Human Activity Recognition in Video Using Deep Neural-Networks (DNN)

Harsh Nitinkumar Shah
Student Id:1211101
School of Engineering and Physical Sciences,
University of Guelph

ABSTRACT: Human activity recognition (HAR) using deep neural networks (DNNs) on image data has gained considerable attention as a promising research area with diverse applications. This project report provides an in-depth overview of the latest advancements in HAR utilizing popular DNN architectures, namely Long-Range Convolutional Neural Networks (LRCNN) and Convolutional Neural Networks with Long Short-Term Memory (CNN-LSTM), in addition to state-of-the-art architectures. Notably, the report delves into the utilization of optimization techniques, such as Frame Selection for Action Recognition, to improve computational efficiency. The technical details, strengths, and limitations of these architectures for HAR with image data are thoroughly discussed, including recent developments and challenges. Moreover, crucial aspects of image data pre-processing, training strategies, and potential applications of HAR with DNNs are highlighted. This comprehensive report serves as resource for researchers and practitioners interested in implementing cutting-edge techniques in HAR with image data.

***Index Terms*—Human Activity Recognition, Deep Learning, Optical flow**

I. Introduction

HUMAN activity recognition (HAR) using computer vision is a dynamic field of study that combines the power of visual data analysis with the ability to understand and recognize human behavior. With the proliferation of cameras and imaging devices, HAR has gained significant attention in various industries, particularly in healthcare. By analyzing visual data, HAR has the potential to assist physicians in making informed decisions, optimize medical resource allocation, and provide real-time monitoring of patients' activities. This can greatly improve patient care and outcomes in healthcare systems.

One of the key applications of HAR is in healthcare, where it can be used to monitor patients' activities and detect anomalous events such as falls or seizures. For example, in assisted living facilities or hospitals, HAR can help in automatically tracking the movements and activities of patients, providing valuable insights into their well-being and safety. This can enable timely interventions in case of emergencies or abnormal behaviors, leading to improved patient care and reduced healthcare costs. Additionally, HAR can aid in the rehabilitation process by monitoring and analyzing the movements and exercises performed by patients, helping them in their recovery journey.

HAR is not limited to healthcare, but also finds applications in personal fitness tracking and elderly care. With the increasing focus on physical fitness and wellness, individuals can use HAR to monitor their physical activities, track their exercise routines, and analyze their performance over time. This can help individuals set and achieve fitness goals, optimize their workouts, and make informed decisions about their health and well-being. Moreover, in elderly care, HAR can play a crucial role in ensuring the safety and well-being of older adults by detecting falls or other abnormal events and providing timely assistance. This can enable older adults to live independently for longer and improve their quality of life.

The primary goal of human action recognition is to analyze videos and accurately identify the actions being performed in them. Videos contain both spatial and temporal aspects, where individual frames represent the spatial information, and the ordering of frames represents the temporal information. While some actions can be accurately identified using only a single frame, such as standing or running, more complex actions, like walking versus running or bending versus falling, may require information from multiple frames to be correctly classified.

In human action recognition tasks, local temporal information plays a crucial role in differentiating between various actions. The ordering of frames and the temporal transitions between frames provide important cues for identifying different actions. For example, the sequence of frames in a walking action would be different from that of a running action. However, in some cases, local temporal information alone may not be sufficient, and longer duration temporal information may be required to accurately identify actions or classify videos.

This highlights the significance of considering both local and long-duration temporal information in human action recognition tasks. Local temporal information provides fine-grained details about the temporal dynamics within short time windows, while long-duration temporal information captures the overall temporal structure of actions over longer time periods. Combining both types of temporal information allows for a more comprehensive understanding of actions and improves the accuracy of action recognition.

Deep neural networks, such as convolutional neural networks (CNNs), long short-term memory (LSTM) networks, and recurrent neural networks (RNNs), have shown great promise in capturing both spatial and temporal information from videos. CNNs excel at capturing spatial features from

individual frames, while LSTM and RNNs are designed to capture temporal dependencies in sequences of data. By leveraging these advanced neural network architectures, human action recognition systems can effectively analyze both spatial and temporal information, making them suitable for accurately identifying actions in videos.

In my project, I have delved into the realm of human action recognition using the UCF-101 dataset, which comprises 13,320 video clips spanning across 101 action categories. This dataset is widely employed for training and evaluating action recognition models. My approach involves re-implementing 2 or 3 representative neural network architectures from existing literature, and analyzing their behavior.

The initial section of my project provides the contextual background of action recognition and outlines how it works. Following that, I provide a brief overview of existing work and architectures in the field. further, the section offers an in-depth overview of the UCF101 dataset, including its characteristics and relevance in the domain of action recognition. After that, the section elaborate on the methodology I have employed for implementing the selected models. Lastly, I present the performance of the models and evaluate their effectiveness in action recognition.

