A Fuzzy Logic-Based System for Soccer Video Scenes Classification

Wei Song

A thesis submitted for the degree of Doctor of Philosophy in Computer Science

School of Computer Science and Electronic Engineering

University of Essex

August 2018

Abstract

Massive global video surveillance worldwide captures data but lacks detailed activity information to flag events of interest, while the human burden of monitoring video footage is untenable. Artificial intelligence (AI) can be applied to raw video footage to identify and extract required information and summarize it in linguistic formats. Video summarization automation usually involves text-based data such as subtitles, segmenting text and semantics, with little attention to video summarization in the processing of video footage only. Classification problems in recorded videos are often very complex and uncertain due to the dynamic nature of the video sequence and light conditions, background, camera angle, occlusions, indistinguishable scene features, etc.

Video scene classification forms the basis of linguistic video summarization, an open research problem with major commercial importance. Soccer video scenes present added challenges due to specific objects and events with similar features (e.g. “people” include audiences, coaches, and players), as well as being constituted from a series of quickly changing and dynamic frames with small inter-frame variations. There is an added difficulty associated with the need to have light weight video classification systems working in real time with massive data sizes.

In this thesis, we introduce a novel system based on Interval Type-2 Fuzzy Logic Classification Systems (IT2FLCS) whose parameters are optimized by the Big Bang–Big Crunch (BB-BC) algorithm, which allows for the automatic scenes classification using optimized rules in broadcasted soccer matches video. The type-2 fuzzy logic systems would be unequivocal to present a highly interpretable and transparent model which is very suitable for the handling the encountered uncertainties in video footages

and converting the accumulated data to linguistic formats which can be easily stored and analysed. Meanwhile the traditional black box techniques, such as support vector machines (SVMs) and neural networks, do not provide models which could be easily analysed and understood by human users. The BB-BC optimization is a heuristic, population-based evolutionary approach which is characterized by the ease of implementation, fast convergence and low computational cost. We employed the BB-BC to optimize our system parameters of fuzzy logic membership functions and fuzzy rules. Using the BB-BC we are able to balance the system transparency (through generating a small rule set) together with increasing the accuracy of scene classification. Thus, the proposed fuzzy-based system allows achieving relatively high classification accuracy with a small number of rules thus increasing the system interpretability and allowing its real-time processing. The type-2 Fuzzy Logic Classification System (T2FLCS) obtained 87.57% prediction accuracy in the scene classification of our testing group data which is better than the type-1 fuzzy classification system and neural networks counterparts. The BB-BC optimization algorithms decrease the size of rule bases both in T1FLCS and T2FLCS; the T2FLCS finally got 85.716% with reduce rules, outperforming the T1FLCS and neural network counterparts, especially in the "out-of-range data" which validates the T2FLCSs capability to handle the high level of faced uncertainties.

We also presented a novel approach based on the scenes classification system combined with the dynamic time warping algorithm to implement the video events detection for real world processing. The proposed system could run on recorded or live video clips and output a label to describe the event in order to provide the high level summarization of the videos to the user.

Acknowledgements

I would like to express the deepest appreciation to my PhD supervisor, Professor Hani Hagras, who has given me this opportunity to continue the study, for his patience, guidance, enthusiastic encouragement and useful critiques of this research. Without his supervision and constant help this dissertation would not have been possible.

I wish to thank various people for their contribution to this project; Dr. Bo Yao, my friend, for his valuable suggestions on this project; Miss Min Liao, my girlfriend, for her help and encouragement in my studying and living; Dr. Tianyu Li, my friend, for his help and his knowledge of the philosophy to guide me when I fell into pain.

On a personal note, I would like to thank my parents and my kin for their affectionate support, patience, and encouragement throughout my research.

Contents

List of Figures

List of Tables

Publications Arising from this work

The following publications have resulted from my work:

<u>Journal Papers</u>

- W. Song, H. Hagras, D. Alghazzawi, "A cloud computing based Big-Bang Big-Crunch fuzzy logic multi classifier system for Soccer video scenes classification", ***Journal of Memetic Computing***, Vol.8, No.4, pp.307-323, December 2016.

<u>Book Chapters</u>

- W. Song, H. Hagras, "A Hybrid Type-2 Fuzzy Based System for Football Scenes Classification within Recorded Video Sequences", **Multimodal Analytics for Next-Generation Big Data Technologies and Applications**, Springer, October 2018.

<u>Conference Papers</u>

- W Song, H Hagras, "A type-2 fuzzy logic system for event detection in soccer videos", ***Proceedings of the 2017 IEEE International Conference on Fuzzy Systems***, Naples, Italy, July 2017
- W. Song, H. Hagras "A Big-Bang Big-Crunch Type-2 Fuzzy Logic Based System for Soccer Video Scene Classification", ***Proceedings of the 2016 IEEE International Conference on Fuzzy Systems****, Vancouver, Canada, July 2016.*
- W. Song, H. Hagras "A Big-Bang Big-Crunch Fuzzy Logic Based System for Sports Video Scene Classification", ***Proceedings of the 2016 IEEE International Conference on Fuzzy Systems****, Vancouver, Canada, July 2016.*

Chapter 1 : Introduction

"The most significant aspect of all this for us is that AlphaGo isn't just an 'expert' system built with hand-crafted rules, but instead uses general machine learning techniques to allow it to improve itself, just by watching and playing games".

— *Silver and Hassabis (2016)*

Artificial intelligence (AI) has been omnipresent and disseminated worldwide over recent decades. People have witnessed huge changes due to AI applications in many domains. AI brings the incredible innovation to the world to refresh the traditional cognition of technology in many applications, such as medical healthcare, autonomous cars, telephone customer services, and education. These applications successfully affect daily lives. One famous AI application in game science presented in 2014 was the "AplhaGo" program developed by DeepMind to play the board game Go aiming to play against human, which then defeated the professional player Lee Sedol in a five-game match in March 2016.

The applications AI have a tendency to be smarter than people in some areas. Go is considered much more difficult for computers to win than other games such as chess, because it has many more branching factors. The victory of 'AlphaGo' has proved the potentiality of AI applications can bring revolutionary changes into the traditional applications. Typical AI perceives its environment and takes action to maximize its chance of successfully achieving its goals [Poole, 1998]. From a scientific point of view, AI applications learn the needed information from the environment, then react to the realistic situations with the corresponding behaviours. These behaviours are the targets for human expectations in applying AI in real-life contexts.

1.1 Artificial Intelligence Systems

There are several significant stages in the history of AI development. AI was born between 1952 to 1956, which was the result of the earliest research into thinking machines, a confluence of ideas that became prevalent in the late 1930s, 1940s, and early 1950s [McCorduck, 2004]. The famous Turing's test was published in 1950 by Alan Turing, whereby he speculated about the possibility of creating machines that think. Turing proposed that a human evaluator would judge natural language conversations between a human and a machine designed to generate human-like responses. The evaluator would be separated from the two partners (i.e. one machine and one human) and in conversation with them; if the evaluator cannot distinguish between the human and machine, the latter is said to have passed Turing's test [Turing, 1950].

The test results do not depend on the ability to give correct answers to questions, only the closely one's answers resemble those a human would give. Turing's test continues to be an important standard assessment for AI to the present. In 1955, Allen Newell and Herbert A. Simon (later a Nobel Laureate) created the "Logic Theorist" (with help from J. C. Shaw), which introduced several concepts that came to be central to AI research, such as reasoning search, heuristics, and list processing. As such, it represented a milestone in the development of AI and our understanding of intelligence in general [Crevier, 1993]. Artificial intelligence was founded as an academic discipline in the Dartmouth Conference of 1956.

The years after Dartmouth Conference were an era of discovery, of sprinting across new ground, called the "golden years" of AI, particularly the period of 18 years from 1956 to 1974. There are some particular breakthroughs on the AI development arising during this time. The "reasoning as search" provided early AI solutions in searching problems instead of the basic algorithm. To achieve some goals (like winning a game or proving a theorem), they proceeded step by step towards it (by making a move or a deduction) as if searching through a maze, backtracking whenever they reached a dead end [Russell, 2003].

Another main application which benefited from AI was natural language processing. An important goal of AI research is to allow machines to communicate in natural languages like English. Joseph Weizenbaum's ELIZA was the first chatterbot in the world [Joseph, 1976]. ELIZA could communicate with users that were so realistic that users occasionally were fooled into thinking they were communicating with a human being and not a program. However, ELIZA did not understand the talking process; she simply gave a canned response or repeated back what was said to her, rephrasing her response with a few grammar rules [McCorduck, 2004]. The most important thing was "machine learning", coined in 1959 by Arthur Samuel. Machine learning evolved from the study of pattern recognition and computational learning theory, exploring the study and construction of algorithms that can learn from and make predictions on data [Kohavi, 1998] – such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions [Bishop, 2006].

AI development atrophied during the mid-1970s as researchers reached the limits of available computer power, facing problems such as the intractability of the combinatorial explosion, conflict between reasoning and common-sense knowledge, and the problems of the frame and qualification [McCarthy, 1981]. In the 1980s, "expert systems" brought the AI program into a development "boom" through to 1987. The "expert systems" were adopted by corporations around the world and knowledge became the focus of mainstream AI research. An expert system is a program that answers questions or solves problems about a specific domain of knowledge, using logical rules that derived from expert knowledge [Crevier, 1993]. At that moment, the "expert systems" were mostly used to solve the real problem and looked very smart. The programs of the "expert systems" were not really smart or had the intelligence like human beings. The actions of the "expert systems" came from the logical rules, designed by experts.

Following the second AI winter (1987-1993), more and more scientists accepted a new program which was called "intelligent agents". An intelligent agent is a system that perceives its environment and takes actions which maximize its chances of success. By this definition, simple programs that solve specific problems are "intelligent agents", as are human beings and organizations of human beings, such as corporations [Russell, 2003]. Recent years have witnessed renewed interest in machine learning, following the recent success of deep learning, which was fuelled by the availability of cheap and powerful hardware.

1.2 Overview of Machine Learning

In the past 20 years, machine learning has achieved great success both in academic and industrial applications. It is very apparent that the potentiality of machine learning represents the possibility of developing true AI systems in the future, which is generally accepted by most people. The concept of machine learning always appears in combination with data, which presents the importance of the relationship between data and learning; a seminal outline of learning problems summarising the process of machine learning was produced by the late 1990s [Mitchell, 1997], based on which scientists have produced many algorithms that are widely used in industry, including in commerce, transportation, and manufacturing.

From the academic point of view, machine learning is generally used to perform classification, regression, and clustering of data. However, in real, applied systems, scientists expect machines to learn the needed information from the data and become intelligent, such that they can output smart feedback to users. The difference of AI application objectives led different learning algorithms being employed in numerous professional fields, for example Hidden Markov Mode (HMM) is used in speech recognition more effectively than other learning algorithms, and Naïve Bayes (NB) is widely used to mark an email as spam or not spam [Mehran, 1998].

Machine learning algorithms are categorized into three categories with different learning processes: supervised learning (SL), unsupervised learning (UL), and reinforcement learning (RL). SL learns the input-output pairs from the training data and produces an inferred function, which can be used for mapping new examples. UL is called “clustering” in some studies, referring to a function that describes the structure of “unlabelled” data. RL does not require training data like SL and UL, rather it is concerned with how software agents ought to take action in an environment so as to maximize some notion of cumulative reward [Sutton, 1998].

The most successful learning task among machine learning applications is SL, which reflects the demand of users in real situations. In logistic regression (LR), with a relatively small number of inputs, there is a very simple relationship between the inputs and outputs. The same applies to the decision trees (DT): people can follow the logic applied by the tree and understand why a certain element was assigned to one class or the other. They are both called white box algorithms with the ability to explain the function specified by the machine learning algorithm. However, in some other learning processes, like neural networks, which are examples of black box algorithms, no one (not even neural networks experts) can easily explain why a certain decision has been taken.

The interpretability of the learning algorithm is very important in the learning process for scientists, which contains two aspects in the different stages. One of the aspects of the interpretability is expressed in the process of training part, wherein the white box learning algorithms explain numerical values with the human understanding translations in the process of training. The other aspect of the interpretability is embodied in the algorithm running after training completed, whereby the process of the new inputs to the outputs could be interpreted to the users with the specific path and components used, enabling people to understand the details of the algorithms' running processes (i.e. the reason for the classification or progress of the regression).

1.3 Uncertainty in the Learning Process

In existing expert systems of learning algorithms, uncertainty is dealt with through a combination of predicate logic and probability-based methods. A serious shortcoming of these methods is that they are not capable of coming to grips with the pervasive fuzziness of information in the knowledge base, thus they are mostly ad hoc in nature [Zadeh, 1983]. When dealing with real-world problems, we can rarely avoid uncertainty. At the empirical level, uncertainty is an inseparable companion of almost any measurement, resulting from a combination of inevitable measurement errors and resolution limits of measuring instruments. At the cognitive level, it emerges from the vagueness and ambiguity inherent in natural language. At the social level, uncertainty even has strategic uses and it is often created and maintained by people for different purposes (e.g. privacy, secrecy, or propriety), as expressed in *Uncertainty-Based Information* [Klir, 1998]:

> *"Uncertainty involved in any problem-solving situation is a result of some information deficiency. Information (pertaining to the model within which the situation is conceptualized) may be incomplete, fragmentary, not fully reliable, vague, contradictory, or deficient in some other way. In general, these various information deficiencies may result in different types of uncertainty"* [Klir, 1998] [Mendel, 2001].

Source data and learning algorithms bring "uncertainty" into the learning process. To increase the applicability of learning approaches and reduce variability, in most methods multiple rounds of cross-validation are performed using different partitions, and the validation results are combined (e.g. averaged) over the rounds to give an estimate of the model's predictive performance. However, cross-validation does not eliminate uncertainty.

1.4 Uncertainty Handling with Fuzzy Logic

An alternative approach to the management of uncertainty, as suggested in this thesis, is based on the use of fuzzy logic, which is the logic underlying approximate (or, equivalently, "fuzzy") reasoning. A feature of fuzzy logic which is of particular importance to the dealing with the uncertainty in expert systems is that it produces a framework to process the fuzzy quantifiers (e.g. infrequently, too few, few, almost all, not very many, many, too many, approximately half etc.). In this way, the fuzzy logic subsumes both predicate logic and probability theory, and makes it possible to manage different types of uncertainty within a single conceptual framework.

In fuzzy logic, the deduction of a conclusion from a set of premises is reduced, in general, to the solution of a nonlinear program through the application of projection and extension principles. This approach to deduction leads to various basic syllogisms which may be used as rules of combination of evidence in expert systems [Zadeh, 1983]. An overview on fuzzy logic is presented in Chapter 3.

1.5 Overview of Intelligent Systems in Video Summarization

Video summarization is a branch subject of the computer vision which extends traditional image processing techniques, aiming to summarize the needed information from the video type data of the living camera capturing and recording videos, in order to provide solutions for intelligent systems in fields including security, entertainment, and robotics. Clear objectives of video summarization were outlined by of 2002 [Ma, 2002]:

- Object detection, which contains the segmentation and object recognition in video processing, whereby the system should find the target object from the videos and extract it from the background.
- Complete the tracking and locating the target objects when moving in the video.
- Recognizing the actions of the target and use human understanding languages to describe the behaviours of the target during the movement, like "a car is parking" or "people are dancing".

The video is the combination of many images which are in the sequence saved in the data to record the visual information for a period of time. The image is a kind of matrix of structural data that records the colour information into each minimum block, called a pixel, which contains three numerical values to describe the red, green, and blue pigmentation of the pixel colour. Therefore, for the machines, images are just numerical values, which differ from what the human eye captures. This makes it difficult for machines to understand the meaning of images presented as they are for humans. The use of the intelligent systems (e.g. AI or expert systems) is a key solution in image and video processing in order for machines to complete high-level detection. However, current intelligent systems cannot provide perfect solutions for the video summarization problem due to the high complexity of video type data and the high uncertainty of the image content.

From the intelligent systems viewpoint, the illegible definition of the objects in the image/video is illegible for machines. Scientists have designed relevant algorithms to undertake detection and recognition tasks for the target objects in the images, in order to complete some basic detection to support the high level detections, like the traditional Sobel, Laplacian, and Marr filters, which find object boundaries; or some extending approaches that combine with intelligent learning algorithms, like Haar Feature-based Cascade Classifier [Viola, 2001] to find human faces, or HoG support vector machine [Dalal, 2005], which finds people in images. However, increasingly complex detection, such as behavioural event detection in videos, becomes more complicated and difficult to implement.

1.6 The Research Objectives

This research aims to present novel techniques which can operate on complicated soccer video sequences and extract events and scenes of interest from them, which could be then used in linguistic summaries. In doing this, we employ type-1 and type-2 fuzzy logic systems to handle the encountered uncertainties in such applications, and novel BB-BC systems to enable maximized accuracy of prediction, and the interpretability of the generated model. The proposed framework employs the following steps:

1. Developing a video scenes classification system based on type-1 fuzzy logic systems, employing generated fuzzy rules with confidence and support in order

to predict the input video frames with the label output, to describe the current scene in the soccer videos. This classification system will be constructed based on the type-1 fuzzy membership functions, which are created from the training data.

2. The type-2 fuzzy logic classification system will then be generated wherein the type-2 fuzzy sets are generated from the type-1 fuzzy sets.
3. The type-1 and type-2 fuzzy systems are optimized by Big-Bang Big-Crunch (BB-BC) algorithm to obtain the best accuracy and reduce the size of the rule base. The fuzzy systems will bring similarity into the fuzzy rules matching in order to minimize the size of rule base by BB-BC optimization.
4. We then produce an event detection system for the videos in order to classify video clips with the label "events". The event detection system is built based on the previous fuzzy logic scenes classification system and employs the dynamic time wrapping (DTW) algorithms to process the video clips, calculating the similarity between the input video and the event classes, labelling the highest possibility event.

1.7 Video Summarization on Soccer Videos

Video summarization has a wildly research domains, the specific of related researches and applications will be presented in Chapter 2. For my research, we aim to implement the objectives on recorded soccer videos. Soccer videos as the represented sports videos have a massive audience, and hence there is a huge commercial potential which necessitates tools that can automatically classify sports video scenes.

Many intelligent applications of soccer videos have been reported based on players and movement detection. For instance, in [Ekin, 2003], an approach was presented for soccer match video analysis and summarization using object-based features like color histograms and image segments to detect the target object in soccer videos. In [Dai, 2005] a framework which detects replay scenes was presented using real-time subtitle text in combination with local image features and text semantics. However, these frameworks are based on supplied subtitle text and local segmental image features, which are not applicable to all kinds of sports videos. In [Zhu, 2009], a method was introduced to extract tactical information from the team moving events in broadcasted soccer video service for coaches' decision making. The system was trained by subtitle text information, player position, semantic events and analysed by the team tactics. However, for most videos such detailed information cannot be collected.

Thus, there is a need to provide a better framework to summarize the content information for the soccer videos and output the linguistics to the audience. However, the black box models such as SVM and neural networks which do not provide models that could be easily analysed and understood by human users. Deep learning algorithms are the extension of the traditional neural networks which has stronger learning ability and popular in currently researches. However, they have no interpretability that people still couldn't analysis and understand. For video summarization problems, the deep learning algorithms require huge dataset with labeling and marking images to train and learn. We could not find the exist database ready for applying the deep learning and it is impossible to build the needed database ourselves.

Fuzzy Logic Classification System (FLCS) employs fuzzy sets and provides a white box technique that can handle the uncertainties associated with soccer video classification. People could analysis the learning and prediction process by fuzzy sets and rule base. The fuzzy inputs and outputs provided the linguistics that people can easily understand.

1.8 Thesis Structure

This thesis is divided into seven chapters. The proposed systems contain two systems, one called “scene classification system”, which can predict the input video frames with the label of the scene; the other named “event(s) detection system(s)” which can predict the video clips event label. Chapter 2 presents a brief overview of video summarization, while Chapter 3 explains fuzzy logic systems.

Chapter 4 presents the process of the fuzzy logic scenes classification system, comprising feature extraction, definition of the training vectors, type-1 fuzzy sets and membership functions creation, the fuzzy rules generation, and the type-2 fuzzy system generation.

In Chapter 5, we will employ the Big-Bang Big-Crunch (BB-BC) algorithm to optimize the fuzzy logic scene classification systems (of type-1 and type-2) in order to improve the performance and reduce the cost of computation in video processing. The specific process of the size of rule base reduction and the parameters of the type-2 fuzzy sets optimization will be presented.

Chapter 6 provides a novel approach based on the existing scenes classification system and Dynamic Time Wrapping (DTW) to detect the events in soccer video clips, including main events like goal, miss goal, foul, free kick, outside ball and compare with non-event video clips.

Chapter 7 presents the thesis conclusions and highlights areas for future work.

1.9 Discussion

This chapter presented a general introduction of the history of AI and an overview of machine learning, then it introduced the concept of uncertainty in the learning process. The objectives of the thesis have been situated in the context of existing research in order to frame the video classification problems to be addressed. The next chapter will present an overview of video summarization systems.

Chapter 2 Overview of Video Summarization

A brief description of video summarization with its difficulties was presented in the first chapter, in this chapter we will present some important related concepts including "Scenes" in video summarization and associated approaches of feature extractions in the computer vision and image processing. Then we will present the detail of the background of video summarization and some critical researches of this subject.

2.1 Scenes

Scene is an important component in video summarization. The scene may have different means in different contexts. In the Webster Dictionary, there are two definitions of "scene":

A subdivision of an act in a dramatic presentation in which the setting is fixed and the time continuous.

One of the subdivisions of a play; as a division of an act presenting continuous action in one place.

In this section, we introduce the concept of "scene" as it pertains to video technology and explain some important applications of applied to scenes. And for these applications we will also present the some important researches from the domains of image processing, computer vision, pattern recognition and video summarization which are relevant to the scenes.

2.1.1 The Concept of Scenes in Videos

A scene may comprise a single physical locale such as a studio, recorded by one or more cameras, or it might have a moving or otherwise dynamic background. Additionally, “a scene may also be defined by the continuity of ongoing actions performed by the actors… a framework can be deployed to find scene boundaries which exploits both colour and motion similarities of shots” [Ekin, 2003]

To explain the definition of scene in detail, we introduce the example shown in Figure 2.1, from a soccer match video. In these three groups, two of them have the same scene and the other one does not. Group A shows some close-up images of people (players and coach), so it should be a scene of people in the same category. However, Group A can be divided into more specific groups, like players and not players. The scene classification depends on the definition of class. Group B shows three images from the main camera from a far view to the field, and they should be in the same class. The images in Group C come from three different angles of cameras without common features, so they do not belong to the same scene.

Figure 2.1: Scenes examples

The scene is an important concept in video summarization and in computer vision. For example, if the user expects the intelligent system to detect some target, like a ball in a soccer match video, the system should be focused on the scene, as in group B. It will lose the computation cost in detecting the target ball in the people close-up scene (group A) or some other scenes as in group C. To ensure the target can be detected successfully, there is a need to classify scenes before some high level detection in video summarization.

2.1.2 The Classification of the Scenes

As described by pervious applications and according to definition of scene, scene can be redefined according to the conditions and classes. Continuing with the example of a soccer match, a scene in the video can be divided into two big groups – scenes of game play and scenes of people – or into more specific categories, as shown in Figure 2.2.

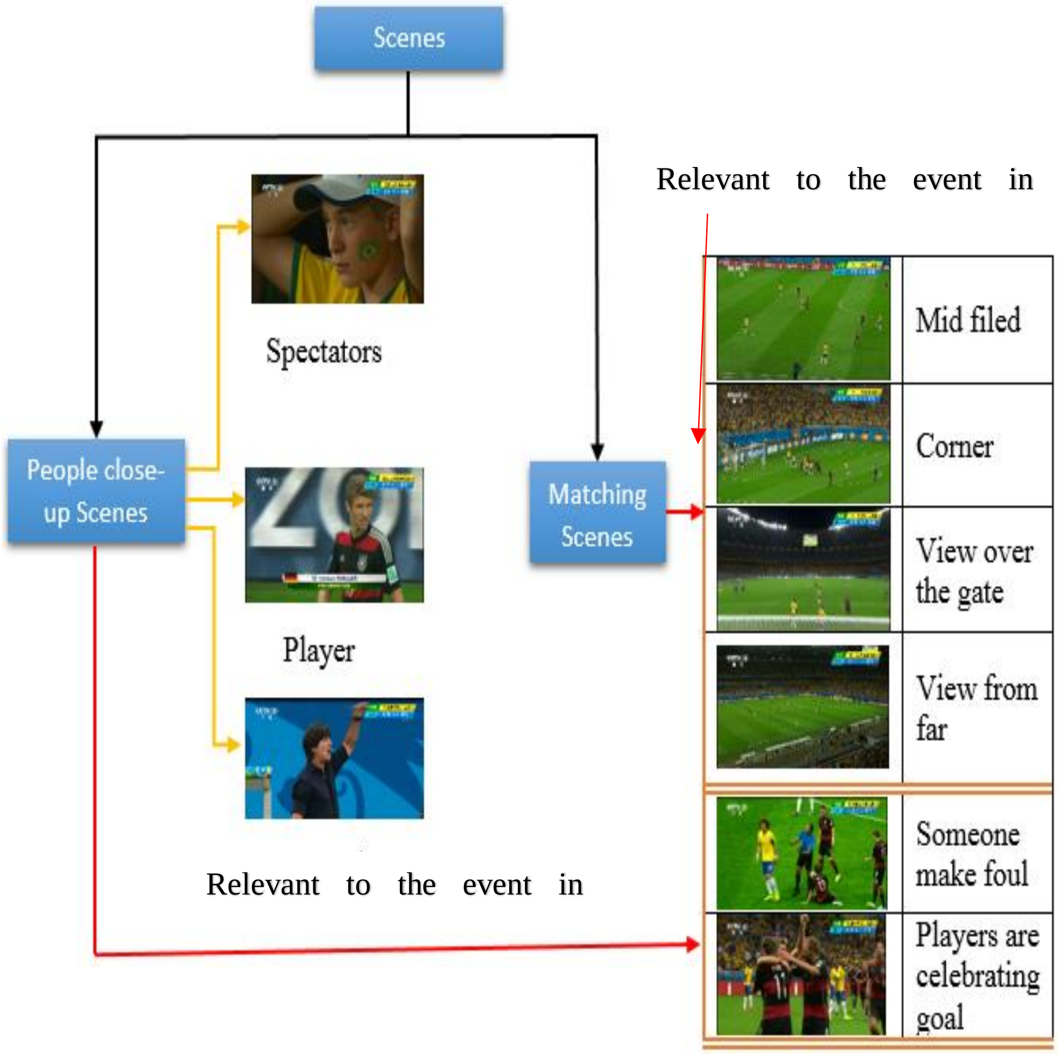


Figure 2.2: Classified scenes in soccer match

In Figure 2.2, it is easy to understand that some close-up scenes are not directly relevant to the event. The director edits the video clips and wants to show more interesting clips to the audience. However, some close-up scenes have relevance to events. In the example of the soccer video, if the player made a foul, the sound of the whistle would pause the match and referees may go to the midfield. The referee might then show the player a yellow card or give a warning. If a team scored a goal in the match, they may celebrate this, and most of time the camera would focus on the goal-scoring player and team players as they get together for celebration. “scene” is the minimum component for high-level events in video summarization.

2.1.3 Relevant Applications in Video Scenes

Many researchers devised plans for event detection or future prediction, but most research faced the fundamental problem of how to increase classification accuracy in video scenes, due to the complexity of scenes classification in videos, particularly for similar scenes. It is difficult for intelligent systems to process videos for high-level detection or summarization because each scene has a unique method to process (there is no general way to process all kinds of scenes). Each expert system is created to process target video scenes. For example, different match scenes come from different cameras with different angles. Cameras are located in different positions around the stadium. However, in this kind of mixed video scenes, the machine does not know how to process such scenes without learning. In [Zhu, 2006], the scientist did the object identification to similar scenes, however this system does not work for other scenes from different video sources. Thus, scene classification is a high priority in video summarization.

