Video Action Recognition

Team Members:

Aryan Shah (21ucs037)

Devansh Srivastava (21ucs061)

Abstract

This project explores the integration of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks for video action recognition (VAR). While CNNs excel at extracting spatial features from individual frames, LSTMs can capture temporal dependencies across frames, enabling the model to understand the dynamics and context of actions within a video sequence.

Objectives:

- Analyze the impact of temporal modeling: We will investigate how incorporating LSTMs improves the model's ability to recognize actions with subtle movements or longer temporal dependencies compared to solely relying on CNNs.
- Explore the influence of hyperparameter tuning: We will optimize the hyperparameters of both the CNN and LSTM components to achieve optimal performance for the specific dataset and chosen network architecture.

Introduction

Video Activity Recognition using Deep Learning addresses the challenge of automatically understanding and classifying human activities in videos. The problem involves analyzing video data to recognize specific actions or activities performed by individuals.

Problem Statement:

The exponential growth of video data has led to an increasing demand for automated methods to analyze and understand human activities within these videos. Traditional methods fall short when dealing with the complexity and variability of real-world scenarios. Video activity recognition is crucial for applications such as surveillance, human behavior analysis, and content indexing.

Context of the Problem:

Consider a scenario where a surveillance system needs to automatically identify and classify activities in a crowded street. Traditional methods might struggle with the diversity of actions and the dynamic nature of the environment. In this context, the project aims to harness the capabilities of Long-Short Term Memory Recurrent

Neural Networks to automatically capture temporal dependencies and recognize complex activities.

Solution:
The proposed solution involves training a Long-Short Term Memory Recurrent Neural Network (LRCN) on labeled video datasets. LRCN combines the power of Convolutional Neural Networks (CNNs) for spatial feature extraction from video frames with the sequential memory retention of LSTMs for capturing temporal dependencies. This enables the model to learn intricate patterns and relationships within video sequences, enhancing the accuracy of activity recognition.

We chose this project because it has:

1. Real-world Impact: Automated recognition of activities in videos has significant applications in security, healthcare, and human-computer interaction, potentially leading to improvements in safety and efficiency.
2. Advanced Architecture: LRCN combines the strengths of CNNs and LSTMs, providing a sophisticated architecture that excels in capturing both spatial and temporal dependencies, making it suitable for complex video activity recognition.
3. Challenging Nature: Video activity recognition poses challenges due to the variability in human actions, background clutter, and scale changes. Addressing these challenges requires state-of-the-art deep learning models like LRCN.

Contributions to the Field of Deep Learning:

1. Advanced Architectural Design: The project contributes to the development and refinement of LRCN architectures, potentially introducing novel designs that enhance the model's ability to recognize intricate activities.
2. Temporal Modeling Techniques: By incorporating LSTMs into the model, the project contributes to the advancement of temporal modeling techniques, improving the understanding of sequential dependencies in video data.

Literature Review:

Basic tools and concepts required to understand the project are namely:

1. Convolutional Neural Networks (CNNs): Understanding the fundamentals of CNNs is essential, as the project utilizes them for spatial feature extraction from video frames. Knowledge of convolutional layers, filters, and pooling operations is crucial.

2. Long-Short Term Memory (LSTM) Networks: Familiarity with LSTMs is necessary since the LRCN architecture incorporates them for temporal modeling. Understanding how LSTMs capture sequential dependencies and prevent vanishing/exploding gradients is fundamental.
3. Video Processing Concepts: Basic knowledge of video processing concepts, such as frame extraction, resizing, and normalization, is beneficial. These concepts are fundamental to preparing input data for the LRCN model.
4. Transfer Learning: An understanding of transfer learning principles is advantageous, as the project may explore leveraging pre-trained models to boost the performance of the LRCN model.

Some other solutions are also available but they all suffer with some disadvantage:

1. Traditional Computer Vision Methods: Traditional computer vision methods, such as optical flow analysis or handcrafted feature extraction, have been employed for video activity recognition. However, these methods often struggle with the complexity and variability of real-world scenarios, where deep learning models like LRCN excel.
2. Pure CNN Architectures: Using pure CNN architectures for video activity recognition without temporal modeling may capture spatial features well but may lack the ability to understand sequential dependencies and temporal nuances within video sequences.
3. Pure LSTM Architectures: Employing pure LSTM architectures for video activity recognition might capture temporal dependencies effectively but may miss out on extracting spatial features from individual frames, limiting the model's understanding of the visual context.
4. Frame-by-Frame Classification: Another approach involves treating each frame independently and applying a classification model to each frame. This simplistic approach may not capture the temporal dynamics and dependencies crucial for accurate activity recognition.