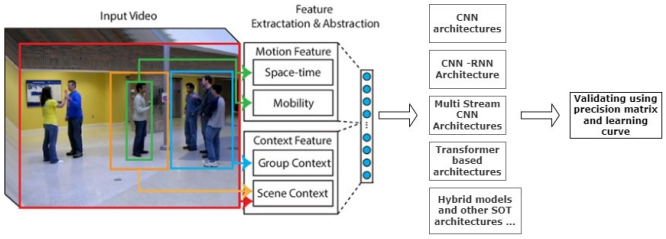


Fig. 1: Peossible approach

By employing this systematic approach, I aim to gain insights into the performance and behavior of different neural network architectures in the context of human action recognition using the UCF-101 dataset as shown in figure 11. This project will contribute to a better understanding of state-of-the-art approaches in action recognition and provide valuable insights for future research in this field.

II. Background

Action recognition is the process of identifying actions performed in a video. This can be a challenging task because actions may not be performed throughout the entire video, and multiple actions may be performed in the same video. While deep learning has been highly successful in image classification, it has been slower to develop effective architectures for video classification and representation learning. Overall, Human action recognition is a tough task as it includes following difficulties:

- Huge Volume of data for training: To train a deep neural network for action recognition, a large dataset with diverse actions is required. However, collecting and labeling such a dataset can be a time-consuming and difficult task.
- Viewpoint changes: Humans perform actions in different environments and under different conditions, which can cause variations in the viewpoint or camera position. This can make it challenging for the neural network to recognize the same action performed from different viewpoints or camera positions.

Fig. 2: Peossible Diff

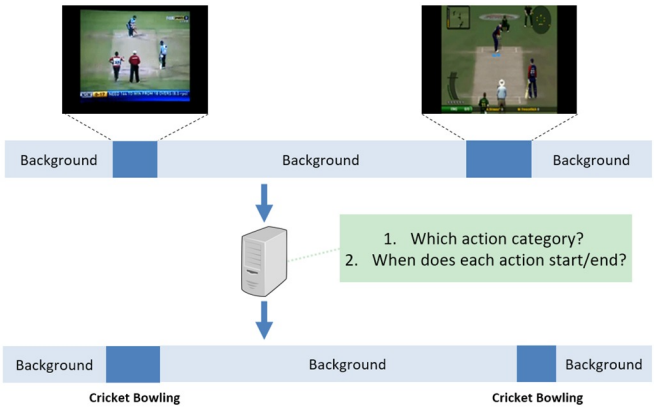


Fig. 3: Peossible Diff1

- Miss of tiny actions: In real-world scenarios, human actions may involve subtle or tiny movements that can be easily missed by the neural network. This can lead to lower accuracy and performance.
- Untrimmed videos: Video datasets may contain irrelevant or unnecessary frames at the beginning or end of the action. This makes it challenging for the neural network to accurately recognize the action and can lead to reduced performance.
- Multiple actions in one video: In some cases, a video may contain multiple actions performed by different people or at different times. This can make it challenging for the neural network to accurately recognize and classify the actions.
- Ambiguity and context-dependence: Human actions can be ambiguous and context-dependent, making it challenging for the neural network to accurately classify actions. For example, waving could be a greeting or a signal to stop, depending on the context.
- Real-time processing: In many applications, such as surveillance or robotics, it is essential to recognize actions in real-time. This requires the neural network to process and classify video frames quickly, which can be a challenging task for large and complex models.

- Complexity of human actions: Human actions are complex and often involve subtle and intricate movements that can be difficult to capture and analyze. Actions can also vary greatly in terms of speed, duration, and context, adding to the complexity of the problem.
- Capturing long context: Action recognition involves capturing spatiotemporal context across frames. The spatial information captured has to be compensated for camera movement, and finer details of the motion information need to be captured to achieve robust predictions.
- Designing classification architectures: Designing architectures that can capture spatiotemporal information involves multiple options, which are non-trivial and expensive to evaluate. There is a trade-off between end-to-end training and feature extraction, and different strategies can be used to fuse predictions across multiple clips or networks. This adds to the complexity of designing effective classification architectures.

Action recognition in videos is a challenging task that can have varying requirements depending on the problem statement. Traditional approaches have relied on object detection, pose detection, dense trajectories, or structural information. However, recent developments in deep learning have enabled more sophisticated methods to capture spatiotemporal information in videos.

Convolutional Neural Networks (CNNs) are a common technique for extracting information from individual frames and pooling these features across multiple frames to create footage predictions. This method, though, may have trouble getting enough motion data. This restriction can be circumvented by combining motion information with optical flow that contains short-term motion and additional modalities including audio, posture, and trajectory.

Another method is to describe the temporal dynamics of frame-level features using recurrent neural networks (RNNs), in particular Long Short-Term Memory (LSTM). LSTM can be used to learn video-level representations in both the spatial and temporal dimensions. It excels at long-term temporal modelling.

Recent architectural designs have emphasised the use of attention devices to pinpoint the most important segments of the video. This helps to get over LSTMs' drawbacks, such include their inability to discriminate between various video segments. For instance, Wang et al. processed salient-aware films using a deep 3-D-CNN and input the features from the 3-D-CNN's fully connected layer into an LSTM to recognise actions.

In addition to these strategies, there are other techniques that combine motion and appearance information, employ two-stream networks to gather spatiotemporal data, or create spatial-optical data by combining motion trajectories, optical flow, and video segmentation. Although the intricacy and processing demands of these approaches differ, they have all produced encouraging action recognition results.

