Feature based Multimedia Data Analysis in Video Games

Sarakatsanos Orestis

SID: 3308170020

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

NOVEMBER 2018

THESSALONIKI, GREECE
Feature based Multimedia Data Analysis in Video Games

Sarakatsanos Orestis

SID: 3308170020

Supervisor: Prof. Rigas Kotsakis

Supervising Committee Members:

SCHOOL OF SCIENCE & TECHNOLOGY
A thesis submitted for the degree of
Master of Science (MSc) in Data Science

NOVEMBER 2018
THESSALONIKI – GREECE
Abstract

This paper is a dissertation work as part of the MSc in the Data Science at the International Hellenic University. The goal of the work was to perform data analysis and classification in Video Games using multimedia features, in order to separate games based on their genre. The most popular game genres and the most popular games of each category are presented, and their main characteristics are discussed and explained.

The main purpose of this paper is to try and identify which video, image and sound features are the most effective in our classification task and of course which classifiers provide the best results. Various tables and diagrams are included that show the comparison results among all the features and the classifiers. Overall, we observed that the video/image features provide very high accuracy results and can be highly dependable, while the sound features show significantly lower accuracy in the genre classification but can still be regarded dependable.

At this point, I would like to offer my sincere thanks to my supervisor, Professor Rigas Kotsakis, for his immediate and constant help, his detailed and sufficient feedback in every step of our work and his encouragement during any problems we faced.
Contents

Abstract ............................................................................................................................................... 3

1 Introduction ...................................................................................................................................... 5

2 Theoretical Part .............................................................................................................................. 6

2.1 Historical Retrospective .............................................................................................................. 6

2.2 Multimedia Feature Analysis ..................................................................................................... 16

2.2.1 Audio Features ...................................................................................................................... 16

2.2.2 Image features ...................................................................................................................... 24

2.2.3 Video features ...................................................................................................................... 28

2.3 Pattern Recognition and Machine learning ................................................................................ 29

2.3.1 Supervised ........................................................................................................................... 29

2.3.2 Unsupervised ....................................................................................................................... 31

3 Experimental Part ........................................................................................................................ 33

3.1 Selected Games and Data Collection ......................................................................................... 33

3.2 Feature Selection ......................................................................................................................... 39

3.2.1 Video Features .................................................................................................................... 39

3.2.3 Audio Features .................................................................................................................... 42

3.3 Video Game Classes and Data Annotation ................................................................................. 44

3.4 Video Classification Models ..................................................................................................... 45

3.4.1 Classifiers ............................................................................................................................ 45

3.4.2 Inner Class Accuracy .......................................................................................................... 48

3.4.3 Visual Features Evaluation ................................................................................................. 50

3.5 Audio Classification Models .................................................................................................... 54

3.5.1 Classifiers ........................................................................................................................... 54

3.5.2 Inner Class Accuracy .......................................................................................................... 60

3.5.3 Audio Features Evaluation ................................................................................................. 63

Conclusions ....................................................................................................................................... 68

Future Work ...................................................................................................................................... 70

References .......................................................................................................................................... 71
1 Introduction

In this context, the classification task will point out/reveal specific audio and visual similarities among the different video game categories, in order to identify specific homogenic properties that consist the respective classes, and therefore dissimilarities in the utilized properties across the different categories. Furthermore, the classification analysis will possibly pinpoint key elements for each game genre, providing essential design guidelines for subsequent game development processes.

The purpose of this work is to classify video games, based on their genre, with the help of multimedia analysis and different multimedia features extracted from them. Taking into consideration existing works regarding the problem and proposed solutions, we aim to identify the stronger video, image and audio features that lead us to good classification scores.

Beginning with a Historical Retrospective of the most popular video game genres in the Theoretical part, we follow with general multimedia feature analysis, explaining the many video, image and audio features, commonly used in multimedia classification problems as well as presenting the most effective algorithms for these problems. In the Experimental part, the results of our tests are presented, showing the most effective classifiers and features we used and comparisons between them.
2 Theoretical Part

2.1 Historical Retrospective

The history of video games starts far back to 1950s when academic computer scientists started designing simulations and simple games for their research. [1], [4]

The first known electronic game was Bertie The Brain [2], created in 1950, which was an arcade tic-tac-toe game, built by Josef Kates. An arcade game or coin-op is a coin-operated entertainment machine typically installed in public businesses.

However, video game industry became more mainstream around 1970s-80s when first arcade games and video game consoles were firstly introduced to the general public, with the Galaxy, Computer Space and Pong games and with the release of the first home video game console, Magnavox Odyssey. After the 70-80s golden arcade era, the arcade industry faced a decline, as more home video game consoles such as the Sony PlayStation and Microsoft Xbox offered increased graphics and game-play capabilities at decreased cost. Nowadays, video games face a constant increase in popularity and the gaming industry plans efficient and intelligent access to them, similar to any other media. Video game industry is considered one of the biggest and continuously growing markets, with a big number of Video games sales annually. What may also come as a surprise to some, is that some competitive video games are into consideration of even joining Olympic Games under a new eSports label. This generation of game companies but also the players and their parents, are in the need of getting familiar and intimate with the gaming industry and understand the information provided to users through current video game access.

Initially the term video game was defined as “a video signal transmitted to a cathode ray tube (CRT) that creates a rasterized image on a screen” [3]. The contemporary meaning of the video game term defines “any game played on hardware built with electronic logic circuits that incorporates an element of interactivity and outputs the results of the player's actions to a display”.
A Video game genre is regarded as one of the most important features for accessing and categorizing video games based on:

I. The different types of information that are represented in the genre labels currently used for the game organization systems.

II. The facets that systematically describe the different types of information existing in each video game genre category.

After research [10], [18], and also heavily based on personal knowledge and experience, we identified some main and general video game genres according to popularity and sales, categorized mainly by their gameplay. However, we should note, that boundaries of video game genres are not 100% distinct and a game can fall to more than one general categories. For the purpose of this work, the collected data were annotated to belong to a single game genre, the one that identifies them the most.

More specifically:

**Action games:**

- **Arcade:** Arcade games were one of the first main video game genres [5], [6], initially introduced as coin-operated entertainment machines, usually installed in public areas like cafés and restaurants. The first appearances of arcade games are dated back to the 1970s with their golden era being around late 70s and mid-80s, with famous Pac-Man, Centipede, Space Invaders being some of the most iconic games of that era. Since arcade games are a big genre it can include other genres that are mentioned below, such as some of the first Fighting, racing games and the first Mario games. For the purposes of this work, we will refer to some specific arcade games, Pac-Man, Tetris and Space Invaders, some of the best-selling arcade games of all time.

Space Invaders was released in 1978 and the player controls a laser cannon and shoots the enemies as they descend from the top of the screen towards the bottom. The game also included a very specific and revolutionary music that interacts with the on-screen animation, influencing the emotions of the player. Pac-Man was released in 1980, and soon after became a social phenomenon and an icon of the 1980s popular culture. In
a Pac-Man game the player navigates through a black-colored maze containing various yellow dots, trying to avoid their enemies in the same time, with the goal being to “eat” all the dots without dying. The player achieves the highest possible score, if he manages to beat every round, by eating all possible dots and avoiding death by getting eaten by their enemies. Tetris was released in 1984 and it is a tile-matching puzzle game, where objects of different shapes appear, and the player has to match them accordingly in order to create solid lines in the bottom.

These three games are some of the most popular games of all time, with billions of estimated profits and hundreds of thousands sales.

- **Fighting**: A fighting game is a video game genre in which the player is involved in an interpersonal combat between a limited number of characters. The goal for the player is to defeat their opponents until the timer expires. A typical fighting game is consisted of multiple rounds taking place in a specific arena and each available character offers a variety of unique skills and abilities. Players must learn to combine different fighting techniques for each character, including good timing of their abilities, blocking, counter-attacking and chaining attacks to perform “combos”. The first fighting game was Heavyweight Champ [7], launched in 1976 and the fighting games genre became one of the most prominent arcade games during 90s. Fighting games utilize a 2D plane of motion, with characters moving left and right and jumping. The character controlled by the player and their opponent appear in the screen, initially one of them being on the left side and the other on the right, with constant movement involving. Usually fighting games are accompanied with some characteristic fast paced music. Some very popular examples of fighting games are Street Fighter series, Mortal Combat and Super Smash Bros.
- **Platform:** In platform games the player controls a character (avatar) to jump on and between platforms and overcome various obstacles. Usually these games feature uneven terrain heights and the player mostly uses the jump button in order to progress. The player might also be allowed to perform some other actions and movements such as swinging from objects or grappling hooks. Initially the platform game genre was developed in a 2D dimension with Donkey Kong [8] in 1981 and Mario Bros in 1983 games being some of the most popular. A sequel of Mario Bros, Super Mario Bros was released in 1985, one of the most popular and best-selling video games of all time, accompanied with one of the most iconic and recognizable video game music. Since the early 90s, a true 3D approach for the platform games was attempted with Alpha Waves being the first true 3D platform game, and new Super Mario series following soon. In the recent years despite platform games having a smaller presence in the gaming industry they are still successful and continuously produced for different consoles. The latest Super Mario game, Odyssey was released in 2017 for the Nintendo Switch console.

