Action Spotting in Football Matches

Department of Computer Science

School of Technology and Engineering

Nottingham Trent University

Under kind Guidance of

Dr. Ismahane Cheheb

Prepared by

Yash Uday Sule

(N1040923)

2023

Project Report submitted in partial fulfilment of the requirements of Nottingham Trent University for the degree of M.Sc. Data Science

DECLARATION

In submitting this work, I confirm that I am aware of, and am abiding by, the University’s expectations for proof – reading.

Name: Yash Uday Sule
NTU ID: N1040923

ABSTRACT

In computer vision, the area of Video understanding is growing and is quite challenging also. Videos related to football games in which spotting action events like “goals”, “substitutions” and “cards” across each video frame is quite difficult due to their sparse nature. In this work, whether a video frame is an action frame or a non – action frame has been investigated. A sub – sample of the SoccerNet benchmark dataset was considered, which contains annotated action events for six football games from English Premier League season 2014/15. Overall, four different pooling techniques have been studied and applied to build deep learning models. In addition to this, using three types of pre – extracted features along with four pooling techniques a total of 12 deep learning models have been designed and a comparative study has been carried out. After performing 5 experimental runs for each model and based on evaluation metrics like Precision, Recall and F1 – score, two models have surpassed other models. One is using ResNET features with NetVLAD pooling technique giving F1 – score of 0.4850 and other using I3D features with NetVLAD pooling technique giving 0.5026 as F1 - score.

ACKNOWLEDGEMENTS

I would like to take this opportunity to sincerely thank all those people, without whomthis major project is incomplete and would not be in good shape.

Firstly, I would like to express my gratitude towards Dr. Ismahane Cheheb for being such a kind and generous supervisor. Her immense knowledge, enthusiastic nature and inclination towards the research topic made me inspire to carry out my efforts towards successful completion of this major project. The discussions during the meetings helped me a lot that made me think in different perspectives that were necessary for the project. Also, her valuable timely suggestions, feedback and comments aid me to towards the completion of major project in time. Thank you for your continuous support and guidance.

Secondly, I would like to thank Major Project module leader, Dr. Temitope Alade for providing with important information regarding the necessary submission requirements and other queries.

Thirdly, I would also like to thank the Department of Computer Science and Nottingham Trent University for provided with the necessary facilities that helped me to complete my work on time.

Finally, I would like to thank my family and friends for their unconditional support throughout the development of my major project.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1: **Introduction**

1.1 Background and Motivation

In the past few years, sports as a sector have turn out to be a money-making area where enormous amount of funds are utilized on players (with regards to fitness, training and overall development) as well as on a team unit as whole(GIANCOLA, S. et al., 2018). According to "Research and Markets" report the worldwide market of sports is expected to grow at an exponential rate approximately from $441 billion (£351 billion) and reach $600 billion (£478 billion) at a compound annual growth rate (CGPA) of 8% by 2025 (RESEARCH AND MARKETS, 2021). In addition to these, if we look only at the Big Five European soccer leagues (English Premier League, La Liga, Ligue 1, Bundesliga and Serie A) then its combined revenue is projected around 6.1 billion Euros for the season 2021/22 (LANGE, D, 2022).

Also, these days the volume of TV broadcasting of sporting events around the globe has increased rapidly (GIANCOLA, S. and Ghanem, B., 2021). This turns out to be the second major revenue generation for football clubs. According to the survey it has been estimated that only from broadcasting the soccer clubs will generate revenue of 3.1 billion British pounds for the season 2021/22. Concurrently, the revenue from deals pertaining to commercials and sponsorships will accumulate to 1.7 billion British pounds and on the matchdays around 700 million Euros will be amounted (LANGE, D, 2022).

Even though there is a scope of tons of earnings from this sector of broadcasting there is also another critical aspect of analysing important events like "goals", "substitutions", "yellow / red cards", "penalties", "offside" and many more which generally occurs in football matches. By doing these, it can help teams to design new strategies and game plan for upcoming matches. Moreover, if efficient machine learning models are developed to captured important actions from these videos, then it will lead to automate the process of generating highlights of the football matches. This motivates me to pursue thisaspect of broadcasting videos.

1.2 Aims and Objectives

Aim: The project aims to study the existing state-of-art deep learning algorithms for action spotting and developed deep learning model for detecting events in football matches by using different metrics to compute the accuracy for the developed model.

Objectives: To achieve the aim of the project, following are the potential objectives which are identified and are mentioned below:

- To carry out an extensive background research of the topic "Video Understanding".
- Understanding the approaches and their methodologies for the current state-of-art algorithms.
- Reviewing and reporting the short comes of the previous state-of-art algorithms.
- Developing new artificial intelligence model based on deep learning techniques.
- Comparing and analysing the different outcomes of the model being implemented by using different performance metrics to calculate the model accuracy.

Now, as aim and objectives of the project have been identified and are well-defined, the next section will elaborate more on the how to carry out and perform certain tasks that could achieve the laid down aim and objectives for the project.

1.3 Planning and Management

1.3.1 Tasks

This section is very crucial as it has been mentioned in (OGBEIWI, O., 2017) that tasks that are not structured well lack rationale and projects have no accountability. Furthermore, all the tasks need to be defined carefully as they play a vital role in achieving the overall objective of the project for a successful completion. Also, they should be monitored through intermediate outcomes either over a short or long period of time. As a result, all the deliverables must follow SMART (Specific, Measurable, Appropriate, Realistic and Time bound) approach as it has become a vital tool for managing research projects (DUCA, G., 2021).

For this project following is the list of tasks which are divided into four phases. They are as follows:

Phase – 1: Project Registration

- Selecting a project from NOW portal from a list of projects, applying for that project and getting an approval confirmation email form the Course Leader.

Phase – 2: Background Research

- Collect and read relevant literatures / articles / conference proceeding papers.
- Analyse the essential information and document it in the form of a literature review.

- Extract and tabulate all the techniques used to design artificial intelligence models using deep learning approach.

Phase – 3: Designing and Testing Models using Deep Learning algorithms

- Participating in online lectures and tutorials for getting depth on the implements of advanced deep learning techniques in a programming language named “Python”.(parallel task)
- Preparing datasets to perform various programming tasks such as pre-processing of the data as well as dividing it into appropriate training and testing datasets.
- Design and implement the developed code using deep learning techniques on datasets for performing “action classification”.
- Report and analyse the accuracy of the designed model.
- Perform appropriate changes in the model to achieve higher accuracy.

Phase – 4: Documentation

- Report detailed finding of the results along with the methodologies used for obtaining these results.
- Prepare a draft version of the Major Project and get a feedback on it from the project supervisor.
- Finalize the whole thesis and submit it on NOW portal.

The next section of this project report document will focus on the sources or resources that are necessary to accomplished all the mentioned tasks and to complete the project.

1.3.2 Sources of information and resources required

This is the second most important section of the project planning. In every project, sources and resources have an immense contribution in order to achieve success towards the completion of a project. Also, it is crucial to select the most adequate tools for the project, ideally even before the project has begun, so that we can make maximum use of them to have smooth running of entire project and increase its profitability in terms of gaining good results. Moreover, it ensures that those tasks which have a higher priority have the right resource at the right time (NÖSEL, I., 2020) (PANCHAL, M., 2021).Thus, for the current project the required sources and resources have been discussed further.

Books / Journals / Conference proceeding articles / Case Studies:

These would be the most prominent sources of information which will form the bases of research for the chosen topic under investigation. So this project will require the access of the "Library One Search" and "Library One Search Pro" facilities which are owned by Nottingham Trent University (NTU) and available for students who are at NTU. It has a very rich database with many capabilities like filtering your search, requesting for book or a journal, paper and targeted search options. Apart from this, "Google Scholar" will be used as an online search tool which has access to all the academic literatures and books from all over the world. It also has some unique filtering options like limit your search based on citations, year and many more.

Resources in terms of Software:

A lot of research work is carried out all over the world with the help of available software technologies. It has been claimed based on their survey which was carried out among 417 researchers from the 15 Russell Group universities that 92% of the academicians use software for their research and 69% researches stated that research would be impractical or they cannot conduct their research if there is no software (HETTRICK, S. et al., 2014). Thus the project will require using the following softwares:

- Jupyter Notebook / Kaggle / Google Colab: Either of these tools will mainly be used for writing codes related to deep learning algorithms using Python programming language.
- Microsoft Word 2021: This tool will be used for documentation of the literature, analyses and results, discussion and conclusions.
- Microsoft Excel 2021: This tool will be used to generate experimental tables and compute some important statistics which can be visualized using graphs.

Hardware requirements:

For this particular project there are not requirements of specific hardware but a large amount of storage space will be required to save the data either on computer or laptop. Other requirements are specified below:

- Operating system: Microsoft Windows 10 (64-bit) or higher.
- RAM: 8GB or more.

- Processor: Intel Core i5(6th Generation) 1 GHz or faster with minimum 4 cores.
- Storage: 1 TB HDD (Hard Disk Drive) or 512 GB internal SSD (Solid State Drive).
- Internet: Active internet connection.

The above requirements have been currently identified it might change in the due course of time as we proceed with actual implement of the research work.

1.3.3 Project risks

This section of the project plan is mainly dedicated to represent the different risks that a project can encountered when it begins its actual execution. Risk is a key factor that one should examine before the commencement of the project. Since, it gives a comprehensive idea of major risks and how to deal with them. This greatly reduces the changes of missing the important milestones of the project. For this project, some of the potential risks are listed below and discussed further in detail.

Time Management Issues: This might happen since being a research student there are lots of distractions which makes it is difficult to concentrate on work and lacks 100% engagement in a task. Along with this, it might happen that there may arise some unforeseen circumstances which are not avoidable such as social engagements. This might delay the project and fail to submit it on time.

Health Issues / Pandemic situations: This is certain if the researcher does not follow the norms laid down by the government for the safety of their citizens. This might affect mentally and will increase stress and other health problems. Since much of the days will be wasted in recovering from such problem sit would not be possible to carry out the work and submit the project at its scheduled time.

Natural Calamities: These events are very rare globally as it has been reported by “Our World in Data” that on an average 45,000 people per year are killed due to such events. Moreover, within the last ten years it accounted for 0.1% of deaths (RITCHIE, H. and Roser, M., 2021). But such a form of risk has been considered since this might affect few authorities of the ongoing project that are closely associated with such a disaster. This would definitely force a delay in the project.

Insufficient research: This might occur if we lack in analysing good sources/ materials for our initial literature survey. The ultimate consequences of this will be reflected in the final output of the project in which the model performs very badly and gives abrupt results.

Difficulty in self – learning: This could be possible since most of the time it is the researcher who tries to learn many of the things on his own by referring to lot of non-reliable online material in the forms of tutorial videos or blogs. This might take a lot of unnecessary time and it may point it to a wrong direction and only after its implementation he might discover that it was a total failure. This would impact the overall objectives of the project and will not give good results at the end.

Technical difficulties: This kind of risk may arise since most of the project work will depend upon the technical resources that should be compatible to handle the problems related to data acquisition, storage, cleaning and preparation.

Data Loss: This form of risk is quite certain if we do not take proper steps for its prevention. The following two losses would impact the project potentially and will never achieve the targeted aim and objectives.

- Loss of programmed code files/ Code files being corrupted.
- Loss of thesis file.

Since, most of the project work will be carried out on laptop or computer the mentioned losses are not acceptable. The reasons for this loss can be any of the following:

- Failure of Hard drive.
- Malware infections caused by computer viruses.
- Damaged caused by water or coffee spilling.

Due to such causes there is a high probability that the project will not be completed on the scheduled time.

1.3.4 Risk Mitigation Plan

This section will discuss about how the mentioned project risks can be avoided if we follow a mitigation plan which will help the researcher to achieve its goals towards project completion.Table1.1 and Table1.2 are the scale indicators to identify the level of risk and risk matrix respectively. Risk matrix is designed as the intersection of the probability and impact of the risk.

Rating Scale	Risk Level (Probability * Impact)
1,2,3	Low
4,6	Medium
9	High

Table 1.1: Risk Level Indicator based on rating scale

		Impact		
		Low	Medium	High
Probability	Very Likely	3	4	9
	Possible	2	4	6
	Unlikely	1	2	3

Table 1.2: Risk Matrix

Table 1.3 indicates how the status of the risk was decided based on the risk level.

Status Key		
Unacceptable	High	Disruption likely - Requires attention and closely managed
Reasonably Acceptable	Medium	Possible disruption - Needs to be managed
Acceptable	Low	No disruption likely - Needs to be monitored

Table 1.3: Indicator for Status Key

Table 1.4 shows the Risk Mitigation Plan which shows the possible solutions for causes of risks. It also specifies status, probability, impact, risk level and the risk owner.

Table 1.4: Risk Mitigation Plan

Risk Number	**Description of Risk**	**Status**	**Probability**	**Impact**	**Risk Level**	**Mitigation plan**	**Owner of risk**
1	Time Management Issues	Reasonably Acceptable	Possible	High	Medium	Strickly follow the time plan designed in the form of Gantt Chart.	Self
2	Health issues / Pandemic situations	Reasonably Acceptable	Possible	Medium	Medium	Take precautions and follow government safety norms.	Self
3	Natural Calamities	Reasonably Acceptable	Unlikely	High	Low	Stay mentally strong and do not loss the focus from the targeted work.	Unknown
4	Insufficient research	Reasonably Acceptable	Possible	High	Medium	Ensure that the sources being referred are academically reputable. Also, spent good amount of time on analysis to make sure the literatures, journals, books, conference papers are reliable.	Self

5	Difficulty in self – learning	Reasonably Acceptable	Possible	Medium	Medium	Immediately contact the supervisor to seek proper advice through email.	Self
6	Technical difficulties	Reasonably Acceptable	Possible	Medium	Medium	Ensure that all the technical resources are latest and working properly before the commensment of the project.	Self
7	Loss of programmed code files / Code files being corrupted	Unacceptable	Very Likely	High	High	Always save your work while working on the project. Take more than one backup of the files.	Self
8	Loss of thesis file	Unacceptable	Very Likely	High	High		Self

Table 1.5 summarises the different types of risks and Figure 1.1 shows the corresponding pie – chart of the risks.

Risk Count	
Low	1
Medium	5
High	2

Table 1.5: Risk Count

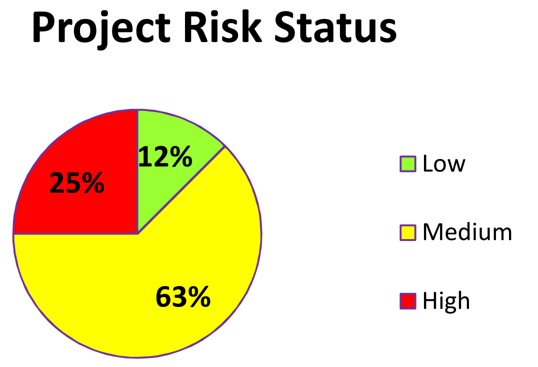


Figure 1.1: Pie Chart of Project Risk Status
(Designed using Microsoft Excel 2007)

1.3.5 PSEL issues

This section discusses about the Professional, Social, Ethical and Legal (PSEL) Issues that needs to be considered while executing the project. It is very important to examine all the aspects related to PSEL issues else the researcher might fall into an unknowingly situation and ultimately will not be able to add any value to the society.

Professional Issues: From the project perspective, it will be dealing in automating the whole process for the topic under investigation, there might have slight adverse effect on the professionals working in that field. Also, it is not possible to design a system to have 100% accuracy so issues regarding its success will be a concerned. Thus, if the system is put into actual practice, it might happen that it detects a wrong action and gives a completely different output than expected.

Social Issues: The work from this research will not impact any religion or community directly since there are no surveys involved in the research and no personal information will be collected or stored as a part of project.

Ethical Issues: In this project, there is no involvement of human participants and animal experiments / trials. Hence, one does not need to get a prior approval before the commencement of the project. But still the researcher needs to obey the following ethics:

- Be sincere and hardworking.
- Critically analyse the results before publishing it.
- Should not control the output of the results by modifying the code files.
- Should avoid copying other researchers work without seeking his permission.

- Do not forget to take necessary permissions to carry out a particular work, if required.

If the above rules are not followed then the researcher needs to face few consequences like copyright of work, patents issues and would turn out to be an academic misconduct and malpractice.

Legal Issues: Although, if the artificial intelligence system developed through deep learning algorithms when put into effect then it will be able to classify events that can be helpful in generating highlights. But it is not possible to design 100 % accurate system thus relying solely on it will generate a risk of misclassification of some actions and hence the highlights would not be generated properly by missing out important events that must had occurred. Thus, the viewers would not be able to enjoy the highlights and might complain against the developed artificial intelligence system.

As a part of this research project, one needs to consider the following legal regulations:

- Copyright, Designs and Patents Act (CDPA)1988.
- Computer Misuse Act 1990.
- General Data Protection Regulation (GDPR) 2018.

This will ensure that the project will not access or use any form of data without prior permissions from the authorities.

