Automatic Detection of TV Commercials

Guilherme Mateus Moreira

Thesis to Obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. Maria Paula dos Santos Queluz Rodrigues
Prof. António José Castelo Branco Rodrigues

Examination Committee
Chairperson: Prof. José Eduardo Charters Ribeiro da Cunha Sanguino
Supervisor: Prof. Maria Paula dos Santos Queluz Rodrigues
Member of the Committee: Prof. Luís Eduardo de Pinho Ducla Soares

June 2018
Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.
Acknowledgments

In first place, I would like to thank my father Afonso, for all the long guiding and supporting conversations on the phone, my mother Teresa, for always helping to stay focused and find my own path while keeping positive, and my brother Francisco, for always making me remember of what is really important. This journey would not be possible without any of you and I could not be more grateful to have you.

A special thanks to Professors Paula Queluz and António Rodrigues, for their availability, advices and support throughout this thesis and for their constructive criticism. Thank you for the immense patience to improve my writing skills.

A special word to Sara, who was always there for me.

To my dearest friends from Terceira Island – these years would not be the same without the three of you.

To my “war companions” of the IT room, thanks for putting up with me in some stressful moments during these last months. All the talks, shared moments and meals helped me to regain my strength and confidence to continue this task the best way possible, and fellow IST colleagues.

To my friends from Abrantes, for the amazing moments we have shared since we were little until now.

I also want to thank Instituto de Telecomunicações for providing me with the required means for the completion of this dissertation.
Abstract

In the world of advertisement, the largest share of investment is in TV commercials; accordingly, for large corporations, this type of advertisement is still of major importance in their main goal of selling products/services to the public. Commercials are also of major importance to public TV stations since it is the result of the revenue they generate which allows those stations to fund a variety of in-house TV programs. Due to the large sums of money being moved on TV commercials, national and international regulations were created, to avoid a wrong use of the commercials by the broadcasters; the enforcement of these laws created certain characteristics specific to TV commercials like, as an example, the maximum share of TV commercials per hour of broadcasting. The automatic detection of commercials may allow to fulfill several needs: i) of the users, who typically wish to eliminate the transmitted commercials from their recorded TV programs; ii) of the regulators, who need to verify if the TV advertising rules are met; iii) of the advertisers, who want to check if their contracts with the broadcasters have been fulfilled.

TV commercials are a special type of video content, with some rather specific characteristics that have been exploited in automatic commercials detection solutions. After an overview of past methodologies for TV commercials detection, a new approach, based on a Convolutional Neural Network (CNN), is proposed; this CNN was trained with thousands of manually selected images, extracted from more than 105 hours of recorded video, from 12 different TV channels. However, the developed system also has the capability of detecting when it is performing poorly (e.g., whenever a new TV channel is being analyzed) and of initiating a re-learning process, without human intervention. The proposed method was assessed with a video dataset that included sequences extracted from TV channels considered in the training stage, as well as new ones. For the TV channels included in the training stage, the minimum observed accuracy in commercials detection was 92%; this value decreases when a new TV channel is analyzed (e.g., a TV channel for which the system was not trained for) but may still reaches accuracy values around 90% in several cases.

Keywords: TV Commercials; commercial blocks; automatic detection of commercials; convolutional neural network; TV content classification.
Resumo

Na área da publicidade, a maior parte do investimento é aplicado em anúncios transmitidos em TV; com efeito, esta solução de marketing é ainda hoje em dia uma forma das maiores empresas darem a conhecer os seus produtos e serviços ao grande público. Os anúncios são também de grande importância para as estações públicas de transmissão de TV, uma vez que é o resultado da receita que geram que permite que essas emissoras financiem uma variedade de programas internos. Em paralelo, foram criadas regulamentações nacionais e internacionais para evitar o uso incorrecto da publicidade; a aplicação dessas leis criou certas características específicas nos anúncios de TV como, por exemplo, a duração máxima de blocos comerciais por hora de transmissão. A detecção automática de anúncios pode permitir satisfazer diversas necessidades: i) dos utilizadores, que tipicamente desejam eliminá-los dos seus programas de TV gravados; ii) dos reguladores, que precisam de verificar se as regras de publicidade são cumpridas; iii) dos anunciantes, que querem verificar se os contratos com as estações emissoras foram cumpridos.

Os anúncios de TV são um tipo especial de conteúdo de vídeo, com algumas características bastante específicas que têm sido exploradas em diversos métodos de detecção automática. Após uma revisão das metodologias existentes para a detecção de anúncios em TV, propõe-se uma nova abordagem baseada em Redes Neurais Convolucionais (CNN); a CNN usada foi treinada com milhares de imagens seleccionadas manualmente, extraídas de mais de 105 horas de vídeo gravado, de 12 canais de TV diferentes. No entanto, o sistema desenvolvido tem também a capacidade de detectar as situações em que o seu desempenho se deteriora (por exemplo, sempre que um novo canal de TV é analisado) e de iniciar um processo de reaprendizagem, sem intervenção humana. O método proposto foi avaliado com um conjunto de dados de vídeo que inclui sequências extraídas de canais de TV considerados na fase de treino da CNN, bem como de novos canais de TV (para os quais o sistema não foi treinado). Para os canais de TV incluídos na fase de treino, a precisão mínima observada na detecção de anúncios foi de 92%; esse valor diminui quando um novo canal de TV é analisado (por exemplo, um canal de TV para o qual o sistema não foi treinado), mas ainda assim atinge valores de precisão próximos de 90% em diversos casos.

Keywords: Anúncios de TV; blocos comerciais; detecção automática de publicidade; rede neural convolucional; classificação de conteúdo de TV.
# Table of Contents

Declaration ...................................................................................................................... iii  
Acknowledgments .......................................................................................................... iv  
Abstract .......................................................................................................................... vi  
Resumo ............................................................................................................................ vii  
Table of Contents ............................................................................................................. viii  
Index of Figures ............................................................................................................... xi  
Index of Tables ................................................................................................................ xiv  
Acronyms........................................................................................................................ xv  
1. Introduction................................................................................................................... 1  
   1.1 Context and Motivation.............................................................................................. 1  
   1.2 Objectives ................................................................................................................. 2  
   1.3 Main Contributions..................................................................................................... 2  
   1.4 Thesis Structure ......................................................................................................... 3  
2. TV Commercials Detection Review ............................................................................ 5  
   2.1 Introduction................................................................................................................. 5  
   2.2 Legal Framework......................................................................................................... 5  
   2.3 Typical Structure of a Commercial Block .................................................................. 5  
      2.3.1 Intrinsic characteristics ......................................................................................... 6  
      2.3.2 Extrinsic characteristics ....................................................................................... 7  
   2.4 Methodologies for Commercials Detection ............................................................. 7  
      2.4.1 Knowledge-based detection ................................................................................. 7  
      2.4.2 Repetition-based detection .................................................................................. 9  
3. TV Logo Detection Methods Overview ..................................................................... 11  
   3.1 Introduction................................................................................................................. 11  
   3.2 Automatic Detection of Commercial Blocks Based on TV Channel Logos Detection. 11  
   3.3 A TV Logo Detection and Recognition Method Based on SURF Feature and Bag-of-  
       Words Model ............................................................................................................. 15  
   3.4 Automatic Video Logo Detection Based on a Bayesian Classifier and Neural Networks  
       ......................................................................................................................................... 17  
   3.5 Spatial HOG Based TV Logo Detection in a Single Frame....................................... 19
4. Proposed Solution for TV Commercials Detection ................................................. 23
   4.1 Introduction ........................................................................................................ 23
   4.2 TV Data Characterization .................................................................................. 23
   4.3 Architecture Walkthrough ............................................................................... 25
   4.4 Shot Change Detection and Video Segmentation ............................................. 26
   4.5 Post-Processing ................................................................................................ 29
      4.5.1 Example Frame Extraction ........................................................................ 29
      4.5.2 Regions of Interest Extraction ................................................................... 30
      4.5.3 Edge Detection ............................................................................................ 32
      4.5.4 Edge Fusion ................................................................................................ 35
      4.5.5 Static Area Detection – Edges Intersection .............................................. 36
   4.6 Automatic Video Shot CNN-Based Classification ........................................... 39
      4.6.1 Convolutional Neural Network – Inception Architecture ......................... 40
      4.6.2 Manual Video Shot Classification .............................................................. 41
      4.6.3 Automatic Video Shot Classification – Classification Phase ...................... 42
   4.7 Temporal Coherency-Based Reclassification ..................................................... 45
      4.7.1 Input Parameters ......................................................................................... 46
      4.7.2 Reclassification ........................................................................................... 46
   4.8 Automatic Learning .......................................................................................... 49
      4.8.1 Classification Consistency Analysis ............................................................ 50
      4.8.2 Representative Images Selection ............................................................... 51
5. Performance Assessment ....................................................................................... 53
   5.1 Introduction ........................................................................................................ 53
   5.2 Video Libraries Database .................................................................................. 53
      5.2.1 Video Library for the SCD Analysis A ....................................................... 53
      5.2.2 Training Video Library ............................................................................... 54
      5.2.3 Testing Video Library .................................................................................. 54
      5.2.4 Automatic Learning Library ....................................................................... 55
   5.3 Performance Assessment Metrics ..................................................................... 55
   5.4 Shot Change Detection Assessment ................................................................ 56
      5.4.1 SCD Analysis A ........................................................................................... 56
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4.2</td>
<td>SCD Analysis B</td>
<td>60</td>
</tr>
<tr>
<td>5.5</td>
<td>Initial CNN-Based Classification Assessment</td>
<td>61</td>
</tr>
<tr>
<td>5.6</td>
<td>Temporal Coherency-Based Reclassification</td>
<td>64</td>
</tr>
<tr>
<td>5.7</td>
<td>Automatic Learning Assessment</td>
<td>67</td>
</tr>
<tr>
<td>6.</td>
<td>Summary and Future Work</td>
<td>73</td>
</tr>
<tr>
<td>6.1</td>
<td>Summary and Conclusions</td>
<td>73</td>
</tr>
<tr>
<td>6.2</td>
<td>Future Work</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td><strong>References</strong></td>
<td>77</td>
</tr>
<tr>
<td>A.</td>
<td>Basic Concepts of Convolutional Neural Networks (CNN)</td>
<td>81</td>
</tr>
<tr>
<td>A.1</td>
<td>CNN Composition</td>
<td>81</td>
</tr>
<tr>
<td>A.2</td>
<td>CNN Learning Process</td>
<td>84</td>
</tr>
<tr>
<td>B.</td>
<td>Training and Testing Video Library</td>
<td>85</td>
</tr>
<tr>
<td>B.1</td>
<td>Training Library</td>
<td>85</td>
</tr>
<tr>
<td>B.2</td>
<td>Testing Library</td>
<td>87</td>
</tr>
</tbody>
</table>
Index of Figures