**First person Shooter:** A First-Person Shooter game (FPS) is a video game genre centered around a gun or another weapon in a first-person perspective, meaning the player sees through the eyes of the protagonist. First FPS games go back to the 70s, with the first highly popular FPS game being Doom, released in 1993. In the 21st century, FPS is one of the most commercially successful video game genres [9] and several FPS games have been popular eSports and competitive game competitions. FPS games utilize a 3D environment, displaying the character’s hands and weapon in the main view, also showing ammunition, player’s health and various location details. One of the most popular and successful FPS games is Counter Strike, initially released in 2000 with many updated versions following since then. Currently, Overwatch is regarded one of the most popular FPS games, based on active players and viewers in Twitch streaming platform.

**Role Playing Games:** A Role-Playing game (RPG) is a big and diverse video game genre where the player controls one or more characters, with various personal abilities and skills, living in a well-defined, usually imaginary world. Players explore the game world, engaging in combat and complete quests following a storyline. One of the main distinguishing features
of the genre is the enhancement of the character’s abilities and level progression. The character increases their attributes by gaining levels each time a certain amount of experience is accumulated, allowing them to use better abilities and higher-level weapons. In general, an RPG game offers higher and more complex character interaction and dialogues with the scripted behavior of non-player characters of the game, compared to other video game genres.

The first RPG games appeared in mid 1970s, mostly inspired by the tabletop Dungeons & Dragons game [11],[12]and other sources like the fantasy writings of authors such as J.R.R Tolkien and even traditional strategy games such as chess.

The first very successful RPG games were Ultima and Wizardry, first released in the early 1980s. Despite the first RPG games being strictly single-player oriented, with the vast rise in popularity of multiplayer modes early to mid-1990s, new RPG multiplayer games started to appear, such as Diablo [13]. In the 21st century, with the emergence of internet, a new subgenre of RPG games emerged as well, the massively multiplayer online role-playing games, (MMORPG) [14]. In these games the players form parties of many characters, exploring the in-game world together, or engaging in boss combats. One of the most successful and highest-grossing video games of all time, World of Warcraft is the main representative of the MMORPG sub-genre.

Usually in RPGs the player navigates the world from a first or third person view, with the Head-up Display (HUD), containing info about the characters level, health, weapons and abilities and a part of the world map. However, there are some alternative views, such as isometric or aerial top-down perspectives that are mostly common in the MMORPGS, in order to provide the player with a clearer view of their party and surroundings.

**Action-Adventure:** This genre is a hybrid genre, combining characteristics from the action genre and the quest/problem solving of a simple adventure game. In this paper we explore it as different category due to the high number of games that can be considered Action-Adventure games. Action adventure games engage both in in-game reflexes and fast gameplay and also include problem-solving and decision-making challenges. The game is based on the character movement following a storyline and contains many characteristics that can be found in a single-player RPG game, in terms of what the player sees in their monitor.
They do however, lack, the dynamic dialogue options, character interactions and progression that an RPG game offers.

An action-adventure game navigates the world usually in a first or third person view, with the HUD usually containing information about the characters health, weapons and map.

The first action-adventure game is considered to be Superman\textsuperscript{[15]} game, produced by Atari in 1979, with many action-adventure games following in the mid to late 90s such as Tomb Raider (1996)\textsuperscript{[16]} and Resident Evil (1996).

One of the most popular and bestselling action-adventure games, are the Grand Theft Auto series firstly released in 1997 with over 235 million sales overall. The Assassin’s Creed series is also one of the most popular games of this genre with more than 100 million copies as of September 2016. The latest hit of this genre is the latest God of War game, released in April 2018 selling more than 5 million copies in the first month of release.

The last year, a new game mode, Battle Royale, falling into the Action category emerged, with Epic Games’ Fortnite being the main representative. The goal in this game mode is to outlive your opponents and be the last man standing on the island. Since its release for Windows in September 2017, Fortnite has exploded as a cultural phenomenon with more than 100 million active players.

Based on the actual games that are currently popular and belong to the 2 above genres, and from a multimedia analysis point of view, an Action adventure game and a third person RPG game, can easily be confused.

The general gameplay is quite similar, and what differentiates them is the main character progress and dialogue interactions that exist in RPG. Therefore, in the aspect of this work, we will combine the above genres, into one and separate the MMORPGs genre as a different one, since it offers a unique style and gameplay mechanics and characteristics.
**Sport games:** It is a video game genre, simulating different sports, most common ones being Football, Basketball and combat sports but games simulating numerous other sports also exist [17]. Sport games challenge the player’s precision capabilities and tactical knowledge of the corresponding game. They try to model the athletic and mental characteristics needed in that sport and also take part in a stadium or arena with clear boundaries. A main characteristic of sport games is that during gameplay, they offer a play-by-play and color commentary through pre-recorded audio lines [18].

These games in general, use and update the real names of teams and players associated with that sport and a new version is released every year in order to keep up with the real-world changes, with improved graphics being an annual goal.

For the purpose of this work, we will explore two specific sport games, *Football* and *Basketball*, while car racing video games will fall into a completely different category explained just below.

Sports game genre is currently dominated by two major video game publishing companies, EA sports and 2k sports. They both hold official real-life league’s licenses in order to feature the real competitions and players.

*Football:* EA sports and KONAMI are the two main companies that have been producing Football video games for more than 20 years for different consoles, FIFA series and Pro Evolution Soccer Series, accordingly. Both these series have been some of best most selling video games series ever and are still maintaining a high popularity. In terms of what the player sees in screen, both these games show negligible differences. The players see the game in camera angle similar to how you watch a game in real life and controls one of the team’s players each time. The HUD includes current player names, the match score and the green color is by far the most dominant one, due to the representation of the pitch. As also mentioned above, in football games there also exists some commentary following each in game action.

*Basketball:* EA sports again and 2K sports are the two big companies associated with basketball games, and more specifically with the simulation of the American NBA league for around two decades. Following the same logic with the football games, the player sees
the match, in a camera similar to how they watch a real basketball game. The HUD includes information again about the match score, the currently controlled character and possibly some other information such as small graphs or tactics panels. Again, a light brown color representing the basketball pitch is very dominant and commentary also supplements the in-game experience.

Both Football and Basketball video game series, are one of the most popular ones, with new versions of them released by their respective publishers every annually, including all changes reflecting to the real-world player transfers and team changes, and usually offering improved graphics and match engine.

**Racing games:** In this type of video game genre the player participates in a racing competition, in various land types and weather conditions. The first racing game Space Race [19] was actually an Arcade game released by Atari in 1973. In the early 1990s racing games with 3D graphics appeared, like Nintendo’s Super Mario Kart (1992). The first installment of the Need for Speed series was released in 1994. Need for Speed game series is one of the most popular and best-selling video game series of all time with the most recent released in November 2017.

In racing games, the player controls the car in a first person or a third person perspective, either simulating a real-world racing league like Formula 1, or taking place in a completely imaginary world. The players car is in the middle of the screen and the HUD usually contains speedometer and a map with location details and the game experience is usually combined with fast paced racing-style music.

**Strategy games:** Strategy games are a video game genre that focuses on the players thorough thinking and strategic planning with the goal being to achieve victory [20], [21]. The player faces one or more opponents and must plan their actions as to eliminate the enemy forces or resources, by outthinking them. Players in strategic games are also involved in small economy management and in game world exploration in order to get an advantage over their opponents. Although strategy games can show similarities with RPG games, their main difference is that in RPG the player usually controls one or a very small number of players, while a strategy game focuses on a significantly higher number of fairly similar, controllable units. Strategy games in general offer single-player or multiplayer gameplay or both.
The first strategy games originate from tabletop strategy games like Chess, and the first console strategy game was Invasion, designed in 1972 for the Magnavox Odyssey. Since then, some various subgenres have emerged, based on small gameplay differences, like Real-Time Strategy games or Real-time Tactics games. Total War series (2000) and Age of Empires (1997), Warcraft (1994), and Starcraft (1998) series are some of the most popular strategy-based games, with millions of sales worldwide.

The games of the genre utilize a “godlike” point of view and the player indirectly controls the game character or characters, in different terrains and situations. The HUD includes numerous information about the player’s world, like Map, explored areas, resources and information about the currently selected units.

A subgenre of the strategy games will be explored separately just below, due to its current massive popularity and player base.

- **Multiplayer online battle area:** MOBA is one of the biggest subgenres of strategy games, initially originated from the real-time strategy subgenre [22], [23]. League of Legends (2009), possibly the most popular video game of the decade belongs in this genre. The player controls a single character in a team of two, who compete against each other with the ultimate goal being the destruction of the enemy team’s main building, making use and strategically manipulating periodically-spawned computer-controlled units that move forward in predetermined paths. Each of the controllable characters for the players, has various and unique abilities and can be improved through the course of the game. Warcraft 3 was the first major hit of this subgenre, with sponsored tournaments taking place. Defense of the Ancients (DotA) is also a popular game, initially a fan-made modification for the Warcraft but the major breakthrough happened in 2009 when Riot Games published the League of Legends (LoL) game. In September 2016 more than 100million players where active every month and is one of the most watched game in streaming media like YouTube and Twitch.tv. LoL game has one of the biggest and widespread competitive scenes with LoL leagues being held in North America, Europe, Korea, Taiwan and Southeast Asia.
Each MOBA game takes part in a typical and specific MOBA map, consisting of 3 main lanes and other “jungle” areas. Each of the two teams, usually consists of 5 players each one having somewhat different role, controlling and protecting one of the 3 lanes and their “jungle” area. The HUD offers information about the player’s-controlled character such as stats and the characters abilities, their Kill-Death-Assists scores in the current game, a map with various location details and some information about the player’s teammates. The game is played in a isometric point of view, so that a player can have a better view of their teammates and their surroundings.

