**Human\_Action\_Recogntion\_using\_CNN\_+\_LSTM**

Anoushka Tripathi

# Abstract

This project explores two approaches for action recognition in videos: ConvLSTM and LRCN. The ConvLSTM approach combines Convolutional and LSTM cells to capture spatial and temporal features efficiently. Meanwhile, the LRCN approach integrates Conv2D layers with LSTM units for combined spatial and temporal modeling. Both methods demonstrate effectiveness in recognizing actions, offering insights into their applicability in real-world scenarios.

Top of Form

# Introduction

Action recognition in videos plays a pivotal role across a multitude of domains, ranging from surveillance and human-computer interaction to sports analysis. As such, the ability to automatically identify and classify actions depicted in videos has garnered significant attention in the field of computer vision. In this project, we delve into two distinct approaches for action recognition: Convolutional Long Short-Term Memory (ConvLSTM) and Long-term Recurrent Convolutional Networks (LRCN). The overarching goal is to investigate the effectiveness of these methods in capturing spatiotemporal features and accurately categorizing actions within video sequences. By implementing and evaluating these approaches, we aim to contribute to the advancement of action recognition techniques and facilitate their application in real-world scenarios.

# Related Work

Several research papers have delved into the realm of action recognition, aiming to enhance model performance and explore innovative architectures. One notable paper in this domain is "Long-term Recurrent Convolutional Networks for Visual Recognition and Description" by Jeff Donahue et al. (CVPR 2015). This paper introduces the Long-term Recurrent Convolutional Network (LRCN) architecture, which integrates convolutional and recurrent layers to capture spatial and temporal information in videos. LRCN demonstrates superior performance in various action recognition tasks, providing inspiration for subsequent studies in the field. Additionally, works such as "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting" by Xingjian Shi et al. (NIPS 2015) have explored the application of Convolutional LSTM networks, a variant of LSTM with convolutional operations, for spatiotemporal data modeling. These architectures have laid the foundation for advancements in action recognition, guiding the development of novel models with improved accuracy and efficiency.

# Dataset and Features

The dataset utilized in this project is the UCF50 Action Recognition Dataset, renowned for its collection of realistic videos sourced from YouTube. It comprises 50 distinct action categories, with each category containing approximately 25 groups of videos. On average, each action category consists of 133 videos, with a mean duration of 199 frames per video. The frame dimensions typically measure 320 pixels in width and 240 pixels in height. Moreover, the videos exhibit an average frame rate of 26 frames per second. This dataset's unique characteristic lies in its realistic nature, as the videos are not staged by actors but are authentic recordings, making it well-suited for training deep learning models for action recognition tasks.

1. Top of Form



Figure 1: Datasets.

# Methods

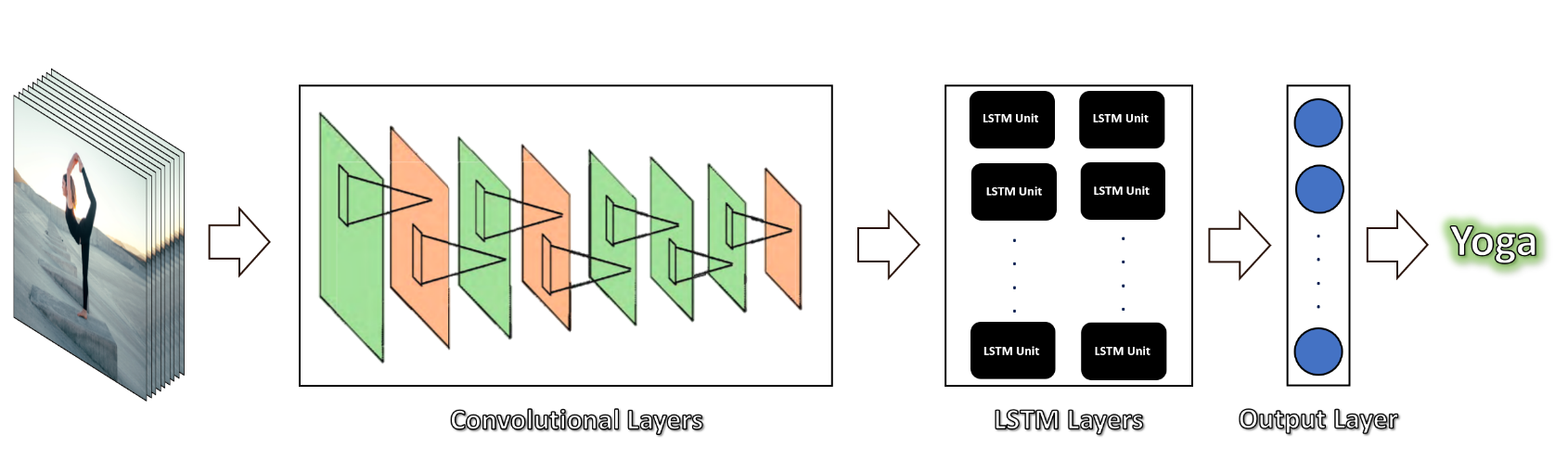
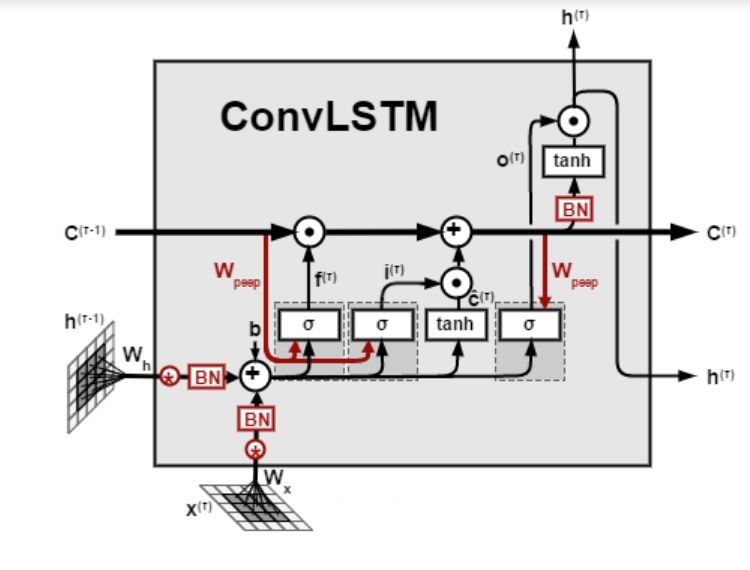