1.3.6 Time Plan

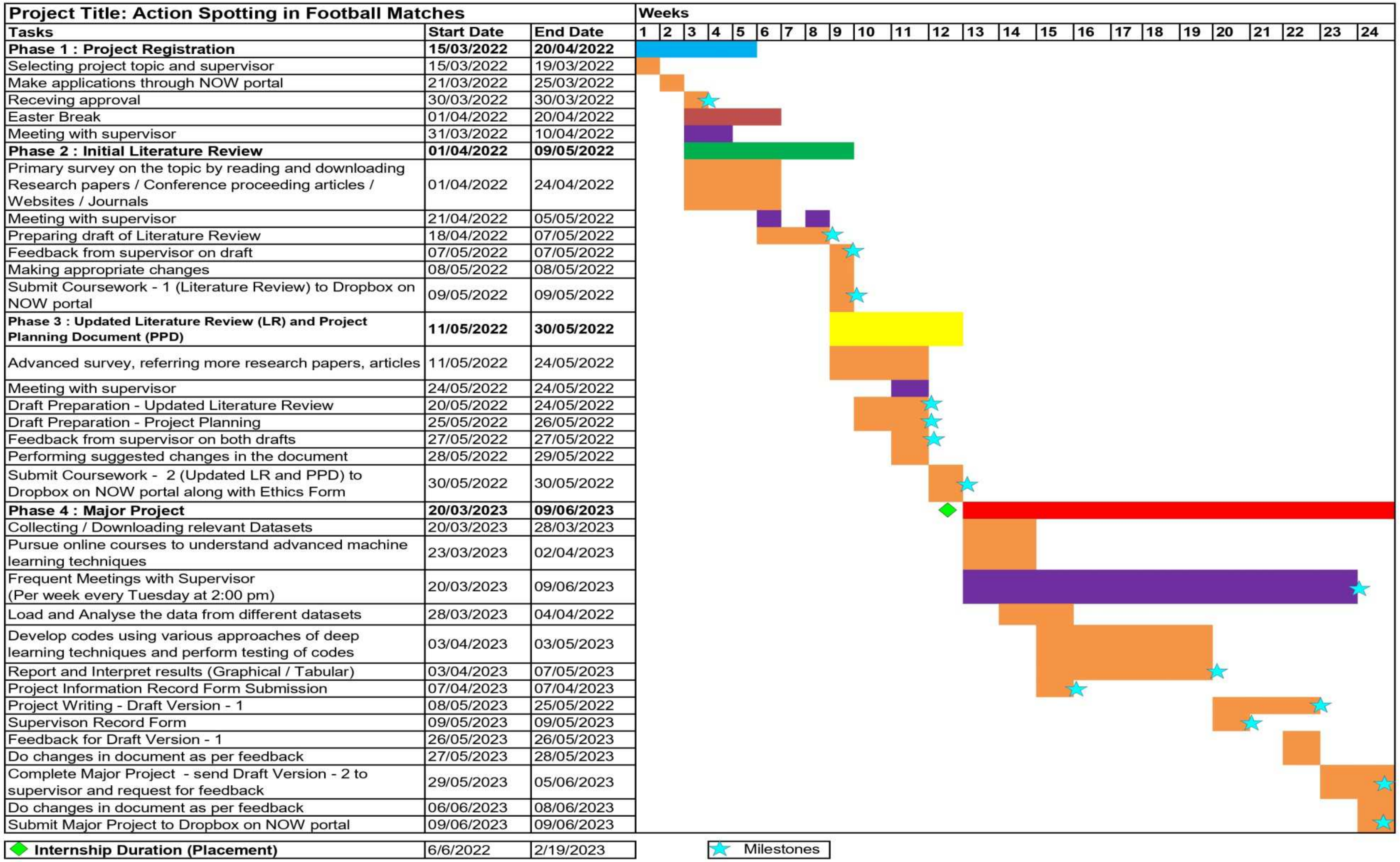
Figure 1.2 below shows the time plan for the project in the form of the Gantt chart. The summarized details about the project are as follows:

Project Title: Action Spotting in Football Matches
Start Date: 15 – 03 – 2022
Internship Period: 06 – 06 – 2022 to 19 – 02 – 2023
Recommencement Date: 20 – 03 – 2023
End Date: 09 – 06 – 2023
Duration:≈24 Weeks (≈5 months)

The follows Table1.6 of indicators will help in understanding the flow of the horizontal bars in the Gantt chart.

Purpose	Indicator
Phase 1 : Project Registration	🟦
Phase 2 : Initial Literature review	🟩
Phase 3 : Updated literature review and project planning document	🟨
Phase 4 : Major Project	🟥
Intermediate tasks	🟧
Meetings with Supervisor	🟪
Easter breaks	🟫

Table 1.6: Indicators with its purpose in Gantt chart



Project Title: Action Spotting in Football Matches		
Tasks	**Start Date**	**End Date**
Phase 1 : Project Registration	**15/03/2022**	**20/04/2022**
Selecting project topic and supervisor	15/03/2022	19/03/2022
Make applications through NOW portal	21/03/2022	25/03/2022
Receving approval	30/03/2022	30/03/2022
Easter Break	01/04/2022	20/04/2022
Meeting with supervisor	31/03/2022	10/04/2022
Phase 2 : Initial Literature Review	**01/04/2022**	**09/05/2022**
Primary survey on the topic by reading and downloading Research papers / Conference proceeding articles / Websites / Journals	01/04/2022	24/04/2022
Meeting with supervisor	21/04/2022	05/05/2022
Preparing draft of Literature Review	18/04/2022	07/05/2022
Feedback from supervisor on draft	07/05/2022	07/05/2022
Making appropriate changes	08/05/2022	08/05/2022
Submit Coursework - 1 (Literature Review) to Dropbox on NOW portal	09/05/2022	09/05/2022
Phase 3 : Updated Literature Review (LR) and Project Planning Document (PPD)	**11/05/2022**	**30/05/2022**
Advanced survey, referring more research papers, articles	11/05/2022	24/05/2022
Meeting with supervisor	24/05/2022	24/05/2022
Draft Preparation - Updated Literature Review	20/05/2022	24/05/2022
Draft Preparation - Project Planning	25/05/2022	26/05/2022
Feedback from supervisor on both drafts	27/05/2022	27/05/2022
Performing suggested changes in the document	28/05/2022	29/05/2022
Submit Coursework - 2 (Updated LR and PPD) to Dropbox on NOW portal along with Ethics Form	30/05/2022	30/05/2022
Phase 4 : Major Project	**20/03/2023**	**09/06/2023**
Collecting / Downloading relevant Datasets	20/03/2023	28/03/2023
Pursue online courses to understand advanced machine learning techniques	23/03/2023	02/04/2023
Frequent Meetings with Supervisor (Per week every Tuesday at 2:00 pm)	20/03/2023	09/06/2023
Load and Analyse the data from different datasets	28/03/2023	04/04/2022
Develop codes using various approaches of deep learning techniques and perform testing of codes	03/04/2023	03/05/2023
Report and Interpret results (Graphical / Tabular)	03/04/2023	07/05/2023
Project Information Record Form Submission	07/04/2023	07/04/2023
Project Writing - Draft Version - 1	08/05/2023	25/05/2022
Supervison Record Form	09/05/2023	09/05/2023
Feedback for Draft Version - 1	26/05/2023	26/05/2023
Do changes in document as per feedback	27/05/2023	28/05/2023
Complete Major Project - send Draft Version - 2 to supervisor and request for feedback	29/05/2023	05/06/2023
Do changes in document as per feedback	06/06/2023	08/06/2023
Submit Major Project to Dropbox on NOW portal	09/06/2023	09/06/2023
◆ **Internship Duration (Placement)**	6/6/2022	2/19/2023

Figure 1.2: Gantt Chart
(Designed using Microsoft Excel 2007)

Chapter 2: Literature Survey

2.1 Introduction

In the last few years, the field of computer vision has grown rapidly and has gone way beyond our imagination. One of the most interesting areas and quite challenging that has become the core domain of research is “Video Understanding”. Researchers want to have an overall understanding of the scenes in a video by performing some analyses over the different patterns of motion in it. In this literature review chapter, we focus our attention on sporting videos. Among these, we will indeed discuss in detail about interpreting videos relating to soccer matches.

Nowadays, everybody can browse to a never-ending hours of video content through services like YouTube, Netflix and Amazon Prime. Also, there are hundreds of TV channels that are running 24 / 7 that broadcast different entertainment shows and live streaming of worldwide sports which can last for hours until it is concluded. As, it has been mentioned in previous section of background and motivation that although streaming of athletic events gives fans more entertainment, but these kinds of videos are used by professional experts to analyse a group of statistics such as player and team performances, overall strengths and weakness and much more.

In this regard, one needs to summarize the interesting events either by compressing the video or by annotating the important actions. For football matches, this summarization would frequently consist of game highlights which will have goals, substitutions and fouls. For these purposes, there are authorized firms such as StatsPerform (STATS PERFORM, 1981), Sportsradar (SPORTRADAR, 2001) and OctaSports (OPTASPORTS, 1996) which have made sports analytics as their fundamental business and had provided with several of products for analysing advanced statistics and for generating highlights.

Currently, in most of the cases to obtain statistical information a lot of manual annotation must be carried out by the creators. Professionals must watch the broadcasts again and again to identify the different events which occur within a game (GIANCOLA, S. et al., 2018). According to one expert as mentioned in this research paper says that it takes over approximately 8 hours to generate around 2000 annotations for each game.

However, it is a very tedious task to detect the soccer actions such as “goals”, “substitution” or “showing Yellow/ Red cards” in a match as compared to other events which are sparse in nature within a video. Therefore, retrieving information about the tasks mentioned just now from soccer broadcasting videos can be looked upon as sub – problem of video understanding.

2.2 Video Understanding

There are different approaches that researchers use these days in order to gain insight about video understanding. Some popular ones are activity recognition, spatio- temporal localization and video classification (MINOURA, H. et al., 2021).

2.2.1 Action Recognition

Action recognition is a problem of detecting action from the videos and further classifying the video segments in their pre – defined action class labels. In 2014, researchers (SIMONYAN, K. and Zisserman, A., 2014) proposed a two – streamed Convolution Neural Network (CNN) trained on multi – frame dense optical flow to recognize action from the videos. The model was evaluated on benchmark datasets such as HMDB51 (KUEHNE, H. et al., 2011) and UCF – 101 (SOOMRO, K. et al., 2012).

Other popular work was the proposed by (CARREIRA, J. and Zisserman, A., 2017), in which a new Two-Stream Inflated 3D ConvNet (I3D) that is based on 2D ConvNet inflation was introduced and evaluated on the same two datasets as done by (SIMONYAN, K. and Zisserman, A., 2014). The results on these algorithms were far better than the ones given by (SIMONYAN, K. and Zisserman, A., 2014).

2.2.2 Video Classification

Video classification is a task of classify the content of the video based on their annotated labels. Back in 2014, the researchers (KARPATHY, A. et al., 2014) have used CNN on large – scale dataset having 1 million video clips from the Youtube from the domain of sports having 487 classes. The CNN architecture was designed in such a way that it learns the spatio – temporal information from the video and use to classify the videos into relevant action classes.

Afterwards in 2015, (DONAHUE, J. et al., 2015) proposed a long – term recurrent CNN model that was end – to – end trainable with temporal dynamics also being updated simultaneously.

The model was tested on benchmark video recognition dataset UCF – 101 (SOOMRO, K. et al., 2012). In most of the datasets a single label is assigned to the entire videos and thus research related to video classification is based on single label.

The most recent algorithms work on three dimensional CNN and Recurrent Neural Networks (RNN) (UR REHMAN, A. et al., 2023) for both approaches.

2.2.3 Datasets for Video Classification

A wide range of datasets are available for understanding various sub – domains of video understanding. The following Table 2.1 shows a descriptive statistics about datasets which are used for video classification.

Proposed by, Year	Dataset	Content	Number of Videos	Number of Instances	Duration (hrs)	Number of Classes
(MARSZALEK, M. et al., 2009)	Hollywood2	Movies	3669	-	20.1	12 (Actions)+ 10 (Scenes)
(KUEHNE, H. et al., 2011)	HMDB51	Movies	6849	-	-	51
(ROHRBACH, M. et al., 2012)	MPII Cooking	Cooking	44	-	-	65
(SOOMRO, K. et al., 2012)	UCF101	General	13320	101	-	25
(RODRIGUEZ, M.D. et al., 2008)	UCFSports	Sports	150	-	0.26	10
(NIEBLES, J.C. et al., 2010)	Olympics Sports	Sports	800	-	-	16
(KARPATHY, A. et al., 2014)	Sports-1M	Sports	1133158	-	-	487
(ABU-EL-HAIJA, S. et al., 2016)	Youtube-8M	General	6100000	18300000	350000	3862
(KAY, W. et al., 2017)	Kinetics	General	495547	495547	1377	600

Table 2.1: Comparison of benchmark datasets for tackling the problem of video classification.

The content of these datasets focuses their attention on the actions in movies, cooking activities, various sports events or general-purpose activities. Out of all these datasets as it can be seen from the Table 2.1 that Youtube-8M and Kinetics are used for performing large scale video classification.

2.2.4 Datasets for Action Localization

The number of benchmark datasets for detecting action localization is listed below in Table 2.2 which gives a very detailed description of them in terms of total durations, number of classes and most importantly the number of events per class.

Literature, Year	Dataset	Content	Number of Videos	Number of Instances	Duration (hrs)	Number of Classes	Avg. Event per class	Avg. Video Length (sec)
(JIANG, Y.G. et al., 2014)	THUMOS14	General	413	6363	24	20	318	209.2
(YEUNG, S. et al., 2018)	MultiTHUMOS	General	400	38690	30	65	595	-
(CABA HEILBRON, F. et al., 2015)	ActivityNet	General	19994	30791	648	200	154	116.7
(SIGURDSSON, G.A. et al., 2016)	Charades	General	9848	66500	82	157	424	30
(GIANCOLA, S. et al., 2018)	SoccerNet	Soccer	1000	6637	764	3	2212	2750.4
(YU, J. et al., 2018)	Yu et al.	Soccer	222	6850	167	11	623	2708.1
(JIANG, Y. et al., 2020)	SoccerDB	Soccer	171191	37715	668.6	11	3428	14.1
(DELIEGE, A. et al., 2021)	SoccerNet-v2	Soccer	1000	110458	764	17	6498	2750.4

Table 2.2: Descriptive comparison of the benchmark datasets available for action localization.

2.3 Computer Vision in Football

Subsequently as a part of research, there are number of computer vision methods related to soccer which aims to address the problems pertaining to low level of understanding in soccer broadcasting. For instance, identification of field and its boundaries locally using images(FARIN, D. et al., 2003 and HOMAYOUNFAR, N. et al., 2017), tracking the trajectories of players and detecting them(MANAFIFARD, M. et al., 2017 and YANG, Y. and Li, D., 2017), estimating the athlete’s pose using footage in the form of video(LEWIS, Bridgeman et al., 2019 and ZECHA, D. et al., 2019), distinguish teams in a match (ISTASSE, M. et al., 2019), possession statistics of players having ball (SARKAR, S. et al., 2019 and THEAGARAJAN, R. et al., 2018) or determining feasible passes among the players (ARBUES-SANGUESA, A. et al., 2020).

In order to generate automated broadcasting, there is a need to understand the advanced high-level gaming of this sport. This leads to understand the spatio-temporal information which in turn could enhance the virtual experience by using augmented reality devices of a sports fan (REMATAS, K. et al., 2018) and to collect data driven information of players (THOMAS, G. et al., 2017).

As mentioned in one of the research article (CIOPPA, A. et al., 2020), in order to automate the annotation process related to the tasks such as generating short summaries, highlights of a match and reporting of human centric actions per game, computer vision should have an depth of understanding regarding the salient features of an sporting game. Thus, it requires an extensive knowledge of time and space within a video. Overall, the most challenging tasks with the present algorithms of video understanding are to localize an action in time in a video.

Among the mentioned datasets in Table 2.2, last four datasets namely SoccerNet (GIANCOLA, S. et al., 2018), (YU, J. et al., 2018) , SoccerDB (JIANG, Y. et al., 2020) and SoccerNet–v2 (DELIEGE, A. et al., 2021) are one of the unique and especially dedicated datasets related to game of soccer. With the appearance of such large-scale datasets, the higher-level understanding of the game becomes easy. This was the need for doing automatic editing in broadcasting videos.

SoccerNet (GIANCOLA, S. et al., 2018) was the first benchmark dataset which comprises of 500 games from the leagues of Europe with 6637 actions annotated among the three different action classes such as goals, substitutions and cards.

Later, (YU, J. et al., 2018) came with a novel dataset having 222 videos where they included annotations related to camera shots and position of players. But which task is to be performed on that dataset is still unknown.

Furthermore, SoccerDB (JIANG, Y. et al., 2020) developed a dataset comprising of 346 games. It was merged dataset having a subset of 270 games from SoccerNet and 76 games from Chinese Super League. They proposed different tasks like object detection, temporal localization of action, action recognition and replay segmentation.

Finally, SoccerNet–v2 (DELIEGE, A. et al., 2021) was an extended version of SoccerNet (GIANCOLA, S. et al., 2018). They proposed several novel tasks that would contribute towards the generation of automatic broadcasting of the soccer videos. The number of classes was extended to 17 in order to gain more depth about the fine-grained details of the game. They proposed new innovative tasks namely boundary detection and camera shot segmentation for purpose related to broadcast editing. Along with this, another task named replay grounding was also proposed which could be used for action retrieval and highlight production.

