Human Action Recognition - Investigating the Performance of Different Deep Learning Approaches

Shantanu Kumar Rahut (7015438) Rushan Mukherjee (7015520)
Sarvenaz Amiri (7027420) Prakash Naikade (7000433)
Saarland University

Abstract

In this work, we investigate and compare the overall performance of different renowned pre-trained deep learning models and their ensembles on the task of Human Action Recognition. We use a publicly available image dataset found from Kaggle [1]. The dataset is comprised of 15 different classes. We purposely used a small dataset to address the overfitting issues with respect to other proposed models and tried to mitigate it using a deep ensemble technique. Our experiments and code can be found here. [1]

1. Introduction

Human Action Recognition (HAR) [2–4] is one of the leading research fields in computer vision. Although videos can offer more clues to recognize actions, namely texting, dancing, laughing, and listening to music, as shown in figure 1, these actions can be represented by a single still image [3]. In some cases, when the motions in the videos are available, some activities shown in figure 1 may need static cue-based approaches [3]. Hence, video-based techniques stated in [5–7] may not be appropriate to recognize these human motions as there are minute action changes with a lack of distinguishing ability. At the same time, classifying human activities is more difficult in still images since these images are often plagued with disturbances, noise, and cluttered backgrounds [8].

It is important to understand the underlying aspects of the HAR techniques as it can provide a researcher with various types of information like the identity of a person, their personality, psychological state, etc. Therefore, HAR models can be employed in video surveillance (for security purposes), Human-Computer Interaction (HCI), video reclamation, and understanding of visual information.

To model human-object interaction, Yao et al. [9] used pose information and objects. Zhao et al. [10] and Gkioxari et al. [11] proposed models, which combine general features with features of various body parts in order to recognize human actions.

Figure 1. Example of Human Action Images from Kaggle dataset.

Deep learning has emerged as a promising technique for identifying human actions in recent years as a result of Convolutional Neural Network's (CNN) spectacular success in computer vision [12–14] and their potent feature extraction capabilities from raw images. By combining CNNs, Poselets [15] and Gkioxari et al. [11] have identified human activities and characteristics. By simply retraining a pre-trained network's classifier, Oquab et al. [16] adopted the transfer learning technique. For identifying activities in still images, Yan et al. [17] adopted the VGG16 [18] network combined with two additional attention branches.

As a part of our investigations with various deep learning models, we use the deep ensemble learning technique, which is a combination of the results of various models, to further increase the action classification accuracy. We examine the following two aspects:

- Comparing the results of three CNN-based pre-trained models with two different vision transformer-based pre-trained models.
- Investigating the performance of a weighted average ensemble of a CNN-based pre-trained model and a transformer-based pre-trained model.

[1] https://github.com/p0l4r/Human-Action-Recognition-HLCV-Summer-2022

2. Related work

Previously existing research focuses on feature engineering such as body part-based modeling or pose-based representation for human action recognition. The bag-of-features approach was used by Delaitre et al. [2] to study human action recognition in still images. The body position was used by Gupta et al. [19] as a cue to identify actions. Moreover, in order to build a hint-enhanced CNN framework, Qi et al. [20] suggested learning posture hints and deep feature extraction simultaneously. Zhang et al. [21] presented a foreground trajectory extraction method based on a saliency sampling strategy with the aim of reducing the reduction of the valid trajectory of action. To execute human action recognition, Kong et al. [22] suggested extracting depth motion maps pyramid descriptor for each action, followed by the classifier of discriminative collaborative representation. To infer human activities, Ko et al. [23] suggested an action poselet-based method and a two-layer classification model.

Overfitting issue frequently arise when a deep CNN model is trained from scratch using a small dataset. A very effective method to tackle overfitting is the usage of a pre-trained network like VGG16 [18]. Therefore, we use this strategy to increase our classification accuracy and reduce overfitting.

In some cases, linear models like VGG16 [18], which has linear stacks of layers may be inflexible. The amount of network overfitting is heavily influenced by the number of trainable parameters. Furthermore, Vision Transformers(VT) outperform CNNs on big datasets due to their better modeling capacity, smaller inductive biases, and global receptive fields. By lowering receptive fields and utilizing hierarchical pyramidal feature maps, contemporary, enhanced, and smaller vision transformers like Swin Transformers(ST), we effectively get closer to the performance of CNNs. Additionally, ensemble learning [24, 25] can combine various models to produce better results. Different classifiers are pooled, and their predictions are taken into account by weighted coefficients to get better outcomes.