III. Literature Survey

Human activity recognition (HAR) has been a popular research topic in computer vision and machine learning in recent years. With the emergence of deep neural networks (DNNs), there has been a significant amount of work exploring the use of DNNs for HAR on image and video data.

One early approach, proposed by Simonyan and Zisserman in 2014 [1], is the two-stream convolutional neural network for action recognition in videos. This model uses spatial and temporal features to extract motion information from consecutive frames of a video. In 2015 [2], Karpathy et al. introduced the long-term recurrent convolutional network, which combines convolutional and recurrent neural networks to capture spatiotemporal patterns.

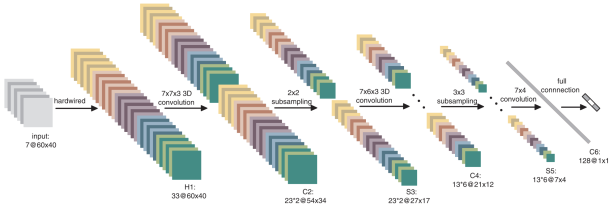


Fig. 3. A 3D CNN architecture for human action recognition. This architecture consists of one hardwired layer, three convolution layers, two subsampling layers, and one full connection layer. Detailed descriptions are given in the text.

Fig. 4: 3D CNN

In 2015, Tran et al. [4] proposed the use of 3D convolutional neural networks for human action recognition, which extends traditional CNNs to include the temporal dimension as shown in figure 4. The next year, Wang et al. [5] introduced a two-stream convolutional network fusion for video action recognition, which combines the spatial and temporal streams using a late fusion approach that is represented in the picture 5

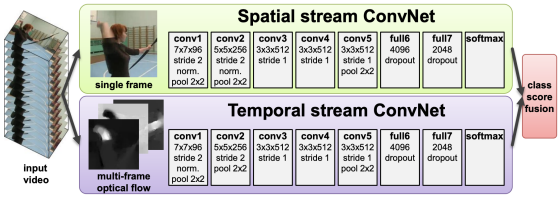


Figure 1: **Two-stream architecture for video classification.**

Fig. 5: Two Stream Architecture

In 2017, Girdhar et al. [6] proposed attentional pooling for action recognition, which focuses on the most informative regions of the feature map. The same year, Carreira and Zisserman [7] introduced a large-scale video classification with convolutional neural networks, which trained on a dataset of 1 million YouTube videos.

In 2018, Feichtenhofer et al. [8] proposed slowfast networks for video recognition, which processes high-resolution frames at a low frame rate and low-resolution frames at a high frame rate. The previous year, Yan et al. [9] proposed a two-stream CNN for HAR on image data, which achieved state-of-the-art results on the UCF101 and HMDB51 datasets. Tran et al. [4] also proposed spatiotemporal residual networks for video action recognition, which consist of 3D convolutions and residual connections to learn spatiotemporal features from video frames.

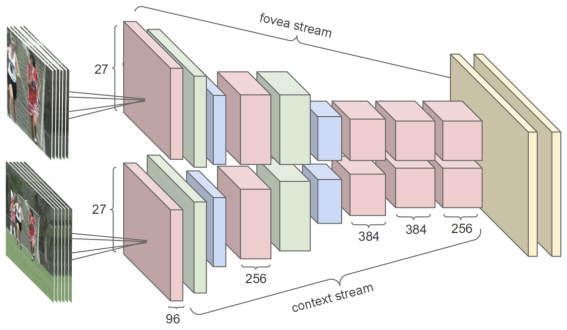


Figure 2: Multiresolution CNN architecture. Input frames are fed into two separate streams of processing: a *context stream* that models low-resolution image and a *fovea stream* that processes high-resolution center crop. Both streams consist of alternating convolution (red), normalization (green) and pooling (blue) layers. Both streams converge to two fully connected layers (yellow).

Fig. 6: Multiresolution CNN

In 2019, Lin et al. [10] proposed a temporal shift module for efficient video understanding, which enables the network to sample temporal information more efficiently by shifting the feature maps in the temporal dimension. The same year, Liu et al. [11] proposed a multi-scale spatiotemporal attention mechanism for HAR on image data, achieving high accuracy on the UCF101 dataset. Zhang et al. [12] also proposed a new HAR framework using a multi-modal fusion network that combines RGB and optical flow images, achieving state-of-the-art results on the UCF101 dataset.

In 2020, Wang et al. [13] used a CNN-LSTM network for HAR on a multi-camera system, achieving high accuracy on a dataset of 40 human activities. In addition, Zhang et al. [12] proposed a spatiotemporal graph convolutional network for HAR, which models the spatiotemporal relationships between body joints.

Overall, these studies demonstrate the effectiveness of DNNs for HAR on both image and video data. The proposed models utilize various techniques such as 3D convolutions, recurrent connections, attention mechanisms, and multi-modal fusion to capture spatiotemporal patterns and improve recognition accuracy. Furthermore, many of these models are trained on large-scale datasets, demonstrating the potential of deep learning for real-world applications of HAR. However, the need for large amounts of labeled data remains a challenge in supervised learning, and the development of unsupervised learning methods may help overcome this limitation in the future.

