University of Sheffield

Sports Video Classification and Summarization

Seunghyun Im

Supervisor: Yoshi Gotoh

A report submitted in fulfilment of the requirements
for the degree of Mcomp Computer Science(with Artificial Intelligence)

in the

School of Computer Science

May 20, 2025

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Seunghyun Im

Signature: Seunghyun Im

Date: 24 Oct 2024

Abstract

This project presents the development of a system for sports video classification and summarisation utilising deep learning model. The rapid growth of media data increased demand for efficient video analysis. Consequently, The study focuses on building a solution capable of both video classification and summarisation. The system trains CNN-LSTM architecture by employing UCF-101 to extract spatial and temporal features from video frames, enabling accurate classification across 56 sports-related actions. The original plan to generate summaries from longer output was adapted due to copyright and access limitations, resulting in a practical approach to upload a merged video with short clips as an input for summary generation. The report illustrates the technical pipline, including dataset optimisation, model configuration, and user interface development with PyQt6. The evaluation resulted high accuracy (over 98%) in classification and positive user feedback for the summarisation feature. Furthermore, all encountered limitations and potential improvements, such as expanding dataset variety, and supporting various features for the summarisation, are stated. Overall, the project demonstrates an effective methodology for sports video classification and summarisation, providing fundamental approaches for future advancements.

Contents

List of Figures

Chapter 1

Introduction

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus. Nullam quis ante. Etiam sit amet orci eget eros faucibus tincidunt. Duis leo. Sed fringilla mauris sit amet nibh. Donec sodales sagittis magna. Sed consequat, leo eget bibendum sodales, augue velit cursus nunc.

1.1 Aims and Objectives

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus. Maecenas tempus,

tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus. Nullam quis ante. Etiam sit amet orci eget eros faucibus tincidunt. Duis leo. Sed fringilla mauris sit amet nibh. Donec sodales sagittis magna. Sed consequat, leo eget bibendum sodales, augue velit cursus nunc.

1.2 Overview of the Report

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus. Nullam quis ante. Etiam sit amet orci eget eros faucibus tincidunt. Duis leo. Sed fringilla mauris sit amet nibh. Donec sodales sagittis magna. Sed consequat, leo eget bibendum sodales, augue velit cursus nunc.

Chapter 2

Literature Survey

The rapid evolution of deep learning techniques and access to large datasets has significantly advanced video classification and summarisation. This chapter discusses fundamental methods and datasets relevant to the project's objectives, focusing on deep learning models, such as Covolutional Neural Networks(CNN), Long Short-Term Memory Networks(LSTM) and HSA-RNN. In addition, the report introduces various datasets such as UCF-101, YouTube-8M, Sum-Me, TVSum and SoccerNet, which are large-scale video datasets used for classification and summarisation purposes.

2.1 Video Classification

As discussed in previous chapter, the amount of media data has dramatically increased in modern times. With the growing diversity of media and broadcasting platforms, the volume of data is expected to continue expanding in the future. The application of deep learning algorithms enables efficient utilisation of this increasing media data. The suggested models, such as CNN and LSTM, are reasonable approaches to the video classification task.

Simonyan and Zisserman (2014) represents use of hybrid Convolutional Neural Network, emphasising its capability for spatial and temporal analysis. As suggested by Wu et al. (2015), the project extended convolution networks by implementing a singular architecture with two recognition features: spatial and temporal. Simonyan and Zisserman (2014) further advanced this approach by applying two CNNs and combining their outputs into a single score. In addition, Wu et al. (2015) developed upon Simonyan and Zisserman (2014)'s methodology by integrating LSTM units with each ConvNet stream. Since the original CNN contains only temporal information, the spatial stream was necessary for effective video classification. This extension included 3 by 3 grid window to incorporate the spatial stream, thereby optimising the architecture. Consequently, the model, initially limited to processing two-dimensional data, was extended to function in a three-dimensional space by adding spatial information. This model was used in projects conduct by Akilan et al. (2019) and Islam et al. (2020), resulting average of 95% and higher percentage in image and 3-D recognition.

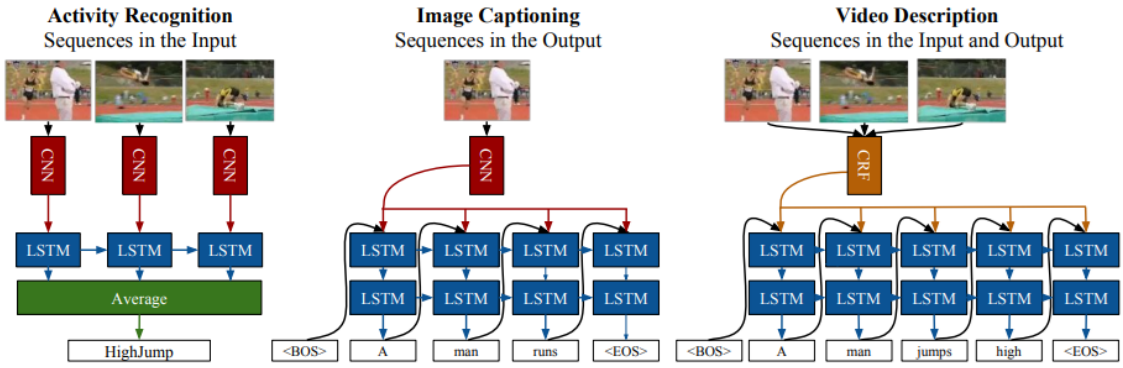


Figure 2.1: Demonstrating usage of LSTM layers for video description tasks (Donahue et al., 2014)

The last considerable model for this task is the addition of LSTM. The integration of LSTM into CNN models for video classification was presented by Wu et al. (2015), while Donahue et al. (2014) highlights the use of RNN and LSTM for video recognition, where CNN is applied first, followed by LSTM layers. The key difference between the two studies is that Wu et al. (2015) separates spatial and temporal components, whereas Donahue et al. (2014) illustrates three distinct approaches: activity recognition, image captioning, and video description. Activity recognition requires multiple CNNs to capture image features, with LSTMs sharing information for action classification. Secondly, image captioning employs a single CNN to generate outputs for multiple LSTMs, leading to diverse recognition results. Figure 2.1 provides a clear illustration of this methodology. Lastly, video description follows a similar method but begins filtering with CRF. These methodologies demonstrate how combining spatial and temporal features advances the effectiveness of video classification tasks. In conclusion, research suggests that deep learning models for video recognition are well-established. Recent approaches modify spatial and temporal features, setting new performance benchmarks on large datasets. The explained models, CNN and LSTM, are extended neural network architectures that demonstrate the effectiveness of these algorithms in achieving high performance metrics, as reported by Akilan et al. (2019) and Islam et al. (2020).

2.2 Video Summarisation

Part B of the project is video summarisation. Video summarisation focuses on extracting selected segments from the video. The system is required to extract only the categories requested by the user. According to Ghosh (2024), sports video summarisation traditionally relied on manual approaches. The task required manpower to identify highlight events and edit the videos. However, by adopting a machine learning system to automate this task, the time and resources consumed can be significantly reduced. Sanabria et al. (2022) utilises HSA-RNN to summarise the dataset called SoccerNet. According to their research,

Sanabria et al. (2022) states that their system splits the video into fixed-size segments using HSA-RNN. The system uses a segment size of 10 and sub-samples at 2 frames per second. As a result, the model achieved better performance than the LSTM model after testing. Moreover, the HSA-RNN model requires long videos containing various segments to achieve successful performance.

2.3 Deep Learning Models

In modern days, the Convolutional Neural Network (CNN) is a foundational deep learning algorithm for image recognition. Ghosh (2024) stated that the spatial and temporal features extracted by the CNN algorithm are suitable for video tasks, as they can effectively capture dynamic movements from the frames represented in video datasets.

2.3.1 CNN-LSTM

Wu et al. (2015) expressed that CNN demonstrated impressive achievements in image recognition tasks. However, the model was initially considered unsuitable for video recognition tasks, as it requires multiple frames to capture temporal dynamics. To address this limitation in large-scale tasks, Karpathy et al. (2014) extended CNN by stacking frames into fixed-size time windows and applying spatial convolution for video classification. The article introduces UCF-101, which contains a wide range of classified sports data. The authors Islam et al. (2020) proved that the extended CNN-based architecture effectively handles sophisticated features in both temporal and spatial video data. Building on this evidence, the project focuses on implementing a convolutional neural network architecture using the UCF-101 dataset.

As suggested earlier, the standard CNN is limited to spatial operations and requires additional methods to effectively recognise video content. Although, Tran et al. (2015) explains that the 3D Convolutional Network (C3D), proposed by Tran et al. (2015), allows both spatial and temporal operations. Wu et al. (2015) introduced a methodology using LSTM with CNN. The author demonstrated that this method employs two CNNs—one for extracting spatial features and another for capturing motion. The LSTM processes these features to capture long-term temporal relationships, providing context and sequence information for improved video classification. Tran et al. (2015) explained that C3D applies convolutional filters across both temporal and spatial dimensions, enabling a unified model for motion analysis in video scenes. This approach is particularly effective for classifying sports videos featuring key actions, such as kicks or serves in racket sports.

According to Wu et al. (2015), video classification tasks can be addressed using LSTM. The research suggests extracting two types of features from videos—spatial and motion—by applying two separate CNNs. These features are then processed by an LSTM to produce a single prediction score.

Tran et al. (2015) introduced an alternative approach by using C3D to extend CNNs into a 3D framework for video classification. This method incorporates a temporal dimension

alongside spatial dimensions, employing 3D convolutional kernels, temporal dynamics, and fixed temporal windows for feature representation.

The studies compared multiple algorithms using the UCF-101 dataset and demonstrated that C3D outperformed LSTM in classification accuracy, particularly for short video clips. However, the fixed temporal window of C3D limits its effectiveness for long-duration videos. Consequently, this limitation suggests a collaborative approach, integrating C3D with LSTM to leverage the long-term memory capability of LSTM for processing longer-duration inputs.

2.3.2 HSA-RNN

Hierarchical Structure-aware Recurrent Neural Network (HSA-RNN) is a model that captures the temporal structure and hierarchical semantic information of a video. Zhao et al. (2018) demonstrates that HSA-RNN is a supervised learning model consisting of three hierarchical levels: frame, shot, and global structure. A shot refers to a sequence of frames occurring before a transition in the video. The model enhances the quality of summarisation by detecting events at the shot level instead of the frame level. This hierarchical structure supports the following workflow:

- Extract frame features from the input video by using CNN
- Frame-level RNN employs the generated frame feature
- Group frames into shots
- Process with a shot-level RNN
- Generate importance scores for each shot
- Extract the video summary

The model is structured to detect input videos at the shot level instead of the frame level. Accordingly, HSA-RNN enhances the quality of detection by extracting crucial shots and generating a video summary based on the importance score assigned to each considered shot. However, the model requires a long video dataset that contains multiple shots to function effectively. Additionally, a dataset annotated with frame-level importance scores supports a more concise and accurate summarisation process.

2.4 Sports dataset

The project required a large-scale dataset containing diverse videos to support both classification and summarisation tasks. Despite the availability of such datasets, it was essential to ensure that the selected dataset did not include copyrighted material or, alternatively, provided open access for research purposes.

2.4.1 UCF-101

Sports datasets are essential to the advancement of this project in sports recognition tasks. These datasets provide the fundamental information necessary for action-spotting and related procedures required to train the model. In particular, the classification task utilises the UCF-101 dataset to train the CNN-LSTM model. UCF-101 contains short clips from 101 action classes, enabling the model to learn and perform action recognition effectively.

2.4.2 Youtube-8M

YouTube-8M is a dataset that provides 230,000 human-verified segment labels across 1,000 classes Abu-El-Haija et al. (2016). Despite its diversity, the dataset only supplies frames extracted from YouTube videos rather than the original video content. All feature and label data are available in TensorFlow format, encouraging researchers to work with the raw data. However, the dataset requires approximately 1.5 TB for frame-level data and 120 GB for video-level data, which poses limitations for researcher with insufficient storage capacity.

2.4.3 SumMe

SumMe is a dataset consisting of 25 short videos, ranging from 1 to 7 minutes in length Gygli et al. (2015). Although it is a suitable dataset for training video summarisation models, the dataset is no longer officially available. While it is still possible to locate this dataset, the sources are hosted on unofficial webpage, raising concerns about reliability and access.

2.4.4 TVSum

TVSum Song et al. (2015) is another dataset that provides 50 videos for video summarisation. However, this dataset has also been deprecated from official online sources. Consequently, the project was required to extend the research procedure to identify a suitable dataset for the intended purpose.

2.4.5 SoccerNet

SoccerNet, proposed by Giancola et al. (2018), includes five hundred games from European football leagues with more than six thousand annotations. According to the article, other existing video recognition datasets are often limited in providing annotations focused on video understanding and do not generate any video as a result. However, SoccerNet annotates all possible actions from 500 games, providing valuable annotations for video editing and enabling salient moment retrieval by linking camera shots with actions. Deliège et al. (2021) mentions a methodology called Context-Aware Loss Function (CALF), published by Cioppa et al. (2021), which addresses a method for video understanding.

2.5 Evaluation of the Project

In order to define the success of the project, the system's robustness and effectiveness must be evaluated using reliable metrics. Previous research by Ghosh (2024); Wu et al. (2015) illustrated that the performance of video classification was evaluated through accuracy measures. Moreover, Zhao et al. (2018) assessed performance based on F-measure, recall, and precision to demonstrate the project's effectiveness.

On the other hand, summarisation employs a different evaluation strategy. Part A includes train and test lists for data training and testing. However, the summarisation task generates a new video, for which no existing dataset is available for training or testing. Therefore, the assessment requires a different measurable method to evaluate the system. Accordingly, the project decided to use a human evaluation method to assess Part B.

2.5.1 Classification

Classification Evaluation Metrics:

- F-Measure:
  - $$\frac{2 \times Preicision \times Recall}{Precision + Recall}$$
  - Definition: A single measure that trades off precision versus recall is the F measure Manning et al. (2009)
- Recall:
  - $$\frac{TruePositive}{TruePositive + FalseNegative}$$
  - Definition: What fraction of the relevant documents in the collection were returned by the system? Manning et al. (2009)
- Precision:
  - $$\frac{TruePositive}{TruePositive + Positive}$$
  - Definition: What fraction of the returned results are relevant to the information need? Manning et al. (2009)
- Accuracy:
  -

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{y}_i = y_{true\_i})$$

Figure 2.2: Formula used to find accuracy for validation and training set Hossin and Sulaiman (2015).

The following equations represent the standard metrics used to evaluate the project's performance Hossin and Sulaiman (2015). These measures reflect the system's effectiveness in producing a summarized video that includes all crucial scenes from the input video.

Visual Representation

- **ROC Curve:** The ROC curve represents the performance of a binary classifier model Vilarino et al. (2006). Researchers use this curve to evaluate multiple classifier models. It plots the true positive rate (recall) against the false positive rate at various thresholds.
- **Confusion Matrix:** Wu (2022) described the confusion matrix as a table that represents the performance of a classification algorithm. Additionally, the table provides a visual representation of the data classification results.

2.5.2 Summarisation

It is important to visually demonstrate the evaluation of the summarisation task. However, since this project's summarisation focuses on generating a new video summary as output, there is no existing test data available to directly evaluate the results. Consequently, an alternative evaluation method—human evaluation—is employed. Participants will complete a questionnaire survey, and their responses will be used to assess the performance of the summarisation system.

Chapter 3

Requirements and analysis

This chapter outlines the key requirements and goals of the project, providing a detailed analysis of the resources necessary for its successful implementation. The project is divided into two parts: Part A, focusing on video classification, and Part B, on video summarization. Each part is analyzed to identify the required models, datasets, software, hardware, as well as functional and non-functional requirements. Additionally, this chapter discusses the evaluation metrics for both Part A and Part B. It also introduces the system structure by specifying the essential features needed for implementation.

3.1 Aims and Objectives

The main aim of the project is to train a deep learning model for sports video classification and summarization. The primary objectives are as follows:

- Objectives:
  - Implement an effective system for sports video classification.
  - Advance the classification model to develop a system for summarisation.
  - Effectively manage the large dataset.
  - Must be capable of managing the intensive computation by a suitable deep learning model.
  - Conduct evaluation for both tasks to represent performance of the implement system.

3.2 Requirements

Based on Chapter 2, the project requires specific models and datasets to implement applications for classification and summarization. To achieve the project objectives, it is necessary to identify the tools, platforms, and resources needed for development. This section

provides an explanation of the models, datasets, software, and hardware required to manage computation for both tasks.

3.2.1 Models for classification and sumarisation

- **CNN:** Chapter 2 highlights the algorithm's effectiveness in processing and analyzing visual data. According to Wu et al. (2015), CNN excels at extracting spatial features from images and video frames, making it ideal for the tasks in this project. The ResNet-18 model is employed to extract spatial feature vectors from each video frame. Its convolutional layers effectively capture the action patterns present in the UCF-101 dataset.

- **LSTM:** Wu et al. (2015) demonstrates that LSTM is a crucial algorithm for capturing the temporal dimension in media data. The spatial feature vectors extracted from each frame serve as sequential inputs to the LSTM architecture, allowing it to capture temporal dependencies effectively. While CNN extracts spatial features from individual frames, the combined CNN-LSTM model enables more effective analysis and improves video classification performance.

- **ResNet-18** ResNet-18 is a deep CNN architecture that facilitates efficient learning in deep networks. The model employs 18 layers to achieve robust image recognition. ResNet-18 is chosen for its high-performance feature extraction, benefits in transfer learning, and training stability. Consequently, this model is essential within the CNN-LSTM framework to enable effective image recognition.

- CNN's ability to efficiently extract spatial visual information, combined with LSTM's capability to learn temporal dependencies, enables accurate action classification and summarisation. Accordingly, the combination of these two models represents an optimal approach for learning and analyzing spatial-temporal patterns in action videos.

3.2.2 Sports Dataset

Video Classification and Summarisation

The selection of datasets is crucial for video recognition and summarisation tasks, as they form a fundamental requirement for training deep learning models. This project focuses on the UCF101 dataset, which contains 101 action classes in videos. UCF101 is well-known for its diversity and relevance in video classification research, particularly in action detection.