Chen built a relational graph by processing a movie. The system architecture has three stages. Obviously in this research relational graph building is the final objective. It is not difficult for the system to get a relational graph by mining knowledge while the input is identified. However, ensuring the accuracy of the final output and the correct result of the relational graph depends on the result of classification.

In the early work of pattern recognition, some approaches were presented to solve the problem in video summarization that aimed to generate a series of visual contents for users to browse and understand the whole information or story of the video efficiently. Actually, what happened or what is happening in the video is the core of video summarization. However, the specific method to implement the video summarization differs according to the distinctive type of videos. For example, in video summarization applied in movies [Chen, 2005], the techniques of video summarization can be roughly classified into the two categories: static storyboard, and dynamic skimming. However, in this example of movie video summarization [Chen, 2009], the scientist showed the event scenes detection, which is much more complicated than simple object detection, because the movie videos have many more components, more complicated backgrounds and more diverse categories of scenes for video summarization. The system employed the movie subtitles and associate movie scenes data to classify some important elements, and then to create a relational graph in order to complete high level detection. This research structuralized the video contents, and discovered important shots and their semantically related shots in the video.

The traditional techniques of video summarization display the key-frames by using the ranking method. It is a classical approach which comes from early computer vision techniques. Intelligent systems are widely used for the detection when processing the sport videos [Zhou, 2006]. The system can locate and track the target objects but could not describe further more information of the content. Another intelligent system in video summarization produced by [Ekin, 2003] employed histograms, frame segmentation, and classification method to classify the scenes of the soccer videos, however, the system could only classify four basic scenes: long shot, in-field medium shot, close-up shot and out-of-field shot. Some researchers focused on other targets in video summarization. For example, in [Rasheed, 2009], they produced an approach to improving the accuracy of scenes classification in high-level segmentation of video with graph cut and SSG algorithms to classify movie scenes and obtain better results.

Sports videos are a popular target for video summarization. Yoshimasa Takahashi and his team used baseball match metadata, with semantic and image sequences. This can summarize the majority events and classify significant player scenes in order to present some video content to the audience [Takahashi, 2005]. Other researchers focused on the combination processing of soccer video and webcast text employing semantic recognition approaches to summarize video information of the video in the paper of [Dai, 2005] and [Xu, 2008]. In [Xu, 2008], they introduced a system which employed the replay time from the soccer video, then it used the semantic recognition approach to identify the content of the webcast text in order to recognize the event of the replay, but this research had too many limitations, including that events could be detected only in replay, not all the time.

[Dai, 2005] extended this research to a system able to process whole videos, not only focused on soccer videos, but also basketball ones. The method was the same employed web-cast semantic recognition approaches to assist in scenes classification. The proposed system in this research was able to detect eight events from the scenes (corner, shot, foul, card, free-kick, offside, goal, and substitution), but it could not detect player behaviour or specific events from videos, and the presence of the web-cast (i.e. subtitles) was an essential condition for the video summarization. Based on the classification results of the previous two papers, a clustering system was developed that could predict video events [Zhu, 2009]. This model actually analysed the movement direction of team attack from the video and trained the cluster with the label of event and time information. Its prediction model was based on earlier analysis from a system to predict the probability of events for a team focused on video and music multi presentation [Xu, 2005].

2.2 Background on Features Extraction

We assume the simplest videos are groups of images presented for 25 frames per second. In the early ages of video summarization, researchers sought to solve the problems of single image processing in order to extract some important information about objects in the image, like location and boundary. These basic image processing ideas continued to be used in studies to the present, alongside extended knowledge of computer vision. Basically, the first step of the video summarization is features extraction. The feature of one image generally contains the colour, boundary, and textures of objects etc. In this section, we present some traditional techniques for image processing applied in our proposed system.

2.2.1 Histograms

Histograms use a graph to describe the pixel colour level distribution of an image. Histograms give a rough sense of the density of the underlying distribution of the data, and often for density estimation: estimating the probability density function of the underlying variable. Imagine that a matrix contains information of an image (i.e. intensity in the range 0-255), as shown in Figure 2.3.

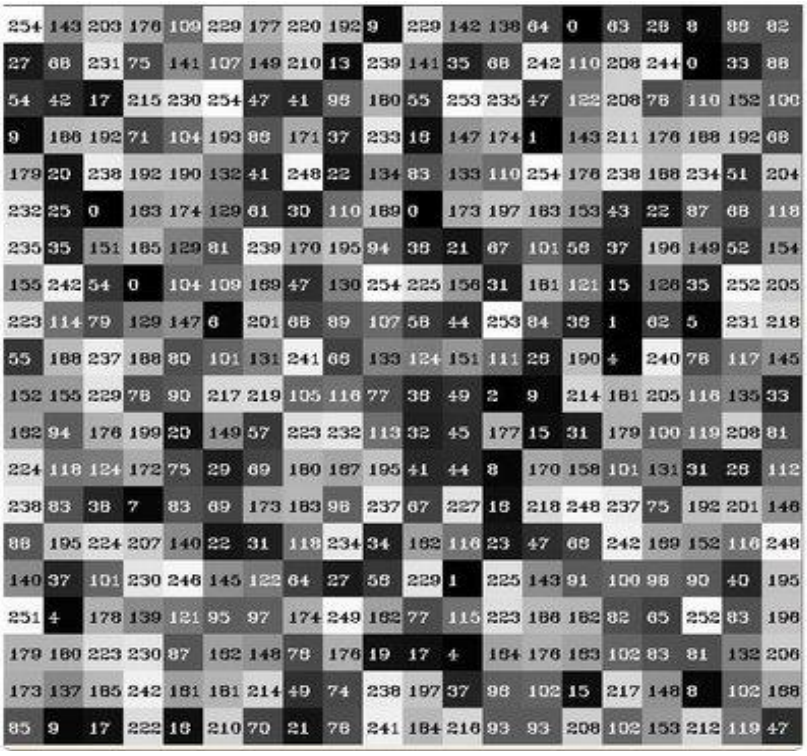


Figure 2.3: Matrix containing image information

Histograms were used in the earliest classifications of scenes and continue to be used in feature extraction in expert systems, for diverse research targets and disparate data. In [Ekin, 2003], the paper focused on soccer video summarization, with histograms shown by red, green, and blue, respectively, then the program can classify the scene and put the images sequence to the groups based on these histograms. As shown in Figure 2.4, here is a diagram of RGB histograms for an image. Histograms generally differ for different scenes in the same kind of video, describing boundary and colour features among different images and then concluding similarity to distinguish them from each other. As shown in Figure 2.4, there are three RGB histograms that present the different features of three groups where the red, green, blue are all different from other groups.

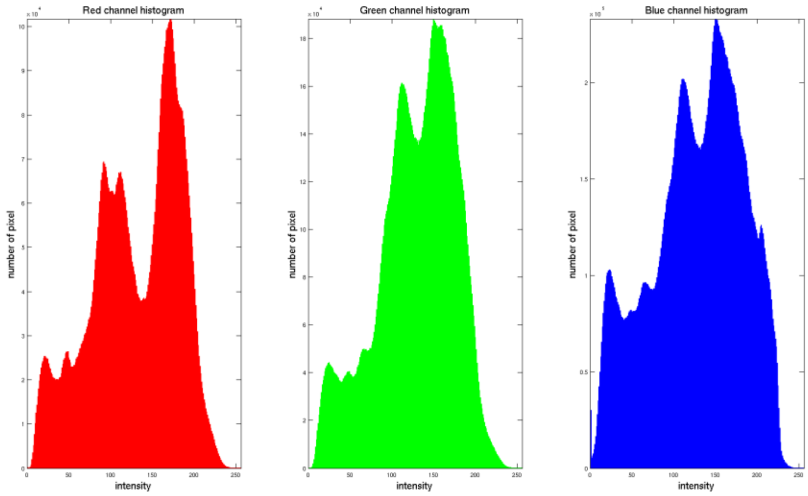


Figure 2.4: General overview of feature extraction using R, G, B histogram

2.2.2 Edges detection

The features of the edges are the most useful features in computer vision. A regular image generally contains many objects inside. In computer vision, people want the machine can recognize and identify objects and do further processing like tracking and segmentation. The edge represents the basic boundary lines of the objects in the image. Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. Thus, applying an edge detection algorithm to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image. Edge detection is one of the fundamental steps in image processing, image analysis, image pattern recognition, and computer vision techniques.

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a computational theory of edge detection explaining why the technique works. Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It has been widely applied in various computer vision systems. Canny has found that the requirements for the application of edge detection on diverse vision systems are relatively similar. Thus, an edge detection solution to address these requirements can be implemented in a wide range of situations.

Figure 2.5: Example of Canny Edge Detection

In figure 2.5, we present an example of one image and with its boundary lines after canny edge detection. Although his work was done in the early days of computer vision, the Canny edge detector (including its variations) is still a state-of-the-art edge detector. Edge detectors that perform better than the Canny usually require longer computation times or a greater number of parameters.

2.2.3 Corner detection

A corner can be defined as the intersection of two edges. A corner can also be defined as a point for which there are two dominant and different edge directions in a local neighbourhood of the point. In computer vision, usually there is a need to find matching points between different frames of an environment. If we know how two images relate to each other, we can use both images to extract information of them.

Corner detectors are not usually very robust and often require large redundancies introduced to prevent the effect of individual errors from dominating the recognition task. One determination of the quality of a corner detector is its ability to detect the same corner in multiple similar images, under conditions of different lighting, translation, rotation and other transforms.

The simplest way to detect corners in the image is using correlation, but this gets very computationally expensive and suboptimal. Moravec is one of the earliest corner detection algorithms and defines a corner to be a point with low self-similarity [Moravec, 1980]. The algorithm tests each pixel in the image to see if a corner is present, by considering how similar a patch centred on the pixel is to nearby, largely overlapping patches. The similarity is measured by taking the sum of squared differences (SSD) between the corresponding pixels of two patches. A lower number indicates more similarity.

The scale-invariant feature transform (SIFT) can be the most successful extension of corner detection. SIFT is an algorithm in computer vision to detect and describe local features in images. It was patented in Canada by the University of British Columbia and published by David Lowe in 1999 [Lowe, 1999]. We present an example of the SIFT in figure 2.6 to demonstrate the function of SITF, in the figure, the left side is a book with black background and taken from the front close-up. The right part is the photo of one room, we can easily find the book appears in this scene. The SIFT build the connection between two books and find the corresponding coordinates points.

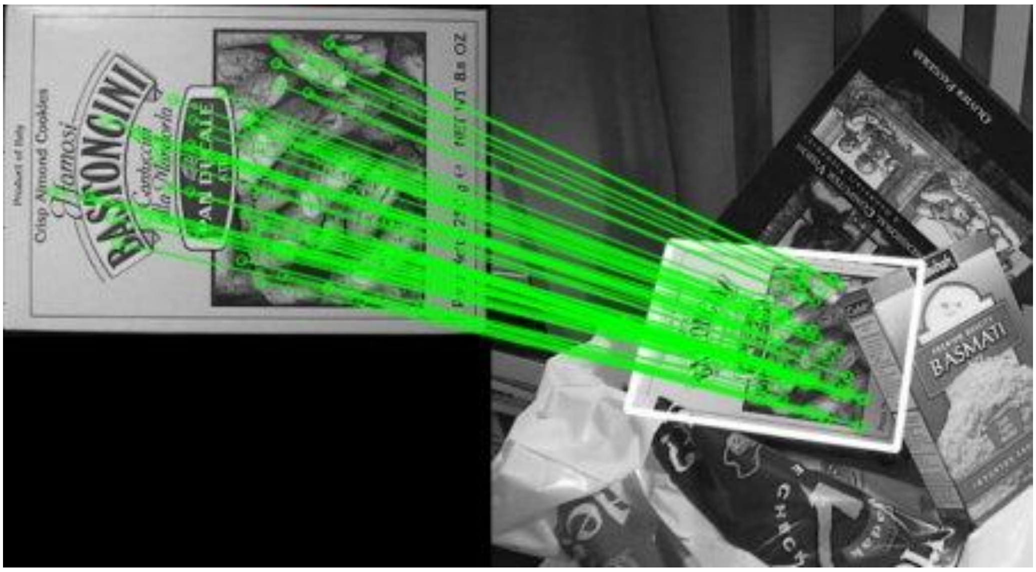


Figure 2.6: Example of scale-invariant feature transform (SIFT)

2.3 Video Summarization Applications

Video summarization includes the research and applications relating to videos, multi-camera captures, and image sequences in order to summarize the key information from the recorded videos and living camera captures. The application of video summarization has been applied on many areas including personal care, surveillance, public security, etc. Generally, the applications are applied on three situations: stationary scene videos, multi-cameras living captures, and movement scene videos, as described below.

Each situation has its own focus unique features.

- In stationary scene videos, the background of the video does not change. This includes applications on CCTV camera and surveillance that the scene does not change too much. The stationary scene keeps the background unchanged and applications mainly focus on the moving objects and people that appears on the screen captures.

- In multi-camera monitoring systems, scenes classifications are not required, as people are willing to detect target objects and track them in order to ensure the security and avoid risk (e.g. in densely populated city centres, where the police want to identify suspicious people and locate them via multi-camera monitoring systems).
- In movement scene videos, the background changes irregularly and the content of the videos changed frequency, whereby a smart video summarization system can find representative frames of the video to demonstrate and output with the linguistic to describe the content of the video. For example, the movie story summarization system could present the important events of the movie in the time bar and output the linguistics to the audience.

We will introduce the video summarization on different applicate domains as follows:

2.3.1 Stationary Scene Videos

There are various applications of video summarization for stationary scene videos. [Zhou, 2008] used the scene of the indoor environment, with the target being elderly people. In this scene the background is stationary while the target object is human. The objective of this research was behaviour detection for elderly people. Zhou Chen, Chung, He, Han and Keller introduced a system focused on human activity detection and analysis in order to record and summarize the behaviours of senior people in the room. The system was subsequently upgraded to address such challenges [Zhou, 2009a] and [Zhou, 2009b] including concentrating on the detection of dangerous behaviour in order to protect elderly people from accidents. For instance, the system would send warning messages when a person falls down in order to alert caregivers and ensure assistance to improve the safety of elders.

2.3.2 Multi-Cameras Monitoring System

Multi-cameras monitoring system requires more intelligence due to the different videos being captured. Target objects must be detected and tracked while moving from one camera capture to others. However, the cameras are set in different positions and face stationary scenes. Some common features in multi-camera monitoring systems include [Black, 2004]:

- Two or more cameras in different locations and capture in living
- The target of the applications is moving objects generally can be human, auto mobiles.
- The objectives of the applications are target tracking or behavioural identification

The scientist installed two cameras at two different positions of a gym, focusing on the treadmill and the person on it. There is a research [Havens, 2009] produced an exercise-feedback computer interface and used image contour tracking and swarm intelligence methods to track the location of the spine and shoulders during three exercises: treadmill, exercise bike, and overhead lateral pull-down. Multi-camera system is also used to track in some particular situations. In one research, they used multiple cameras to track detected automobiles on the street. This research technique is based on mathematics, with position calculation and algorithms of computer vision [Black, 2006].

2.3.3 Movement Scene Videos

Movement scene videos can be named as dynamic scene videos. The research of this kind of videos focuses challenge with scenes change as the most difficult problem in the video summarization research. In this research direction, scientists try to extract the useful information from the videos including low level summarizations like positions, objects and high level summarizations like behaviours and events.

Some scholars have focused on scene classification [Bosch, 2008] or target scene detection [Ekin, 2003] [Rasheed, 2005]. From these researches, scene is the most important part in the movement scene videos. The objectives of such research are implemented based on the scenes and key-frames summarization.

2.4 Discussion

In this chapter, we briefly elaborated on the background of the video summarization and its widespread applications. We presented some previous work on video summarization and demonstrated and analysed them from many aspects in order to present the difficulties of this area.

In this chapter we surveyed the major techniques applied in computer vision and video summarization. These techniques contain the colour feature extraction and object boundary feature extraction approaches that have been wildly used. Some basic techniques will be applied to our system in order to accomplish the objectives of our proposed systems in video summarization, like the histogram. The next chapter provides an overview of fuzzy logic systems.

Chapter 3 Overview of Fuzzy Logic Systems

3.1 Introduction to Fuzzy Logic

In traditional, classical logic, conclusions are always either true or false. However, in our world there are many propositions with variable answers, such as one might find when asking a group of people to share their feelings about the dishes in a restaurant. In this example, ‘true’ means people are completely satisfied with this meal and extremely happy with this restaurant’s services. Meanwhile, ‘false’ represents some people have a horrible feeling about the meal and being totally disappointed with the services. Fuzzy logic provides the degrees of truth and probabilities range between 0 and 1. In such instances, the feedback scores should be variable from 0 to 1, which means that the diner’s opinion on the meal and restaurant services would be between total disappointment and complete satisfaction. Fuzzy logic has been applied to many fields, including control theory, decision making, and AI [Gupta, 1977].

3.1.1 A Brief History of Fuzzy Logic

The idea of fuzzy logic was first introduced by Dr. Lotfi Zadeh at the University of California at Berkeley the 1965 and it has been continually expanded by researchers worldwide to the present. However, it was not initially welcomed in the academic community due to some of its underlying mathematics not yet being explored, thus associated applications developed slowly. One of the most famous objectors to fuzzy logic was Professor William Kahan, a colleague of Dr. Zadeh at UC Berkeley.

However, fuzzy logic found acceptance among mathematicians in the east. Japan was the first place to fully accept fuzzy logic, because it really worked. In the early 1980s, the success of many fuzzy logic-based products led to a revival in the US. In 1987 Hitachi employed a fuzzy controller in the Sendai train control, which used an innovative system and worked efficiently. In the same year the first commercial fuzzy controller was developed by Omron. Thus the year 1987 is considered the "fuzzy boom", due to the large number of products based on fuzzy logic to be popularised.

In 1993, Fuji fuzzy logic was applied to control chemical injection water treatment plants for the first time in Japan, and widespread deployments of fuzzy logic were institutionalised in Japan and South Korea between government, universities, and industries.

3.1.2 Fuzzy Sets

In mathematics a set is a collection of things that belong to some definition. Any item either belongs to that set or does not. Consider the set of room temperature, with people feeling the temperature is cold in a room lower than 15 degrees. This set can be represented graphically, as shown in Figure 3.1. The function shown describes the membership of the 'cold' set. This sharp edged membership functions works nicely for binary operations and mathematics, but do not work as neatly in describing the real world. This membership function makes no representation between different people's feelings.

For example, consider two people, one of whom feels that 14 ℃ is definitely cold and the other of whom feels 12 ℃ is definitely cold; both are cold. The other side of this limitation is the distinctions between 14.9 ℃ and 15.1 ℃. While this is a negligible temperature difference, this membership function can say one temperature is cold and the other is not. The fuzzy set approach to the set of COLD feeling provides a better description of the temperature feeling of people, as shown in Figure 3.2.

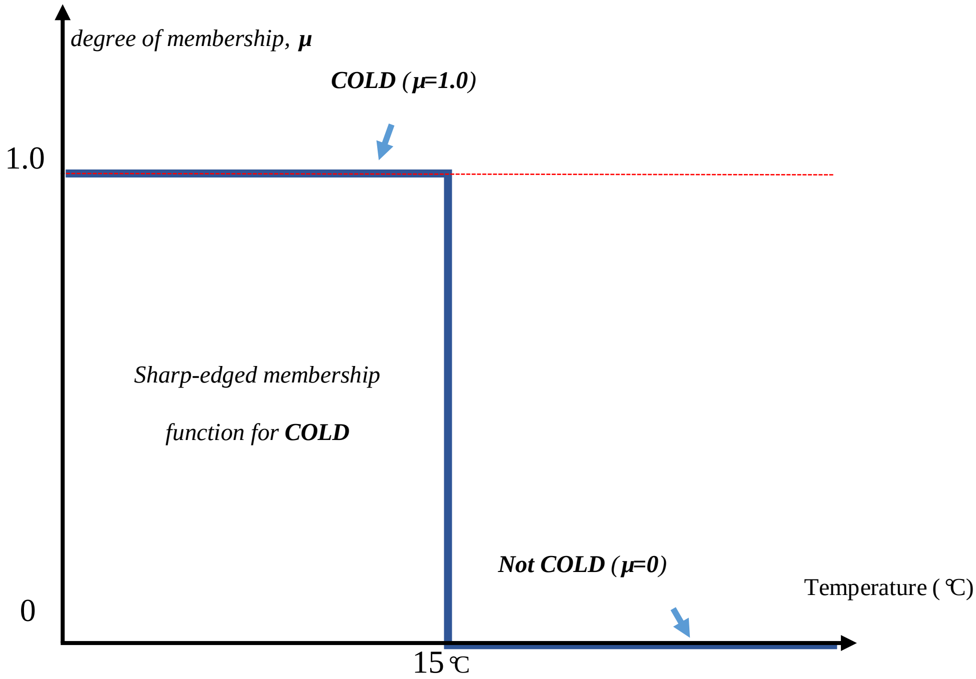


Figure 3.1: Example of traditional set

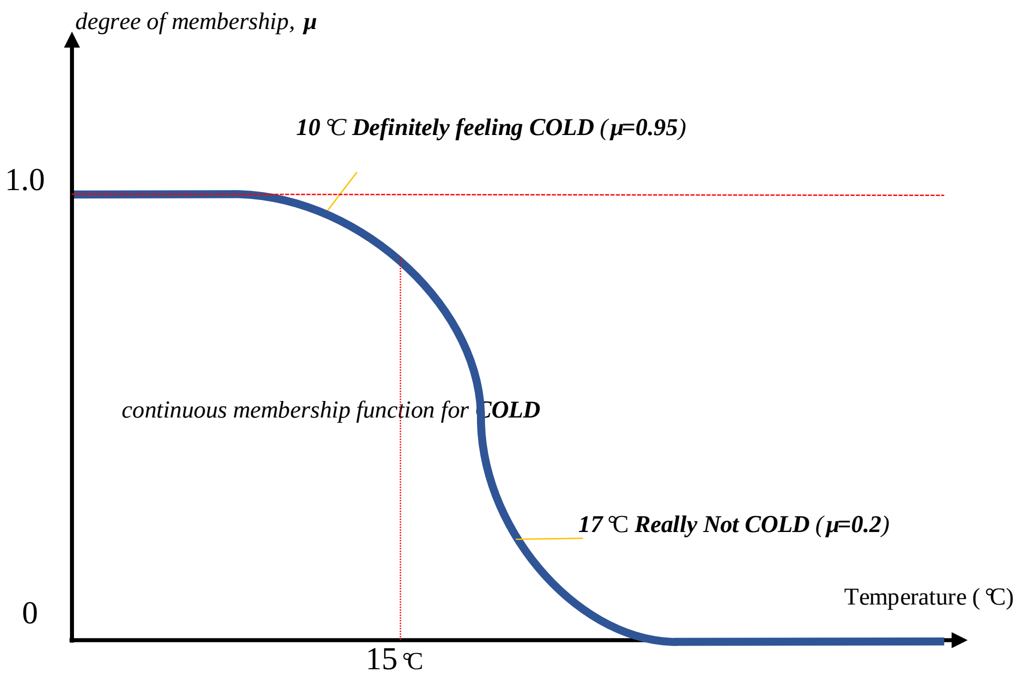


Figure 3.2: Example of the fuzzy set

The membership function defines the fuzzy set for the possible values underneath it on the horizontal axis. The vertical axis, on a scale of 0 to 1, provides the membership value of the height in the fuzzy set. Thus, for the different temperature feeling above, 10 ℃ has the membership function of 0.95, which is totally "COLD", while 17 ℃ has the membership function of 0.2, which is "Not COLD".

The concept of fuzzy sets was first proposed by Zadeh in 1965 in his paper *Fuzzy Sets*, which laid the foundation for all fuzzy logic that followed by mathematically defining fuzzy sets and their properties. The original definition of a fuzzy set is:

> *"Let **X** be a space of points, with a generic element of **X** denoted by **x**. Thus **X** = {**x**}.*
>
> *"A fuzzy set **A** in **X** is characterized by a membership function $\boldsymbol{f_A(x)}$ which associates with each point in **X** a real number in the interval **[0,1]**, with the values of $\boldsymbol{f_A(x)}$ at **x** representing the "grade of membership" of **x** in **A**. Thus, the nearer the value of $\boldsymbol{f_A(x)}$ to unity, the higher the grade of membership of **x** in **A**".*
>
> *----from Fuzzy Sets, by Lofti A. Zadeh [1965]*

3.1.3 Fuzzy Logic Concept

Fuzzy logic is an extension of classical binary logic, with the purpose of providing a framework for approximate reasoning. In practical terms, fuzzy logic is based on the theory of fuzzy sets, which extend the value of true and false from the classical logic to an arbitrary degree of truth, with specific values between 0 and 1. The basic concept of fuzzy logic theory applied the set theory with the main purpose of demonstrating the implementation of mathematical functions in hardware circuits [Wilkinson, 1963]. Fuzzy logic was implemented with maximum and minimum operations in hardware using diodes and resistor circuits to form a “logical block”, in a process dubbed “analogue logic”.

3.1.4 Fuzzy Logic Applications in the Real World

Since the first sensational application of fuzzy logic in a Japanese railway system in 1987, fuzzy logic controller system and associated fuzzy systems were quickly applied in industrial solutions it has been over 30 years [Timothy, 1995]. Today, the use of fuzzy logic in real-world applications has spread globally, with the products of fuzzy logic-based solutions including air conditioners, washing machines, rice cookers, auto mobiles and other manufacturing products [Samsung, 2018].

The fuzzy logic controller system represents the most successful application in the real world, with optimum performance, productivity, simplicity, and cost efficiency. For example, fuzzy logic washing machines apply a fuzzy logic controller during the washing process, controlling water intake, temperature, wash time, rinse performance, and spin speed. This optimizes the life span of the washing machine. More sophisticated machines weigh the load (so one cannot overload the washing machine), advise on the required amount of detergent, assess cloth material type and water hardness, and check whether the detergent is in powder or liquid form. Some machines even learn from past experience, memorizing programs and adjusting them to minimize running costs [Garrido, 2012].

3.2 Type-1 Fuzzy Logic

Type-1 fuzzy logic systems are the most common type of fuzzy logic systems today. Type-1 fuzzy sets were introduced by Zedeh as classical type-1 fuzzy logic systems. In this section, an overview of type-1 fuzzy logic is presented to describe the progress of type-1 fuzzy logic controller system.

3.2.1 Type-1 Fuzzy Sets

Consider a crisp set *A* in a universe of discourse *X*. An element *x* in *X* can either belong to the set *A* or not, expressed more formally using the notation to express the membership of *x* in *A*:

$$\mu_A(x) = \begin{cases} 1 \; if \; x \in A \\ 0 \; if \; x \notin A \end{cases} \tag{3-1}$$

A fuzzy set *F* (as depicted in Figure 3.3) is a generalization of a crisp set and is characterized by a membership function (MF) and resulting membership value $\mu_F(x) \in [0,1], \forall x \in X$. As such, an element x can have 0, 1 or partial membership function in a fuzzy set *F*, which can be represented as a set of ordered pairs of its elements *x* in *X* and his membership value $\mu_F(x)$ as below [Zadeh, 1965]:

$$F = \{(x, \mu_A(x)) | x \in X\} \tag{3-2}$$

F can be also defined in a continuous universe of discourse as follows:

$$F = \int \mu_F(x) \; dx \tag{3-3}$$

Or on a discretize universe of discourse as:

$$F = \sum \mu_F(x) \tag{3-4}$$

while fuzzy sets can be characterized by any type of membership function.