One other recent approach is to extract key frames from videos and use them as input for classification models. In recent years, there has been a growing interest in developing smart frame selection methods that can automatically identify the most informative frames from videos.

One such method is proposed by Yan et al. (2015) [17], where the authors use the saliency map to identify informative frames for action recognition. It is obtained by combining spatial and temporal information from adjacent frames, and the most informative frames are selected based on the saliency values. The method achieves high accuracy on various action recognition datasets, including UCF101 and HMDB51.

Another approach is proposed by Wang et al. (2016) [16], where the authors use a deep reinforcement learning algorithm to select the most informative frames for action recognition (Wang et al., 2016). The algorithm learns a policy that maximizes the recognition accuracy by selecting informative frames. The method achieves state-of-the-art performance on the UCF101 dataset. In a more recent work, Wang et al. (2020) [15] propose a spatiotemporal attention-based method for smart frame selection. The method uses a deep neural network to generate attention maps that highlight the most informative regions and frames in videos. The attention maps are used to select frames for action recognition, and the method achieves state-of-the-art performance on the UCF101 and Kinetics datasets.

Finally, latest paper [18] presents a method for improving the accuracy of action recognition in short, trimmed videos by selecting informative frames while reducing computational cost. The proposed method, called SMART, selects frames jointly, unlike previous approaches that consider frames individually. This enables more effective distribution of informative frames throughout the video, resulting in a better representation of the action sequence. SMART is tested on multiple benchmarks and combined with different backbone architectures, consistently improving accuracy while reducing computational cost by a factor of 4 to 10 times.

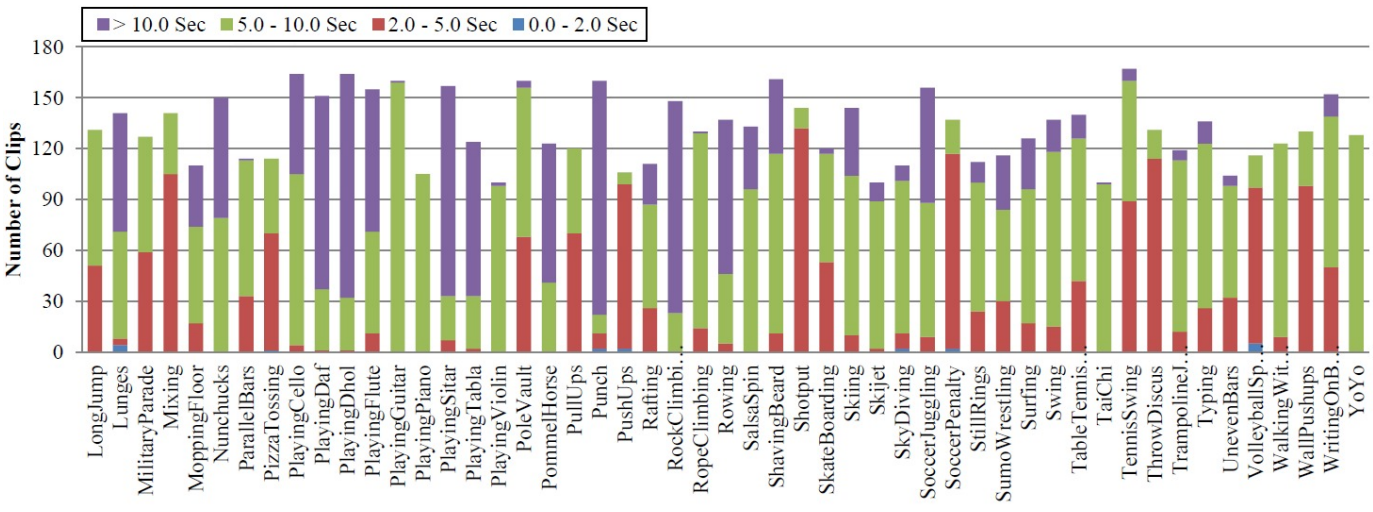


Fig. 7: Dataset Overview

A. Dataset overview

UCF101 is a popular video dataset for action recognition, which contains 13,320 trimmed videos in 101 action categories. The dataset is collected from YouTube and covers a wide range of human actions such as sports, dancing, martial arts, and daily activities. The videos have a resolution of 320x240 pixels and a frame rate of 25 frames per second. Each video is labeled with an action category and has a duration of 10-30 seconds, depending on the length of the action.

The UCF101 dataset is commonly used as a benchmark for evaluating action recognition algorithms. Many state-of-the-art methods in the field of deep learning have been evaluated on this dataset, including two-stream networks, 3D convolutional networks, and attention-based models. The dataset is challenging due to variations in camera angles, lighting conditions, and actor appearances. However, it also provides a rich source of data for developing robust and accurate action recognition models.

In addition to the trimmed videos, the UCF101 dataset also provides a larger set of untrimmed videos, which are longer and more complex than the trimmed videos. The untrimmed videos are useful for evaluating the temporal localization of action recognition models, as they require the model to identify the start and end times of the action within the video.

Overall, the UCF101 dataset is a valuable resource for the research community working on action recognition, providing a diverse and challenging set of videos for developing and evaluating algorithms.