UCF101

UCF101, proposed by Soomro et al. (2012), consists 13,320 video clips categorized into 101 distinct actions derived from various sports videos. This dataset established a standard benchmark for action recognition. Its 101 action categories were classified using an SVM model applied to resources from the YouTube. Tran et al. (2015) demonstrated that UCF101

is highly effective for action recognition, achieving over 80 percentages accuracy with models like C3D, confirming its reliability for advanced implementation.

3.2.3 Hardware Requirements

Since this project employs large video datasets and deep learning models, it is anticipated to be computationally intensive. Therefore, it is necessary to specify the hardware requirements to support efficient execution. The following section outlines the essential hardware components needed for the project.

Part A: Classification

- **Computer Performance:** A high-performance computing setup capable of efficiently training large video datasets, such as UCF-101, using deep learning models.
  - **RAM:** A minimum of 16 GB, or 32 GB of RAM as a standard requirement.
  - **CPU:** Powerful multi-core CPU, minimum of Intel Core i7.
  - **GPU:** At least RTX 3060 is required with sufficient VRAM, to use optimal CUDA environment and pyTorch.
  - **Storage Requirement:** A minimum of 200 GB, or standard of 300 GB SSD is required for efficient loading and processing UCF-101.
- **Display:** The task requires a standard resolution for development and display to monitor training progress and results (accuracy and loss function). Additionally, the program should be capable of providing an estimated time of completion (ETA) to allow users to track the duration of the training process.
- **Runtime:** Based on the hardware requirements outlined above, training a deep learning model on the modified UCF-101 dataset is estimated to take a minimum of 60 hours for 100 epochs.
- **Configuration:** The task is implemented using the Python programming language within the VS Code environment, allowing the integration of computer vision libraries.

Part B: Summarisation

- **Computer performance:** The video summarization task requires a high-performance computer equipped with a minimum of 16 GB of RAM, with 32 GB recommended as the standard. A powerful CPU, such as an Intel i7, is necessary to efficiently handle frame extraction, segment scoring, and summarization. Additionally, an RTX 3060 GPU is the minimum requirement to support deep learning operations.
- **Storage:** A minimum of 200 GB of SSD storage is required to manage the large video datasets and the extracted frames generated from these videos.

- **Display:** An application interface is required to allow users to interact with the system. Additionally, features such as a loading percentage, progress bar, and other indicators are necessary to clearly represent the system's progress. After generating the output, the system should display the results to the users, and the summarized video should be easily recognizable and understandable.
- **Runtime:** Since Task B uses the same deep learning algorithm as Task A, no additional training is required. The task is expected to deliver efficient runtime performance in input recognition, summarized video generation, and testing procedures.
- **Configuration:** The task requires Python platforms, such as VSCode, which support libraries like PyTorch, NumPy, PyQt6, and others, ensuring compatibility with deep learning workflows.

3.2.4 Software Requirement

Based on Chapter 2, this section explains the required software components to implement the sports video classification and summarization system.

- Common requirements for video classification and summarisation:
  - **Programming environment and framework:**
    - Python: Both tasks A and B will be implemented in the Python environment. This framework offers high compatibility with external libraries, such as deep learning frameworks, OpenCV, and PyQt6, which are utilized in this project. This choice allows for a flexible and unrestricted use of libraries.
  - Deep learning framework:
    - PyTorch: PyTorch is a deep learning framework that provides a flexible and dynamic computational environment. It simplifies the development of the CNN-LSTM architecture, which is the key model for this project, making it ideal for both tasks. Its dynamic computation graph enables easier debugging and experimentation, while its extensive libraries, such as Torchvision, support and accelerate the deep learning workflow.
      - The framework for defining the CNN-LSTM model is implemented by importing relevant PyTorch libraries.
  - Torchvision: This library is part of PyTorch and offers a wide range of utilities, datasets, pre-trained models (such as ResNet-18), and transformation functions for image and video processing. The torchvision.transforms module is used for tasks like image resizing, tensor conversion, and normalization.
  - OpenCV: It is a comprehensive computer vision library designed for real-time applications. It handles video file processing, including tasks such as color space conversion and overlaying text on extracted frames.

- PyQt6 : This framework is used for developing graphical user interfaces (GUIs). It is an essential tool for implementing visual applications that enable user interaction, such as input uploading, classification, summarized video playback, and mo
  - Data management
    - Numpy: It is a fundamental Python library for efficient array operations. Given that the project involves handling various multidimensional arrays, this library is essential for efficient processing, transformation, and storage of large data.

3.2.5 Functional Requirements

This section outlines the essential components required for the system's operation and defines its core functionalities.

Input Handling

- Accept various video formats (e.g. .mp4, .avi, etc).
- Allow all video formats without any validation checks.

Video classification

- Input video
- Utilise CNN to extract spatial features and LSTM to capture temporal patterns.
- Trained CNN-LSTM checkpoint model (.pth) file to classify action class label.
- Display notification if the system fails.
- Display probability of the prediction.
- Allow random video classification

Video Summarisation

- Input: merged video with five random selected clips.
- Trained CNN-LSTM check point model (.pth) file to classify action patters.
- Display progress bar for the users to wait.
- Display notification if the system fails.
- List all detected actions.
- Require users to selection one of the action in the list.

- Based on user selected action, generate output video including title of the selected action displayed in the top-left corner of the output.
- Automatically display output video after program generation.

3.2.6 Non-functinoal Requirements

This section outlines the fundamental procedures of the system, highlighting its performance, usability, and reliability aspects.

- Performance:
  - Both summarisation and classification should not exceed 4 minutes runtime per video on average.]
- Usability:
  - Interactive GUI for the users.
- Reliability:
  - System should not crash.
  - Avoid copyright issues.
  - Efficient mechanism in adopting dataset.
  - System should progress without error crash.

3.3 Evaluation

This section outlines the evaluation strategy for the classification and summarization components. It defines the key metrics used to assess the performance of the CNN-LSTM model on the UCF-101 dataset. These metrics are essential for presenting the results of both training and validation.

3.3.1 Classification

Classification Evaluation Metrics

Hossin and Sulaiman (2015) defines evaluation metrics for classification problems.

- Accuracy: Measures how often the classifier gets the correct results.
- Precision: Demonstrates how many of the positive predictions were correct.
- Recall: Illustrates how many of the actual positive samples were predicted.
- f1-measure: It is the mean of precision and recall to represent imbalance.

Confusion Metrics

The confusion matrix is required to visually illustrate true and predicted classifications for each class, Wu (2022).

ROC Curve

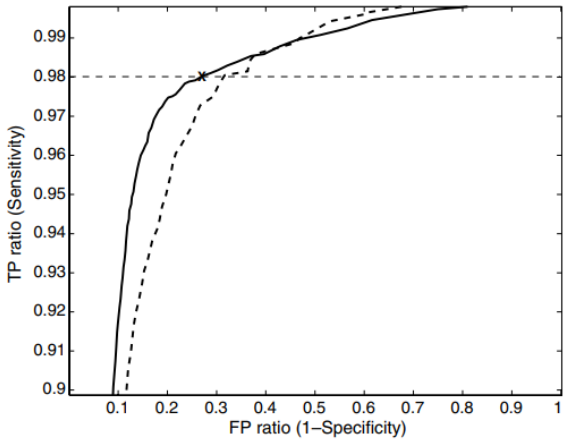


Figure 3.1: Example diagram representing ROC Curve, which visually represents class separability, Vilarino et al. (2006).

Additionally, it demonstrates the model's performance in distinguishing between similar sports actions, Vilarino et al. (2006).

3.3.2 Summarisation

Since the summarization task does not have a test dataset, it requires different metrics to evaluate the system's results. Accordingly, the project will conduct human evaluation by collecting survey responses. A questionnaire will be provided via Google Forms to gather participant feedback based on five criteria: relevance and accuracy, completeness, coherence, usefulness, and overall feedback. The collected responses will then be used to evaluate the system's performance.

Ethical Issue

Since this project incorporates human interaction to evaluate the performance of the summarisation, it involves ethical considerations related to participants. The ethical review will address the following key areas:

- **Personal Safety:**
  This term refers to any risks or threats to the participant's safety while conducting the survey.

- **Potential Participants**
  Description on invited participants
- **Advertising Methods**
  Methodology to invite participants
- **Payment**
  Explanations of whether the participants will be compensated for their time and contributions.
- **Potential Harm to Participants**
  Describe the possible harm to the participants during the survey procedure.
- **Reporting of safeguarding concerns or incidents**
  Refers to the potential risks could arise during the survey evaluation.
- **Use of personal data**
  Description of the restriction on personal data to respect participants' human rights.
- **Managing personal data**
  Describes how personal data will be stored, accessed, and deleted after the research.
- **Third-party services**
  Refers to external data involvement.
- **Security of computers, devices and software**
  Description of the requirements for participants to access the survey.

3.4 Conclusion

The previous explanations indicate that large, time-annotated datasets are well-suited for evaluating the performance of various video classification models, particularly those involving temporal segmentation and multi-label classification. These datasets have significantly advanced sports video classification by offering diverse and compressed video clips. This highlights that complex and extensive datasets like UCF-101 enable the development of effective video classification models.

Although SoccerNet was originally essential for developing the summarisation task, certain limitations in its use led to the decision to employ a different dataset for Part B. To maintain methodological consistency between the two tasks, the project utilized the same dataset as in the classification task. Consequently, the summarisation task was developed using a compressed version of the UCF-101 dataset, containing 56 action classes. This approach ensured continuity with the classification methodology and provided a consistent data source for implementing the Part B system described in Chapter 2.

Chapter 4

Design of the project

This chapter outlines the initial and final designs of the video classification and summarisation system, explaining the reasons behind modifications made to the original plan. It also provides detailed descriptions of the model architectures, data preparation, and implementation procedures for both tasks.

4.1 Plans for Summarisation

This section discusses the reasons for altering the initial plan for summarisation and provides a detailed explanation of the modified approach. It also evaluates why the revised summarisation method is considered more suitable for the project.

4.1.1 Plan A : SoccerNet

According to Chapter 2, the HSA-RNN provides shot-level recognition, which accelerates detection by grouping frames. This grouping improves feature detection efficiency by reducing the number of individual frames the model needs to process. Consequently, the initial plan for the summarisation task was to train the HSA-RNN model using the SoccerNet dataset, which contains long videos. However, due to issues accessing the dataset, the initial plan had to be modified.

Data Access Issue

Giancola et al. (2018) implemented a scalable dataset for action spotting; however, access to the dataset requires a password. To obtain the password, researchers must submit a Non-Disclosure Agreement (NDA) form, declaring that the project focuses on research excluding commercial purposes. Although SoccerNet is publicly available for research and non-commercial use, no response has been received from the organization aside from advertising emails.

Additional Dataset Research

Due to access issues, the project required additional research to find an alternative dataset for summarisation purposes. Two datasets were identified during this process: SumMe and TVSum. Although both datasets were developed for video summarisation, they are now deprecated from official sources. Furthermore, these datasets are not specifically designed for sports, which does not align with the project's objectives. Consequently, further investigation was necessary to secure a suitable dataset for the summarisation task.

Copyright Issue

Several sports-related video datasets were identified, such as BBC Sports, Sports-1M, and Olympics. However, all videos in these datasets are copyrighted, limiting access as they are not publicly available. Since the project does not utilize any external information beyond the dataset, it is essential to examine the copyright status of the data in use. Fortunately, UCF-101 is openly accessible for research purposes, eliminating potential copyright concerns. Accordingly, the project finalized UCF-101 as the dataset for the summarisation task and implemented an alternative plan that utilizes the same dataset for classification.

4.1.2 Plan B : Modified Plan

Final Plan for summarisation

The initial plan for Part B focused on implementing HSA-RNN (Hierarchical Structure Adaptive RNN), as proposed by Zhao et al. (2018), as the training model. HSA-RNN processes video inputs using a two-layer hierarchy, detecting crucial frames by distinguishing between local and global timelines. This mechanism enables the model to focus on relevant segments rather than simply following a sequential order, allowing for more precise and momentary summarisation.

Accordingly, the summarisation was designed for long video detection, such as full soccer matches lasting 90 minutes. Consequently, HSA-RNN was considered a suitable model due to its hierarchical structure and processing capabilities.

HSA-RNN essentially requires long-duration videos for effective performance. However, most sports videos are protected by copyrights, and the only dataset identified during research was SoccerNet. Access to SoccerNet is granted for research purposes upon submission of a Non-Disclosure Agreement (NDA) form, after which a password is provided to access the data. Despite multiple NDA submissions, only promotional responses were received, and dataset access was never granted.

The research process continued to find alternative datasets without copyright restrictions, including those beyond football. However, this exploration was unsuccessful, necessitating a change in plan. Accordingly, the revised objectives for Task B were:

- Must analyse all video frames
- Require to perform summarisation

- Must produce a extracted output from long input
- Should include only specified actions in the output
- Users must be able to recognize the extracted actions

Due to copyright limitations, the only available dataset was UCF-101, which was already used in the classification task. However, UCF-101 consists of short video clips, typically 4 to 12 seconds long, making it unsuitable for HSA-RNN, which is designed to handle extended input sequences.

As an alternative, the project decided to select five random action categories from the UCF-101 dataset and merge their clips to create longer videos (approximately 30 seconds), enabling more complex analysis. Nevertheless, this duration is still relatively short for processing with a model like HSA-RNN. Therefore, a more suitable and efficient model was required for this task.

The plan concluded that a CNN-LSTM model would be more appropriate for this task. Since CNN-LSTM effectively captures both temporal patterns and spatial structures of each frame, it was considered a suitable architecture for detecting content with varied patterns and short runtimes Mutegeki and Han (2020).

In conclusion, the initial plan to utilize HSA-RNN for summarizing longer videos was revised. The final approach employs a CNN-LSTM model to summarize merged short clips by extracting only the relevant actions into the output video.

CNN-LSTM Plan Explanation

The modified plan employs a CNN-LSTM architecture, which is better suited for short clips and diverse action patterns. Additionally, the selected dataset is UCF-101. Since the CNN-LSTM model is already trained on UCF-101, this approach creates a connection between Task A and Task B, while eliminating the need for additional training data.

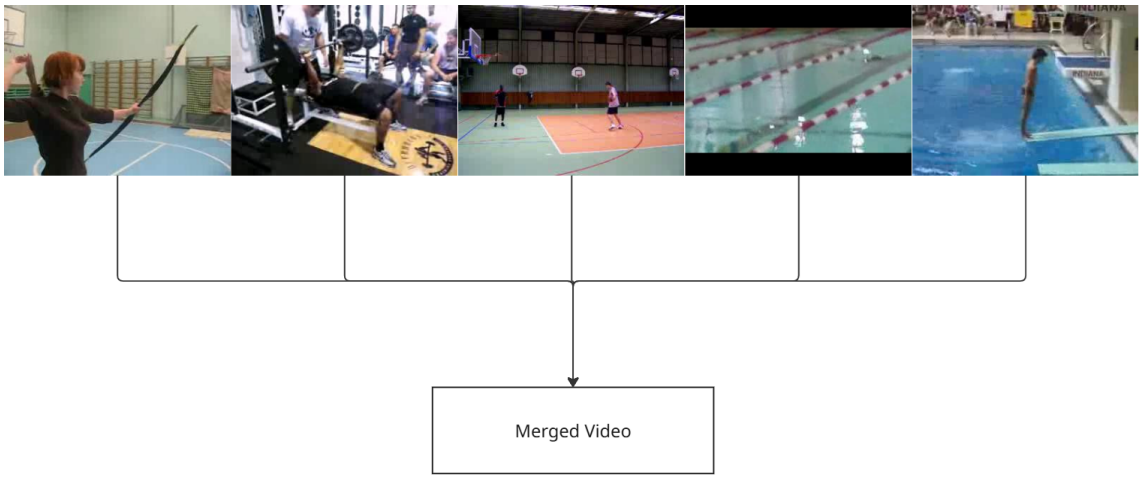


Figure 4.1: Merging five different actions into a single video. The merged video will be used as input for a summarization task.

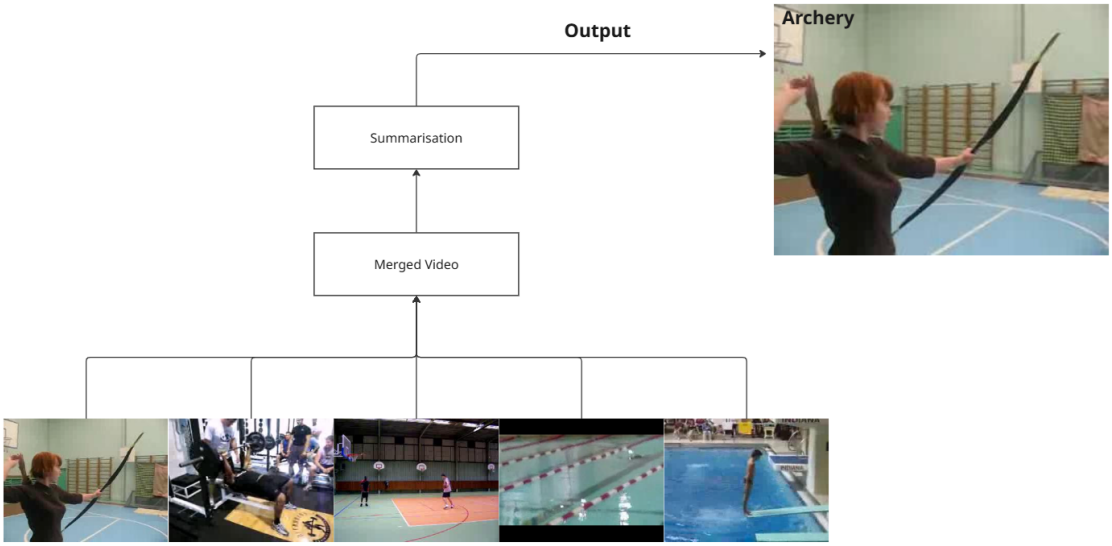


Figure 4.2: A diagram representing the usage of the merged video: The merged video is used as input for the summarization system, which processes it and generates a summarized output.