3. Method

3.1. Dataset

We used the dataset sourced from Kaggle [1] which features 15 various classes of human activities, namely calling, clapping, cycling, dancing, drinking, eating, fighting, hugging, laughing, listening-to-music, running, sitting, sleeping, texting, and using-laptop. There are 8442 train and 4158 test images. All the classes are equally distributed in the training set.

3.2. Approach

The method of using a fine-tuned pre-trained model for solving a task of the same genre is known as Transfer Learning [26]. Ensemble methods use multiple learning algorithms to obtain better predictive performance than what could be obtained from any of the constituent learning algorithms alone [27]. We investigated several methods to determine the best procedure for human action recognition from images. The methods include pre-trained CNN-based models such as ResNet50, VGG16, and ResNet152; and transformer-based pre-trained models such as Swin Transformer Base [2] and Tiny model[3]. The main difference between the Base and Tiny models of Swin Transformer is just the difference in trainable parameters. We also tried using a combination or ensemble of the aforementioned models, using a weighted average.

All the CNN models mentioned above are pre-trained on the ImageNet dataset. The Swin Transformer model that we used is also a pre-trained model which was trained on the ImageNet-1K dataset. We did not use the version of Swin Transformer that is pre-trained on the ImageNet-22K dataset because it is too large. Further, apart from finding the best accuracy, we were also focusing on the time it takes to train the model. Every model takes 224x224x3 images as input. We used the same image size for all models. Every model is fine-tuned on the same dataset that we found in Kaggle.

In the best interest of fine-tuning and adding regularization, we used an extra sequential layer on top of the original pre-trained model. These extra sequential layers used in different models are shown in figure 2.

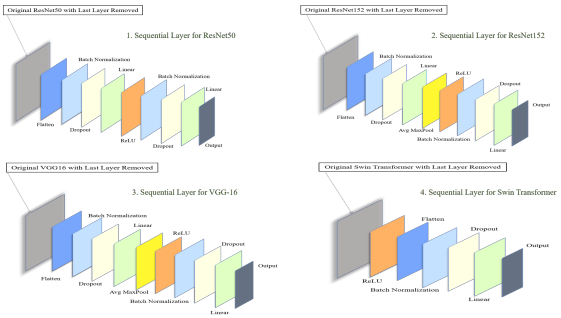

Figure 2. Overview of Sequential Layers added on top of different models.

The dataset is split into training and validation parts with 80-20 ratios respectively. With the help of torch vision transforms, we applied the following transformations to the dataset: Resize and Normalization. We used the same loss function for all models i.e. CrossEntropyLoss as our loss function. For ensemble learning, we used one CNN-based pre-trained model from ResNet50, ResNet152, and VGG16; with Swin Transformer Base or Swin Transformer Tiny architecture. For deciding the output of our ensemble,

[2]Microsoft/swin-base-patch4-window7-224
[3]Microsoft/swin-tiny-patch4-window7-224

we used the weighted average technique shown in figure 3 and the equation below:

$$((\text{Prediction}_1 * Weight_1) + (Prediction_2 * Weight_2))/2 \quad (1)$$

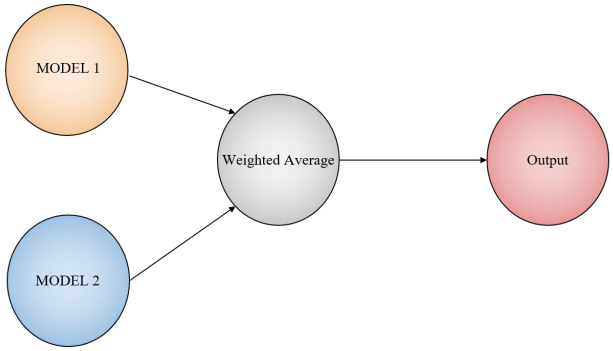

Figure 3. Illustration of Weighted Average of Models involved.

4. Experiment results and analysis

4.1. Hardware Setup

We used hardware support provided by Kaggle. The hardware includes Intel(R) Xeon(R) with four cores and 2.3GHz frequency, 16GB RAM, Nvidia P100 with 16GB of GPU memory, and 9.3 TFLOPS clock performance.

