**Part A**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Group Number 79**   |  |  |  | | --- | --- | --- | | **Member Names** | **BITS Id** | **Contribution** | | M S ANJANA | 2023ac05498 | 100 % | | LADDHA SACHIN RAMBILAS | 2023ac05564 | 100 % | | NAKUL V | 2023ac05741 | 100 % | | NEGI NUPUR RAVINDERSINGH | 2023ac05812 | 100 % | |
| **Domain** - Human Activity Recognition from videos |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Paper 1** | **Paper 2** | **Paper 3** |
| **Title of the paper** |  | ARNets: Action Recurrent networks for Human Action Recognition | ViT-ReT: Vision and Recurrent Transformer Neural Networks for Human Activity Recognition in Videos |
| **Authors** |  | Guangjun Zhang, Xiaobo Cai, Guangyu Gao, Zihua Yan, Lian Shu, Zhihui Hu | James Wensel, Hayat Ullah, and Arslan Munir |
| **Year of publication** |  | 2023 | 2023 |
| **Architecture of Deep Learning** |  | The architecture of the deep learning model described in the paper is based on a Recurrent Neural Network (RNN) framework enhanced with Convolutional Long Short-Term Memory (ConvLSTM) units, which are specifically designed for processing sequential video data.   The model consists of two ConvLSTM layers that capture temporal dependencies across sequences of frames, enabling the network to process both spatial and temporal features. The architecture uses tanh activation functions for updating memory cells and sigmoid activations in the input, forget, and output gates to control the flow of information through the recurrent layers. A softmax layer at the output generates class probabilities for multi-class action recognition tasks.  Another feature of the architecture is its integration of a pre-trained BN-Inception network as a backbone, which extracts spatial features from the video frames. These spatial features are then processed by the ConvLSTM layers to learn temporal relationships.   The design includes six alternative design configurations to optimize performance for different action recognition tasks. | ViT-ReT - Vision Transformer - Recurring Transformer Vision Transformer (ViT)  * Purpose: This component replaces traditional Convolutional Neural Networks (CNNs) to extract spatial features from video frames. * How it works:   + Video frames are divided into smaller patches, such as 8×88 \times 88×8 pixel grids. These patches are then embedded into a sequence of vectors.   + A self-attention mechanism processes these vectors to learn the relationships between different patches, effectively capturing spatial features.   + The ViT uses pre-trained weights from large datasets like ImageNet, making it efficient for transfer learning. * Unique Features:   + It is lightweight compared to deep CNNs like ResNet, making it ideal for resource-constrained environments.   + Unlike CNNs, it doesn't rely on convolutional layers; instead, it treats image patches as sequences, akin to words in a sentence.  Recurrent Transformer (ReT)  * Purpose: This component replaces traditional Recurrent Neural Networks (RNNs) to model temporal dependencies in video data. * How it works:   + The ReT processes sequences of video frames with positional encodings to maintain the order of the frames.   + Unlike RNNs, which process sequences step-by-step, the Transformer encoder can parallelize computations, greatly improving efficiency. * Unique Features:   + It avoids the bottleneck associated with RNNs, making it faster and more scalable for long sequences.   By integrating these two components, the ViT-ReT framework efficiently handles the dual challenge of spatial and temporal feature extraction in video-based human activity recognition. |
|
|
|
|
| **How is the network helping the overall task?** |  | The network is designed to excel in human action recognition by addressing the critical need to analyse both spatial and temporal aspects of video data.  It achieves this by integrating spatial feature extraction and temporal modelling into a unified framework.  Spatial features, such as static object shapes and scene layouts, are captured using the pre-trained BN-Inception network, which processes individual video frames. Temporal dependencies, such as the sequence and flow of actions, are modelled through ConvLSTM layers, which analyse the relationships between consecutive frames. This combination enables the network to understand not just what objects or scenes are present in a video but also how they change over time, which is essential for recognizing dynamic actions.  The network’s ability to process RGB frames, optical flow, and RGB differences ensures that it captures both static and motion-related features, enhancing its understanding of human actions. By leveraging recurrent layers, the network can identify patterns across frames, such as a person lifting an object or jumping, rather than relying on single-frame analysis. This temporal reasoning capability makes the network particularly suited for complex tasks where actions unfold over time.  