Figure 2.1 – Typical Structure of a Commercial Block. ................................................................. 6
Figure 2.2 – **Left:** Initial separator broadcasted by “RTP”; **Center:** “SIC” soap opera self-promotion; **Right:** Final separator broadcasted by “RTP”. ........................................................................ 6
Figure 2.3 – Illustration of text detection results [14]. ................................................................. 9
Figure 3.1 – Flowchart of the commercials detection method proposed in [3][4]. .................... 12
Figure 3.2 – Heat map of the chrominances variance [4]. ......................................................... 13
Figure 3.3 – Block diagram for the TV logo detection and recognition algorithm proposed in [17]. ................................................................................................................................. 16
Figure 3.4 – Flowchart of the video logo detection method proposed in [21]. ......................... 19
Figure 3.5 – Flowchart of the TV logo detection method proposed in [22]. ............................. 21
Figure 4.1 – “TVI” channel, with visible logo during a self-promotion. .................................. 24
Figure 4.2 – “Correio da Manhã TV” channel with some DoG elements: TV logo, Clock, Top and Bottom bar. ........................................................................................................ 24
Figure 4.3 – “RTP 1” channel detail, with the channel website, clock, bottom bar and sign language translator .............................................................................................................. 24
Figure 4.4 – Global Solution flowchart. ..................................................................................... 25
Figure 4.5 – SCD section Flowchart. ........................................................................................... 27
Figure 4.6 – A window (orange) with a WS of five. ................................................................... 27
Figure 4.7 – Luminance histogram comparison in an initial temporal window. ....................... 28
Figure 4.8 – Graphical representation of $WTh$ versus $\mu Fd$. ................................................ 29
Figure 4.9 – Post-Processing flowchart. ..................................................................................... 30
Figure 4.10 – One video frame from the SIC channel. ............................................................ 31
Figure 4.11 – The four ROIs stitched together, after being extracted with $ScaleX = 5$ and $ScaleY = 4$........................................................................................................................................ 31
Figure 4.12 – **Left:** “Sport TV+” channel logo and **Right:** “Nickelodeon” channel logo. **Top:** original channel logo; **Middle:** upper right corner ROI detail; **Bottom:** ROI detail after edge detection. 31
Figure 4.13 – “RTP1” channel ROIs stitched together. **Left:** frame content; **Right:** edges extracted with the Canny edge detector. .............................................................................. 32
Figure 4.14 – A frame of the “RTP 1” channel, with a TV logo with low contrast with the background. ...................................................................................................................... 32
Figure 4.15 – Image gradient and level-line example [25]. ....................................................... 33
Figure 4.16 – **Left:** original image **Center:** level-line field **Right:** line support regions (colored) [25]. ......................................................................................................................... 34
Figure 4.17 – Region growing error [25]. .................................................................................. 34
Figure 4.18 – ROI edges obtained with: **Left:** Canny edge detector; **Right:** LSD edge detector ........................................................................................................................................... 35
Figure 4.19 – Eurosport 1 channel simplified edged fusion example **Top**: four stitched ROIs with their respective canny edge detection results; **Bottom**: high detail edge fusion of the frames above. .......................................................... 36

Figure 4.20 – Trailing effect example. **Top**: two example frames (initial and final) of the same shot; **Bottom**: resulting binary image after the OR logical operation between edges images. ............. 36

Figure 4.21 – “Disney Channel” edge intersection simplified example **Top**: four shot images; **Bottom**: edge intersection of the shot images above. ........................................................... 37

Figure 4.22 – “Discovery Channel” simplified edge intersection example. **Top**: four shot images, the first three from regular programming and the last from non-regular programming; **Bottom**: edge intersection of the frames above. ........................................................... 38

Figure 4.23 – Edge intersection sliding window with a NSW of five, to obtain SI 5. ................. 38

Figure 4.24 – Edge intersection to obtain SI 6. ................................................................... 38

Figure 4.25 – Automatic CNN-based video shot classification flowchart. .......................... 39

Figure 4.26 – A final shot image from the “RTP1” channel, labeled as regular shot, with Teletext code, clock, and sign language translator. ......................................................... 41

Figure 4.27 – Example of a discarded regular shot image, from the “TVI” channel - TV Logo is unrecognizable due to unwanted edge elements......................................................... 42

Figure 4.28 – A final shot image obtained for a video shot of the “RTP1” channel. ................. 42

Figure 4.29 – **Left**: “RTP” symbol removed; **Center**: TV logo number removed; **Right**: Teletext number removed. .............................................................................. 43

Figure 4.30 – Removed elements - Top Left: TV logo and teletext number; Top Right: clock; Bottom Left: sign language translator; Bottom Right: website ........................................... 44

Figure 4.31 – Temporal Coherency-based Reclassification flowchart. ................................. 46

Figure 4.32 - Anchor Shot Based Reclassification approach example (in gray: non-regular shots; in orange: regular shots). .............................................................................. 48

Figure 4.33 – Simple Isolation Detection and Correction Approach example, (gray) non-regular, (orange) regular.............................................................................................. 48

Figure 4.34 – **Left**: “SIC Noticias” normal TV logo and **Right**: alternative logo. .................. 49

Figure 4.35 – Automatic Learning flowchart........................................................................ 50

Figure 4.36 – “Vicious cycle” of misclassified training shots.................................................. 51

Figure 5.1 – Discovery Channel TV logo detail. ................................................................. 55

Figure 5.2 – “Correio da Manhã TV” channel example of a preliminary shot image resulting from a false negative. ........................................................................... 57

Figure 5.3 – “Rtp1comPub” video Recall, Precision and F1 scores when changing the window size. ....................................................................................................................... 58

Figure 5.4 – “aBolaComPub” video Recall, Precision and F1 scores when changing the low threshold .................................................................................................................. 59

Figure 5.5 – “Rtp1ComPub” video Recall, Precision and F1 scores when changing the high threshold .................................................................................................................. 59
Figure 5.6 – “Biggs” video Recall, Precision and F1 scores when changing the number of histogram bins. ........................................................................................................ 59
Figure A.1 – Features matching [41].......................................................................................................................... 81
Figure A.2 – **Left:** Example image binary matrix **Center:** Filter matrix **Right:** Resulting Output matrix [40]........................................................................................................ 82
Figure A.3 – Max pooling operation on a **Left:** Original matrix with **Right:** Output matrix [40].. 83
Figure A.4 – Fully Connected Layer Example [42]........................................................................................................ 84
Index of Tables

Table 2.1 – Comparison between EU and Portuguese TV commercials legal framework [3]. .... 5
Table 3.1 – Global system performance results [4]........................................................................ 14
Table 3.2 – Video logo detection results [21]............................................................................... 19
Table 4.1 – DOGs elements removal effect on the confidence level. ........................................... 43
Table 4.2 – Cumulative elements removal impact on the confidence level. ................................. 44
Table 5.1 – Testing library for the SCD analysis A. ................................................................. 54
Table 5.2 – Automatic learning library. ....................................................................................... 55
Table 5.3 – SCD parameters used in [3]. ....................................................................................... 57
Table 5.4 – Final SCD parameters. .............................................................................................. 59
Table 5.5 – SCD analysis A results. ............................................................................................. 60
Table 5.6 – SCD Assessment with videos from the testing library. ............................................. 61
Table 5.7 – Parameters for each test condition. .......................................................................... 62
Table 5.8 – CNN training parameters. ....................................................................................... 62
Table 5.9 – Classification accuracy results. .................................................................................. 63
Table 5.10 – Computation time of SCD plus Post-processing modules. ...................................... 64
Table 5.11 – Temporal coherency-based classification parameter. ............................................. 65
Table 5.12 – Anchor shot approach accuracy improvements over initial classification. ............ 65
Table 5.13 – Anchor shot plus single isolation detection combined approaches improvements. ..... 66
Table 5.14 – Final temporal coherency-based classification accuracy results. ........................... 67
Table 5.15 – Testing condition 1 classification consistency analysis results, ordered by average classification change. .............................................................................. 68
Table 5.16 – Testing condition 6 classification consistency analysis results, ordered by average classification change. .............................................................................. 69
Table 5.17 – Automatic Learning module parameters. ............................................................... 70
Table 5.18 – Automatic learning results for the “Discovery_HD_CA” video. ............................... 70
Table 5.19 – After learning Discovery channel summarized results on the testing library. ......... 71
Table 5.20 – Accuracy results per video sequence, after each procedure. ................................. 72
Table B.1 – Training Library. ....................................................................................................... 85
Table B.2 – Testing Library Already-Trained subset.................................................................... 87
Table B.3 – Testing Library Not-Trained subset. .......................................................................... 88
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acc</td>
<td>Accuracy</td>
</tr>
<tr>
<td>ASCI</td>
<td>Audio Scene Change Indicator</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional Neural Network</td>
</tr>
<tr>
<td>Cb</td>
<td>Blue Chrominance</td>
</tr>
<tr>
<td>Cr</td>
<td>Red Chrominance</td>
</tr>
<tr>
<td>ERC</td>
<td>Entidade Reguladora para a Comunicação Social</td>
</tr>
<tr>
<td>EF</td>
<td>Example Frames</td>
</tr>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>F1</td>
<td>F1-Score</td>
</tr>
<tr>
<td>FN</td>
<td>False Negative</td>
</tr>
<tr>
<td>FNR</td>
<td>False Negative Rate</td>
</tr>
<tr>
<td>FPR</td>
<td>False Positive Rate</td>
</tr>
<tr>
<td>HOG</td>
<td>Histogram of Oriented Gradients</td>
</tr>
<tr>
<td>ITU-R</td>
<td>International Telecommunication Union Radio Regulation</td>
</tr>
<tr>
<td>LSD</td>
<td>Line Segment Detector</td>
</tr>
<tr>
<td>SCD</td>
<td>Shot Change Detection and Segmentation</td>
</tr>
<tr>
<td>SIC</td>
<td>Sociedade Independente de Comunicação – Portuguese private television station</td>
</tr>
<tr>
<td>SPM</td>
<td>Static pixels map</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machines</td>
</tr>
<tr>
<td>SURF</td>
<td>Speeded Up Robust Features</td>
</tr>
<tr>
<td>RELU</td>
<td>Rectified Linear Unit</td>
</tr>
<tr>
<td>RGB</td>
<td>Red, Green, Blue</td>
</tr>
<tr>
<td>RNN</td>
<td>Recurrent Neural Network</td>
</tr>
<tr>
<td>RTP</td>
<td>Rádio e Televisão Portuguesa – Portuguese public television station</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of Interest</td>
</tr>
<tr>
<td>TV</td>
<td>Television</td>
</tr>
<tr>
<td>TVI</td>
<td>Televisão Independente – Portuguese private television station</td>
</tr>
<tr>
<td>TN</td>
<td>True Negative</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>TP</td>
<td>True Positive</td>
</tr>
<tr>
<td>TPR</td>
<td>True Positive Rate</td>
</tr>
<tr>
<td>ILSVRC</td>
<td>ImageNet Large Scale Visual Recognition Challenge</td>
</tr>
</tbody>
</table>
1. Introduction

1.1 Context and Motivation

In the world of advertisement, the largest share of investment is in TV commercials, as stated by the Wall Street Journal in [1]; in fact, for large corporations, this type of advertisement is still of major importance in their main goal of selling products/services to the general public. The pinnacle of this are the Super Bowl Commercials, which in 2017 had an average cost of 5 million dollars per 30 seconds of transmission [2]. For large corporations to spend this amount of money, they expect a return on their investment; in the case of Super Bowl, it was watched, in recent years, by 110 to 115 million viewers [2], which gives companies assurance that their message was shown to a great number of people, and possibly convert some of them into buyers of products and services.

Commercials are also of major importance to public TV stations since it is the result of the revenue they generate which allows those stations to fund a variety of in-house TV programs. Because of the large sums of money being moved on TV commercials, national and international regulations were created, to avoid a wrong use of the commercials by the broadcasters; the enforcement of these laws created certain characteristics specific to TV commercials like, as an example, the maximum share of TV commercials in an hour of TV broadcast.