Table 2.3 shows a comparison of classes between the four benchmark datasets.

SoccerNet (GIANCOLA, S. et al., 2018)	(YU, J. et al., 2018)	SoccerDB (JIANG, Y. et al., 2020)	SoccerNet-v2 (DELIEGE, A. et al., 2021)
Goal	overhead kick	background	ball out of play
Cards	solo drive	injured	throw – in
Substitutions	Goal	Saves	indirect free – kick
	Shot	red / yellow cards	foul
	Corner	Shot	Clearance
	free kick	substitution	shots on target
	penalty kick	free kick	shots off target
	red card	corner	Corner
	yellow card	Foul	substitution
	Foul	Goal	kick – off
	Offside	penalty kick	direct free kick
			Offside
			yellow card
			Goal
			Penalty
			red card
			yellow -> red card

Table 2.3: List of action classes among the four benchmark datasets

2.4 Action Spotting

Currently, action spotting is a well-known technique that has been researched widely by many researchers which is very essential for video understanding. It is a task where we have to identify a particular action event within a video.

This term originated in research article (GIANCOLA, S. et al., 2018)which was defined as identifying and localizing an instantaneous event that was anchored with single timestamp wherein five different pooling techniques where used as a part of research findings. In this an event, in particular, represents an action which occurs within a long video under some tolerance (δ) which ranged from 5 to 60 seconds.

In this paper, SoccerNet dataset was used for evaluating the designed models. Three feature extraction approaches such as ResNET (HE, K. et al., 2016), C3D (TRAN, D. et al., 2015) and I3D (CARREIRA, J. and Zisserman, A., 2017) were used to extract features from the football matches. For this purpose the videos were trimmed and sized to 224 x 224 resolution and at each second 25 frames were extracted using these three techniques. For the action spotting task using these baseline approaches they achieve a mean Average Precison (mAP) of 49.7%. This paper serves as the first baseline for analysing soccer broadcasting videos.

But as mentioned in (GIANCOLA, S. and Ghanem, B., 2021), their program showed a weak optimization over the period and this lead to affect the pooling performances. This gave (VANDERPLAETSE, B. and Dupont, S., 2020) an opportunity to enhance the baseline where they combined the audio and visual characteristics of different soccer games. They also carried out two tasks namely activity classification and action spotting in which mAP metric showed an improvement of 7.43% and 4.19% respectively.

Later on, (RONGVED, O.A.N. et al., 2020) used 18 –layered 3D ResNet (Residual Network), directly on videos frames, which is a Convolutional Neural Network (CNN) architecture inspired from (TRAN, D. et al., 2018). This algorithm was trained on a machine which comprise of 512 gigabytes of memory but it performed poor mostly because they trained the encoder from scratch. Then, (VATS, K. et al., 2020) presented a multi – scale CNN that processed information at different temporal scales in order to take care of unpredictability in terms of events spatial and temporal details.

Furthermore, an innovative approach was put forward by (CIOPPA, A. et al., 2020) which was based on loss function that explicitly captured the temporal context of the actions that surrounds them. The most current work of (TOMEI, M. et al., 2021) used a fine tune and easily deployable network for spotting appropriate events in matches. They used regression and masking strategies along with RMS – Net.

Lastly, (GIANCOLA, S. and Ghanem, B., 2021)proposed an innovative pooling method for extracting features which was based on NetVLAD (ARANDJELOVIC, R. et al., 2018) that was modified to NetVLAD++. In this algorithm the feature learning is done based on the context of past and futures frames of time around a specific action.

Overall, a lot of different architectures have been proposed and implemented on different datasets of soccer by various researchers all around the world. Most of them are relating to modify the pooling techniques after extracting the best features from the soccer videos so that one can get handful of understanding about the different events that are occurring during a match. These can help the latest machine learning algorithms to annotate the videos automatically and can benefit in generating highlights and analytics from a competitive game of soccer.

Since, many researchers have applied different pooling mechanism; in this project I will try to study different pooling techniques. Also, will use them in building deep learning models that can identify whether a video frame from a football match video is an action frame or a non – action frame.

The next chapter will brief about the methodology adopted for designing deep neural networks for the project.

Chapter 3: Methodology Adopted

3.1 Introduction

In this chapter will briefly discuss about the different pooling techniques that have been implemented in the developed deep learning model. Also, brief information regarding the basic terminologies of video that are necessary for understanding computer vision will be explained.

In this project, since the hardware requirements were not meet due to which, was obliged to use the available pre – extracted features and labels for the considered dataset. Also, previously the researchers (VANDERPLAETSE, B. and Dupont, S., 2020andGIANCOLA, S. and Ghanem, B., 2021) have used at hand extracted features and labels to later designed their own pooling methodologies and developing new deep learning architectures.

The general pipeline of designing deep neural networks using pooling techniques is as follows:

Step – 1: Extracting features from videos or images.
Step – 2: Assigning annotations for those features.
Step – 3: Applying Pooling Techniques to obtain new reduced feature matrix.
Step – 4: Passing newly obtained reduced feature matrix to the fully-connected layers.
Step – 5: Adding activation functions, dropout and output layers.
Step – 6: Compiling the model with the set parameters.
Step – 7: Train the designed deep neural network model and predict the outcomes.
Step – 8: Evaluate the performance of the model.
Step – 9: Plot necessary graphical visualizations as per the problem domain.

These steps will be discussed in detail in chapter 5 and 6. Since, steps 1 and 2 are not applicable for this project the later sections of this chapter will brief about the pooling techniques.

3.2 Pooling, Padding and Striding[1]

3.2.1 Pooling

In the branch of artificial intelligence, pooling techniques are very common while we are developing models based on machine learning and deep learning. The main purpose of performing a pooling operation is to reduce the overall dimensionality of

[1] Referred (AGGARWAL, C., 2018) and (GÉRON, A., 2019) book chapters.

input feature map. This indeed helps to reduce the computational load, memory utilization and reduction in parameters.

Technically, it is a sliding window that slides over the entire feature map and outputs a completely new shrink in size feature map. Thus, we can consider it to be a pooling layer of size (p, p) which moves over the feature map and is usually smaller in size as compared to the size of feature map, say (M, N), where p < M and p < N. For example, in figure 3.2 the size of the input feature map is (M x N) = 5 x 5 whereas the pooling layer can be of size (p x p) = 2 x 2 or 3 x 3 matrix. Thus, using these two pooling layers on a 5 x 5 feature map will reduce its size to a new 4 x 4 feature map and to a new 2 x 2 feature map respectively.

3.2.2 Padding and Striding

These are the two most common techniques that one uses while performing a pooling operation. As we use pooling to reduce the size of feature map, we might loss some information along the borders of the feature map and that might lead to decrease the accuracy of the designed network. This can be resolved by using the padding technique.

In padding, one can add an extra set of features, which are generally “pixels” in case of a color image, around the borders of the feature map. The values of these features are set to zero by default. There are two terminologies: “valid padding” and “same padding”. Valid padding implies that padding technique is not used. Whenever, we need the output dimension of feature map to be same as the input dimension in those case we use same padding. Figure 3.1 shows an example of “same padding” and figure 3.2 highlights an example of “valid padding”.

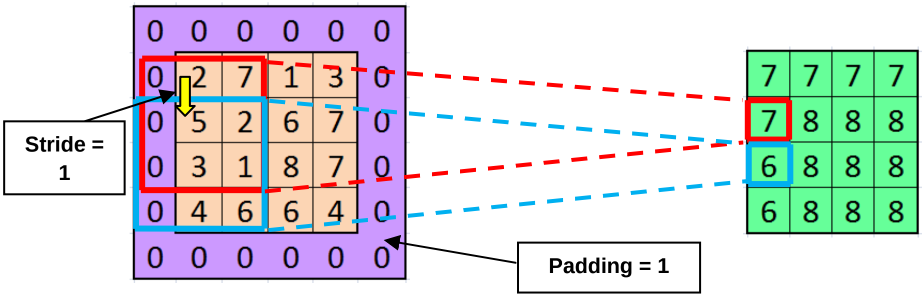


Figure 3.1: Example of Padding and Striding

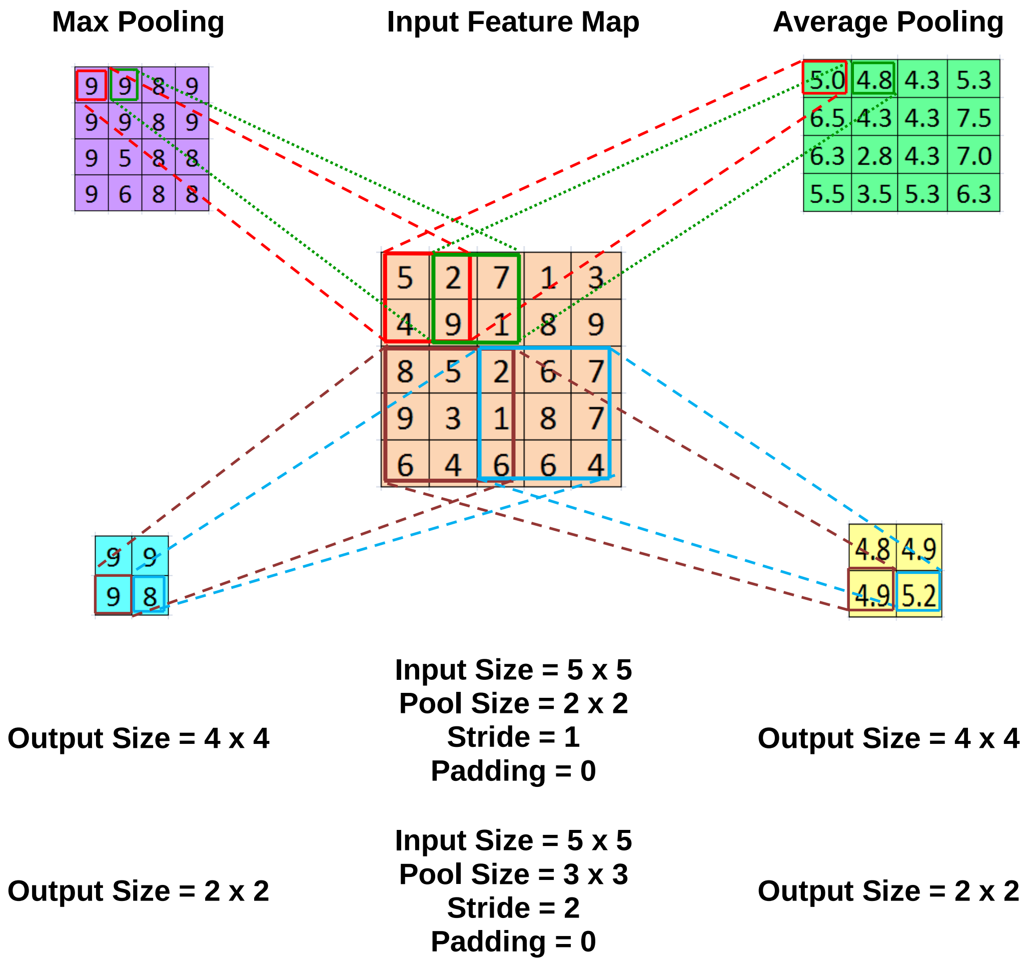
Next important concept is striding. It is basically moving a pooling window with appropriate jumps that slides over the entire feature map in all directions. Figure 3.1 shows an example of stride = 1 whereas in Figure 3.2 an illustration with stride = 2 is demonstrated. Both these techniques of pooling operation help in reduction of spatial position across the feature maps.

Now, that the required terminologies have been explained, for this project four pooling techniques have been selected and applied in the designed deep neural network model. They are as follows:

- Max and Average Pooling
- Mixed Pooling
- NetVLAD Pooling

3.3 Max Pooling and Average Pooling[2]

The two most common pooling techniques which are widely used are Max and Average pooling. In Max pooling, from every patch over the feature map it takes the maximum value among the values present inside that patch whereas in Average pooling, it takes average of the values. Figure 3.2 demonstrates an example of Max and Average pooling techniques.



9	9	8	9
9	9	8	9
9	5	8	8
9	6	8	8

5	2	7	1	3
4	9	1	8	9
8	5	2	6	7
9	3	1	8	7
6	4	6	6	4

5.0	4.8	4.3	5.3
6.5	4.3	4.3	7.5
6.3	2.8	4.3	7.0
5.5	3.5	5.3	6.3

9	9
9	8

4.8	4.9
4.9	5.2

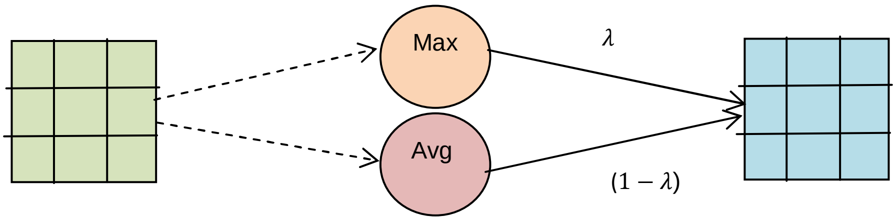
Figure 3.2: Example of Max and Average Pooling

From figure 3.2 it can observe that, if we have a kernel of size 2 x 2 applied on an input image of 5 x 5 then using max(5,2,4,9) we get 9 in the output matrix of size 4 x 4(Refer red box and line). Similarly, we can calculate other elements of the output feature map using either max or average pooling.

[2] Referred (AGGARWAL, C., 2018), (DI, W. et al., 2018) and (GÉRON, A., 2019) book chapters.

3.4 Mixed Pooling

This pooling technique was proposed by (YU, D., Wang, H., Chen, P. and Wei, Z., 2014). It is basically a combination of Max and Average Pooling in a stochastic manner. Figure 3.3 illustrates a pictorial viewpoint of how the operation of mix pooling takes place over a given feature map. Mathematically, it has been proposed by (YU, D., Wang, H., Chen, P. and Wei, Z., 2014).



Feature Map (M x N) Mixed Pooling Operation Output (P)

Figure 3.3: Example of Mixed Pooling Technique

In simple terms, it can be understood as follows:

Let us assume that P is the outcome of mixed pooling operation applied on an input feature map of size, $M\ x\ N$. Also, let M_{max} and Avg be the Max and Average Pooling operation applied on an input feature map. Let, λ be the stochastic variable. Thus, we get equation (3.1),

$$P = \lambda * M_{max} + (1 - \lambda) * Avg\ , where\ 0 \leq \lambda \leq 1. \qquad ----(3.1)$$

3.5 NetVLAD

NetVLAD (Network Vector of Locally Aggregated Descriptors) (ARANDJELOVIC, R. et al., 2018) is a differentiable pooling technique that is inspired from VLAD (JÉGOU, H., Douze, M., Schmid, C. and Pérez, P., 2010). It is a method which is very popular in deep learning for tasks such as retrieval of images and recognition of places. The neural network structure of the NetVLAD is given in figure 3.4.

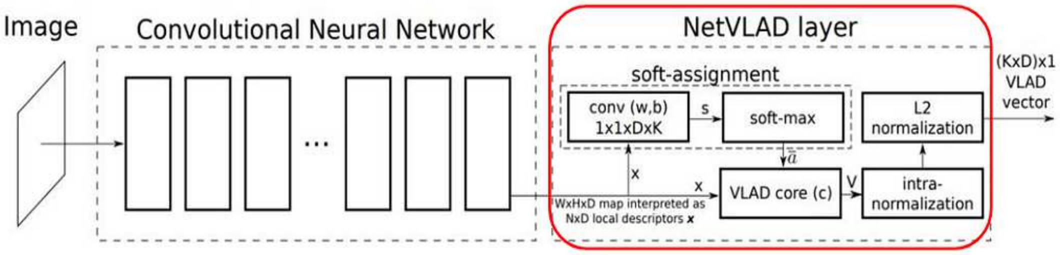


Figure 3.4: Structure of NetVLAD pooling layer (YOONWOOJEONG, 2020)

Since, NetVLAD has learnable parameters it can be easily integrated into other Convolutional Neural Networks (CNN) architectures. Moreover, all the features of NetVLAD are differentiable thus providing an excellent end-to-end methodology when applied on networks.

3.6 Interpretation of Video in computer vision

The purpose of introducing this section is to make everyone familiarize with the terminologies that are used to process a video using a computer. Also, how one should make the computer understand that the input will be a video format that has to be processed and give desired results as per the requirements of the user. Furthermore, the dataset under investigation deals with football matches videos.

A video is nothing but a sequence of consecutive frames in time or it can be describe as an arrangement of moving visuals in time. Each frame in any video can be considered as an individual image and thus if all the successive images are arranged in a particular manner, then it takes the form of a video. Let us understand this by considering an example[3].

Consider a video having at least 10 frames each of size 224 x 224 resolution (the number of pixels contained in each frame) over a time period of 10 seconds. This means that in a video it has a sequence of 10 images whose length and breadth is 224 and there is an image at every second. Moreover, if we consider coloured image then it is a combined of three channels which are red, green and blue. Thus, each frame or image will be of the size 224 x 224 x 3. Therefore, the total size of a 10 second video segment will be 10 x 224 x 224 x 3. Figure 3.5 gives a pictorial explanation in order to gain in-depth idea about how one should think of video from the perspective of computers.