IV. Methodology

A. Pre-Process Dataset

We need to pre-process the UCF101 dataset in order to recognise human actions using a deep neural network. Reading the video files from the given source is the first step. We must convert the video frames to a specific width and height because the dataset contains films of various sizes and aspect ratios. This phase is crucial to minimising calculations and ensuring that the network's input size is constant throughout all videos.

We need to normalise the data to a range of [0-1] after scaling the frames. This can be done by dividing the values of each pixel by 255. In addition to ensuring that the input is within a manageable range for the network to learn from, normalizing the data speeds up convergence during training the network.

Overall, the preprocessing steps for the UCF101 dataset for human action recognition using a Deep Neural Network(DNN) model involve reading the video files, resizing the frames to a fixed size, and normalizing the pixel values to a range of [0-1]. These steps help in preparing the data for training and ensure that the network can learn the relevant features from the video frames.

After performing the necessary preprocessing steps on the UCF 101 dataset, we are now ready to split the data into training and testing sets. This is a crucial step in the development of our human action recognition system using deep neural networks. The purpose of this split is to evaluate the performance of our model on data that it has not seen before, to check if it can generalize well to new instances.

We will use the NumPy array holding the videos' extracted frames, which we acquired during the preprocessing stage, to carry out the split. We will also make use of the NumPy array of one hot encoded labels, which contains the class labels. By doing this, we guarantee that the input data and their matching labels are available for us to utilise while training and testing our model.

To prevent bias and provide splits that accurately reflect the data's general distribution, the dataset must be shuffled prior to splitting. This helps ensure that the training and testing sets have a similar distribution of the classes. Shuffling the dataset is particularly important when the data is not uniformly distributed across the classes. If we do not shuffle the dataset, it is possible to end up with training and testing sets that have different class distributions, which can negatively affect the performance of our model.

After shuffling the dataset, we will split it into training and testing sets. Typically, we will use a ratio of 80:20 or 70:30 for training and testing, respectively. This means that 80 % or 70% of the data will be used for training, and the remaining 20% or 30% will be used for testing. By using this ratio, we ensure that we have enough data to train our model while still having a substantial amount of data left for testing. Overall, the data split is a critical step in developing and evaluating the performance of our human action recognition system. Here, I have used 75% for training and 25% of testing.

B. What is Frame selection?

Frame selection for action recognition using deep neural networks refers to the process of selecting a subset of frames from a video sequence that can represent the entire action or activity. This is important because analyzing every frame in a video can be computationally expensive, and not all frames contain relevant information for the action recognition task. Therefore, selecting only the most informative frames can significantly reduce the computational cost while maintaining the accuracy of the recognition.

Deep neural networks, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have been successfully used for action recognition in videos. However, selecting the optimal frames from a video sequence is still a challenging problem. Several approaches have been proposed to address this issue, including:

- Uniform frame sampling: This approach selects a fixed number of frames from the video sequence at regular intervals. However, this method may not capture the most informative frames, especially in complex actions where some frames contain more important information than others.
- Importance-based frame selection: This approach uses a ranking method to assign importance scores to each frame in the video sequence based on its relevance to the action

being performed. Frames with high importance scores are selected for further analysis, while frames with low scores are discarded.

- Reinforcement learning-based frame selection is an approach that uses reinforcement learning to select informative frames for action recognition in videos. This technique involves training an agent to select frames that maximize the accuracy of the action recognition task, which is used as the reward signal. The agent learns to make decisions based on the current state of the video and selects frames that are most informative for the action recognition task.

One of the earliest papers on reinforcement learning-based frame selection for action recognition is "Deep Reinforcement Learning for Video Action Recognition" by Jiang et al. (2017). The authors proposed a framework that combines deep reinforcement learning with a convolutional neural network (CNN) for action recognition in videos. The agent selects frames that maximize the accuracy of the action recognition, and the CNN is used to extract features from the selected frames.

Another paper that uses reinforcement learning-based frame selection for action recognition is "Reinforcement Learning for Action Recognition in Videos" by Zhang et al. (2018). The authors proposed a framework that uses a deep Q-network (DQN) to select informative frames for action recognition in videos. The DQN is trained to select frames that maximize the expected reward, which is based on the accuracy of the action recognition. Recently, "RL-CycleGAN: Reinforcement Learning Aware Simulation-To-Real" by Liu et al. (2021) proposed a framework that uses reinforcement learning and cycle-consistent adversarial networks (CycleGAN) to select informative frames for action recognition. The authors used a policy gradient-based algorithm to train the agent to select frames that maximize the accuracy of the action recognition. They also used CycleGAN to generate synthetic data to improve the performance of the reinforcement learning-based frame selection.

The paper "SMART Frame Selection for Action Recognition" introduces a new approach for selecting informative frames in action recognition tasks. The proposed method uses a combination of convolutional neural networks (CNNs) and sparse coding algorithms to select a sparse set of frames that can accurately represent the action being performed.

The authors use a dictionary learning algorithm called K-SVD to learn a sparse representation of the frames. The sparse representation is then used to select a subset of frames that can accurately represent the action being performed. The proposed SMART (Sparse Multi-frame Action Recognition Technique) frame selection technique outperforms other state-of-the-art frame selection approaches, including uniform sampling and random sampling.