In terms of video content, TV commercials show quite distinctive characteristics when compared to other forms of programming, like high cut rates, loud sounds, recognizable audio jingles and high motion scenes, which are used to attract the audience attention as much as possible; some characteristics can also be imposed by legislation, like packing the commercials together into blocks, or removal of the TV station logo during the commercial block, being this last one of major importance to the work developed in this dissertation.

With the advent of digital TV, the barrier to develop and implement systems that detect a variety of specific content on TV has lowered significantly. In particular, many systems to automatically detect TV commercials were developed in recent years. The automatic detection of commercials may allow to fulfill several needs: i) of the users, who typically wish to eliminate the transmitted commercials from their recorded TV programs; ii) of the regulators, who need to verify if the TV advertising rules are met; iii) of the advertisers, who want to check if their contracts with the broadcasters have been fulfilled, checking clauses like “which”, “when” and “how many times” a given commercial is broadcasted [3].
1.2 Objectives

TV commercials detection is a research field were much work has been done in the past; however, with the evolution of both TV commercials and regular programs through the years, most of the previously developed systems are not optimized to today’s TV content. Therefore the main objective of this thesis is to develop a TV commercials detection system that reliably distinguishes commercial blocks from regular television programming, in current TV content, exploiting recently proposed Machine Learning (ML) algorithms.

As the main objective, the developed system should be able to accurately discriminate between commercials and programs, in TV channels for which it has been trained for; as a secondary, and more challenging objective, it should also detect the less reliable classifications – that may happen when a new TV channel is analyzed - and, without human intervention, gather additional information from the TV content to be used in a re-learning step, to improve the classification accuracy.

As a starting point, the structure of the TV commercials detection algorithm proposed in [3][4] is used, seeking to solve some of the limitations of this algorithm through the use of ML based solutions.

1.3 Main Contributions

Three main contributions can be identified in this dissertation:

- A vast library of video sequences captured from 21 different TV channels, amounting to more than 150 hours of video sequences, was created. For each video, a ground truth with the transitions between commercials and regular programming was manually obtained. Both the video library and ground truth will be made available for future works.

- The solution developed in this thesis, although with some steps based on prior work [3], presents a novel combination of ML algorithms with traditional time persistence based methods.

- The developed solution has the capability of detecting when it is performing poorly and of initiating a re-learning process, without human intervention.
1.4 Thesis Structure

This thesis is structured in six chapters. After the introduction, Chapter 2 describes the structure of a TV commercial block and overviews some relevant methods for TV commercials detection; this chapter also emphasizes that, due to the rapid development of new regulations and the abandonment of old practices by broadcasters, some of the proposed TV commercials detection methods no longer would produce reliable and accurate results. Chapter 3 is focused on TV commercials detection methods that exploit the presence/absence of the TV logo, with special emphasis on the solution proposed in [3][4]; all the other reviewed algorithms have a features extraction step followed by a machine learning model that performs the classification. In Chapter 4, the developed solution is described section-by-section, with the reasoning for every major design decision explained in detail. Chapter 5 presents the performance assessment methodology and results, from the video libraries assembled and the used metrics, till the assessment of every major step of the solution, ending with the obtained results analysis. Chapter 6 concludes this dissertation, with a summary of the developed work and some suggestions for future work.
2. TV Commercials Detection Review

2.1 Introduction

In this chapter, the main concepts about TV commercials are introduced, starting with the legal framework that regulates commercials broadcasted in TV stations (section 2.2). The structure of the commercial block is explained in section 2.3, along with a description of the main characteristics of TV commercials. In section 2.4, several TV commercials detection methods are overviewed.

2.2 Legal Framework

When analyzing the TV commercials legal framework that applies to Portugal, two sets of laws almost equal in substance can be identified. The European Union (EU) directive [5], stipulates that per hour of broadcasted TV no more than 20% can be spent on commercials and also the need of clear visual separators between regular programs and commercials. The Portuguese law expands the EU directive by requiring the inclusion of the word “Publicidade” in the initial separator and by creating a restriction in the sound level of the commercial, that must remain the same as in the regular programs [6][7] (cf. Table 2.1).

Table 2.1 – Comparison between EU and Portuguese TV commercials legal framework [3].

<table>
<thead>
<tr>
<th>Condition</th>
<th>Application Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum percentage of broadcasted commercials</td>
<td>20% per hour</td>
</tr>
<tr>
<td>Commercial block identification</td>
<td>Clearly distinguishable by optical and/or spatial means</td>
</tr>
<tr>
<td>Sound volume difference between commercials and regular programs</td>
<td>Not specified</td>
</tr>
<tr>
<td>Sound volume difference between commercials and regular programs</td>
<td>Not specified</td>
</tr>
<tr>
<td>Sound volume difference between commercials and regular programs</td>
<td>Not specified</td>
</tr>
</tbody>
</table>

2.3 Typical Structure of a Commercial Block

TV commercials are always packed and broadcasted together in a so-called commercial block, containing a certain number of individual commercials; four main elements can be identified in these blocks, depicted in Figure 2.1:

- An initial separator which, in Portugal, should contain the word “Publicidade” (cf. Figure 2.2 -Left)).
• A set of individual commercials, corresponding to institutional commercials or to ordinary commercials.
• Broadcaster self-promotion (cf. Figure 2.2-Center).
• An end separator (cf. Figure 2.2 -Right).

Concerning their duration, commercials are typically produced in multiples of five seconds as observed in [4], and nearly 100% of them have a duration between 5 to 60 seconds.

2.3.1 Intrinsic characteristics

The commercials intrinsic characteristics are specific proprieties of the TV commercials content, that are used by commercials producers in order to attract the viewer´s attention:

• **High scene cut rates** – because of their shorter duration in comparison with regular programming, one of the tricks used by advertisers to catch the viewer´s attention is the use of multiple video content transitions; these transitions are implemented through hard cuts and soft cuts, as fades and dissolves.

• **Text presence** – a short and clear way to provide information about a certain product is to include text on the commercial content; this text can be placed in any position of a frame, and this position may change (and typically does) along the commercial.

• **Audio level** – to allow a higher differentiation between a commercial block and a regular programming, the audio level is often increased during the commercial; however, this is now illegal in Portugal and in some other countries. In Portugal, *Entidade Reguladora para a Comunicação Social* (ERC) uses a solution [7] to assess audio level infractions based on ITU-R recommendation BS.1770-2 [8].
• **Audio jingles** – most commercials have background music, and some major brands (e.g., McDonald’s) have recurrent audio jingles.

### 2.3.2 Extrinsic characteristics

The extrinsic characteristics of a commercial are not related to the commercial content, being introduced in a post-production phase:

- **Commercial block separator** – as mentioned before, this is a specific video content that may exist at the beginning and end of a commercial block and serves to separate the aforementioned commercial block from regular programming. This characteristic can be used to help identify the beginning and end of a commercial block.

- **TV logo absence** – in several countries (Portugal included) the TV channel logo should be removed during commercials. Accordingly, the TV channel logo absence can be used as an important hint for commercials detection.

- **Time duration** – usually, a TV commercial has a duration that is a multiple of five seconds, typically between 5 to 60 seconds. This, along with the maximum duration of commercials during an hour of broadcast, imposed by EU regulation, can be used by the commercial detection system.

- **Commercials repetition** – a commercial is usually repeated several times during a day, and sometimes it may be also repeated during the same commercial block; this characteristic has been exploited by the so-called “repetition based” commercials detection schemes.

- **Black frames** – in the past, monochromatic dark frames were commonly used between individual commercials; nowadays, the use of these black frames has been abandoned in several countries.

### 2.4 Methodologies for Commercials Detection

The TV commercials detection schemes available on the literature can be classified into two major groups: the knowledge-based approaches and the repetition-based approaches. Some relevant schemes of each group are overviewed in the next sections.

#### 2.4.1 Knowledge-based detection

Knowledge-based solutions for commercial detection exploit the presence of intrinsic and/or extrinsic characteristics.
A. Black frames and silence

One of the most exploited characteristics on early solutions for commercials detection was the existence of delimiting black frames and silence periods.

In [9], the author proposes a simple method to access the average intensity value of a frame, in order to detect black frames in a video stream; a frame is considered as “black” whenever its average intensity is below a predetermined threshold value. The author concluded that 99.98% of the black frame sequences identified as potentially belonging to commercial blocks were indeed part of a commercial block.

Later, Sadlier et al. [10] improved the black frames detection and added the detection of silent audio frames, by examining the average audio volume level of the audio track. With the information from the detection of the black frames combined with the detection of the audio silence, the chances of successfully detecting the beginning and ends of commercial blocks were improved.

B. High video cuts rate

An intrinsic characteristic worth to be exploited for commercials detection is the video cuts rate, since this is typically much higher in commercials than on regular programming; in fact, as the producers of the commercial have limited time to engage the audience, they tend to implement very fast cutting and jumping from different video shots, in a small amount of time. This characteristic was exploited by a commercial detection solution proposed in [9].

C. TV logo detection

In those countries where the TV channel logo is mandatorily removed during commercials, its absence can be used as an important hint in commercials detection. Of course, this requires the development of accurate logo detection solutions, able to differentiate the TV logo channel from other types of logos (e.g., brand logos that are nowadays sent with some commercials). The method proposed in [11] is one of the first solutions exploiting this characteristic.

D. Motion analysis

A high action level in a shot, which can be measured by the motion inside the shot, together with a high cuts rate, can provide a way to distinguish commercials from regular programming, which tend to have lower levels of temporal activity. This idea forms the basis of the method proposed by Z. Feng et al. in [12].

E. Audio analysis

The analysis of background music can be useful for detecting commercials both on radio and on TV broadcasting, by searching for sudden silences that are common between regular programs and commercial blocks, or even between different commercials of the same commercial block.
In [13], an Audio Scene Change Indicator (ASCI) is proposed to help the detection of the commercials boundaries. An audio scene is often modeled as a miscellaneous grouping of audio sources, some more dominant than others; an audio change is said to have happened when the more dominant sources change.

F. Text Detection

As TV commercials have a need to relay as much information to the viewer as possible, in a small-time interval, they tend to have more text on the screen than regular programming.

In [14], the authors propose an algorithm based on both shot change and text detection. The text detection component works essentially to detect when the trademark of a certain product appears on screen (see Figure 2.3); it computes the gradient of the luminance component, relying on the fact that text, even on a highly textured background, has both large positive and negative gradients in a local region, due to the equally distributed character strokes.

The results of the text detection component of the algorithm show a 96.44% precision, for a total of 100 color video frames with spread text. When coupled with the shot change detection, an improvement in the detection results was observed, in contrast to applying the shot change detection alone.

![Figure 2.3 – Illustration of text detection results [14].](image)

2.4.2 Repetition-based detection

As stated in section 2.3.2, the same TV commercial may be broadcasted multiple times during a day. Accordingly, some detection systems have been proposed that exploit this characteristic.

In [9], the authors established the concept of a commercial fingerprint, with the following requirements:

- The ability to make fine distinctions between similar TV commercials.
- Tolerance to small differences, between two fingerprints, computed from the same commercial but broadcasted at distinct times.
• The ability to quickly calculate and rely on only a few values, so that the computation does not take too much time.

Once the fingerprint of a video segmented is extracted, a matching algorithm is applied between it and an already computed (and stored) fingerprint.