All the above-mentioned video game genres face some very distinct differentiations between them in terms of multimedia aspect analysis. Information captured in a video game screenshot, like the color variations and the game’s HUD is an initial approach to separate and classify different video games in the above genres. However, in order to achieve better accuracy but also precision and recall scores, we need to utilize all the available multimedia features we can extract. Therefore, audio derived from different video games will be analyzed as well. Luckily, most of the genres, in general, have some particular and characteristic audio, music and songs that are embedded in them, like commentary as aforementioned for sport games, car engine sounds for racing games and distinct music in fighting games. However, we are in the need of identifying the specific features that actually result in good classification results. Dynamic ranges of both audio and image will also be taken into consideration, and also small duration videos of in-game gameplay will be used in order to separate fast-paced games from other slower and with smaller frame differences between each other.
2.2 Multimedia Feature Analysis

2.2.1 Audio Features

A big variety of researches ([24], [25]) has been conducted to identify and evaluate numerous audio features aiming to classify different kind of audio signals, such as human voice, music classification and semantic separation. The features that can be extracted from a digital audio file are mainly based on the sampling frequency and the quantization levels. More specifically the audio features can be separated in three classes [26], [27], [28], [29], [30].

Time-Based audio features

The signal envelope of a signal is a curve (Picture 1) passing through its extremes, giving a more general view of the signal’s amplitude changes.

\[
\bar{x} = \frac{1}{n} \sum_{i=1}^{n} t_i = \frac{1}{n} (t_1 + \cdots + t_n)
\]

![Audio waveform](image1)

*Audio waveform of ragtime excerpt*

![Corresponding envelope of the ragtime excerpt](image2)

*Corresponding envelope of the ragtime excerpt*

Picture 1: Signal Envelope
A relevant to the signal envelope feature is the E85 feature which refers to the level of signal envelope that the signal has reached after 85% of the observation time. The audio samples that do not exceed the E85 threshold constitute the silence period.

*RMS energy* is the root mean square of the signal’s energy.

\[
x_{rms} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2
\]

The RMS energy Curve (Picture 2) is following a similar pattern to the envelope one:

![Picture 2: RMS energy](image)

*Audio Low Energy* is the ratio of the audio samples below the average signal energy.

*Signal attack time and attack slope* is time Interval and slope (Picture 3) for Signal Increase/’Elevation.’

![Picture 3: Signal Attack Time and Attack Slope](image)

Based on the information theory the *Shannon entropy* [26], [27] is the average rate that the signal is created by stochastic source of data. It is a strong indicator of whether the signal
contains predominant peaks. A flat curve indicates a maximum uncertainty around the random variable’s output.

\[ H(X) := - \sum_{i=1}^{n} p(x_i) \log_b p(x_i) \]

Where \( b \) stands for the base of the logarithm. However, a better measurement of entropy, independent of the sequence length is given by:

\[ H(x) = - \frac{\sum_{i=1}^{n} p(x_i) \cdot \log(p(x_i))}{\log(length(p(x)))} \]

**Zerocrossing** is the number of signal’s zero values (Picture 4) directing to sign alternations of the signal envelope.

![Audio waveform](image)

**Picture 4: Zerocrossing**

**Signal Autocorrelation** which corresponds to the similarity between the signal observations based on the time lag function between them. It’s the division of the signal’s peak amplitude and the RMS value.
Crest factor shows the ratio of the signal’s peaks values to the effective values indicating how strong and extreme the peaks in the signal are.

Loudness is the logarithmic expression of normalized recording level of the sound, measured in dB.

Some features based on the signal peaks (Picture 5) can also be extracted from the audio file such us the number of the peaks, their average values and their variance.

![Picture 5: Signal peaks](image-url)
**Spectral-Based audio Features:**

*Spectral roughness* is the average distance between all possible pairs of peaks of the spectrum.

*Spectral irregularity* is defined as the variance of the successive peaks and is a measure of the spectrum’s non-normality. Its calculated as the sum of the squared distances between two successive spectral peaks

\[
\text{irregularity} = \frac{\left( \sum_{k=1}^{N} (a_{k+1} - a_k)^2 \right)}{\sum_{k=1}^{N} a_k^2}
\]

where \( a_k \) and \( a_{k+1} \) are the amplitudes of successive peaks

*Spectral brightness* is the percentage of the signal’s energy withheld above a certain frequency, with the 1500Hz being the default one.[26](Picture 6)

![Picture 6: Spectral Brightness](image)
The *Roll-off* feature detects the high frequencies of the audio signal (Picture 7). It is related to the maximum frequency within a selected fraction of the total signal energy.

![Spectrum](spectrum.png)

*Picture 7: Roll-Off*

*Spectral event density* indicates how often an audio event happens on the audio signal. An audio event can be a specific audio sound, such as a door slam, a door knock or human voice and laughter.

The signal’s *Fundamental frequency* is the signal’s lowest frequency and is used to analyze the other existing frequencies as multiples of the fundamental. All the other frequencies are named *overtones*. A derivative of the fundamental is the *inharmonicity* of the signal. It is defined as the degree to which the overtones can be referred as multiples of the fundamental.

*Spectral-spectrum centroid* (Picture 8) is the average frequency of the audio spectrum and is regarded as a good predictor of the sound brightness and an indicator of the tone color.

![Spectrum Centroid](spectrum_centroid.png)

*Picture 8: Spectrum Centroid*
Accordingly, *Spectral spread* (picture 9) is the variance of the signal around the spectral centroid and *spectral skewness* (picture 10) is the “asymmetry” of the signal distribution around the centroid. A final statistical feature is *Spectral Kurtosis* (picture 11), indicating the intensity of the values distribution.
Spectral flatness is a measure of the signal’s smoothness, indicating how much noise exists in the signal opposed to it being tone-like and it is computed as a ration between the signal’s geometric mean and the arithmetic mean.

\[
\text{Flatness} = \frac{\sqrt[N]{\prod_{n=0}^{N-1} x(n)}}{\frac{\sum_{n=0}^{N-1} x(n)}{N}}
\]

Delta spectrum magnitude (DFI) are defined as the successive transitions of spectral amplitude.

\[
DF_i = \sum_{k=1}^{N/2-2} \|X(k) - X(k+1)\|
\]

Cepstral-based audio Features:

**MEL FREQUENCY CEPSTRAL COEFFICIENTS (MFCCs)** is the single Cepstral feature and consists the calculation of the frequency bands with the implementation of Fourier Transform. It offers a description of the signal’s spectral shape. (Picture 12)

The absolute values of the signals are extracted and are logarithmically matched to Mel-Scale, which better resembles the human auditory system. A Discrete Cosine Transform is then applied mainly focused on the low-frequency components.
2.2.2 Image features

Image classification is a major computer vision topic and the quality of the extracted image features is significant for scene classification, object recognition and detection tasks[31], [32], [33]. Most of the image features originate from the basic image properties and are extracted from the information that is stored in Digital Images, based on grayscale and the RGB color model. [27], [28], [29], [30], [34], [35]

In a grayscale image each pixel is usually represented by 8 bits, therefore each pixel can present 256 (0-255) different levels of gray. Some grayscale images also make use of fewer bits per pixel (3,4,6), resulting in fewer levels of gray represented as shown in Picture 13.

6 bit: $2^6=64$ levels of gray  
3 bit: $2^3=8$ levels of gray

4 bit: $2^4=16$ levels of gray

Picture 13: 3, 4 and 6-bit levels of gray
The RGB model is an additive color model (picture 14), utilized in electronic and digital systems and is based on the human perception of colors (every color can be composed by three components), with three components, Red-Green-Blue, added together in numerous combinations, resulting in the final colored image.

A common RGB image uses 8 bits (1 byte) for the representation of each color, meaning a total of 24 bits is needed for each image pixel. However, some other encodings exist, utilizing 1, 2, 4, 5 and 16 bits per color. The number of bits used for each RGB component is called the color depth of the image. A common 24-bit image can present $2^{24} = 16777216$ colors and each of the 3 color components has 256 different intensities. Similar to how the grayscale image works, the number 0 indicates total absence of color (black) and 255 indicates the maximum intensity of each color.
The image features can be separated in two categories based on the image’s version they were extracted from. More specifically:

**RGB-based features:**

*RGB color percentage:* Each color component can be isolated from the others. In order to make use of this feature the image’s resolution needs to be taken into consideration and calculate the percentage of the specific color in the total number of the image pixels. This is useful in image classification where a big part of the image is expected to have a specific color.

*Average color intensities:* The intensity of the color components is an important feature for separating images with very bright colors from others with darker content.

*Color histogram:* It is used to visualize the average intensities of the color components. By itself this feature may not look very significant, but it can be a good indicator of the image properties and helps adjust the other features accordingly.

**Grayscale-based features:**

The initial RGB image is transformed into a grayscale, where each pixel corresponds to the amount of light in the image. It is an intensity image that includes only the luminance values of each pixel.

Two very simple and important features are extracted by computing the *mean value of the luminance* and the *standard deviation* of it, being very effective in distinguishing images with big color variations and different color intensities.

*Image contrast* is a measurement of the intensity contrast between a pixel and its neighbor. A contrast of 0 derives from a constant image.

*Image Energy* corresponds to the rate of change in the color or intensity of a pixel compared to its neighbor, across the whole image. It’s a measure of randomness and can take values from 0 to 1(constant image.)
Image *Homogeneity*, a measure of the “color distance”, and *Image Correlation* which corresponds to the correlation between a pixel and its neighbor, across the whole image, are two similar features. *Homogeneity* has values between 0 and 1 while, and as expected, correlation is valued from -1 to 1.