4.2. Experiments and Results

We began our investigations by building two convolutional neural network(CNN) models, Resnet50 and VGG16. Due to our limited GPU availability, we decided on training every model for 20 epochs. Resnet50 model encountered an overfitting problem after 20 epochs. It was the same case with VGG16, as it overfitted on our dataset. The loss and accuracy graphs for the above two models are depicted in figure 4.

On the other hand, both the Swin transformer models: Tiny and Base, underfitted on our dataset. The loss and accuracy graphs for both the Swin Transformer models are shown in figure 5.

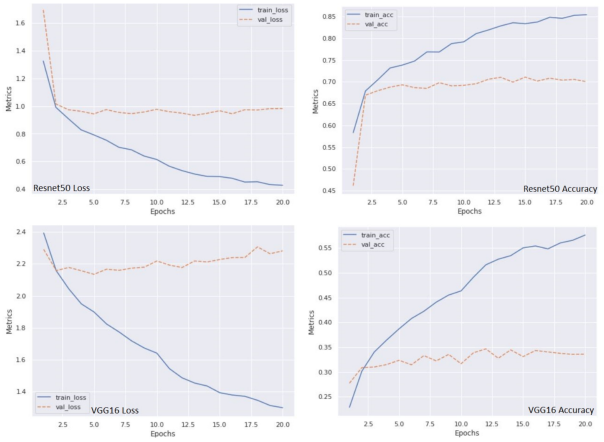

Figure 4. Loss and Accuracy Graph for Resnet50 and VGG16.

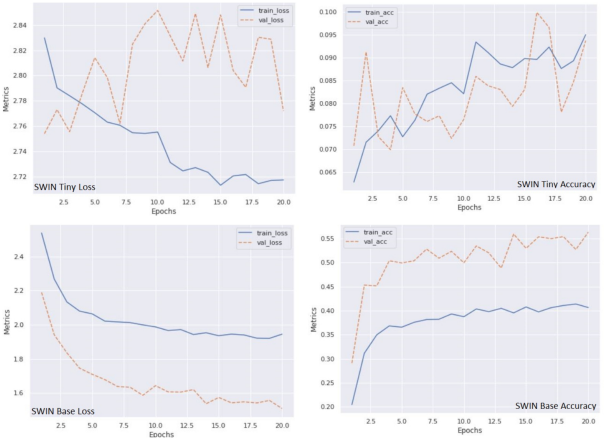

Figure 5. Loss and Accuracy Graph for Swin Transformer Tiny and Base.

Further, we dabble in using an ensemble technique with the Resnet50-Swin Transformer Tiny model and VGG16-Swin Transformer Tiny model respectively. These models show an increase in performance and a decrease in overfitting. Nevertheless, these are not a good fit as we can see from the loss and accuracy graphs in figures 6 and 7.

Our next approach increased the parameters in the Swin Transformer model by replacing the Tiny model with the Base model. As we can see from the figures 6 and 7, these models with both CNNs: Resnet50 and VGG16 perform better than those with the Swin Transformer Tiny model in terms of increased validation accuracy. Still, none of these models were close to a good fit.

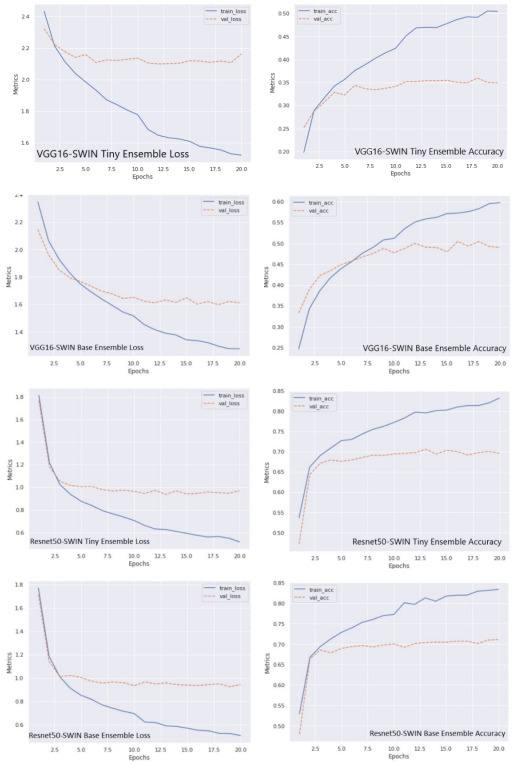

Figure 6. Loss and Accuracy Graphs for Resnet50 and VGG16 Ensemble Models.