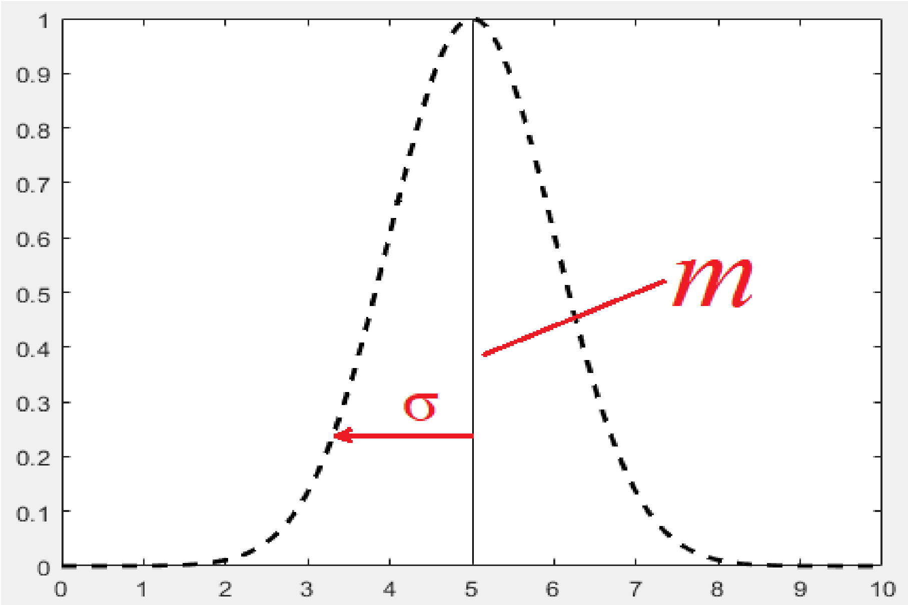


Figure 3.3: Type-1 Gaussian membership function where σ *denotes the standard deviation and* m *denotes the mean*

3.2.2 Type-1 Fuzzy Logic Controller

In the real world, the most common use of fuzzy sets is in fuzzy logic control applications. Type-1 Fuzzy Logic Controller System (T1FLCs) represents the Fuzzy Logic Controller (FLC) implementation. The FLC accepts crisp inputs from the system and output the crisp values, which are used to control the system (Figure 3.4).

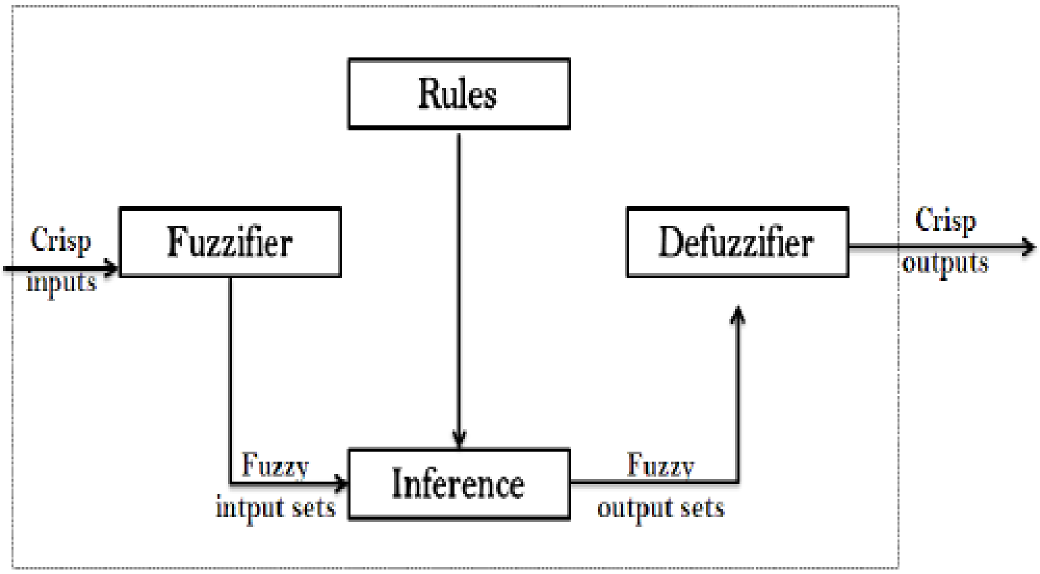


Figure 3.4: Structure of a type-1 FLC

Figure 3.4 shows the structure of a type-1 FLC working progress, in which the crisp inputs are fuzzified in the fuzzifier, and the resulting fuzzy input sets are processed by the inference engine using the rule base. The resulting fuzzy output sets are defuzzified in the defuzzifier to generate the crisp outputs, which are fed to the system. Each component of the type-1 FLC is described in detail below.

3.2.2.1 Fuzzifier

The fuzzifier generates the crisp input vector with p inputs $x = \left(x_1, \dots, x_p\right)^T \in X_1 \times \dots \times X_p \equiv X$ into type-1 fuzzy input sets F_x. As fuzzy logic controllers operate frequently in time critical applications, the Gaussian fuzzification is generally used for the fuzzification stage. In Gaussian fuzzification, the fuzzy input set contains the Gaussian membership function:

$$\mu_A(x) = exp\left[-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2\right] \tag{3-5}$$

3.2.2.2 Rule Base

Rules are used to describe linguistically the complex relationship between the input and output fuzzy variables in the form of IF–THEN statements. Typically, the rule is composed of several antecedents in the IF statement and one or several consequents in the THEN statement.

Assuming a type-1 FLC has a crisp input vector with p inputs $x = \left(x_1, \dots, x_p\right)^T \in X_1 \times \dots \times X_p$, and a crisp output vector with c outputs $y_1 \in Y_1 \dots \times y_c \in Y_c$ [Mendel, 2001], the ith rule for this Multi-Input Multi-Output (MIMO) FLC containing M rules can be written as follows:

$$R^i_{MIMO}: IF\ x_1 is\ F^i_1 and \dots and\ x_p is\ F^i_p\ THEN\ y_1 is\ G^i_1 \dots\ y_c is\ G^i_c, i = 1, \dots, M \tag{3-6}$$

3.2.2.3 The Fuzzy Inference Engine and Defuzzifier

The process of deducing the "strength" of these consequents from the strength of the antecedents is called rule inference. The fuzzy inference engine matches the fuzzy inputs and fuzzy outputs from the fuzzy sets based on the rule base. The most commonly used inference methods are correlation minimum, correlation product, and MIN–MAX [Heske, 1996]. In the defuzzifier step, FLC outputs a crisp value from the fuzzy output set(s). The capacity of inference engine and defuzzifier was reviewed in detail by [Mendel, 2001], but this is not of primary concern to the current research except as a component of the fuzzy logic control system.

3.3 Type-2 Fuzzy Logic

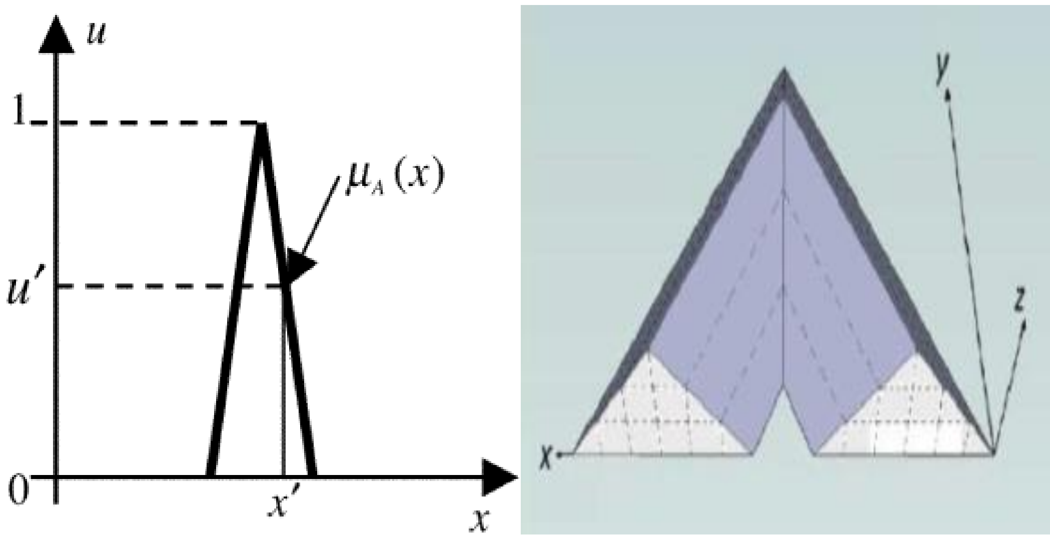
Type-2 fuzzy systems are based on the type-2 fuzzy sets, which are an extension of type-1 fuzzy sets. The main difference between type-1 and type-2 is in their fuzzy sets (FS) and membership functions (MF). First, T1FLS is two-dimensional; it is a plane coordinate in which the x-axis has crisp input values, and the y-axis is the unit from 0 to 1. Conversely, T2FLS is three-dimensional. In this section, type-2 fuzzy sets, especially interval type-2 fuzzy sets and interval type-2 fuzzy logic system are described in detail.

3.3.1 Type-2 Fuzzy Sets and FOU

In type-2 fuzzy logic system, the FS is different from type-1; in practical terms, it is an extension of type-1. While a type-1 fuzzy set F is characterized by its membership function μF(x), where x $\in$ X and $\mu F(x) \in [0,1]$, a general type-2 FS $\tilde{F}$ is characterized by a general type-2 MF $\tilde{\mu} F(x,u)$, where x $\in$ X and u $\in J_x \subseteq [0,1]$ [Mendel, 2005]. It can be expressed as:

$$\tilde{F} = \int_{x\in X}\int_{u\in J_x} \tilde{\mu}F(x,u)\ /(x,u) \qquad (3\text{-}7)$$

where $\iint$ denotes union over all admissible x and u. A 3-dimentional graph of a type-2 fuzzy set is shown in Figure 3.5(b), in comparison to the type-1 fuzzy set in Figure 3.5(a):



(a) type-1 (b) type-2

Figure 3.5: Fuzzy logic sets

It is not easy to express 3D images in two-dimensional forms. To visualize T2 FS, a good way is to sketch in 2D (plot) its footprint of uncertainty (FOU) of 2D domain of its FS. The heights of T2 MF sit atop FOU. Figure 3.6 shows a general type-2 FS in two-dimensional form. In this triangle, we call the top boundary upper membership function (UMF) and the inside boundary lower membership function (LMF). FOU is the grey area, which includes all embedded FS. Suppose there are N triangles, then at each value of x (on x-axis) they can be up to N grades (values): $MF_1(x)$ $MF_2(x)$ $MF_3(x) \ldots MF_N(x)$, each of MF grades has a weight assigned to it, W_{x1}, W_{x2}, $W_{x3} \ldots W_{xN}$ [Mendel, 2007; Wagner, 2010].

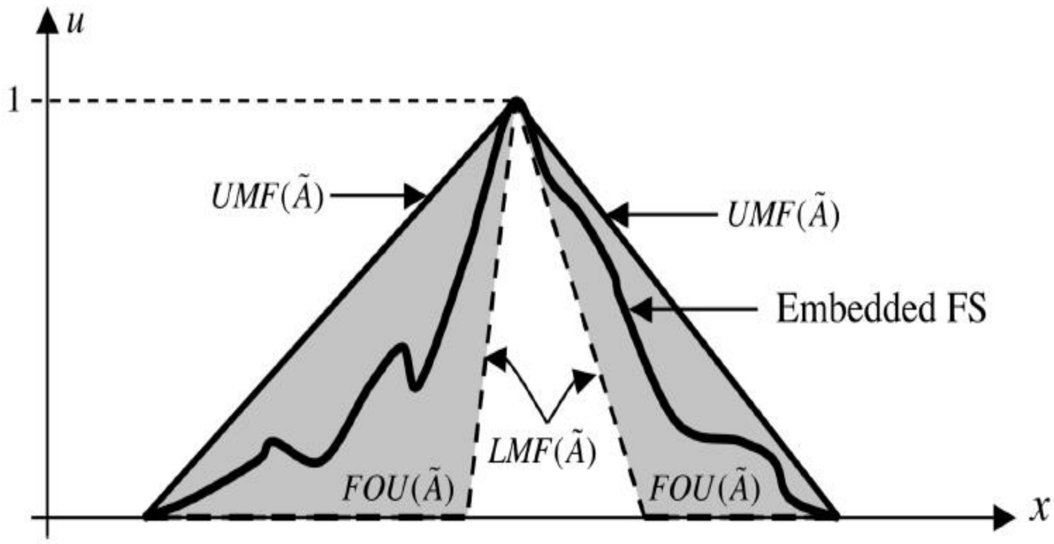


Figure 3.6: Front view of general type-2 fuzzy set

3.3.2 Interval Type-2 Fuzzy Logic

As described above, the Interval Type-2 Fuzzy Sets (IT2FSs) are a simplification of general type-2 sets, where for each primary membership the value in secondary membership is equal to 1. As such, considering Equation (3-7), the IT2FSs can be written as follows:

$$\tilde{I} = \int_{x \in X} \int_{u \in J_x} 1 \ \ \mathrm{d}(x, u), Jx \subseteq [0,1] \tag{3-8}$$

3.3.3 Interval Type-2 Fuzzy Logic Controller

The difference between T1 and T2 is not only in the FS; T2TLCs is also the extension from T1FLCs. In the T1FLCs, the whole system contains the fuzziflier, inference, defuzzifier and rules engine. The T2FLCs has a new model called "type-reducer" between the inference and the defuzzifier. It reduces the fuzzy output sets to a type-reduced set which is a type-1 set, then the data goes to the defuzzifier. The general flow chart is shown in Figure 3.7.

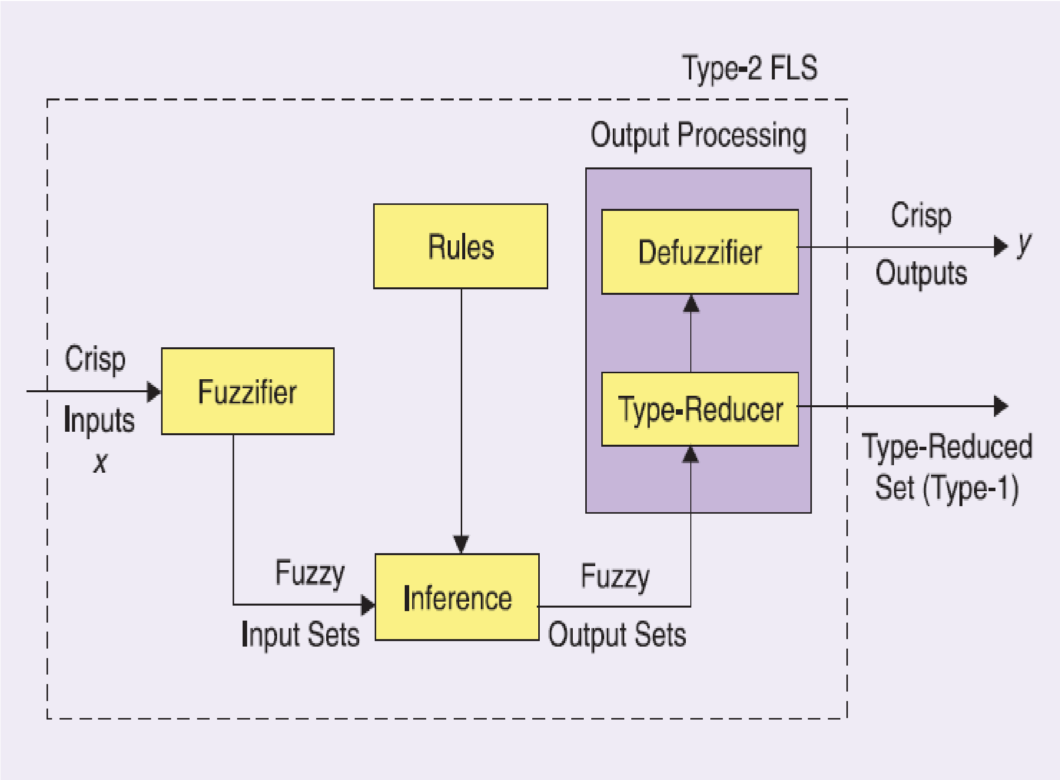


Figure 3.7: Type-2 FLCs

3.4 Fuzzy Logic Classification System (FLCS)

Fuzzy logic classification system (FLCS) is an extension applied approach of fuzzy logic based on the “Training-to-Learning” principle. It learns from the training data and class label which can give a prediction to determine which class the new coming input(s) belong to. FLCSs are also based on the fuzzy sets and rule base knowledge, however some components are changed and different from the diagram of fuzzy logic control system shown in Figure 3.7.

3.4.1 Background of Fuzzy Logic Applications in Classification

Fuzzy logic was first applied to solve classification problems in the 1990s. Fuzzy logic was only used in control system and associated applications, however scientists tried to generate fuzzy rules by learning from the data, aiming to make fuzzy logic systems intelligent, and able to handle similar data computation [Wang, 1991].

They then upgraded the original rule generating method to distributed representation [Ishibuchi, 1992].

The concept of rule confidence and support was first introduced in [Agrawal, 1994], however they were not initially relevant to fuzzy logic systems. People started to utilize these concepts to generate fuzzy rules in [Ishibuchi, 2001] and give specific generating progress in [Ishibuchi, 2005]. These fuzzy rules generating approaches led to the development of FLCS.

3.4.2 Theory of FLCS

3.4.2.1 Training the FLCS to learn the Confidence and Support

FLCS employs the concept of confidence and support in each generated fuzzy rules [Ishibuchi, 2005]. The confidence and support are two important parameters in the fuzzy logic classification system which have the similar functions like the “weight” in neural networks. In the training stage of FLCS, the system learns the needed information from the data in order to obtain the confidence and support for each fuzzy rule. The fuzzy rule in a fuzzy classification system is written as:

$$\text{Rule } R_q\text{: } IF\ x_1\ is\ A_{q1} \ldots x_n\ is\ A_{qn} \text{ then class } C_q \tag{3-9}$$

where in the rule q, it has n crisp inputs x and n fuzzy sets A_q. If each crisp input x belongs to each fuzzy set A_q, and the class label is C_q. The fuzzy system could learn the confidence and support from the rule q. The confidence can be viewed as measuring the validity. It can be also viewed as a numerical approximation of the conditional probability. The support can be viewed as measuring the coverage of training patterns. For each rule q, the confidence of R_q is represented $s(A_q \Rightarrow C_q)$ and the support of R_q is represented $c(A_q \Rightarrow C_q)$.

Assuming in the fuzzy set A_{qp}, it has M membership functions $\{A_{qp1}, \dots, A_{qpM}\}$, and for the input x_p, the system calculates the firing strength $w_q(x_p)$ from the fuzzy set A_{qp}, the $w_q(x_p)$ can be represented as below:

$$w_q(\boldsymbol{x_p}) = \min(\mu_{A_{qp1}(x_p)}, \dots, \mu_{A_{qpM}(x_p)}) \quad \textit{(3-10)}$$

where $w_q(x_p)$ represents the firing strength of the rule q to a crisp input x_p, $\mu_{A_{qp1}(x_p)}$ represents membership values of the crisp input x_p to the membership function A_{qp1} from the fuzzy sets A_{qp} in type-1 fuzzy sets for the rule q, and x_p is one crisp input of the input vector $\{x_1, \dots, x_n\}$, n is the size of input vector.

Meanwhile, for type-2 fuzzy sets, $\overline{w_q}(x_p)$ and $\underline{w_q}(x_p)$ represent the upper and lower firing strengths of the rule q to a crisp input x_p, which can be written as:

$$\overline{w_q}(\boldsymbol{x_p}) = min(\overline{\mu}_{A_{qp1}(x_p)}, \dots, \overline{\mu}_{A_{qpM}(x_p)}) \quad \textit{(3-11)}$$

$$\underline{w_q}(\boldsymbol{x_p}) = min(\underline{\mu}_{A_{qp1}(x_p)}, \dots, \underline{\mu}_{A_{qpM}(x_p)}) \quad \textit{(3-12)}$$

where $\overline{\mu}_{A_{qp1}(x_p)}$ and $\underline{\mu}_{A_{qp1}(x_p)}$ represent the upper membership value and lower membership value of the crisp input x_p to the upper and lower membership function of A_{qp1} respectively from the fuzzy sets A_{qp} in type-2 fuzzy sets for the rule q, and x_p is one crisp input of the input vector $\{x_1, \dots, x_n\}$, n is the size of input vector.

In the training stage, the fuzzy logic classification system learns the confidence and support from the training data. The confidence of the type-1 fuzzy logic classification system for the rule q can be obtained as follows [Ishibuchi, 2005]:

$$c\left(A_q \Rightarrow C_q\right) = \frac{\sum_{x_p \in \text{Class } C_q} w_q(x_p)}{\sum_{p=1}^{m} w_q(x_p)} \tag{3-13}$$

where *m* is the number of rules in the rule base. And the support of the rule *q*:

$$s\left(A_q \Rightarrow C_q\right) = \sum_{x_p \in \text{Class } C_q} w_q(x_p) \tag{3-14}$$

The confidence and support also employed by the type-2 fuzzy sets, with the upper confidence and lower confidence written as:

$$\overline{c}\left(A_q \Rightarrow C_q\right) = \frac{\sum_{x_p \in Class\ C_q} \overline{w_q}(\boldsymbol{x_p})}{\sum_{p=1}^{m} \overline{w_q}(\boldsymbol{x_p})} \tag{3-15}$$

$$\underline{c}\left(A_q \Rightarrow C_q\right) = \frac{\sum_{x_p \in Class\ C_q} \underline{w_q}(\boldsymbol{x_p})}{\sum_{p=1}^{m} \underline{w_q}(\boldsymbol{x_p})} \tag{3-16}$$

where *m* is the number of rules in the rule base. And the upper support and lower support can be written as:

$$\overline{s}\left(A_q \Rightarrow C_q\right) = \sum_{x_p \in Class\ C_q} \overline{w_q}(\boldsymbol{x_p}) \tag{3-17}$$

$$\underline{s}\left(A_q \Rightarrow C_q\right) = \sum_{x_p \in Class\ C_q} \underline{w_q}(\boldsymbol{x_p}) \tag{3-18}$$

The fuzzy logic classification system learns the confidence $s\left(A_q \Rightarrow C\right)$ and support $c\left(A_q \Rightarrow C\right)$ for rules in the training stage based on the input vector $\{x_1, \dots, x_n\}$, class label C_q and the fuzzy sets $\{A_{q1}, \dots, A_{qn}\}$.

3.4.2.2 Output Computing and Data Classifying

The confidence and support are employed by type-1 and type-2 FLCS during the testing phase to predict the class label for the new inputs. The system uses the confidence and support from the rule base during the testing phase to classify the incoming data. For type-1 FLCS, the crisp inputs would then be fuzzified and the firing strength of each rule computed using $w_q(V_U^i)$, and the strength of each class is given by:

$$O_{Class_h}(V_U^i) = \sum_q \left(w_q(V_U^i) * c(A_q \Rightarrow C_q) * s(A_q \Rightarrow C_q)\right) \quad \text{(3-19)}$$

where the output classification will be the class with the highest class strength. In type-2 FLCS, the crisp inputs would be fuzzified and we will compute the upper firing strength $\overline{w_q}(V_U^i)$ and lower firing strength $\underline{w_q}(V_U^i)$ respective to each fired rule. The strength of each class is given by:

$$\overline{O}_{Class_h}(V_U^i) = \sum_q \left(\overline{w_q}(V_U^i) * \overline{c}(A_q \Rightarrow C_q) * \overline{s}(A_q \Rightarrow C_q)\right) \quad \text{(3-20)}$$

$$\underline{O}_{Class_h}(V_U^i) = \sum_q \left(\underline{w_q}(V_U^i) * \underline{c}(A_q \Rightarrow C_q) * \underline{s}(A_q \Rightarrow C_q)\right) \quad \text{(3-21)}$$

$$O_{Class_h}(V_U^i) = \frac{\overline{O}_{Class_h}(V_U^i) + \underline{O}_{Class_h}(V_U^i)}{2} \quad \text{(3-22)}$$

The highest class strength would be the winner of all classes as the output classification.

3.5 Discussion

In this chapter, we presented a brief introduction and the background of the type-1 and type-2 fuzzy logic. The importance of fuzzy logic has been described. We have also produced the specific process of the fuzzy rules generation with confidence and support in order to learn from the data. This approach will be applied to our proposed system, to learn required knowledge from the data in order to summarize the information from the target videos.

Chapter 4 The Fuzzy Logic Classification System for Linguistic Video Summarization of Soccer Videos

In this chapter, we will demonstrate the process of our scenes classification system based on fuzzy logic and some basic techniques from computer vision in order to learn from video data and predict scenes of the soccer video.

4.1 Problem Description

Video scenes can be regarded as continuous sequences of images, but the classification problem is much more complicated than single image classification due to the dynamic nature of the video sequence and the associated changes in light conditions, background, camera angle, occlusions, and indistinguishable scene features, etc. Video scene classification forms the basis of linguistic video summarization, which is an open research problem with major commercial importance. Soccer video scenes present added challenges due to the existence of specific objects and events which have high similar features like audience and coaches as well as being constituted from a series of quickly changing and dynamic frames with small inter-frame variations.

In addition, there is an added difficulty associated with the need to have lightweight video classification systems which can work in real time, with the massive data sizes associated with video analysis applications, especially in soccer videos. Furthermore, video classification can turn into an untenable labour-intensive human task, whereby humans watch or manually extract key video information, warranting urgent attention to devise automated solutions [Song, 2016a] [Song, 2016b].

4.1.1 The Objective of Proposed System

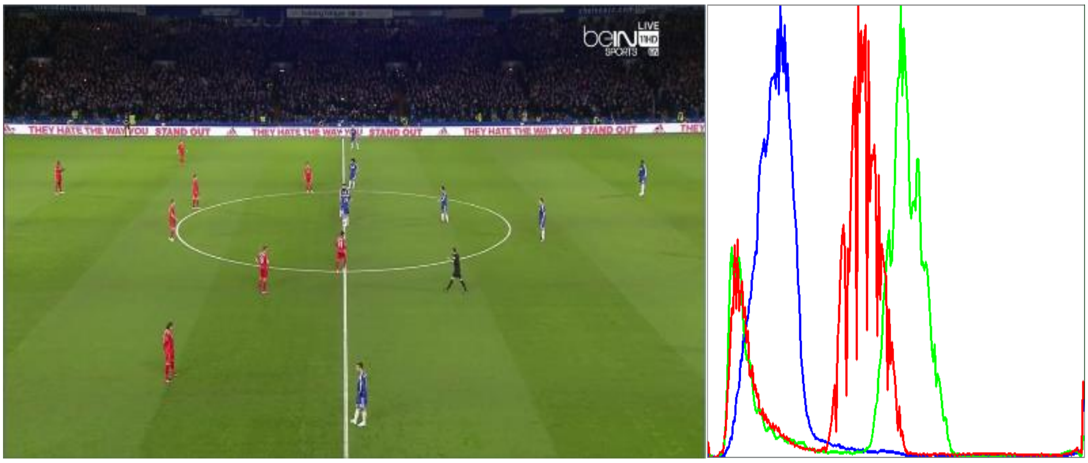
The proposed system aims to process the real video data as inputs and output a label to predict the current video scene in real-time based on the well-trained FLCS. The system has to complete three phases when testing: video data transform, feature extraction, fuzzy classification system computation. However, before the real-time video data testing, the system has to be trained, learning from the required training data. The system is also applied on out-of-range data to test compatibility, flexibility and practicality.