Figure 4.1 illustrates the components contained in the input video. Five action clips are randomly selected from the dataset and merged into a single video. This setup enables the system to detect a mixture of multiple action patterns and extract a specific action as the output. Moreover, Figure 4.2 depicts the extraction procedure from the large information model.

The fundamental objective of video summarization is to extract the crucial segments

from a long video. However, the term "crucial segments" is somewhat vague and can vary depending on context. For example, in a football match, goal moments are typically considered crucial, but highlight footage often includes other important segments such as fouls, offsides, or injuries. Given this variability, this project defines "crucial segments" as the specific actions selected by the user. This approach provides a consistent and controllable framework, allowing the system to extract only the desired content.

Based on this rationale, the selected plan requires the user to choose specific action patterns to include. As shown in Figure 4.2, the system will then generate an output video featuring recognizable symbols overlaid on the relevant segments.

Lastly, video summarisation generates new outputs that cannot be directly compared with existing test datasets. This highlights the need for an alternative evaluation methodology. Consequently, the project relies on user feedback to assess the quality of both the system and its outputs.

Limitation

Despite the discussions presented in the previous sections, this plan ultimately represents a modification of the initial plan. Consequently, it entails certain limitations that reduce the system's performance and variety, as follows:

- **Variety in dataset:**
  Although UCF-101 provides a variety of videos for each action class, employing one dataset limits the system's ability to adapt to different environments, such as variations in filming angles, resolution, and weather conditions.

- **Limited testing data:**
  Since the project employs one dataset, the system performs optimally within the context of that dataset. However, it may not produce the same results with inputs other than UCF-101.

- **lack of runtime in single video:**
  Since the task first aimed to summarise a long video into shorter output containing crucial segments, if the system uses input approximating 30 seconds and generates a 4-12 second video, it is considerably off from the project's objective.

- **Action label limitation:** Plan B requires the user to select a single class label; however, the system must process and combine all relevant action labels for the chosen sports. For instance, if the user uploads a football video, the system should identify all crucial segments, such as penalties, goals, and other key segments, and compile them into a single output. Although Plan B employs an input composed of a mixture of actions rather than a continuous video, user selection is required to display a single action from the various detected class labels. Therefore, the system functions more as a classification generator, producing a summary of the selected action class from a variety of clips, rather than extracting crucial segments from a single sports video.

Benefits

Although the plan presents several limitations, it also provides benefits. This section outlines the advantages provided by Plan B.

- **Reduced Training Epoch:**
  If the system included additional datasets and models, it would require further training, resulting in an increased number of epochs. Consequently, Plan B reduces the time required for model training.

- **Lower Storage Requirements:**
  No additional datasets are required, as the project utilises a single dataset, UCF-101, which consequently minimises storage requirements.

- **Low Computation:**
  The duration of the merged video is approximately 30 seconds, which is significantly shorter than the 90 minutes provided by SoccerNet. As a result, the project does not require a longer dataset, thereby reducing computation during system operation, such as feature extraction and image recognition.

- **Avoiding Copyright** Although Plan B presents limitations in achieving the objective of the summarisation task entirely, these constraints primarily derive from copyright restrictions associated with most long video datasets. UCF-101 was the only dataset identified that was still existing and open for research purposes, making it the exclusive option to mitigate the risk of copyright infringement.

4.1.3 Final Decision

Considering various problems such as dataset accessibility, copyright restrictions, and computational limitations, the project ultimately adopted Plan B as the final design for the video summarisation. Although this approach involves reduced dataset variety, short input duration, and dependence on user-selected actions, it avoids problems in further implementation.

The UCF-101 dataset will allow the system to avoid copyright issues and enables efficient training through a reduced number of training epochs, lower storage requirements, and manageable computation power. The CNN-LSTM architecture produces an efficient methodology for extracting features from the input clips, which consist of mixed actions.

On the other hand, while Plan B does not fully achieve the initial objectives of the summarisation task, it represents a realistic approach to implementation. This decision addresses both classification and summarisation objectives within an optimised data and model framework. Accordingly, the implementation will contain fundamental features for future project development.

4.2 Design of the Sports Classification and Summarisation

The task trains a deep learning model to utilise a classification application. This section explains the specific methodology and architectural components of Task A.

4.2.1 Video Classification

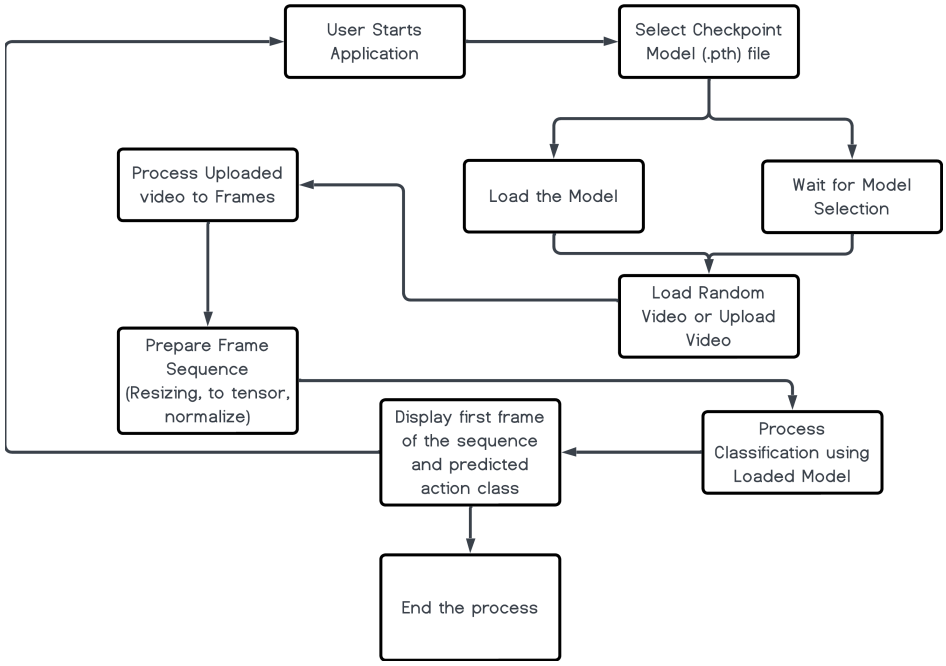


Figure 4.3: A flowchart illustrating the expected user journey in a classification application.

Figure 4.3 illustrates the application structure for the classification task. As described in the functional requirements in Chapter 3, the program requires the user to upload a checkpoint model (.pth) file. After uploading this file, the training model will be loaded into the system and will be available for the classification task. After loading the model, the application requires the user to upload input or click on random input selection.

After receiving the input, the system performs frame-level feature extraction based on the training model and displays the most accurate class label and the first frame of the input. Then, the system remains idle until the user provides further instructions.

Classification App

Figure 4.3 flowchart illustrates the structure presented in the functional requirements described in Chapter 2. Since the objective of the classification task is to develop an interactive application, the project will implement a GUI using the PyQt6 framework. The model will be continuously evaluated and optimised through repeated epochs until the evaluated accuracy is satisfactory. Moreover, the accuracy will be displayed on the application screen.

4.2.2 Video Summarisation

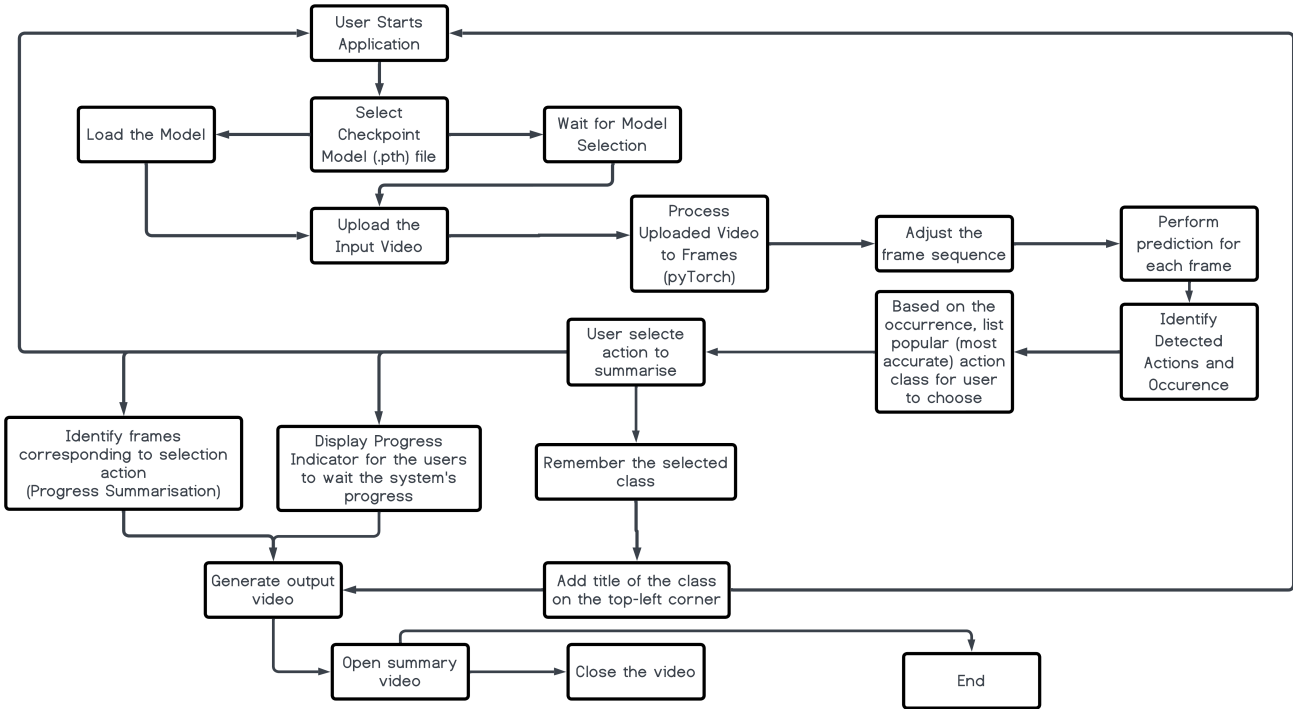


Figure 4.4: A flowchart illustrating the expected user journey in a summarisation application.

Figure 4.4 presents the user interaction flow for the summarisation application. Since Plan B is a modified strategy derived from classification, Figure 4.4 shares similarities with the classification flow. As described in the "Plan B" section, the program requires the user to upload a merged video as input. Using PyTorch, the program performs feature extraction on each frame and predicts the action class.

Based on the detected occurrences of actions, the system generates a list of identified action labels. The list is presented as selectable buttons. When the user selects one of the actions and clicks the "generate summary" button, the system produces the summarised output.

Summarisation App

The flowchart shares a similar mechanism to that presented in the "Classification App". The illustrated flow will be presented using PyQt6 and visually represented to the users.

4.2.3 Optimised UCF-101

To achieve efficient model training, the UCF-101 dataset must be optimised. Currently, UCF-101 includes 101 action classes. However, it includes irrelevant actions, such as applying

eye makeup, playing instruments, typing, etc. Consequently, these irrelevant labels may introduce noise into the classification process and reduce accuracy.

Furthermore, the large and varied action classes can result in increased data storage and inefficiencies in model training due to the size of the classes and frames. Therefore, all non-sports-related action classes will be removed from the dataset to enhance the computation environment.

4.3 Evaluation

As previously mentioned, Tasks A and B will employ different evaluation methodologies. Although the methods are different, each method will be represented using charts, diagrams, or tables to visually illustrate the evaluation results.

4.3.1 Evaluating Classification

The classification task will be evaluated using classification evaluation metrics, Hossin and Sulaiman (2015). These metrics will be used to assess the system's training and validation performance.

In addition, ROC curves and Confusion matrix diagrams will be presented to visually illustrate the evaluation results.

4.3.2 Summarisation Evaluation

Due to the absence of a testing dataset for direct comparison, the results of Part B will be evaluated by user responses. A questionnaire will be provided to participants via Google Forms, and their responses will be used to generate charts and diagrams to discuss the results of the summarisation task.

The result may present ambiguous evaluations; however, by analysing user feedback, the project aims to identify the strengths and weaknesses of the program and suggest potential features to be developed for future advancements in video summarisation.

The evaluation is structured as an online questionnaire requesting participants' answers to the questions. The participants will have five sessions to complete; each session will contain different input and output videos. The questionnaire will contain 7 questions per session and 5 common questions, which means there are 40 questions for the participants to fill in. Each question will be stated across 5 different criteria, representing the relevance and accuracy of action detection, completeness, coherence, usefulness, and feedback. The current structure allows the participants to assess the performance of the system based on these 5 criteria. Moreover, participants will respond to questions using a scale from 1 to 5, where higher numbers indicate greater overall relevance to the question. The details of the provided questionnaire follow:

- **Questions**

  - **Session Questions**
    * Did the output video clearly focus on one specific action (e.g., basketball, soccerPenalty)? (Relevance & Accuracy of Action Detection)
    * Did you find any irrelevant frames included in the output video? (Relevance & Accuracy of Action Detection)
    * Was the content of the output video relevant to the action title displayed in the top-left corner? (Relevance & Accuracy of Action Detection)
    * Was the structure of the output video easy to understand? (Completeness)
    * Were there any abrupt transitions or missing segments in the output video? (Completeness)
    * Was the title of the action easy to understand? (Completeness)
    * Was the title of the action clearly displayed? (Coherence)
  - **Feedback Questions**
    * Would this system be useful for compressing future media datasets? (Usefulness)
    * Would this system be useful for summarising longer videos? (Usefulness)
    * How confident are you in the system's ability to automatically identify and extract this type of action? (Coherence)
    * How accurately do you think the system selected relevant scenes? (Feedback)
    * Do you have any additional comments or suggestions about this system? (Feedback)

- **Definitions of Each Criterion**
  - **Relevance & Accuracy of Action Detection**: Evaluates how well the summarised video captures the intended action frames from the original video, including:
    * Whether the summary focuses on a single, significant action.
    * Whether irrelevant frames are included.
    * Whether the title of the action class matches the summarised content.
  - **Completeness**: Measures the structure and clarity of the summary, including:
    * Whether the video is easy to follow.
    * Whether there are abrupt cuts or missing segments.
    * Whether the class title is clearly displayed and understandable.
  - **Coherence**: Assesses alignment between the displayed title and video content, including:
    * Whether the title is visually clear and not distorted.
    * Whether the system consistently aligns detected actions with the class title.
  - **Usefulness**: Evaluates the practical value of the system, including:
    * Its potential for compressing sports media datasets.

    - Its usefulness in summarising longer sports videos.
  - **Feedback**: Collects participants' overall impressions and suggestions, including:
    - Perceived system performance.
    - Any improvement suggestions or concerns.

- **Marking Scheme (Criteria)**
  - **5 – Excellent:** The system is well-constructed and clearly understood by users based on the session questions.
  - **4 – Great:** The system is well-designed but may require additional features to enhance user clarity.
  - **3 – Moderate:** The system performs adequately but needs improvement in clarity and usability.
  - **2 – Poor:** The system causes confusion for users in relation to the evaluation questions.
  - **1 – Bad:** The system fails to communicate effectively, leading to significant user misunderstanding.

Chapter 5

Implementation

This chapter will provide a detailed explanation of the methodologies, configuration, and equations used to train the CNN-LSTM model. Since Chapter 4 illustrates the design of the project, a detailed description of the actual progress is absent. Therefore, this chapter will outline the implementation process and the limitations encountered during the development of the classification and summarisation applications, based on the structure presented in the previous chapter. Moreover, it will cover additional components, such as user input, error handling, and a progress indicator, to discuss application completion and enhance user satisfaction.

5.1 Approach in Achieving Fundamental Objectives

Before illustrating application implementation, this section introduces the preparatory procedure required to establish the fundamental components.

5.1.1 Dataset Preparation

As Chapter 2 mentioned, the system requires optimising the dataset to enhance model training computation. This section outlines the approaches taken in dataset extraction and management.

Dataset extraction

The project required extracting all clips of UCF-101 into frames for efficient pattern computation. Accordingly, the project extracted the frames of each clip and stored them in a folder titled after the footage.

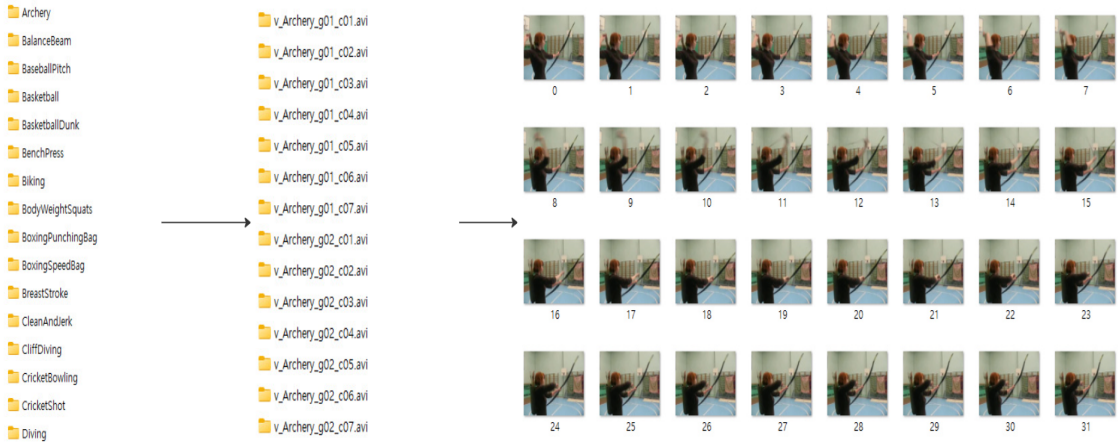


Figure 5.1: A diagram representing the data structure. Extracted frames from each video are stored in a folder named after the video. The videos belonging to each action class are stored in a folder labeled with the corresponding action.

Figure 5.1 represents the structure of the extracted frames. It shows that each action includes a folder for each video, and the extracted frames are stored in each folder corresponding to their clips.

The CNN extracts frames of the video to extract features in a loop based on the number of epochs presented in the configuration. On the other hand, if the prepared dataset is already arranged into frames, the model just needs to load the dataset and proceed with computation. Consequently, the dataset is prepared into frames before the training section.