Figure 2: LCRN Approach



## Architecture

**ConvLSTM Approach:** In this approach, ConvLSTM cells are utilized, which are a variant of LSTM networks containing convolutional operations. This architecture effectively captures both spatial and temporal relations in videos. The model is constructed using ConvLSTM2D layers, which have filters, kernel sizes, and recurrent dropout parameters specified. MaxPooling3D layers are incorporated to reduce dimensionality, while Dropout layers prevent overfitting. Finally, a Dense layer with softmax activation outputs the probability of each action category.

**LRCN Approach:** The Long-term Recurrent Convolutional Network (LRCN) architecture combines CNN and LSTM layers in a single model. Convolutional layers extract spatial features from video frames, followed by MaxPooling2D and Dropout layers. The feature vectors are then flattened and fed into an LSTM layer for temporal sequence modeling. A Dense layer with softmax activation predicts the action category. TimeDistributed wrapper layers are used to apply layers to each frame independently, enabling end-to-end training for robust action recognition.

## Metrics

The metrics used for evaluating the performance of the action recognition models include:

1. **Loss:** This metric measures the error between the predicted probabilities and the actual labels. It is typically calculated using categorical cross-entropy, which penalizes the model more for confidently incorrect predictions.
2. **Accuracy:** This metric indicates the proportion of correctly classified samples out of the total number of samples. It provides an overall measure of how well the model is performing across all classes.

These metrics are computed during both training and evaluation phases. During training, the model aims to minimize the loss by adjusting its parameters using optimization algorithms like Adam. Early stopping is employed based on validation loss to prevent overfitting and ensure that the model generalizes well to unseen data.

.

*I*

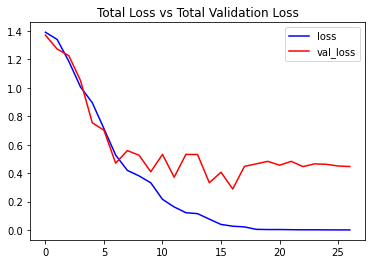
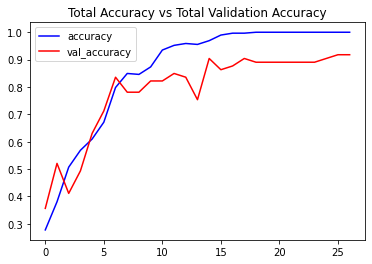
**LRCN Approach:**

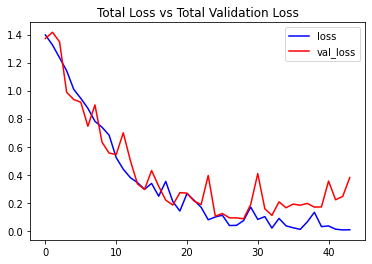
* Loss: 0.2242
* Accuracy: 92.62%

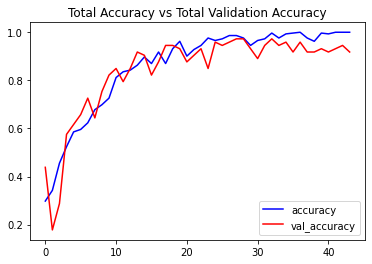
**ConvLSTM Approach:**

* Loss: 0.8976
* Accuracy: 80.33%

These results indicate that the LRCN approach outperformed the ConvLSTM approach in terms of both loss and accuracy. The LRCN model achieved a lower loss value and a higher accuracy rate compared to the ConvLSTM model.