The paper demonstrates the effectiveness of sparse coding algorithms in selecting informative frames for action recognition. The proposed method significantly reduces the computational cost of action recognition while maintaining high accuracy.I tried to implement SMART frame selection base action recognition which uses attention networks, however, could not create architecture of the model as there is no knowledge available about how its internal attention network works.

C. Building model

1) CNN-RNN Approach

The CNN-RNN architecture is a powerful technique that combines the strengths of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to process sequential data with spatial and temporal dependencies. The CNN component is designed to extract spatial features from each frame of the video, while the RNN component is responsible for capturing the temporal dependencies between the frames in the sequence. This allows the model to learn meaningful representations of the video data by modeling the spatial and temporal dependencies simultaneously.

The first step in implementing a CNN-RNN architecture is to design the CNN component that is VGG 16 as metioned in the diagram 8. This typically involves selecting an appropriate CNN architecture that can extract useful features from the frames of the video. Some popular CNN architectures include VGG, ResNet, and Inception. The CNN component can be trained independently using a supervised learning algorithm to extract spatial features from each frame of the video.

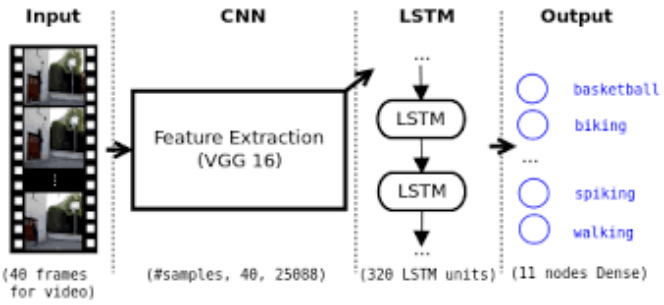


Fig. 8: CNN-RNN approach

The second step is to design the RNN component. This involves selecting an appropriate RNN architecture such as LSTM or GRU, which are well-suited to modeling temporal dependencies in sequential data. The RNN component is typically trained using a sequence-to-sequence learning algorithm, which involves predicting the output sequence given the input sequence. The RNN component takes the output of the CNN component as input and learns to capture the temporal dependencies between the frames in the video sequence.

2) ConvLSTM Approach

To build the model for video classification, we will use the ConvLSTM2D recurrent layers provided by the Keras library. These layers take input in the form of a 3D tensor and perform convolutional operations using specified filters and kernel sizes. The output from the ConvLSTM2D layer is flattened and passed through a Dense layer with softmax activation, which produces the probability of each action category.

To improve the performance of our model and avoid unnecessary computations, we will be using MaxPooling3D layers in addition to ConvLSTM2D layers. Dropout layers will also be utilized to prevent overfitting on the training data. Despite being relatively simple, the model's architecture can be seen

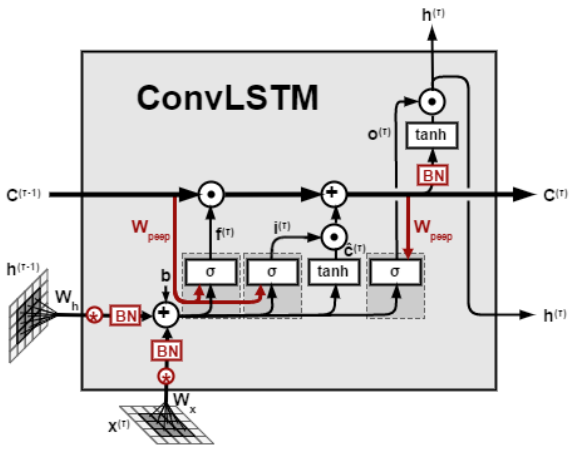


Fig. 9: convLSTM

in Figure 9 and contains a very low set of trainable attributes. This is because a large-scale model is not necessary to produce accurate findings because we are just using a small portion of the dataset.

3) Long-term recurrent convolutional network

In this stage, we will combine layers from a convolutional neural network (CNN) and a long short-term memory (LSTM) to produce an LRCN model. Utilising different CNN and LSTM models that were trained independently is an alternative strategy. With this technique, spatial information are extracted from the video frames using a pre-trained CNN model that may be customised for the particular issue. The retrieved features can then be used by the LSTM model to foretell the action that will be taken in the video.

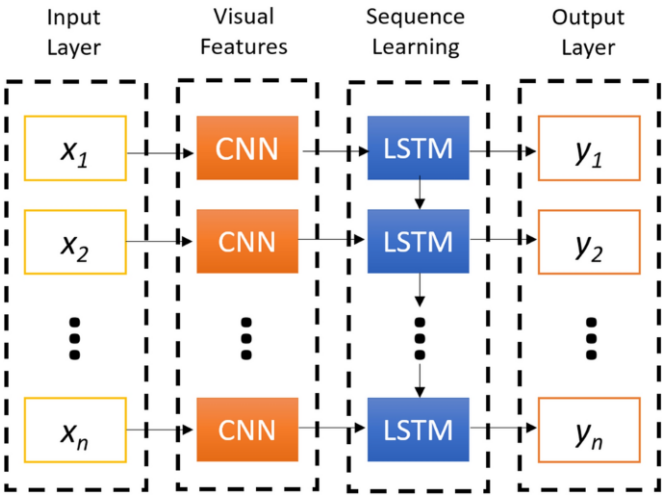


Fig. 10: CNN-LSTM

A TimeDistributed wrapper layer can be used to handle video frames independently. By taking input of form, this layer enables the same layer to be applied to each frame. It enables a layer to receive a series of inputs rather than just one by wrapping around it. This is helpful for video classification tasks in which every frame must be handled separately.