The softmax layer at the output enables classification of the extracted features into specific action categories, fulfilling the overall task of action recognition. | Human activity recognition involves understanding not just what is visible in individual frames but also how these frames relate over time. For example, recognizing a “high jump” requires analyzing the motion and sequence of the activity.   * Feature Extraction:   + The ViT handles spatial feature extraction, analyzing what is happening in each individual frame, such as identifying objects or body parts. * Temporal Modelling:   + The ReT looks at how these features change over time, making it possible to understand actions like walking, jumping, or waving. * Efficiency:   + This approach significantly reduces the computational cost compared to traditional CNN-RNN models while maintaining or improving accuracy. |
| **Training procedures** |  | Data Preprocessing:  The input data is pre-processed to include sequences of 10 consecutive video frames. These frames are processed in three forms: RGB frames for capturing spatial details, optical flow to emphasize motion, and RGB differences to highlight changes between frames. These pre-processed features provide the foundation for learning both static and dynamic aspects of the input.  Batch processing: During training, batches of 10 frames are fed into the network, ensuring that temporal relationships across the sequence are adequately modelled. The spatial features are first extracted using the pre-trained BN-Inception backbone, which is fine-tuned during training to adapt to the specific action recognition task. These extracted features are passed to the ConvLSTM layers, which specialize in capturing temporal dependencies by analysing sequential data.  Optimization:  The optimization is performed using Stochastic Gradient Descent (SGD  Learning rate:  The ConvLSTM layers, which are trained from scratch, use a higher learning rate of 1e-2 to speed up their optimization. Conversely, the backbone network, which is already pre-trained, is fine-tuned with a lower learning rate ofe-3to avoid disrupting the learned weights.  Momentum:  To further stabilize training, a momentum value of 0.9 is applied.  Regularisation:  Regularization is incorporated into the training process through a weight decay of 5e-4  Other notes: The training is conducted in an end-to-end manner, allowing the spatial and temporal features to be jointly optimized. This unified approach ensures that the network learns meaningful representations that integrate both motion and static information, critical for recognizing human actions. | Data Preprocessing:   * Video frames are resized to 224×224224 \times 224224×224 pixels, and only 20 frames per video are sampled. This reduces computational load without sacrificing key information. * Frames are preprocessed and stored to save time during training.   Optimization Algorithm:   * The Adam optimizer is used for faster and more stable convergence.   Hyperparameters:   * Batch size: 4 (small batches are used due to the computational complexity of Transformers). * Training epochs: 50.   Regularization:   * Dropout and normalization layers are employed to prevent overfitting.   Separate Training for Components:   * The Vision Transformer is pre-trained on ImageNet, and the Recurrent Transformer is trained on extracted features from video datasets. |
| **Evaluation / Performance metric used** |  | The evaluation of the network’s performance focuses on its ability to classify human actions accurately across different datasets. The primary metric used for evaluation is mean classification accuracy, which measures the proportion of correctly classified actions among all test samples. This metric was used to assess the network’s performance on datasets such as UCF-101, HMDB-51, and Kinetics.  Additionally, the paper highlights the use of Class Activation Maps (CAMs) as a visualization tool to understand the network’s focus during predictions. | * Accuracy: The percentage of correctly classified activities. * Loss: Measured using categorical cross-entropy. * Efficiency Metrics:   + Training Time: The time required to train the model.   + Throughput Time: How quickly the model can make predictions.   + Memory Usage: The amount of memory required for inference.   UCF101: A dataset of 13,320 videos across 101 categories (e.g., sports activities, gestures).   * URL: UCF101 Dataset.   Additional Datasets:   * YouTube Action Dataset. * HMDB51 Dataset. * UCF50 Dataset. |
| **Name of Dataset used. If a public dataset, provide the URL.** |  | The network was evaluated using several public datasets. Only the names of the dataset are given in the paper, URLs are not mentioned. We found the URL on the google.  1. UCF-101**:** [UCF-101 Dataset](https://www.crcv.ucf.edu/data/UCF101.php)  2.HMDB-51:  [HMDB-51 Dataset](https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/)  3. Kinetics: [Kinetics](https://deepmind.com/research/open-source/kinetics)  4. Moments in Time: [Moments in Time Dataset](http://moments.csail.mit.edu/) | <https://www.kaggle.com/datasets/matthewjansen/ucf101-action-recognition> |