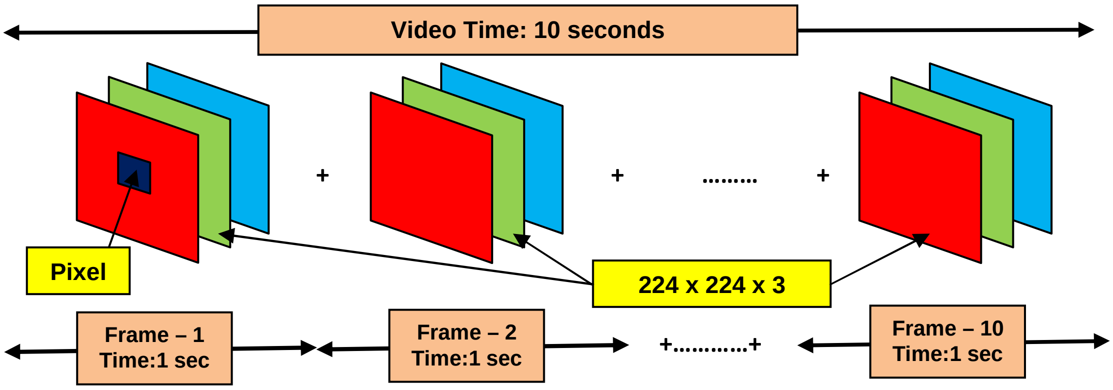


Figure 3.5: Sequence of frames or images in time

[3] Referred (AGGARWAL, C., 2018) section from a book in a chapter.

Chapter 4: Versions of SoccerNet Dataset

In this chapter will discuss about the different SoccerNet dataset versions.

4.1 First Version of SoccerNet

Almost 5 years back, in 2018, the first version of SoccerNet was released by a team of four researchers from the King Abdullah University of Science and Technology (KAUST), Saudi Arabia and till date it is considered as benchmark for action spotting in soccer videos and for developing new deep learning algorithms to gain depth about the different events that takes place during an ongoing game of football.

The dataset contains 500 games which are compiled from six European Championships covering three seasons for the years 2014 to 2017. Every game has two untrimmed videos that correspond to two half-time periods in a football match. The total duration of all these videos is around 764 hours and almost 4TB (Terabyte) of space is consumed for its storage. Almost 6637 actions were temporally annotated among the 3 different actions classes such as goals, red/ yellow cards and substitutions. Table 4.1 summarizes about the number of games in each season and across each league.

League	Seasons			Total
	14 / 15	15 / 16	16 / 17	
EN – EPL	6	49	40	**95**
ES – LaLiga	18	36	63	**117**
FR – Ligue 1	1	3	34	**38**
DE – BundesLiga	8	18	27	**53**
IT – Serie A	11	9	76	**96**
EU – Champions	37	45	19	**101**
Total	**81**	**160**	**259**	**500**

Table 4.1: Summary of 500 games along 3 seasons from six European Championships

The researchers have randomly split the games into 300, 100 and 100 for training, testing and validation purposes while ensuring that there is a similar distribution of events between the classes and datasets. Table 4.2 summarizes about the train, test and valid split among the 6637 events for 3 classes.

Split	Events			Total
	Goals	Cards	Substitutions	
Train	961	1296	1708	**3965**
Valid	356	396	562	**1314**
Test	326	453	579	**1358**
Total	**1643**	**2145**	**2849**	**500**

Table 4.2: Events information among Train, Test and Valid datasets

4.2 Second Version of SoccerNet

After a span of 3 years from their first version of the SoccerNet dataset, in 2021, an extended version "SoccerNet-v2" was released by team of nine researchers that belong to three different universities (KAUST, University of Liège and Aalborg University). The dataset has approximately 300k manually annotations for 17 types of actions that occurred during any football matches for the same 500 games. A list of these actions has been previously mentioned in Table 2.3.For the process of annotation, 33 annotators were hired who were frequent football observers which took them almost 1600 hours of annotations. All these annotations were divided among three categories namely actions, camera shots and replays (DELIEGE, A. et al., 2021).

Therefore, with the release of this dataset two new tasks other than action spotting were introduced such as camera shot segmentation with boundary detection and replay grounding. The scope of understanding of these two tasks is way beyond the chosen topic. Also, they release another dataset which comprises of additional 50 games which were part of challenge in the CVPR 2021 International Challenge on Activity Recognition Workshop. The developed deep learning models were then applied on the challenge dataset and winners of the challenge got amazing prizes sponsored by Sports Analytics Company.

4.3 Third Version of SoccerNet

Recently one year back, in 2022, the latest version SoccerNet-v3, another extended version of SoccerNet dataset has been released by a team of five researchers from two universities (KAUST and University of Liège. The following dataset has 1300k annotations on approximately 34k soccer images. Again, they had hired 83 students from the University of Liège, Belgium to perform the annotation task. It took them almost 3 months and approximately 6000 hours. This makes it to be the largest dataset available for multi – view soccer analysis. This dataset consists of 400 SoccerNet games with train, test and valid splits as 290, 55 and 55 games respectively. With this, new set of tasks and challenges were released in 2023. There are 7 tasks that are split into three broad themes.

Looking at all the versions of SoccerNet dataset, it can be said that it can be exciting and at the same time very challenging to work on such huge datasets.

Even though, multiple tasks are being added in the recent versions of SoccerNet dataset, the pre – extracted features, labels and benchmark codes are made available in each version which one can use to develop their own architecture for providing better results in the field of video understanding.

Chapter 5: Designing and Experiments

5.1 Introduction

In this chapter will discuss about the sub – sample of the dataset that was taken to developed deep learning model. Also, it covers a process that has been carried out in different stages while designing Deep Neural Network model that could detect whether a video frame is an action frame or a non – action frame. Thus, it can identify whether an event is an action event or not.

For this project, the entire code was developed using Python programming language in Kaggle which is a web-based workbench for building machine learning and deep learning models in the field of data – science.

5.2 Dataset Description

For the current project, the sub – sample of the dataset from first version of SoccerNet was utilized. As described in Table 4.1 there are 95 games in total in the English Premier League for the three seasons, I have taken 6 games from the season 2014 / 15. The size of the dataset is approximately 2 GB. The following Figure 5.1 gives an outline of structure of the dataset that has been maintain for all the 6 games.

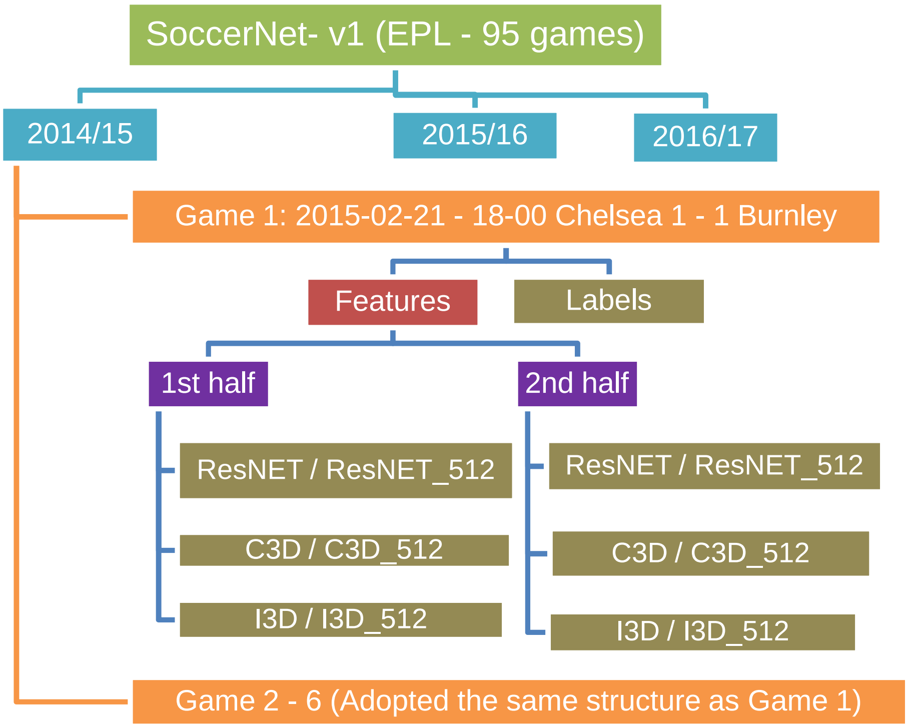
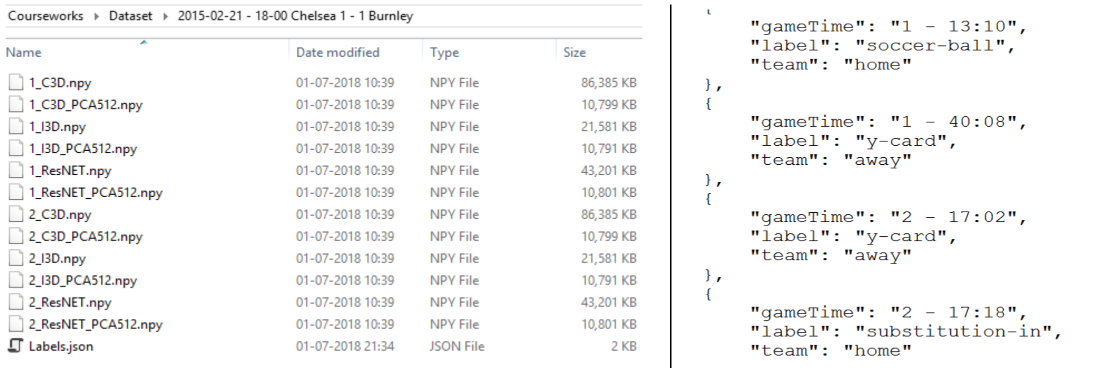


Figure 5.1: Structure of datasets

As it can be seen from figure 5.1, for each game of football that has two halves and therefore each will have pre – extracted features those were extracted using three approaches ResNET, C3D and I3D (refer section 2.4) and only one file of labels.

The features are in .npy format (Numpy) and labels in the form of .json format (Java Script Object Notation).Thus, in total there are 72 features files and 6 files of labels for six football games. Figure 5.2 gives a clear picture regarding features and labels.



Courseworks ▸ Dataset ▸ 2015-02-21 - 18-00 Chelsea 1 - 1 Burnley

Name	Date modified	Type	Size
1_C3D.npy	01-07-2018 10:39	NPY File	86,385 KB
1_C3D_PCA512.npy	01-07-2018 10:39	NPY File	10,799 KB
1_I3D.npy	01-07-2018 10:39	NPY File	21,581 KB
1_I3D_PCA512.npy	01-07-2018 10:39	NPY File	10,791 KB
1_ResNET.npy	01-07-2018 10:39	NPY File	43,201 KB
1_ResNET_PCA512.npy	01-07-2018 10:39	NPY File	10,801 KB
2_C3D.npy	01-07-2018 10:39	NPY File	86,385 KB
2_C3D_PCA512.npy	01-07-2018 10:39	NPY File	10,799 KB
2_I3D.npy	01-07-2018 10:39	NPY File	21,581 KB
2_I3D_PCA512.npy	01-07-2018 10:39	NPY File	10,791 KB
2_ResNET.npy	01-07-2018 10:39	NPY File	43,201 KB
2_ResNET_PCA512.npy	01-07-2018 10:39	NPY File	10,801 KB
Labels.json	01-07-2018 21:34	JSON File	2 KB

```
        "gameTime": "1 - 13:10",
        "label": "soccer-ball",
        "team": "home"
    },
    {
        "gameTime": "1 - 40:08",
        "label": "y-card",
        "team": "away"
    },
    {
        "gameTime": "2 - 17:02",
        "label": "y-card",
        "team": "away"
    },
    {
        "gameTime": "2 - 17:18",
        "label": "substitution-in",
        "team": "home"
```

(a) Game Folder with features(.npy files) and labels(.json file)

(b) Information inside .json file of label.

Figure 5.2: Sample screenshots of game folder and informative content within labels.

5.3 Dataset Exploration

Among the 72 files of features, the files with an extension "***_PCA512" (Figure5.2(a)) were used for developing deep learning models. To understand the elements and dimensions of each feature and label, an extensive investigation was carried out with the help of Python inbuilt functions. The below Tables 5.1, 5.2 and 5.3 will summarize the information that was obtained after executing few lines of code using Python programming language.

ResNET Approach					
Sr. No.	**Game Name**	**Half**	**Rows**	**Columns**	**Labels**
1	2015-02-21 - 18-00 Chelsea 1 - 1 Burnley	1	5400	512	2
	2015-02-21 - 18-00 Chelsea 1 - 1 Burnley	2	5400	512	8
2	2015-02-21 - 18-00 Crystal Palace 1 - 2 Arsenal	1	5534	512	4
	2015-02-21 - 18-00 Crystal Palace 1 - 2 Arsenal	2	5948	512	9
3	2015-02-21 - 18-00 Swansea 2 - 1 Manchester United	1	5450	512	3
	2015-02-21 - 18-00 Swansea 2 - 1 Manchester United	2	5874	512	11
4	2015-02-22 - 19-15 Southampton 0 - 2 Liverpool	1	5400	512	2
	2015-02-22 - 19-15 Southampton 0 - 2 Liverpool	2	5640	512	10
5	2015-05-17 - 18-00 Manchester United 1 - 1 Arsenal	1	5400	512	1
	2015-05-17 - 18-00 Manchester United 1 - 1 Arsenal	2	6086	512	8
6	2015-04-11 - 19-30 Burnley 0 - 1 Arsenal	1	5400	512	3
	2015-04-11 - 19-30 Burnley 0 - 1 Arsenal	2	5846	512	3

Table 5.1: Dimensions of features and labels for ResNET Approach

C3D Approach					
Sr. No.	**Game Name**	**Half**	**Rows**	**Columns**	**Labels**
1	2015-02-21 - 18-00 Chelsea 1 - 1 Burnley	1	5399	512	2
	2015-02-21 - 18-00 Chelsea 1 - 1 Burnley	2	5399	512	8
2	2015-02-21 - 18-00 Crystal Palace 1 - 2 Arsenal	1	5533	512	4
	2015-02-21 - 18-00 Crystal Palace 1 - 2 Arsenal	2	5947	512	9
3	2015-02-21 - 18-00 Swansea 2 - 1 Manchester United	1	5449	512	3
	2015-02-21 - 18-00 Swansea 2 - 1 Manchester United	2	5873	512	11
4	2015-02-22 - 19-15 Southampton 0 - 2 Liverpool	1	5399	512	2
	2015-02-22 - 19-15 Southampton 0 - 2 Liverpool	2	5639	512	10
5	2015-05-17 - 18-00 Manchester United 1 - 1 Arsenal	1	5399	512	1
	2015-05-17 - 18-00 Manchester United 1 - 1 Arsenal	2	6085	512	8
6	2015-04-11 - 19-30 Burnley 0 - 1 Arsenal	1	5399	512	3
	2015-04-11 - 19-30 Burnley 0 - 1 Arsenal	2	5845	512	3

Table 5.2: Dimensions of features and labels for C3D Approach

I3D Approach					
Sr. No.	**Game Name**	**Half**	**Rows**	**Columns**	**Labels**
1	2015-02-21 - 18-00 Chelsea 1 - 1 Burnley	1	5395	512	2
	2015-02-21 - 18-00 Chelsea 1 - 1 Burnley	2	5395	512	8
2	2015-02-21 - 18-00 Crystal Palace 1 - 2 Arsenal	1	5529	512	4
	2015-02-21 - 18-00 Crystal Palace 1 - 2 Arsenal	2	5943	512	9
3	2015-02-21 - 18-00 Swansea 2 - 1 Manchester United	1	5445	512	3
	2015-02-21 - 18-00 Swansea 2 - 1 Manchester United	2	5869	512	11
4	2015-02-22 - 19-15 Southampton 0 - 2 Liverpool	1	5395	512	2
	2015-02-22 - 19-15 Southampton 0 - 2 Liverpool	2	5635	512	10
5	2015-05-17 - 18-00 Manchester United 1 - 1 Arsenal	1	5395	512	1
	2015-05-17 - 18-00 Manchester United 1 - 1 Arsenal	2	6081	512	8
6	2015-04-11 - 19-30 Burnley 0 - 1 Arsenal	1	5395	512	3
	2015-04-11 - 19-30 Burnley 0 - 1 Arsenal	2	5841	512	3

Table 5.3: Dimensions of features and labels for I3D Approach

From the summarized tables it can be observed that for each game the number of rows varies while the columns size remains unchanged. The reason behind this is while performing feature extraction three separate approaches were used and since each approach has a distinct algorithm for extracting the features from the videos of the football matches the number of rows as well as column differs initially.

Afterwards, a technique named “Principal Component Analysis (PCA)” was applied to each obtained matrices, due to which the number of columns had reduced to 512 principal components. Thus, we get same number of columns for each match, but the rows differ because eventually rows correspond to frames in a video.
Moreover, from the labels column it can be observed that more actions take place in the 2nd half of the match rather than in 1st half since an outcome in the form of win, loss or tie should occurred for an on-going football match.

5.4 Data Interpretation and Preprocessing

From the previous section 5.3 and from Figure 5.2(a), there are separate files for features and labels. Thus, to develop deep learning model, we need to combine the features and labels so that it can form a dataset which can be then further divided into train and test datasets that can be passed through the model. For this purpose, a lot of preprocessing steps are required to be executed and hence there is a need for using the inbuilt libraries and modules which are available in python language (refer table 5.5).