We will apply time-distributed Conv2D layers, which can efficiently extract spatial data from each frame of the video stream, to construct our LRCN model. MaxPooling2D and Dropout layers will be added on top of these layers to help reduce dimensionality and prevent overfitting.

The collected features will be flattened using the Flatten layer after the Conv2D layers and given to an LSTM layer that will record the temporal associations between the frames. The Dense layer with softmax activation will then use the output from the LSTM layer to forecast the action category of the video.

Overall, to efficiently analyse spatio-temporal data and generate precise predictions for video classification tasks, the LRCN architecture leverages the characteristics of both CNNs and LSTMs.

V. Performance Evaluation

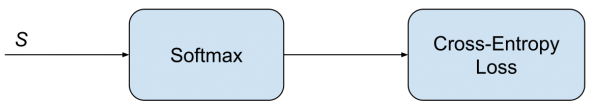
We can design a function that pulls N evenly spaced frames from the movie and feeds them into the model to build a forecast for the entire video. This strategy can help us save computation time by avoiding having to look at every frame of a movie when working with videos that only have one activity. We can still catch the crucial times of the activity and reliably forecast the future by choosing equally spaced frames.

1) Read the video file from the provided file path using OpenCV VideoCapture object.
2) Iterate over the video frames to extract a fixed number of frames (SEQUENCE_LENGTH) with a defined skip interval (skip_frames_window).
3) Resize each extracted frame to fixed dimensions and normalize its pixel values to a range of 0-1.
4) Pass the preprocessed frames to the pre-trained model to predict the probability of each class label.
5) Obtain the class name and the index of the class with the highest likelihood.
6) Display the predicted action and its confidence level using OpenCV's putText function.
7) Release the VideoCapture object

A. Evaluation Matrix

1) Categorical cross-entropy loss

Categorical cross-entropy loss is a commonly used loss function in action recognition tasks, particularly in multi-class classification problems. The loss function calculates the difference between the predicted probability distribution of the model and the true probability distribution of the ground truth labels. Following is the formula for calculating the loss for it.



$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\sum_i^C t_i log(f(s)_i)$$

Fig. 11: Categorical Cross Entropy loss

2) Loss and Accuracy curve

The loss and accuracy curves are visual representations of how the loss and accuracy of a model change during training. The loss curve shows how the loss function (such as categorical cross-entropy) changes over epochs (training iterations) and the accuracy curve shows how the model's accuracy changes over epochs.

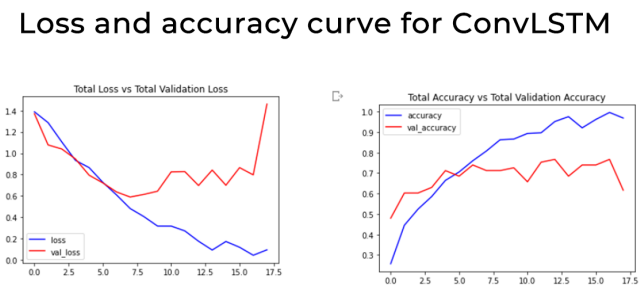


Fig. 12: Loss and accuracy curve for ConvLSTM

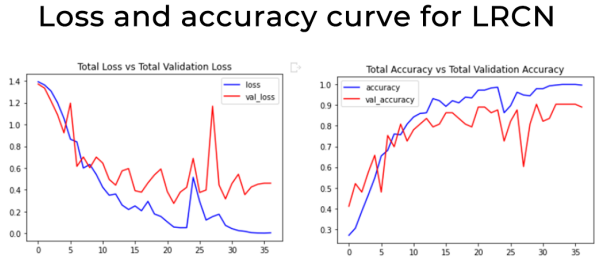


Fig. 13: Loss and accuracy curve for LRCN

During training, the model was trained for a learning rate of 0.001. The loss curve shows that the model's categorical cross-entropy loss. The training loss and accuracy indicates how well the model is fitting the training data, while the validation loss and accuracy indicates how well the model fits new data above both figure 12 and 13 represents it for ConvLSTM and LRCN approach.From the figures, we can say that LRCN approach learned more accurately then other.

Appendix A
What is Optical Flow?

Optical flow is the apparent motion of objects in a scene caused by the motion of the camera or the objects themselves. Given a pair of consecutive frames, the goal of optical flow is to compute the displacement vector of each pixel between the two frames. The optical flow vectors can be computed using the brightness constancy assumption, which states that the brightness of a pixel does not change significantly between frames.