Dataset management, UCF-101

Since the project is designed to optimise UCF-101 by reducing irrelevant action classes, this section describes the approaches to managing UCF-101.

After loading the dataset, it was observed that it contained actions irrelevant to sports. Accordingly, the project decided to delete the unnecessary classes manually.

BlowDryHair	FieldHockeyPenalty	HeadMassage	Skijet	
BlowingCandles	FrisbeeCatch	HighJump	SumoWrestling	
Bowling	GolfSwing	HorseRace	TaiChi	
BrushingTeeth	Haircut	HorseRiding	Typing	
CuttingInKitchen	Hammering	HulaHoop	UnevenBars	PoleVault
ApplyEyeMakeup	JugglingBalls	PizzaTossing	PlayingGuitar	PommelHorse
ApplyLipstick	Lunges	PlayingCello	PlayingPiano	Rafting
BabyCrawling	MilitaryParade	PlayingDaf	PlayingSitar	SalsaSpin
BandMarching	Mixing	PlayingDhol	PlayingTabla	SkateBoarding
Billiards	MoppingFloor	PlayingFlute	PlayingViolin	

Figure 5.2: List of Manually Removed Classes: A total of 45 classes, identified as irrelevant to sports, were deleted from the dataset.

Figure 5.2 shows the list of deleted classes from UCF-101, which includes 45 action classes. Most of these are irrelevant to sports, such as ApplyEyeMakeup, ApplyLipstick, etc. Although a few labels such as Bowling, Golf swing, etc., were included, these were excluded due to potential confusion caused by dominant dataset patterns and colour (green for grass, brown board). Accordingly, they were excluded to improve computation performance and model accuracy.

After reducing 45 classes, the dataset now contains 56 action labels. The storage size for the 56 action videos was reduced by 3 GB (from 7 GB to 4 GB), and the corresponding frames were reduced by 16 GB (from 31 GB to 15 GB). Moreover, the number of frame images was reduced from approximately 2,500,000 to 1,316,800. Accordingly, this reduction significantly decreased the number of image files required for extraction, training, and computation by half.

Dataset for Video Summarisation

The input for summarisation is a merged video with 5 videos from random action classes. Accordingly, the project implemented a script that selects five random actions and videos and generates a merged video.

```
selected_actions = random.sample(available_folders, 5)
selected_actions = [action for action in selected_actions
    if action in available_folders]
video_clips = []

# Pick one random video from selected action folder
for action in selected_actions:
    action_path = os.path.join(base_path, action)
    video_files = [f for f in os.listdir(action_path)
```

```
            if f.endswith(('.avi', '.mp4', '.mov'))]

        selected_video = random.choice(video_files)
        video_path = os.path.join(action_path, selected_video)
        print(f"Selected video from {action}: {video_path}")
        video_clips.append(video_path)

# Set frame size as the first video
first_video = cv2.VideoCapture(video_clips[0])
width = int(first_video.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(first_video.get(cv2.CAP_PROP_FRAME_HEIGHT))
first_video.release()
```

The following code selects five random videos from five random action classes. After the selection, the script generates a sequentially merged video, which will be uploaded for the summarisation process.

5.2 Flowchart Illustrating the Configuration for Classification and Summarisation

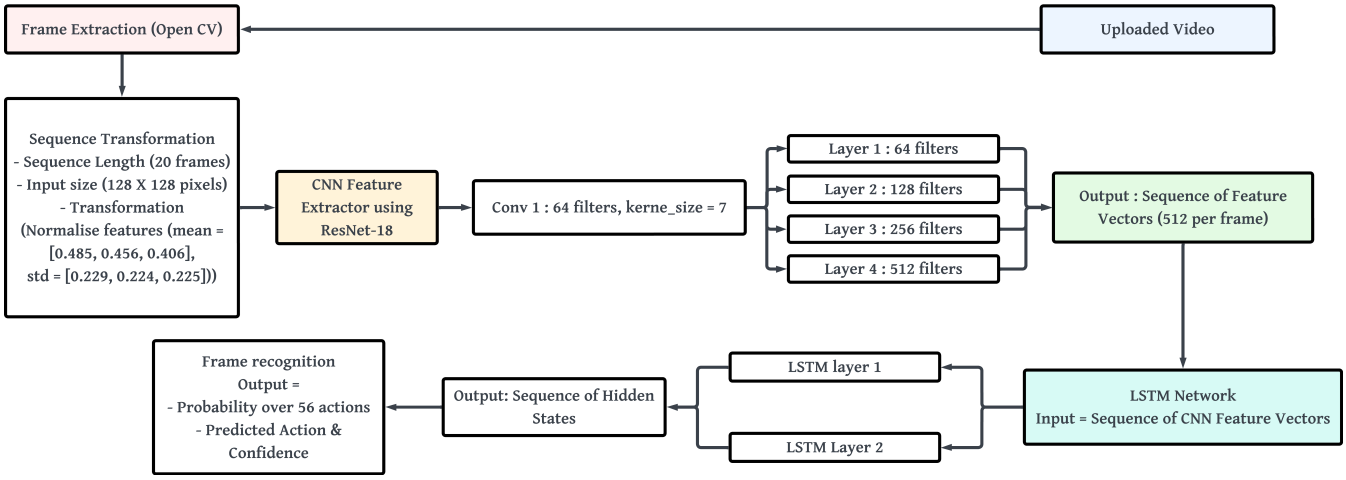


Figure 5.3: The flowchart illustrates the functionality of the system, highlighting both the model configuration steps for classification and summarisation.

According to Figure 5.3, the project employs a CNN with 4 layers where the number of filters increases up to 512. After the CNN process, the output is uploaded into two LSTM layers, which generate a sequence of hidden states. The **hidden state** is a vector that stores information about the temporal sequence the model processed. Furthermore, the model collects these results to predict the probability distribution over 56 action classes for each frame and outputs the predicted action based on the trained CNN-LSTM architecture.

As Chapter 3 mentioned, summarisation is an advanced version of classification that

modifies the input requirement, listing process, and output generation. Accordingly, the functionality illustrated in Figure 5.3 is shared for both classification and summarisation.

5.3 Sports Video Classification

The project trains a deep learning model to be utilised in the classification application. This section explains the specific methodology and architectural components employed in implementing Task A system.

- **Extracting frames**: Figure 5.3 represents how each video from the dataset is extracted into frames.
- **Data splitting**: The extracted frames are provided as inputs to the model. The model splits the frame files into training and testing sets based on the train and test list text files. The system consistently processes to account for variations in video resolution and frame rate during frame extraction for all videos.
- **Label mapping**: The system loads mapping information, which contains title of the action classes and their corresponding numeric labels.
- **Sequence generation and padding**: Configuration file involves in generating fixed length frame as inputs. The fixed sequence length was 40 frames, which was ensured to be consistent input size for the LSTM model. The shorter videos than the required length repeatedly padded the last frame.
- **Data transformation and normalisation**: The torchvision.transforms is used to resize, 224 X 224 pixels, the each image to match the model's dimensions and converted into a tensor format. Additionally, each image tensor(multi-dimensional array) is normalised to optimise the performance of the CNN, using ImageNet dataset's mean ([0.485, 0.456, 0.406]) and standard deviation ([0.229, 0.224, 0.225]).

5.3.1 Model architecture

As Section 3.2.1 represents, the CNN-LSTM is the model employed to effectively learn both spatial and temporal features for this project. The architecture of the model is defined by the following list:

- **CNN**: The convolution layers of an imported ResNet-18 model are used to extract spatial features from each video frame. The preceding feature map is taken as output after removing the fully connected layer of ResNet-18.
- **LSTM**: The sequence of feature vectors extracted by the CNN is sent to the LSTM to model temporal dependencies. The system used three LSTM layers with a hidden dimension of 1024 and bi-directionality.

- **Classification Layer**: The output is then passed through a classification layer to predict the final action class. A total of 56 classes are defined, and the system configuration contains class labels for recognition and reference.

5.3.2 Training procedure

The system uses the following procedure to train the CNN-LSTM model:

- **Parameter Configuration**: Essential training parameters such as the learning rate, batch size, number of epochs, etc., are stated in the configuration to optimise the model's performance.
- **Loss Function and Optimiser**: Cross-entropy loss is employed as the loss function for each iteration of the model training. The resulting loss function demonstrates the current performance of the system.
- **Training Loop**: The project completed 105 iterations of the total training session, achieving reasonable results in accuracy and loss function, which took 50 hours of runtime.
- **Model evaluation and check-pointing**: The system performs model evaluation at each batch iteration and computes the overall evaluation results using the test dataset. Checkpoints are generated at the end of every epoch. Moreover, the program also supports resuming training from a selected checkpoint.
- **Logging Display**: TensorBoard is used to provide a visual representation of training progress, including the loss function, accuracy, and ETA (estimated time of completion), to track progress throughout the training process.

5.3.3 Classification application

PyQt6 was employed to develop the GUI for action classification. The implemented application enables users to upload and classify their desired inputs or click on the 'load random video' option to classify random footage. This interface enhances the user experience by providing an interactive application.

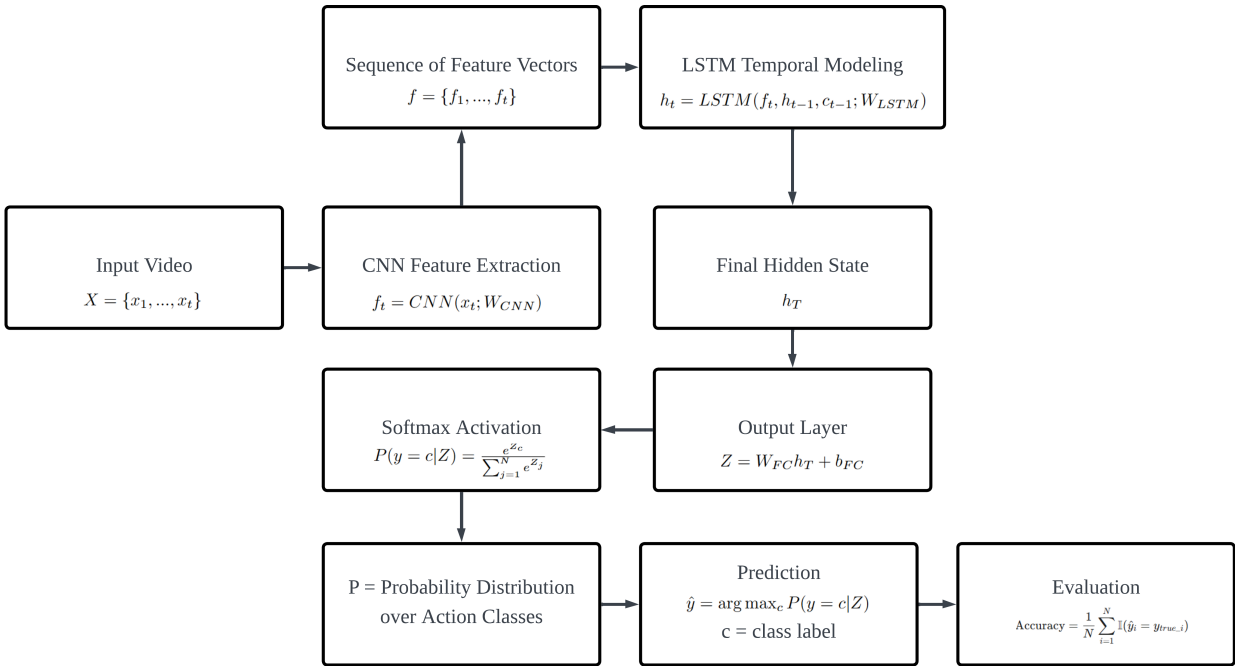


Figure 5.4: A flowchart illustrating the system's functionality with equation integration at each stage of the classification process.

- **Upload model checkpoint**: The user is required to upload a checkpoint file (.pth) to load the trained model.
- **Upload video**: Users can provide a local video file or choose the random video option to classify the input video. OpenCV is used to convert the uploaded video into a sequence of frames. Each extracted frame is resized to the same dimensions (128x128 pixels) and undergoes the same normalisation process. OpenCV is also used for resizing to minimise quality loss.
- **Frame screening**: The first frame of the input video is displayed on the application screen to help users recognise two features: the frame and the classified action. This feature is essential for verifying the accuracy of the classified action and its correspondence with the uploaded video.

The **runtime** of the implemented system was less than 0.5 seconds, which demonstrates the efficiency of the system.

Equation Explanation: Classification

Figure 5.4 following equations:

- **Input Frame Sequence**
  $X = \{x_1, ..., x_t\}$

  - X represents the full input sequence consist of individual inputs x_1 to x_t.

- **CNN Feature Extraction**
  $f_t = CNN(x_t; W_{CNN})$
  - Each input frame x_t is passed through a CNN to extract a feature vector f_t. CNN uses learned weights W_{CNN} to find useful patterns, such as edges, objects, etc.

- **Output: Sequence of Feature Vectors**
  $f = \{f_1, ..., f_t\}$
  - After feature extraction, f is the sequence of all feature vectors generated as an output.

- **LSTM Temporal Modeling**
  $h_t = LSTM(f_t, h_{t-1}, c_{t-1}; W_{LSTM})$
  - LSTM collects parameters f_t as current feature vector, h_{t-1} as previous hidden state, c_{t-1} as cell state and compute h_t. This represents input changes over time.

- **Hidden State (Vector Containing Temporal Sequence)**
  h_T
  - Final hidden state after the LSTM completed the entire sequence.

- **Output layer (Linear Transformation)**
  $Z = W_{FC} h_T + b_{FC}$
  - Combine hidden state with weight and bias, pass it to the fully connected layer, to produce logits Z.

- **Logit(Raw score)**
  Z
  - A vector of raw score of each class before converting to probabilities.

- **Softmax Activation**
  $P(y = c|Z) = \frac{e^{Z_c}}{\sum_{j=1}^{N} e^{Z_j}}$
  - Softmax converts logit Z into probabilities.

- **Probability**
  P
  - Probabilities for each class, calculated from the softmax.

- **Prediction**
  $\hat{y} = \arg\max_c P(y = c|Z)$
  - $\hat{y}$ is the final predicted class

5.4 Sports Video Summarisation

PyQt6 is also employed in this task to provide a GUI for video summarisation. The following list demonstrates the features and functionality included in the summarisation application.

5.4.1 Summarisation Application

The implemented system is an action recognition-based summarisation approach that generates an output video based on a specific action selected by the user.

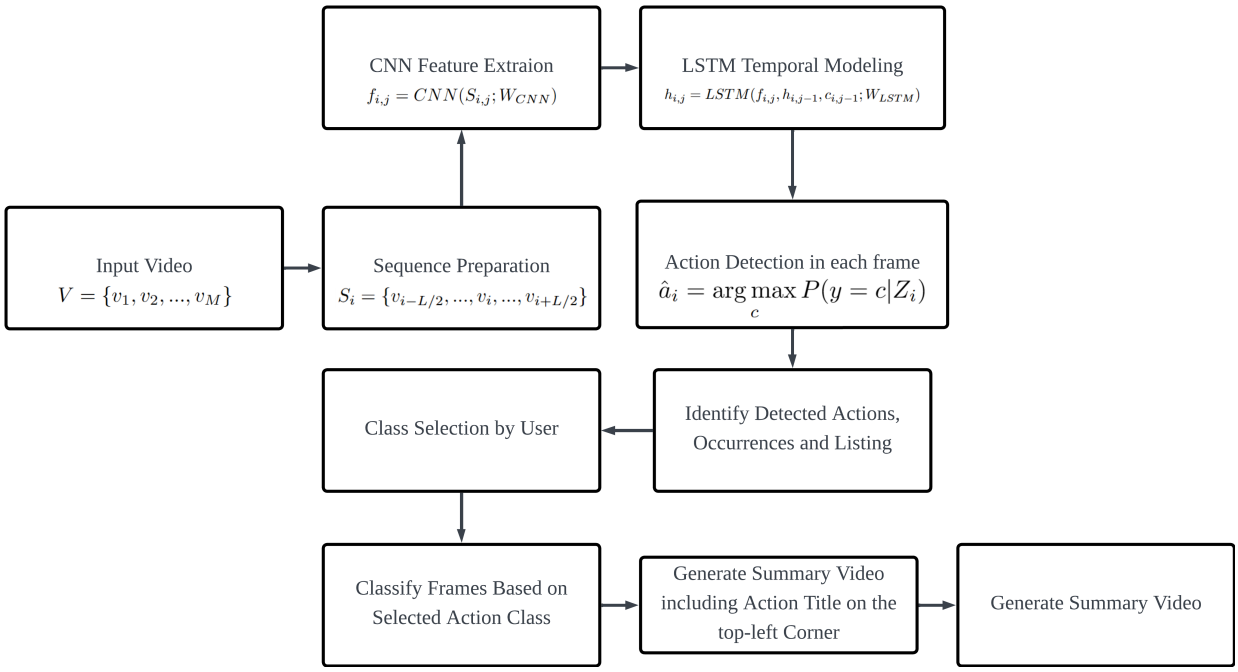


Figure 5.5: A flowchart illustrating user flow and equation integration in the summarisation process. The diagram does not include as many equations as the classification process due to procedural similarities.

- **Upload video**: After a user uploads a video, OpenCV is used to extract the video into a sequence of frames. The system allows differing original frame rates during frame extraction. The resolution of extracted frames is adjustable or can maintain the original video quality.
- **Action Recognition per Frame**: Each frame of the uploaded video is passed through the trained CNN-LSTM to predict the corresponding action class. Frames are resized and normalised to match the input format used during training (128x128 pixels). The model generates a probability distribution over all classes, and the five classes with the highest probabilities are displayed in a list.
- **Detected action listing and selection**: A list of the five actions with the highest probabilities is provided to the user, who may select one option from the list.