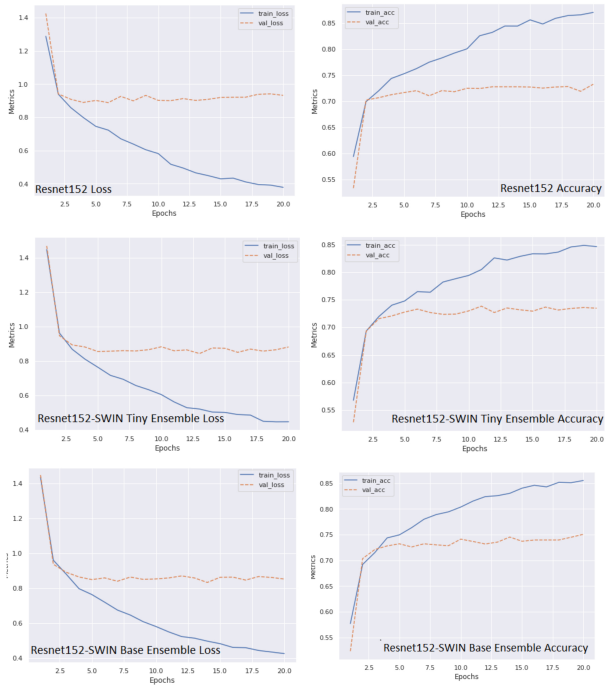

Figure 7. Loss and Accuracy Graphs for Resnet152 based models.

Our final approach with Resnet152 and Swin Transformer ensemble leads us closer to a good fit model. Resnet152-Swin Transformer Tiny model trained in the least amount of time and gave a satisfactory validation accuracy score on our dataset. Whereas, the Resnet152-Swin Transformer Base model took the most amount of time among all the previously tested models but also gave the best training and validation accuracy score. An overview of the performance of different models can be seen in figure 8.

MODEL	Training Loss	Training Accuracy(%)	Validation Loss	Validation Accuracy(%)	Training Time(Mins)
Only VGG16	1.299	57.6	2.282	33.6	25.48
Only Resnet50	0.429	85.4	0.982	70	26.48
Only Resnet152	0.383	87	1.028	70.6	34.5
Only Swin Tiny	2.717	9.5	2.772	9.4	26.83
Only Swin Base	1.945	40.6	1.509	56.3	41.81
VGG16(0.6) with Swin Tiny(0.4)	1.441	52.6	2.15	35.1	35.33
VGG16(0.4) and Swin Base(0.6)	1.274	59.8	1.609	49.1	52.85
Resnet50(0.4) with Swin Base(0.6)	0.506	83.4	0.944	71.1	48.23
Resnet50(0.4) with Swin Tiny(0.6)	0.52	83.2	0.972	69.5	33
Resnet50(0.6) with Swin Base(0.4)	0.477	84.6	0.903	72.1	47.58
Resnet152(0.6) with Swin Tiny(0.4)	0.448	84.7	0.883	73.5	41.4
Renset152(0.6) Swin BASE(0.4)	0.425	85.5	0.852	75.1	56
Resnet152(0.4) Swin Base(0.6)	0.445	84.7	0.922	72.1	56.96

Figure 8. Results of the performance of used models in Tabular Format.

4.3. Analysis

Every CNN-based pre-trained model that we used individually encountered overfitting in the training phase. Possible reasons for this is most likely that the dataset is not large enough and the model is too complex for the data. On the other hand, Swin Transformer is a relatively new Vision Transformer architecture that has shown promising results in the ImageNet classification task. However, for the task of Human Action Recognition with our dataset, we observed that Swin Transformer is underfitting in the training phase. We tried adding more sequential layers on top of it to remove underfitting issue, however this approach made the model overfit because of the dataset size. Ensemble of CNN-based pre-trained model that is too complex (overfitting) for the dataset, and Swin Transformer that is too simple (underfitting) for the dataset, reduce the overall error and gives us a model closer to good fit.

