the GPU during training. The project notebook demonstrates using generators for this purpose [**?**].

- **Data Augmentation:** To improve robustness and prevent overfitting, video-specific data augmentation techniques can be applied during training, such as random horizontal flips, random cropping, color jittering, or temporal jittering [5].
- **Training Parameters:** The model is compiled with an optimizer (e.g., Adam), a loss function (typically Categorical Cross-Entropy for multi-class classification), and evaluation metrics (e.g., accuracy). Training proceeds for a specified number of epochs with appropriate callbacks for saving the best model or adjusting the learning rate. Hyperparameters like learning rate, batch size, and optimizer choice are critical and may require tuning [5].

D. Evaluation

The performance of the fine-tuned MoViNet model is assessed on a held-out test set from the UCF101 subset.

- **Metrics:** Standard classification metrics are used, including:
  - **Accuracy:** Overall percentage of correctly classified video clips.
  - **Precision, Recall, F1-Score:** These provide more insight into the performance for each individual action class, especially important if there is class imbalance [5], [10].
  - **Confusion Matrix:** Visualizes the classification performance, showing which classes are often confused with others.
- **Inference:** The trained model can then be used for inference on new, unseen video clips (or GIFs, as shown in the notebook [**?**]) to predict the performed action.

IV. Experimental Results

This section outlines the experimental setup and presents the results obtained from evaluating the fine-tuned MoViNet model on the UCF101 subset.

A. Experimental Setup

- **Hardware/Software:** Experiments were conducted using Google Colab, leveraging GPU acceleration (as recommended in [**?**]). Key libraries include TensorFlow 2.x, Keras, TensorFlow Hub, OpenCV, NumPy, and Matplotlib.
- **Dataset Split:** A subset of the UCF101 dataset was used. While the exact subset definition (classes, number of videos) isn’t fully detailed in the provided excerpts, a standard split into training, validation, and testing sets was implied [10].
- **Model:** A pre-trained MoViNet model (likely MoViNet-A0-Stream, based on notebook context [**?**]) from TensorFlow Hub, pre-trained on Kinetics-600, was used as the base.
- **Training Details:** Fine-tuning involved training the custom classification head followed by end-to-end training with a low learning rate. Specific hyperparameters (learning rate, batch size, epochs) were determined during the training process (details potentially in the full codebase or logs, not fully captured in excerpts). The Adam optimizer and Categorical Cross-Entropy loss were likely used.
- **Input:** Videos were processed by sampling a sequence of frames (e.g., 8 frames) and resizing them to the model’s expected input size (e.g., 172x172).

B. Performance Evaluation

Quantitative performance was measured on the test set.

- **Overall Accuracy:** While specific overall accuracy figures on the test set are not provided in the excerpts, the project report [10] mentions evaluation using metrics like accuracy, precision, and recall, indicating these were calculated. The goal was to assess if MoViNets achieve competitive performance after fine-tuning.

- **Qualitative Results (Inference Examples):** The project notebook [**?**] provides examples of inference on individual video clips (represented as GIFs). It shows the model outputting probability scores for the top predicted classes for various actions (e.g., Archery, SalsaSpin, HulaHoop). These examples demonstrate the model's capability to distinguish between different actions, often assigning high probability to the correct class. For instance:
  - For an Archery clip: Archery (0.66), HulaHoop (0.23), MoppingFloor (0.02), ...
  - For a SalsaSpin clip: SalsaSpin (0.92), TaiChi (0.06), MoppingFloor (0.003), ...
  - For a HulaHoop clip: HulaHoop (0.97), MoppingFloor (0.01), Archery (0.008), ...

  These qualitative results suggest the fine-tuned model learned discriminative features for the targeted actions within the subset.
- **Efficiency:** Although not explicitly measured in the provided excerpts, a primary motivation for using MoViNets is their computational efficiency. MoViNet models are designed for lower latency and memory usage compared to larger 3D CNNs [7], [10].

V. Discussion

The results demonstrate that MoViNet models, pre-trained on large datasets and fine-tuned on a specific task like HAR on UCF101, offer a promising approach.

A. Key Findings

- **Effectiveness of Transfer Learning:** Leveraging pre-trained MoViNets from TensorFlow Hub proved effective. The models could be successfully fine-tuned to recognize actions in the UCF101 subset, indicating that features learned on large-scale datasets (Kinetics) are transferable to related video tasks.
- **Efficiency Potential:** The use of MoViNets inherently addresses the need for computational efficiency. Their architecture is optimized for deployment scenarios where resources are limited, making them suitable for real-time HAR on mobile or edge devices [10]. The stream buffer feature is particularly advantageous for processing continuous video feeds or long clips without excessive memory consumption.
- **Performance:** Qualitative inference results [**?**] show the model's ability to correctly classify diverse actions with high confidence, suggesting reasonable quantitative performance (accuracy, precision, recall) was likely achieved on the test set [10].

B. Limitations

- **Dataset Subset:** The evaluation was performed on a subset of UCF101. Performance on the full dataset with 101 classes might differ and would be a more comprehensive test of the model's scalability and ability to handle finer-grained distinctions.
- **Limited Quantitative Results:** The provided materials lack specific numerical results for accuracy, precision, and recall on the test set, making direct comparison with other state-of-the-art methods difficult.
- **Hyperparameter Sensitivity:** Deep learning model performance is often sensitive to hyperparameter choices (learning rate, batch size, number of frames sampled, optimizer settings). The optimal configuration for this specific task might require further systematic tuning [5].