- **Summary Selection**: Based on the user's selection, the system identifies frames corresponding to the selected class. To enhance the summary, the system excludes the first and last 10 frames, allowing the system to focus on the essential part of the action.
- **Summary Video Generation**: The selected frames are compiled into a sequence to generate a summarised video. OpenCV is used to save the summary with a specified codec, frame rate, and resolution, as defined in the configuration. PIL is used to overlay the action class title on the top-left corner of the video to help the user verify whether the summarisation has succeeded.
- **Generation check**: The generated output can be immediately viewed using the system's default video player.

Equation Explanation: Summarisation

- **Input Video Frame Sequence**
  $V = \{v_1, v_2, ..., v_M\}$
  - V is the full video, represented as a sequence of M frames.
  - v_i demonstrates single frame from the video.
- **Sequence Preparation**
  $S_i = \{v_{i-L/2}, ..., v_i, ..., v_{i+L/2}\}$
  - S_i represents video clip centered around frame v_i with length L.
- **CNN Feature Extraction (for each sequence S_i)**
  $f_{i,j} = CNN(S_{i,j}; W_{CNN})$
  - For every frame S_{ij} int the sequence S_i, a CNN is applied to extract a feature vector f_{ij}.
- **LSTM Temporal Modeling (for each sequence of features)**
  $h_{i,j} = LSTM(f_{i,j}, h_{i,j-1}, c_{i,j-1}; W_{LSTM})$
  - LSTM processes the feature sequence $f_{i,1}, ..., f_{i,L}$ one by one.
  - This captures change in appearance across time in the clip S_i
- **Frame-Level Action Prediction (for each window centered around a frame)**
  $\hat{a}_i = \arg\max_c P(y = c|Z_i)$
  - Model makes a prediction of the action **at that specific frame**.
  - Z_i provides logit output for frame i, and softmax generate the probabilities P.
  - $\hat{a_i}$ is the predicted action at frame v_i

5.5 Additional Features

- **Progress indicator**: A progress bar and progress percentage, are the features added for users to recognise if the system is still computing.
- **Error alerts**: Error handlers are implemented to support users to recognise any errors.

The estimated **runtime** of the this application was 5 minutes, including video generation.

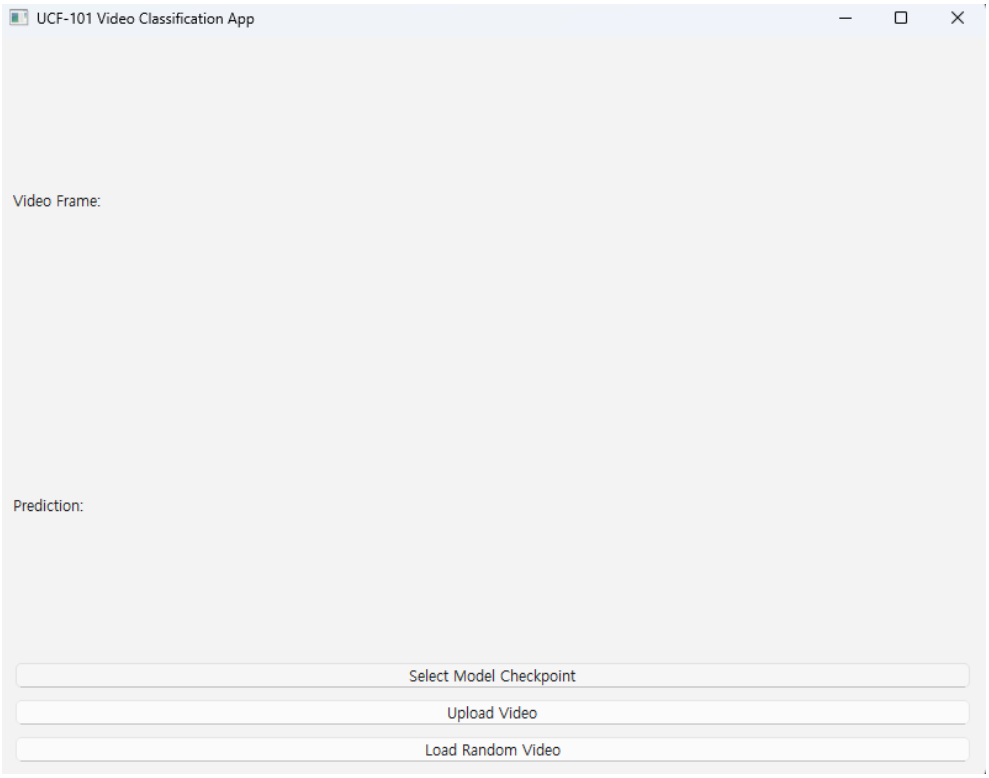
5.6 Applications Representation

The project implemented applications to demonstrate the systems to users with a safe and clear representation, generating results using the same device that was used to train the model. Furthermore, storing a large amount of data in an online system could overwhelm the system's functionality and cause runtime delays. Accordingly, an app was the initial model to develop. To ensure a clear and safer demonstration, the project initially developed two systems using PyQt6, providing a visual interface. Moreover, the project aims to evolve into a web or mobile application to provide users with more portable and accessible approaches.

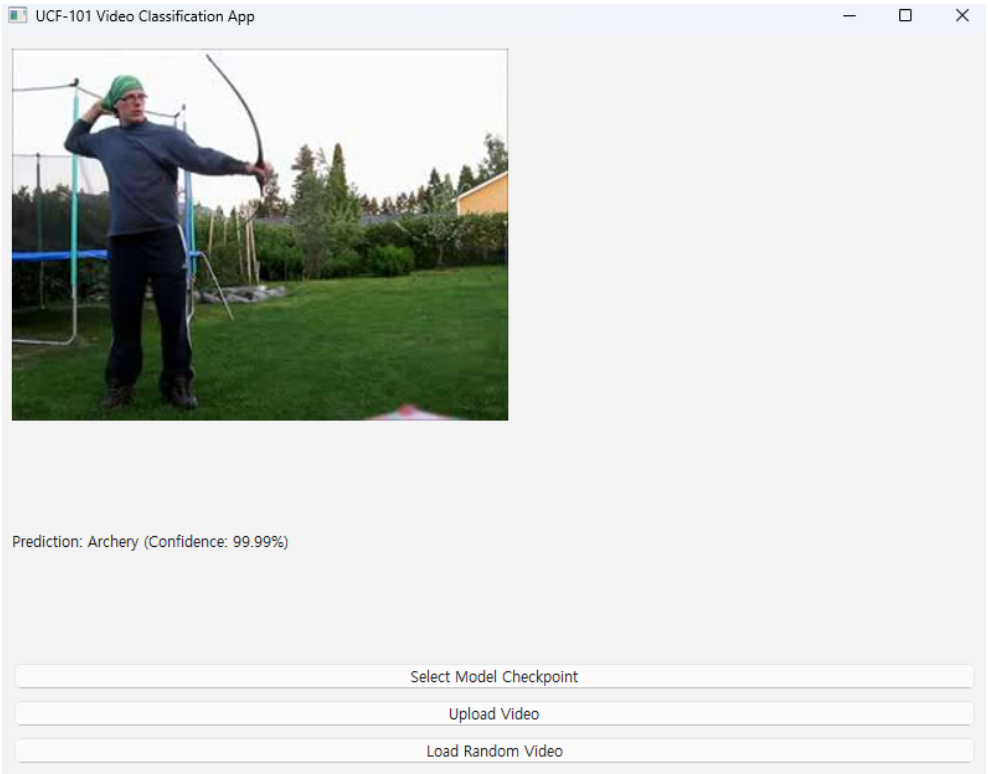
5.7 Application Demonstration

This section demonstrates the implemented applications for classification and summarisation by illustrating application images and results.

5.7.1 Part A: Classification



(a) Implemented application: The program presents the ideal stage when the application is started.



(b) Implemented application: The program demonstrates the result after uploading a video for classification.

Figure 5.6 represents the implemented application for the classification task. The image on the right shows an example of the system in execution. Section 4.3.3 illustrated that the classification app includes an image field, a prediction field, and three buttons for interaction. The right image of Figure 5.6 (a) shows that the system successfully predicted the action class of the uploaded video as "archery" with the corresponding confidence score. This graphical user interface facilitates intuitive recognition of both the system's functionality and its output.

5.7.2 Part B: Summarisation

As Section 5.5.1 presented, providing visual tools is essential for user interaction. Task B implemented an application performing summarisation. Section 4.4.2 explained that a progress indicator is required as an additional feature due to the longer processing time compared to classification. The system informs users that it is actively running by displaying a progress indicator.

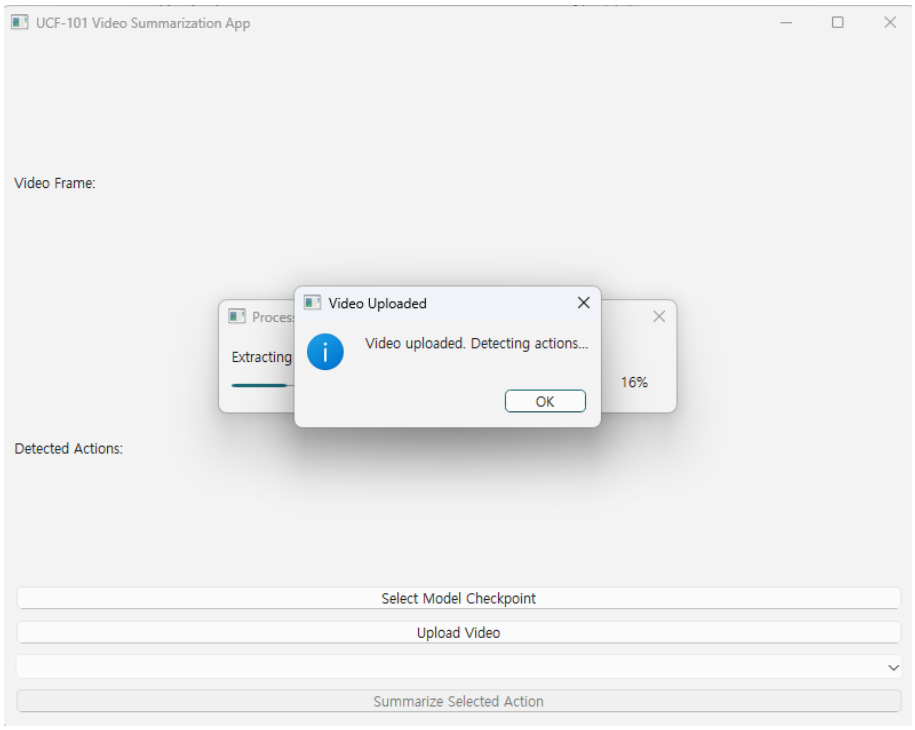


Figure 5.7: Summarisation app: Generated alert after uploading an input video. The message indicates that the program is extracting frames to detect the actions in the input.

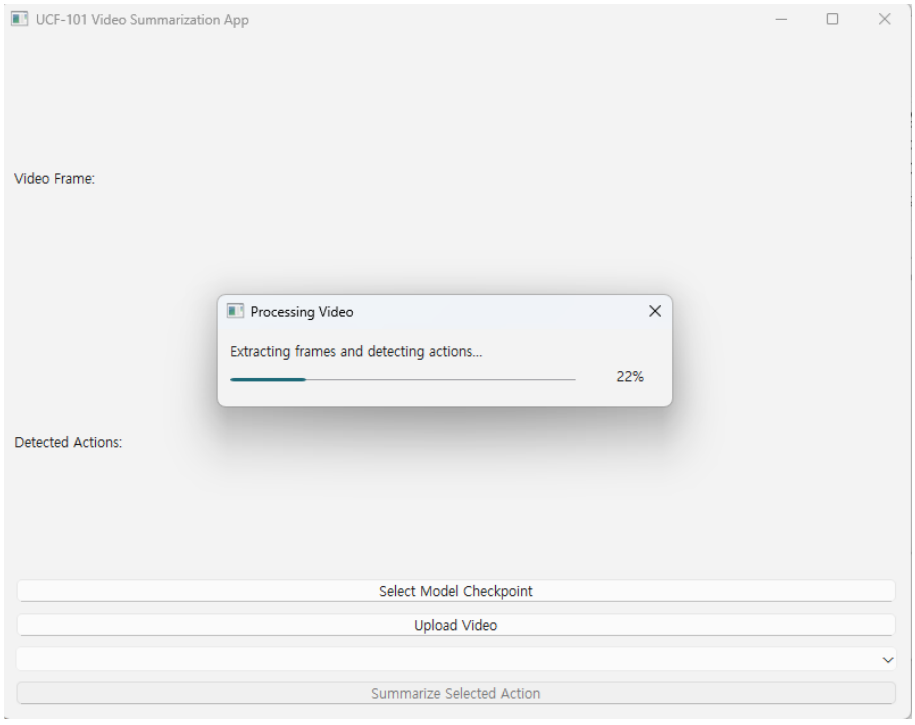


Figure 5.8: Summarisation app: Progress indication illustrating feature extraction status, including a progress bar and percentage representing the system's progress.

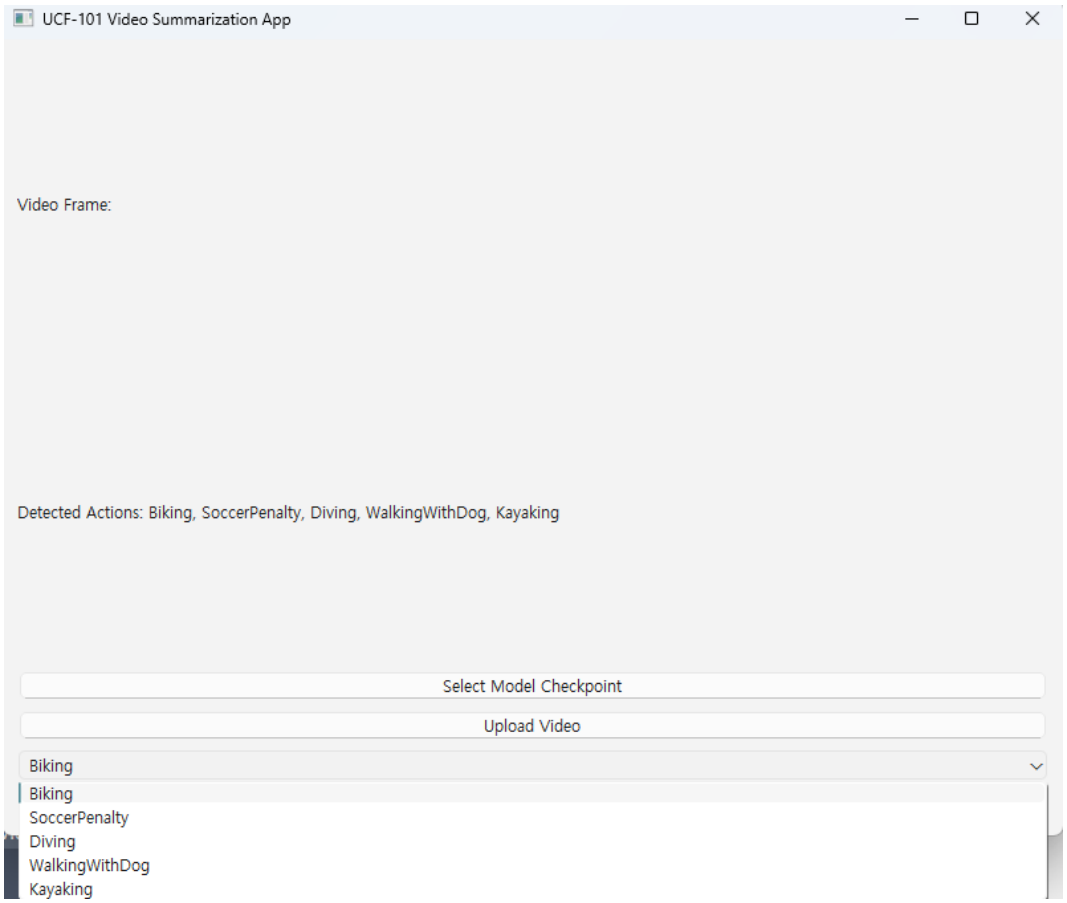


Figure 5.9: Summarisation app: List of detected action classes after video recognition based on the trained model. The program is waiting for the user to select one of the actions.

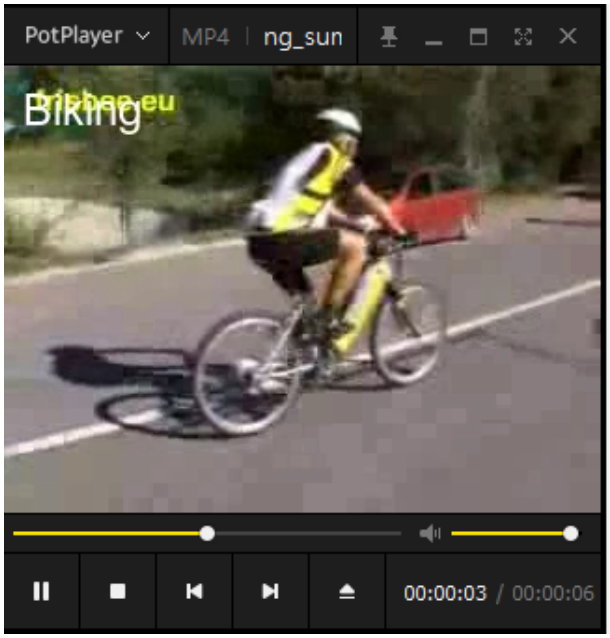


Figure 5.10: Summarisation result: Generated output video example, including the selected action segment with the class label displayed in the top-left corner.

Figure 5.8 and Figure 5.10 demonstrate the interaction of the developed application. Figure 5.8 (b) illustrates the progress indicator displayed after the input video is uploaded. After the extraction, the system provides a list of detected action classes for the users to select. Based on the selected class, the system generates a summary of the input video, as represented in Figure 5.10 (b). Figure 5.10 (b) demonstrates the structure of the output video generated by the summarisation system. The video consists of frames corresponding to the selected action, with the class title displayed in the top-left corner. Consequently, this outcome indicates that the system effectively achieves the objectives and produces the desired output defined in Chapter 2, the Literature Survey.