*Discrete Cosine Transform* is also technique often used for image processing, widely used in image compression, like the JPEG method. Each pixel of the image is converted into its frequency value and a NxN matrix is created, with the value of N depending on the image’s type. For a 24bit image a N=24 is needed with the cost of time complexity. As a result, *DCT* is used on a single-color component, with block size N=8.
2.2.3 Video features

All the features mentioned for the image processing can be expanded for the video analysis as well, although with a major difference. Unlike the image classification, video classification is based on a sequence of frames. [27], [29], [35]

According to the MPEG-1 standard, the analysis of the video frames can be executed with macroblocks, a processing unit of 8x8 pixels. The changes in the macroblocks between successive video frames showcases the motion it contains. Image histograms are also used for measuring the changes between frames, with confined success, due to the difficulties of selecting a threshold suitable for the whole video duration. To tackle this, some methods have been proposed for frame switching detection, using adaptive thresholds, in order to identify both steep and slow frame changes, based on fade-in, fade-out and dissolve.

Finally, Edge and Corner detection [31], [32], [33] are some important image analysis methods that are mainly used for video and are based on identifying specific interest points within the image. They are important in image classification and analysis tasks, offering particularly in areas of image matching, object recognition within the image, feature and motion detection, video tracking and 3D modelling and are used in many developed image and video classification techniques existing.

Depending on the nature of the classifying task that needs to be executed, some of the above-mentioned features are more suitable than others. The feature values must be inserted in algorithmic models, in order to detect similarities and differences between the different values and extract conclusions depending on the classification product.
2.3 Pattern Recognition and Machine learning

In the field of multimedia data analysis, a good number of machine learning algorithms and data mining techniques are used, utilizing the multimedia features presented earlier. According to the nature of each task the machine learning models classify the content into the existing classes using the temporal or spectral audio features and the visual features as inputs. Of course, not all the features are equally critical for each classification problem, but there are techniques that identify the most important and effective ones for each problem. The different multimedia features then, can be exploited as inputs for different algorithms and a combination of more than one machine learning methods for multimedia classification problems can improve the final product’s accuracy results.

Both supervised and unsupervised machine learning techniques are used for multimedia analysis and classification [30], [37], [38]. In general, the main methods used are:

2.3.1 Supervised

**K-Nearest Neighbors** [39]: Every element of the set is classified based on its “similarity” with the K-nearest neighbors. It’s a simple rather optimal method, resulting however in acceptable results and is often combined with additional methods to achieve the desired scores.

**Decision Trees** [40], [41]: Decision trees utilize a tree-like graph form, with decision nodes at the different levels. The end nodes represent the different classes and the nodes in the previous levels control one or more input features and lead to the next level with the general rule being: *if condition1 and condition2 and condition3 then outcome*. The input features are evaluated based on criteria and metrics (info gain, gini, entropy) in order to construct the optimal decision tree with the minimum possible levels and nodes, resulting in a simple and comprehensible representation of the classification task.

Decision trees are one of the most successful algorithms for radio content discrimination, based on entropy features. They are also used for binary classification tasks, such as television context isolation, using the image features based on MPEG standard and the DCT coefficients.
Random Forest methods is also used as an improvement of the simple decision trees trying to eliminate the overfitting problems often appearing in them.

Statistical Regression [35]: A dependent variable is modeled as a linear combination of one or more independent variables. The function can be either linear or not. More specifically:

- **Linear Regression**: The function is the in form of \(y=ax+b\) where \(y\) is the dependent variable(output) and the value of \(a\) and \(b\) are usually calculated by the Least Squares method. Multimedia data however, are usually represented by multiple features therefore Linear Regression is not optimal.

- **Multiple Linear Regression**:

  \[y = a_1x_1 + a_2x_2 + a_3x_3 + \ldots \ldots a_nx_n + c\]

  Again, \(y\) is the independent variable but, in this case, multiple independent variables \(x_i\) exist. Multiple regression is widely used in knowledge extraction problems being able to handle the correlation of multiple variables.

- **Logistic Regression**: It’s a non-linear regression method using logarithmic conversion resulting in the model’s output taking values in the \([0,1]\) space, allowing it to be used as a probability estimator of an event.

Genetic Algorithms (GAs) [42], [43]: GAs are inspired by the process of natural selection and evolution. They are widely used for optimization and knowledge extraction tasks using both numeric and categorical values and the extracted knowledge is following a binary representation.

Neural Networks (NNs) [29]: Simulating how the human brain works the NNs are based in a network of nodes that learn to perform tasks in machine learning and deep learning problems. Neural Networks consist of three operating layers:

Input layer, hidden-intermediate layer and the output layer. One of their main characteristics is they are capable of training themselves and are independent of any prior knowledge.
For multimedia analysis and classification tasks, Neural Networks are successfully applied for context analysis of audio content and offer high accuracy scores in TV content classification mainly utilizing the spectral audio features of the program.

**Support Vector Machines (SVM)** [38]: SVM models are used for data preparation aiming for classification and regression tasks. Although they are mainly used for linearly separable tasks they can also perform non-linear classification utilizing kernel tricks.

### 2.3.2 Unsupervised

**K-means**[44],[45]: It is the most famous unsupervised learning method widely used for signal processing, separating the input objects in predetermined number of k clusters. The objects are partitioned to the cluster with the nearest mean. The algorithm can work only with numerical values and in case of non-numerical objects they need to be converted into numeric. The algorithm operates in multiple rounds, each round identifying the nearest clusters for each of the objects. The algorithm ends either in a selected number of steps or when all the objects are stable within their cluster.

**Gaussian Mixture Models (GMMs)**[46]: They are probabilistic models and linear combination of Gaussians capable of approximating any continuous density.

Additionally, although not falling under either supervised or unsupervised learning, **Hidden Markov Models** are stochastic models that are extensively used for temporal pattern recognition tasks. More specifically, alongside *GMMs* and other unsupervised machine learning techniques such as *K-means* offer high accuracy scores in audio classification using both temporal and spectral domain features. These models are highly effective as well, when text features are used as inputs and are mostly used for context discrimination of TV News programs and movie genres.
Again, in audiovisual content classification tasks, a combination of \textit{k-Nearest Neighbors}, \textit{Decision trees} and \textit{Support Vector Machines (SVMs)} offer acceptable results, while for more specific human voice recognition in videos tasks, the \textit{Gaussian Mixture Models (GMMs)} with the addition of some \textit{MFCCs} offer good results as well. For more complex tasks, SVM models are being used initially for classifying the audiovisual context in some generic classes and on the next step they further classify them in subclasses following a hierarchical approach. Finally, image and video classification features like the RGB components and the frame speed of the video are used by Markovian models and Genetic Algorithms for television program classification, successfully classifying the content in News, ads, music programs or sports.
3 Experimental Part

In this part we will introduce and explain the steps followed in order to complete our video game genre classification problem. As previously mentioned, the goal of this work is to perform a video game genre classification based on video, image and sound features. According to the categories we introduced earlier, some of the main and most popular game representatives were picked to proceed in the feature extraction and create the needed dataset. Additionally, we will present and discuss the results from the various tested classifiers and evaluate the efficiency of each video and sound feature.

3.1 Selected Games and Data Collection

The first step was to identify and collect the data to create the dataset for the classification work and establish the “ground truth”. Based on the main video game categories that were introduced above, the aim was to find and collect gameplay footage from various games belonging to these genres. We explored 10 different video game genres with total 23 games tested for these categories. These 23 games were selected as the main representatives of each video game genre, based mostly on their popularity and the number of their active players.

We already mentioned some of the most popular representatives of each category in the Historical Retrospective part of this work and some of these games are further described below. For each of the selected games a representative gameplay image is given. More specifically:
1. Action-Adventure/RPG:

**Assassins creed:**

![Assassins Creed Odyssey game](Picture 15)

**Grand Theft Auto:**

![Grand Theft Auto V](Picture 16)

Picture 15: Assassins Creed Odyssey game

Picture 16: Grand Theft Auto V

**Witcher:**

![The Witcher 3](Picture 17)

**God of War:**

![God of War](Picture 18)

Picture 17: The Witcher 3

Picture 18: God of War

2. Arcade:

**Pac Man**

![Pac Man](Picture 19)

**Space Invaders**

![Space Invaders](Picture 20)

**Tetris**

![Tetris](Picture 21)

Picture 19: Pac Man

Picture 20: Space Invaders

Picture 21: Tetris
3. **Fighting:**

**Mortal Combat:**

![Mortal Combat](image1)

**Super Smash Bros:**

![Super Smash Bros](image2)

4. **First Person Shooter:**

**Counter Strike:**

![Counter Strike Global Offensive](image3)

**Overwatch:**

![Overwatch](image4)
5. MMO-RPG:

![World of Warcraft](image)

Picture 28: World of Warcraft

6. Multiplayer Online Battle Arena (MOBA):

**DOTA:**

![Dota](image)

Picture 29: Dota

**League of Legends:**

![League of Legends](image)

Picture 30: League of Legends

7. Platform:

**Donkey Kong**

![Donkey Kong](image)

Picture 31: Donkey Kong

**Super Mario**

![Super Mario](image)

Picture 32: Super Mario
8. Racing:

Picture 33: Need for Speed

9. Sports:

FIFA series

Pro Evolution Soccer series

NBA 2K series

Picture 34: Pro Evolution Soccer

Picture 35: NBA 2K19

10. Strategy:

Picture 36: Age of Empires
After picking the games we wanted to include in our classification, taking into consideration the possible differences among them in order to make the task as less biased as possible, we proceeded to the data collection and gameplay footage for each game.