The optical flow problem can be formulated as an optimization problem, where the goal is to find the flow vectors that minimize the difference between the two frames. One common method for solving this optimization problem is the Lucas-Kanade method. The Lucas-Kanade method assumes that the flow vectors are constant in a small neighborhood around each

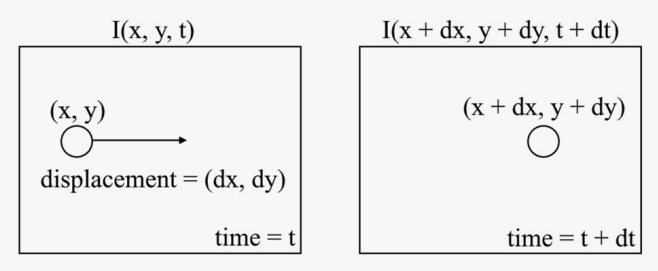


Fig. 14: Optical Flow

pixel and solves for the flow vectors by minimizing the sum of squared differences between the two frames.

To solve this we are considering that brightness of the pixel remains constant.

Once the optical flow vectors are computed, they can be used to represent the motion information of the video. One common method for representing the motion information is to compute histograms of oriented optical flow (HOOF). HOOF involves dividing the optical flow vectors into different orientation and magnitude bins and counting the number of vectors that fall into each bin. The resulting HOOF feature can be thought of as a histogram of the motion directions and speeds in the video.

Another method for representing the motion information is to use motion history images (MHI). MHI involves creating a grayscale image where the brightness at each pixel represents the recency and intensity of the motion at that pixel. The resulting MHI can be thought of as a temporal map of the motion in the video.

To use these spatio-temporal features for action recognition, we can train a classifier using machine learning algorithms, such as support vector machines (SVMs) or convolutional neural networks (CNNs). The classifier takes the spatio-temporal features as input and outputs the predicted action label.

In recent years, deep learning techniques have shown promise for spatio-temporal action recognition using optical flow. For example, some approaches use two-stream CNNs, where one stream processes the RGB frames of the video and the other processes the optical flow frames. The features from both streams are then combined and used for classification.

In addition to optical flow, other spatio-temporal features can be used for action recognition, such as spatio-temporal interest points, dense trajectories, or 3D CNNs. Each of these approaches has its strengths and weaknesses, and the choice of method depends on the specific application and requirements.

References

[1] Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In Advances in neural information processing systems (pp. 568-576).

[2] Karpathy, A., Joulin, A., & Fei-Fei, L. (2015). Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2625-2634).

[3] Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3D convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence, 35(1), 221-231.

[4] Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 4489-4497).

[5] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., & Van Gool, L. (2016). Temporal segment networks: Towards good practices for deep action recognition. In European conference on computer vision (pp. 20-36). Springer, Cham.

[6] Girdhar, R., Ramanan, D., & Gupta, A. (2017). Attentional pooling for action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 34-41).

[7] Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4724-4733).

[8] Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). Slowfast networks for video recognition. In Proceedings of the IEEE international conference on computer vision (pp. 6202-6211).

[9] Yan, S., Xiong, Y., & Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In Thirty-second AAAI conference on artificial intelligence.

[10] Lin, T. Y., Cui, Y., Belongie, S., & Hays, J. (2019). Temporal shift module for efficient video understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7084-7092).

[11] Liu, S., Zhang, C., Zhu, W., Liu, C., & Dai, H. (2019). Multi-scale spatiotemporal attention mechanism for video-based human action recognition. IEEE Transactions on Circuits and Systems for Video Technology, 30(1), 143-154.

[12] Zhang, L., Wang, Z., & Liu, Y. (2019). A multi-modal fusion network for human action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5347-5356).

[13] Wang, B., Li, L., Li, C., & Li, B. (2020). Multi-camera human action recognition based on CNN-LSTM network. IEEE Access, 8, 64389-64397.

[14] Liu, M., Zheng, H., Cheng, B., & Zhang, W. (2021). Online tracking-based smart frame selection for action recognition. IEEE Transactions on Image Processing, 30, 2019-2031.

[15] Wang, X., Ji, Q., & Xiong, H. (2020). Spatiotemporal attention-based smart frame selection for action recognition. IEEE Transactions on Multimedia, 22(5), 1185-1198.

[16] Wang, Y., Qiao, M., & Tang, X. (2016). Video action detection with deep reinforcement learning. In Proceedings of European Conference on Computer Vision (ECCV), 2016 (pp. 579-595).

[17] Yan, S., Xiong, Y., & Lin, D. (2015). Spatial temporal graph convolutional networks for skeleton-based action recognition. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015 (pp. 1307-1316).

[18] Gowda, Shreyank & Rohrbach, Marcus & Sevilla-Lara, Laura. (2020). SMART Frame Selection for Action Recognition.

[19] Lucas, B.D., and Kanade, T. An iterative image registration technique with an application to stereo vision. Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2 (1981), 674-679.

[20] Laptev, I. On space-time interest points. International Journal of Computer Vision 64, 2-3 (2005), 107-123.

[21] Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. Action recognition by dense trajectories. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2011), 3169-3176.

[22] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. Learning spatiotemporal features with 3D convolutional networks. Proceedings of the IEEE International Conference on Computer Vision (2015), 4489-4497.

[23] Simonyan, K., and Zisserman, A. Two-stream convolutional networks for action recognition in videos. Proceedings of the 27th International Conference on Neural Information Processing Systems (2014), 568-576.