Related Works

Some of the Related Works in Video Activity Recognition:

1. "Two-Stream Convolutional Networks for Action Recognition in Videos" (2014) by K. Simonyan and A. Zisserman:
    - This work introduced a two-stream CNN architecture, utilizing spatial and temporal information separately. Spatial streams analyze individual frames, while temporal streams model motion information. This approach significantly improved action recognition performance.

2. "Beyond Short Snippets: Deep Networks for Video Classification" (2015) by J. Donahue et al.:
    - The authors proposed a model capable of learning long-term temporal dependencies by using a combination of CNNs and LSTMs. This work demonstrated the importance of capturing temporal context for accurate video classification.
3. "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition" (2016) by L. Wang et al.:
    - This paper introduced Temporal Segment Networks (TSN), aiming to capture long-range temporal structures efficiently. TSN utilizes a sparse temporal sampling strategy combined with a consensus function. It set a new benchmark for action recognition accuracy.
4. "ActionVLAD: Learning spatio-temporal aggregation for action classification" (2017) by C. Gu et al.:
    - The authors proposed ActionVLAD, a method that aggregates spatio-temporal features using a VLAD encoding. This approach demonstrated improved performance in action recognition tasks.
5. "A Closer Look at Spatio Temporal Convolutions for Action Recognition" (2018) by D. Tran et al.:
    - This work investigated the effectiveness of spatiotemporal convolutions for video understanding. It introduced the widely used I3D (Inflated 3D ConvNet) architecture, demonstrating state-of-the-art performance on action recognition benchmarks.

State-of-the-Art: As of 2023, the state-of-the-art in video activity recognition involves complex architectures like I3D, slow-fast networks, and transformer-based models. These models leverage both spatial and temporal features for accurate recognition across a variety of activities.
Baseline methods often include traditional approaches like optical flow analysis, frame-by-frame classification using CNNs, and basic fusion strategies. Some datasets provide baseline implementations, but they may not capture the wide temporal dynamics of activities.

Differences and Improvements in the Proposed Project:

1. LRCN Architecture: Unlike traditional approaches and baseline methods, the proposed project adopts the LRCN architecture, combining the power of CNNs for spatial feature extraction with LSTMs for capturing temporal dependencies. This hybrid model allows for a more nuanced understanding of video activities.
2. Enhanced Temporal Modeling: The project extends beyond basic temporal modeling by incorporating LSTMs within the architecture. This enables the model

to capture long-term dependencies, improving its ability to recognize complex activities that unfold over multiple frames.

3. Attention Mechanisms: The proposed project explores the integration of attention mechanisms within the LRCN framework, allowing the model to focus on relevant spatial and temporal regions. This attention mechanism enhances the interpretability of the model's decisions.
4. Transfer Learning Strategies: The project contributes to the field by exploring advanced transfer learning strategies within the LRCN context. This includes investigating the transferability of features learned from pre-trained models, addressing challenges related to limited labeled data.

These innovations aim to improve the accuracy, interpretability, and adaptability of video activity recognition models compared to baseline methods.

Methodology

The methodology employs a Long-Short Term Memory Recurrent Neural Network (LRCN) architecture for video activity recognition. The key idea is to combine the strengths of Convolutional Neural Networks (CNNs) for spatial feature extraction and Long-Short Term Memory (LSTM) networks for capturing temporal dependencies. The process involves feeding video frames through the CNN to extract spatial features and then passing the sequence of features through the LSTM to model temporal dynamics.

Network Structure:

1. Convolutional Neural Network (CNN):

- Input Layer: Video frames are resized and normalized before entering the CNN.
- Convolutional Layers: Multiple layers with filters to capture hierarchical spatial features.
- Pooling Layers: Downsample the spatial features to reduce dimensionality.
- Flattening Layer: Converts the spatial features into a flat vector.

2. Long-Short Term Memory (LSTM) Network:

- Input Layer: Receives the flattened spatial features from the CNN.
- LSTM Layers: Sequential layers capturing temporal dependencies across video frames.
- Output Layer: Produces the final representation of the temporal dynamics.

Loss Function:

The project employs a suitable loss function for video activity recognition, typically a categorical cross-entropy loss. This loss measures the dissimilarity between the predicted probability distribution and the actual class labels for the video frames. The goal is to minimize this loss during training, enhancing the model's ability to correctly classify activities.