The goal was to perform two different classification tasks separately, one including the video features and a second one with the sound features, comparing results between them and trying to identify the most successful one.

Therefore, for the needed dataset, both video and sound footage was essential. To achieve that, for each of the above-mentioned games, YouTube was used to search and find various gameplay footages being as much representative of each video game as possible. For each game, a video of ideally around five to ten minutes was selected and downloaded featuring representative gameplay footage. However, as we explain later, for the video classification we used only around 3 minutes of video footage for every genre while for sound more than 6 minutes.

Ideally, the same videos were used to extract the sound of the game, in order to follow with the sound classification. It was not always possible to find videos having both representative gameplay and sound, because some of the videos had the player’s commentary on it or even some advertisements. Fortunately, most of the videos that were selected were suitable for both video and sound extraction, but for the ones that were not, some additional representative video game sound and music was further searched for.
3.2 Feature Selection

Since we wanted to perform to separate classification tasks, two different datasets were created based on each multimedia modality, Video and Sound. Specifically:

3.2.1 Video Features

For each genre, around 3 minutes of video gameplay was used aiming in creating the video dataset for all the genres. The most clear and representative part of gameplay was selected from each of the downloaded videos and created the dataset based on the video frames.

The idea was to pick different frames from each video, essentially being images, and create a dataset based on the image features extracted with around 30-40 instances for each genre.

The extraction of the video features was performed in MATLAB 2017a. Each selected video had a framerate of 30 and in order to extract the visual features, the videos were divided in frames, sampled and averaged every 0.5 seconds with a step of 5 seconds each time, meaning that for every minute of video, 13 feature inputs were created, with a total of 341 inputs for all the genres.

The extracted features are specifically:

**Color Features:**

- **Mean** of the Red, Green and Blue components: Simply the mean value of each color component.
- The **amount** of Red, Green and Blue pixels surpassing a selected intensity value.

The above are the only color-based features. Some games are based on constant background with always the same color while others have some colors appearing more often than others like Football games having a constant green pitch appearing in the majority of the screen. Therefore, we thought that measuring the average intensity of each RGB component will help identify games with very specific coloring. Additionally, since as we know each RGB component takes values from 0 to 255, we decided to keep values above the 120 value threshold. The idea behind it was to measure the number of pixels in an image that possess a quite bright value of the respective color.
Grayscale Features:

- **Mean value of the Grayscale:** Giving the average luminosity of the image, trying to separate bright images from darker ones.

- **The standard deviation of the grayscale image,** indicating how much intensity variation exists in each image.

Mean gray value gives the average luminosity of the image and the standard deviation gives the variety of different luminosities. Both these attributes help distinguish images with big color variations and different color intensities and we regarded them as key attributes. Arcade games for example are based on a very dark-black background while other games are much brighter with big intensity variations.

- **The average length of the edges** in each image. After computing all the edges with the Sobel method, we decided to keep only a feature with the average length of them instead of the exact number or location. Based on the nature of each game, we decided that the exact positions of the edges wouldn’t be beneficial as a feature, but instead we kept just the number of them. In games with somewhat constant background, most of the lines would be in the same positions, but in games with fast and drastic changes in every frame, that would have led to the confusion of the classifier.

- **The number of corners,** using the Harris-Stephens method. Again, following the same idea as before, we kept just the number of the corners appearing in each image.

- The **image contrast**
- Image **Homogeneity**
- Image **Correlation.**
- Image **Energy**.
Contrast measures the intensity difference between a pixel and its neighbor across the image while correlation measures the correlation of each pixel with each neighbor. Homogeneity is a measure of distance of pixels across the whole image and finally energy is a measure of “randomness” in the pixels of the whole image. These features were expected to play a significant role separating fast paced games, with big and fast variations in what appears in the screen every moment, opposed to more static games, like MOBAs or Strategy and of course arcades.

The above features are mainly image features since they are depending on each frame of the video. To make use of the video and the motion property an extra feature was used.

- The **Euclidean distance** of the image intensity between frames, with half a second difference. This feature tries to identify and separate fast paced games like FPS from other slower and more stable games like MOBAs and Sports (match arena is constantly the same color) and it is a measure of change between frames. Specifically, we measured the Euclidean distance for each pixel from one frame, to each respective pixel of a frame 0.5 seconds later (15 frames after).
3.2.3 Audio Features

MATLAB was again used for the extraction of the audio features and more specifically operators and functions from the MIR toolbox. Audio classification requires much more features and inputs and as a result a much bigger dataset with audio features was created. The features we used have already been described in detail in the *Multimedia Feature Analysis* part of this paper.

- **Mirpeaks**
- **Mirrms**
- **Mirlowenergy**
- **Mireventdensity**
- **Mirzerocross**
- **Mirrolloff:** 4 different fraction thresholds were selected, 0.3, 0.5, 0.7 and 0.9, each of them making for an additional feature.
- **Mirbrightness:** 7 different frequency thresholds were used for this feature, 500, 100, 1500, 2000, 3000, 4000 and 8000, adding in total 7 features. We expected brightness to be a significant feature for the audio classification and that is why we selected that many different values.

**Statistics:**

- **Mircentroid**
- **Mirspread**
- **Mirskewness**
- **Mirkurtosis**
- **Mirflatness**
- **Mirentropy**
Timbre:

- **MFCCS**: 13 different Mel Frequency Cepstrum Coefficients were calculated and introduced as separate features.

In total 35 features were created and used for the audio classification task. For each category around 5 to 7 minutes of audio was used for the feature extraction. Two different window lengths rates were used. One every half a second and another every second. Their results will be presented later.
3.3 Video Game Classes and Data Annotation

In both cases, MATLAB created a csv file containing all the input attributes for both the video and audio datasets. The next step was to continue with the actual Machine Learning task. We decided to use Weka [47], [48] to test various classifiers and therefore the extracted CSV files were converted to ARFF file type compatible with Weka. The different video game genres were given a number indicating their class. For the 10 different games the classes were number 1-10 as shown in table 1.

Table 1: Video Game Classes

<table>
<thead>
<tr>
<th>Genre</th>
<th>ClassID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action-adventure/RPG</td>
<td>1</td>
</tr>
<tr>
<td>Arcade</td>
<td>2</td>
</tr>
<tr>
<td>Fighting</td>
<td>3</td>
</tr>
<tr>
<td>First Person Shooter</td>
<td>4</td>
</tr>
<tr>
<td>MMORPG</td>
<td>5</td>
</tr>
<tr>
<td>MOBA</td>
<td>6</td>
</tr>
<tr>
<td>Platform</td>
<td>7</td>
</tr>
<tr>
<td>Racing</td>
<td>8</td>
</tr>
<tr>
<td>Sports</td>
<td>9</td>
</tr>
<tr>
<td>Strategy</td>
<td>10</td>
</tr>
</tbody>
</table>

The aim was the compare various classifiers in both the video and the sound datasets we had created, try to identify the best ones and see which of the selected features are the most important and affect the final accuracy score the most.
3.4 Video Classification Models

3.4.1 Classifiers

After the dataset was loaded in Weka, we tested different classifiers and recorded their results using different number of folds for cross validation, 10, 5 and 3. We decided to keep the four best classifiers with scores higher than 80% accuracy.

The results are summarized in table 2 and the following diagrams for the different number of selected folds.

Table 2: Video Classifiers’ comparison

<table>
<thead>
<tr>
<th></th>
<th>MLP</th>
<th>Random Forest</th>
<th>Logistic Regression</th>
<th>KNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-Fold</td>
<td>85</td>
<td>90.9</td>
<td>83.3</td>
<td>88.3</td>
</tr>
<tr>
<td>5-Fold</td>
<td>86.2</td>
<td>90</td>
<td>80.9</td>
<td>83.9</td>
</tr>
<tr>
<td>3-Fold</td>
<td>83.3</td>
<td>88</td>
<td>78</td>
<td>83.9</td>
</tr>
</tbody>
</table>

Accuracies for each fold
10 fold Cross Validation

- MLP: 85
- Random Forest: 90.9
- Logistic Regression: 83.3
- KNN: 88.3

5 fold Cross Validation

- MLP: 86.2
- Random Forest: 90
- Logistic Regression: 80.9
- KNN: 83.9

3 fold Cross Validation

- MLP: 83.3
- Random Forest: 88
- Logistic Regression: 78
- KNN: 83.9
Overall as it is shown in the diagrams all the classifiers performed slightly better with more folds in cross validation. **Multilayer Perceptron** was the only classifier that performed better in 5folds compared to 10folds but again the performance slightly decreased for 3 folds.

Overall the highest accuracy was achieved with **Random Forest** classifier and 10 folds, 90.9% while **Logistic Regression**, in 3 folds offered the lowest score of 78%.

**K-Nearest Neighbors** classifier was the second best, very close to Random Forest’s performance. Different numbers of closest neighbors were tested in Weka but the best results were given by 3 neighbors and again 10 fold cross validation.

**Multilayer Perceptron** showed good results as well, and surprisingly it was the only classifier as mentioned before with higher accuracy in 5folds compared to 10.