  
Figure 4:ConvLSTM Approach





# Conclusion

In our experiments, we have demonstrated that SRCNN surpasses non-neural methods for the task of super-resolution, achieving a test PSNR of 26.442 dB, surpassing the baseline bicubic PSNR of 23.226 dB. In the output images, Figure [8,](#_bookmark6) we see that the model increases

Figure 5: Comparison of validation loss and PSNR with tanh and no tanh activation functions

Figure 6: Comparison of validation loss and PSNR with batchnorm and no batchnorm. The experiments with batchnorm have much higher jitter and perform worse than the corresponding versions without.

sharpness and adds naturalistic detail. There are some remaining issues with pixels that are very different in color than they should be, possibly due due overflow or clipping. We also see some remaining pixelation, for instance, in the corn leaves at the upper right corner of the first photo in Figure [8.](#_bookmark6)

Our work demonstrates the promise of deep neural architectures for achieving state-of-the-art performance on the task of image super- resolution. We also expect that having the computing resources and time to train on the full DIV2K dataset of 800 images would improve the model’s performance. The lack of availability of GPUs also limited us to a relatively shallow neural network.

While the SRCNN architecture offers significant improvements over the baseline, other models such as Fast SRCNN (FSRCNN) as well as adversarial networks improve over SRCNN’s performance. In future work, we would like to approach the super-resolution task using a UNet++ architecture as well as a generative adversarial network.

# Contributions

We performed the coding, analysis, and writing as a team.

We would like to thank our mentor Guanzhi Wang for his advice.

|  |  |  |  |
| --- | --- | --- | --- |
| Experiment | Max Val PSNR (dB) | Test Loss | Test PSNR (dB) |
| Basic SRCNN | 27.455 | **0.0064324** | **26.442** |
| + Regularization | **27.459** | 0.0064412 | 26.429 |
| + Tanh | 27.336 | 0.0066215 | 26.282 |
| + Batch Norm | 25.530 | 0.0100244 | 24.373 |
| Bicubic | 25.433 | 0.0066051 | 23.226 |
| Bilinear | 25.220 | 0.0074545 | 23.182 |
| Nearest neighbor | 22.198 | 0.014747 | 20.551 |

Table 1: Comparison of model performance, where learning rate is 0.001.

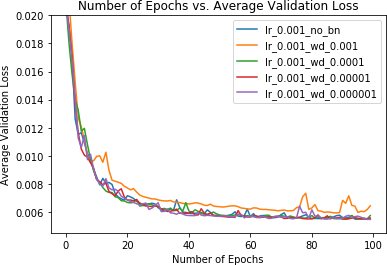
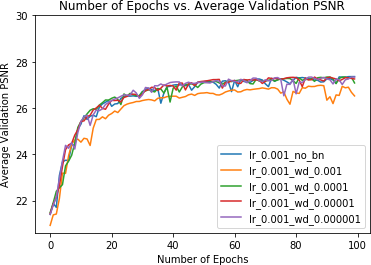
 

Figure 7: Comparison of validation loss and PSNR with different regularization scaling factors. These are comparable to the performance of the non-regularized model.



Figure 8: Output of model with 0.001 learning rate. (left: input images. center: predicted images. right: target images.) The predictions produce significantly sharper edges and appear to have higher dynamic range than the inputs.

# References

1. Eirikur Agustsson and Radu Timofte. “NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. July 2017.
2. Chao Dong et al. “Image Super-Resolution Using Deep Convolutional Networks”. In: *CoRR* abs/1501.00092 (2015). arXiv:

[1501.00092](https://arxiv.org/abs/1501.00092). URL: <http://arxiv.org/abs/1501.00092>.

1. R. Keys. “Cubic convolution interpolation for digital image processing”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 29.6 (1981), pp. 1153–1160.
2. Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. “Accurate image super-resolution using very deep convolutional networks”. In:

*Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1646–1654.

1. Bee Lim et al. “Enhanced deep residual networks for single image super-resolution”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 136–144.
2. Wenzhe Shi et al. “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network”. In:

*Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1874–1883.

1. Wenming Yang et al. “Deep learning for single image super-resolution: A brief review”. In: *IEEE Transactions on Multimedia* 21.12 (2019), pp. 3106–3121.