4.1.2 Video Data Pre-Processing

In practical terms, video data can be segmented to the continuous images. In most types of living and recorded video, they have 25 frames per second (FPS), which represents 25 continuous images shown to the viewer in each second. Thus, in our experiments, the system learns from each frame and gives the incoming target image the predicted label. The images have to be extracted and translated into numerical values for the system needs.

4.1.2.1 Feature Extraction

The features required by the system are extracted from the original video frame, then the histogram distances representing the difference between two images are computed and fed as crisp inputs to the multi classification systems. Figure 4.1 shows colour histograms extracted from represented original video scenes.



(a) Centre field scene

(b) People scene

(c) Player close-up scene

Figure 4.1: Scenes and histograms

The training and testing data consist of three significant scenes classes from broadcast soccer videos: which are centre, close-up and people scenes. The centre scenes, as shown in Figure 4a, represent the scenes in the centre field with an overview from afar, which comprises most soccer video scenes. Figure 4b represents people scenes, which include audience and coaches with no direct relationship to the football match. Figure 4c shows the close-up scenes where we see a zoom shot of one or several players during the match.

To get the inputs the system needs, we process the frame histograms first. The system requires highly summarized inputs and histograms that only the describe the information of colour distribution in graphical form. To enable histograms processing, histogram distance is introduced and some mathematical characteristics are applied before feeding this to the multi-classification system. The idea of computing two value distances instead of a single value is widely used in some complicated applications. First, we compute dcs, which is the colour histograms' Chi-square correlations, written as:

$$dcs(H_a, H_b) = \sum_{i=0}^{I} \frac{(h_{ai} - h_{bi})^2}{h_{ai}} \quad \text{(4-1)}$$

where H_a and H_b are two different histograms of two different frames in the same colour channels. Generally, H_a is a standard scene image that represents the scene. H_b is one frame from the training video. I is the range of pixel level (generally 256) and i is the sequence. Thus h_{ai} and h_{bi} are ith values for two image colour histograms H_a and H_b respectively. dco represents the correlation between two histograms H_a and H_b. It can be written as:

$$dco(H_a, H_b) = \frac{\sum_{i=0}^{I}(h_{ai} - \overline{H_a})(h_{bi} - \overline{H_b})}{\sqrt{\sum_{i=0}^{I}(h_{ai} - \overline{H_a})^2 \sum_{i=0}^{I}(h_{bi} - \overline{H_b})^2}} \tag{4-2}$$

where $\overline{H_x}$ represents the average colour level of a histogram from image x; it can be written as:

$$\overline{H_x} = \frac{1}{N}\sum_{i=0}^{I} h_{xi} \tag{4-3}$$

where *N* is the total number of histogram bins, and h_{xi} is *i*th values. $\overline{H_a}$ and $\overline{H_b}$ are average colour levels for image histograms H_a and H_b. dbd represents the Bhattacharyya distance, which is widely used in statistics to measure the difference between two discrete or continuous probability distributions. It can be written as:

$$dbd(H_a, H_b) = \sqrt{1 - \frac{1}{\sqrt{\overline{H_a H_b} N^2}} \sum_{i=0}^{I} h_{ai} h_{bi}} \tag{4-4}$$

where i, I, N, $\overline{H_a}$, $\overline{H_b}$, h_{ai} and h_{bi} are the same as in Equation (4-2). din represents the intersection, which denotes the similarity between the two images. din could be written as:

$$din(H_a, H_b) = \sum_{i=0}^{I} min(h_{ai}, h_{bi}) \tag{4-5}$$

where i, I, $\overline{H_a}$, and $\overline{H_b}$ are the same as in Equation (4-2).

4.1.2.2 System Inputs Vector

To train and test the classification system in sports video scenes, the inputs of the system include Chi-square distance (*dcs*) correlation (*dco*) intersection (*din*) and Bhattacharyya distance (*dbd*) for the colours red (*R*), green (*G*) and blue (*B*). The histogram generally summarizes the colour distribution as a symbol of image features and is widely used to describe images and videos. RED, GREEN and BLUE (RGB) are the basic colours of all images from which other colours are made. To train and test the classification system in sports video scenes, the inputs of the system include dcs, dco, din, dbd. Thus, the primary input vector V_T^i of the scene classification system at the ith frame can be written as:

$$V_T^i = \left\{ \begin{matrix} dco_B^i, dco_G^i, dco_R^i, dcs_B^i, dcs_G^i, dcs_R^i, \\ din_B^i, din_G^i, din_R^i, dbd_B^i, dbd_G^i, dbd_R^i \end{matrix}, O^i \right\} \tag{4-6}$$

where O^i is the class label of the scene at i frame.

4.2 The Proposed System of Type-1 Fuzzy Logic Classification

Fuzzy sets and their membership functions are always the core of any fuzzy logic system. Fuzzy C-means (FCM) clustering technique is employed to create the type-1 fuzzy sets and associated membership functions first [Bezdeck, 1984]. The system then uses training data and type-1 fuzzy sets to generate the rule base with the associated confidence and support (Figure 4.2).

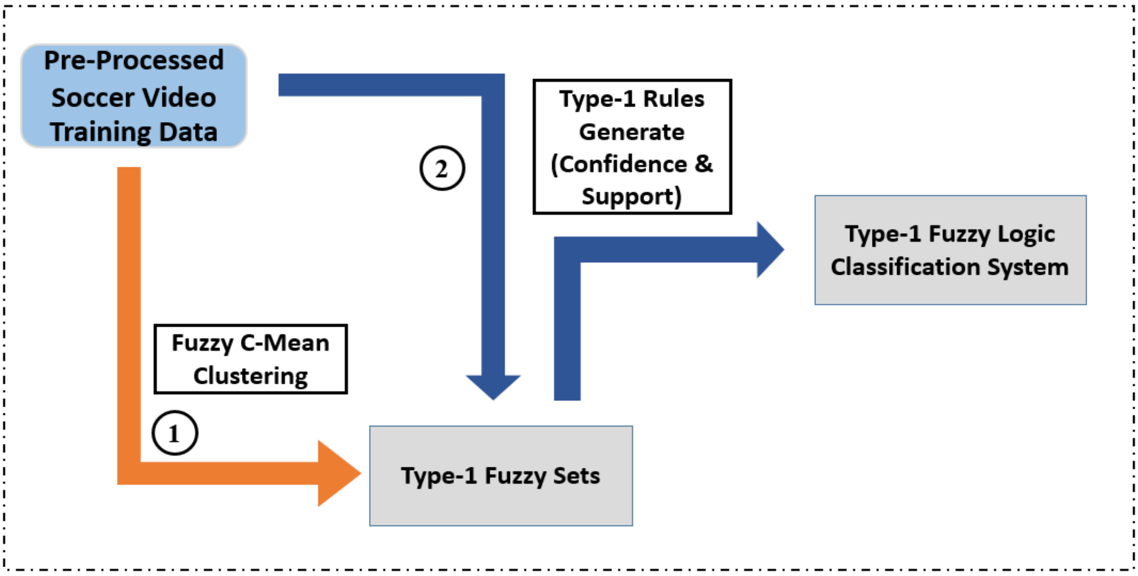


Figure 4.2: The progress of type-1 FLCS build

4.2.1 Type-1 Fuzzy Sets for Scene Classification system

4.2.1.1 Fuzzy C-Mean Clustering

Fuzzy sets and their membership functions are always the core of any FLCS. The membership functions are generally set by human prior knowledge in some experiments and real-world applications. In this paper we employ the Fuzzy C-means (FCM) [Bezdeck, 1984] clustering technique to obtain parameters of the fuzzy sets used in the fuzzy classification system. FCM algorithm is an unsupervised clustering method to classify the unlabelled data by minimizing an objective function, using fuzzy partitioning such that each data point belongs to a cluster to a certain degree modelled by a membership degree in the range [0, 1], which indicates the strength of the association between that data point and a particular cluster centroid. Let $X = \{x_1, x_2, \dots, x_N\}$ be a set of given data points, and $V = \{v_1, v_2, \dots, v_C\}$ be a set of cluster centres. The idea of the FCM is to partition the N data points into C clusters based on minimization of the following objective function [Bezdeck, 1984]:

$$J(X; U, V) = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - v_j \right\|^2 \qquad \text{(4-7)}$$

where *m* is used to adjust the weighting effect of membership values, $\|\cdot\|$ is the Euclidean norm modelling the similarity between the data point and the centre, and $U = (u_{ij})_{C\times N}$ is a fuzzy partition matrix, subject to [Bezdeck, 1984]:

$$\sum_{i=1}^{N} u_{ij} = 1, \forall j = 1, \dots, C \qquad \text{(4-8)}$$

$$u_{ij} \in [0,1], \forall i = 1, \dots, N, \forall j = 1, \dots, C \qquad \text{(4-9)}$$

where u_{ij} is the membership degree of point u_i to the cluster j.

FCM is performed via an iterative procedure with Equation (4-7), updating u_{ij} and c_j. In this paper, FCM is used to compute the clusters of each feature to generate the type-1 fuzzy membership functions for the FLCS. Figure 4.3 shows how we approximate the raw function generated by the FCM with a Gaussian type-1 fuzzy set.

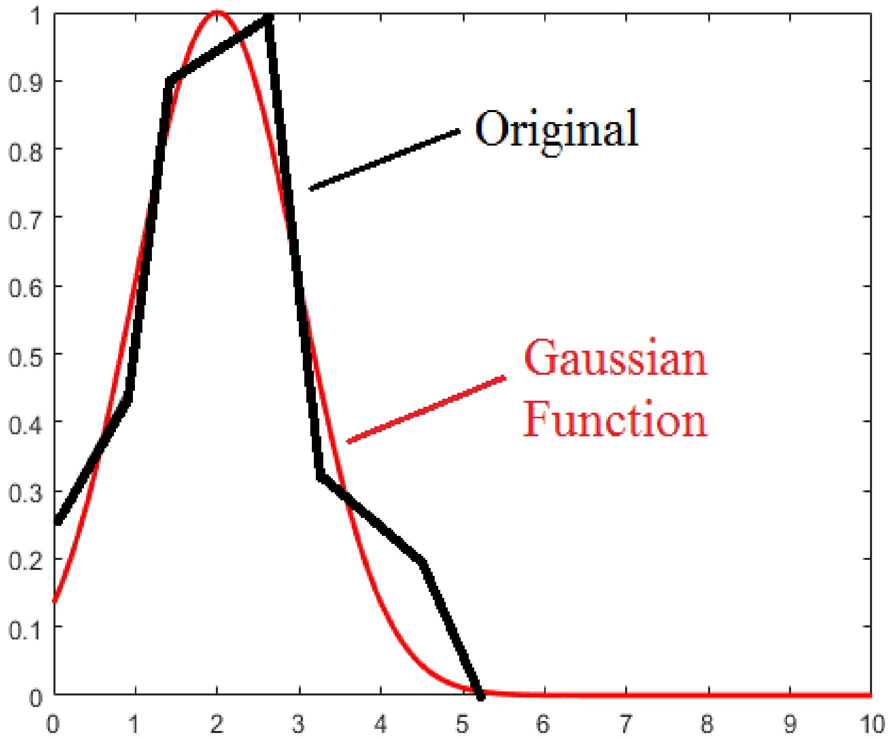
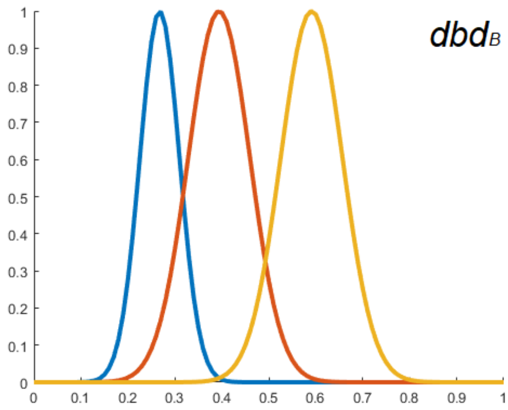


Figure 4.3: Schematic diagram of approximating the raw function generated by FCM with a Gaussian type-1 fuzzy set

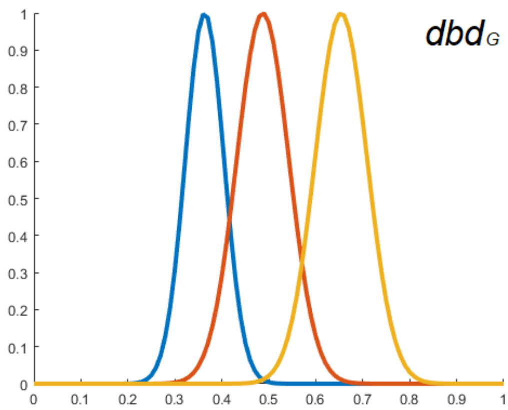
4.2.1.2 Generated Type-1 Fuzzy Sets

The FCM algorithm is used to cluster the data for each input into three (Low, Medium, High) Gaussian fuzzy sets (clusters). In this type-1 FLCS, it has 12 inputs, as described in Equation (4-6), thus, our FLCS has 12 fuzzy sets and each set is created by each relevant input.

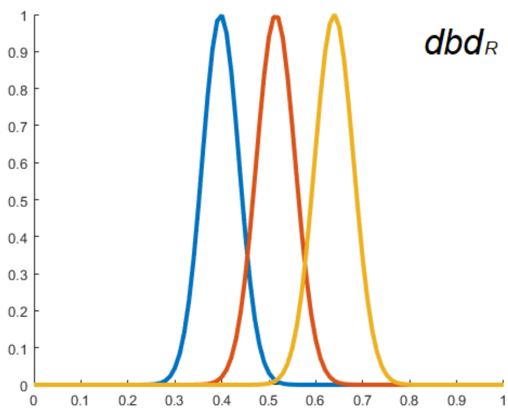
The primary type-1 fuzzy sets are shown in Figure 4.4, representing the results of fuzzy c-mean. Each fuzzy set is generated from each input vector. For example, fuzzy set dco_B is created by the input sequences dco_B^i.



(a) Fuzzy set of dbd_B

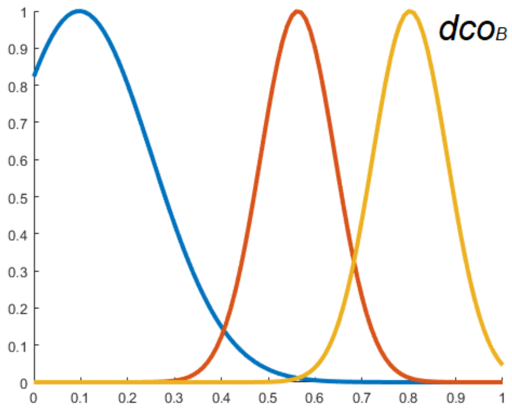


(b) Fuzzy set of dbd_G

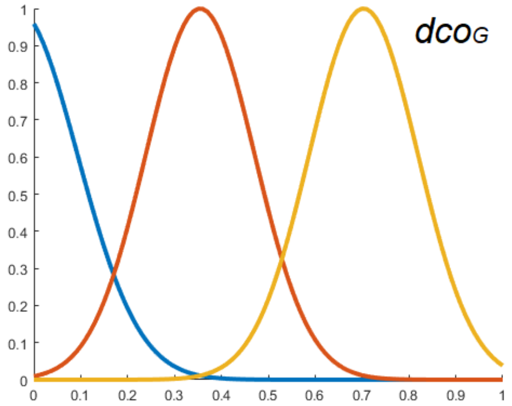


(c) Fuzzy set of dbd_R

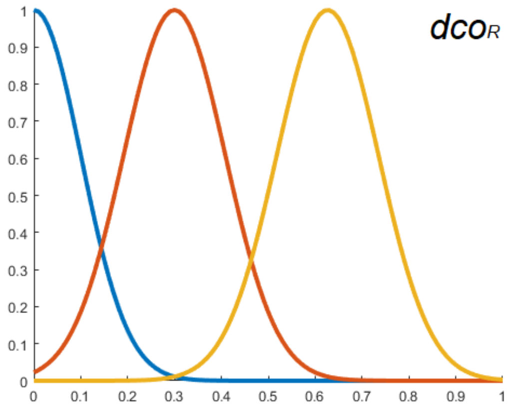
Figure 4.4: Type-1 fuzzy sets and membership functions for FLCS



(d) Fuzzy set of dco_B

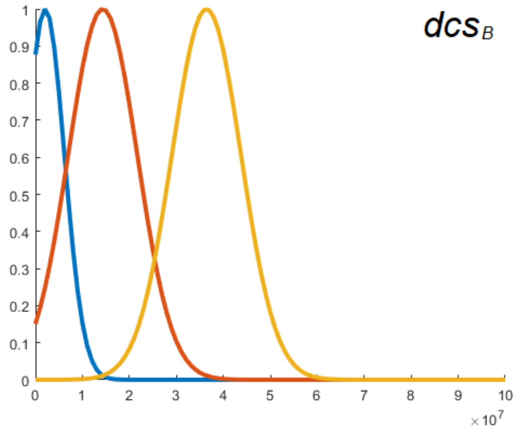


(e) Fuzzy set of dco_G

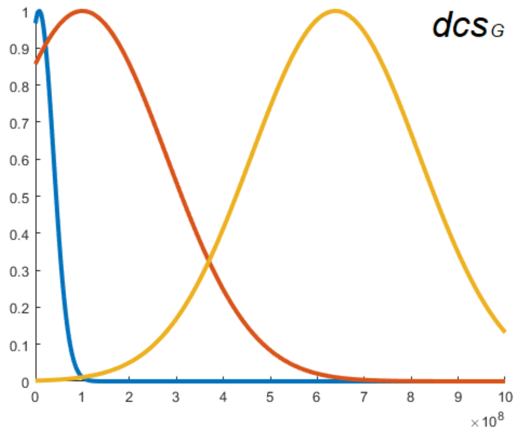


(f) Fuzzy set of dco_R

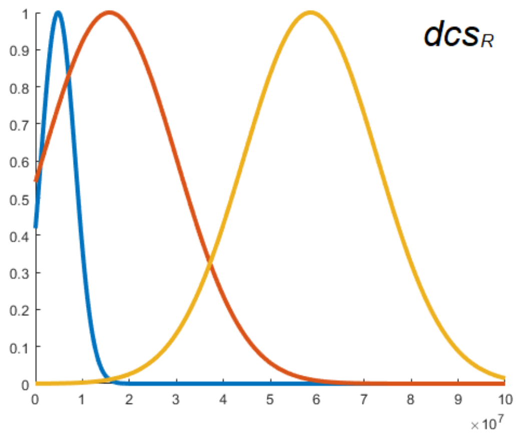
Figure 4.4: Type-1 fuzzy sets and membership functions for FLCS (cont.)



(g) Fuzzy set of dcs_B

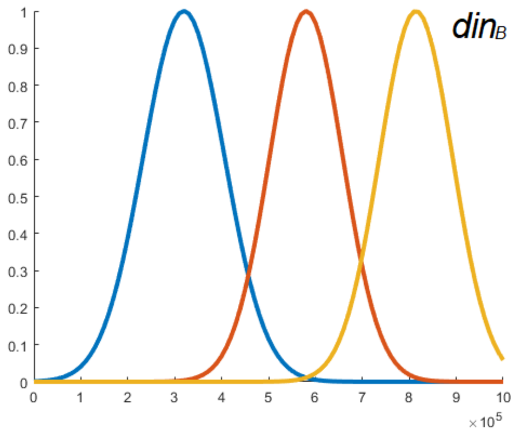


(h) Fuzzy set of dcs_G

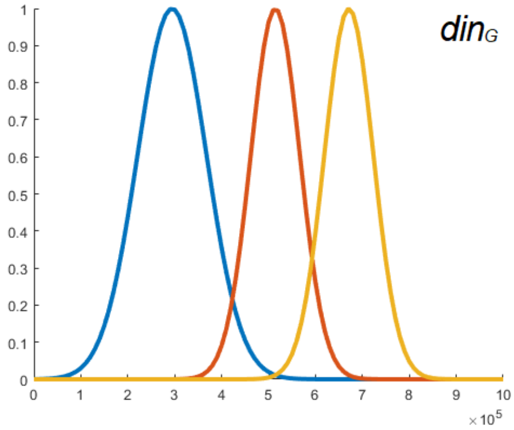


(i) Fuzzy set of dcs_R

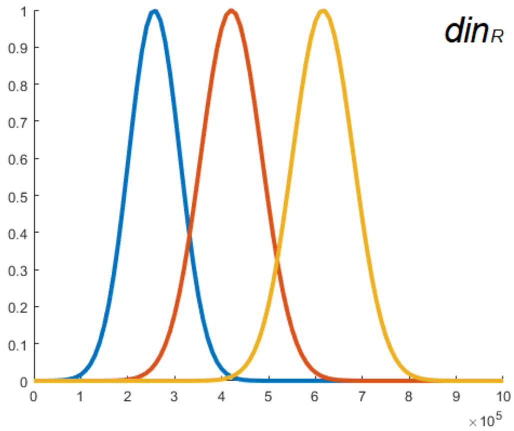
Figure 4.4: Type-1 fuzzy sets and membership functions for FLCS (cont.)



(j) Fuzzy set of din_B



(k) Fuzzy set of din_G



(l) Fuzzy set of din_R

Figure 4.4: Type-1 fuzzy sets and membership functions for FLCS (cont.)

4.2.2 Type-1 Rule Base

The Type-1 Fuzzy rules are generated from the training data and type-1 fuzzy sets. The specific fuzzy rules generation process is described in section 3.4. The fuzzy rules for the type-1 fuzzy classification system have the rule antecedents and associated confidence and support. The fuzzy classification system rule base can be represented as shown in Figure 4.5.

$\boldsymbol{m_1^1}$	$\boldsymbol{\dots}$	$\boldsymbol{m_p^1}$	$\boldsymbol{C_1^1}$	$\boldsymbol{S_1^1}$	$\boldsymbol{\dots}$	$\boldsymbol{C_q^1}$	$\boldsymbol{S_q^1}$	$\boldsymbol{\dots}$	$\boldsymbol{m_1^R}$	$\boldsymbol{\dots}$	$\boldsymbol{m_p^R}$	$\boldsymbol{C_1^R}$	$\boldsymbol{S_1^R}$	$\boldsymbol{\dots}$	$\boldsymbol{C_q^R}$	$\boldsymbol{S_q^R}$		

Figure 4.5: Rule base of type-1 FLCS

As shown in Figure 4.5, m_j^r are the antecedents and C_k^r and S_k^r are the confidence and support respectively of each rule, where $j = 1,\dots, p$, j is the number of antecedents; $k = 1,\dots, q$, k is the number of output classes labels; and $r = 1,\dots, R$, and r is the number of the rules to be tuned. The input vector of the T1FLCS is 12, each input vector has unique fuzzy set and the label of output has 3 classes, thus the p =12 and q=3 for the figure 4.5

4.3 The Upgrade of Type-1 FLCS to Type-2 FLCS

The type-2 FLCS is generated from the type-1 FLCS based on the fuzzy sets and membership functions upgraded. The upgrade process is shown in the Figure 4.6. Type-2 fuzzy sets firstly are created by the type-1 fuzzy sets applied in the type-1 FLCS. In each fuzzy set, the original type-1 Gaussian membership functions are added the upper and lower standard deviation, then the type-2 fuzzy sets are employed to the training process of the type-2 fuzzy rules generation, to create the upper and lower confidence and support, respectively.

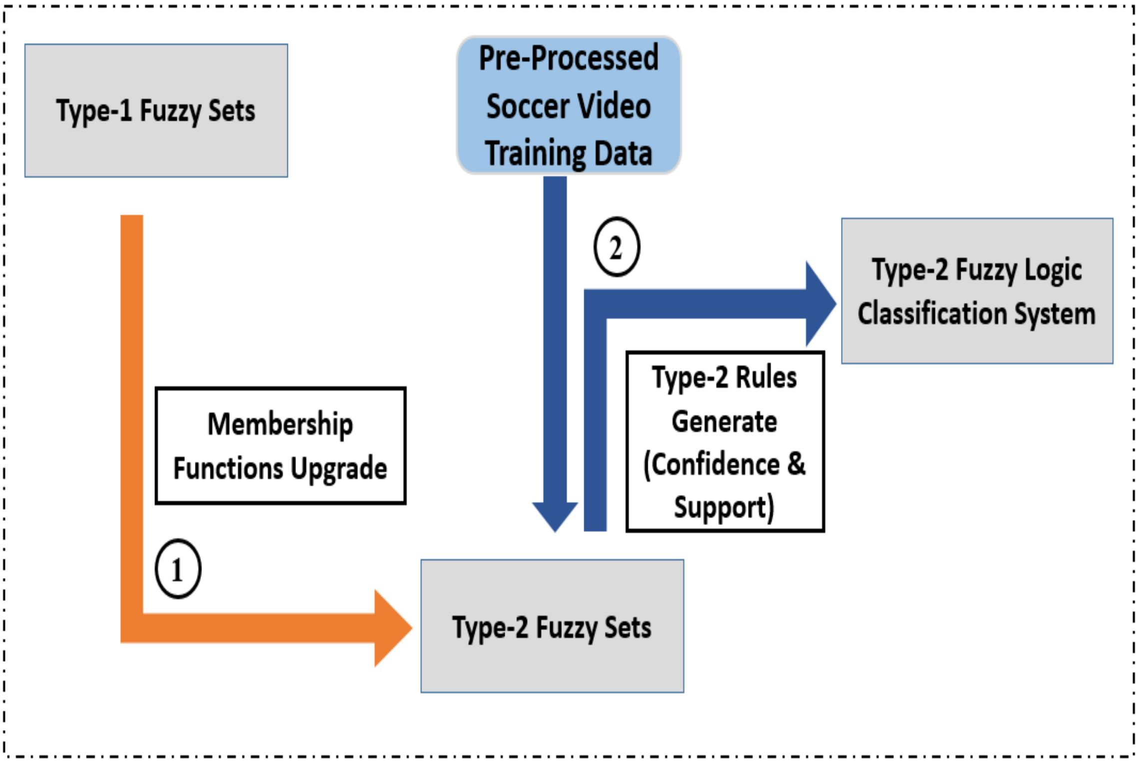
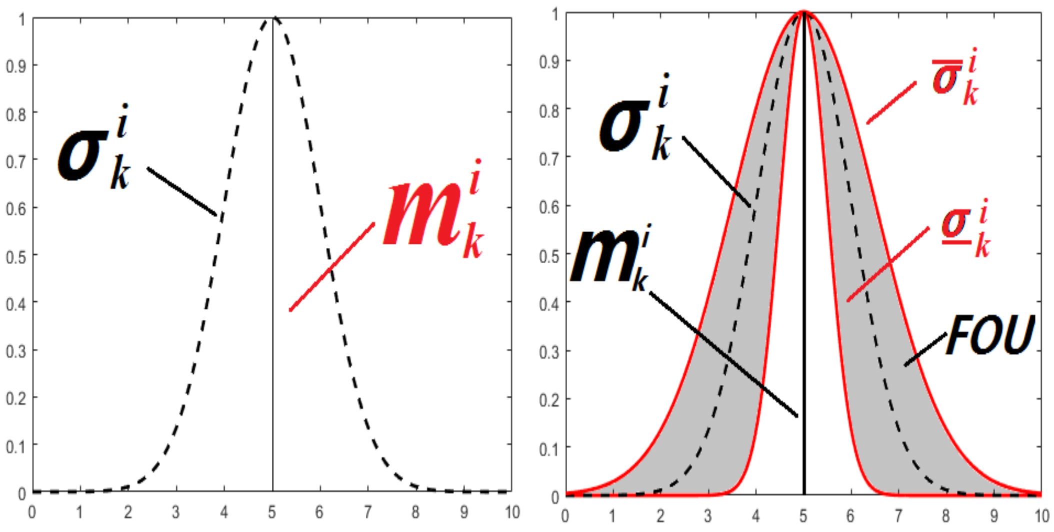


Figure 4.6: Process of Type-2 FLCS build

4.3.1 Type-2 Fuzzy Sets Generated

To obtain the interval type-2 fuzzy set from the extracted type-1 fuzzy set, we blur the mean of the type-1 fuzzy set σ_k^i equally to the upper and lower with a distance d_k^i, to generate the upper and lower standard deviation ($\overline{\sigma_k^i}$ and $\underline{\sigma_k^i}$ respectively) for the generated Gaussian type-2 fuzzy set with uncertain standard deviation (Figure 4.7).