Finally, **Logistic regression** slightly surpassed the 80% benchmark in the 10 and 5 folds run and it came 2% short in the 3 folds.
3.4.2 Inner Class Accuracy

At this point, we wanted to go more in depth of the results and try figure the misclassified objects and the reasoning behind it. The following table diagram shows the inner class accuracy for every different classifier. The results of the 3 different folds were summarized for each classifier as shown in table 3.

Table 3: Inner Class Accuracy for Video Classifiers

<table>
<thead>
<tr>
<th></th>
<th>MLP</th>
<th>Random Forest</th>
<th>Logistic</th>
<th>KNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>66%</td>
<td>80%</td>
<td>50%</td>
<td>76%</td>
</tr>
<tr>
<td>2</td>
<td>100%</td>
<td>100%</td>
<td>87%</td>
<td>100%</td>
</tr>
<tr>
<td>3</td>
<td>81%</td>
<td>95%</td>
<td>69%</td>
<td>92%</td>
</tr>
<tr>
<td>4</td>
<td>68%</td>
<td>73%</td>
<td>78%</td>
<td>60%</td>
</tr>
<tr>
<td>5</td>
<td>95%</td>
<td>97%</td>
<td>78%</td>
<td>89%</td>
</tr>
<tr>
<td>6</td>
<td>92%</td>
<td>99%</td>
<td>95%</td>
<td>91%</td>
</tr>
<tr>
<td>7</td>
<td>95%</td>
<td>81%</td>
<td>87%</td>
<td>88%</td>
</tr>
<tr>
<td>8</td>
<td>76%</td>
<td>77%</td>
<td>67%</td>
<td>62%</td>
</tr>
<tr>
<td>9</td>
<td>87%</td>
<td>96%</td>
<td>91%</td>
<td>92%</td>
</tr>
<tr>
<td>10</td>
<td>94%</td>
<td>96%</td>
<td>93%</td>
<td>96%</td>
</tr>
</tbody>
</table>
As shown in the diagram, the most misclassified objects, belong to class 1 (Action/Adventure games), class 4 (First Person Shooter games) and class 8 (racing games). This can be partially explained by taking into consideration the nature of these 3 video game genres.

All of them, are fast-paced games, with the environment constantly and quickly changing. No specific and constant colors appear in them and all these games utilize a screen HUD containing a map and other game related information, and the overall screen is based on what the character or object controlled by the player, “sees”.

On the other hand, we notice, that the Arcade genre, had almost perfect accuracy which can be explained by the unique nature of these games and the very simple and specific game screen during the gameplay.

MLP and Logistic Regression also showed some poor results in the Fighting game classification, which we think is resulted again from the fast-changing image contrast and energy of the fighting games.

Overall the diagrams show that Random Forest was the most stable classifier, outperforming the others for every class object, while Logistic Regression was the most varying one scoring even as low as 50% for the Action/Adventure games genre.
3.4.3 Visual Features Evaluation

Table 4 and 5 show the evaluation results for all the video features that were used:

Info-Gain: It evaluates how much information each feature gives us about the class. In Weka the information gain is calculated by:

\[
\text{InfoGain(Class,Attribute)} = H(\text{Class}) - H(\text{Class} | \text{Attribute})
\]

where \( H \) is the entropy.[47],[48]

Table 4: Video features Info-Gain evaluation

<table>
<thead>
<tr>
<th></th>
<th>Feature</th>
<th>Info-Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Energy</td>
<td>1.3419</td>
</tr>
<tr>
<td>2</td>
<td>Mean of Blue Pixels</td>
<td>1.2951</td>
</tr>
<tr>
<td>3</td>
<td>Mean of Green Pixels</td>
<td>1.2904</td>
</tr>
<tr>
<td>4</td>
<td>Green pixels</td>
<td>1.2683</td>
</tr>
<tr>
<td>5</td>
<td>Mean of Grayscale</td>
<td>1.2479</td>
</tr>
<tr>
<td>6</td>
<td>Correlation</td>
<td>1.1436</td>
</tr>
<tr>
<td>7</td>
<td>Mean of Red pixels</td>
<td>1.1259</td>
</tr>
<tr>
<td>8</td>
<td>Contrast</td>
<td>1.0877</td>
</tr>
<tr>
<td>9</td>
<td>Homogeneity</td>
<td>1.0429</td>
</tr>
<tr>
<td>10</td>
<td>Blue Pixels</td>
<td>0.9495</td>
</tr>
<tr>
<td>11</td>
<td>Red Pixels</td>
<td>0.9314</td>
</tr>
<tr>
<td>12</td>
<td>Number of Corners</td>
<td>0.8093</td>
</tr>
<tr>
<td>13</td>
<td>Average Length of Edges</td>
<td>0.7517</td>
</tr>
<tr>
<td>14</td>
<td>Standard deviation of Grayscale</td>
<td>0.7282</td>
</tr>
<tr>
<td>15</td>
<td>Euclidean Distance</td>
<td>0.5916</td>
</tr>
</tbody>
</table>
All the features have high info-gain values, meaning all of them played a significant part in the classification process. However, *Energy* was the most important one, while the *Euclidean Distance* of the pixels every half a second (measure of fast paced games) had surprisingly the least significant role of all. We also see that the mean value of the RGB components is ranked quite high. In general there is no big variation in the significance according to the Info-Gain criterion with only Euclidean Distance ranking noticeably lower than the rest.
OneR: Is an evaluation algorithm that tries to identify the feature that makes the fewest prediction errors in a classification process.

Table 5: Video features OneR evaluation

<table>
<thead>
<tr>
<th></th>
<th>Feature</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Energy</td>
<td>44.2815</td>
</tr>
<tr>
<td>2</td>
<td>Contrast</td>
<td>41.6422</td>
</tr>
<tr>
<td>3</td>
<td>Green Pixels</td>
<td>40.176</td>
</tr>
<tr>
<td>4</td>
<td>Mean of Blue Pixels</td>
<td>40.176</td>
</tr>
<tr>
<td>5</td>
<td>Mean of Green Pixels</td>
<td>39.8827</td>
</tr>
<tr>
<td>6</td>
<td>Blue Pixels</td>
<td>39.2962</td>
</tr>
<tr>
<td>7</td>
<td>Red Pixels</td>
<td>38.1232</td>
</tr>
<tr>
<td>8</td>
<td>Mean of Grayscale</td>
<td>37.5367</td>
</tr>
<tr>
<td>9</td>
<td>Homogeneity</td>
<td>37.2434</td>
</tr>
<tr>
<td>10</td>
<td>Mean of Red pixels</td>
<td>36.0704</td>
</tr>
<tr>
<td>11</td>
<td>Standard deviation of Grayscale</td>
<td>32.8446</td>
</tr>
<tr>
<td>12</td>
<td>Average Length of Edges</td>
<td>31.9648</td>
</tr>
<tr>
<td>13</td>
<td>Correlation</td>
<td>31.085</td>
</tr>
<tr>
<td>14</td>
<td>Number of Corners</td>
<td>29.0323</td>
</tr>
<tr>
<td>15</td>
<td>Euclidean Distance</td>
<td>20.2346</td>
</tr>
</tbody>
</table>
Again, OneR shows *Energy* is the most significant video feature, while *Contrast* is ranked a bit higher than in Info-gain. The OneR diagram show even less variation in the significance of each feature but again the Euclidean Distance one is ranked last, yet still significant.

Overall the evaluation criteria show that the features selected for the video classification were very effective and aptly selected which also explains the very high accuracy results in this task.
3.5 Audio Classification Models

3.5.1 Classifiers

As mentioned above, for the sound classification task we required much a much bigger dataset, since sound classification is more a demanding task by nature. For the 5 to 7 minutes of sound for each genre, we used two window lengths, one every second and one every half a second, creating two datasets with 3707 and 7412 instances respectively. All the sound files used a sampling rate 44100Hz και 16bit dynamic range. In general, we expected the accuracy of the classifiers in this task to significantly lower than ones used for the video features. Many different classifiers were tested and evaluated in WEKA, but we decided to keep only the ones that provided more than 70% accuracy scores. Again, we tested 3 different numbers of folds in cross validation, 10, 5 and 3 as shown in table 6:

Table 6: Sound classifiers’ comparison

<table>
<thead>
<tr>
<th>Window Length</th>
<th>Logistic Regression</th>
<th>MLP</th>
<th>Random Forest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 second</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10-Fold</td>
<td>73.7</td>
<td>74.2</td>
<td>77.3</td>
</tr>
<tr>
<td>5-Fold</td>
<td>73.4</td>
<td>72.8</td>
<td>76.6</td>
</tr>
<tr>
<td>3-Fold</td>
<td>72.2</td>
<td>72.9</td>
<td>75.6</td>
</tr>
<tr>
<td>0.5 seconds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10-fold</td>
<td>69.3</td>
<td>73.4</td>
<td>75.9</td>
</tr>
<tr>
<td>5-fold</td>
<td>69.3</td>
<td>72.9</td>
<td>75.6</td>
</tr>
<tr>
<td>3-fold</td>
<td>69.2</td>
<td>72.1</td>
<td>74.9</td>
</tr>
</tbody>
</table>
We go into more detail for each of the two different windows lengths shown in tables 7 and 8.