Although of more generic use, repetition-based solutions tend to be more computationally expensive than knowledge-based ones.
3. TV Logo Detection Methods Overview

3.1 Introduction

As mentioned in Chapter 2, in several countries the TV channel logo cannot be present during commercials; accordingly, several methods proposed on the literature for TV commercials detection exploits the presence or absence, in the broadcasted video content, of the channel logo [11][15]. However, nowadays, a TV channel logo has become only one of the many recurrently used graphical elements in the screen known as Digital on-Screen Graphic (DoG). Thus, the methods proposed in [11][15] cannot discriminate the different types of DoGs, preventing an effective TV channel logo detection. In this context, the solution proposed in [3][4], and described in section 3.2, is based on a DoGs detection and classification mechanism, targeting to detect and distinguish TV channel logos from other types of DoGs. Although the assessment results reported in [3][4] are quite good, the method presents clear limitations, some of which have been identified by the author. To get some insight on possible solutions to overcome those limitations, sections 3.3 to 3.5 overviews three different approaches for logo detection, proposed in the last years.

3.2 Automatic Detection of Commercial Blocks Based on TV Channel Logos Detection

Approach

The approach proposed in [3][4] for commercial blocks detection assumes that the TV channel logo disappears during the full duration of the commercial block; accordingly, a DoGs detection and classification mechanism is proposed, targeting to detect and distinguish TV channel logos from other types of DoGs. A DoGs database containing the DoGs acquired along the time is built and continuously updated. A systematic analysis and control of the DoGs database is performed to conclude about the nature of each detected DoG, with the final target of classifying each video segment as Regular Program or Commercial Block.

Technical Solution

Figure 3.1 depicts schematically the architecture of the method proposed in [3][4]; it has four main sections, the first dealing with segmenting a video stream into smaller parts (video shots) and the others three dealing with DoGs detection and classification.
A - Shot change detection and segmentation

The TV broadcaster has to clearly separate TV commercials from other types of programming, due to the previously mentioned EU directive [5]. Therefore, in [5] a simple histogram-based approach was used to detect hard cuts, as it strikes a good balance between computational burden and accuracy. For comparing the frame histograms, the Chi-squared distance was used; a hard-cut is detected if this distance is higher than a threshold. To adjust the threshold to the video content and, consequently, to obtain a better trade-off between false detections and missed detections, an adaptive threshold was adopted.

B - DoG acquisition

This module involves all the algorithmic procedures aiming to detect and characterize the DoGs; it is composed by two main procedures: Video Segment Edges & Color Analysis and DoG Detection.

The Video Segment Edges & Color Analysis procedure consists in the following steps:

1. Example frames extraction – To reduce the computational burden, only a few selected frames (“example frames”, EF) of each video segment are used in the following steps.

2. Edges detection and region of interest definition – The Canny edge detector [16], with predefined-thresholds, is applied to the four corners (considered as the regions of interest) of each EF.

3. Edges fusion – For each corner, the EFs edges maps obtained in each video segment are fused by applying a logical “OR” operation, pixel-by-pixel; the main goal here is to avoid incomplete edge detection, by adding redundancy to the detected edges map.

4. Color Analysis – A dilation (with a 3x3 structuring element) is firstly applied to the previously fused map edges; the variances of the two chrominance components are then computed for every pixel belonging to the dilated edge map, using all the EFs in the
5. **Static pixels detection** – For a pixel, belonging to the dilated edge map, to be considered as *Static*, the variance of both the chrominances components needs to be below a predefined threshold, resulting in a binary map called *static pixels map* (SPM).

The *DoG detection* procedure decides if a static area can be considered as representing a DoG. This function has the following steps:

1. **SPMs interception** – A DoGs characteristic that distinguishes it from any other static area in a video segment (e.g., a static background) is its presence in consecutive video segments. Therefore, in this step, when a new SPM is obtained, it is intersected (i.e., a logical ‘AND’ operation is applied) with other previously obtained SPMs, using a moving window that includes a pre-defined number of video segments. The number of produced static pixels is then evaluated: if higher than a threshold, they are considered for the next analysis; if lower than the threshold, they are discarded unless they have similarities with a previous DoG already stored in the DoGs database (DDB).

2. **DoGs presence verification** – This step aims to assure that no frame corners with a DoG are undetected, provided they have a sufficient number of static pixels that are worth being analyzed. This is done by comparing the potential DoG with those already in the database, to assess if a similarity is found. Running through all the DoGs in the DDB, a comparison between edges, and color values can result in: *i)* detection of an already existing DoG; *ii)* creation of a new DoG on the DDB; *iii)* discarding of the potential DoG under analysis.

**C - DoGs database updating & DoG type decision**

The DoGs database (DDB) is a structure designed with the purpose of storing the necessary information about the DoGs detected over time; it is a central element in the algorithm because the data it contains is crucial to decide the final classification of each video segment.

The detected DoGs can belong to one of three categories: *TV channel logo*, *non-TV channel logo*, and *undefined*. The category is decided mainly based on time persistence: a high time persistence implies that the DoG should be classified as *TV channel logo*; a low time persistence corresponds
to a non-TV channel logo; the undefined category is used as a temporary classification that can be changed to any of the other two.

D - Video segment classification

The last part of the algorithm classifies the video segments as Regular Programming, if a DoG from the classes TV channel logo or undefined was detected on the video segment, or as Commercial Block, if no DoG was detected or if all detected DoGs are of the non-TV channel logo type.

To decrease the occurrence of false commercial detections, the algorithm reclassifies as Regular Programming, isolated video segments that were previously detected as a Commercial Block. In fact, there are no commercials with a single video segment.

Performance assessment

For the assessment of the global system, a library of video files was created composed by three video clips, corresponding to the three Portuguese TV channels; each clip contains examples of regular programs and commercials (among these, some with a brand logo on the screen). The total duration of the video library is 1149 seconds. All videos have a spatial resolution of 1920×1080 pixels and a frame rate of 25 frames per second.

Table 3.1 – Global system performance results [4].

<table>
<thead>
<tr>
<th>TV Channel</th>
<th># frames</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIC</td>
<td>6824</td>
<td>97.7</td>
</tr>
<tr>
<td>TVI</td>
<td>4215</td>
<td>93.9</td>
</tr>
<tr>
<td>RTP</td>
<td>17708</td>
<td>97.0</td>
</tr>
</tbody>
</table>

The reported accuracy values in commercials detection (shown in Table 3.1) are all above 93%. Most false negatives and positives come from the incorrect association between the extracted DoG and the DoGs already in the DDB, which the author refers as a point to be improved.

Although the assessment results are quite impressive, the method presents clear limitations, some of which have been identified by the author:

- Since the matching between a detected DoG and the DoGs already in the DDB are based on a direct pixel-by-pixel comparison, results are quite sensitive to the position of the DoG. As an example, even small errors of about 1-2 pixels may cause a loss of accuracy in the results.

- The DoGs matching mechanism needs improvement because it causes false positives and false negatives.
• The DoGs acquisition procedure assumes static (or quasi-static) DoGs; therefore, with dynamic TV logos, the detection will not work properly.

• The algorithm was tested with short video sequences (the largest one has around 12 minutes duration) and considering just three different TV channels (RTP, SIC, and TVI). A better assessment of the proposed solution should be performed with larger datasets, and longer video sequences.

3.3 A TV Logo Detection and Recognition Method Based on SURF Feature and Bag-of-Words Model

A. Approach

In [17], a TV logo detection and recognition method is proposed, based on several key steps, namely: i) TV logo extraction from a video source using time persistence; ii) SURF (Speed up Robust Features) features extraction from the TV logo images; and iii) SVM (Support Vector Machine) based classification that completes the logo recognition procedure.

B. Technical Solution

Figure 3.3 depicts the block diagram of the method proposed in [17], with has two main phases: SVM training and SVM-based logo recognition.

Training Phase

The training phase has, at its input, TV logo templates pre-selected manually. The following steps are then sequentially performed:

1. **SURF Feature Extraction** – The features extraction is performed on the pre-selected TV logo templates, using the SURF feature extraction proposed in [18], resulting in feature descriptors that will be used in the next step.

2. **Construction of a Visual Dictionary** – To construct the visual dictionary, the authors choice was the bag-of-words model, replacing the normally associated concepts in written documents with the ones appropriate to this solution; therefore, a document is replaced by an region of interest (ROI), and words are replaced by the previously mentioned SURF features of the TV channel logos. The visual vocabulary is formed with the previously extracted feature descriptors, and by applying a k-means clustering algorithm to join the features in k groups, each group representing a visual word; the collection of visual words make the visual dictionary.

3. **SVM Training** – For each channel an SVM classifier is trained. Because SVMs were originally designed for binary classification, a one-to-all strategy [19] is deployed, meaning every SVM classifier is trained with two sets of visual words: the first set is
composed by visual words from the pre-selected logo templates of the TV channel; the second set is assembled from visual words extracted from the pre-selected logo templates from all the other channels in the TV logos template library. This strategy results in an SVM classifier that either recognizes the channel from the first set, a positive answer, or it gives a negative answer. This process is repeated for all TV channels, resulting in an SVM classifier for each TV channel.

![Diagram](image)

**Figure 3.3** – Block diagram for the TV logo detection and recognition algorithm proposed in [17].

**Testing Phase**

After the training phase is completed, the algorithm is prepared to recognize the TV channel to which a given broadcasted video belongs to. The testing phase consists of the following steps:

1. **Example Frames Extraction** – In order to reduce the computational time, the number of video frames is decreased by a selection of equidistant example frames (EFs), from the query video sequences.

2. **TV Channel Logo Extraction** – The method assumes that the TV channel logo is located on the top left corner of the video frame (an assumption that cannot be applied to the Portuguese TV channels); therefore, it treats this part of the frame as a region of interest (ROI) for template matching and recognition. Each example frame ROI is then converted to a binary image, based on the Otsu [20] image segmentation method. To remove unwanted elements, the authors rely on the time persistence of
the TV channel logo, that is expected to be higher than for the non-desired elements. The used method computes the average pixel values, using several example frames, the expected value of this average is one, when the pixel belongs to a TV logo area, and lower than one for pixels outside the TV logo area. Finally, to obtain the closest representation of the actual TV channel logo, an AND operation with the video frame is applied, resulting in a close approximation of the TV channel logo, now with color.

3. **SURF Feature Extraction** – The features extraction is performed on the TV channel logo image resulting from the previous step, using the SURF feature extraction proposed in [18]. This step is identical to the one in the training phase.

4. **Classification** – All the trained SVMs, one per channel, receive the extracted features; then, if the detection is successful, only one SVM will give a positive answer, meaning one SVM detects the channel it was trained for. When more than one SVM classifier gives a positive answer then, the detection is considered wrong.

**C. Performance Assessment**

To evaluate the performance of this solution, the authors collected a video library from 12 TV stations, consisting of 20 video sequences per station, each sequence with a duration of 30 seconds. Each video had a spatial resolution of $424 \times 240$ pixels, and the frame rate is 25 Hz. For each TV channel, 10 TV logos templates were extracted from the top left corner of each frame, and manually labeled for use as training samples; three of the TV channels had transparent logos.

In total, 240 sequences were tested, with 229 been correctly recognized by the algorithm, which gives an average recognition rate of 95.4%. Therefore, the proposed solution has a high recognition rate in both transparent and opaque TV channel logos; also, it shows resilience to non-TV channel logo elements that were not eliminated in the TV logo extraction step, which can be attributed to the SVM proprieties. However, the assumption that the TV channel logo is going to be present in the top left corner, only, is not valid in many countries; for instance, for the three main Portuguese TV channels, at certain times the logo changes position from the top left corner to the top right corner; other channels (e.g., Eurosport 1 and Fox) have their logo positioned in the top right corner. Also, and most important, a system that only recognizes the logos for which it has been trained is of very limited use.