5.8 Video Summarisation

Task B implemented a system to generate a summary of the input video. However, there is no readily available test data that directly corresponds to the system's output. Accordingly, the task required a different evaluation method. Since the system generates new data distinct from the existing dataset, the application necessitates a methodology that doesn't rely on a pre-existing dataset. Among various alternative measures, the project decided to conduct a questionnaire to collect user evaluations. However, due to human engagement in the project, it was necessary to submit a risk assessment. The ethics review for the human evaluation was approved to conduct an online questionnaire with more than 10 participants; Appendix B includes a screenshot as evidence of this approval. Consequently, the evaluation was conducted immediately with 10 participants.

Chapter 6

Results and Discussion

Even if a successful methodology or application was developed, unsuccessful model training results can lead to increased inaccuracy in the project's recognition capabilities. This chapter will provide a detailed explanation of the outcomes of model training, focusing on the methodologies, models, and datasets used. Accordingly, it will present the recorded metrics, such as accuracy and loss functions, resulting from the training process. In addition, it will derive the equations presented in Chapter 2 to represent the results of the classification task. Since the evaluation of the summarisation task employs a different evaluation methodology, the testing results for this task will not be determined. However, this chapter will rely on human evaluation results to assess the success of that task.

6.1 Model training results

The components used to generate the testing results are the CNN-LSTM model and the optimised UCF-101 dataset. Since both the classification and summarisation tasks utilise the same model and dataset, they share the same loss function and accuracy for training and validation results.

6.1.1 Loss function and Accuracy

Measures	value
Training loss function	0.002
Training accuracy	99.44%
Training f1-measure	0.9904
Training Precision	0.9906
Training Recall	0.9905

Table 6.1: Evaluated training results of the proposed CNN-LSTM model on the optimized UCF-101 dataset at the 140^{th} epoch.

Table 6.1 outlines the training results obtained after the 140th epoch. The loss function converged to 0.002, indicating a low prediction error and a strong fit to the training data. Furthermore, the model achieved a training accuracy of 99.44%, demonstrating outstanding classification performance. In addition, all f1-measure, precision, and recall exceeded 0.99, illustrating that the model is highly effective at correctly identifying positive instances while minimising both false positives and false negatives.

6.1.2 Evaluation Matrix for Both Training and Validation

Measures	Value
Training Loss	0.0606
Validation Loss	0.0520
Training Accuracy	98.65
Validation Accuracy	98.97
Training F1-score	0.9866
Validation F1-score	0.9898
Training Precision	0.9874
Validation Precision	0.9907
Training Recall	0.9865
Validation Recall	0.9897

Table 6.2: Comparison of classification evaluation metrics for the training and validation sets over 140 epochs.

Since training results indicate the performance of the model on the training data, validation results are evaluated to estimate its generalisation ability and its expected performance in real-world applications. Table 6.2 outlines the training and validation metrics after evaluation. Although the accuracy exceeds 95%, the results are slightly lower than those in Table 6.1. This is because Table 6.2 reflects averaged results over all 140 training epochs. Consequently, while the f1-score, precision, and recall are slightly below 0.99, the model still achieved strong performance, with accuracy exceeding 98% and loss values approaching 0.06 for both training and validation.

6.1.3 Diagrams Representing Evaluated Accuracy

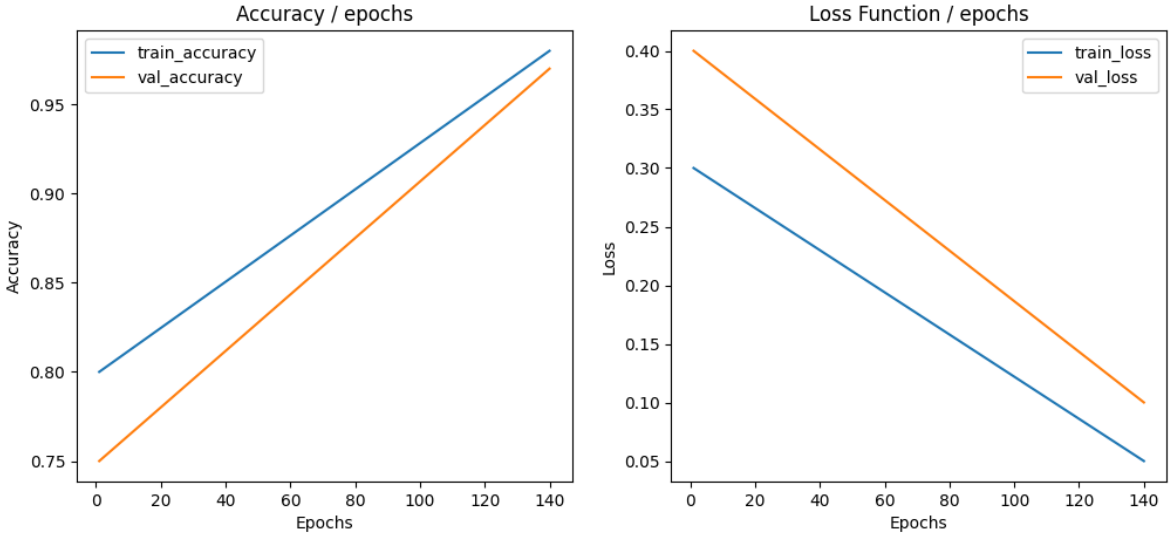
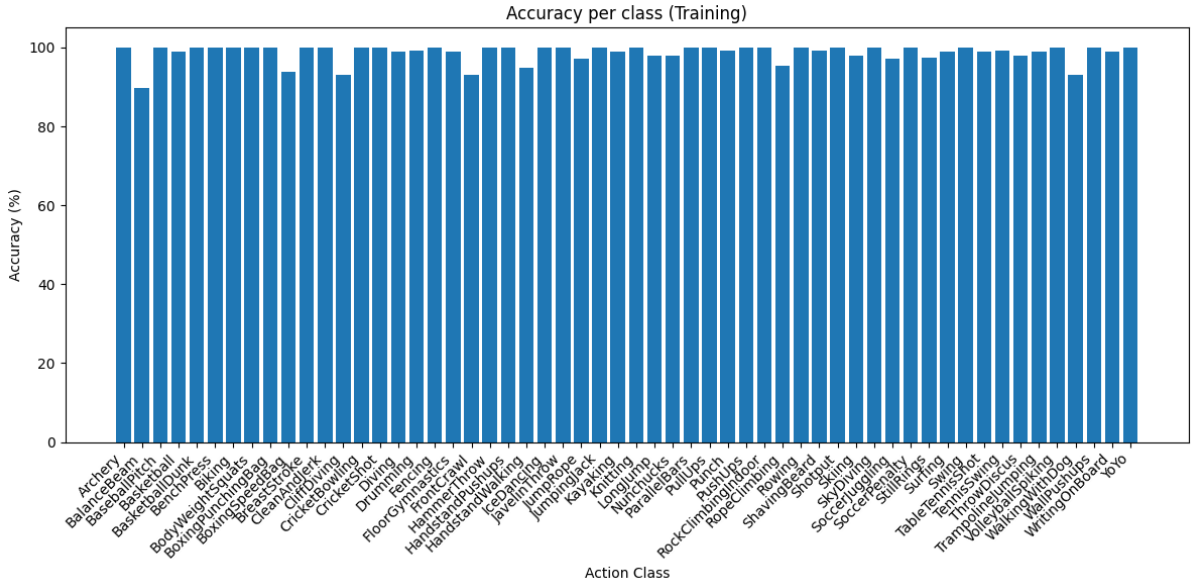
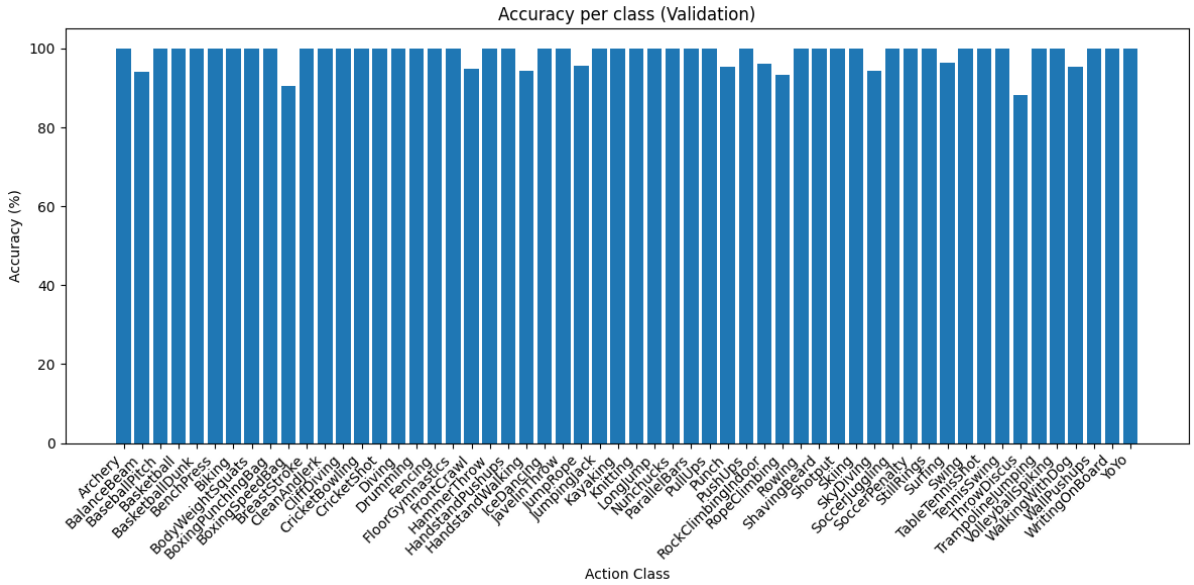


Figure 6.1: A diagram representing the results of accuracy and loss functions as the number of epochs increases.

Figure 6.1 provides evidence of the model's training progress over 140 epochs. It illustrates a generally increasing trend, showing that accuracy linearly approaches 90%, while the loss for validation decreases toward zero. The validation accuracy converges closely with the training accuracy at the final epoch. However, the validation loss reaches 0.06 in Table 6.2 and 0.1 in the diagram, which is noticeably higher compared to the training outcome.



(a) A diagram representing the accuracy for the training set (left) for each class (bottom).



(b) A diagram representing the accuracy for the validation set (left) for each class (bottom).

Figure 6.2: A bar chart representing accuracy per class for both training and validation, which provides a clear understanding of the confusion matrix.

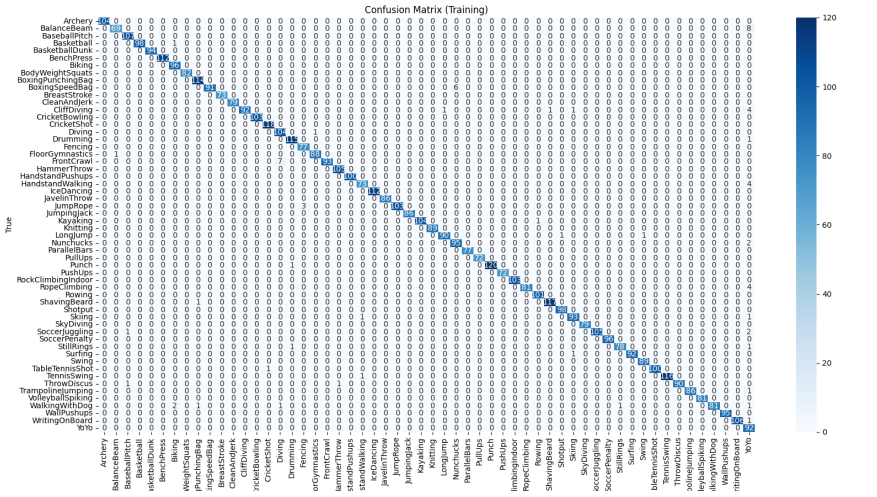
Despite the overall results represented in Figure 6.1, the majority of classes achieved accuracy above 90%. However, a few classes, such as BalanceBeam and ThrowDiscus, recorded lower performance. This observation may be attributed to the limitations of the dataset quality. Since the dataset was originally designed for pattern recognition tasks, the

video resolution is relatively low and lacks clarity, providing noisy data that may lead to inaccurate classifications.

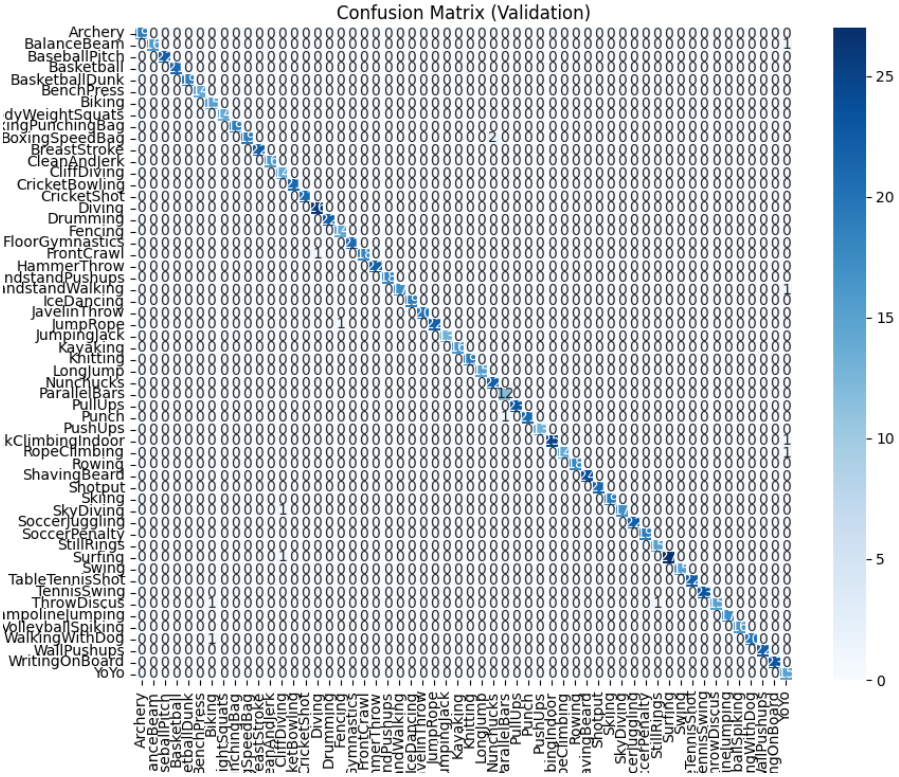
On the other hand, the validation set includes more classes that achieved 100% accuracy than the training set, as illustrated in the per-class accuracy diagram 6.2. This can be attributed to the presence of simpler samples in the validation set and differences in pre-processing. As a result, the validation performance appears stronger across several classes.

6.1.4 Visual Diagrams

Confusion Matrix



(a) Confusion matrix comparing training action classes to demonstrate linear correction



(b) Confusion matrix comparing validate action classes to demonstrate linear correction

Figure 6.3: A confusion matrix for both the training and validation sets, showing a clear diagonal line that indicates correct predictions.

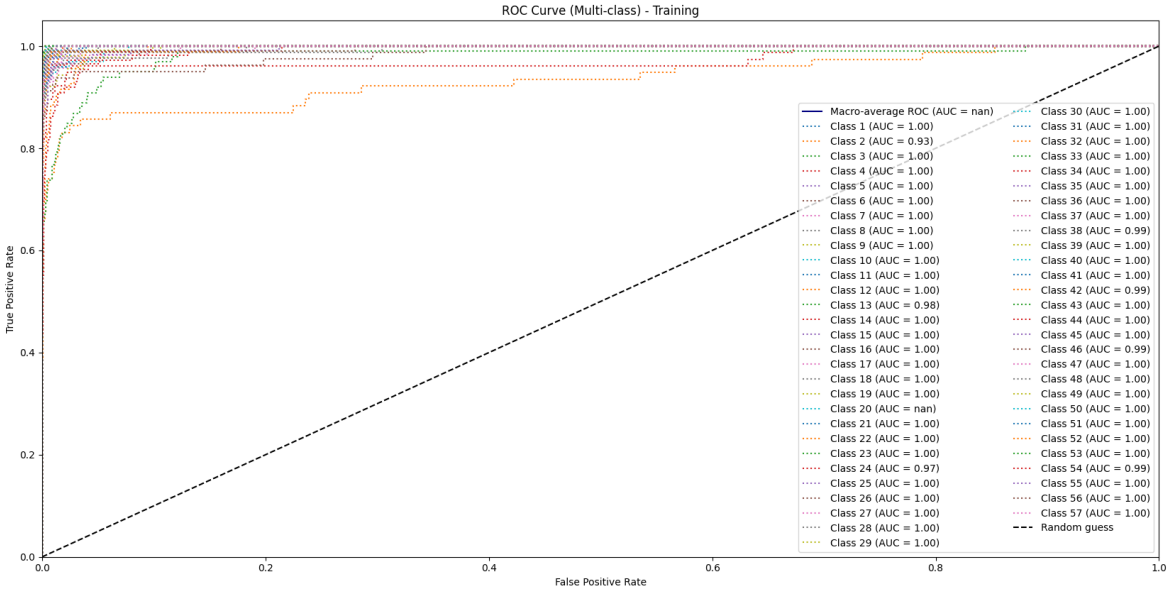
According to Wu (2022), the structure of matrices from Figure 6.3 includes information as follows:

- **Row** = True labels (Actual)
- **Column** = Predicted labels
- **Diagonal cells** = Correct predictions
- **Apart from the diagonal cells** = Incorrect classification

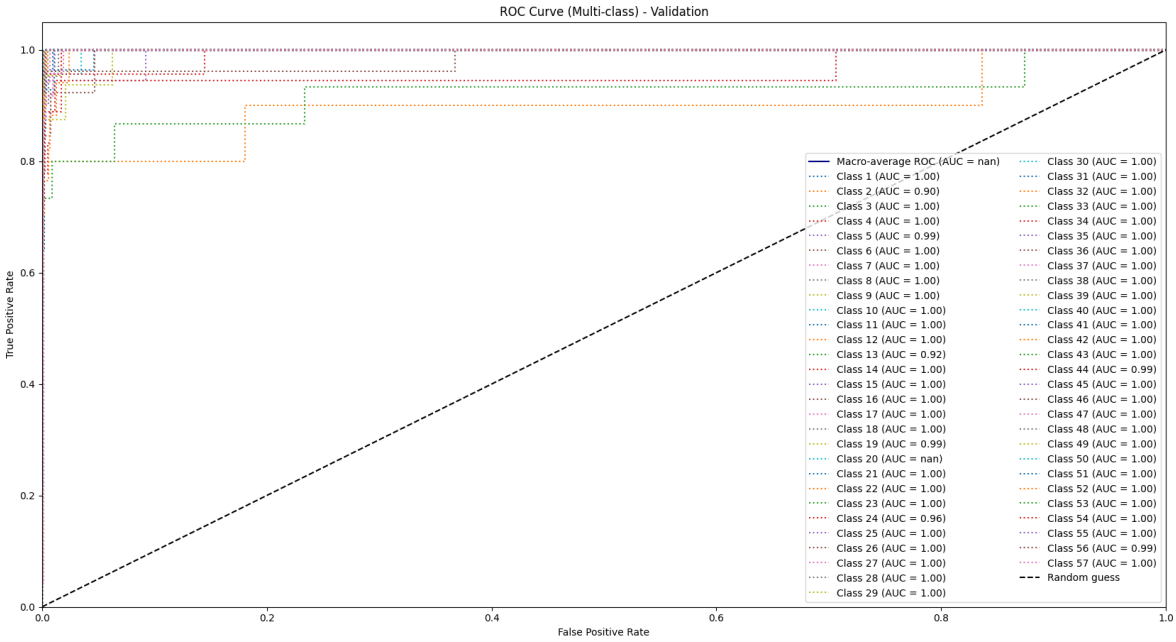
Based on the information provided, Figure 6.3 illustrates clear diagonal lines, which indicate correct predictions for both the training and validation datasets. Each cell's intensity corresponds to the number of predictions for that class, with darker shades of blue representing a higher number of correct classifications. Accordingly, the straight diagonal line in a darker blue shade defines accurate class predictions.