Table 7: 1 second window length

<table>
<thead>
<tr>
<th>Window Length</th>
<th>Logistic Regression</th>
<th>MLP</th>
<th>Random Forest</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-fold</td>
<td>73.7</td>
<td>74.2</td>
<td>77.3</td>
</tr>
<tr>
<td>5-fold</td>
<td>73.4</td>
<td>72.8</td>
<td>76.6</td>
</tr>
<tr>
<td>3-fold</td>
<td>72.2</td>
<td>72.9</td>
<td>75.6</td>
</tr>
</tbody>
</table>

The results are summarized in the following graphs:

The following diagrams shows the overall accuracy of all the classifiers for the 1 second window length.
10 fold classification (1 second)

- Logistic Regression: 73.7
- MLP: 74.2
- Random Forest: 77.3

5 fold classification (1 second)

- Logistic Regression: 73.4
- MLP: 72.8
- Random Forest: 76.6

3 fold classification (1 second)

- Logistic Regression: 72.2
- MLP: 72.9
- Random Forest: 75.6
For all the 3 different number of selected folds, Random Forest algorithm provided the best accuracy, while Logistic Regression and MLP provided quite similar results, with MLP surpassing Logistic for 3 and 10 folds while Logistic, surpassed MLP for 5 folds cross validation. The other different classifiers tested in WEKA, like SVM and KNN scored less than 70% so we decided to ignore them.

Table 8: 0.5 seconds window length

<table>
<thead>
<tr>
<th></th>
<th>Logistic Regression</th>
<th>MLP</th>
<th>Random Forest</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-fold</td>
<td>69.3</td>
<td>73.4</td>
<td>75.9</td>
</tr>
<tr>
<td>5-fold</td>
<td>69.3</td>
<td>72.9</td>
<td>75.6</td>
</tr>
<tr>
<td>3-fold</td>
<td>69.2</td>
<td>72.1</td>
<td>74.9</td>
</tr>
</tbody>
</table>
Again, the higher number of folds selected, the higher the accuracy of each classifier was, with Random Forest overperforming the others, achieving slightly above 77%.
Comparisons:

As shown in the diagrams above, for the 0.5 seconds window length Logistic Regression showed a significant decrease in accuracy, falling under the 70% threshold. MLP classifier was the least affected by the sample rate, decreasing only slightly for 10 and 3 folds, and actually scoring 0.1% higher for 5 folds. Random Forest witnessed a drop of around 1% for each fold but was still the best classifier for both the window lengths.
3.5.2 Inner Class Accuracy

Although the overall accuracy in the sound classification wasn’t as good as in video, we still wanted to see in more detail the reason behind it and try to identify and explain the most misclassified genres.

Table 9 shows the accuracy of each classifier for all the video game classes.

Table 9: Sound classifiers’ inner class accuracy

<table>
<thead>
<tr>
<th></th>
<th>MLP</th>
<th>Random Forest</th>
<th>Logistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>70%</td>
<td>76%</td>
<td>70%</td>
</tr>
<tr>
<td>2</td>
<td>95%</td>
<td>96%</td>
<td>89%</td>
</tr>
<tr>
<td>3</td>
<td>67%</td>
<td>73%</td>
<td>62%</td>
</tr>
<tr>
<td>4</td>
<td>55%</td>
<td>63%</td>
<td>54%</td>
</tr>
<tr>
<td>5</td>
<td>56%</td>
<td>54%</td>
<td>62%</td>
</tr>
<tr>
<td>6</td>
<td>63%</td>
<td>64%</td>
<td>70%</td>
</tr>
<tr>
<td>7</td>
<td>77%</td>
<td>80%</td>
<td>77%</td>
</tr>
<tr>
<td>8</td>
<td>83%</td>
<td>88%</td>
<td>85%</td>
</tr>
<tr>
<td>9</td>
<td>90%</td>
<td>92%</td>
<td>91%</td>
</tr>
<tr>
<td>10</td>
<td>65%</td>
<td>65%</td>
<td>59%</td>
</tr>
</tbody>
</table>
Arcade Games (class 2) showed an extremely high accuracy. This can be explained by the very distinctive music and sound in these games and was something we expected to happen.

Racing and Sports games (class 8 and 9 respectively), also had a very high class accuracy. Racing games contain loud care engine sounds and fast paced music, which resulted in a good accuracy, while as mentioned, Sport games are accompanied by commentary, human voice, which led to more than 90% accuracy.

Platform games (class 7), although not great, they scored slightly below 80%. The platform games selected, similar to the arcade ones, do have some unique sounds, but the gameplay also contains many various ones that we think reduced the overall accuracy.

The other genres, Action/RPG, Fighting, FPS, MMORPG, MOBA, Strategy showed significantly lower accuracy scores. Most of these games don’t have any specific and unique sounds, and generally the sound and music depend on the exact moment of the gameplay and what kind of action the player performs at that time.

Checking the confusion matrices in WEKA, we saw apart from the 3 best class scores we mentioned, instances of the other genres were indeed missclassified from one to another.

Below is the confusion matrix produced in Weka for the highest Scoring classifier, Random Forest (1 second window length and 10 fold cross validation).

```
    a  b  c  d  e  f  g  h  i  j <-- classified as
  a  318 3  22 32 3 18  4  3  5  6 | a = 1
  b  4395 2  0  0  0  6  2  0  2  0 | b = 2
  c   17 1 255 27 9 10  6  7  4  9 | c = 3
  d   39 1 31 231 15 10  2  3 12  4 | d = 4
  e   18 1 24 24 167 15 16 13  8 14 | e = 5
  f   37 0 14 9 12 211 1 23 14  5 | f = 6
  g   12 4 11 5 18 1 371 18  2 20 | g = 7
  h    0 0  7 0  5  5 13 293  2  6 | h = 8
  i   17 0 4  8 1  6  0  4  427  3 | i = 9
  j    9 4  5 16 16 3 22 10 18 197 | j = 10
```
As we see, classes 1, 8 and 9 have extremely high accuracy scores, and very few false positive instances. On the other hand, in the other classes we see many False positive instances.

Additionally, as opposed to what we saw for the video classifiers, for audio the case is not exactly the same. While for the video part, Random Forest was the most stable classifier outperforming the others for every class, for the audio classification we see that for some classes MLP and Logistic Regression offered better results. Random Forest actually equalized the overall lowest inner class accuracy of 54%, sharing it with Logistic Regression.

However, Random Forest was overall the best classifier for the audio classification as well.
3.5.3 Audio Features Evaluation

This time, for the audio classification more than double the features compared to the video task were used. Therefore, their significance was expected to vary much more. The same two evaluation criteria was selected again, Info Gain and OneR as presented in tables 10 and 11.

Table 10: Audio features Info-Gain Evaluation

<table>
<thead>
<tr>
<th></th>
<th>Feature Description</th>
<th>Info Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RMS</td>
<td>0.8259</td>
</tr>
<tr>
<td>2</td>
<td>Spread</td>
<td>0.4553</td>
</tr>
<tr>
<td>3</td>
<td>ZeroCross</td>
<td>0.4357</td>
</tr>
<tr>
<td>4</td>
<td>RollOff (threshold 0.3)</td>
<td>0.4226</td>
</tr>
<tr>
<td>5</td>
<td>RollOff (threshold 0.9)</td>
<td>0.4156</td>
</tr>
<tr>
<td>6</td>
<td>Entropy</td>
<td>0.3772</td>
</tr>
<tr>
<td>7</td>
<td>RollOff (threshold 0.7)</td>
<td>0.3733</td>
</tr>
<tr>
<td>8</td>
<td>1(^{st}) MFCC</td>
<td>0.3638</td>
</tr>
<tr>
<td>9</td>
<td>RollOff (threshold 0.5)</td>
<td>0.3628</td>
</tr>
<tr>
<td>10</td>
<td>Brightness (threshold 4000)</td>
<td>0.3618</td>
</tr>
<tr>
<td>11</td>
<td>Kurtosis</td>
<td>0.3562</td>
</tr>
<tr>
<td>12</td>
<td>Brightness (threshold 8000)</td>
<td>0.3559</td>
</tr>
<tr>
<td>13</td>
<td>Brightness (threshold 500)</td>
<td>0.3513</td>
</tr>
<tr>
<td>14</td>
<td>Skewness</td>
<td>0.3398</td>
</tr>
<tr>
<td>15</td>
<td>2(^{nd}) MFCC</td>
<td>0.333</td>
</tr>
<tr>
<td>16</td>
<td>Brightness (threshold 3000)</td>
<td>0.3287</td>
</tr>
<tr>
<td>17</td>
<td>Centroid</td>
<td>0.3254</td>
</tr>
<tr>
<td>18</td>
<td>Peaks</td>
<td>0.3129</td>
</tr>
<tr>
<td>19</td>
<td>Brightness (threshold 2000)</td>
<td>0.2953</td>
</tr>
<tr>
<td>20</td>
<td>Brightness (threshold 1500)</td>
<td>0.28</td>
</tr>
<tr>
<td>21</td>
<td>Brightness (threshold 1000)</td>
<td>0.2746</td>
</tr>
<tr>
<td>22</td>
<td>6(^{th}) MFCC</td>
<td>0.2143</td>
</tr>
<tr>
<td>23</td>
<td>7(^{th}) MFCC</td>
<td>0.2103</td>
</tr>
<tr>
<td>No.</td>
<td>Feature</td>
<td>Value</td>
</tr>
<tr>
<td>-----</td>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>24</td>
<td>Flatness</td>
<td>0.2072</td>
</tr>
<tr>
<td>25</td>
<td>3rd MFCC</td>
<td>0.1733</td>
</tr>
<tr>
<td>26</td>
<td>13th MFCC</td>
<td>0.1674</td>
</tr>
<tr>
<td>27</td>
<td>4th MFCC</td>
<td>0.1638</td>
</tr>
<tr>
<td>28</td>
<td>10th MFCC</td>
<td>0.1283</td>
</tr>
<tr>
<td>29</td>
<td>11th MFCC</td>
<td>0.1175</td>
</tr>
<tr>
<td>30</td>
<td>9th MFCC</td>
<td>0.1</td>
</tr>
<tr>
<td>31</td>
<td>5th MFCC</td>
<td>0.0932</td>
</tr>
<tr>
<td>32</td>
<td>Low Energy</td>
<td>0.0874</td>
</tr>
<tr>
<td>33</td>
<td>12th MFCC</td>
<td>0.0836</td>
</tr>
<tr>
<td>34</td>
<td>Event Density</td>
<td>0.0796</td>
</tr>
<tr>
<td>35</td>
<td>8th MFCC</td>
<td>0.0694</td>
</tr>
</tbody>
</table>
Overall the most significant feature was by far the RMS (root mean square) energy of the signal which is a Time-Based feature.