**3.4 Automatic Video Logo Detection Based on a Bayesian Classifier and Neural Networks**

A. **Approach**

The logo detection solution proposed in [21] starts with the determination of the TV logo location, based on the time persistence at pixel level, followed by a feature extraction process. A Bayesian
classifier selects some of the extracted features, based on their location. Finally, a pre-trained Neural Network determines if those features are part of a TV logo.

B. Technical Solution

Figure 3.4 depicts the main building blocks of the algorithm proposed in [21], which are detailed below:

1. **Initial Logo Location Computation** – Exploiting TV channel logos time persistence, a rough estimation of logo location is determined by comparing pixel differences (between neighboring frames) with a pre-defined threshold.

2. **Feature Extraction** – The initial logo region located in the previous step is split in 12×12 pixel regions and, for each pixel, its RGB color values are extracted to construct a feature vector (resulting in a total of 12×12×3 elements per feature vector).

3. **Logo-let Location Assessment** – A Bayesian classifier uses the prior knowledge about the location of TV channels logos and the previously described features vector and its location, to determine if each 12×12 pixel-sized region is a logo-let region or a non-logo-let region; a logo-let is considered as a logo fragment.

4. **Logo-let Classification** – The logo-let candidates (those that are inside logo-let regions), are then assessed by a Neural Network (NN) classifier. This NN was trained with a previously built database containing 8 573 logo-let features and 28 831 non-logo-let features, from manually selected video sequences. The NN classifier receives as input the logo-let candidate features vector and decides if the logo-let candidate is, or not, a true logo-let.

5. **Final Logo Detection** – Two final steps are then applied:
   
   a. Merging all the neighboring logo-lets.
   
   b. Removing all isolated logo-lets.

   If after removing the isolated logo-lets a collection of logo-lets remains, the TV channel logo is considered detected.

C. Performance Assessment

To test the proposed method, a library of 236 frames, taken from 23 video sequences (each one from a different TV channel) was built. In these 236 frames, 226 frames have TV logos. As can be seen in Table 3.2, the detection rate is above 82%; the authors emphasize the robustness of their detection system, mainly because some of the considered logos are transparent.
Table 3.2 – Video logo detection results [21].

<table>
<thead>
<tr>
<th>Total logos</th>
<th>226</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detected logos</td>
<td>187</td>
</tr>
<tr>
<td>Missed logos</td>
<td>39</td>
</tr>
<tr>
<td>False alarms</td>
<td>36</td>
</tr>
<tr>
<td>Detection rate</td>
<td>82.7%</td>
</tr>
</tbody>
</table>

3.5 Spatial HOG Based TV Logo Detection in a Single Frame

A. Approach

In [22], a TV logo detection system is proposed that achieves its goal with only a single frame. This is achieved with a prior phase in which the system is trained with features, extracted from a training library, that include both TV logo positioning and texture information; these features are then stored in a database. The detection phase consists in comparing the features extracted from an query video frame, with the ones previously stored in the database.

B. Technical Solution

Figure 3.5 depicts the main building blocks of the logo detection system proposed in [22], and where two global procedures can be identified: training phase and detection phase.
Training phase

The training phase consists of the following steps:

1. **TV Logo Template Generation** – This step assumes that the TV logo has a few characteristics that are usually stable - as position, size, brightness, and pattern - in contrast with the rest of the video frame. Thus, TV logos can be extracted from video sequences of the training library, by looking for unchanged video brightness. The method used to obtain a threshold below which the brightness variations are considered as not meaningful is the Otsu’s method [20]. The result of this step is a TV logo template.

2. **Key Point Detection** – The algorithm starts with the corners of the TV logos, using the Harris corner detector [23], resulting in key points; a threshold is set to limit the maximum number of key points.

3. **Features Extraction** – The histogram of oriented gradients (HOG) [24] is extracted at key points, one feature per key point. The extracted HOG features contain only information about the texture information; therefore, a novel concept is introduced - the spatial HOG (SHOG) - which combines the HOG feature, with positioning information. The positioning information consists in the spatial relationships between each detected key point and the center of the logo.

4. **Logo Library Creation** – All of the previously extracted SHOGs are grouped according to the corresponding TV logo, building a library for the detection phase.

Detection phase

The detection phase consists on the following steps:

1. **Region of Interest definition** – The four corners of the frame are selected as regions of interest (ROI), to guarantee the capture of the TV channel logo.

2. **Key Points Detector and Feature Extraction** – for each ROI, the HOG features are extracted and transformed into SHOGs.

3. **Features Matching** – The SHOGs acquired in the previous step are matched with the SHOG features stored in the logo library, with the goal of determining if a ROI has a target TV logo.

4. **Voting Scheme** – In order to obtain a final classification, the previously mentioned features matching is performed for all the TV channels in the library; the logo with the highest matching is chosen to label the query frame.
C. Performance Assessment

In this paper, the goal is the recognition of the correct TV channel logo: when a logo is detected but it is wrongly labeled, then it is considered as a negative event. The training library consists of seven videos, one per TV channel.

The authors separated the performance assessment of the method into two phases:

- In the first phase, a testing library consisting on about 2000 frames, from seven TV stations, each one with ten videos, was used. The results show a precision, in all channels, above 98% and a recall above 69%.

- In the second phase, the testing library was expanded to 1,681,106 frames obtained from 18 TV stations. The results show an false positive rate (FPR) of 0.09% and an false negative rate (FNR) of 3.7%.

Although the results are interesting, the method only works with TV logos for which it has been trained for, limiting its applications.
4. Proposed Solution for TV Commercials Detection

4.1 Introduction

In this chapter, the proposed solution for TV commercials detection is described. It can be considered as an evolution of the technique proposed in [3], by the integration of machine learning techniques – namely, convolutional neural networks - seeking to circumvent some of the drawbacks reported in [3]. The chapter starts with the characterization of the relevant TV data; the global architecture of the proposed solution is then described; finally, an in-depth section by section description of the various key steps present in the architecture, is presented.

4.2 TV Data Characterization

The proposed solution was designed to classify the TV content in two categories:

- **Regular Programming** – This class encompasses all the TV content where the TV channel logo is present.

- **Non-Regular Programming** – This class encompasses TV commercials, broadcaster-self-promotion, beginning and end separators of a commercial block, and where the TV channel logo is not present; by EU directive [5], it cannot represent more than 20% of the broadcasted content, per hour.

To accomplish this classification, the focus is on detecting easily distinguishable elements - known as digital on screen graphics (DoG) - that, typically, are only present in regular programs; some examples of these elements are:

- **TV Channel Logo** – The most important element, since when correctly detected should enable automatic detection of a regular program. However, there are a few exceptions, as in the case of the Portuguese “TVI” channel case which, occasionally, uses its logo on self-promotion segments (shown in Figure 4.1), although placed in a different position and/or with a different scale, comparatively to the logo used in regular programs.

- **Clock** – Some TV channels have the time displayed, often close to the TV logo, as depicted in Figure 4.2, and less often away from the TV logo, as can be seen in Figure 4.3; the presence of this element is usually associated with regular programs.
Figure 4.1 – “TVI” channel, with visible logo during a self-promotion.

Figure 4.2 – “Correio da Manhã TV” channel with some DoG elements: TV logo, Clock, Top and Bottom bar.

- **Sign Language Translator** – Typically present in regular programs (as depicted in Figure 4.3), although not exclusively; in fact, it can also be seen in some commercials such as infomercials.

Figure 4.3 – “RTP 1” channel detail, with the channel website, clock, bottom bar and sign language translator.

- **Bottom Bar** – Usually present in TV news and also morning and afternoon talk shows, with text sliding from right to left, as shown in Figure 4.2 and Figure 4.3.
- **Top Bar** – Very similar to the Bottom Bar, but less frequent; shown in Figure 4.2.
- **Program Specific Logo** – Typically found in soap operas, and morning talk shows.
- **Brand Logos** – Present in commercials, sometimes located in the usual TV channel logo position, and adding an extra challenge to the commercials detection.
- **Website** – Some TV channels send its respective website address during news segments, as can be seen in the left side of Figure 4.3.
In designing the commercials detection system, the main concern was to detect as much as possible the aforementioned elements, while reducing unnecessary information. Another observation that lead to a reduction in the information used through the system is that all these elements are mainly present in the frame corners, even if not completely, as is the case of the top and bottom bars.

4.3 Architecture Walkthrough

The architecture of the proposed solution for TV commercials detection is depicted in Figure 4.4, and a summary of every key procedure is presented afterwards.

![Figure 4.4 – Global Solution flowchart.](image)

The first step is the Shot Change Detection and Video Segmentation (SCD); in fact, all the frames belonging to the same shot should have the same classification, thus segmenting the video in shots may help the classification procedure. This block receives a video, computes the luminance histogram of each video frame and, based on a histogram similarity metric, splits the video in groups of consecutive frames with similar histograms, each group being considered as a shot. This procedure is described in section 4.4.
The Post-Processing block starts by selecting a small number of frames from each shot and, for each selected frame, extracts and stores their four corners, that are considered as the regions of interest (ROI); subsequently, it applies an edge detection step to the ROIs, followed by a set of logical operations between detected edges; this results in a single binary image per shot. This procedure is described in section 4.5.

The Automatic Video Shot CNN-Based Classification receives the binary image produced by the previous step (one image per video shot) and classifies it as regular or non-regular programming, using a trained Convolutional Neural Network (CNN). Besides the class of each shot, the CNN also returns the confidence level associated to the classification. This CNN was previously trained (in the “CNN Training” block) with video shots from a training video library that were manually selected and classified (in the “Manual Video Shot Classification” block). This procedure is described in section 4.6.

The Temporal Coherency-Based Reclassification block evaluates the results of the previous block, seeking to identify the lack of temporal coherency in the classification of consecutive video shots; whenever unlikely classifications are identified, the class is changed. This procedure is described in section 4.7.

A last module, named Automatic Learning, encompasses two procedures. The Classification Consistency Analysis block seeks the identification of new scenarios (e.g., a new TV channel) for which the CNN was not trained for. Whenever a new scenario is detected, a set of representative binary images are then extracted (by the Representative Images Selection block) and used to retrain the CNN. This procedure is described in section 4.8.

4.4 Shot Change Detection and Video Segmentation

The objective of the Shot Change Detection and Video Segmentation (SCD) procedure, depicted in Figure 4.5, is to split the input video in segments with similar spatial content. The SCD was designed for detecting hard cuts only, because this is the shot cut type that occurs in regular to non-regular programming transitions, and vice-versa [9]. Its design follows an approach similar to the one adopted in [3]. However, some changes were made on the conditions to determine the adaptive threshold that triggers a hard-cut detection, in order to guarantee continuity between them (they were discontinues in [3]).

The hard cuts detection is based on the differences in luminance values between frames, inside a temporal window. The luminance component was chosen due to three reasons: i) it is directly available on the color space typically used in TV broadcast, \((YC_bC_r)\); ii) it carries more information than the chrominance components; iii) it is less computationally demanding to work with just one color component, then with the three available components.

If the SCD module fails in the detection of a hard cut, it may result in a video segment containing the two types of programming; accordingly, the SDC was designed to have a very low rate of false negatives.
A. Frame Luminance Histogram Computation
The SCD starts by computing the luminance histogram values of every frame inside a temporal window, which has a pre-determined size (WS). Figure 4.6 depicts, in orange, an example of a window with a WS of five.

![Flowchart](image)

Figure 4.5 – SCD section Flowchart.

![Window Example](image)

Figure 4.6 – A window (orange) with a WS of five.