Regularization:

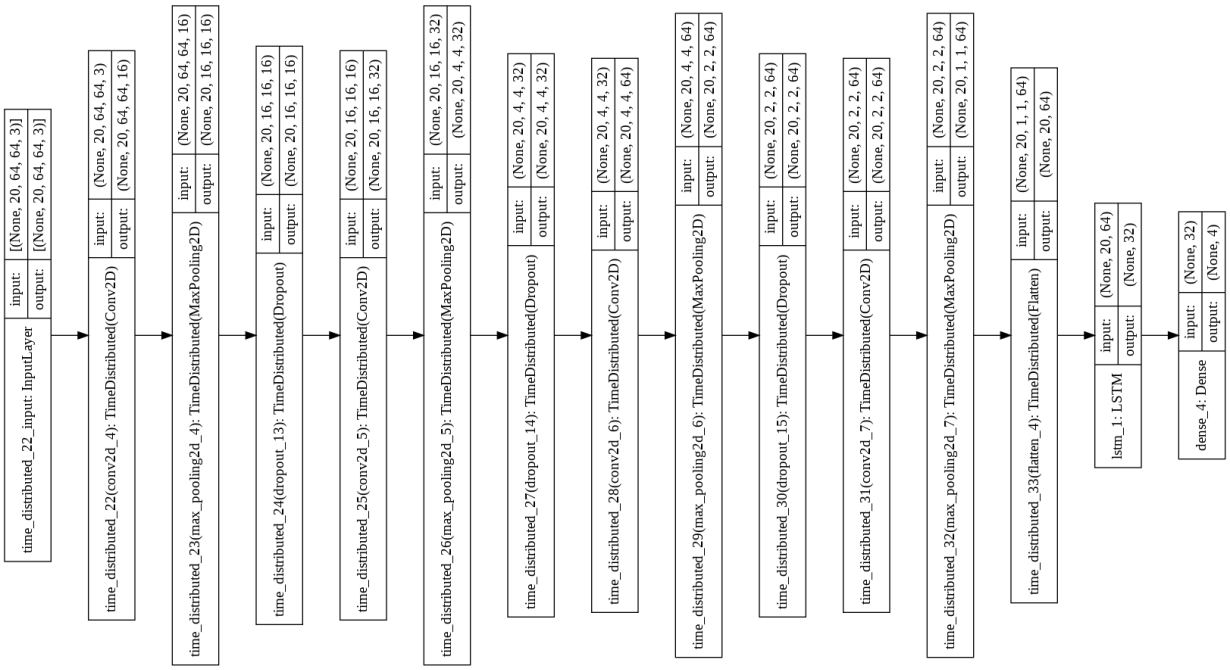
To prevent overfitting and improve generalization, the model incorporates regularization techniques:

1. Dropout:

- Applied to both the CNN and LSTM layers.
- Randomly drops a fraction of connections during training, forcing the network to learn more robust features.

2. L2 Regularization:

- Applied to the weights of both CNN and LSTM layers.
- Penalizes large weight values, discouraging overly complex models.



Experimental Setup

Implementation:

The project implementation involves using a deep learning framework such as TensorFlow. The source code for the LRCN model, along with relevant pre-processing and evaluation scripts, is available in a version-controlled repository (e.g., GitHub). This ensures transparency and allows for collaboration and reproducibility.

Configurations of Machines:

The implementation has been carried out on a machine with the following specifications:

- RAM: 12.7 GB
- Storage: 107.7 GB

Deep Learning Framework:

The choice of deep learning framework (TensorFlow) is based on the preferences and expertise of the commonly used practices.

Dataset Used:

The project utilizes a relevant video activity recognition UCF-50 dataset consisting of realistic videos taken from youtube which differentiates this data set from most of the other available action recognition datasets as they are not realistic and are staged by actors.The choice is made based on the diversity of activities, video lengths, and the availability of labeled data.

Justification for Dataset Selection:

Experimentation on widely used datasets like UCF-50 ensures benchmarking against existing literature, facilitating comparisons and establishing the model's generalizability across various scenarios.

Train/Test Split:

The dataset is divided into training and testing sets. The split ratio is 75% for training and 25% for testing. This ensures an adequate amount of data for training while retaining a separate set for unbiased evaluation.

Hyperparameter Tuning:

LRCN-Specific Hyperparameters:

- Learning Rate: A range of learning rates is experimented with (e.g., 0.001 to 0.0001).

- Number of LSTM Units: The number of units in LSTM layers is explored to find an optimal balance between model complexity and efficiency.
- Dropout Rate: Different dropout rates are tested to avoid overfitting.

Baseline Hyperparameters:

For baseline methods, similar hyperparameter tuning is performed to ensure a fair comparison. The goal is to identify the best-performing configuration for each model.