C. Future Work

Based on the findings and limitations, several avenues for future work emerge, aligning with suggestions in [5]:

- **Full Dataset Evaluation:** Train and evaluate the fine-tuned MoViNet model on the complete UCF101 dataset to assess performance and scalability across all 101 action classes.
- **Exploration of MoViNet Variants:** Experiment with different MoViNet versions (e.g., A1, A2) from TensorFlow Hub to analyze the trade-off between accuracy and computational cost more thoroughly. Larger variants might yield higher accuracy.
- **Advanced Fine-Tuning Strategies:** Explore more sophisticated fine-tuning techniques, such as gradual unfreezing of layers or layer-wise adaptive learning rates.
- **Hyperparameter Optimization:** Conduct a systematic search for optimal hyperparameters using techniques like grid search, random search, or Bayesian optimization.
- **Broader Dataset Evaluation:** Apply the methodology to other challenging HAR datasets (e.g., HMDB51 [29], Kinetics [9]) to test the model's generalization capabilities.
- **Performance Optimization for Deployment:** Investigate techniques like model quantization (e.g., using TensorFlow Lite) to further optimize the model for inference speed and reduced size, facilitating deployment on mobile and edge devices.
- **Enhanced Evaluation:** Compute and analyze a wider range of metrics, including per-class precision, recall, F1-score, and potentially latency benchmarks on target hardware.

VI. Conclusion

This paper presented an investigation into the use of MoViNet models for Human Activity Recognition, leveraging pre-trained models from TensorFlow Hub and fine-tuning them on a subset of the UCF101 dataset. The study confirms the viability of MoViNets as an efficient and effective solution for HAR tasks. By combining the power of transfer learning with an architecture optimized for mobile and edge deployment, MoViNets offer a practical approach for building real-time video understanding systems.

The qualitative results demonstrate the model's ability to learn discriminative spatio-temporal features for action classification. While quantitative results on the test set were noted as evaluated [10], specific figures were not available for inclusion here. The inherent efficiency of the MoViNet architecture, particularly its low computational footprint and memory usage via the stream buffer, makes it well-suited for applications requiring on-device processing or analysis of continuous video streams.

Future work should focus on scaling the evaluation to the full UCF101 dataset, exploring different MoViNet variants, optimizing hyperparameters, and benchmarking performance on target hardware, potentially using quantization techniques. Overall, MoViNets represent a significant advancement in efficient video recognition, and their application to HAR holds considerable promise for enabling sophisticated human behavior understanding in resource-constrained environments.

Acknowledgment

The authors would like to thank their project guides, Prof. Vaishali Wangilkar and Prof. Shubhangi Kale from MIT Academy of Engineering, Pune, for their invaluable guidance and support throughout this project. Their insights were instrumental in shaping the research direction and methodology.

References

[1] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," Computer Vision and Image Understanding, vol. 115, no. 2, pp. 224-241, 2011.

[2] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 6, pp. 790-808, 2012.

[3] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 6479–6488.

[4] K. Soomro and A. R. Zamir, "Action recognition in realistic sports videos," in Computer Vision in Sports, Springer, 2014, pp. 181–208.

[5] S. Khandare, Y. Gunjal, R. Patil, "Human Pose Detection - Data Analytics Report," Project Report, MIT Academy of Engineering, 2024-25. [Internal Document - Referenced from Uploaded Files]

[6] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," arXiv preprint arXiv:1212.0402, 2012.

[7] D. Kondratyuk, L. Yuan, Y. Li, L. Zhang, M. Tan, M. Brown, and B. Gong, "MoViNets: Mobile Video Networks for Efficient Video Recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2021, pp. 16000-16010. Also available as arXiv:2103.11511.

[8] TensorFlow Hub. [Online]. Available: https://tfhub.dev

[9] 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," arXiv preprint arXiv:1705.06950, 2017.

[10] Y. Gunjal, S. Khandare, R. Patil, "Project Report: Human Activity Recognition," Project Report, MIT Academy of Engineering, 2024-25. [Internal Document - Referenced from Uploaded Files]

[11] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 1, pp. 221–231, 2013.

[12] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in Proc. Advances in Neural Information Processing Systems (NIPS), 2014, pp. 568–576.

[13] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2015, pp. 4489–4497.

[14] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in Proc. British Machine Vision Conference (BMVC), 2008.

[15] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in Proc. ACM Int. Conf. Multimedia (MM), 2007, pp. 357–360.

[16] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 2625–2634.

[17] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020.

[18] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?," in Proc. Int. Conf. Machine Learning (ICML), 2021.

[19] Z. Tong, Y. Song, J. Wang, and L. Wang, "VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training," arXiv preprint arXiv:2203.12602, 2022.

[20] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 4510–4520.

[22] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 6848–6856.

[23] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. Int. Conf. Machine Learning (ICML), 2019, pp. 6105–6114.

[24] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 6299–6308. [Note: This paper introduced I3D, often used with efficient backbones]

[25] C. Feichtenhofer, "X3D: Expanding architectures for efficient video recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 203–213.

[26] A. Howard, M. Sandler, G. Chu, L. C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for MobileNetV3," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2019, pp. 1314–1324.

[27] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 7132-7141.

[28] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015. Software available from tensorflow.org. [Accessed: Apr 26, 2025]. Available: https://www.tensorflow.org/about/bib

[29] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2011, pp. 2556–2563.

[30] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980, 2014.