5. Conclusion

In our experiments, we construct two ensemble models that show us a close to good fit model. They are known as Resnet152-Swin Tiny Ensemble and Resnet152-Swin Base Ensemble models. The Tiny Ensemble model trains in the least amount of time and gives good training and validation accuracy scores. On the contrary, the Resnet152-Swin Base Ensemble model takes a great deal of extra time to train and displays the best training and validation accuracy. Consequently, we conclude that Ensemble learning is a viable option for Human Action Recognition, specifically when the dataset provided is just static images and the dataset size is comparatively small.

References

[1] "Human Action Recognition (HAR) Dataset", *Available: https://www.kaggle.com/datasets/meetnagadia/human-action-recognition-har-dataset*, 2022. 1, 2

[2] V. Delaitre, I. Laptev, and J. Sivic, "Recognizing human actions in still images: a study of bag-of-features and part-based representations," in *BMVC 2010-21st British Machine Vision Conference*, 2010. 1, 2

[3] G. Yao, T. Lei, and J. Zhong, "A review of convolutional-neural-network-based action recognition," in *Pattern Recognition Letters*, vol. 118, pp. 14–22, 2019. 1

[4] W. Xu, Z. Miao, J. Yu, and Q. Ji, "Action recognition and localization with spatial and temporal contexts," in *Neurocomputing*, vol. 333, pp. 351–363, 2019. 1

[5] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1933–1941, Las Vegas, NV, USA, 2016. 1

[6] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi, "Action recognition with dynamic image networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2799–2813, 2018. 1

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

[8] Xiangchun Yu, Zhe Zhang, Lei Wu, Wei Pang, Hechang Chen, Zhezhou Yu, Bin Li, "Deep Ensemble Learning for Human Action Recognition in Still Images," Complexity *https://doi.org/10.1155/2020/9428612*, vol. 2020, Article ID 9428612, 23 pages, 2020. 1

[9] B. Yao and L. Fei-Fei, "Modeling mutual context of object and human pose in human-object interaction activities," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 17–24, 2010. 1

[10] Z. Zhao, H. Ma, and S. You, "Single image action recognition using semantic body part actions," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3391–3399, 2017. 1

[11] G. Gkioxari, R. Girshick, and J. Malik, "Actions and attributes from wholes and parts," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2470–2478, 2015. 1

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012. 1

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014. 1

[14] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015. 1

[15] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3D human pose annotations," in *IEEE 12th International Conference on Computer Vision*, pp. 1365-1372, doi: 10.1109/ICCV.2009.5459303, 2009. 1

[16] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1717–1724, 2014. 1

[17] S. Yan, J. S. Smith, W. Lu, and B. Zhang, "Multibranch attention networks for action recognition in still images," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 4, pp. 1116–1125, 2017. 1

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *https://arxiv.org/abs/1409.1556*, 2014. 1, 2

[19] A. Gupta, A. Kembhavi, and L. S. Davis, "Observing human-object interactions: using spatial and functional compatibility for recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1775–1789, 2009. 2

[20] T. Qi, Y. Xu, Y. Quan, Y. Wang, and H. Ling, "Image-based action recognition using hint-enhanced deep neural networks," in *Neurocomputing*, vol. 267, pp. 475–488, 2017. 2

[21] G. Zhang, S. Jia, X. Zhang, and X. Li, "Saliency-based foreground trajectory extraction using multiscale hybrid masks for action recognition," *Journal of Electronic Imaging*, vol. 27, no. 5, Article ID 053049, 2018. 2

[22] J. Kong, B. Zan, and M. Jiang, "Human action recognition using depth motion maps pyramid and discriminative collaborative representation classifier," *Journal of Electronic Imaging*, vol. 27, no. 3, Article ID 033027, 2018. 2

[23] B. Ko, J. Hong, and J.-Y. Nam, "Human action recognition in still images using action poselets and a two-layer classification model," *Journal of Visual Languages and Computing*, vol. 28, pp. 163–175, 2015. 2

[24] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: a survey," *Information Fusion*, vol. 37, pp. 132–156, 2017. 2

[25] X.-L. Zhang and D. Wang, "A deep ensemble learning method for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 967–977, 2016. 2

[26] "What is Transfer Learning? Exploring the Popular Deep Learning Approach," *https://builtin.com/data-science/transfer-learning*, 2022. 2

[27] "What is Ensemble learning?, Wikipedia," *https://en.wikipedia.org/wiki/Ensemble_learning*, 2022. 2