After that, the influence of the following features didn’t show any great variations until reaching the mid of the table with the top 10 positions being filled with both Time-Based and Spectral-Based features.

The different thresholds of the Roll-Off frequencies were ranked pretty high while the different Spectral Brightness one’s had a varying significance, but overall, they were all ranked above average.

Despite the first Mel Frequency Cepstral Coefficient (MFCC) ranking very high (8th) the rest were ranked below the 15th and actually most of them were in the bottom positions accompanied by the Low Energy and Event Density features.

Table 11: Audio Features OneR Evaluation

<table>
<thead>
<tr>
<th></th>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RMS</td>
<td>34.2325</td>
</tr>
<tr>
<td>2</td>
<td>RollOff (threshold 0.3)</td>
<td>23.7928</td>
</tr>
<tr>
<td>3</td>
<td>Spread</td>
<td>23.3342</td>
</tr>
<tr>
<td>4</td>
<td>ZeroCross</td>
<td>23.0375</td>
</tr>
<tr>
<td>5</td>
<td>Roll Off (threshold 0.5)</td>
<td>22.444</td>
</tr>
<tr>
<td>6</td>
<td>Roll Off (threshold 0.9)</td>
<td>22.2822</td>
</tr>
<tr>
<td>7</td>
<td>Roll Off (threshold 0.7)</td>
<td>21.9045</td>
</tr>
<tr>
<td>8</td>
<td>2nd MFCC</td>
<td>21.7426</td>
</tr>
<tr>
<td>9</td>
<td>Brightness (threshold 8000)</td>
<td>21.6347</td>
</tr>
<tr>
<td>10</td>
<td>Brightness (threshold 4000)</td>
<td>21.6347</td>
</tr>
<tr>
<td>11</td>
<td>Entropy</td>
<td>21.2031</td>
</tr>
<tr>
<td>12</td>
<td>Peaks</td>
<td>20.5827</td>
</tr>
<tr>
<td>13</td>
<td>Brightness (threshold 1500)</td>
<td>20.4748</td>
</tr>
<tr>
<td>14</td>
<td>Centroid</td>
<td>20.2859</td>
</tr>
<tr>
<td>15</td>
<td>Skewness</td>
<td>19.9083</td>
</tr>
<tr>
<td>16</td>
<td>Kurtosis</td>
<td>19.5846</td>
</tr>
<tr>
<td>17</td>
<td>Brightness (threshold 3000)</td>
<td>19.3148</td>
</tr>
<tr>
<td></td>
<td>Feature</td>
<td>Value</td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>18</td>
<td>1st MFCC</td>
<td>19.072</td>
</tr>
<tr>
<td>19</td>
<td>6th MFCC</td>
<td>18.9911</td>
</tr>
<tr>
<td>20</td>
<td>Brightness (threshold 2000)</td>
<td>18.3437</td>
</tr>
<tr>
<td>21</td>
<td>Brightness (threshold 500)</td>
<td>18.1279</td>
</tr>
<tr>
<td>22</td>
<td>Event Density</td>
<td>18.0469</td>
</tr>
<tr>
<td>23</td>
<td>7th MFCC</td>
<td>18.0469</td>
</tr>
<tr>
<td>24</td>
<td>Brightness (threshold 1000)</td>
<td>17.8581</td>
</tr>
<tr>
<td>25</td>
<td>4th MFCC</td>
<td>17.8311</td>
</tr>
<tr>
<td>26</td>
<td>Flatness</td>
<td>17.6963</td>
</tr>
<tr>
<td>27</td>
<td>Low Energy</td>
<td>17.4535</td>
</tr>
<tr>
<td>28</td>
<td>13th MFCC</td>
<td>16.86</td>
</tr>
<tr>
<td>29</td>
<td>3rd MFCC</td>
<td>16.4823</td>
</tr>
<tr>
<td>30</td>
<td>10th MFCC</td>
<td>15.3493</td>
</tr>
<tr>
<td>31</td>
<td>9th MFCC</td>
<td>14.7019</td>
</tr>
<tr>
<td>32</td>
<td>5th MFCC</td>
<td>14.4861</td>
</tr>
<tr>
<td>33</td>
<td>12th MFCC</td>
<td>14.3243</td>
</tr>
<tr>
<td>34</td>
<td>11th MFCC</td>
<td>14.2703</td>
</tr>
<tr>
<td>35</td>
<td>13th MFCC</td>
<td>13.5959</td>
</tr>
</tbody>
</table>

![OneR Diagram](image-url)
With the OneR criterion, once again RMS is by far the most significant feature. Roll Off frequencies and spectral Brightness are again ranked pretty high. In general a similar pattern is followed, with most features ranking around the same area. We observe however that with OneR criterion, the most important MFCC is the 2\textsuperscript{nd} and the rest are actually ranked pretty low.
Conclusions

The goal of the work presented in this paper was to perform a Multimedia data analysis and classification using multimedia features for Video Game genre classification. The first step was to determine the video game genres we would use for our work and explore the possible already existing works similar to our task’s nature in order to have some initial guidelines.

Based on the video game material we gathered, we created two datasets, one containing the video features and a second one with the audio ones. We wanted to run and test various classifiers for both datasets, trying to identify the most accurate ones in each case, while also providing evaluations for all the features we used. After creating the datasets in MATLAB, Weka was used to train and test various classifiers and the most accurate ones were identified. In both cases Random Forest classifiers score the best accuracy results, with MLP and Logistic Regression scoring slightly lower. The classification tasks were performed using different number of folds, for cross validation, 10, 5 and 3, and almost in every case, the more folds selected the higher the accuracy was, for all the classifiers. For the sound classification we also tested two different window lengths rates for the selected sounds for each game. The 1 second length, despite resulting into a smaller dataset (exactly half the size) offered better accuracy scores for all the classifiers tested.

The overall results showed that video/image classification was much more accurate than audio classification, achieving as high as around 91% of accuracy while for the sound we barely surpassed 77%.

For video, even the classes with the most misclassified instances can be explained, as we mentioned, by the nature of the games that fall in each of these genres. FPS and Action/Adventure games for example don’t have any major differences apart from the player’s point of view during gameplay since both utilize fast paced movement and use a HUD with similar information.

The evaluation of the features we used, using both InfoGain and OneR methods, showed that all the features we used played an important role in the final scores, with the Energy feature ranking top in both cases.
We didn’t use any object and shape recognition features for the video classification and in a future expansion of this work, these features can possibly help increasing the accuracy even more.

Sound classification on the other hand, is a more demanding and difficult task and requires big differences among the classes in order to achieve an extremely high accuracy score, similar to the video one. Despite some of the genres we used for this work, having some very unique and special sounds accompanying them, like Arcade games genre, most do not, so as expected the classifiers could not separate games like Action, Fps or Fighting very successfully. On the other hand, for the genres like Arcade with the very iconic and distinct audio, the results showed that the features we selected provided almost a perfect score. Overall, despite the sound classification results being quite lower than the video ones, the inner class accuracies presented, provided us with the reason behind it and we still regard them as effective.

The evaluation of the audio features using the same two methods, InfoGain and OneR, showed that the most effective feature we used was the RMS energy. The different Roll-Off frequencies and Brightness thresholds selected were in general quite important features, while surprisingly MFCCs were ranked somewhat lower than expected.
**Future Work**

In terms of what can be further used and explored in order to improve the accuracy of our classification, *motion vectors* can be implemented. For the video classification, we witnessed that the poorest feature was the Euclidean distance one, which was actually conceived and manually constructed by us. This shows that a better feature regarding the motion estimation and the frame succession is needed to possibly further increase the classification accuracy.

The steps of our work and the process we followed can of course be used for the classification of any other game as long as it is assigned a class. Because many classes offer many similarities with each other, in a future implementation sub-classes can be introduced. After the initial classification of a game, in a main class, a new classification process and be executed that classifies the games into each sub-class. However, since the genre of each game can be partially considered a subjective issue, genre similarities are hard to be extinguished in every case.
References


[33] B. V. V. Sri Raj Dutt, P. Agrawal, S. Nayak, “Scene Classification in Images”, 2005

[34] B. T. Truong, C. Dorai, Automatic Genre Identification for Content-Based Video Categorization, 2000


[38] X. Zhou, J. Wang, “Feature Selection for Image Classification Based on a New Ranking Criterion”, 2015


[48] Weka 3: Data Mining Software in Java, available at:
https://www.cs.waikato.ac.nz/ml/weka/