(a) (b)

Figure 4.7: (a) Type-1 fuzzy sets, (b) Type-2 fuzzy sets

The mean value of $\boldsymbol{m_k^i}$ for the generated type-2 fuzzy set will be the same as the corresponding type-1 fuzzy set. Thus the upper ($\overline{\boldsymbol{\mu}}_{\boldsymbol{A}}(\boldsymbol{x})$) and lower ($\underline{\boldsymbol{\mu}_{\boldsymbol{A}}}(\boldsymbol{x})$)MFs for the generated type-2 fuzzy set set can be written as:

$$\overline{\mu}_A(x) = N\left(\boldsymbol{m_k^i}, \overline{\boldsymbol{\sigma_k^i}}; x\right) \qquad (4\text{-}10)$$

$$\underline{\mu_A}(x) = N\left(\boldsymbol{m_k^i}, \underline{\boldsymbol{\sigma_k^i}}; x\right) \qquad (4\text{-}11)$$

The type-2 FLCS employed the constant value of the d_k^i. However, the primary type-2 FLCS for the soccer video scenes classification is not fitting in the best performance. The value d_k^i for each interval type-2 fuzzy set can be optimized via BB-BC (section 5.2). The type-2 fuzzy logic classification employs the type-2 fuzzy sets and learnt type-2 rule base, with the associated confidence and support (section 3.4).

4.3.2 Type-2 Rule Base

Type-2 fuzzy rules are the extension generated from the training data and type-2 fuzzy sets. The fuzzy classification system rule base can be represented as shown in Figure 4.8.

m_1^1	...	m_p^1	$\overline{C_1^1}$	$\underline{C_1^1}$	$\overline{S_1^1}$	$\underline{S_1^1}$	...	$\overline{C_q^1}$	$\underline{C_q^1}$	$\overline{S_q^1}$	$\underline{S_q^1}$	...	m_1^R	...	m_p^R	$\overline{C_1^R}$	$\underline{C_1^R}$	$\overline{S_1^R}$	$\underline{S_1^R}$	...	$\overline{C_q^R}$	$\underline{C_q^R}$	$\overline{S_q^R}$	$\underline{S_q^R}$

Figure 4.8: Type-2 rule base

As shown in Figure 4.8, m_j^r are the antecedents and $\overline{C_k^r}$ and $\underline{C_k^r}$ are the upper and lower confidence of the rule; $\overline{S_k^r}$ and $\underline{S_k^r}$ are the upper and lower support respectively of each rule, where $j = 1,\ldots, p$, $p=12$ is the number of antecedents; $k = 1,\ldots, q$, $q = 3$ which the number of output classes labels; $r = 1,\ldots, R$, and R is the number of the rules to be tuned.

4.4 Experiments and Results

4.4.1 Data of Video Scenes Pre-processing

We performed various experiments using the selected data from over 30 football videos from the English, Spanish, German and French soccer leagues as our database. All videos had a resolution of 1280*720 pixels with frame rates of up to 25 frames/second. First, we separated the soccer match videos to two groups, 85% of the videos is used to be our “training and testing”, the other 15% is used to be the “out-of-range” dataset. The “training and testing” dataset and “out-of-range” dataset have no common video sources. And the soccer videos in “out-of-range” dataset has the significant difference from “training and testing” dataset. For example, in our “out-of-range” dataset, there are some videos from the Spanish afternoon soccer matches which have the strong sunshine on the ground and players, and the appearances of the objects in the videos have much more brightness colour than normal soccer videos. These videos are not in our training and testing dataset.

The purpose of building the “out-of-range” dataset is to evaluate the flexibility of the systems if the system is able to give the correctly prediction for the inputs which have the differences from the training and testing data. However, it is not the transferring learning process, the system would not obtain any update in the evaluation process of the “out-of-range” dataset. The predict results of the system in “out-of-range” dataset give us the accuracy which compare to the “testing” dataset, in order to describe the flexibility of the system when process different data.

The total video data comprises over 800,000 frames, “training and testing” data set takes 85% of them where were randomly divided into the training data (70%), used to build the system’s needed parameters and the rule base, and testing data (15%), used to evaluate the classification accuracy of the system. Approximately 15% of matches to be the “out-of-range” dataset in order to test the flexibility and capability of classification system when running with the video data from difference video sources. In order to train, test and evaluate our systems, we have to pre-process the sources of “videos” to be the readable data of numerical values. Thus, we developed a GUI program named “VideoInfo” to load the videos and chop videos to small video clips as the first step. Secondly, we labelled the small video clips with the scene classes (“Centre Field”, “Players Close-up” and “Audiences & other people”). At last, we extracted features of the video clips to capture histograms and compute distance by using OPENCV library.

The process of the experiments for the fuzzy logic scenes classification in soccer video generally can be regarded as a two-stage operation, video data pre-processing and type-1 FLCS training (section 4.2) and testing.

The “VideoInfo” GUI program is coded for chopping the needed data from the download video sources. This program imported the original soccer video and make them into short time video slips (normally 3-15 seconds) which representing groups of continuous video frames. The continuous video clips generally belong to the same scene. However, the double check is needed during the label marking in the video clips. The mixed scenes video will be removed from the catalogue of classes, and the data will be separated into the training dataset, testing dataset, and “out-of-range” dataset, which are fed to train the type-1 FLCS. The training and testing video clips come from the same video resources, and the “out-of-range” dataset is from the different video resources. In the experiment, the “out-of-range” dataset is used to test the flexibility of FLCS, where trained by a different data source.

4.4.1.1 The process of building the dataset

The “VideoInfo” GUI program is a user interface similar to a video player, with the functions of video play, pause, and resume buttons, other basic functions and a window to show the current frame histogram with RGB colour distribution information. The video is loaded after the user selects the directory and the time information is stored in a text file to record the time information of each video clip. The screenshot of “VideoInfo” GUI program interface is shown in Figure 4.9.

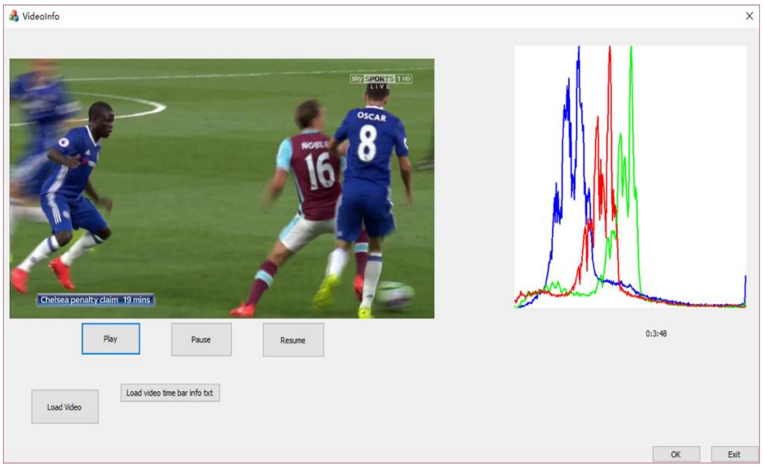
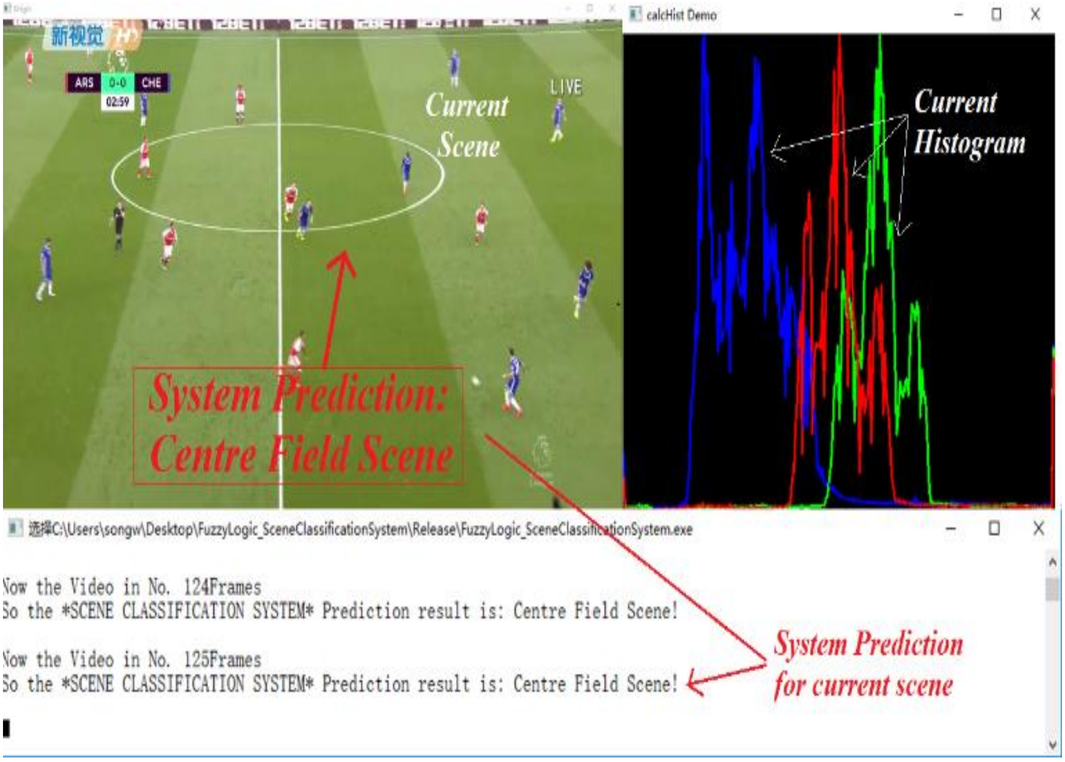


Figure 4.9: The "VideoInfo" GUI program

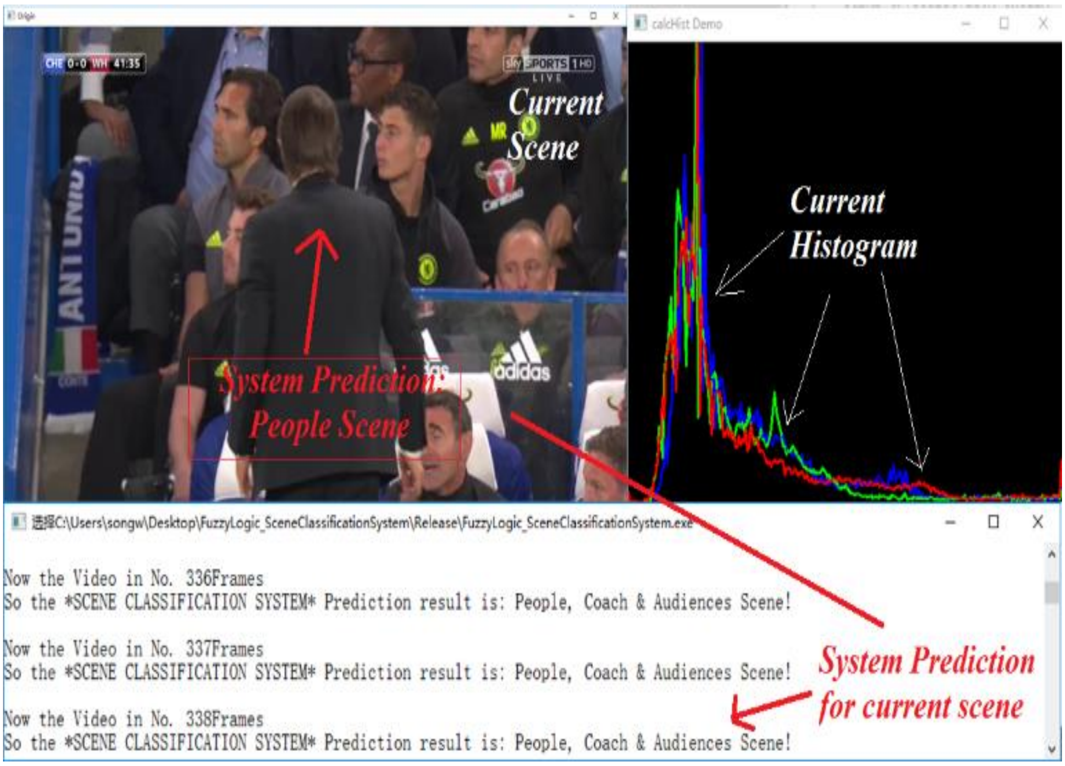
"VideoInfo" GUI program system only complete the segmentation function to chop the original video source to short time video clips. The videos are separated by the "VideoInfo" due to the rapidly changes between two continues frames. The video can be regarded as continuous sequences of images; thus two histograms of the image scenes will have significantly changes especially in the frame of current scene changes rapidly to another scene in the video. However, this segmentation could not build good enough dataset for our system required, the video clips generally are mixed with different scenes. We have to remove the scenes-mixed video clips manually and labelling the qualified video clips with the scene class.

4.4.1.2 Video Scenes Classification System

Figure 4.10 presents some screenshots of our system operation for scene classification, showing the prediction class label in "Centre" scene detection (a) and "People" scene detection (b). This program is the testing phase of type-1 FLCS prediction and also was used for type-2 FLCS and back propagation (BP) neural network to output the current video scene and predict its class. The prediction is shown as the text in the bottom of the program, and the current scene is in the top-left, with its histogram in the top-right.



(a) “Centre Field” scene detection



(b) “People” scene detection

Figure 4.10: Scene classification system

4.4.2 The Prediction Accuracy of the Scene Classification Systems

The experimental results are described in the tables below. We evaluate the system performance mainly based on the prediction accuracy of testing dataset and out-of-range dataset. We built a comparison group from Back Propagation Neural Network (BP-NN) classification system to learn the needed information and give the predictions of the training dataset, testing dataset and "out-of-range" dataset. The group of "BP-NN" is consist of one input layer, two hidden layers, one output layers and biases. The size of each hidden layer is the triple of input nodes which is 36 nodes in each hidden layer. We present a diagram of our BP-NN structure in figure 4.11, we use the circles to express the nodes and the lines to express the weight between the nodes to nodes.

In the BP-NN training, we employ the SoftMax training approach to complete the classification task in neural networks training. The weights between nodes to nodes are randomly initialized to 0 to 1 and the activate functions are "sigmod". The training process of the neural network is the back-propagation and the training process would be finished while the value of the loss function decreased to a stable level.

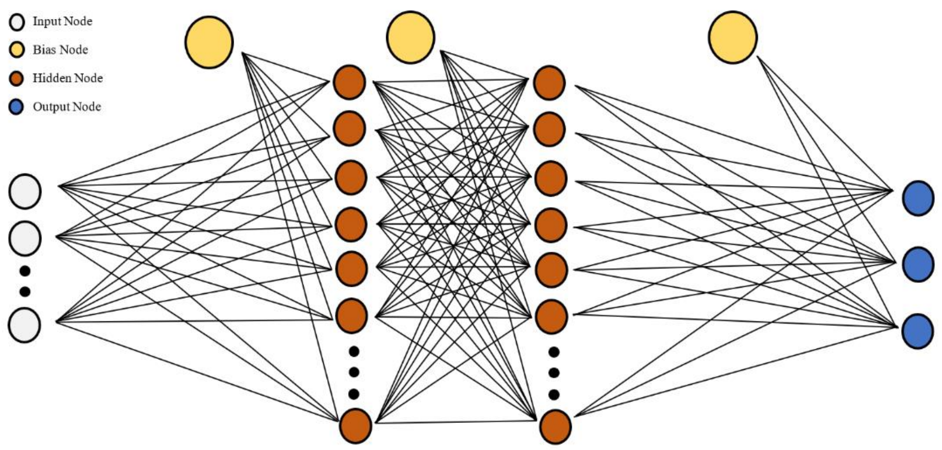


Figure 4.11: Diagram of BP-NN Scene classification system

Table 4.1 shows the comparison group from Back Propagation Neural Network (BP-NN) classification system**.** Table 4.2 shows another comparison over the training dataset, testing dataset and out-of-range dataset of the proposed scenes classification system using T1FLCS. The rule base of the T1FLCS in Table 4.2 is the full rules type-1 fuzzy system has 296,028 rules which are generated in the training process.

As can be seen from Table 4.2, T1FLCS outperforms the BP-NN from Table 4.1 on most classifications of overall, average and individual class. The BP-NN only outperforms the T1FLCS in the prediction accuracy of the scene “Close-up” in testing dataset and out-of-range dataset. However, in the other groups of the comparison, T1FLCS is much better than BP-NN, the difference value of overall accuracy in testing dataset and out-of-range dataset are both outperforms over 12% meanwhile the difference value of average accuracy in testing dataset and out-of-range dataset are outperforms over 16% and 7% respectively. As can be seen from Tables 4.2 and Table 4.3, the T2FLCS outperforms the T1FLCS in testing dataset and out-of-range dataset. T2 achieved an overall accuracy uplift of 0.4% and 0.1% compared to the T1 systems for the full rule base in the testing dataset and out-of-range dataset. In the prediction of average accuracy, T2 achieved an overall accuracy uplift of 0.4% and 0.9% compared to the T1 systems in the testing dataset and out-of-range dataset. However, the primary results are not optimized, and the fuzzy based system can be improved the performance by optimizing the parameters. An optimization process is introduced in the next chapter, for FLCSs (both type-1 and type-2) in relation to the fitness of fuzzy sets and the rule base.

BP-NN	Centre	People	Close-up	Overall	Average
Training dataset	73.1645%	68.8574%	70.7103%	71.9346%	70.9107%
Testing dataset	61.4736%	63.6586%	73.4074%	64.1461%	66.1798%
Out-of-range dataset	64.5819%	47.7024%	39.5120%	56.6065%	50.5987%

Table 4.1: BP-NN classification system for scenes classification

Type-1 FLCS	Centre	People	Close-up	Overall	Average
Training dataset (FLCS in full rule base)	95.3804%	85.3642%	74.1028%	90.1855%	84.9491%
Testing dataset (FLCS in full rule base)	93.7818%	83.5201%	70.6501%	87.1553%	82.6506%
Out-of-range dataset (FLCS in full rule base)	84.5601%	56.9450%	32.6095%	69.6532%	58.0382%

Table 4.2: Type-1FLCS scenes classification

Type-2 FLCS	Centre	People	Close-up	Overall	Average
Training dataset (FLCS in full rule base)	94.6135%	87.1903%	72.3802%	89.7939%	84.7280%
Testing dataset (FLCS in full rule base)	93.9607%	85.6233%	69.5595%	87.5675%	83.0478%
Out-of-range dataset (FLCS in full rule base)	83.7021%	54.7832%	38.5502%	69.6641%	59.0118%

Table 4.3: Type-2 FLCS scenes classification

4.5 Discussion

In this chapter, we presented our fuzzy logic scenes classification system building process. First, we introduced the feature extraction in section 4.1, presented the extraction process, and produced the input vector for the system that is to be learnt. We then employed the fuzzy c-mean algorithm to create the needed fuzzy sets and membership functions for the proposed type-1 fuzzy logic scenes classification system, and described the learning process of the system and the fuzzy rules generation from the input data and the fuzzy sets. We showed the upgrade to the type-2 fuzzy logic scenes classification system from type-1 fuzzy sets and membership functions. We proved the type-2 FLCS is better than type-1 FLCS in our researches, that is important for type-2 fuzzy systems researchers.

The results for the proposed systems based on experiments were compared with a BPNN scenes classification system. The type-1 and type-2 FLCS were thus proven to perform better than BPNN, and they have the interpretability to demonstrate the working process in detail to users with human understanding of the presentations. In the next chapter we employ the BB-BC algorithm to optimize our proposed fuzzy logic scenes classification systems in order to enhance the applicability.

Chapter 5 Big Bang-Big Crunch Optimization of Fuzzy Logic Classification Systems for Linguistic Video Summarization of Soccer Video

5.1 The purpose of optimization for fuzzy systems

The fuzzy logic scene classification system gives the higher accuracy prediction result than neural networks. However, the computation speed of the fuzzy logic classification systems could not bring the systems into the real applications. The minimum of frame display requires 25 images per second in the most video streams. However, the computation cost of fuzzy logic classification systems is too difficult for the real time prediction. The output prediction images are presenting at a low PFS and bring the latency for the input video streams. The real time computing function could not be implemented due to the large rules matching calculation. Also, users are unable to read so many rules in converting the system to a black box one.

In order to improve the prediction ability of fuzzy system and achieve the required objectives, there is a need to optimize the relevant parameters in the fuzzy system. The type of membership functions varies according to the employed system. The subjectivity involved in interpreting the linguistic variables exists because of the variation of human interpretation. Thus, in order to sketch subjective membership functions in a very complex system, we employ an optimization system in order to find the optimized membership functions and rule sets. The parameters to be optimized in the fuzzy systems are not only restricted on the membership functions. The original type-1 and type-2 fuzzy systems have a huge rule base (296,028 rules), sharing the same rule antecedents. However, this size of rule base brings the difficulty in rule generation and burdens the classification system with expensive computing costs.

There have been some designs of fuzzy logic systems using genetic algorithm (GA), and a hybrid architecture system based on fuzzy logic was tested to solve the problem of a servomechanism with nonlinear backlash, by obtaining better membership functions [Cazarez-Castro 2010]. Another publication presented an approach to design modular fuzzy controllers by using GA to optimize membership functions in order to improve simulation results of the proposed modular control method [Cervantes, 2013].

Particle Swarm Optimization (PSO) is another popular algorithm in optimization approaches, which has been used to increase the performance of a fuzzy control system by generating an optimal membership function. A fuzzy control system was applied to automatically back up a truck to a specified point on a loading dock from any starting position [Permana 2010].

The Big Bang-Big Crunch (BB-BC) optimization is a heuristic population based evolutionary approach presented by Erol and Eksin [Erol, 2006]. Fast convergence, ease of implementation and low computational cost are its main advantages. BB-BC has also been applied in many experiments and applications, for example, in [Yesil, 2012], the BB-BC has been successful used in type-2 fuzzy PID load frequency controller to obtain the better parameters. Yesil provided a good optimization framework of the use of BB-BC to obtain the fuzzy logic parameters. In this chapter, we will introduce the BB-BC algorithm to optimize the membership functions and rule base in our type-2 fuzzy systems.

5.2 Overview of the Big Bang Big Crunch (BB-BC) Algorithm

The theory of BB-BC is inspired by the Big Bang theory in physics, and this global optimization was built on two main steps:

- The first step is the “Big Bang” phase, where candidate solutions are randomly distributed over the search space. The initial Big Bang population is randomly generated over the entire search space just like the other evolutionary search algorithms. All subsequent “Big Bang” phases are randomly distributed about the centre of mass or the best fit individual in a similar fashion [Kumbasar, 2011].
- The second step is the “Big Crunch” phase, where a contraction procedure calculates the centre of mass for the population [Erol, 2006]. In this phase, the contraction operator takes the current positions of each candidate solution in the population and its associated cost function value and computes a centre of mass [Kumbasar, 2011].

The procedures of the BB-BC are as follows [Kumbasar, 2011]:

Step 1 (Big-Bang phase): An initial generation of N candidates is generated randomly in the search space, similar to the other evolutionary search algorithms.

Step 2: The cost function values of all the candidate solutions are computed.

Step 3 (Big Crunch phase): This phase comes as a convergence operator. Either the best fit individual or the centre of mass is chosen as the centre point. The centre of mass is calculated as:

$$x_c = \frac{\sum_{i=1}^{N} \frac{x_i}{f^i}}{\sum_{i=1}^{N} \frac{1}{f^i}} \tag{5-1}$$

where x_c is the position of the centre of mass, x_i is the position of the candidate, f^i is the cost function value of the i^{th} candidate, and N is the population size.

Step 4: New candidates are calculated around the new point calculated in Step 3 by adding or subtracting a random number whose value decreases as the iterations elapse, which can be formalized as:

$$x^{new} = x_c + \frac{\gamma\rho(x_{max} - x_{min})}{k} \tag{5-2}$$

where γ is a random number, ρ is a parameter limiting search space, x_{min} and x_{max} are lower and upper limits, and k is the iteration step.

Step 5: Return to Step 2 until stopping criteria have been met.

5.3 The Optimization of FLCS in Soccer Video Scenes

The T1FLCS was described in the previous chapter, which explained that this system achieves the basic objects of video scenes prediction. However, there is still a need to optimize the system to improve the performance of the classification accuracy and increase the system interpretability by decreasing the number of rules in the rule base.

5.3.1 Objective of BB-BC Optimization

The previous T1FLCS was created with training data including fuzzy sets, membership functions and the rule base. In the optimization stage, the training data is also used to improve the system performance via the BB-BC algorithm. We aim to improve the system though optimizing the relevant parameters of membership functions and reduce the size of the rule base.

In T2FLCS, the most important is the type-2 fuzzy sets and their membership functions fitness. In the chapter 7, it was shown that the process of T2FLCS generation includes creating the type-1 fuzzy sets with the initialized parameters d_k^i. The system will employ BB-BC to find the best d_k^i for each fuzzy sets. The second step for type-2 FLCS is mostly like the optimization process of type-1, which aims to decrease the size of type-2 rule base.

5.3.2 BB-BC Optimization for Type-1 FLCS

The generation of rules for all the available input combinations can result in the generation of a huge number of rules due to the curse of dimensionality problems of fuzzy systems. The huge rule base brings the high computation cost to the system and the system is not able to process 25 frames as the minimum video FPS, rendering it unsuitable for real-time video classification.

The BB-BC technique is applied to optimize the type-1 rules of the classification system in order to allow the classification system to maintain a high classification accuracy while minimizing the rule base. To optimize the rule base of the fuzzy classification system, the parameters of the rule base are encoded into a form of a population. The fuzzy classification system rule base can be represented as shown in Figure 5.1.

m_1^1	…	m_p^1	O_1^1	…	O_q^1	…	m_1^R	…	m_p^R	O_1^R	…	O_q^R

Figure 5.1: Population representation for the parameters of the rule base

In Figure 5.1, m_j^r are the antecedents and O_k^r are the consequents of each rule, where j = 1,…, p, p=12 is the number of antecedents; k = 1,…, q, q=3, which the number of output classes labels; r = 1,…, R, and R is the number of the rules to be tuned. However, the values describing the rule base are discrete integers while the original BB-BC supports continuous values. Thus, instead of Eq. (5-2), the following equation is used in the BB-BC paradigm to round off the continuous values to the nearest discrete integer values, modelling the indexes of the fuzzy set of the antecedents or consequents:

$$D^{new} = D_c + round\left[\frac{\gamma\rho(D_{max} - D_{min})}{k}\right] \tag{5-3}$$

where D_c is the center of the mass, γ is a random number, ρ is a parameter limiting search space, D_{min} and D_{max} are lower and upper bounds, and k is the iteration step.

5.3.3 BB-BC Optimization for Type-2 FLCS

The optimization of fuzzy sets and membership functions is crucially needed in a fuzzy system to find the best parameters in order to achieve the required objectives. Type-2 FLCS is upgraded from type-1 fuzzy system. The use of type-1 fuzzy sets is to get the type-2 fuzzy sets with their membership functions. However, the boundary of upper and lower membership functions in type-2 fuzzy sets is still not easy to set up. Thus, it is important to get the optimized boundaries in the type-2 membership functions.