The luminance histograms of each pair of consecutive frames are then compared using the Chi-Square distance, formally described by (4.1)
where \( k \) is the number of histogram bins, and \( H_m, H_n \) are the luminance histograms of two consecutive frames, \( m \) and \( n \). The histogram distance computations that occur inside a window with a WS of five can be seen in Figure 4.7.

\[
D_{m,n} = \sum_{i=1}^{k} \left( \frac{H_m(i) - H_n(i)}{H_m(i)} \right)^2 
\]

(4.1)

Figure 4.7 – Luminance histogram comparison in an initial temporal window.

B. Adaptive Threshold Computation

When all the histograms distances within a window are computed, an average of histogram distance (\( \mu_{FD} \)) is calculated, using (4.2).

\[
\mu_{FD} = \frac{\sum_{i=1}^{WS-1} D_{i,i+1}}{WS - 1} 
\]

(4.2)

A hard cut is detected whenever the histogram distance, \( D_{m,n} \), is higher than a threshold. Similarly to what was considered in [3], the threshold is adapted to the temporal activity within each window, by changing its value according to the value of \( \mu_{FD} \), namely:

\[
AdapTH = W_{Th} \times \mu_{FD} 
\]

(4.3)

i.e., the higher the temporal activity inside the temporal window, the higher is the threshold. \( W_{Th} \) is given by:

\[
W_{Th} = \max(\frac{\mu_{FD}}{N_F}, 2) 
\]

(4.4)

and is graphically represented in Figure 4.8.

Equation (4.4) shows a downward slope starting at a high threshold, \( H_{Th} \), and reducing at a rate dictated by the normalizing factor, \( N_F \), and by \( \mu_{FD} \). Windows with low \( \mu_{FD} \) are too sensitive to small luminance changes that may happen due, for instance, to compression errors, so \( W_{Th} \) should be higher in these cases, to reduce the number of false hard cuts detections; on the contrary, \( W_{Th} \) should be lower for high values of \( \mu_{FD} \), to avoid false negatives in windows with high temporal activity. It is worth to note that windows with too low values of \( \mu_{FD} \) (in this thesis results, lower than 4000) are immediately considered as not containing a hard cut.
C. Cut Decision

In this step, the decision to split the video into segments is based on a comparison between the previously computed $AdapTH$ and each histogram difference. A hard cut is detected between frames $m$ and $n$, according to:

$$Cut\ Decision(m,n) = \begin{cases} \text{Hard Cut Detected,} & \text{if } D_{m,n} > AdapTH \\ \text{No Cut Detected,} & \text{else} \end{cases}$$

(4.5)

After analyzed, the window moves forward $WS - 1$ frames, and the process described above is repeated.

4.5 Post-Processing

The post-processing goal is to transform a video shot into one single image, a shot image, representative of all frames inside the shot. Figure 4.9 shows a flowchart depicting the main steps of this procedure, which are detailed in the following sections.

4.5.1 Example Frame Extraction

To represent a given segmented shot in its entirety, a small number of frames – example frames, (EF) – are extracted from the shot; this allows the following steps to work with a much smaller number of frames, when comparing with the original number of frames in a given shot, thus reducing the computational cost.

The number of EF ($N_{EF}$) to extract per shot is given by:

$$N_{EF} = \begin{cases} \text{ceiling } (0.1 \times SL), & \text{if } SL < SL_{Th} \\ \text{round}(0.1 \times SL_{Th}), & \text{if } SL \geq SL_{Th} \end{cases}$$

(4.6)

where:

- $SL$ is the shot length, in frames;
- $SL_{Th}$ is a parameter that sets the boundary between what is considered a short and a long shot.
For each shot, the first frame is picked and the remaining $N_{EF} - 1$ frames are chosen as equidistant as possible from each other. This selection process aims to capture the video content variations that may happen inside a shot and to have a higher chance of capturing the desired DoGs elements.

### 4.5.2 Regions of Interest Extraction

After the extraction of the EFs, the regions of interest (ROIs) are extracted, one for each corner of the frame, with size of $H \times V$ pixels, were:

$$H = \frac{\text{frame width}}{\text{ScaleX}}, \quad V = \frac{\text{frame height}}{\text{ScaleY}}$$

(4.7)
An example of an original frame is presented in Figure 4.10; Figure 4.11 presents its four ROIs stitched together. The values of \( \text{ScaleX} \) and \( \text{ScaleY} \), five and four respectively, were chosen based on the size of TV channel logos from various TV stations, extracted for the training library. The chosen values were validated when assembling a more diverse library for testing purposes and all, but two, TV channel logos were fully extracted. The exceptions were “Sport TV+” channel logo, with a minor cut, and “Nickelodeon” channel logo, which is almost cut at the middle, although remaining easily distinguishable from all the other TV channels logos. Examples of the two exceptions are depicted in Figure 4.12.

![Figure 4.10 – One video frame from the SIC channel.](image1)

![Figure 4.11 – The four ROIs stitched together, after being extracted with \( \text{ScaleX} = 5 \) and \( \text{ScaleY} = 4 \).](image2)

![Figure 4.12 – Left: “Sport TV+” channel logo and Right: “Nickelodeon” channel logo. Top: original channel logo; Middle: upper right corner ROI detail; Bottom: ROI detail after edge detection.](image3)
4.5.3 Edge Detection

The reason to deploy edge detection, therefore reducing the information given to the CNN classifier, comes from the necessity to select only relevant information for the classifier to be trained and tested with. The reduction in the amount of information given to the CNN results in less time to train, and also helps the CNN to learn only desirable features from the TV channels, e.g. the TV channel logo which is preserved after edge detection.

After the ROIs are obtained, an edge detection is performed, with the goal of retaining only the information needed for classifying the original shot. The edge detection is performed on each ROI separately, to avoid creating false edges from stitching together different frame contents; this also allows adaptive thresholds (for the edge/non-edge decision) to be computed for each ROI.

Since the classifier will support its decisions based solely on the extracted edges, the method used to perform the edge detection is of crucial relevance. A failed detection of an important element, like the TV channel logo, will probably originate a wrong classification.

The first tested edge detection method was the Canny edge detector [16], with an adaptive threshold computed using the Otsu method [20]. The Canny edge detector involves the following steps (applied to the luminance component of the image):

1. Noise reduction performed using a Gaussian filter to smooth the ROI.
2. Gradient magnitude computation.
3. Hysteresis-based edge decision using two thresholds, high\(TH\) and low\(TH\), given by the Otsu method:
   a. If the pixel gradient is higher that high\(TH\), the pixel is considered as an edge.
   b. If the pixel gradient is lower than low\(TH\), the pixel is considered as a no-edge.
   c. If the pixel gradient is between the two thresholds, it is considered as an edge only if is 8-connected to an edge pixel.

Figure 4.13 shows, on the left, the ROIs of an original frame and, on the right, the resulting edges.

![Figure 4.13 – “RTP1” channel ROIs stitched together. Left: frame content; Right: edges extracted with the Canny edge detector.](image_url)
A second edge detection method, the Line Segment Detector (LSD) [25], was also tested, seeking better results for low contrast between TV logo and image background, as shown in Figure 4.14. The main advantage of the LSD detector is that this method does not need the tuning of any parameter.

Figure 4.14 – A frame of the “RTP 1” channel, with a TV logo with low contrast with the background.

The LSD method consists on the following steps:

1. **Image Scaling** – The image is scaled by 80% in each axis, to reduce any aliasing and/or quantization artifacts in the original image.

2. **Gradient Computation** – The image gradient is computed on every pixel with a $2 \times 2$ mask; the level line angle is then obtained - and example is presented in Figure 4.15.

3. **Gradient Pseudo-Ordering** – The previously computed gradients are ordered into a list according to their gradient magnitude.

4. **Gradient Thresholding** – Because small gradient magnitude corresponds to slow gradients or flat zones, those pixels are discarded.

5. **Region Growing** – Starting with a pixel (seed) from the previously ordered list, a region growing algorithm is used to build a **line-support region** (a set of pixels). The pixels that are inside the region are assessed by the similarity between their level line angle and the **line-support region** angle; the most similar ones are selected to be part of a **line-support region**. The **line-support region** angle begins as the angle of the seed pixel; it is then updated as an average of all the angles from all the pixels that are selected within the **line-support region**. A simplified example of a **line-support region** definition process is shown in Figure 4.16.
6. **Rectangular Approximation** – In this step, a rectangle is matched to the line-support region. The resulting rectangle still needs to be validated.

7. **Rectangle Validation** – To validate a rectangle as an edge, the pixels inside the rectangle need to have a similar level line angle with the rectangle; if the pixels that do not meet this criterion are below a certain proportion of the pixels that meet the criterion, the rectangle is validated as an edge.

8. **Aligned Points Density** – This step seeks to correct a possible mistake that comes from the tolerances used in the prior steps; an example of a region growing error can be observed in Figure 4.17. In this particular case, the line-support region should be cut into two smaller and thinner ones. This is achieved by reducing the angle tolerances and removing further away pixels, only when a certain selected pixel density threshold is achieved this step is completed.

![Figure 4.17 – Region growing error [25].](image)

9. **Rectangle Improvement** – The rectangles that are not considered valid in the Rectangle Validation step, go through another equivalent process where, by changing some parameters and tolerances, some of them can be considered as valid.

To compare both edge detectors in low contrast scenarios, the original video frame depicted in Figure 4.14 was used. Figure 4.18 shows, on the left side, the edges resulting from the Canny edge detector, where it is visible:

- The sign language translator, with a well-defined enclosing box.
- The news title “NA TAÇA”, although the surrounding box was not fully detected.
- A very detailed texture (corresponding to the tree), which is an unwanted element.

The TV channel logo, the most important element, is not visible at all; the same happens with the Teletext number and its surrounding box.

Figure 4.18 shows, on the right side, the edges resulting from the LSD detector; in this case, besides being visible all the elements detected with the Canny detector, the "RTP1" logo, as well as the accompanying Teletext code and surrounding box, are also detected.
In this case, the most successful edges detector is the LSD method. However, it also produces unnatural edges in curved forms - the sign language translator edges are an example of this behavior.

### 4.5.4 Edge Fusion

The objective of using edge fusion is to work around the cases where the TV logo, and other relevant DOGs elements, have low contrast with the background, as in the case shown in Figure 4.14. In those situations, the edge detection may fail to catch some parts of the DOGs. By applying the OR between edges of the same shot, it is expected to obtain a resulting edges map containing sufficient edges to define the DOG.

After applying the edge detector to the four ROIs of an example frame (EF), the four resulting binary images are stitched together, as shown in the example of Figure 4.18. To obtain a single image that represents the shot in its entirety, a OR logical operation is performed between the binary images resulting from every EF in a given shot.

In Figure 4.19 a simplified example of edge fusion is depicted - the last frame and respective edges (the fourth from the left, with highly detailed background on the logo area) results in a unrecognizable TV channel logo; by applying the OR logical operation with the other EFs from the shot, the TV channel logo is fully “reconstructed”.

There are also disadvantages when using edge fusion, as the generation of unwanted “fused” edges in the final image. Another disadvantage is shown in the example of Figure 4.20 where, inside a shot, the text changes within the shot, creating a “trailing” effect, which may confuse the CNN classifier.
Figure 4.19 – Eurosport 1 channel simplified edged fusion example **Top:** four stitched ROIs with their respective canny edge detection results; **Bottom:** high detail edge fusion of the frames above.

Figure 4.20 – Trailing effect example. **Top:** two example frames (initial and final) of the same shot; **Bottom:** resulting binary image after the OR logical operation between edges images.