ROC Curve

To visually represent the performance of the developed multi-class classification model, Vilarino et al. (2006), ROC curves were generated for both the training and validation datasets.



(a) ROC curve the majority of the curves are concentrated on the top left corner for training results



(b) Validate results outlines the high but lower performance than ROC Curve-training

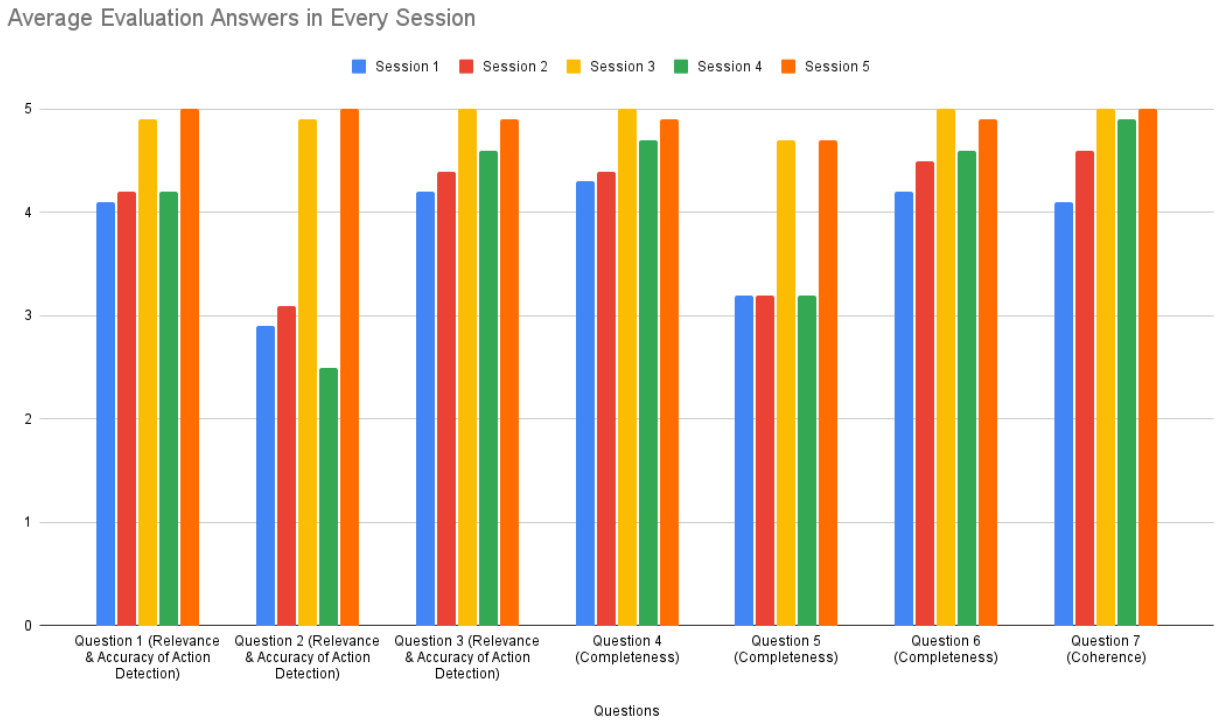
Figure 6.4: A ROC curve for both training and validation sets, representing slightly lower performance on the validation set.

Figure 6.4 illustrates these curves, plotting the TPR (True Positive Rate) against the FPR (False Positive Rate) at different classification thresholds, demonstrating the model's ability to distinguish between action classes. Figure 6.4 contains ROC curve diagrams for both the Training and Validation sets.

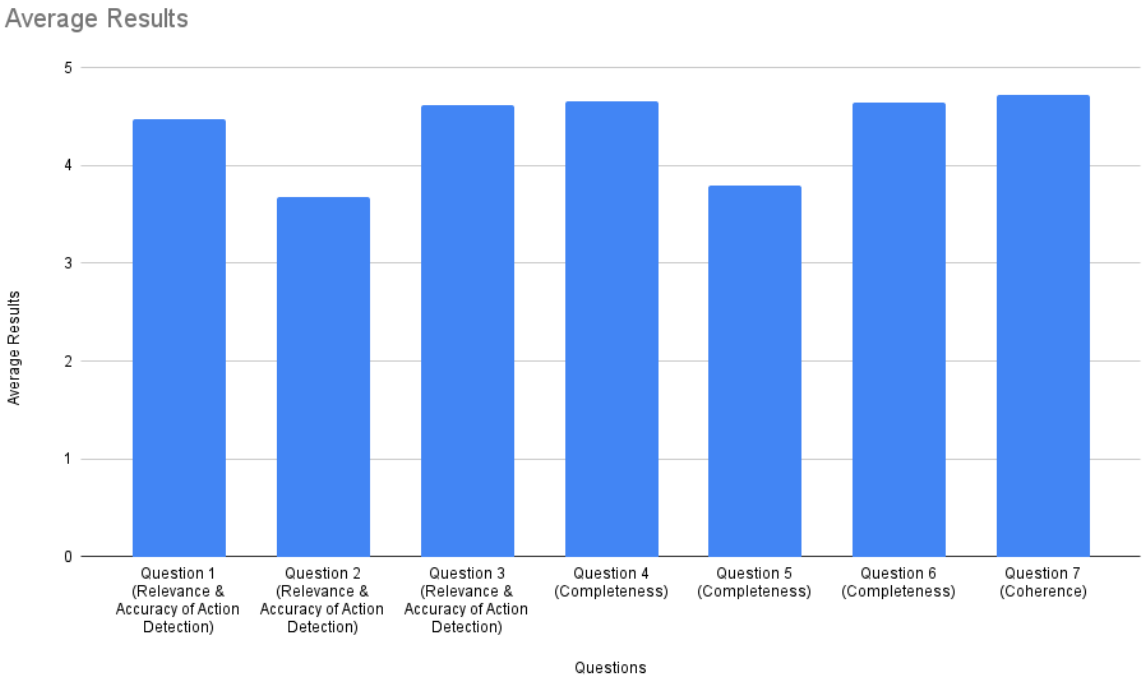
Figure 6.4 (a) shows a strong concentration of the ROC curve towards the top-left corner of the diagram for training results. This indicates successful performance for the majority of action classes. On the other hand, the validation Figure 6.4 (b) represents less concentration towards the top-left corner. Although the curve resulted in a lower AUC (Area Under Curve) than 1, this is expected as a model's performance mostly degrades slightly when applied to unseen data. Despite the reduction in AUC, both training and validation curves provide evidence to demonstrate the high performance of the system.

6.2 Summarisation Evaluation: Result Representation

This section presents the responses collected from 10 participants and explains what the results reveal about the system's functionality.



(a) A bar chart showing the average evaluated results for each session across all questions.



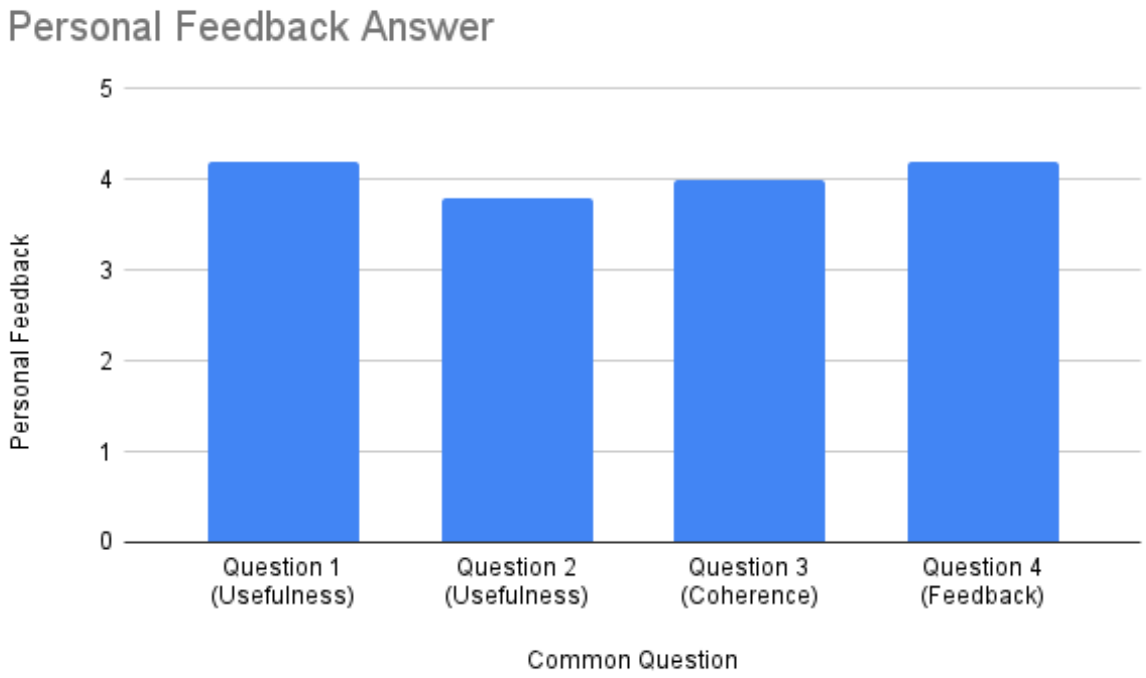
(b) Average session responses for each question.

Figure 6.5: Bar charts representing evaluation responses for each question.

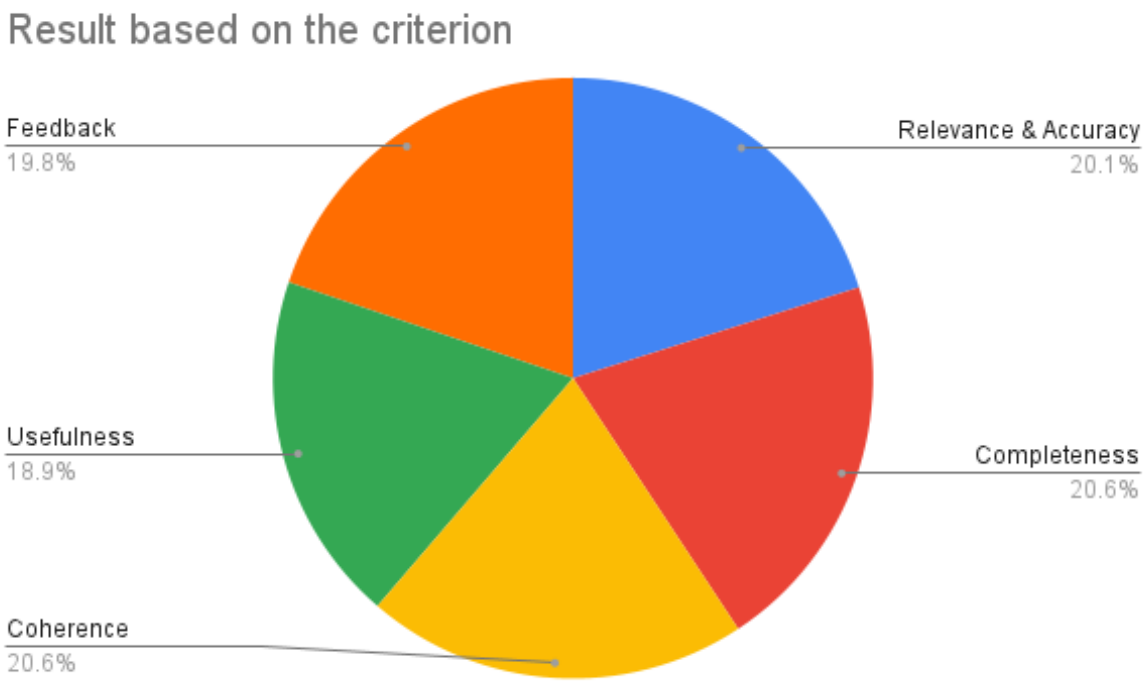
Figure 6.5 presents a visual representation of the average responses across each session as well as overall responses. According to Figure 6.5 (a), the responses can be distinguished into two groups: sessions 1, 2, and 4, and sessions 3 and 5. Sessions 1, 2, and 4 resulted in average scores above 4 for most questions, except questions 2 and 5, which addressed relevance and accuracy, and completeness criteria. The participants indicated that the input and output videos in these sessions were too short to effectively achieve the objectives outlined in the participant information sheet.

In contrast, sessions 3 and 5 received higher average responses due to the clear identification presented to users. Consequently, the responses formed two distinct groups in their evaluation of the system across the five criteria.

On the other hand, Figure 6.5 (b) combines the results from all sessions into a single bar chart per question. While two questions received an average score below 4, the overall average across all sessions was 4, indicating strong performance according to the evaluation criteria outlined in the Section, Summarisation Evaluation.



(a) Average session responses to the four personal feedback questions.



(b) Pie Chart: demonstrating the balance of the responses based on the five criteria

Figure 6.6: Figures representing feedback responses from the participants.

During the evaluation, participants responded to two types of questions: session-specific questions and general feedback questions. Figure 6.6 (a) presents the average score for the feedback questions, which was approximately 4, indicating "great" based on the suggested criteria.

In addition, Figure 6.6 (b) displays a pie chart illustrating the distribution of scores across the evaluation criteria. It demonstrates that the results are evenly distributed, each accounting for approximately 20%, suggesting a balanced performance without any particular area of weakness. This indicates that the system achieved moderate functionality, fulfilling the objective of establishing a fundamental infrastructure for a video summarisation system. Ultimately, the evaluation supports the successful completion of the task based on the stated objectives.

Do you have any additional comments or suggestions about this system? (Feedback)
N/A
Some output videos start with a bit of the previous content. It would be better if that part could be trimmed or improved
Three of the output videos had an irrelevant segment of the prior video from the input. Refining the system to only display the video segment that is relevant to the title displayed would be useful.
No
Summarising sports videos is an intriguing approach. While many highlight-focused contents are currently created manually, an automated, topic-based summarisation method could be both engaging and practical. If developed into a service, it has strong potential to attract interest.
Some irrelevant frames are included in some outputs. Most sports shown are correctly classified in the end so I am very confident in the system.
It would be better to use higher resolution videos.
Does not segment all the clips in the larger input clip - just takes one scene
Although the clips often started too early, in general the model performed well with identifying the classified activities in each output clip.

Figure 6.7: Feedback provided by the participants at the end of the questionnaire.

Figure 6.7 summarises the additional feedback provided by the participants. A common observation suggested that the output video often started with irrelevant segments. This issue was addressed by increasing the epoch size during the model training, which generates enhanced summaries.

Another suggestion concerned the video resolution. Since the project employed the UCF-101 dataset as both input and output, and this dataset is designed for research purposes, the resolution quality was inherently limited. While this constraint is acknowledged as a

limitation, future advancements will aim to incorporate a wider range of datasets to support varying resolutions for improved pattern recognition. Furthermore, future iterations of the system will aim to generate output videos with higher resolution, enhancing the user's ability to interpret the content effectively.

6.3 Discussion

6.3.1 Limitations

The suggested limitations for this project are following:

- **Computation Power:** The CNN-LSTM architecture was trained using the researcher's personal device. However, the project required a large number of iterations to learn features from each frame. Although classification was successfully achieved, the hardware limitations restricted the inclusion of additional features that could have improved training efficiency and execution performance. Consequently, the dataset size and the features used for model training were limited.

- **Dataset Limitation:** The UCF-101 dataset was suitable for this project. However, it has certain limitations in representing diversity. While it includes a range of action patterns, the videos are not long enough to capture variations in those patterns. For example, in the Archery class, the dataset only shows the moment when the player releases the bow. A more diverse dataset could include failed attempts, resting, or various camera angles. Due to limited dataset access, these variations were not available. Although an attempt to include a required dataset, such as SoccerNet, was unsuccessful, future work should focus on extending the dataset to fully evaluate the summarisation system's capabilities.

- **Model Limitation:** While the CNN-LSTM model produced successful results, no comparisons were made with alternative models to validate its relative performance. Although the model achieved over 98% accuracy, other models may offer better performance or shorter runtime. This project employed only one model and dataset, limiting the justification of CNN-LSTM as the optimal choice. Although previous research supports CNN-LSTM as an effective model, presenting executed results with alternative models would strengthen the selection.

- **Not Containing Crucial Segments for Single Sports:** Although Task B modified its objectives to fit the changed plan, the ultimate aim of summarisation is to generate a summary from a long video input, including crucial action patterns. For example, in a football match, the system should ideally include key moments such as goals, offsides, fouls, and penalties in a single summarised output. However, the current system is generating a summary video presenting only a single action.