The chapter 4 has fully described type-2 FLCS and the specific process of the type-2 fuzzy sets construction. BB-BC is employed to optimize the system performance and search for the best fitness parameters for the type-2 fuzzy sets with the Gaussian membership functions.

5.3.3.1 Type-1 Fuzzy Sets and Membership Functions Optimization

The use of value of d_k^i can make the type-2 membership functions have upper and lower boundaries. However, the value of d_k^i is based on the prior knowledge that may not represent the best performance of the type-2 fuzzy system. Thus, the use of the BB-BC could improve the system to achieve performance, making the prediction of outputs more accurate and producing more readable explanations for human users. In order to optimize the fuzzy sets and membership functions, we employ the BB-BC algorithm to obtain better parameters. As shown in Figure 5.2, we optimize d_k^i for each membership function in order to optimize the deviation of each type-2 Gaussian functions. The results are shown in Figure 5.3.

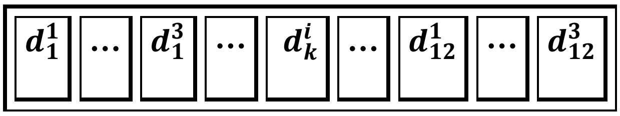
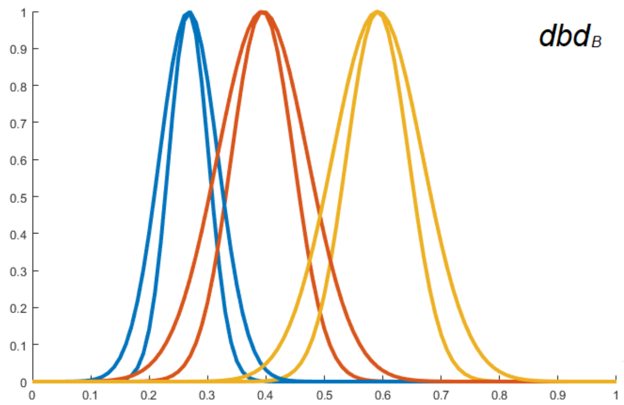
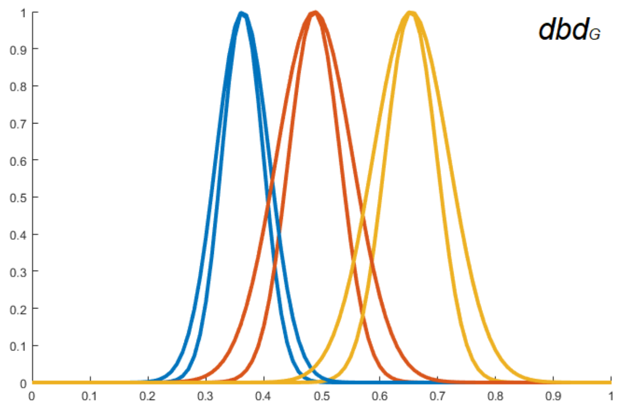


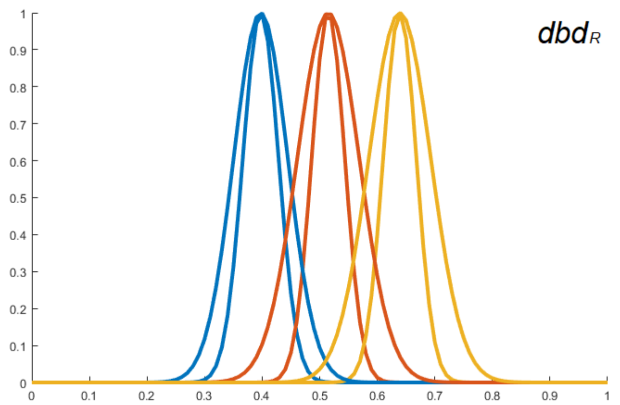
Figure 5.2: Population representation for the parameters of the type-2 fuzzy sets



(a) Type-2 fuzzy set of dbd_B

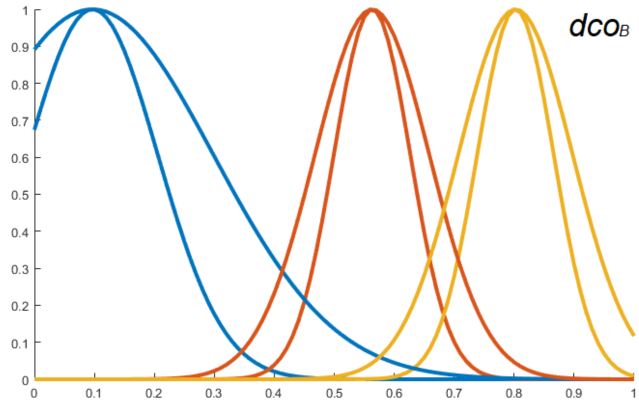


(b) Type-2 fuzzy set of dbd_G

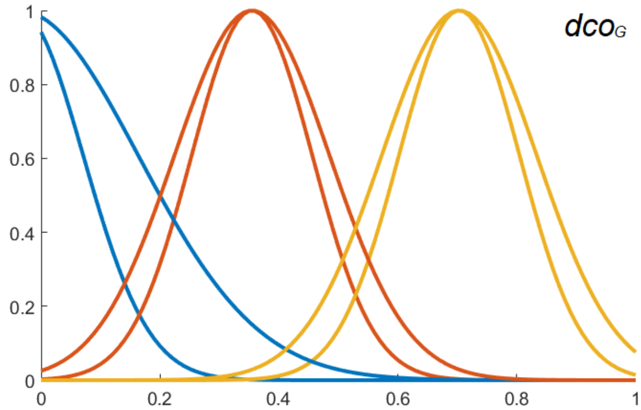


(c) Type-2 fuzzy set of dbd_R

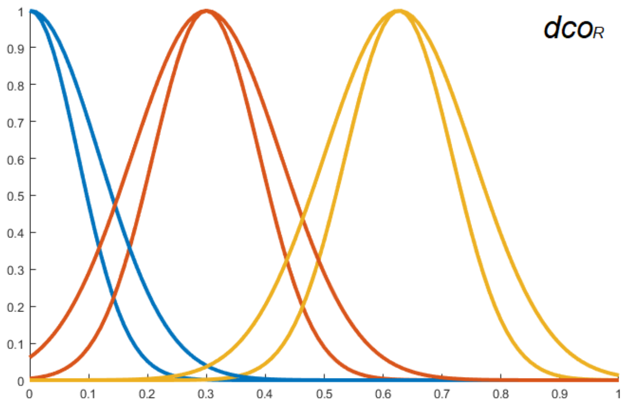
Figure 5.3: Optimized type-2 fuzzy sets and membership functions



(d) Type-2 fuzzy set of dco_B

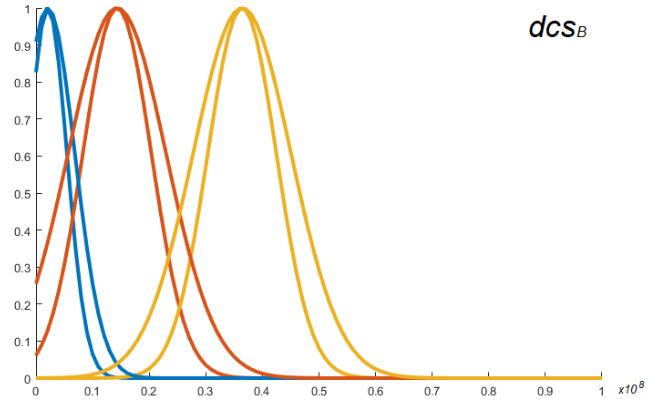


(e) Type-2 fuzzy set of dco_G

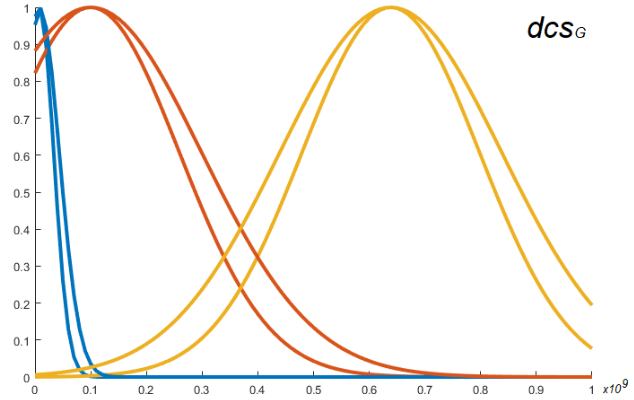


(f) Type-2 fuzzy set of dco_R

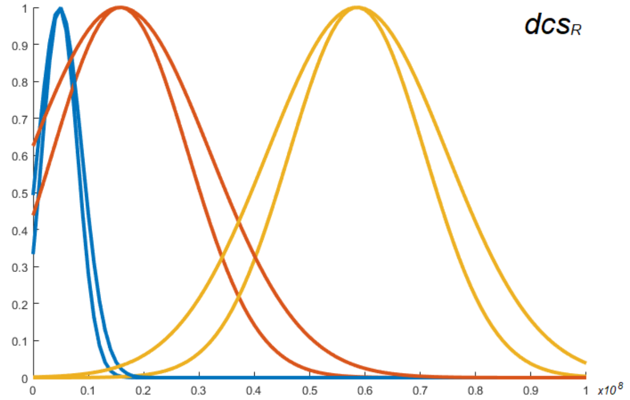
Figure 5.3: Optimized type-2 fuzzy sets and membership functions (cont.)



(g) Type-2 fuzzy set of dcs_B

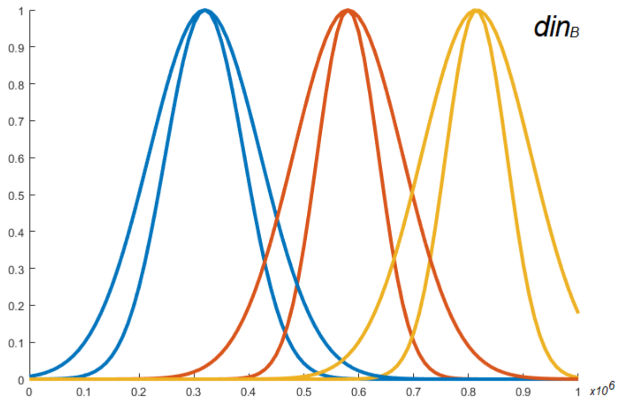


(h) Type-2 fuzzy set of dcs_G

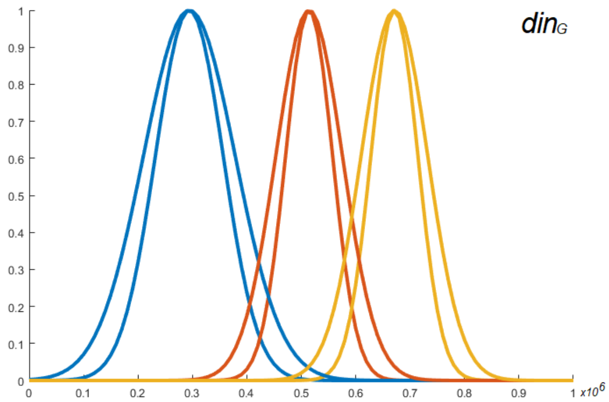


(i) Type-2 fuzzy set of dcs_R

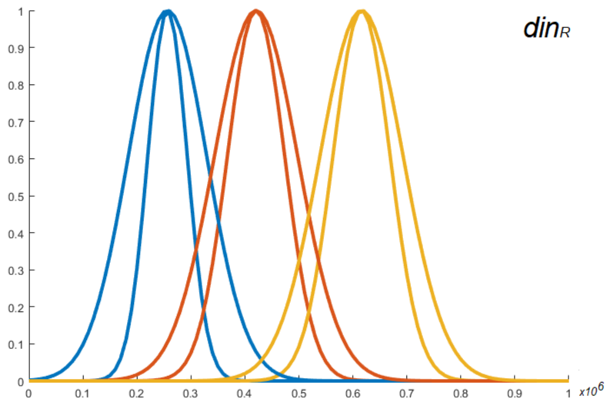
Figure 5.3: Optimized type-2 fuzzy sets and membership functions (cont.)



(j) Type-2 fuzzy set of din_B



(k) Type-2 fuzzy set of din_G



(l) Type-2 fuzzy set of din_R

Figure 5.3: Optimized type-2 fuzzy sets and membership functions (cont.)

5.3.3.2 Type-2 Rule Base Optimization

The BB-BC algorithm was employed to optimize the rules in type-2 FLCS in order to reduce the size of rule base. The approach is identical to the optimization in type-1 rule base. Type-2 rules optimization program and membership functions optimization program are run in different steps. We firstly run BB-BC to obtain the parameters of type-2 fuzzy sets and membership functions, then the BB-BC is used to reduce the rule base. The whole optimization process has to separate into two steps due to the huge rule base, large number of parameters in membership functions and the training dataset. The computation process took a long time for optimization and each step at least over 4 days. The one-step optimization strategy is available if the optimization program running in a super machine.

In this process of optimizations, the BB-BC firstly obtains better fuzzy membership functions, then continues to optimize the rule base.

5.3.3.3 Similarity for Rule Base Match with Incomplete Rule Base

As the BB-BC results in the reduction of the rule base, there are situations where the input vector does not fire any rules in the rule base. In this situation, we employ the similarity metric, which enables production of a classification output from similar rules in the rule base [Garcia-Valverde, 2012]. In order to calculate the similarity in the antecedent parts between the rule generated by the input ${x_i}^{(t)}$ and each rule R_q, a function distance is defined as $\mathfrak{D}(A_i^{k^*}, A_i^{(q)})$, where $A_i^{k^*}\left({x_i}^{(t)}\right)$ represents the fuzzy set matched by an input $x_i(t)$, and $A_i^{(q)}$represents the antecedent fuzzy set for rule q.

For example, in our application, we have five fuzzy sets for each inputs which are {Very Low, Low, Medium, High, Very High}; each label can be encoded as an integer, whereby Very Low is 1, Low is 2, … Very High is 5. So $\mathfrak{D}$ (High, High) = 0,

$\mathfrak{D}$ (High, Low) = 2, $\mathfrak{D}$ (Medium, Very High) = 2. With this aim, we define a distance that find the difference between the linguistic labels which are coded. Using this distance, the similarity between the rules created by the input $x^{(t)}$ with each rule R_q is calculated as [Garcia-Valverde, 2012]:

$$S\left(x^{(t)}, R_q\right) = \frac{\sum_{i=1}^{n}\left(1 - \frac{\mathfrak{D}\left(A_i^{(t)}, A_i^{(q)}\right),}{V-1}\right)}{n} \tag{5-4}$$

where $S\left(x^{(t)}, R_q\right) \in [0,1]$, V is the number of fuzzy sets and $i = 1, \dots, n$, where *n* is the number of values of the inputs which is the number of antecedents of the rule [Garcia-Valverde, 2012].

The similarity works if the incoming input vector does not match any of the existing rules in the rule base. The process starts by the crisp input vectors matching to different various fuzzy sets, and then combining the matched fuzzy sets results in generated fuzzy rules which do not exist in the fuzzy classification rule base. We then employ similarity to compare each of the generated fuzzy rules with the existing fuzzy rules in the fuzzy classification system rule base. For instance, assuming the rule base has only two rules, which are R_1{LOW, LOW, LOW} and R_2{HIGH, HIGH, HIGH}, if the incoming crisp input vector $x^{(t)}$ generates a fuzzy rule {LOW, LOW, HIGH}, which we cannot find in the rule base, then similarity *S* for R_1 and R_2 is:

$$S_{(x,R_1)} = \frac{\left(1 - \frac{\mathfrak{D}\,(LOW\,,LOW\,),}{5-1}\right) + \left(1 - \frac{\mathfrak{D}\,(LOW\,,LOW\,),}{5-1}\right) + \left(1 - \frac{\mathfrak{D}\,(LOW\,,HIGH\,),}{5-1}\right)}{3} = 0.83$$

and

$$S_{(x,R_2)} = \frac{\left(1 - \frac{\mathfrak{D}\,(LOW\,,HIGH\,),}{5-1}\right) + \left(1 - \frac{\mathfrak{D}\,(LOW\,,HIGH\,),}{5-1}\right) + \left(1 - \frac{\mathfrak{D}\,(LOW\,,HIGH\,),}{5-1}\right)}{3} = 0.5$$

So in this case, $S_{(x,R_1)}$is bigger, thus crisp input $\boldsymbol{x}$ applies rule R_1{LOW, LOW, LOW} as if it was the matched rule.

5.3.4 Experiments Comparing Original and Optimized Fuzzy Systems

In order to optimize the FLCS, we employed the BB-BC algorithm to reduce the size of rule base in type-1 and type-2 fuzzy systems and also find the better fitness with the parameters in type-2 fuzzy sets. The use of BB-BC improves system performance, including classification accuracy, with the less rules and membership functions and computation speed for the video running in real time.

The main differences between the two fuzzy classification systems are the size of the fuzzy rule base and the type-2 fuzzy sets' membership functions. The main purpose is to reduce the latency in order to achieve real-time video running; the second target is to increase the system predictions accuracy; and the third is to make the rule base readable for users. Thus, the optimized FLCSs have the better accuracy with less rules, faster computation speed, and more readable fuzzy rules than the original FLCS.

The experimental results are described in the tables below. Table 5.1 shows the comparison group from BP-NN classification system. Table 5.2 shows another comparison over the testing data and out-of-range testing data of the proposed Scenes Classification System using T1FLCS. The results in Table 5.2 use the full rule base (296,028 rules), with no BB-BC tuning, and we present groups of reduced rule base with BB-BC optimization T1FLCSs. As can be seen from Table 5.2, the proposed T1 outperforms the BP-NN from Table 5.1 on most classifications of average and individual class.

BP-NN	Centre	People	Close-up	Average
Training	73.1645%	68.8574%	70.7103%	71.9346%
Testing	61.4736%	63.6586%	73.4074%	64.1461%
Out-of-range testing	64.5819%	47.7024%	39.5120%	56.6065%

Table 5.1: BP-NN classification system for scenes

As can be seen from Tables 5.2 and 5.3, the T2FLCS outperforms the T1FLCS in testing and out-of-range testing groups. T2 achieved an average accuracy uplift of 0.4% and 0.1% compared to the T1 systems for the full rule base in the testing and out-of-range testing. However, T2FLCS performs better when the rule base is decreasing from full rules to 50 rules. As can be seen, the T2FLCSs can give a very close performance, with only 1000 rules (thus enabling the real-time and maximum interpretability), as opposed to using the full rule base of 296,028 rules, which cannot enable interpretability or real-time performance. The IT2FLC outperforms the T1FLC by about 1% for the testing data, and by about 5% for the out-of-range data, which verifies the ability of IT2FLC to handle the faced uncertainties and produce resilient performance in the face of high uncertainty levels.

Type-1	Centre	People	Close-up	Average
Training (full rule base)	95.3804%	85.3642%	74.1028%	90.1855%
Testing (full rule base)	93.7818%	83.5201%	70.6501%	87.1553%
Testing-1000 rules	93.3201%	78.2921%	65.5671%	84.7178%
Testing-200 rules	80.2350%	64.9031%	57.2034%	72.4097%
Testing-100 rules	70.0562%	41.3084%	35.7032%	59.9732%
Out-of-range data testing (full rule base)	84.5601%	56.9450%	32.6095%	69.6532%
Out-of-range data testing-1000 rules	82.6507%	37.2615%	25.4658%	62.9436%
Out-of-range data testing-200 rules	75.7023%	39.2953%	27.2059%	59.4484%
Out-of-range data testing-100 rules	58.6911%	33.4255%	23.4029%	47.1337%

Table 5.2: Type-1FLCS on scenes classification

Type-2	Centre	People	Close-up	Average
Training (full rule base)	94.6135%	87.1903%	72.3802%	89.7939%
Testing (full rule base)	93.9607%	85.6233%	69.5595%	87.5675%
Testing-1000 rules	92.6501%	81.3656%	69.1742%	85.7162%
Testing-200 rules	82.2314%	66.4139%	58.8425%	74.2252%
Testing-100 rules	69.3527%	42.0644%	38.6799%	57.2824%
Out-of-range data testing (full rule base)	83.7021%	54.7832%	38.5502%	69.6641%
Out-of-range data testing-1000 rules	83.2684%	44.5811%	40.2099%	67.4372%
Out-of-range data testing-200 rules	77.6121%	43.6174%	37.8850%	63.3797%
Out-of-range data testing-100 rules	58.7037%	39.2567%	31.6022%	49.8181%

Table 5.3: Type-2 FLCS scenes classification

5.4 Discussion

In this chapter, we presented the optimization process for our proposed fuzzy logic scenes classification systems (both of type-1 and type-2) for the better parameters optimization and rule base reduction, in order to improve the system prediction accuracy when it processes the input videos, and also in order to reduce the computational cost of the fuzzy rules matching the rule base. We employed BB-BC to optimize the fuzzy sets and rule base for our type-1 and type-2 scene classification systems.

The optimization process is divided into two steps due to the huge rule base, large number of parameters in membership functions and the training dataset. Optimization program require at least 4 days to complete each step, the first step is to obtain the better parameters of membership functions and the second step is to reduce the rule base. The combined one-step optimization is available if the program running in a super machine.

From the results of experiments, the type-2 fuzzy logic scenes classification system was the best. And, the optimized type-2 fuzzy logic system also performs well while keeping accuracy at a high level, while reducing the size of rule base from 296,028 to a few hundred. In the next chapter, we introduce a novel approach based on scenes classification system to build a different system in order to detect some high-level events from the video clips.

Chapter 6 The Proposed System for Event Detection Within Linguistic Video Summarization of Soccer Videos

In this chapter, we present a novel framework based on the previous scenes classification system, combined with the Dynamic Time Warping algorithm, in order to classify the input video clips and predict the event for it. The use of FLCS and DTW is necessary for our approach. We will present the specific process of building our proposed event detection system in the following sections. We will introduce the DTW in section 6.1 with its background. Then we will present the system in section 6.2 and present the experiments and results in section 6.3.

6.1 Introduction to Dynamic Time Warping (DTW)

Dynamic time warping (DTW) is an algorithm for measuring similarity between two sequences of time series, which may vary in speed. It has been applied to temporal sequences of the recorded video, audio and any data that can be turned into a single like sequence based on the time changes.

6.1.1 Principles of DTW Algorithm

In general, DTW is an algorithm that computes an optimal match between two sequences (e.g. time series) with certain restriction and rules:

1. Every index from the first sequence must be matched with one or more indices from the other sequence, and vice versa.
2. The first index from the first sequence must be matched with the first index from the other sequence (but it does not have to be its only match).
3. The last index from the first sequence must be matched with the last index from the other sequence (but it does not have to be its only match).

4. The mapping of the indices from the first sequence to indices from the other sequence must be monotonically increasing, and vice versa.

For example, if $j > i$ are indices from the first sequence, then there must not be two indices $l > k$ in the other sequence, such that index i is matched with index l, and index j is matched with index k, and vice versa. The optimal match is denoted by the match that satisfies all the restrictions and the rules and that has the minimal cost, where the cost is computed as the sum of absolute differences, for each matched pair of indices, between their values [Vintsyuk, 1968].

6.1.2 Process of DTW

To understand the theory of DTW, we present a sample of the process of DTW. We can arrange the two sequences of observations on the sides of a grid (Figure 6.1) with the sequence *X* on the bottom (16 observations in the example) and the stored template up the left hand side with sequence *Y* (same 16 observations). Both sequences start on the bottom left of the grid. Inside each cell we can place a distance measure comparing the corresponding elements of the two sequences.

To find the best match between these two sequences *X* and *Y*, we can find a path through the grid which minimizes the total distance between them. The path shown in Figure 6.1 gives an example. Here the first and second elements of each sequence match together at the left bottom grid, then the second of sequence Y starts to increase while X does not change. They then have the same change in the following 7 grids and the matrix is a long slash to the 9^{th} of X and the 7^{th} of Y. The line in the matrix will keep slashing when X and Y have same changes. Once an overall best path has been found, the total distance between the two sequences can be calculated for this stored template.

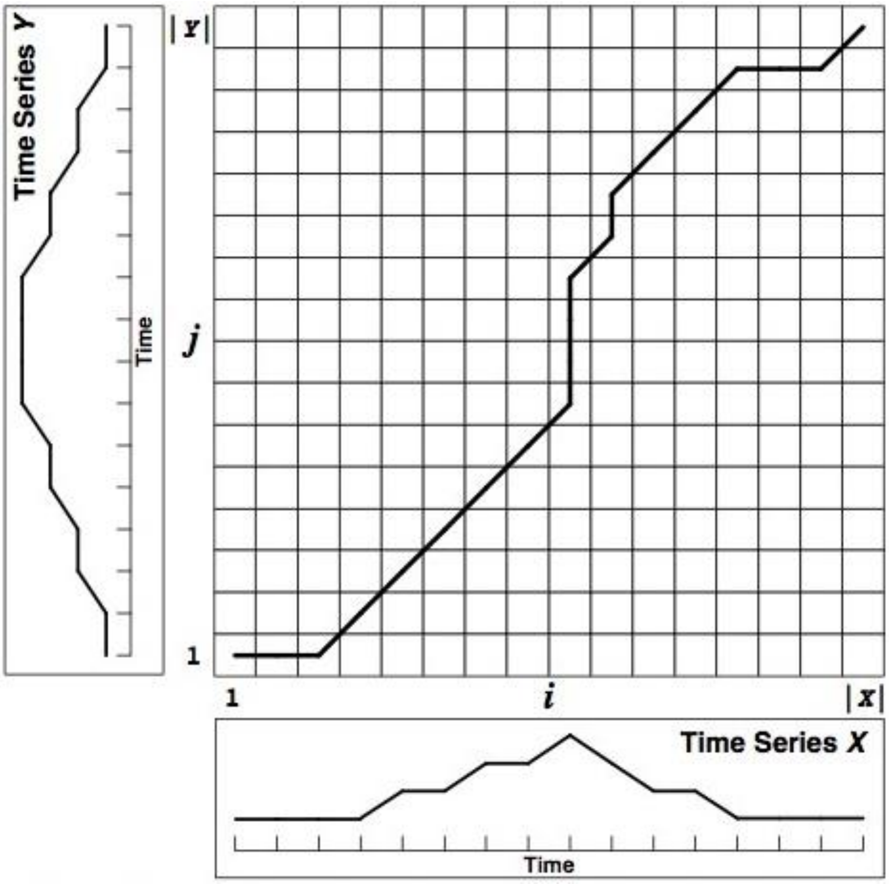


Figure 6.1: Cost matrix with minimum-distance wrap path traced through it

6.1.3 Definition of Distance of DTW

The DTW algorithm inherited the idea of dynamic programming, which is uniformly stretched or compressed undetected speech region to the same length as the reference template, so as to solve the matching problems caused by different pronouncing length [Jansen, 2009].

The DTW distance $d_{DTW}(X,Y)$ between the two sequences $X = \{x_1, x_2, x_3, \dots, x_n\}$ and $Y = \{y_1, y_2, y_3, \dots, y_m\}$ is defined as:

$$d_{DTW}(X,Y) = D(i*, j*) \qquad (6\text{-}1)$$

$$D(i*, j*) = d_{Eu}(i-j) + min\begin{cases} D(i-1, j) \\ D(i, j-1) \\ D(i-1, j-1) \end{cases} \quad (6\text{-}2)$$

where $x_i \in X$ and $y_j \in Y$, n and m are the maximum size of sequence X and Y respectively, and $d_{Eu}(i-j)$ is the Euclidean distance between two sequences.