To combine features and labels, let us first understand how one should interpret the labels and features files. Also, let us assume that ideally a football match has duration of 90 minutes where each half is further divided into 45 minutes each. As mentioned, that rows of the matrices correspond to frames in a video. These frames were extracted at a frame rate of 2 frames per second using the stated three approaches. Thus, in 60 seconds it will have 120 frames and eventually it will have 5400 frames over duration of 45 minutes. This number can be seen in Table 5.1 in the form of rows. Also, as stated that each pre – extracted feature algorithm is different thus it will have slight variations in terms of frames.

Now figure 5.2(b) gives a snapshot of what kind of information is stored inside the labels file. It is stored in a json(JavaScript Object Notation) format. It also called as key – value format. For example, “gametime”, “label” and “team” corresponds to keys and “1 – 13:10”, “soccer-ball” and “home” represents the values for those keys. Hence, this information can be interpreted as “A goal was scored by the home team in the first half at time 13:10 minutes”. This represents a particular action that has occurred during a football match.

Let us now connect this action with the frames. Since, the action has occurred during a particular time, there is need to identify exactly which frame corresponds to an action.

So, the time from the label can be used to refer to a particular frame. For example, if the event has occurred at 13:10 minutes then the corresponding frame is 1580 out of the 5400 frames. The mathematical calculation is shown in Table 5.4.

1 min – 120 frames 13 min - ? = 13 x 120 = 1560 frames	1 sec – 2 frames 10 sec - ? = 10 x 2 = 20 frames
Therefore, 1560 + 20 = 1580 frames (if the time is 13:10 minutes)	

Table 5.4: Identify the frame from the given time.

Finally, in this way time and frame should be interpreted. Thus, from the table 5.1, among 5400 frames there are only 2 frames in which an action has occurred in the first half and for the second half there are only 8 frames out of 5400 in which an action takes place for the very first game of football. Hence, one can now understand how rows are related to frames and labels are related to actions.

As 6 games were considered this eventually means that the dataset will have 6 game folders that will have some features and labels files. Therefore, from the viewpoint of preprocessing, several functions were developed that can read and extract the information from the files of features and labels to smooth the overall process of preprocessing.

Table 5.5 shows a summarized list of libraries that were used along with their purposes.

Libraries Used	**Purpose**
Os	For navigating through different folders which are present in the operating system.
Json	For handling json files (In this project, Labels are in json format)
Numpy	For performing numerical computations
tenserflow, keras	For developing deep learning models
matplotlib.pyplot, seaborn	For visualization
sklearn.metrics	For measuring the accuracy of the models

Table 5.5: Libraries used along with their purpose.

5.5 Generation of Train, Test, Valid datasets

By this time, a clear understanding must have been developed about the dataset structure and how features are related to the label. Now, the next step is to create train, test and valid datasets. Among the 6 games mention in Table 5.1 – 5.3, the first 4 games were considered for training the deep learning model while 5th and 6th game was considered for testing and validating the deep learning model respectively.

For this dataset we have 3 classes of action as mentioned in section 4.1 which are goals, yellow / red cards and substitutions. Also, it has been briefed in section 5.4 about relating an action with a frame.

Our aim and objective are to detect whether in a football match an action has occurred or not by developing a deep learning model. Since, the features are in the form of a numpy matrix, the labels those are in json format had to be converted into a column matrix that would be added at the end of the features matrix. This will form a dataset having features and targets (action / non - action labels) which is needed to build the desired deep learning model.

For this project, if in a frame an action occurs then it has been re-labelled as 1 otherwise 0 if the action does not take place. Thus, for annotations having label values with actions such as “soccer-ball”, “y-card”, “r-card” and “substitution-in” as can be seen in Figure 5.2(b), the labels have been reassigned to 1 and for all other frame where no information is available in terms of time where one cannot relate it to any frame has been labeled as 0(assumed as backgrounds frames).Some functions were created using python programming which can convert the labels to 0 and 1 respectively giving an output as a column matrix.

Now, we can easily combine features matrix with label column matrix. For all the three approaches, Table 5.6 displays the dimensions of Train, Test and Valid games datasets along with the number of labels in each dataset for all the 3 approaches.

Matrix	ResNET Approach			C3D Approach			I3D Approach		
	Rows	Columns	Labels	Rows	Columns	Labels	Rows	Columns	Labels
Train_game	44646	513	49	44638	513	49	44606	513	49
Test_game	11486	513	9	11484	513	9	11476	513	9
Valid_game	11246	513	6	11244	513	6	11236	513	6

Table 5.6: Train, Test and Valid games datasets for 3 approaches.

Overall, the splitting percentages of Train, Test and Valid datasets for all the three approaches are 66%, 17% and 17% respectively.

Moreover, it can be observed from Table 5.6 that the dimension of column has been changed from 512 to 513. The last column indicates a column matrix with values either 0 (non – action) or 1(action).

The next important observation from Table 5.6 is that there are just 49, 9 and 6 actions taking place among the 44646, 11486 and 11246 frames for ResNET approach. This implies that the number of counts of 1 will be fewer as compared to the counts of 0. The same can be seen in case of C3D and I3D approaches. This leads us to some limitations of the dataset which are discussed further.

5.6 Limitations of datasets

For the dataset under investigation, the following three limitations have been identified. They are as follows:

Data Imbalanced: Data Imbalance is a situation that is greatly observed in deep learning problems. It is a situation where the distribution of the target values is skewed towards one of the target values. In this case, target values are 0 and 1. Also, as stated in section 5.5 there are more 0’s than 1’s. For example, in Train game dataset for ResNET approach there are just 49 frames of 1’s (action) while 44597 frames of 0’s (no-action) which is almost 900 times as compared to 1’s. Hence, the dataset is skewed towards 0’s.

Biasness: It is a phenomenon that leans the outcomes of the algorithm towards the most favor ones. In this case it might lean towards 0’s since they are more in number as compared to 1’s.

Overfitting: It is that behavior of machine learning which is undesirable. The network model gets train in such a way that it gives accurate predictions for train data but not for the test data. In such cases, the train error is small but the validation error is high.

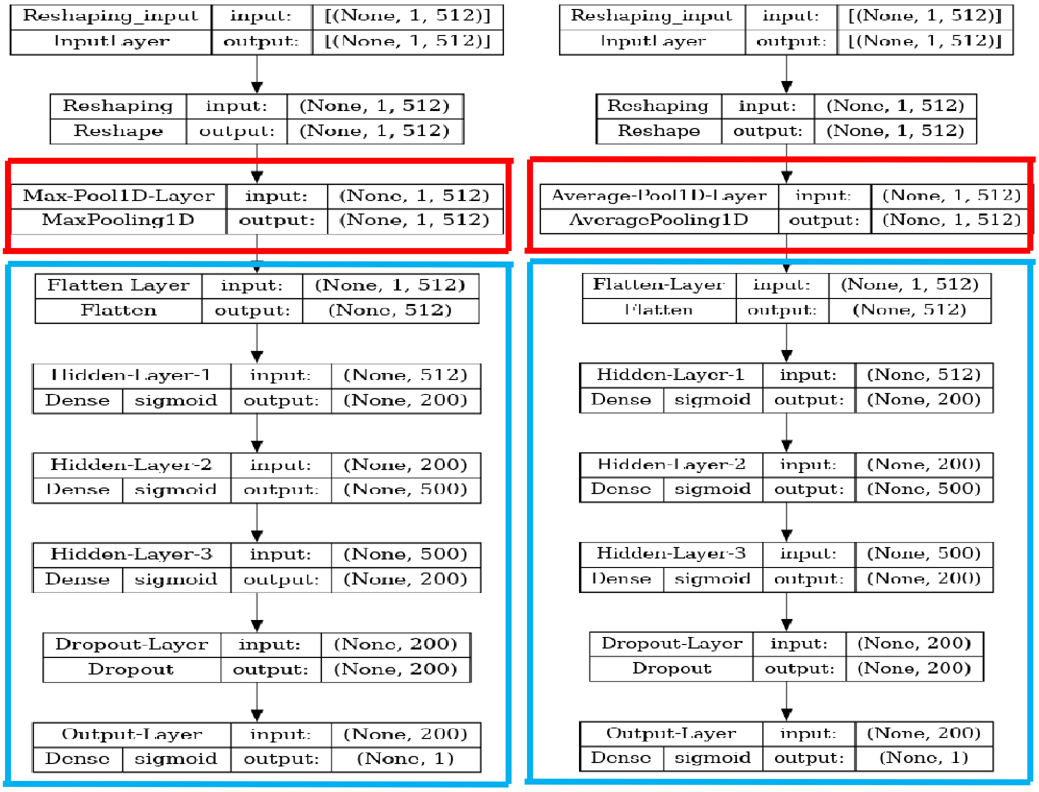
Hence, the dataset is highly imbalanced in terms of target values, and this might make the model bias during the training of the model which later can negatively impact the overall performance of the designed model.

5.7 Architecture of Deep Learning Models

This section will brief about the architecture of the deep neural network that was adopted for generating the required outcome which is to identify whether a particular frame is categorized as an action frame or non – action frame.

It has already been revealed that for this project the pre – extracted features from 3 approaches (ResNET, C3D and I3D) have been considered and in Chapter 3 in total 4 pooling techniques have been discussed. Thus, in total 4 (pooling techniques) * 3 (approaches) = 12 deep learning models have been developed and later compared to analyze which model gives the best outcome for the considered dataset. The architecture of all the models is shown in the figures 5.3 and 5.4.

ResNET, C3D and I3D Approaches

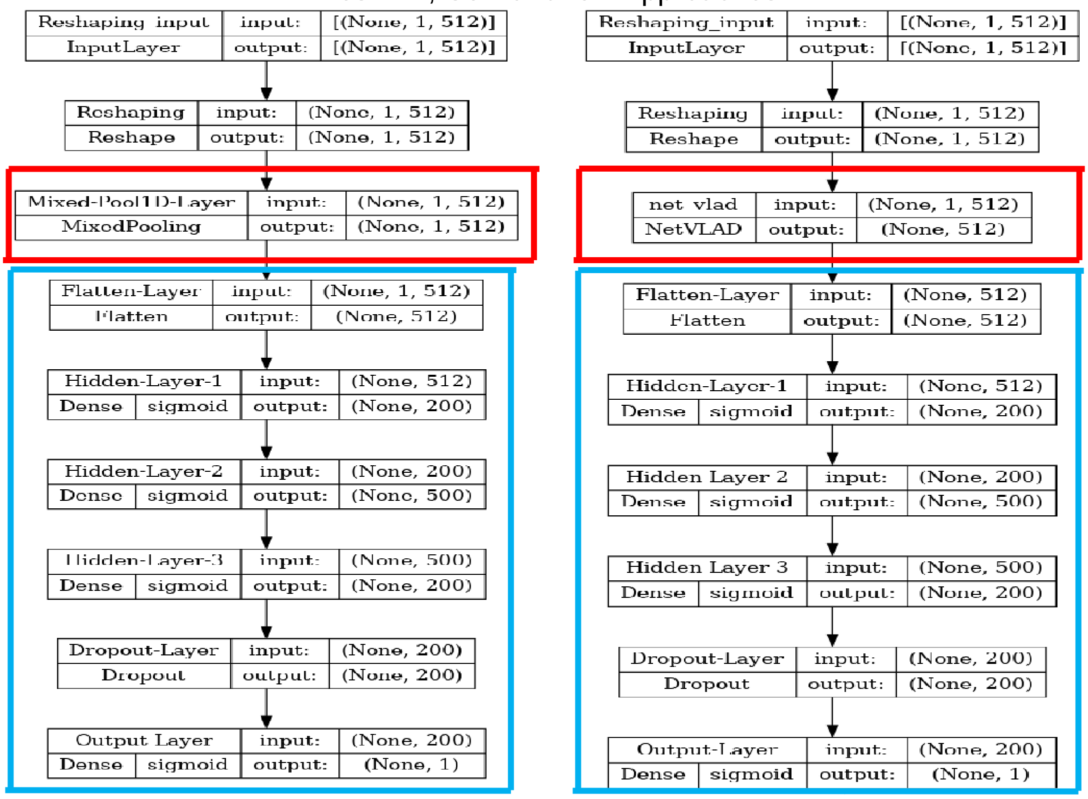

Reshaping_input | input: | [(None, 1, 512)]
InputLayer | output: | [(None, 1, 512)]
Reshaping | input: | (None, 1, 512)
Reshape | output: | (None, 1, 512)
Max-Pool1D-Layer | input: | (None, 1, 512)
MaxPooling1D | output: | (None, 1, 512)
Flatten Layer | input: | (None, 1, 512)
Flatten | output: | (None, 512)
Hidden-Layer-1 | input: | (None, 512)
Dense | sigmoid | output: | (None, 200)
Hidden-Layer-2 | input: | (None, 200)
Dense | sigmoid | output: | (None, 500)
Hidden-Layer-3 | input: | (None, 500)
Dense | sigmoid | output: | (None, 200)
Dropout-Layer | input: | (None, 200)
Dropout | output: | (None, 200)
Output-Layer | input: | (None, 200)
Dense | sigmoid | output: | (None, 1)
Reshaping_input | input: | [(None, 1, 512)]
InputLayer | output: | [(None, 1, 512)]
Reshaping | input: | (None, 1, 512)
Reshape | output: | (None, 1, 512)
Average-Pool1D-Layer | input: | (None, 1, 512)
AveragePooling1D | output: | (None, 1, 512)
Flatten-Layer | input: | (None, 1, 512)
Flatten | output: | (None, 512)
Hidden-Layer-1 | input: | (None, 512)
Dense | sigmoid | output: | (None, 200)
Hidden-Layer-2 | input: | (None, 200)
Dense | sigmoid | output: | (None, 500)
Hidden-Layer-3 | input: | (None, 500)
Dense | sigmoid | output: | (None, 200)
Dropout-Layer | input: | (None, 200)
Dropout | output: | (None, 200)
Output-Layer | input: | (None, 200)
Dense | sigmoid | output: | (None, 1)

(a) Max Pooling (b) Average Pooling

Figure 5.3: Architecture of ResNET, C3D and I3D models having Max and Average Pooling Layers

It can be seen from Figure 5.3 and Figure 5.4 that only the pooling layers for all the 12 model architecture approaches are changing while the rest of the structure remains as it is. The pooling layers are indicated in red boxes while blue boxes indicate a fully connected layer. It has 3 hidden layers having 200, 500 and 200 neurons in it. A dropout layer with 60% probability has been added to avoid the problem of overfitting and biasness. Finally, an output layer is constructed which determines whether a video frame is an action frame or a non-action frame.

ResNET, C3D and I3D Approaches


Reshaping input | input: | [(None, 1, 512)]
InputLayer | output: | [(None, 1, 512)]
Reshaping | input: | (None, 1, 512)
Reshape | output: | (None, 1, 512)
Mixed-Pool1D-Layer | input: | (None, 1, 512)
MixedPooling | output: | (None, 1, 512)
Flatten-Layer | input: | (None, 1, 512)
Flatten | output: | (None, 512)
Hidden-Layer-1 | input: | (None, 512)
Dense | sigmoid | output: | (None, 200)
Hidden-Layer-2 | input: | (None, 200)
Dense | sigmoid | output: | (None, 500)
Hidden-Layer-3 | input: | (None, 500)
Dense | sigmoid | output: | (None, 200)
Dropout-Layer | input: | (None, 200)
Dropout | output: | (None, 200)
Output Layer | input: | (None, 200)
Dense | sigmoid | output: | (None, 1)
Reshaping_input | input: | [(None, 1, 512)]
InputLayer | output: | [(None, 1, 512)]
Reshaping | input: | (None, 1, 512)
Reshape | output: | (None, 1, 512)
net vlad | input: | (None, 1, 512)
NetVLAD | output: | (None, 512)
Flatten-Layer | input: | (None, 512)
Flatten | output: | (None, 512)
Hidden-Layer-1 | input: | (None, 512)
Dense | sigmoid | output: | (None, 200)
Hidden Layer 2 | input: | (None, 200)
Dense | sigmoid | output: | (None, 500)
Hidden Layer 3 | input: | (None, 500)
Dense | sigmoid | output: | (None, 200)
Dropout-Layer | input: | (None, 200)
Dropout | output: | (None, 200)
Output-Layer | input: | (None, 200)
Dense | sigmoid | output: | (None, 1)

(a) Mixed Pooling (b) NetVLAD pooling

Figure 5.4: Architecture of ResNET, C3D and I3D models having Mixed and NetVLAD Pooling Layers

Apart from the parameters mentioned, at each hidden layer and output layer the activation function used was sigmoid. A kernel regularizer has been added in the 1st and 3rd hidden layer having penalty 0.01 to prevent overfitting. The total number of trainable parameters (weights) for 6 models having Max and Average pooling layers are 303,501 while next 3 models with Mixed pooling layers are 303,502. The remaining 3 models having NetVLAD pooling layer has 17,146,317 trainable parameters. The detail parameters of each pooling layer are given in Tables 5.7(a) and 5.7(b).