### 4.5.5 Static Area Detection – Edges Intersection

After the edges fusion step, a preliminary shot image that represents the relevant edges of the shot ROI areas is obtained. This image occasionally contains too much details, that do not result only from the (DOGs) desired elements, which have a destabilizing effect on the classifier, leading to wrong classifications. To reduce these details, an AND logical operation is applied between consecutive binary shot images.
Figure 4.21 shows an example where four preliminary shot images are intersected between each other resulting in a cleaner final shot image, where the main desired DoG element are maintained and the undesired elements are removed.

This approach also has disadvantages, as shown in Figure 4.22 - by applying the edge intersection in transitions between non-regular and regular programming, or vice-versa, the TV channel logo has parts of it that disappear because it was not present in the last, non-regular shot image.

The AND operation is applied inside a sliding window with $N_{SW}$ binary shot images - each new shot image is intercepted with the previous $N_{SW} - 1$ shot images, as represented in Figure 4.23. When a new shot is processed, the window advances one position (as depicted in Figure 4.24) and the process is repeated.
Figure 4.22 – “Discovery Channel” simplified edge intersection example. **Top:** four shot images, the first three from regular programming and the last from non-regular programming; **Bottom:** edge intersection of the frames above.

Figure 4.23 – Edge intersection sliding window with a $N_{SW}$ of five, to obtain SI 5.

Figure 4.24 – Edge intersection to obtain SI 6.

The final binary shot image resulting at the end of the post-processing chain, should represent the entirety of the corresponding shot in one image, with most unwanted elements removed and the wanted elements easily recognizable. If the final shot image meets this goal, the learning and testing phases of the classifier will show a high performance.
4.6 Automatic Video Shot CNN-Based Classification

The Automatic Video Shot CNN-Based Classification module, depicted in Figure 4.25, encompasses every task that involves the CNN classifier (Appendix A presents a brief overview of CNNs). It contains two distinct phases: training phase and classification phase. On both phases the videos go through the SCD and Post-Processing modules, to maintain the consistency between the training and testing data.

For the training phase, a library of videos from 12 TV stations, among the most watched in Portugal, was assembled, amounting to 105 hours of video (this library is described in Appendix B). Every video was processed by the SCD and Post-Processing sections, resulting in groups of final shot images; each group is composed by the binary shot images from the same video, and was stored in a separate folder. The shot images were then manually labeled in two classes - regular and non-regular programming. With the training material finalized, the CNN model was trained for the two classifications. Upon the completion of the training phase, a trained CNN model is obtained, and the classification phase can start.

The classification phase consists in using the trained CNN to classify final shot images, taken from the testing library; this library amounts to 38 hours of video, covering 21 different TV channels: the 12 used for training and 9 additional ones (see Appendix B). When a query shot image is given to the CNN, it classifies the shot as regular or non-regular program, and gives the associated confidence level, in percentage.

Figure 4.25 – Automatic CNN-based video shot classification flowchart.
4.6.1 Convolutional Neural Network – Inception Architecture

When developing a system that requires a CNN based classifier, one of two options can be taken: either to design a CNN from scratch or to use a CNN that has most of the layers already pre-trained. In this dissertation, it was decided to use a pre-trained CNN, namely the Inception model-v3 [26][27], developed by Google for image recognition tasks. The main points justifying this option were:

- **Limited Hardware** – The Inception model was trained on 50 Nvidia Kepler GPU [26], but the hardware available in this work was just one mid-tier GPU (GTX 1060 3GB).

- **Sample Size** – The number of video samples collected during this dissertation are enough for retraining the CNN, but not for training the CNN from the scratch; for this, a much higher number of samples would be necessary. As an example, in the ILSVRC-2012 database [28], which was used for the Inception initial training, there were more than one million images.

- **Classification Accuracy** – It was the winner of ILSVRC challenge in 2014 [29], a multi-class classification challenge.

When using a pre-trained CNN, only the weights and connection strengths of the last layer are changed. This approach is detailed in Donahue et al [30].

Finally, the Inception architecture is known for greatly reducing the number of parameters compared to other CNNs. As an example, the winner of ILSVRC 2012, AlexNet [31], employed 60 million parameters [26], while the Inception architecture, used by GoogLeNet [32], employed five million, which results in a much lower computational cost.

In the case of this thesis, the CNN retraining requires the specification of two parameters, namely: the learning rate, and the training steps. The best values for these parameters were experimentally established, as will be described in section 5.5 The code used for retraining the CNN classifier is available at [33], provided by Google.

At this point, it is important to highly that some of the decisions taken in the previous modules (SCD and Post-Processing) seek to lower the computation time during the training stage, by reducing the number of information that the classifier receives:

- **Binary images** – The use of binary images limits the number of permutations in the CNN during the training phase.

- **ROIs** – Reducing the number of pixels that the input layer receives, reduces the computational cost; also, because the maximum resolution image the Inception architecture can receive is $299 \times 299$ pixels, it reduces the magnitude of the Resize operation needed, lowering the risk of losing important elements.

- **Final Shot image** – The classifier receives only one image per shot, reducing the computation cost when training/classifying a shot.
4.6.2 Manual Video Shot Classification

In the training phase, the CNN classifier receives two sets of manually selected final shot images, representing the regular and non-regular classes.

During the manual labeling, some shot images were quite challenging to classify. When it is not obvious the classification to give to a certain shot image, the following considerations were applied:

- Even without a recognizable TV Channel Logo, the shot image is used (and labeled as regular program) if other DoGs are visible, such as: the clock below the TV channel logo, the Teletext code that normally resides next to the TV logo or the sign language translator. An example of a final shot image with these characteristics is depicted in Figure 4.26.

![Figure 4.26](image)

Figure 4.26 – A final shot image from the “RTP1” channel, labeled as regular shot, with Teletext code, clock, and sign language translator.

- If the shot image is completely blank it should not be used. These images are typically originated from non-regular programs, and often used as a separation between each individual commercial of a commercial block.

- If the TV Channel Logo is present but some unwanted elements make the resulting TV Channel Logo unrecognizable, then the shot image is discarded. An example of this case is shown in Figure 4.27.
4.6.3 Automatic Video Shot Classification – Classification Phase

In this stage, the trained model is used to classify shot images that are obtained from the test videos library. The process consists on classifying one final shot image at a time; for each query shot image, the CNN produces two outputs:

- **Class** – The CNN attributes a classification to the query image: regular or non-regular programming.

- **Confidence level** – The confidence level associated to the classification, in percentage; the higher this value, the more likely is that the classification is correct.

To better understand which conditions the confidence level, a shot image was manually manipulated in order that only some of the (considered as) relevant DOGs are present. The initial shot image (before manipulation), obtained from the testing library, is presented in Figure 4.28 - using the trained CNN, this image was classified as *regular programming*, with a confidence level of 99.95%.
Some elements were then removed, to access its influence on the classification. In Figure 4.29, three examples of a single element removal are presented.

![Figure 4.29](image-url) **Left:** “RTP” symbol removed; **Center:** TV logo number removed; **Right:** Teletext number removed.

The classification confidence level results, for the manipulated shot image are presented in Table 4.1.

<table>
<thead>
<tr>
<th>Element(s) Removed</th>
<th>Confidence Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel Number</td>
<td>99.8%</td>
</tr>
<tr>
<td>Teletext Box</td>
<td>99.6%</td>
</tr>
<tr>
<td>Sign Language Translator</td>
<td>99.1%</td>
</tr>
<tr>
<td>Clock</td>
<td>99.0%</td>
</tr>
<tr>
<td>Channel Symbol</td>
<td>98.9%</td>
</tr>
<tr>
<td>Channel Logo (Symbol+Number)</td>
<td>98.5%</td>
</tr>
<tr>
<td>Website</td>
<td>97.4%</td>
</tr>
<tr>
<td>Channel Logo + Teletext Box</td>
<td>94.6%</td>
</tr>
</tbody>
</table>

The results presented in Table 4.1 show that the CNN classifier does not rely on only one DOG element - even in the case where all the elements present in “RTP1” logo are removed, the confidence level is still 94.6%; the classifier uses the different desired elements, making it a potential robust solution, even when some relevant DOG elements are not correctly extracted.

Another analysis was conducted taken Figure 4.28 as a starting point and progressively removing elements, until the classification was close to change from regular to non-regular. Figure 4.30 depicts the progressive elements removal.

The results of this analysis are presented in Table 4.2. After removing the “RTP.PT” website, just a few edges remain, and the shot is classified as regular but with a confidence level of 52% (which is close to a random classification). This is expected, because in this case even a human could not determine the class of the resulting image.
A final scenario where the TV channel logo was moved from the original top left corner to the top right corner resulted in a regular classification with a confidence level of 69.5%, meaning that the classifier is potential robust to a substantial shift in the DOGs position.

Table 4.2 – Cumulative elements removal impact on the confidence level.

<table>
<thead>
<tr>
<th>Cumulative Elements Removal</th>
<th>Confidence Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel Logo + Teletext Box</td>
<td>94.6%</td>
</tr>
<tr>
<td>Clock</td>
<td>82.7%</td>
</tr>
<tr>
<td>Sign Language Translator</td>
<td>74.4%</td>
</tr>
<tr>
<td>Website</td>
<td>52.3%</td>
</tr>
</tbody>
</table>

Finally, it is worth to note that since the CNN classifies one shot image at a time, it does not consider in its decision the temporal correlation that exists between shot images belonging to the same program or to the same commercial block. An additional step was then thought and implemented, seeking to overcome this limitation and correct misclassifications, which is described in the next section.
4.7 Temporal Coherency-Based Reclassification

The need for evaluating and changing the classification produced by the CNN classifier arises from various shortcomings, already referred in previous sections:

- **Lack of Temporal Coherency** – The CNN classifier does not take into account the temporal coherency, meaning that it classifies a query shot image without taking into consideration the classification of previous (and neighbor) shots; this may lead to erratic classifications during a certain time interval, where the class of consecutive shots presents frequent changes.

- **Disregard of the Typical Programming Structure** – Regular programming is much more frequent than non-regular programming, with a ratio close to 4:1 [5], respectively. Also, shots of the non-regular type tend to be concentrated in time, in the so-called Commercial Blocks. The CNN does not have this knowledge.

- **Failed Edge Detection** – The edge detection occasionally fails to capture any element of interest; the CNN classifier, alone, cannot compensate for it, resulting in a misclassification.

These faults were taken into consideration for developing the reclassification strategy, with emphasis for the CNN lack of temporal coherency.

The Temporal Coherency-Based Reclassification procedure, depicted in Figure 4.31, receives the classification from the CNN classifier and, by evaluating the temporal coherence between the classes of consecutive shots, reclassifies some of them. This procedure relies in three inputs - the shot class, the classification confidence level and the shot duration - and is applied on a moving temporal window. This window has a predefined length and is composed of a certain amount of consecutive shots, dependent on their total time duration. The window is analyzed to identify if it contains part of a Commercial Block, looking for a potential beginning and an end of a commercial segment; the corresponding boundary shots are designated as Anchor Shots.

Besides the above described approach, another procedure (also considered in [3]) was implemented, that relies on detecting isolated shot images classifications.
4.7.1 Input Parameters

The Reclassification process works with the following input parameters:

- **Type of Programming** – The class that was given by the CNN to a shot image: *regular* or *non-regular* programming.
- **Confidence Level** – Higher confidence level are assumed to correspond to more accurate classifications.
- **Shot Duration** – Experimentally, it was verified that longer shots result in a more accurate classification.