6.3.2 Future Work

Based on the addressed limitations, the future works to enhance this system are:

- **Employ Additional Dataset for Enhanced Pattern Recognition:** Future work will involve employing additional datasets to capture a wider range of patterns within each action class. This will enable the system to enhance generalisation and improve prediction accuracy by learning from various representations of similar actions.
- **Train Various Models for Comparison** The project should employ various deep learning models for training. Comparing the performance of multiple architectures will highlight the strengths and weaknesses of each, providing a clearer justification for selecting a particular model. This approach would enhance the system's reliability and adaptability.
- **Utilise HPC for implementation:** Rather than relying on a limited computational environment, the project should utilise High-Performance Computing (HPC). HPC will allow for efficient training, training on larger datasets, and enable the exploration of more complex models. Consequently, using HPC would significantly enhance both development speed and model performance.
- **Employ Longer, Copyright-free Videos for Summarisation** Despite Plan B requiring a merged video of five different videos, the system ideally requires longer, copyright-free videos to achieve the ultimate objective. Consequently, a longer input is required to allow the system to learn from a broader set of patterns and produce more varied summaries, achieving the project's fundamental aim.
- **Generate Multi-Event Sport Summaries:** The system should aim to include multiple relevant segments within a single sport summary. For example, a football summary should capture various events such as penalties, fouls, and goals. This would ensure that the output provides a more comprehensive overview of the input video, enhancing its usefulness and realism.

Chapter 7

Conclusions

In conclusion, this project effectively designed and implemented a deep learning-based system for classifying and summarizing sports videos. The system demonstrated high classification accuracy and useful video summarization capabilities through an interactive user interface by utilizing the CNN-LSTM architecture and the optimized UCF-101 dataset. The project achieved its main objectives and includes positive feedback from user evaluations, despite difficulties with dataset access, computational resources, and the requirement to modify the initial technical plan. The encountered limitations, such as the system's current focus on single-action summaries and the lack of dataset diversity, suggest important directions for future work. By adding more diverse models or datasets and expanding the system's capability to handle longer sports videos containing multiple events, the system's performance and applicability can be further improved.

Bibliography

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. In *CVPR*, September 2016.

Thangarajah Akilan, Qingming Jonathan Wu, Amin Safaei, Jie Huo, and Yimin Yang. A 3d cnn-lstm-based image-to-image foreground segmentation. *IEEE Transactions on Intelligent Transportation Systems*, March 2019. URL `https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=8671459`.

Anthony Cioppa, Adrien Deliège, Silvio Giancola, Bernard Ghanem, Marc Van Droogenbroeck, Rikke Gade, and Thomas B. Moeslund. A contest-aware loss function for action spotting in soccer videos. In *CVPR*, March 2021. URL `https://arxiv.org/pdf/2011.13367`.

Adrien Deliège, Anthony Cioppa, Silvio Giancola, Meisam J. Seikavandi, Jacob V. Dueholm, Kamal Nasrollahi, Bernard Ghanem, Thomas B. Moeslund, and Marc Van Droogenbroeck. Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In *CVPR*, March 2021. URL `https://arxiv.org/pdf/2011.13367`.

Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, November 2014. URL `https://arxiv.org/pdf/1411.4389`.

Swapnanil Ghosh. Sports video classification and summarization. *Unpublished*, pages 1–10, September 2024.

Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. In *Computer Vision Foundation*, April 2018. URL `https://arxiv.org/pdf/1804.04527`.

Michael Gygli, Helmut Grabner, and Luc Van Gool. Video summarization by learning submodular mixtures of objectives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. URL `https://www.cv-foundation.org/openaccess/content`$_c vpr_2 015/papers/Gygli_V ideo_S ummarization_b y_2 01$

M Hossin and M.N. Sulaiman. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining and Knowledge Management Process*, 5(2), March 2015.

Md. Zabirul Islam, Md. Milon Islam, and Amanullah Asraf. A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images. *Informatics in Medicine Unlocked*, 2020.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. URL `https://ieeexplore.ieee.org/document/6909619`.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *An Introduction to Information Retrieval*. Cambridge University Press, 2009. URL `https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf`.

Ronald Mutegeki and Dong Seog Han. A cnn-lstm approach to human activity recognition. *IEEE*, April 2020. URL `https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=9065078`.

Melissa Sanabria, Frédéric Precioso, Pierre-Alexandre Mattei, and Thomas Menguy. A multi-stage deep architecture for summary generation of soccer videos. In *CVPR*, May 2022. URL `https://arxiv.org/pdf/2205.00694`.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Visual Geometry Group, University of Oxford*, 2014. URL `https://papers.nips.cc/paper`$_{files/paper/2014/file/00ec53c4682d36f5c4359f4ae7bd7ba1-Paper.pdf}$.

Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. URL `https://openaccess.thecvf.com/content`$_{cvpr2015/papers/Song_TVSum_Summarizing_Web2015_CVPR_pap}$

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. Technical report, Center for Research in Computer Vision, University of Central Florida, November 2012. URL `https://www.crcv.ucf.edu/papers/UCF101`$_{CRCV-TR-12-01.pdf}$.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, October 2015. URL `https://arxiv.org/pdf/1412.0767`.

Fernando Vilarino, Ludmila I. Kuncheva, and Petia Radeva. Roc curves and video analysis optimization in intestinal capsule endoscopy. *Pattern Recognition Letters*, June 2006.

Ming-Te Wu. Confusion matrix and minimum cross-entropy metrics based motion recognition system in the classroom. *Scientific Reports*, 2022. URL `https://www.nature.com/articles/s41598-022-07137-z`.

Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Yue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. *Shanghai Key Lab of Intelligent Information Processing*, pages 461–470, October 2015. URL `https://dl.acm.org/doi/abs/10.1145/2733373.2806222`.

Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Xi'an Institute of Optics and Precision Mechanics, CAS, 2018. URL `https://openaccess.thecvf.com/content`$_c vpr_2018/papers/Zhao_H SA - RNN_H ierarchical_S tructure - Adaptive_C VPR_2018_p aper.pdf$.

Appendices

Appendix A

Appendix: Acknowledgment of Generative AI Support

This document highlights the use of generative AI. The tool is used to enhance the readability and represent academical structure of the written contents. The AI support was focused only on refining grammar, style, and clarity to represent readable and understandable report.

This appendix highlights that all ideas, concepts, and objectives presented in this reports are entirely provided by the author. The model was applied strictly to support presentation of contents, without contributing to the generation of any ideas, methodologies, or structure of the report.

The integration of AI was carefully executed to maintain the originality and core contents, following ethical academic standards. This acknowledgment signifies transparency in the document's development.

Appendix B

Appendix: Ethics Review

The University Of Sheffield.

Downloaded: 20/05/2025
Approved: 28/04/2025

Seunghyun Im
Registration number: 200178857
Computer Science
Programme: Mcomp Computer Science(with Artificial Intelligence)

Dear Seunghyun

PROJECT TITLE: Sports video classification and summarisation
APPLICATION: Reference Number 067571

On behalf of the University ethics reviewers who reviewed your project, I am pleased to inform you that on 28/04/2025 the above-named project was **approved** on ethics grounds, on the basis that you will adhere to the following documentation that you submitted for ethics review:

- University research ethics application form 067571 (form submission date: 23/04/2025); (expected project end date: 05/05/2025).
- Participant information sheet 1151943 version 3 (21/04/2025).
- Participant consent form 1151944 version 4 (23/04/2025).

If during the course of the project you need to deviate significantly from the above-approved documentation please inform me since written approval will be required.

Your responsibilities in delivering this research project are set out at the end of this letter.

Yours sincerely

Luke Whitham
Ethics Admin
Computer Science

Please note the following responsibilities of the researcher in delivering the research project:

- The project must abide by the University's Research Ethics Policy: https://www.sheffield.ac.uk/research-services/ethics-integrity/policy
- The project must abide by the University's Good Research & Innovation Practices Policy: https://www.sheffield.ac.uk/polopoly_fs/1.671066!/file/GRIPPolicy.pdf
- The researcher must inform their supervisor (in the case of a student) or Ethics Admin (in the case of a member of staff) of any significant changes to the project or the approved documentation.
- The researcher must comply with the requirements of the law and relevant guidelines relating to security and confidentiality of personal data.
- The researcher is responsible for effectively managing the data collected both during and after the end of the project in line with best practice, and any relevant legislative, regulatory or contractual requirements.

Figure B.1: Ethics Review Approval Document authorised by the University of Sheffield.

Participant Information Sheet

Project title: Sports Video Classification and Summarisation
Researcher: Seunghyun Im
Email Contact: sim6@sheffield.ac.uk
Supervisor: Yoshi Gotoh

You are invited to take part in my undergraduate dissertation at the University of Sheffield. Before the beginning of the assessment, please read the following information carefully. If there are any questions based on the questionnaire, please contact the email sim6@sheffield.ac.uk.

What is the purpose of this research?
The project aims to evaluate the effectiveness and reliability of a system that generates summarised versions of sports video content. You are invited as a participant to help assess whether these summaries accurately capture the genre of the entire video.

What is my part in this research?
You will watch multiple videos and evaluate whether the generated videos are relevant to the topic that is displayed.

During this assessment, you will be asked to:

1. Watch an original sports video via a provided link.
2. Watch a summarised version of that same video, generated by the system.
3. Complete a short questionnaire provided from the Google Form.

The evaluation will take around 10-15 minutes to complete. Please find a safe place to begin the assessment.

Will I be paid to participate?
No. The participation is entirely voluntary.
You can withdraw the evaluation at any time before submitting your questionnaire, and your responses will be deleted and not used in the research.
Once you submit your response, the data will be anonymised and your data cannot be withdrawn.

Is there any risk or harm in taking part?
No. There is no predicted risk or harm associated with participation.
The study involves only watching videos and giving your opinion through the Google Form.

Confidentiality and Data Handling
Your responses will be collected via a Google Form and will not contain any identifiable personal data.
If any personal data is accidentally disclosed when contacting the researcher, it will be deleted immediately.
Only the researcher and the project supervisor will have access to the raw data.
All data will be securely stored in researcher's University Google Drive.
All data will successfully be deleted after completion of the project.

The University of Sheffield is the Data Controller for this research. Data will be processed under the legal basis of "public interest" as stated in Article 6(1)(e) of the GDPR. For more information, please see the University's Privacy Notice:
https://www.sheffield.ac.uk/govern/data-protection/privacy/general

If there is any questions or concerns:
Please contact to sim6@sheffield.ac.uk about further information or if there is any assistance required to your questionnaire.

Thank you very much for your participation.

Figure B.2: Participant Information Sheet provided to study participants.

Figure B.1 shows the official approval of the ethics application by the University of Sheffield review board. Following this approval, participant recruitment and evaluation activities commenced, in accordance with the guidelines outlined in the Participant Information Sheet (Figure B.2), which was made available to all participants through the Google Form used for the study.

My application form for the ethics review is as follows:

Application 067571

Section A: Applicant details

Date application started:
Thu 17 April 2025 at 17:24

First name:
Seunghyun

Last name:
Im

Email:
sim6@sheffield.ac.uk

Programme name:
Mcomp Computer Science(with Artificial Intelligence)

Module name:
COM3610 Dissertation Project

Last updated:
28/04/2025

Department:
Computer Science

Applying as:
Undergraduate / Postgraduate taught

Research project title:
Sports video classification and summarisation

Has your research project undergone academic review, in accordance with the appropriate process?
Yes

Similar applications:
- not entered -

Section B: Basic information

Supervisor

Name	Email
Yoshi Gotoh	y.gotoh@sheffield.ac.uk

Proposed project duration

Start date (of data collection):
Thu 17 April 2025

Anticipated end date (of project)
Mon 5 May 2025

3: Project code (where applicable)

Project externally funded?
No

Project code
- not entered -

Suitability

Takes place outside UK?
No

Involves NHS?
No

Health and/or social care human-interventional study?
No

ESRC funded?
No

Likely to lead to publication in a peer-reviewed journal?
No

Led by another UK institution?
No

Involves human tissue?
No

Clinical trial or a medical device study?
No

Involves social care services provided by a local authority?
No

Is social care research requiring review via the University Research Ethics Procedure
No

Involves adults who lack the capacity to consent?
No

Involves research on groups that are on the Home Office list of 'Proscribed terrorist groups or organisations?
No

Indicators of risk

Involves potentially vulnerable participants?
No
Involves potentially highly sensitive topics?
No

Section C: Summary of research

1. Aims & Objectives

The research aims to evaluate the effectiveness of a system designed to generate summarized versions of video content, by using human evaluators to assess the informativeness of the summaries. Due to the absence of standardized test sets for the summarization, human evaluation is necessary to verify the system's performance. The answer suggested by the small group of evaluators depends on the quality of summarization of the system.

2. Methodology

To conduct the evaluation, I will provide Google Forms and video links to the 10 participants to evaluate the system's output. Each particpant will complete a total of five sessions, and each session will involve watching two videos.
1. Input video (75 seconds) - This video is a compliation of five randomly selected action category from a dataset containing 57 actions (football, basketball, archer, etc.).
2. Output video (20 seconds) - The system will request the user to upload the input video and select one action category. If the selected action exists within merged video, the system will extract output video with 20 seconds of runtime.

After watching both videos in a session, participants will answer 5-10 evaluation questions through Google Form provided.

In total, each participants will complete 5 sessions, watch approxiamtely 575 seconds of video, and answer 25 to 50 questions.

Based on the result, I will analyse the evaluation of the summarisation.

3. Personal Safety

Have you completed your departmental risk assessment procedures, if appropriate?

No

Raises personal safety issues?

No

This assessment does not consider potential harm to participants due to the online questionnaire format. Moreover, the study only requires evaluators' personal opinions, no harm is expected.

Section D: About the participants

1. Potential Participants

Since the purpose of this evaluation is to assess the output video generated by the system,
I intend to recruit participants who have prior experience or at least watched the relevant
sport. This describes that they are capable of recognizing whether the output video accurately represents the key highlights of the match. Additionally, by selecting individuals who can evaluate the relative importance of different moments within the sport, I aim to determine whether all crucial events have been appropriately included in the generated summary.

2. Recruiting Potential Participants

The potential participants will receive a link to the evaluation directly from me. I intend to share the link to people who have played particular sports at least once, as their familiarity with the sport is important for this assessment. Accordingly, everyone with knowledge in the sports events can participate, without considering such factors as age and gender.

2.1. Advertising methods

Will the study be advertised using the volunteer lists for staff or students maintained by IT Services? No

- not entered -

3. Consent

Will informed consent be obtained from the participants? (i.e. the proposed process) Yes

The consent form will be given to all participants through Google Form links, asking their agreement in continuing the project.

4. Payment

Will financial/in kind payments be offered to participants? No

5. Potential Harm to Participants

What is the potential for physical and/or psychological harm/distress to the participants?

Since the project concerns sports videos and their summarisation, there would be no physical and/or psychological harm/distress to the participants.

How will this be managed to ensure appropriate protection and well-being of the participants?

Since this task will be conducted online, there is no expected physical and mental stress could casue to the participants. However, the participants may be tired by repetitive assessment. Therefore, the Participant Information Sheet will highlight the information where there is no time limit for completing the task, the participants may take breaks at any time during the evaluation if they wish to do so.

6. Potential harm to others who may be affected by the research activities

Which other people, if any, may be affected by the research activities, beyond the participants and the research team?

There will be no other people affected by the research activity, because the project will be progressed anonymously and will not be spread to third party. The data will be only reviewed by the supervisor and researcher of the project.

What is the potential for harm to these people?

N/A

How will this be managed to ensure appropriate safeguarding of these people?

N/A

7. Reporting of safeguarding concerns or incidents

What arrangements will be in place for participants, and any other people external to the University who are involved in, or affected by, the research, to enable reporting of incidents or concerns?

The participants should inform the researcher of the project by using the contact information. If not satisfied, contact the supervisor.

Who will be the Designated Safeguarding Contact(s)?

The Designated Safeguarding Contact will be the dissertation supervisor and the Head of the School

How will reported incidents or concerns be handled and escalated?

Any reported incidents or concerns will be escalated to the supervisor of the project, if not satisfied, then to Head of the School

Section E: Personal data

1. Use of personal data

Will any personal data be processed or accessed as part of the project?
Yes

Will any 'special category' personal data be processed or accessed as part of the project?
No

Provide the number of people whose personal data you expect to process or access.
10

2. Managing personal data

Which organisation(s) will act as data controller(s) of the personal data?
University of Sheffield only

Who will have access to the personal data?
Only the researcher will have access to the personal data, however the personal data will immediately be destroyed after submission of the Google Form.

What measures, processes and/or agreements will be put in place to manage the personal data?
Participants' responses will be collected via a Google Form, no personal data will be stored in evaluation record. After Google Form submission, the results will be recorded. However the personal data and the Google Form will be destroyed immediately after the collection. During the assessment, if any types of personal data, such as email addresses, is accidentally shown to the researcher, it will be deleted immediately.

Will all identifiable personal data in digital or physical format be destroyed within a defined period after the project has ended?
Yes

When will the identifiable personal data be destroyed?
The answers to the question will be recorded, however the personal data will be destroyed after the Google Form submission and reviewed by the researcher.

3. Third-party services

Will any external third-party services not provided by the University be used to process or access personal data during the project?
No

4. Security of computers, devices and software

Will personal data be processed or accessed on any computers or devices that are not managed by the University of Sheffield?
Yes

Will all computers and devices that are not managed by the University of Sheffield be secured in accordance with the IT Code of Connection?
Yes

Will any software not approved by the University of Sheffield be used to process or access data?
No

Will any software be written or developed in order to process or access the personal data?
No

Section F: Supporting documentation

Information & Consent

Participant information sheets relevant to project?
Yes

Document 1151943 (Version 3) All versions

Consent forms relevant to project?
Yes

Document 1151944 (Version 4) All versions

Additional Documentation

External Documentation

- not entered -

Section G: Declaration

Signed by:
Submit
Date signed:
Wed 23 April 2025 at 22:30

Official notes

- not entered -