Pooling Layer	**Pool Size**	**Strides**	**Padding**	**Alpha**
Max	1 x 2	1	same	NA*
Average				NA*
Mixed				0.7

(a) Max, Average and Mixed Pooling parameters(*Not Applicable)

Pooling Layer	**Parameters**
NetVLAD	feature size = 512 output dimension = 512 max samples = 1 cluster size = 64 gating = False batch normalization = False is_training = False

(b) NetVLAD pooling parameters

Table 5.7: Parameters that were considered in each pooling layer techniques.

To optimize the trainable parameters “Adam” optimizer was considered along with a binary loss function since the target values has only two outcomes 0(non – action) and 1(action).

All the 12 models were trained for 25 epochs with batch size 120 using an accelerator GPU T4 x 2 in Kaggle Notebook. To overcome the problem of data imbalanced, class weights = {0:1,1:892} were provided. During the training of the model the validation was done using the valid dataset. Once the model was trained then it was evaluated using the test dataset. Finally, new labels were predicted using the features of test data sets.

Since, we obtained the predicted labels from the designed model we want to understand the reliability and efficiency of the model. For this purpose, the next section will elaborate upon testing the performance of the model using different evaluation metrics.

5.8 Evaluation Metrics

In this section will brief about the different metrics that are used to check the performance of the model as well as it helps in deciding how the model behaves. Also, if through analysis is carried out then it can help in checking whether the model is reliable and can it be used in real world application.

Since, our objective of the project is to identify whether a particular video frame is actionable or non – actionable, this kind of problem are referred to as classification problems in the field of machine learning. Hence, most of the metrics that will be discussed will fall under the category of classification metrics from the sklearn library in python.

For this project, in total 7 evaluation metrics[4] have been implemented to check the model performance from all aspects of deep learning. They are as follows:

Confusion matrix: A confusion matrix is a matrix form representation which summarizes all the predictions for a list of available labels. For example, if in a model there are two classes of labels, Class A and Class B. Then, the general idea is to count the number of times instances of Class A that are being classified as Class B.

[4] Referred (GÉRON, A., 2019) book chapter.

Thus, it helps in understanding that the designed model is being confused and has classified Class A as other Class B. Hence, it reveals how many predictions are classified as correct and incorrect per class. Figure 5.5 shows a confusion matrix that has two classes which represents a binary classification problem.

		Predicted Class	
		Class A	**Class B**
Actual Class	**Class A**	True Positives (TP)	False Negatives (FN)
	Class B	False Positives (FP)	True Negatives (TN)

Figure 5.5: Confusion matrix

Here, each row represents an actual class while the column indicates predicted class.

Let us consider a simple example as shown in Figure 5.6 to understand how to interpret the information generated by confusion matrix.

		Predicted Class	
		0	**1**
Actual Class	**0**	7933	3544
	1	3	6

Figure 5.6: Example of Confusion matrix

		Predicted Class	
		0	**1**
Actual Class	**0**	11477	0
	1	0	9

Figure 5.7: Perfect Confusion matrix

From Figure 5.6, it can be observed that there are two classes 0's and 1's.The first row of this matrix considers 0's (Positives class): 7933 instances of 0's was predicted correctly as 0's (they are known as **True Positives**) while the remaining 3544 instances were wrongly predicted as 1's (**False Negatives**).The second row of this matrix considers 1's (**Negatives class**): 3 instances of 1's was predicted wrongly as 0's (known as **False positives**) while the remaining 6 instances were correctly predicted as 1's (**True Negatives**).

A confusion matrix that has only values at True Positives and True Negatives and values at False Positives = False Negatives = 0, will mean that the model is well – trained and can perfectly identify both the labels, which has been illustrated in Figure 5.7.

Precision: Precision is defined as a ratio that is given by equation (5.1).

$$Precision = \frac{TP}{TP + FP} \qquad ---- \ (5.1)$$

Precision refers to the rate of positive predictions which means among the positively predicted values how many were exactly true.

Recall: Recall is defined as a ratio which is given by equation (5.2).

$$Recall = \frac{TP}{TP + FN} \quad ---- \ (5.2)$$

Recall refers to the rate of actual observations which means among the total actual positive values how many were predicted correctly.

F1 score: Often it is difficult to compare two models having low recall and high precision or vice versa. To solve this problem, one can use F1 score. F1 score is a combination of Precision and Recall. It is a harmonic mean of precision and recall given by equation (5.3).

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad ---- \ (5.3)$$

Accuracy: If one wants to measure how often the model makes a correct prediction when put into effect then we need to know the accuracy of the model. It is also a ratio given by equation (5.4).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad ---- \ (5.4)$$

Classification report: It is a summarized report that has precision, recall and F1 score for each class label and displays the accuracy score.

ROC Curve: A ROC curve (receiver operating characteristic curve) is a graphical representation that indicates the performance of a designed model at different threshold levels. It plots two parameters, True positive rate (TPR) and False positive rate (FPR). True positive rate is another name of the measure recall. The formula of False positive rate is given by equation (5.5).

$$False\ Positive\ rate = \frac{FP}{FP + TN} \quad ---- \ (5.5)$$

Area Under the Curve (AUC): It calculates the entire 2-dimensional area that is beneath the ROC curve starting from (0,0) to (1,1) using the trapezoidal rule.
Figure 5.8 demonstrates a clear idea about both ROC curve and AUC.

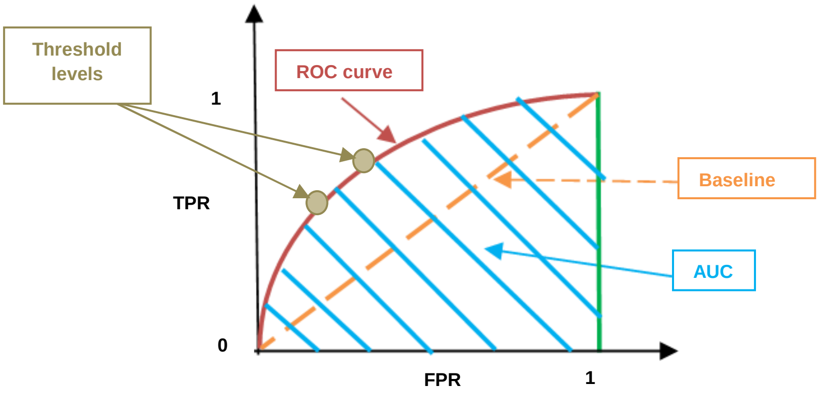


Figure 5.8: ROC and AUC diagram

Thus, it can be seen that a lot of performance metrics can be used to evaluate the performance of the model.

The next chapter will discuss about the outcomes that were obtained after training all the 12 models successfully.

Chapter 6: Results and Discussions

6.1 Introduction

This chapter will highlight the outcomes those were obtained after successfully training all the 12 models. Also, in-total 5 experiments per model (training each model 5 times => 12 models* 5 times = 60 experiments) were carried out to deeply understand which model outperforms among the designed 12 deep neural network models. The discussion of the results will be based on evaluation metrics those were discussed in section 5.8. Furthermore, what inference one can establish from the different graphical visualizations such as accuracy curves, loss curves and ROC curves that will be elaborated in detail.

6.2 Accuracy and Loss Curves

During the training process of a deep learning neural network model, there are many important input parameters and factors that one should take care of such as loss function, accuracy metrics, optimizers, epochs, batch size, etc. Among those the most important thing is to check the behavior of Accuracy and Loss curves as the training progresses for the given number of epochs. The accuracy of the model should increase exponentially while the loss should exponentially decrease. This ensures that the training of the model is being carried out appropriately.

Figure 6.1 is outcome of accuracy and loss curves those were obtained during the training using ResNET features with max pooling techniques were as figure 6.2 is of ResNET features with NetVLAD pooling techniques[5].

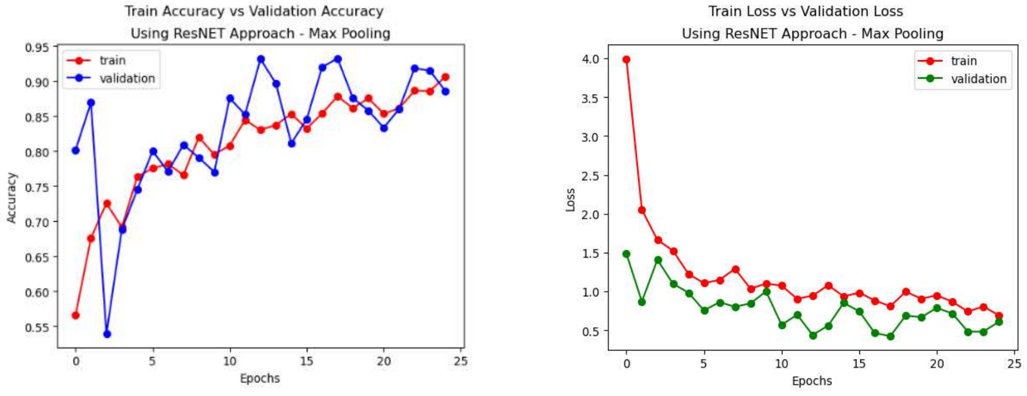


Figure 6.1: Accuracy and Loss curves using ResNET features and Max Pooling

From these two figures it can be visualized that there is an exponential behavior among the different models. There are some variations among few models[1], but still, they do follow this kind of patterns during the training of those models.

[5] All the resulted graphs can be found in the auxiliary files in NOW Dropbox.

In these two figures, it can be noticed that during training of a model (red-line) it is having the same characteristics as we have for exponential curves. On the other hand, while validating the model (blue / green lines) there are some spikes in between where it slightly deviates from its actual mannerism. Since, as explained in section 5.5 there are very few labels with 1's (action) as compared to 0's (non – action) in all the datasets.

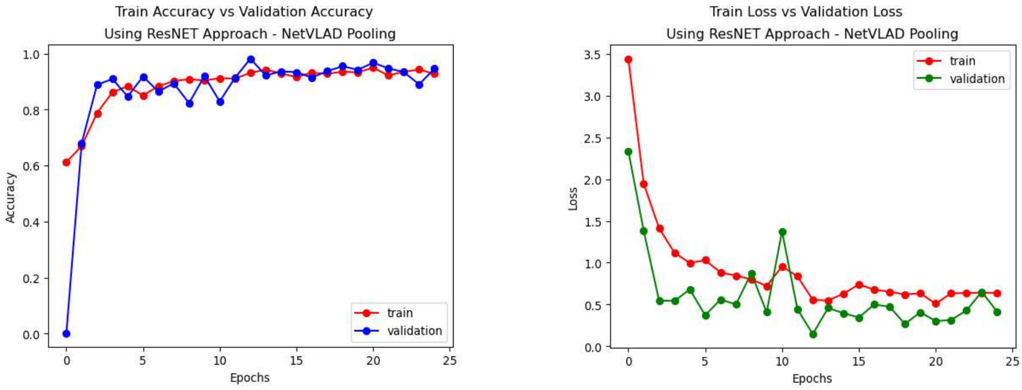


Figure 6.2: Accuracy and Loss curves using ResNET features and NetVLAD Pooling

From these graphs, it is quite difficult to judge which model among the 12 models is the best for classifying a frame as an action frame or not. Hence, we need to use some evaluation metrics that can help us in finding which model has surpass others in terms of performance.

6.3 Based on Confusion matrix

As explained in section 5.8, that the confusion matrix tries to give an idea about among the predicted labels how many of them are correctly classified and which among them are labeled incorrectly.

Tables 6.1 and 6.2 are the confusion matrices[6] that were generated after using the respective trained models for predicting the labels for the test dataset.

Features: ResNET **Pooling Layer: NetVLAD**				
Experiments	**Confusion Matrix**			
	True Positive (TP)	**False Positive (FP)**	**False Negative (FN)**	**True Negative (TN)**
1	10713	7	764	2
2	10630	8	847	1
3	11260	9	217	0
4	10222	4	1255	5
5	10824	7	653	2

Table 6.1:Confusion matrix obtained using ResNET features and NetVLAD pooling.

[6] Refer Appendix to see all the outcomes obtained after 5 experiments.

Features: I3D **Pooling Layer: MixedPooling1D**				
Experiments	**Confusion Matrix**			
	True Positive (TP)	**False Positive (FP)**	**False Negative (FN)**	**True Negative(TN)**
1	11416	9	51	0
2	10543	3	924	6
3	11381	9	86	0
4	9629	3	1838	6
5	11314	9	153	0

Table 6.2: Confusion matrix obtained using I3D features and Mixed pooling.

Recall that the project wishes to identify whether a particular frame is an action frame or not. Thus, label 0 (positives) represents that the frame is not having an action and 1 (negatives) means it is an action frame.

Tables 6.1 and 6.2 are in total 10 confusion matrices since each model were trained 5 times. Hence, from table 6.2, out of the 5 experiments only for 2 experiments it can classify 6 frames as action frames while remaining frames as non – action frames. In the remaining 3 experiments the model got trained in such a way that all the labels tend towards 0's and thus were not able to classify at least 1 frame as an action frame among the 9 action frames present in the test dataset.

Thus, among the two tables 6.1 and 6.2, from the values of confusion matrix in table 6.1 obtained using ResNET features and NetVLAD pooling, after performing 5 successful experiments, performs better than the results shown in Table 6.2. Similar kind of interpretation can be done for all the developed deep learning models.

Hence, again from this evaluation metric it is quite hard to determine which model is the best model for detecting the actions correctly from the game of football. So, the next section will brief about the remaining performance measuring metrics and will try to identify the best model.

6.4 Based on Precision, Recall, F1 score and Accuracy

The basic definitions of these metrics are explained in section 5.8. These four evaluation metrics are the most crucial performance measures that one can use to make a decision about the performance of model.

Tables 6.3, 6.4 and 6.5 are the summarized results obtained after running each model 5 times and then performing an average over each of precision, recall, f1-score and accuracy for respective model over each feature.

		ResNET			
		Precision	**Recall**	**F1 Score**	**Accuracy**
Pooling Technique	**Max**	0.50108	0.6785	0.4605	0.84568
	Average	0.50094	0.672664	0.44938	0.8118
	Mixed	0.50132	0.73238	0.45308	0.82054
	NetVLAD	0.50072	0.57852	0.48502	0.93428

Table 6.3: Results obtained using ResNET Features

		C3D			
		Precision	**Recall**	**F1 Score**	**Accuracy**
Pooling Technique	**Max**	0.50098	0.7534	0.42314	0.72906
	Average	0.50068	0.66828	0.43268	0.75872
	Mixed	0.50088	0.65728	0.4542	0.82546
	NetVLAD	0.50096	0.62432	0.46724	0.87048

Table 6.4: Results obtained using C3D Features

		I3D			
		Precision	**Recall**	**F1 Score**	**Accuracy**
Pooling Technique	**Max**	0.50396	0.69274	0.4987	0.9629
	Average	0.50244	0.63296	0.48714	0.9325
	Mixed	0.50064	0.60666	0.48756	0.94618
	NetVLAD	0.50344	0.58092	0.5026	0.98348

Table 6.5: Results obtained using I3D Features

From these three tables, it can be observed that for all the 12 models the precision is approximately 51%. It means that, if the model predicts that there is an action frame then it is correct 51% of the time. This implies that the quality of the build models is not good. It still needs further improvement.

On the other hand, there are different recall scores for all the 12 models, and they are higher than precision scores. Whenever, in the developed model if recall is higher than the precision it means that there are many predicted labels which are classified incorrectly as compared to the actual labels.

Since we are in a situation where there is low precision and high recall, as explained in section 5.8, F1 – score will be helpful in determining which model is best among these 12 models. It can be observed that the F1 – score for two models is higher compared to others. Those two are deep learning models build using ResNET and I3D features with NetVLAD pooling technique having F1 – score of 0.4850 and 0.5026 respectively. These models were selected not merely based on F1 – score but also their precision and recall scores where much closer to each other than the remaining 10 models.

Among these 10 remaining models, two models did not perform well which are build using ResNET features with Mixed pooling and using C3D features with Max pooling. It can be clearly seen from the Table 6.3 and 6.4 that there are huge differences in their precision and recall scores. Every other model had an average perform.

The accuracy that is reported in all the three tables is evaluated on the test dataset in which there were only 9 action labels and rest were all non – action labels. Since, accuracy is a measure in which only true positives and true negatives are considered in numerator according to equation 5.4 in section 5.8, in an imbalanced dataset like this, it may tend to show biasness. This can be seen in Table 6.2, for the experiments 1, 3 and 5, values tend towards only true positives and all the true negatives are having zero value. This shows that the model was biased towards 0's labels. Hence, this evaluation metric is also not being considered for deciding best model.

6.5 Based on ROC and AUC score

These two metrics can be utilized to decide the performance of the model based on graphical visualization. Figures 6.3 and 6.4 are the outcomes of ROC curves along with AUC scores obtained from two such models[7].