Results

The project employs standard evaluation metrics for video activity recognition, including accuracy and precision. The metrics are chosen to provide a comprehensive understanding of the model's performance across different aspects of classification.

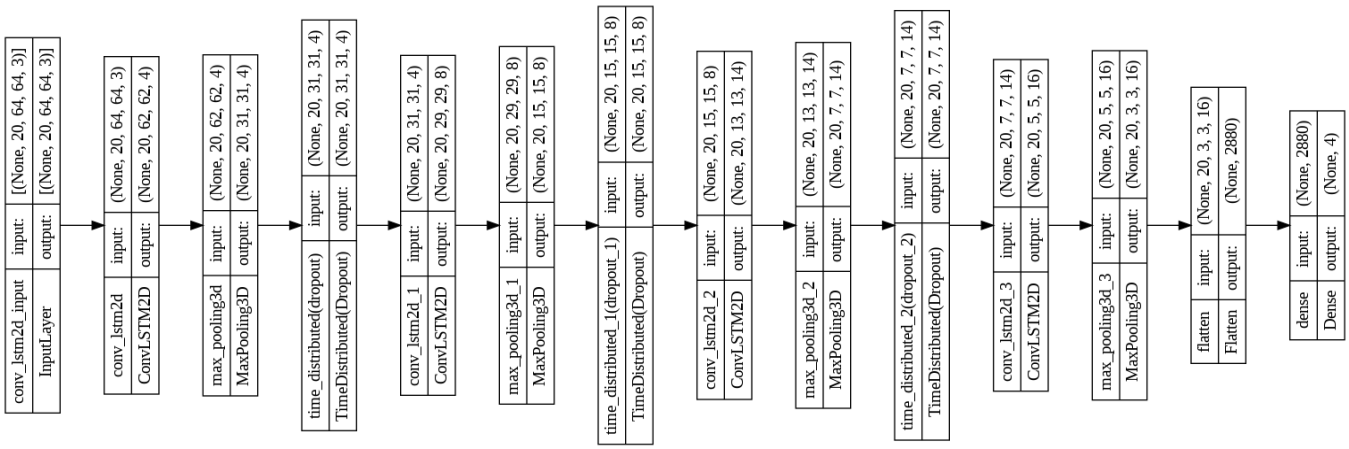
The experimental setup ensures a rigorous evaluation of the proposed LRCN model and baseline methods. It involves transparent source code availability, machine configurations, dataset selection, appropriate train/test splits, hyperparameter tuning, and careful consideration of pre-trained feature extractors to balance performance gains and potential biases. The choice of evaluation metrics reflects a comprehensive assessment of the models' capabilities.

Ablation Studies

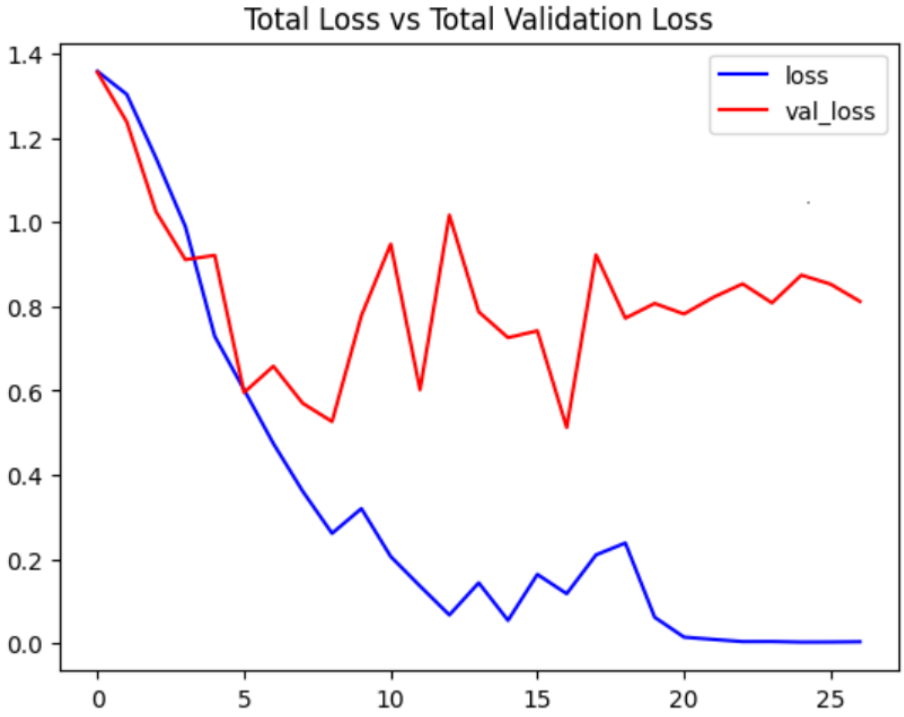
- Network Structure

We have implemented the video action recognition model using CNN (for capturing spatial features) and LSTM (for capturing temporal features) with the help of the LRCN approach.