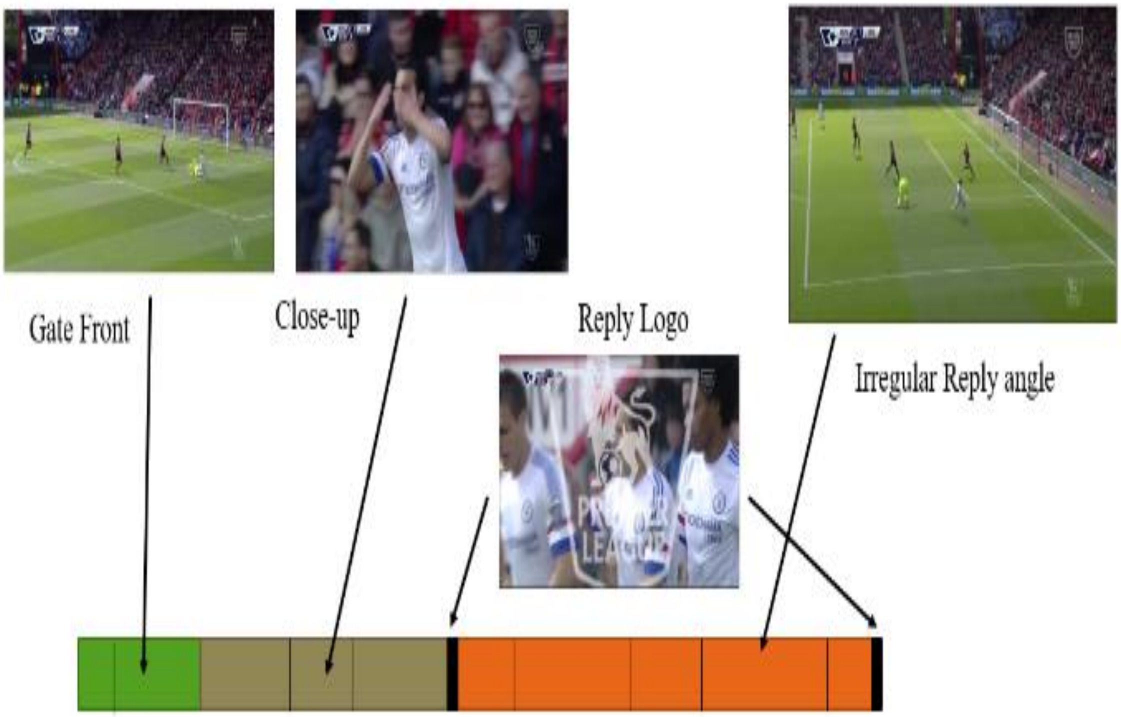
6.2 Proposed Event Detection System for Soccer Videos

In the first section we introduced the dynamic time warping algorithm. In this section we produce the event detection system based on the fuzzy logic scenes classification systems presented in chapter 4, to detect events for the soccer video clips.

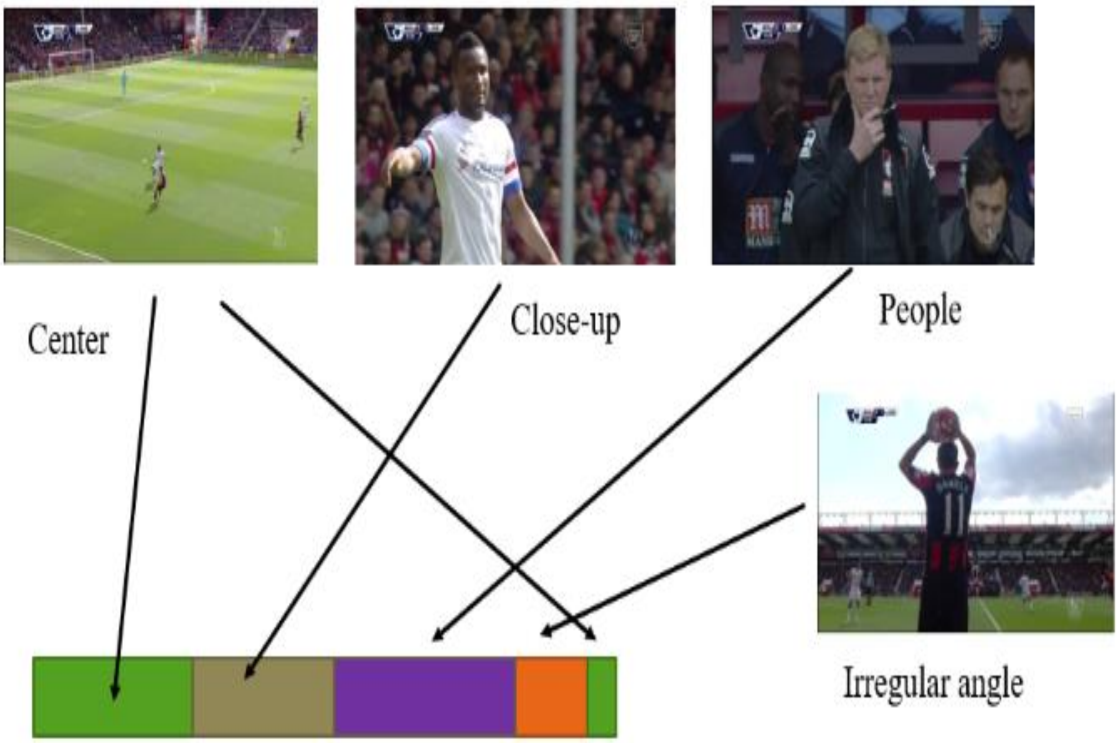
6.2.1 Soccer Video Clips

To construct the proposed system, we have to declare its objectives. The system learns the needed information from the video data in order to output the label of input video clip that predicts the event. The events in the soccer game represent the most important information of the video. Thus, our system should summarize the information of these events for users, which is also the final objective in our research. The event detection system learns and processes the video clips, noting some notable events in the soccer videos, as shown in Figure 6.2.

Figure 6.2(a) shows a goal event, consisting of a few centre scenes, close-up scenes, and some irregular scenes, where we see from different angles. Figure 6.2(b) represents an “outside” event, which means the football is outside and the player goes to throw the ball back into the field to resume play (i.e. a “throw in”). The events consist of many small clips with different video scenes, which our system focuses on in combination, learning the principles from the video clips.



(a) “Goal”



(b) “Outside”

Figure 6.2: “Goal” and “Outside” soccer events

6.2.2 Structure of Proposed System

The event detection system is the combination of scene classification system and dynamic time warping. A diagram of the event detection system is presented in Figure 6.3. The use of scenes classification systems is the core for event detection system. First, in the diagram of the EDS structure in figure 6.3, the input is a video clip, which is a sequence of images, each of which needs to be classified according to scene classes. Then the scenes classification system outputs the numerical values representing the strength of scenes, for each scene class. The numeric values of the strength of scene are used to feed the DTW, which learns the distribution of the scenes in video clips.

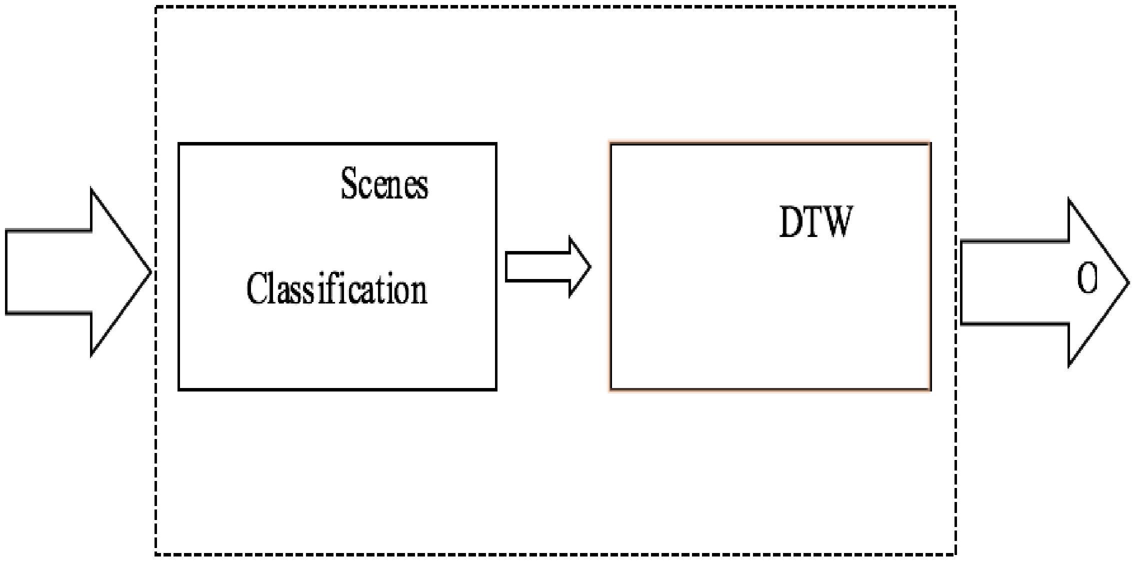


Figure 6.3: The structure of Event Detection System

Having completed a fuzzy logic scene classification system, the DTW learns the outputs of the system from new video events training data. The input vector of the DTW is defined as:

$$V_E = \{O_{Class_h}(V_U^1), O_{Class_h}(V_U^2), O_{Class_h}(V_U^3), \dots, O_{Class_h}(V_U^i)\} \qquad (6\text{-}3)$$

where $O_{Class_h}(V_U^i)$ is the scene evaluation values explained in Chapter 3 – Equation (3-19) for type-1 and Equation (3-22) for type-2, $i = 1,2,3, \ldots, I$, where I is the total frames of one video sequence. In this case, the scene classes have 3, thus the input vector of DTW also can be written as follows:

$$V_E = \begin{bmatrix} O_{Class_1}(V_U^1) & \cdots & O_{Class_3}(V_U^1) \\ \vdots & \ddots & \vdots \\ O_{Class_1}(V_U^I) & \cdots & O_{Class_3}(V_U^I) \end{bmatrix} \tag{6-4}$$

Thus, we select some samples from video events data as the training data for our Events Classification System (ECS). For each class in video events, we have

$$S_{Train}^h = \{{V_{E1}}^h, {V_{E2}}^h, \ldots, {V_{E3}}^h, \ldots, {V_{EN}}^h\} \tag{6-5}$$

where S_{Train} is the training sets, h is the class label of training data, ${V_{En}}^h$ is the training sequences, and N is the maximum size. Then, we employ the k-nearest neighbours (KNN) to process the S_{Train}^h to obtain the ${V_{Pose}}^h$:

$${V_{Pose}}^h = \{V_{Pose}1^h, V_{Pose}2^h, V_{Pose}3^h, \ldots V_{Pose}x^h\} \tag{6-6}$$

where h is the class label of training data, x is the consequence of the KNN. Thus DTW learnt the information and created the sets $D_{Trained}$, defined as:

$$D_{Trained} = \{{V_{Pose}}^1, {V_{Pose}}^2, {V_{Pose}}^h, \ldots, {V_{Pose}}^H\} \tag{6-7}$$

where H is the maximum of the event class. Assume the new input vector V_{EO} is a new video sequence after scene classification system processing. The system computes all DTW distance ${d_o}^h$, using Equation (6-6) for each class h. We then find the minimum DTW distance ${O_{DTW}}^h$ in each class, which can be defined as:

$$O_{DTW}{}^{h} = \min\{|V_{EO} - V_{Pose}{}^{h}|\} \quad (6\text{-}8)$$

where $O_{DTW}{}^{h}$ represents the similarity between the current sequence $V_{EO}{}^{h}$ and each predicted class. The system will output a class label h, which has the minimum of the $O_{DTW}{}^{h}$.

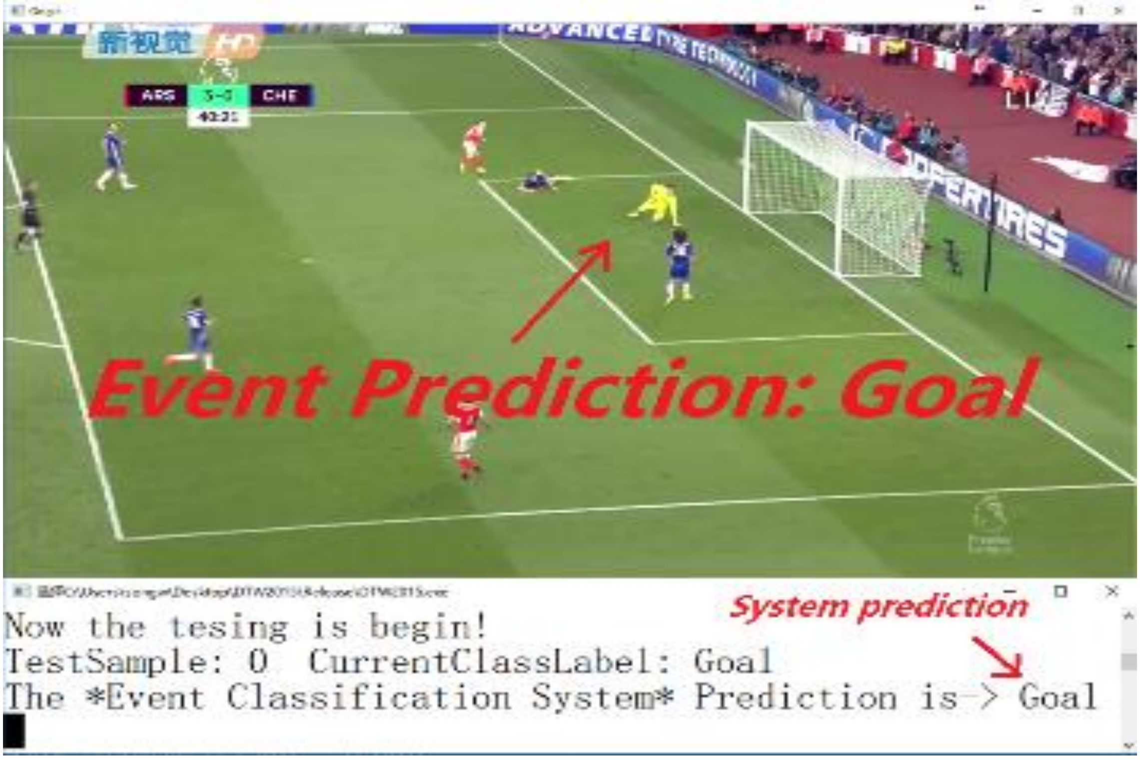
6.3 Experiments and Results

In this section, we present the experiments of our event detection system. Firstly, we list the events of the soccer which our proposed system employs, then focus on the reality of the event detection system. We present the accuracy of the proposed system and test it with different scenes to compare classification systems' performance in real-time processing.

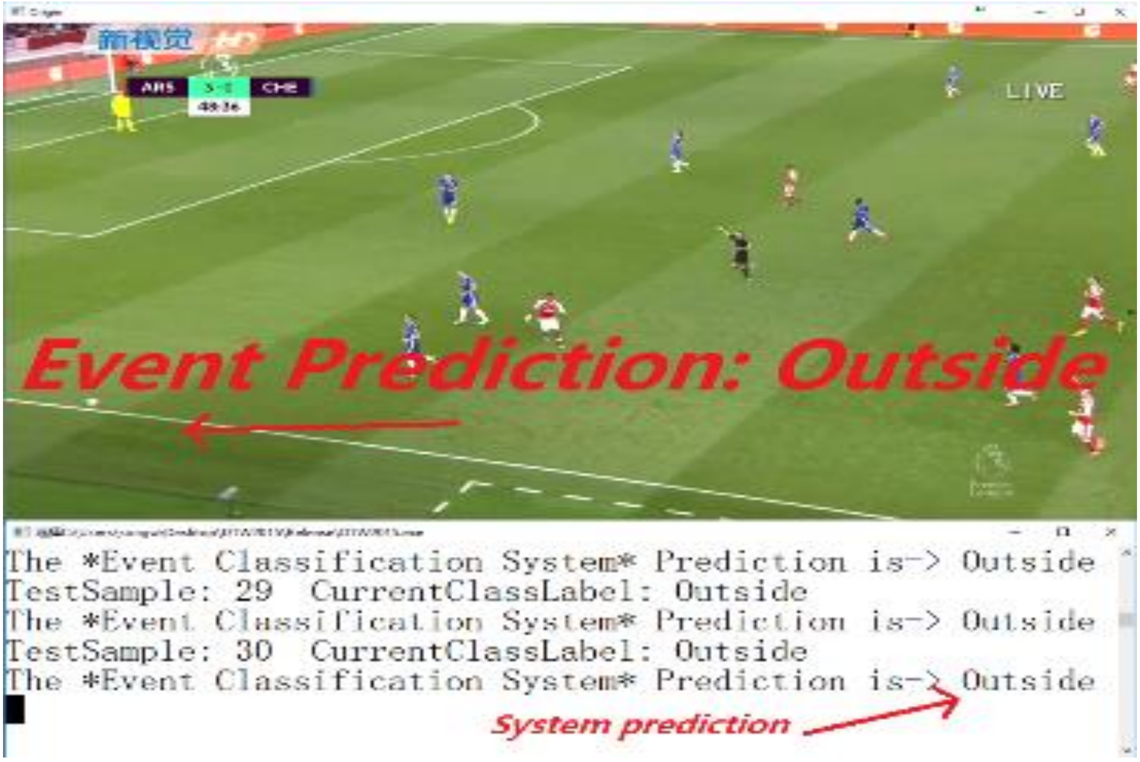
6.3.1 Video Clips in Soccer Data

We performed various experiments using the selected data from over 30 football videos from the English, Spanish, German and French soccer leagues. All videos had a resolution of 1280*720 pixels with frame rates of up to 25 frames/second. We extracted 1270 events from the soccer videos for six classes of the most important events in soccer: *Goal, Miss goal, Foul, Outside ball, Free-kicks (includes some corner ball)* and one comparison group with non-event group.

In Figure 6.4 our system calculates the DTW distance and predicts the new input sequence event for “Goal” and “Outside” events.



(a) “Goal”



(b) “Outside”

Figure 6.4: System prediction of “Goal” and “Outside” events

6.3.2 Video Event Detection Systems Results

We employed the Type-1 fuzzy logic (full rules and optimized rule), Type-2 fuzzy logic (full rules and optimized rule) and BP-NN scenes classification systems for five scenes to detect video events in order to find the best solutions for this problem. The results of each approach are presented below from Table 6.1 to Table 6.5. We also present the average accuracy for each EDS in training and testing dataset and out of range dataset in Table 6.6. The figure 6.5 and 6.6 are provided to visualize the prediction results of EDS and followed by the tables.

The training and testing dataset is used to feed the EDS for training and testing. The "Overall accuracy" represents the system prediction accuracy when running on the dataset of both training and testing meanwhile the "Testing Accuracy" is only the result of the testing dataset. The purpose of this experiments is to evaluate EDS on learning. The additional group means the EDS running the "out-of- range" data which is excluded and separated from the training and testing dataset. The prediction accuracy of additional group is used to evaluated the flexibility and capability of EDS on difference data source.

Event name	Goal	Miss goal	Foul	Outside ball	Free kick (including some corner kicks)	Non-event video clips
Overall accuracy	97% (87/90)	69% (165/240)	67% (183/270)	67% (195/290)	61% (109/180)	66% (101/200)
Testing accuracy	95% (19/20)	66% (46/70)	80% (60/75)	67% (50/75)	68% (27/40)	88% (44/50)
Additional group accuracy	90% (18/20)	54% (38/70)	51% (38/75)	72% (54/75)	45% (18/40)	60% (30/50)

Table 6.1: Event detection system based on the type-2 fuzzy logic scenes classification system in full rule base (191,578 rules)

Event name	Goal	Miss goal	Foul	Outside ball	Free kick (including some corner kicks)	Non-event video clips
Overall accuracy	91% (82/90)	71% (171/240)	63% (170/270)	66% (192/290)	57% (102/180)	66% (101/200)
Testing accuracy	85% (17/20)	70% (49/70)	75% (56/75)	64% (48/75)	68% (27/40)	88% (44/50)
Additional group accuracy	80% (16/20)	59% (41/70)	49% (37/75)	75% (56/75)	40% (16/40)	60% (30/50)

Table 6.2: Event detection system based on the type-1 fuzzy logic scenes classification system in full rule base (191,578 rules)

Event name	Goal	Miss goal	Foul	Outside ball	Free kick (including some corner kicks)	Non-event video clips
Overall accuracy	48% (43/90)	33% (80/240)	22% (60/270)	38% (95/290)	28% (102/180)	29% (58/200)
Testing accuracy	25% (5/20)	37% (26/70)	20% (15/75)	32% (24/75)	35% (14/40)	36% (18/50)
Additional group accuracy	20% (4/20)	27% (19/70)	23% (17/75)	36% (27/75)	23% (9/40)	22% (11/50)

Table 6.3: Event detection system based on neural networks scenes classification system (3 layers version)

We can see obviously that fuzzy logic based event detection systems are overall both better than BB-NN based EDS in overall and each specific class from table 6.1 to table 6.3. In the prediction accuracy of the testing dataset, the Type-2 fuzzy logic event detection system (full rule base) performs the mostly best among the three groups with the best accuracy in the event “Goal” with 97%, “Outside Ball” with 61%, “Free Kick” with 67% and “None Event” with 66%. However, type-1 fuzzy logic event detection system (full rule base) only outperform 4% in “Miss goal” and obtain the same prediction accuracy in “Free kick” and “non-event”.

Event name	Goal	Miss goal	Foul	Outside ball	Free kick (including some corner kicks)	Non-event video clips
Overall accuracy	97% (87/90)	66% (155/240)	81% (220/270)	49% (143/290)	51% (109/180)	56% (101/200)
Testing accuracy	95% (19/20)	60% (42/70)	81% (61/75)	51% (38/75)	53% (21/40)	66% (33/50)
Additional group accuracy	90% (18/20)	51% (36/70)	82% (58/75)	45% (34/75)	40% (16/40)	54% (27/50)

Table 6.4: Event detection system based on the type-2 fuzzy logic scenes classification system in optimized rule base (1000 rules)

Event name	Goal	Miss goal	Foul	Outside ball	Free kick (including some corner kicks)	Non-event video clips
Overall accuracy	90% (81/90)	63% (151/240)	75% (203/270)	45% (131/290)	57% (102/180)	59% (117/200)
Testing accuracy	85% (17/20)	64% (45/70)	77% (58/75)	48% (36/75)	45% (19/40)	72% (36/50)
Additional group accuracy	80% (16/20)	44% (31/70)	66% (50/75)	40% (30/75)	38% (15/40)	48% (24/50)

Table 6.5: Event detection system based on the type-1 fuzzy logic scenes classification system in optimized rule base (1000 rules)

Event name	T2EDS (full rules)	T1EDS (full rules)	BP-NNEDS	T2EDS (1000 rules)	T1EDS (1000 rules)
Average accuracy of overall	71.17%	69.00%	33.00%	66.67%	64.83%
Stand Deviation	11.8098	10.6927	8.28654	17.3077	14.311
Average accuracy for out-of-range dataset	62.00%	60.50%	25.17%	60.33%	52.67%
Stand Deviation	15.0665	13.8173	5.2731	18.8208	15.2607

Table 6.6: Average accuracy for each event detection system in original and out-of-range data

The primary results of the event detection systems are shown from Table 6.1 to Table 6.3. From the prediction accuracy, type-2 fuzzy logic based EDS perform best among three systems. However, this is a need to calculate the optimized fuzzy logic based EDS in order to evaluate the prediction accuracy which to compare with the BP-NN. The group type-2 optimized rule base had the best detection accuracy in the “Foul” event, with the 81% outcomes, followed by type-1 optimized rule base with the 75%. The optimized type-2 fuzzy system also obtained the best accuracy in “Goal” event detection with the same accuracy as the type-2 full rule base. Neural network has the worst accuracy among the groups, meanwhile the results indicate that white box approaches have the best performance.

The average accuracy of the T2 full rules system is 71.17%, which is better than the type-1 system (full rule base), which obtained 64.41% from the event detection. The average accuracy of optimized type-2 event detection system is 66.67%, where the rule base has only 1000 rules, which outperforms the type-1 event detection system (64.83%) with nearly 2% better performance, while the BP-NN performs the worst, with 33.00% average accuracy. In the “out-of-range” data experiments, the T2EDS with 1000 rules obtain 60.33% in average accuracy, which is about 7.5% better than the T1EDS, with 52.67%.

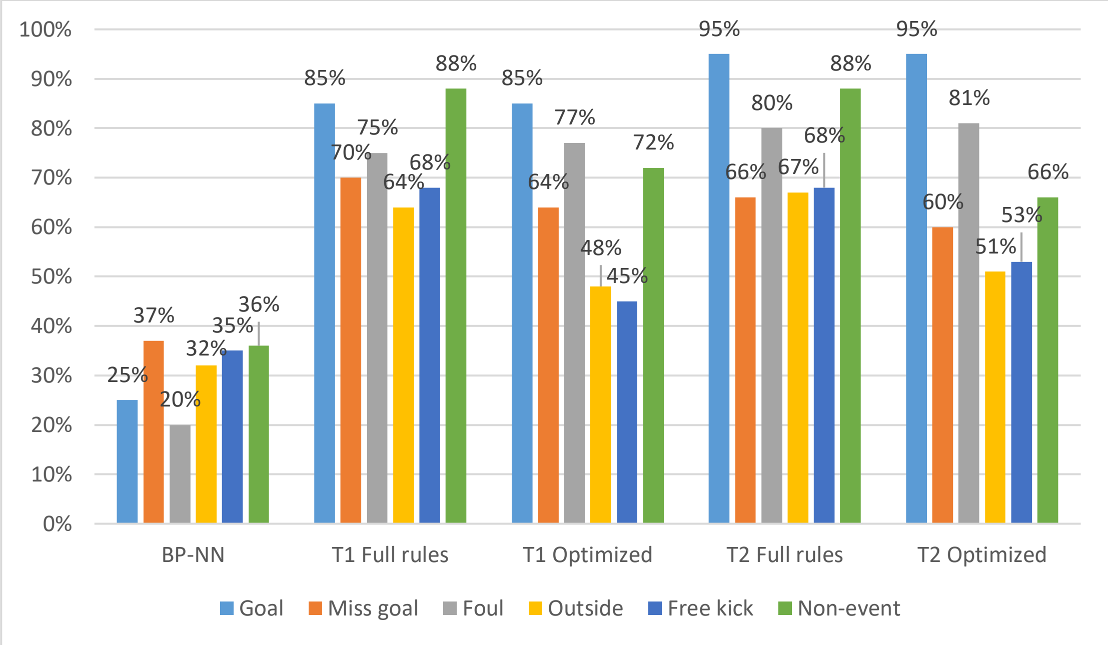


Figure 6.5: EDS prediction accuracy on Testing dataset

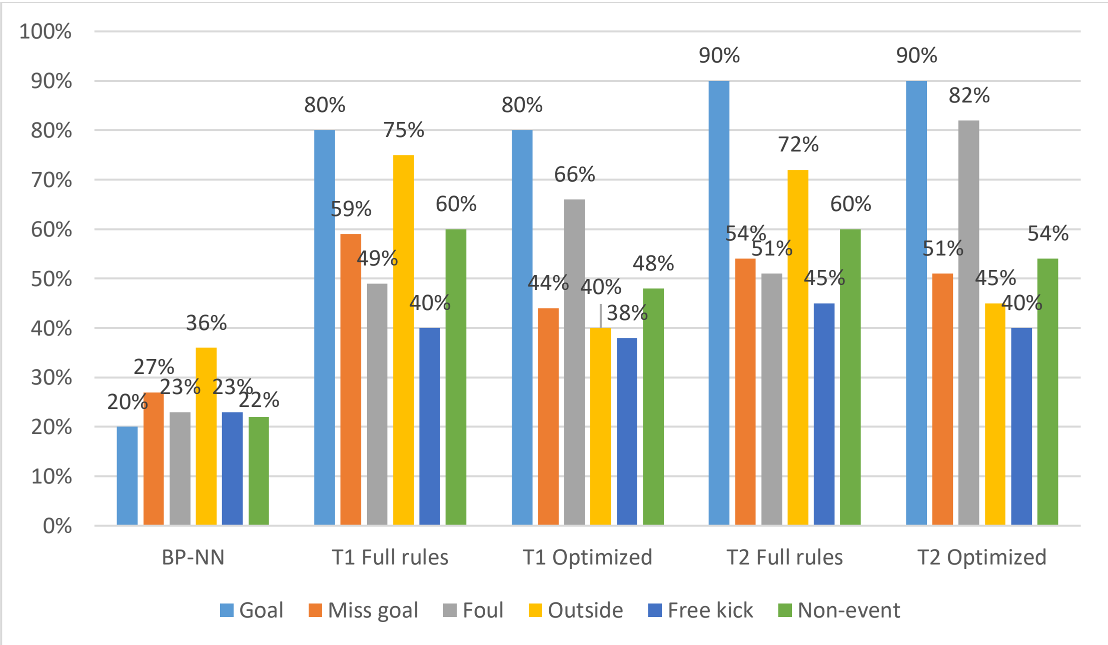


Figure 6.6: EDS prediction accuracy on out-of-range dataset

6.3.3 Evaluation of the proposed Video Event Detection Systems Real Time Processing

The purpose of the video summarization is to extract the content information from the videos for the user. However, for most applications of video summarization, it is difficult to process the videos in real time due to the fact that traditional computer vision algorithms are too expensive in their time complexity, like the Gaussian blur algorithm in $O(n \times r^2)$. In real situations, the image processing progress may require the needed algorithms to be run several times in order to complete some objectives. For example, the blur algorithm is widely used to support the Canny algorithm in the boundary detection of the image objects. Furthermore, the video must present 25 images (frames) each second, which increases the difficulty levels to the video processing. Finally, the intelligent systems have to run their unique learning algorithm at the same time in order to summarize the needed information from the extracted features. This makes the time complexity of the video summarization progress incredibly expensive.