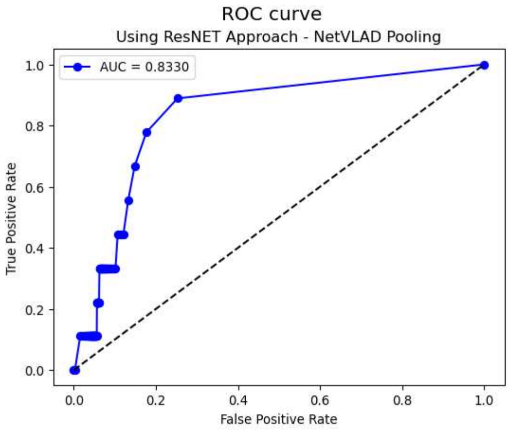


Figure 6.3: ROC curve using ResNET features and NetVLAD pooling

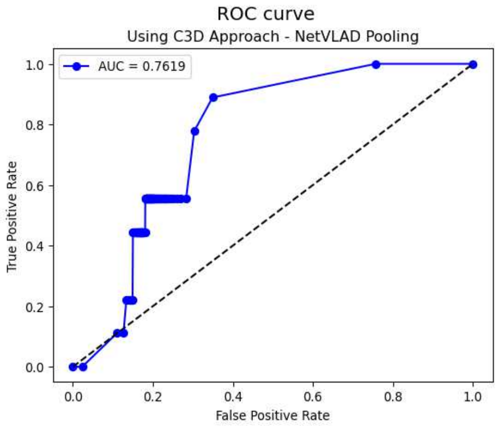


Figure 6.4: ROC curve using C3D features and NetVLAD pooling

[7] All the resulted graphs can be found in the auxiliary files in NOW Dropbox.

Figures 6.3 and 6.4 shows all the thresholds that were obtained while plotting ROC curves. Along with that, AUC score was also calculated and shown in the graph. In figure 6.4, there are few thresholds that are below the baseline which is not a good sign. If all the thresholds are above the baseline as in figure 6.3, then it is said to have a good nature of ROC curve. But again, it is difficult to analyse all the graphs obtained from 12 models with our naked eyes.

As AUC score are calculated, the average of those scores over 5 experiments for each model are summarized in table 6.6.

		AUC Score		
		ResNET	**C3D**	**I3D**
Pooling Technique	**Max**	0.76734	0.8408	0.83566
	Average	0.76288	0.80874	0.79692
	Mixed	0.78432	0.82428	0.79804
	NetVLAD	0.76998	0.7087	0.75196

Table 6.6: Average AUC score for all the models

There are some misleading values in Table 6.6 since, AUC measures the area under the curve irrespective of what threshold is selected. Thus, from figures 6.3 and 6.4 if there are more thresholds nearer to 1 in such a way it forms a plateau then the value of AUC curve will be much higher than the once having a curve of the nature as in figure 5.8 from section 5.8. Hence, this metric also cannot be used to decide the best model.

Overall, it can be concluded that since there are some limitations of the dataset considered for this project as mentioned in section 5.6, few of the evaluation metrics must be discarded and only Precision, Recall and F1 – score had to be used to determine the best model for classifying a given frame of a video as an action frame or not.

Chapter 7: Conclusion and Future Work

7.1 Conclusion

In this major project, the whole ideology was to understand and study the new research topic "Action spotting" in sporting videos related to the game of soccer and to build deep learning models to detect whether a video frame from a football match is an action frame or not.

During the execution of this project, there were few challenges that I came across which were tackled effectively on day-to-day basis. Since the hardware specifications such as processor and storage were not meet, extraction of features from the videos was not possible. Because of which, pre – extracted features were used for developing deep neural network models. Next in line was to decode how features and labels are related since both were having different file formats. The explained regarding which is provided in section 5.4.

Furthermore, the traditional deep learning algorithms having inputs in the form of images or videos but for this project there were pre-extracted features from the videos of football matches and annotated labels. Hence, a different approach was needed for developing deep learning model. For this, understanding of different pooling mechanisms were required which are further connected to the fully connected layers in order to determine an outcome. This has been explained in chapters 3 and 5.

Lastly, since the features that were captured those were of two halves of a football match, the challenge was to combine those features along with the labels to form a complete game of football. This was handled successfully and has been briefed in sections 5.4 and 5.5. This eventually generated the train, test and valid datasets. There were few limitations to the dataset that are discussed in section 5.6 that made this dataset more challenging.

Finally, series of thorough experiments were carried out for developed 12 deep neural network models to analyse which model outplays among them. The results based on performance metrics such as Precision, Recall and F1 – score suggests that there were two best models. Those were build using ResNET features with NetVLAD pooling with an F1 – score of 0.4850 and other was build using I3D features with NetVLAD pooling giving an F1 – score of 0.5026 for the dataset that was under investigation.

7.2 Future Scope

In the current work, pre – extracted features were used because of the limitation of the hardware specifications. In terms of future work, if the hardware requirements are meet then will try to extract features directly from the videos of the football matches. It could be either using available deep learning model architectures by applying the concept of transfer learning or developed a whole new architecture for extracting them which will be a challenging task to be accomplished. Thus, using these features will create more accurate deep learning models with new pooling techniques.

Another work that can be undertaken is to find out the best parameters using hyperparameter tuning for the currently developed deep learning models. Moreover, current models can only help in classifying whether a frame from a video is having an action event or not. This can further be used to carry out action localization tasks.

Furthermore, different versions of SoccerNet datasets can be used to build more accurate deep learning models that can help in understanding the soccer broadcasting and help them to generate automate highlights of a football game.

REFERENCES

ABU-EL-HAIJA, S., N. KOTHARI, J. LEE et al. 2016. Youtube-8m: A large-scale video classification benchmark. *In*: *arXiv preprint arXiv:1609.08675.*

ARANDJELOVIC, R., P. GRONAT, A. TORII et al. 2018. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **40**(6), pp.1437 - 1451.

ARBUES-SANGUESA, A., A. MARTIN, J. FERNÁNDEZ et al. 2020. Using Player's Body-Orientation to Model Pass Feasibility in Soccer. *In*: *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. Seattle, WA, USA: IEEE, pp.3875 - 3884.

CABA HEILBRON, F., V. ESCORCIA, B. GHANEM, and J. CARLOS NIEBLES. 2015. Activitynet: A large-scale video benchmark for human activity understanding. *In*: *In Proceedings of the ieee conference on computer vision and pattern recognition*. Boston, MA, USA: IEEE, pp.961-970.

CARREIRA, J. and A. ZISSERMAN. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. *In*: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp.6299-6308.

CIOPPA, A., A. DELIEGE, S. GIANCOLA et al. 2020. A context-aware loss function for action spotting in soccer videos. *In*: *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Seattle, WA, USA: IEEE, pp.13126 - 13136.

DELIEGE, A., A. CIOPPA, S. GIANCOLA et al. 2021. Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. *In*: *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*., pp.4508-4519.

DONAHUE, J., L. ANNE HENDRICKS, S. GUADARRAMA et al. 2015. Long-term recurrent convolutional networks for visual recognition and description. *In*: *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, pp.2625-2634.

DUCA, G. 2021. Design of the Smart Objectives System in the Management of a Research Project. *In*: *In International Conference on Management Science and Engineering Management, ICMSEM 2021,Part of Lecture Notes on Data Engineering and Communications Technologies*. Springer, Cham, pp.452-462.

FARIN, D., S. KRABBE, and W. EFFELSBERG. 2003. Robust camera calibration for sport videos using court models. *In*: *In Storage and Retrieval Methods and Applications for Multimedia 2004*. United States: SPIE, pp.80 - 91.

GIANCOLA, S., M. AMINE, T. DGHAILY, and B. GHANEM. 2018. Soccernet: A scalable dataset for action spotting in soccer videos. *In*: *In Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. Salt Lake City, UT, USA, pp.1711-1721.

GIANCOLA, S. and B. GHANEM. 2021. Temporally-aware feature pooling for action spotting in soccer broadcasts. *In*: *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*., pp.4490 - 4499.

HETTRICK, S., M. ANTONIOLETTI, L. CARR et al. 2014. *Uk research software survey 2014.* Southampton.

HE, K., X. ZHANG, S. REN, and J. SUN. 2016. Deep residual learning for image recognition. *In*: *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, pp.770-778.

HOMAYOUNFAR, N., S. FIDLER, and R. URTASUN. 2017. Sports field localization via deep structured models. IEEE Conference on Computer Vision and Pattern Recognition, pp.5212-5220.

HOMAYOUNFAR, N., S. FIDLER, and R. URTASUN. 2017. Sports field localization via deep structured models. *In*: *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, HI, USA: IEEE, pp.5212 - 5220.

ISTASSE, M., J. MOREAU, and C. DE VLEESCHOUWER. 2019. Associative embedding for team discrimination. *In*: *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. Long Beach, CA, USA: IEEE, pp.2477 - 2486.

JÉGOU, H., Douze, M., Schmid, C. and Pérez, P. 2010. Aggregating local descriptors into a compact image representation. *In*: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Francisco, CA, USA: IEEE, pp.3304-3311.

JIANG, Y., K. CUI, L. CHEN et al. 2020. Soccerdb: A large-scale database for comprehensive video understanding. *In*: *In Proceedings of the 3rd International Workshop on Multimedia Content Analysis in Sports*., pp.1 - 8.

JIANG, Y.G., J. LIU, A.R. ZAMIR et al. 2014. THUMOS challenge: Action recognition with a large number of classes.

KARPATHY, A., G. TODERICI, S. SHETTY et al. 2014. Large-scale video classification with convolutional neural networks. *In*: *In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. Columbus, OH, USA: IEEE, pp.1725-1732.

KAY, W., J. CARREIRA, K. SIMONYAN et al. 2017. The kinetics human action video dataset. *In*: *arXiv preprint arXiv:1705.06950*.

KUEHNE, H., H. JHUANG, E. GARROTE et al. 2011. HMDB: a large video database for human motion recognition. *In*: *In 2011 International conference on computer vision*. Barcelona, Spain: IEEE, pp.2556-2563.

LANGE, D. 2022. *'Big Five' European football leagues revenue 1996-2019 | Statistic*. [online]. [Accessed 09 May 2022]. Available from World Wide Web: <https://www.statista.com/statistics/261218/big-five-european-soccer-leagues-revenue/>

LANGE, D. 2022. *Premier league revenue streams England 2014/15-2019/20*. [online]. [Accessed 09 May 2022]. Available from World Wide Web: <https://www.statista.com/statistics/874020/revenue-of-premier-league-football-clubs-by-stream/>

LEWIS, Bridgeman, Volino MARCO, Jean-Yves GUILLEMAUT, and Adrian HILTON. 2019. Multi-Person 3D Pose Estimation and Tracking in Sports. *In*: *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Long Beach, CA, USA: IEEE, pp.2487-2496.

MANAFIFARD, M., H. EBADI, and H.A. MOGHADDAM. 2017. A survey on player tracking in soccer videos. *Computer Vision and Image Understanding*. **159**, pp.19-46.

MARSZALEK, M., I. LAPTEV, and C., SCHMID. 2009. Actions in context. *In*: *In 2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp.2929-2936.

MINOURA, H., T. HIRAKAWA, T. YAMASHITA et al. 2021. Action Spotting and Temporal Attention Analysis in Soccer Videos. *In*: *In 2021 17th International Conference on Machine Vision and Applications (MVA)*. Aichi, Japan: IEEE, pp.1 - 6.

NIEBLES, J.C., C.W. CHEN, and L. FEI-FEI. 2010. Modeling temporal structure of decomposable motion segments for activity classification. *In*: *In European conference on computer vision*. Berlin, Heidelberg.: Springer, pp.392 - 405.

NÖSEL, I. 2020. *4 main steps for an effective project resources management*. [online]. [Accessed 28 May 2022]. Available from World Wide Web: <https://www.appvizer.co.uk/magazine/operations/project-management/project-resources-management>

OGBEIWI, O. 2017. Why written objectives need to be really SMART. *British Journal of Healthcare Management*. **23**(7), pp.324-336.

OPTASPORTS. 1996. *Opta data from Stats Perform*. [online]. [Accessed 09 May 2022]. Available from World Wide Web: <https://www.statsperform.com/opta/>

PANCHAL, M. 2021. *The Benefits & Importance of Resource Management*. [online]. [Accessed 28 May 2022]. Available from World Wide Web: <https://guide.quickscrum.com/the-benefits-importance-of-resource-management/>

REMATAS, K., I. KEMELMACHER-SHLIZERMAN, and B. and Seitz, S. CURLESS. 2018. Soccer on your tabletop. *In*: *In Proceedings of the IEEE conference on computer vision and pattern recognition*. Salt Lake City, UT, USA: IEEE, pp.4738 - 4747.

RESEARCH AND MARKETS. 2021. *Sports Global Market Report 2021: COVID-19 Impact and Recovery to 2030*. [online]. [Accessed 09 May 2022]. Available from World Wide Web: <https://www.researchandmarkets.com/reports/5240315/sports-global-market-report-2021-covid-19-impact?utm_source=GNOM&utm_medium=PressRelease&utm_code=wks2vx&utm_campaign=1513400+-+Global+Sports+Market+Report+(2021+to+2030)+-+COVID-19+Impact+and+Recovery&ut>

RITCHIE, H. and M. ROSER. 2021. *Natural Disasters*. [online]. [Accessed 29 May 2022]. Available from World Wide Web: <https://ourworldindata.org/natural-disasters#:~:text=a%20new%20tab.-,Summary,from%200.01%25%20to%200.4%25.>

RODRIGUEZ, M.D., J. AHMED, and M. SHAH. 2008. Action mach a spatio-temporal maximum average correlation height filter for action recognition. *In*: *In 2008 IEEE conference on computer vision and pattern recognition*. Anchorage, AK, USA: IEEE, pp.1 - 8.

ROHRBACH, M., S. AMIN, M. ANDRILUKA, and B. SCHIELE. 2012. A database for fine grained activity detection of cooking activities. *In*: *In 2012 IEEE conference on computer vision and pattern recognition*. Providence, RI, USA: IEEE, pp.1194-1201.

RONGVED, O.A.N., S.A. HICKS, V. THAMBAWITA et al. 2020. Real-time detection of events in soccer videos using 3D convolutional neural networks. *In*: *In 2020 IEEE International Symposium on Multimedia (ISM)*. Naples, Italy: IEEE, pp.135 - 144.

SARKAR, S., A. CHAKRABARTI, and D. PRASAD MUKHERJEE. 2019. Generation of ball possession statistics in soccer using minimum-cost flow network. *In*: *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. Long Beach, CA, USA: IEEE, pp.2515 - 2523.

SIGURDSSON, G.A., G. VAROL, X. WANG et al. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. *In*: *In European Conference on Computer Vision*. Springer, Cham., pp.510-526.

SIMONYAN, K. and A. ZISSERMAN. 2014. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems 27*.

SOOMRO, K., A.R. ZAMIR, and M. SHAH. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *In*: *arXiv preprint arXiv:1212.0402*.

SPORTRADAR. 2001. *The driver of ever deeper engagement in sport worldwide*. [online]. [Accessed 09 May 2022]. Available from World Wide Web: <https://www.sportradar.com/about-us/group-set-up/>

STATS PERFORM. 1981. *Sports Data - Sports AI, Technology, Data Feeds*. [online]. [Accessed 09 May 2022]. Available from World Wide Web: <https://www.statsperform.com/#:~:text=We%20take%20the%20richest%20live>

THEAGARAJAN, R., F. PALA, X. ZHANG, and B. BHANU. 2018. Soccer: Who has the ball? Generating visual analytics and player statistics. *In*: *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Salt Lake City, UT, USA: IEEE, pp.1830 - 1838.

THOMAS, G., R. GADE, T.B. MOESLUND et al. 2017. Computer vision for sports: Current applications and research topics. *Computer Vision and Image Understanding*. **159**, pp.3 - 18.

TOMEI, M., L. BARALDI, S. CALDERARA et al. 2021. Rms-net: Regression and masking for soccer event spotting. *In*: *In 2020 25th International Conference on Pattern Recognition (ICPR)*. Milan, Italy: IEEE, pp.7699 - 7706.

TRAN, D., L. BOURDEV, R. FERGUS et al. 2015. Learning spatiotemporal features with 3d convolutional networks. *In*: *Proceedings of the IEEE international conference on computer vision*. IEEE, pp.4489-4497.

TRAN, D., H. WANG, L. TORRESANI et al. 2018. A closer look at spatiotemporal convolutions for action recognition. *In*: *In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE, pp.6450 - 6459.

UR REHMAN, A., S.B. BELHAOUARI, M.A. KABIR, and A. KHAN. 2023. On the Use of Deep Learning for Video Classification. *Applied Sciences*. **13**(3), p.2007.

VANDERPLAETSE, B. and S. DUPONT. 2020. Improved soccer action spotting using both audio and video streams. *In*: *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. Seattle, WA, USA: IEEE, pp.3921 - 3931.

VATS, K., M. FANI, P. WALTERS et al. 2020. Event detection in coarsely annotated sports videos via parallel multi-receptive field 1d convolutions. *In*: *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. Seattle, WA, USA: IEEE, pp.882 - 883.

YANG, Y. and D. LI. 2017. Robust player detection and tracking in broadcast soccer video based on enhanced particle filter. *Journal of Visual Communication and Image Representation*. **46**, pp.81 - 94.

YEUNG, S., O. RUSSAKOVSKY, N. JIN et al. 2018. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*. **126**(2), pp.375-389.

YOONWOOJEONG. 2020. *NetVLAD: CNN Architecture for Weakly Supervised Place Recognition*. [online]. [Accessed 09 Jun 2023]. Available from World Wide Web: <https://towardsdatascience.com/netvlad-cnn-architecture-for-weakly-supervised-place-recognition-ce64b08bebaf>

YU, D., Wang, H., Chen, P. and Wei, Z. 2014. Mixed pooling for convolutional neural networks. *In*: *Rough Sets and Knowledge Technology: 9th International Conference, RSKT 2014*. Shanghai, China: Springer International Publishing, pp.364-375.