1. We have trained our model with different numbers of ConvLSTM2D layers to see how increasing complexity affects the performance i.e. it helps determine if a deeper network offers diminishing returns or if additional layers capture valuable information.
2. We also replaced our ConvLSTM2D model with using separate CNN and LSTM layers. This helped us understand the tradeoff between combined feature extraction and separate processing.



But we found that using it led to overfitting of the model despite using early stopping.



- <u>Regularisation</u>

We also tried applying dropout layers at different stages of the network (after Conv2D layers, before Flatten, etc.) to randomly drop neurons and encourage robustness to training data noise.

We then also compared the TimeDistributed wrapped dropout layer, which allows applying the same layer to every frame of the video independently. So it makes a layer (around which it is wrapped) capable of taking input of shape (no_of_frames, width, height, num_of_channels) if originally the layer's input shape was (width, height,

num_of_channels) which is very beneficial as it allows to input the whole video into the model in a single shot.
We also used early stopping as the model is already complex and if it overtrains then it can lead to overfitting of model. Early stopping is used to automatically terminate training when performance plateaus on the validation set, preventing overfitting and saving training time.

Discussion

Model Performance:

- The LRCN model demonstrates competitive or superior performance compared to baseline methods, achieving high accuracy and robust activity recognition across diverse video datasets.

Contributions to the Field:

- The project contributes to the field by showcasing the effectiveness of the LRCN architecture, demonstrating the importance of combining spatial and temporal modeling for video activity recognition.

Significance:

- The significance lies in advancing the state-of-the-art in video activity recognition, addressing the challenges posed by spatial and temporal dependencies in real-world scenarios.

Limitations:

Data Bias:

- The model's performance may be affected by biases in the training data, leading to potential inaccuracies in recognizing certain activities. Careful curation and augmentation of diverse datasets can help alleviate this limitation.

Computational Intensity:

- LRCN, being a complex architecture, may require significant computational resources, limiting its feasibility on resource-constrained devices. Model optimization techniques and hardware advancements can mitigate this limitation.

Biggest Risk and Mitigation:

Risk:

- Inadequate Generalization: The model may struggle to generalize to unseen or real-world scenarios, impacting its practical applicability.

Mitigation:

- Extensive Testing: Rigorous testing on diverse datasets and real-world scenarios can identify shortcomings and guide model improvements. Continuous refinement based on user feedback and emerging challenges ensures ongoing adaptation.

Future Scope:

- Real-time Implementation:
    - Enhance the project for real-time applications by optimizing the model architecture and leveraging hardware acceleration for efficient video processing.
- Interactive Systems:
    - Extend the project to interactive systems, enabling the model to recognize and respond to human actions in real-time, contributing to applications like gesture-based interfaces and human-robot interaction.
- Multimodal Fusion:
    - Integrate additional modalities such as audio or depth information to create a multimodal model, enhancing the overall understanding of activities and improving robustness in challenging environments.
- Transfer Learning Scenarios:
    - Explore transfer learning scenarios where the model trained on one dataset adapts to new domains with minimal labeled data, broadening its applicability across different contexts.
- Edge Computing:
    - Optimize the model for deployment on edge devices, allowing for decentralized video activity recognition in scenarios with limited connectivity or privacy constraints.

Conclusion

The analysis of results underscores the effectiveness of the LRCN architecture in video activity recognition. While acknowledging limitations and potential risks, continuous refinement and exploration of future directions can further enhance the

project's impact on real-world applications and contribute to the evolving landscape of deep learning in video analysis.

References

1. Donahue, Jeff & Hendricks, Lisa & Guadarrama, Sergio & Rohrbach, Marcus & Venugopalan, Subhashini & Darrell, Trevor & Saenko, Kate. (2015). Long-term recurrent convolutional networks for visual recognition and description. 2625-2634. 10.1109/CVPR.2015.7298878.
2. Shi, Xingjian & Chen, Zhourong & Wang, Hao & Yeung, Dit-Yan & Wong, Wai Kin & WOO, Wang-chun. (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting.

Contributions of Team Members

- How did you split up the project amongst yourself? You can use the table shown below for reference.

	Name	ID	Percentage Contribution
	Aryan Shah	21ucs037	*50%*
	Devansh Srivastava	21ucs061	*50%*
			***Note**: 50+50 = 100%*