6.3.3.1 Latency for DTW System

The DTW starts to work when one video clip has ended. In order to test the DTW processing time, we employ a machine with intel I7-6700k CPU in normal clock rate and Windows 10 Pro operator system. We tested the video event detection system under this environment and recorded the DTW real-time processing (Figure 6.7). From the results in Table 6.7 it is obvious that the DTW processing time is based on the length of the video clips. It takes only 2.35 seconds for the DTW to predict non-event videos but it requires over 5.7 seconds to predict "Goal" event, on average. Longer videos require longer processing times.

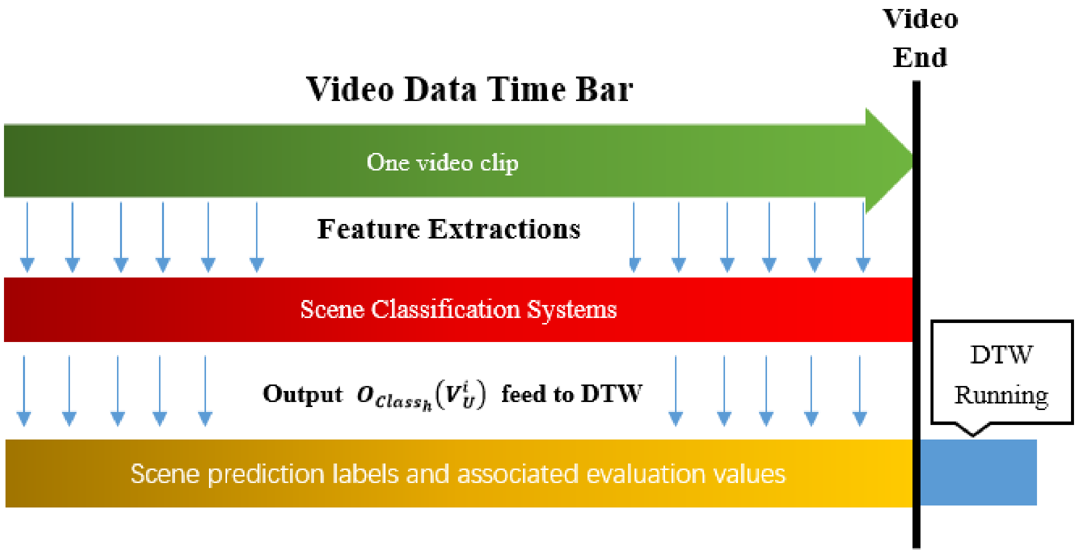


Figure 6.7: Video event detection systems in real-time processing

Video events classes	Goal	Miss goal	Foul	Outside	Free kick	None
Average length of video clips	54.9s	31.2s	19.2s	18.5s	15.6s	13.9s
Average processing time of DTW	5.77s	4.18s	3.01s	2.89s	2.46s	2.35s

Table 6.7: Latency of DTW system for each event detection

6.3.3.2 Latency for Scenes Classification System

The DTW algorithm requires a few seconds to conclude the prediction of the video event. In our real-time experiments, the videos are played in real time, which means that the scene classification systems can process 25 frames per second. However, there is still a need to know the specific time cost for the scene classification system in order to estimate the time cost if the system running in different machines. Thus, we designed another experiment that only recorded the time cost of the scene classification system with batch files. The diagram is shown in Figure 6.8.

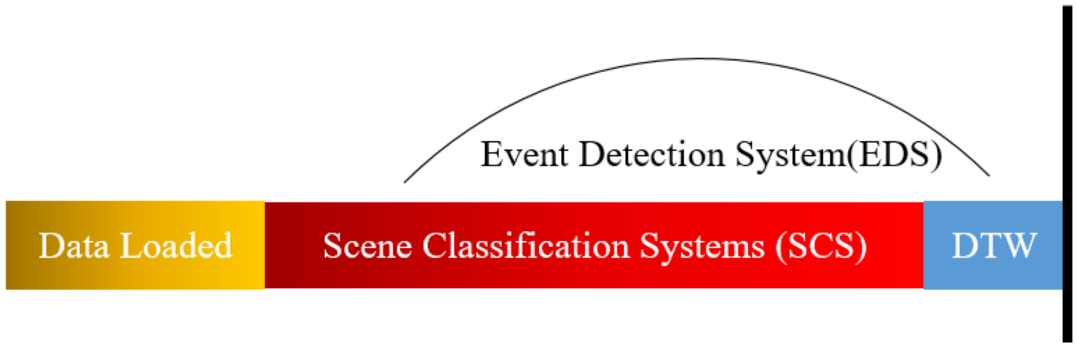


Figure 6.8: Experiments of time latency for scene classification systems

In this experiment, we converted the video data to the feature extracted data into the batch files (.csv) before the SCS started. The systems then loaded the batch files and calculated the needed $O_{Class_h}(V_U^i)$ for DTW. We recorded the time cost of the SCS required. The results are shown in Table 6.8.

Video events classes		Goal	Miss goal	Foul	Outside	Free kick	None
Average length of video clips		54.9s	31.2s	19.2s	18.5s	15.6s	13.9s
Classification system average running time (.csv batch file)	T1SCS (full rules)	10.15s	7.05s	5.11s	5.18s	3.91s	3.65s
	T2SCS (full rules)	10.57s	7.63s	5.24s	5.40s	4.21s	3.92s
	T1SCS (optimized 1000 rules)	7.37s	4.29s	4.01s	4.15s	3.50s	3.27s
	T2SCS (optimized 1000 rules)	7.88s	4.41s	4.17s	4.27s	3.79s	3.43s
	BPNN	4.45	3.23	2.89	2.89	2.87	2.87s

Table 6.8: Experiments of the latency for SCS in batch files

The latency of the scene classification systems for all video events are shown in Table 6.8. The BP-NN performs the best due to the simple feed-forward structure of the neural networks. The fuzzy systems are much slower than the BP-NN due to the size of the rule base and the inference calculations. However, from the results, our fuzzy scenes classification systems are all able to process the video in real time. The difficulties left for real-time processing are only those concerning feature extraction.

6.4 Discussion

In this chapter, we presented a novel approach to detect the events from the soccer video clips. The proposed system is the combination of the scenes classification system and the dynamic time warping. It processes the input video and predict the event in order to accomplish the goal of video summarization, whereby the system summarizes the information from the video for the user. From the experimental results, we also conclude that the performance of the event detection system is based on the performance of the scenes classification system.

T2EDS (full rules) obtain the average accuracy with 71.17% for overall groups and 62.00% for additional groups, and achieves the best results amongst the five systems. The average accuracy of optimized type-2 event detection (using only 1000 rules) system is 66.67%, while the type-1 event detection system has 64.83%, and BP-NN performed the worst with 33.00% average accuracy. However, we also calculated the stand deviation of EDS for overall and additional group. The optimization process would increase the stand deviation of system prediction accuracy. These increasing stand deviations represent fuzzy logic based EDS would be more imbalance on all prediction classes due to the reward functions of the optimization process. In chapter 5, the reward function of the optimization process of the scenes classification system is based on the overall prediction accuracy, that is good on improvement of the system prediction accuracy but not good enough for keep the prediction balance on each class.

We also evaluated the latency for the real-time video processing. In the experiments, we evaluated the time cost for DTW system and scene classification systems separately, and concluded that our technique can produce real-time results when operating on scene classification.

Chapter 7 Conclusions and Future Work

7.1 Study Conclusion

Chapter 1 explained the background of intelligent systems and their development, with particular reference to “uncertainty” and the difficulties and challenges of some learning algorithms. It then presented a brief description of video summarization techniques and their challenges. The objectives of the research were listed, to create a fuzzy learning based system for handling uncertainty in video summarization, and to generate systems with optimized classification accuracy and interpretability.

Next, the background of the video summarization and an overview description of the video study of the computer vision was introduced in Chapter 2. The chapter also presented the concept of the “scene” and demonstrated its importance in video study, including research and applications. Video scene is the key element to achieve some high-level recognition tasks in video summarization. Some important video applications were detailed as well as their research consequences in order to understand the difficulty of the video summarization. Following this, Chapter 3 gave an overview of fuzzy logic systems.

Chapter 4 introduced three steps of the process of the proposed fuzzy logic scenes classification system. Section 4.1 concerned feature extraction, clarifying the extraction process and producing the input vector to be learnt for the proposed system. Fuzzy C-mean algorithm is then used to generate the fuzzy sets and membership functions required to build the type-1 fuzzy logic scenes classification system. The learning process of the system and the fuzzy rules generation from the input data and fuzzy sets were specified, followed by the process of constructing the type-2 fuzzy logic scenes classification system (from the type-1 fuzzy sets and membership functions). From the primary results of the fuzzy logic scenes classification system and the comparison BP-NN scenes classification system, it was found that type-2 performed slightly better than type-1, while both show much better results than their neural networks counterpart.

Chapter 5 highlighted the process of optimization of the fuzzy logic scenes classification systems (both of type-1 and type-2) via BB-BC algorithm, which optimized the fuzzy sets and rule base. The use of BB-BC optimization algorithm obtained better parameters for fuzzy sets and reduced the size of rule base in order to improve prediction accuracy. From the experimental results, the type-2 optimized scenes classification system ran with a good accuracy after rules decreased from nearly 300,000 to a few hundred, while outperforming type-1 and neural networks. The IT2FLC(1000rules) outperformed the T1FLC(1000rules) by about 1% for the testing data and by about 5% for the out-of-range data, which verifies the IT2FLC capability to handle the faced uncertainties and produce resilient performance in the face of high uncertainty. After the rules were decreased to 200, it could be concluded that the optimized T2 could also keep the system performance at a good level in testing, with average accuracy of 74.2252%, which is better than T1 optimized system (72.4097%) and BP-NN (64.1461%).

Chapter 6 produced a novel approach to detect the events from the soccer video clips, to create a video activity detection system for soccer videos by using a scene classification system of T2FLCS working with a DTW. In order to automatically obtain the optimized parameters of the type-2 fuzzy sets and decrease the size of rule base of the T2FLCS (to increase the system interpretability and allow for real-time processing), an optimization approach was employed based on the BB-BC algorithm. The results of our soccer video classification experiments show that the proposed system with T2FLCS outperforms the T1FLCS for scene and event classification accuracy. It was shown that the type-2 fuzzy logic event detection system (full rule base) performed best amongst the five tested systems, with total accuracy of optimized type-2 EDS of 66.67% (with 1000 rules), outperforming the type-1 event detection system (with accuracy of 64.83%), while the BP-NN performed the worst (33.00% total accuracy). In the "out-of-range" data experiments, the T2EDS with 1000 rules obtained 60.33% average accuracy, which is much better than the T1EDS with 52.67%.

The latency for real-time video processing was also analysed. In the experiments, the time cost for DTW system and scene classification systems were evaluated separately, revealing that the scene classification happens in real time.

7.2 Future Work

In ongoing research, we intend to extend the proposed system to employ more classification systems to handle the higher uncertainty levels available in more complicated classifications, such as object summarization and behaviour detection. Fuzzy logic classification system is a white box approach which have the interpretability that could explain the details of reasoning process in giving predictions. This can be very important because of most learning approaches are not white box model, such as SVM, neural networks and the deep learning. The black box model could not show the details when it gives out the prediction, and it is not very good for people understand. Even if the deep learning approach is the most popular learning algorithm now, the black box model still could not show something people want to know, for example, the deep learning is wildly used in the unmanned car, many big companies employed deep learning on the unmanned car to recognize the road situation and make discussions of driving strategy like Google, Uber, etc. However, the real working of unmanned car always has the technique problems and, in these cases, the deep learning system could not make the correct decisions (Uber car crashed, March, 2018). We aim to expand the video activity detection system to more functions within sports videos, in order to allow us to move to real-time video classification and summarization. The white box model is possible to provide the better solutions in real world applications due to its interpretability.

We also intend to change our systems to operate on GPU processing using more inputs, more complex fuzzy sets. In this thesis, we have optimized our fuzzy logic systems to obtain the better parameters and reduce the rule base in order to make the application can be run in the real time. However, the hardware improvement could also accelerate our fuzzy logic systems to run in the real time. GPU accelerate processing enable the deep learning system can be completed in amount of time. The GPU may help the fuzzy systems to run much faster that fuzzy systems could run with large rule base at lower l. That could enable people build complex fuzzy systems with large amount of fuzzy sets and huge size of rule bases to process more complicated problems.

References

[Agrawal,1994] R. Agrawal, R. Srikant, "Fast algorithms for mining association rules", Proc. of 20th International Conference on Very Large Data Bases, pp. 487-499, September 1994.

[Bezdeck, 1984] J. Bezdeck, R. Ehrlich, and W. Full, “FCM: The Fuzzy C-means Clustering Algorithm”, Computers & Geosciences, Vol. 10, No. 2-3, pp. 191-203, 1984.

[Bishop, 2006] C. M. Bishop, “Pattern Recognition and Machine Learning”, Springer, ISBN 0-387-31073-8, 2006.

[Black, 2004] J. Black, T. Ellis and D. Makris, "Wide area surveillance with a multi camera network," in Intelligent Distributed Surveillance Systems (IDSS-04), 2004.

[Black, 2006] J. Black and T. Ellis, "Multi camera image tracking," Image and Vision Computing, vol. 24, no. 11, p. 1256–1267, 2006.

[Bosch, 2008] A. Bosch, "Scene Classification Using a Hybrid Generative/Discriminative Approach," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 30, no. 4, pp. 712 - 727, 04 April 2008.

[Cazarez-Castro 2010] N. R. Cazarez-Castro, L. T. Aguilar and O. Castillo, “Fuzzy logic control with genetic membership function parameters optimization for the output regulation of a servomechanism with nonlinear backlash,” Expert System with Applications, 2010, vol.37, pp. 4368-4378

[Cervantes 2013] L. Cervantes and O. Castillo, “Genetic Optimization of Membership Functions in Modular Fuzzy Controllers for Complex Problems,” Recent Advances on Hybrid Intelligent Systems,2013, vol451, pp 51-62

[Chen, 2009] B.-W. Chen, J.-C. Wang and J.-F. Wang, "A Novel Video Summarization Based on Mining the Story-Structure and Semantic Relations Among Concept Entities," Multimedia, IEEE Transactions on, vol. 11, no. 2, pp. 295 - 312, Feb. 2009.

[Crevier, 1993] Daniel Crevier, "AI: The Tumultuous Search for Artificial Intelligence", New York, NY: BasicBooks, ISBN 0-465-02997-3, pp. 44–46, 1993.

[Dai, 2005] J. Dai, L. Duan, X. Tong and C. Xu, "Replay Scene Classification in Soccer Video Using Web Broadcast Text," in Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on, Amsterdam, 2005.

[Dalal, 2005] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", CVPR, 2005.

[Ekin, 2003] A. Ekin, A. Tekalp and R. Mehrotra, "Automatic soccer video analysis and summarization," Image Processing, IEEE Transactions on, vol. 23, no. 7, pp. 796 - 807, July 2003.

[Erol, 2006] O. Erol and I. Eksin, "A new optimization method: Big Bang–Big Crunch" Advances in Engineering Software, Vol. 37, No. 2, pp. 106-111, 2006.

[Garcia-Valverde, 2012] T. Garcia-Valverde, A. Garcia-Sola and H. Hagras, "A Fuzzy Logic-Based System for Indoor Localization Using WiFi in Ambient Intelligent Environments" IEEE Transactions on Fuzzy Systems, vol. 21 No. 4, 702-718, November, 2012.

[Garrido, 2012] Angel Garrido, "A Brief History of Fuzzy Logic", BRAIN. Broad Research in Artificial Intelligence and Neuroscience, Vol 3, No 1, 2012.

[Gupta, 1977] Madan M. Gupta, "Fuzzy Automata and Decision Processes", pp. 89-102, North-Holland, NY,1977.

[Havens, 2009] T. Havens, G. Alexander, C. Abbott and J. Keller, "Contour tracking of human exercises," in Computational Intelligence for Visual Intelligence, 2009. CIVI '09. IEEE Workshop on, Nashville, TN, March 30 2009-April 2 2009.

[Heske, 1996] T. Heske, and J. Heske, “Fuzzy Logic for Real World Design. 1st ed”, Annabooks, 428 pp, 1996.

[Jansen, 2009] A. Jansen, P. Niyogi “Point process models for spotting keywords in continuous speech”, IEEE Transactions on Audio Speech & Language Processing, 2009, Vol. 17, No. 8, pp. 1457-1470.

[Ishibuchi, 1992] H. Ishibuchi, K. Nozaki, H. Tanaka, "Pattern classification by distributed representation of fuzzy rules", Proc. FUZZ-IEEE '92, pp. 643-650, 1992-March.

[Ishibuchi, 2001] H. Ishibuchi, T. Yamamoto and T. Nakashima, “Fuzzy data mining: Effect of fuzzy discretization,” in Proc. 1st IEEE Int. Conf. Data Mining, San Jose, CA, Nov. 2001, pp. 241–248.

[Ishibuchi, 2005] H. Ishibuchi, and T. Yamamoto, “Rule Weight Specification in Fuzzy Rule-Based Classification Systems”, IEEE Transactions on Fuzzy Systems, Vol. 13, No. 4, pp. 428-435, 2005.

[Joseph, 1976] Joseph Weizenbaum, “Computer power and human reason: from judgment to calculation”, W. H. Freeman and Company, 1976.

[Klir, 1998] G.J. Klir and M.J. Wierman, “Uncertainty-Based Information”, Heidelberg, Germany: Physica-Verlag, 1998.

[Kumbasar, 2011] T. Kumbasar, I. Eksin, M. Guzelkaya and E. Yesil, “Adaptive Fuzzy Model Based Inverse Controller Design using BB-BC Optimization Algorithm” Expert Systems with Applications, Vol. 38, No. 10, pp. 12356–12364, September, 2011.

[Kohavi, 1998] Ron Kohavi, Foster Provost, "Glossary of terms". Machine Learning, pp. 271–274, 1998.

[Liu, 1996] Hongping Liu, V. Chandrasekar, “Classification of Hydrometeors Based on Polarimetric Radar Measurements: Development of Fuzzy Logic and Neuro-Fuzzy Systems, and in Situ Verification”, Colorado State University, Fort Collins, Colorado, 1996.

[Lowe, 1999] Lowe, David G. (1999). "Object recognition from local scale-invariant features" (PDF). Proceedings of the International Conference on Computer Vision. 2. pp. 1150–1157.

[Ma, 2002] Yufei. Ma, Lie Lu, Hongjang Zhang and Mingjing Li, "A user attention model for video summarization," in International Multimedia Conference, New York, 2002.

[McCarthy, 1969] J. McCarthy, P. J. Hayes, "Some philosophical problems from the standpoint of artificial intelligence", Machine Intelligence, Edinburgh University Press, pp. 463–502, 1969.

[McCorduck, 2004] P. McCorduck, “Machines Who Think (2nd ed.)”, Natick, MA: A. K. Peters, Ltd., ISBN 1-56881-205-1, 2004.

[Mehran, 1998] Mehran Sahami, Susan Dumais, David Heckerman and Eric Horvitz, “A Bayesian approach to filtering junk e-mail”, AAAI'98 Workshop on Learning for Text Categorization, 1998.

[Mendel, 2001] J. M. Mendel, "Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions", Upper Saddle River, Prentice Hall PTR, 2001.

[Mendel, 2005] J. Mendel, R. I. John and F. Liu, "Interval Type-2 Fuzzy Logic Systems Made Simple," IEEE TRANSACTIONS ON FUZZY SYSTEMS, pp. VOL. 14, NO. 6, 12 2006.

[Mendel, 2007b] J. Mendel, "Type-2 Fuzzy sets and system: an overview," IEEE computational intelligence magazine, vol. 2, pp. 20-29, 2007.

[Mitchell, 1997] Tom M. Mitchell, “Machine Learning”, McGraw-Hill Science/Engineering/Math, ISBN: 0070428077, March 1, 1997.

[H. Moravec, 1980] H. Moravec "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover". Tech Report CMU-RI-TR-3 Carnegie-Mellon University, Robotics Institute, 1980.

[Permana 2010] K. E. Permana, and M. Ramachandran, “Fuzzy Membership Function Generation using Particle Swarm Optimization,” International Journal Open Problems Computation Mathematics, 2010, vol. 3, no. 1, pp.27-41.

[Poole, 1998] D. Poole, A. Mackworth and R. Goebel “Computational Intelligence: A Logical Approach.” New York, Oxford University Press, 1998, ISBN 0-19-510270-3.

[Rasheed, 2005] Z. Rasheed and M. Shah, "Detection and representation of scenes in videos," Multimedia, IEEE Transactions on, vol. 7, no. 6, pp. 1097 - 1105, 2005.

[Russell, 2003] Stuart J. Russell, P. Norvig, “Artificial Intelligence: A Modern Approach (2nd ed.)”, Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2, 2003.

[Samsuang, 2018] [Web resources] Samsung Fuzzy Logic Washing Machine. https://www.samsung.com/in/support/home-appliances/what-is-fuzzy-logic-in-a-washing-machine/3.2 accessed on May, 2018.

[Silver, 2016] David Silver and Demis Hassabis, "AlphaGo: Mastering the ancient game of Go with Machine Learning", Google Research Blog. 27 January 2016.

[Song, 2016a] Wei Song and Hani Hagras "A Big-Bang Big-Crunch Type-2 Fuzzy Logic Based System for Soccer Video Scene Classification", proceeding of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Vancouver, Canada 2016.

[Song, 2016b], Wei Song and Hani Hagras "A big-bang big-crunch Type-2 Fuzzy Logic based system for football video scene classification", proceeding of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Vancouver, BC, Canada, 2016.

[Song, 2017] Wei Song and Hani Hagras "A type-2 fuzzy logic system for event detection in football videos", proceeding of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Naples, Italy, 2017.

[Sugeno, 1986] M. Sugeno, "Industrial applications of fuzzy control", Elsevier Science Inc., NY,1985.

[Sutton, 1998] Richard S. Sutton, Andrew G. Barto, "Reinforcement Learning: An Introduction", MIT Press. ISBN 0-262-19398-1, 1998.

[Takahashi, 2005] Y. Takahashi, N. Nitta and N. Babaguchi, "Video Summarization for Large Sports Video Archives," in Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on, Amsterdam, 2005.

[Timothy, 1995] J. Ross Timothy, "FUZZY LOGIC WITH ENGINEERING APPLICATIONS", University of New Mexico, USA, 1995.

[Turing, 1950] Alan Turing, "Computing Machinery and Intelligence", Mind, LIX (236): 433–460, ISSN 0026-4423, October, 1950.

[Vintsyuk, 1968] T.K Vintsyuk,"Speech discrimination by dynamic programming", Cybernetics and Systems Analysis, Vol 4, No. 1, pp. 52-57, 1968.

[Viola, 2001] Paul Viola and Michael Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", CVPR, 2001.

[Wagner, 2010] C. Wagner, H. Hagras, "Toward General Type-2 Fuzzy Logic Systems," IEEE TRANSACTIONS ON FUZZY SYSTEMS, pp. 637 – 660, VOL. 18, NO. 4, 2010.

[Wang, 1991] L. X. Wang, J. M. Mendel, "Generating fuzzy rules from numerical data with applications", 1991.

[Wilkinson, 1963] R.H. Wilkinson "A method of generating functions of several variables using analog diode logic" IEEE transactions on Electronic Computers. EC12, pp.112-129, 1963.

[Xu, 2005] C. Xu, X. Shao, N. C. M. Kankanhalli and M. S., "Automatic music video summarization based on audio-visual-text analysis and alignment," in Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, Salvador, 2005.

[Xu, 2008] C. Xu, Y.-F. Zhang, G. Zhu and Y. Rui, "Using Webcast Text for Semantic Event Detection in Broadcast Sports Video," Multimedia, IEEE Transactions on, vol. 10, no. 7, pp. 1342 - 1355, Nov. 2008.

[Yesil, 2012] Engin Yesil “Interval type-2 fuzzy PID load frequency controller using Big Bang–Big Crunch optimization” Applied Soft Computing Volume 15, Pages 100-112, February 2014.

[Zadeh, 1965] L.A. Zadeh, “Fuzzy sets”, Information and Control, Vol. 8, pp. 338-353, 1965.

[Zadeh, 1983] L.A. Zadeh, "The role of fuzzy logic in the management of uncertainty in expert systems", Fuzzy Sets and Systems, Volume 11, Issues 1–3, 1983, Pages 199-227.

[Zhou, 2008] Z. Zhou, X. Chen, Y.-C. Chung, Z. He, T. X. Han and J. M. Keller, "Activity Analysis, Summarization, and Visualization for Indoor Human Activity Monitoring," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 18, no. 11, pp. 1489 - 1498, 2008.

[Zhou, 2009a] Z. Zhou, X. Chen, Y.-C. Chung and Z. He, "Video-based activity monitoring for indoor environments," in Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on, Taipei, 2009.

[Zhou, 2009b] Z. Zhou, W. Dai, J. Eggert and J. Giger, "A real-time system for in-home activity monitoring of elders," in 2009, Minneapolis, MN, Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE.

[Zhu, 2006] G. Zhu, C. Xu, Q. Huang and W. Gao, "Automatic Multi-Player Detection and Tracking in Broadcast Sports Video using Support Vector Machine and Particle Filter," in Multimedia and Expo, 2006 IEEE International Conference on, Toronto, Ont., 9-12 July 2006.

[Zhu, 2007] G. Zhu, Q. Huang, C. Xu, L. Xing, W. Gao and H. Yao, "Human Behavior Analysis for Highlight Ranking in Broadcast Racket Sports Video," Multimedia, IEEE Transactions on, vol. 9, no. 6, pp. 1167 - 1182, 2007.

[Zhu, 2009] G. Zhu, C. Xu, Q. Huang and Y. Rui, "Event Tactic Analysis Based on Broadcast Sports Video," Multimedia, IEEE Transactions on, vol. 11, no. 1, pp. 49 - 67, January 2009.