YU, J., A. LEI, Z. SONG et al. 2018. Comprehensive dataset of broadcast soccer videos. *In*: *In 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. Miami, FL, USA: IEEE, pp.418-423.

ZECHA, D., M. EINFALT, and R. LIENHART. 2019. Refining joint locations for human pose tracking in sports videos. *In*: *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. Long Beach, CA, USA: IEEE, pp.2524-2532.

BIBLIOGRAPHY

AGGARWAL, C. 2018. Applications of Convolutional Networks: Video Classification. *In*: C. AGGARWAL, (ed). *Neural Networks and Deep Learning: A Textbox*, Switzerland: Springer International Publishing, p.367.

AGGARWAL, C. 2018. The Basic Structure of a Convolutional Network. *In*: C. AGGARWAL, (ed). *Neural Networks and Deep Learning: A Textbook*, Switzerland: Springer, pp.318-328.

DI, W., A. BHARDWAJ, and J. WEI. 2018. Getting Started with Neural Networks. *In*: S. KOLTE, V. PAGARE, A. SINGH et al., (eds). *Deep Learning Essentials*, Birmingham: Packt Publishing Ltd, pp.71-75.

GÉRON, A. 2019. Classification. *In*: N. TACHE, (ed). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, Sebastopol: O'Reilly Media, Inc., pp.92-100.

GÉRON, A. 2019. Deep Computer Vision Using Convolutional Neural Networks. *In*: N. TACHE, (ed). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, Sebastopol: O'Reilly Media, Inc., pp.442-444.

APPENDICES

Appendix A: Results obtained after performing 5 experiments.

Features: ResNET
Pooling Layer: MaxPooling1D
Fully Connected Network: 1 Flatten(Input) Layer, 3 hidden layers, 1 Output Layer

Experiments	Confusion Matrix				Precision	Recall	F1 score	Accuracy
	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)				
1	9975	5	1502	4	0.501	0.6567	0.4675	0.8687
2	9168	4	2309	5	0.5008	0.6771	0.4461	0.7986
3	9368	4	2109	5	0.5009	0.6858	0.4516	0.816
4	10647	6	830	3	0.5015	0.6305	0.4846	0.9272
5	9389	3	2088	6	0.5012	0.7424	0.4527	0.8179

Table A.1:Using ResNET Features and Max Pooling technique.

Features: C3D
Pooling Layer: MaxPooling1D
Fully Connected Network: 1 Flatten(Input) Layer, 3 hidden layers, 1 Output Layer

Experiments	Confusion Matrix				Precision	Recall	F1 score	Accuracy
	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)				
1	8678	2	2797	7	0.5011	0.767	0.433	0.7562
2	8472	2	3003	7	0.501	0.758	0.427	0.7383
3	9273	4	2202	5	0.5009	0.6818	0.4491	0.8079
4	7163	0	4312	9	0.501	0.8121	0.3864	0.6245
5	8244	2	3231	7	0.5009	0.7481	0.4202	0.7184

Table A.2: Using C3D Features and Max Pooling technique.

Features: I3D
Pooling Layer: MaxPooling1D
Fully Connected Network: 1 Flatten(Input) Layer, 3 hidden layers, 1 Output Layer

Experiments	Confusion Matrix				Precision	Recall	F1 score	Accuracy
	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)				
1	11195	4	272	5	0.5088	0.7659	0.5113	0.9759
2	10700	3	767	6	0.5037	0.7998	0.4903	0.9329
3	11049	5	418	4	0.5045	0.7039	0.4998	0.9631
4	10899	5	568	4	0.5032	0.6974	0.494	0.95
5	11392	9	75	0	0.4996	0.4967	0.4981	0.9926

Table A.3: Using I3D Features and Max Pooling technique.

Features: ResNET
Pooling Layer: AveragePooling1D
Fully Connected Network: 1 Flatten(Input) Layer, 3 hidden layers, 1 Output Layer

Experiments	Confusion Matrix				Precision	Recall	F1 score	Accuracy
	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)				
1	10211	5	1266	4	0.5013	0.667	0.4738	0.8893
2	7822	3	3655	6	0.5006	0.6741	0.4068	0.6815
3	9671	5	1806	4	0.5008	0.6435	0.4593	0.8423
4	9962	5	1515	4	0.501	0.6562	0.4671	0.8676
5	8934	3	2543	6	0.501	0.7225	0.4399	0.7783

Table A.4: Using ResNET Features and Average Pooling technique.

Features: C3D
Pooling Layer: AveragePooling1D
Fully Connected Network: 1 Flatten(Input) Layer, 3 hidden layers, 1 Output Layer

Experiments	Confusion Matrix				Precision	Recall	F1 score	Accuracy
	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)				
1	7716	2	3759	7	0.5007	0.725	0.4038	0.6725
2	9601	5	1874	4	0.5008	0.6405	0.4575	0.8363
3	8566	3	2909	6	0.5008	0.7065	0.4294	0.7464
4	8649	4	2826	5	0.5006	0.6546	0.4314	0.7535
5	9010	5	2465	4	0.5005	0.6148	0.4413	0.7849

Table A.5: Using C3D Features and Average Pooling technique.

Features: I3D
Pooling Layer: AveragePooling1D
Fully Connected Network: 1 Flatten(Input) Layer, 3 hidden layers, 1 Output Layer

Experiments	Confusion Matrix				Precision	Recall	F1 score	Accuracy
	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)				
1	11088	8	379	1	0.5009	0.539	0.4939	0.9662
2	10878	8	589	1	0.5004	0.5298	0.4883	0.9497
3	11304	7	163	2	0.5057	0.604	0.5077	0.9851
4	9221	2	2246	7	0.5014	0.7909	0.4487	0.8041
5	10984	5	483	4	0.5038	0.7011	0.4971	0.9574

Table A.6: Using I3D Features and Average Pooling technique.

Features: ResNET
Pooling Layer: MixedPooling1D
Fully Connected Network: 1 Flatten(Input) Layer, 3 hidden layers, 1 Output Layer

Experiments	**Confusion Matrix**				**Precision**	**Recall**	**F1 score**	**Accuracy**
	True Positive (TP)	**False Positive (FP)**	**False Negative (FN)**	**True Negative (TN)**				
1	10162	4	1315	5	0.5016	0.7204	0.4732	0.8851
2	8131	3	3346	6	0.5007	0.6875	0.4163	0.7084
3	9899	3	1578	6	0.5017	0.7645	0.467	0.8623
4	9468	3	2009	6	0.5013	0.745	0.4549	0.8248
5	9437	3	2040	6	0.5013	0.7445	0.454	0.8221

Table A.7: Using ResNETFeatures and Mixed Pooling technique.

Features: C3D
Pooling Layer: MixedPooling1D
Fully Connected Network: 1 Flatten(Input) Layer, 3 hidden layers, 1 Output Layer

Experiments	**Confusion Matrix**				**Precision**	**Recall**	**F1 score**	**Accuracy**
	True Positive (TP)	**False Positive (FP)**	**False Negative (FN)**	**True Negative (TN)**				
1	9598	5	1877	4	0.5008	0.6404	0.4574	0.8361
2	9090	4	2385	5	0.5008	0.6738	0.444	0.7919
3	9715	5	1760	4	0.5008	0.6455	0.4606	0.8463
4	8833	5	2642	4	0.5004	0.6071	0.4363	0.7695
5	10142	4	1333	5	0.5016	0.7196	0.4727	0.8835

Table A.8: Using C3D Features and Mixed Pooling technique.

Features: I3D
Pooling Layer: MixedPooling1D
Fully Connected Network: 1 Flatten(Input) Layer, 3 hidden layers, 1 Output Layer

Experiments	**Confusion Matrix**				**Precision**	**Recall**	**F1 score**	**Accuracy**
	True Positive (TP)	**False Positive (FP)**	**False Negative (FN)**	**True Negative (TN)**				
1	11416	9	51	0	0.4996	0.4977	0.4986	0.9947
2	10543	3	924	6	0.503	0.793	0.4853	0.9192
3	11381	9	86	0	0.4996	0.4962	0.4979	0.9917
4	9629	3	1838	6	0.5014	0.7531	0.4596	0.8395
5	11314	9	153	0	0.4996	0.4933	0.4964	0.9858

Table A.9: Using I3D Features and Mixed Pooling technique.

Features:ResNET
Pooling Layer: NetVLAD
Fully Connected Network: 1 Flatten(Input) Layer, 3 hidden layers, 1 Output Layer

Experiments	Confusion Matrix				Precision	Recall	F1 score	Accuracy
	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)				
1	10713	7	764	2	0.5009	0.5778	0.4852	0.9328
2	10630	8	847	1	0.5002	0.5186	0.4818	0.9255
3	11260	9	217	0	0.4996	0.4905	0.495	0.9803
4	10222	4	1255	5	0.5017	0.7231	0.4749	0.8903
5	10824	7	653	2	0.5012	0.5826	0.4882	0.9425

Table A.10: Using ResNET Features and NetVLADPooling technique.

Features: C3D
Pooling Layer: NetVLAD
Fully Connected Network: 1 Flatten(Input) Layer, 3 hidden layers, 1 Output Layer

Experiments	Confusion Matrix				Precision	Recall	F1 score	Accuracy
	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)				
1	9868	7	1607	2	0.5002	0.541	0.4634	0.8594
2	11174	8	301	1	0.5012	0.5424	0.4963	0.973
3	10474	5	1001	4	0.5017	0.6786	0.481	0.9123
4	9085	4	2390	5	0.5008	0.6736	0.4438	0.7915
5	9369	4	2106	5	0.5009	0.686	0.4517	0.8162

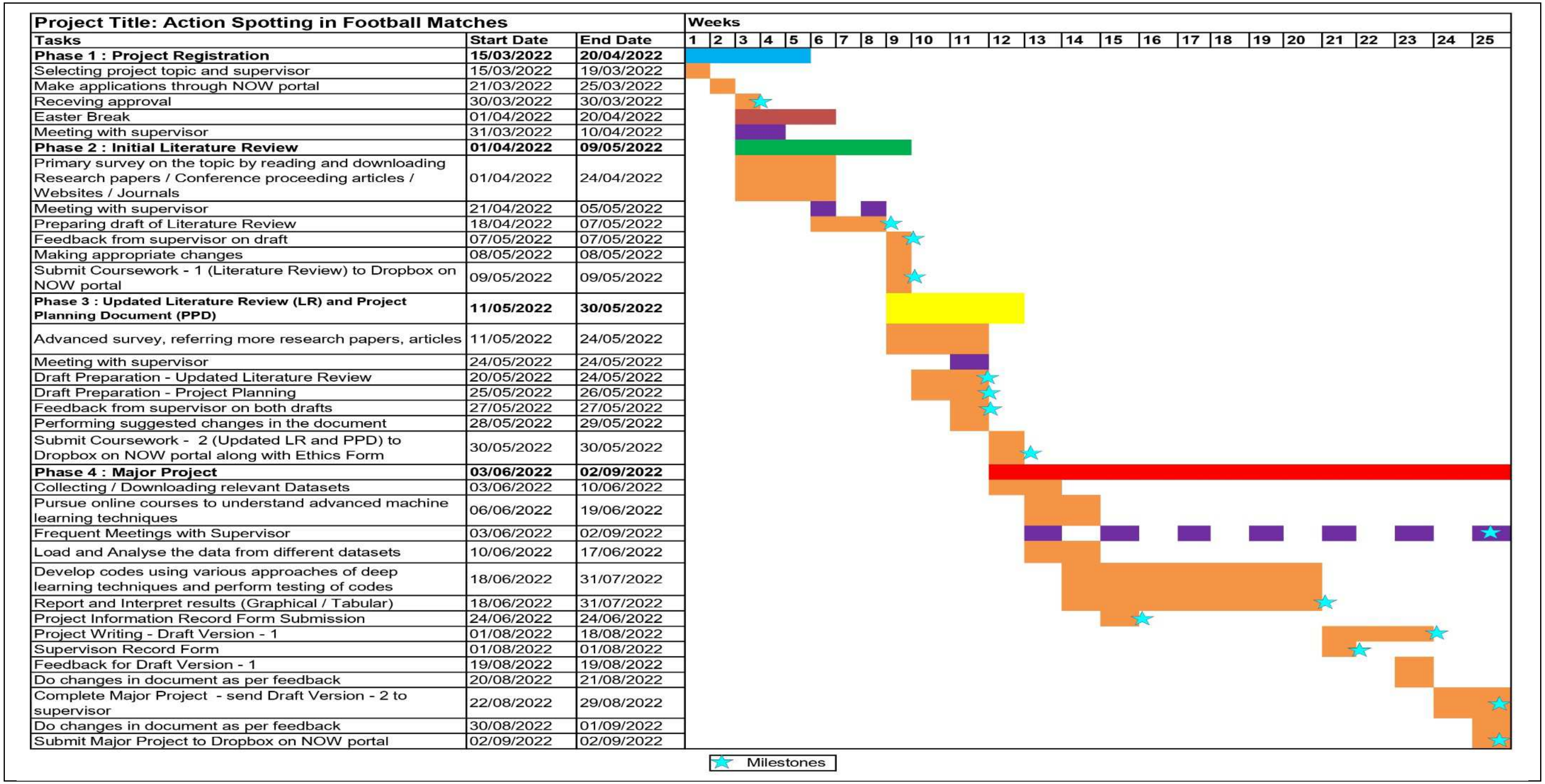
Table A.11: Using C3D Features and NetVLAD Pooling technique.

Features : I3D
Pooling Layer: NetVLAD
Fully Connected Network: 1 Flatten(Input) Layer, 3 hidden layers, 1 Output Layer

Experiments	Confusion Matrix				Precision	Recall	F1 score	Accuracy
	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)				
1	10871	3	596	6	0.5048	0.8073	0.4964	0.9478
2	11418	8	49	1	0.5096	0.5534	0.5157	0.995
3	11342	8	125	1	0.5036	0.5501	0.5044	0.9884
4	11381	9	86	0	0.4996	0.4962	0.4979	0.9917
5	11413	9	54	0	0.4996	0.4976	0.4986	0.9945

Table A.12: Using I3D Features and NetVLAD Pooling technique.

Appendix B: Old Gantt Chart



Project Title: Action Spotting in Football Matches		
Tasks	**Start Date**	**End Date**
Phase 1 : Project Registration	**15/03/2022**	**20/04/2022**
Selecting project topic and supervisor	15/03/2022	19/03/2022
Make applications through NOW portal	21/03/2022	25/03/2022
Receving approval	30/03/2022	30/03/2022
Easter Break	01/04/2022	20/04/2022
Meeting with supervisor	31/03/2022	10/04/2022
Phase 2 : Initial Literature Review	**01/04/2022**	**09/05/2022**
Primary survey on the topic by reading and downloading Research papers / Conference proceeding articles / Websites / Journals	01/04/2022	24/04/2022
Meeting with supervisor	21/04/2022	05/05/2022
Preparing draft of Literature Review	18/04/2022	07/05/2022
Feedback from supervisor on draft	07/05/2022	07/05/2022
Making appropriate changes	08/05/2022	08/05/2022
Submit Coursework - 1 (Literature Review) to Dropbox on NOW portal	09/05/2022	09/05/2022
Phase 3 : Updated Literature Review (LR) and Project Planning Document (PPD)	**11/05/2022**	**30/05/2022**
Advanced survey, referring more research papers, articles	11/05/2022	24/05/2022
Meeting with supervisor	24/05/2022	24/05/2022
Draft Preparation - Updated Literature Review	20/05/2022	24/05/2022
Draft Preparation - Project Planning	25/05/2022	26/05/2022
Feedback from supervisor on both drafts	27/05/2022	27/05/2022
Performing suggested changes in the document	28/05/2022	29/05/2022
Submit Coursework - 2 (Updated LR and PPD) to Dropbox on NOW portal along with Ethics Form	30/05/2022	30/05/2022
Phase 4 : Major Project	**03/06/2022**	**02/09/2022**
Collecting / Downloading relevant Datasets	03/06/2022	10/06/2022
Pursue online courses to understand advanced machine learning techniques	06/06/2022	19/06/2022
Frequent Meetings with Supervisor	03/06/2022	02/09/2022
Load and Analyse the data from different datasets	10/06/2022	17/06/2022
Develop codes using various approaches of deep learning techniques and perform testing of codes	18/06/2022	31/07/2022
Report and Interpret results (Graphical / Tabular)	18/06/2022	31/07/2022
Project Information Record Form Submission	24/06/2022	24/06/2022
Project Writing - Draft Version - 1	01/08/2022	18/08/2022
Supervison Record Form	01/08/2022	01/08/2022
Feedback for Draft Version - 1	19/08/2022	19/08/2022
Do changes in document as per feedback	20/08/2022	21/08/2022
Complete Major Project - send Draft Version - 2 to supervisor	22/08/2022	29/08/2022
Do changes in document as per feedback	30/08/2022	01/09/2022
Submit Major Project to Dropbox on NOW portal	02/09/2022	02/09/2022

Figure B: Old Gantt Chart