4.7.2 Reclassification

The reclassification process starts by grouping together shots into a temporal window, with a maximum time duration of $TempWindowTH$. The reclassification occurs inside the window, and the process is repeated for every window. There is no overlapping between consecutive windows, to circumvent the possibility of reclassifying the same shot twice.

When a reclassification occurred, not only the class is changed but the confidence level is also modified, according to (4.8).

\[
New \, \text{Confidence Level} = 100\% - \text{Old Level of Confidence} \quad (4.8)
\]
Anchor Shots Based Approach

The algorithm starts by verifying if different classes exist in the same window; if yes, the window is analyzed forward and backward, looking for (respectively) the start and end shots (or anchor shots) of a possible commercial block segment; these anchor shots are detected if the following conditions are met, for both anchor shots:

1. They had to be classified as \textit{non-regular}.
2. Their confidence level is above \textit{AnchorLoConfTH}.
3. Their minimum duration is \textit{AnchorShotDuratTH} seconds.
4. They have nearby classifications that support the original classification, if at least one of the following criteria is met:
   a. 80\% of their neighbor shots are classified as \textit{non-regular}.
   b. 60\% of their neighbor shots are classified as \textit{non-regular} and with a confidence level that should not be more than 15\% lower than the confidence level of the candidate anchor shot (i. e., a shot that verifies the three first conditions).

Let \( i \) represent the position of a candidate anchor shot in the temporal window. If it is the starting anchor shot, its neighborhood is defined by:

\[
\text{Neighborhood}(i) = \begin{cases} 
  i - \text{beforeNshots} \\
  i + \text{afterNshots} 
\end{cases} \quad (4.9)
\]

where, \textit{beforeNshots} and \textit{afterNshots}, represent the number of shots before and after the anchor shot candidate position; in the experimental part, these values were set to three and eight, respectively. For the second anchor shot candidate, the parameters \textit{beforeNshots} and \textit{afterNshots} have their values swapped, to avoid analyzing too many shots outside the temporal window.

If a start and an end anchor shot have been detected, the reclassification takes place inside the temporal window in analysis - all shots between the two anchor shots are classified as \textit{non-regular}; the remaining shots inside the temporal window are classified as regular. This strategy seeks mainly to deal with classification errors in the limits of the commercial blocks, that are more likely due to the shot images intersection (AND operation). An example of the overall procedures is shown in Figure 4.32.

When the algorithm encounters a window were only one anchor shot is found, no action is taken.
Simple Isolation Detection and Correction Approach

A simpler reclassification algorithm was developed to detect and reclassify single isolated shots; these shots have a different classification from four shots in its vicinity, two before and two after, as illustrated in Figure 4.33. When these conditions are met, the isolated shot is reclassified to match the classification of the shots in its vicinity.

In this approach, the algorithm does not alter the classification of a regular shot with a duration higher than twenty seconds; in fact, it is quite unlikely that such a long shot could be part of a commercial.
Final Report

The Temporal Coherency-Based Reclassification produces a final report for each window; if the original classification (i.e., resulting from the CNN) was changed during reclassification, this is recorded and will be used to assist the Automatic Learning procedure, described in the next section.

4.8 Automatic Learning

The CNN classifier was trained on a training library, which is limited in channel diversity and time duration and, although there is a dedicated procedure to correct potential misclassifications resulting from the CNN (described in the previous section), it needs to be adapted to new scenarios for which it was not trained, such as:

- **New TV Channel** – The CNN was originally trained with twelve TV channels. When the system is applied to a new channel, the classification accuracy typically decreases; this is the main scenario for the automatic learning process.

- **Known TV Channel with multiple logos** – Although not frequent, the same TV channel may use different logos, as happens in the case of the "SIC Notícias" channel, depicted in Figure 4.34: the left side presents the logo obtained from the training library; the right side presents a logo obtained from the testing library.

![Figure 4.34](image)

Figure 4.34 – **Left**: "SIC Notícias" normal TV logo and **Right**: alternative logo.

The Automatic Learning algorithm, depicted in Figure 4.35, implements two important tasks: the detection of new scenarios, which is performed by the Classification Consistency Analysis, and the selection of new training images to retrain the CNN, which is performed by the Representative Image Selection.

The Classification Consistency Analysis detects when the system should learn a new scenario; it performs this task based on the frequency of the classification change in consecutive shots. In fact, when this frequency increases, usually the resulting classification errors also increase.

When a new scenario is detected, the system needs to automatically acquire new training material, corresponding to that scenario. It is worth to note that if the CNN is retrained with misclassified shot images, a “vicious cycle” is created, resulting in a CNN that after retraining produces worst results. To acquire suitable training data, the Representative Image Selection uses the classification confidence level and time duration of the new shot images.
4.8.1 Classification Consistency Analysis

As mentioned before, TV commercials tend to be grouped together into commercial blocks. Accordingly, the shot class resulting from the CNN should persist during several consecutive shots. To determine when the system should capture new information (i.e. a new TV channel), the following metrics was developed, that measures the average number of class changes per detected shot:

\[
Average\text{ClassificationChange} = \frac{\sum_{i=1}^{W_{size}-1} ClassificationChange(i)}{N_{shots} - 1}
\]  

(4.10)

where,

\[
ClassificationChange(i) = \begin{cases} 
0, & \text{if ClassOfShot}(i) = ClassOfShot(i + 1) \\
1, & \text{else} 
\end{cases}
\]  

(4.11)

and \(N_{shots}\) is the number of detected shots in the temporal window, with a time duration, \(LearningDecisionTH\).

The decision to enter in the retraining process is taken whenever \(Average\text{ClassificationChange}\) is higher than a predefined threshold, \(ClassChangeTH\). The choice of a value to this threshold is of major importance because it need to be low enough to enable the retraining process to start, while being high enough to avoid unnecessary training of the CNN. Also, it reduces the risk of the
CNN becoming too specialized, meaning it would only classify one channel well, because with infinitely repeated retraining, it would forget features from other TV channels, resulting a less useful solution.

4.8.2 Representative Images Selection

A major difficulty in the Automatic Learning algorithm is the selection of images to retrain - if retraining is necessary, the CNN is certainly wrongly classifying shots, and if those wrongly classified shot images are then selected for retraining, the results will be worst, as depicted in Figure 4.36.

![Figure 4.36 – “Vicious cycle” of misclassified training shots.](image)

To obtain suitable retraining images (final shot images), the following considerations were applied:

- The new training images should have not been reclassified by the Temporal Coherency-Based Reclassification procedure. This is performed because: when a shot image is reclassified its confidence level is altered in a way that makes the new confidence level irrelevant, e.g. a shot image that was classified by the CNN as a regular with a 90% confidence level, after reclassification (by the Temporal Coherency-Based Reclassification procedure) turn into a non-regular with a 90% confidence level.

- The selected retraining images should verify the following criteria:
  - Classification confidence level higher than $AutoLevelTH$.
  - Associated time duration higher than $AutoDurationTH$.

The retraining images are extracted from the window where the detection in the Automatic Learning Detection occurred, after the selection, usually less than 10% of the extracted shot images are selected, from the new video. This occurs because wrongly classified shot images can provoke the positive feedback cycle described previously, and by having strict selection parameters this is minimized.

To minimize the errors in the classification that may occur until the CNN is trained for the new scenarios, and to avoid the CNN forgetting features from the original training library, the retraining
time is much lower than the initial training, and this is achieved by reducing the learning rate, and the number of training steps.

The Automatic Learning Process allows the solution to adapt itself to new scenarios, making it much more useful to a potential user. If this section was not implemented, for each new scenario with subpar classification accuracy the user would need to make a frame-by-frame analysis, of each captured video for every new scenario. The process of creating a training set manually is extremely time-consuming and by avoiding it, the solution has a very useful feature.
5. Performance Assessment

5.1 Introduction

In this chapter, an assessment of the proposed solution for TV commercials detection is conducted, with each main module being individually evaluated to determine if it is performing as expected. For this assessment, a database of video sequences was assembled, which is described in section 5.2; the used assessment metrics are presented in section 5.3.

For the SCD evaluation, described in section 5.4, two analyses were performed: a first one to assess the detection of hard cuts, regardless of its position on the video sequence, and a second one to determine if the hard cuts between regular and non-regular programming, or vice-versa, are successfully detected.

The CNN-Based Classification was firstly evaluated without both the temporal reclassification and automatic learning; this initial assessment is described in section 5.5. Several testing scenarios, consisting of different combinations of the SCD and Post-Processing parameters, are considered. The trained CNN’s are tested with two sub-sets of videos from the testing library: the first sub-set with videos of the same TV channel used for the CNN training; the second set is composed by videos from TV channels not used for training.

The Temporal Coherency-Based Classification is the module that creates the final report; its assessment is presented in section 5.6, including a comparison with the initial classification given by the CNN.

Finally, the Automatic Learning module is evaluated in section 5.7.

All tests were conducted on a computer with a 64-bit operating system, with an Intel i7-4790 @ 3.60 GHz with 16 GB of RAM and a GTX 1060 3GB video card.

5.2 Video Libraries Database

During the development of this Master Thesis, three video libraries were used: the first one, and also the smallest one, was taken from the videos dataset assembled in [3] and is used exclusively in the first assessment of the SCD (SCD Analysis A). The other two – referred as the training and testing video libraries - were collected with the purpose of capturing longer and more diverse video sequences, when comparing to the video dataset of [3], and are used for the assessment of all the main modules of the proposed solution.

5.2.1 Video Library for the SCD Analysis A

The video library used for the first assessment of the SCD module is composed by four small videos taken from the video library used in [3], which were provided with accompanying files with
the frame number where a hard cut occurred. The relevant video characteristics are present in Table 5.1.

Table 5.1 – Testing library for the SCD analysis A.

<table>
<thead>
<tr>
<th>Video Sequence</th>
<th>TV Channel</th>
<th># of Frames</th>
<th># of Hard Cuts</th>
<th>Content Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biggs</td>
<td>Panda Biggs</td>
<td>4581</td>
<td>49</td>
<td>Kids shows</td>
</tr>
<tr>
<td>aBolaComPub</td>
<td>Bola TV</td>
<td>12901</td>
<td>71</td>
<td>Talk show</td>
</tr>
<tr>
<td>Rtp1comPub</td>
<td>RTP 1</td>
<td>8787</td>
<td>72</td>
<td>News</td>
</tr>
<tr>
<td>Hw1</td>
<td>Hollywood</td>
<td>3933</td>
<td>55</td>
<td>Movies</td>
</tr>
</tbody>
</table>

5.2.2 Training Video Library

The training library was assembled with the top twelve most viewed TV channels in Portugal [34]. The video sequences were captured within almost two weeks, with a capture time between 9 AM to 11 PM. Ideally, this should result in only one video per channel and with the same time duration; however, the capture card had problems whenever it reached three hours of continuous recording, which results in videos with different time length.

The training library final time duration is, roughly, 105 hours and it is described in Appendix B. There are two main motivations for such a long time duration:

- **Content Diversity** – A longer time enables more diverse TV content, giving the CNN more scenarios to be trained.

- **Generalized Learning** – With a library composed of twelve TV channels, the CNN acquires a more generalized knowledge, avoiding a specialization that would produce worst results when the CNN classifier encounters new scenarios, i.e. new TV channels.

5.2.3 Testing Video Library

The training library can only be used to train the CNN; in fact, it would be of little value to demonstrate a CNN classifying the same videos used in the training phase. Therefore, a new library was assembled for testing, composed of 26 videos from 21 TV channels, with more than 38 hours of video, and captured with more than a month of distance to the training library capture. It is separated into two subsets:

- **Already-Trained** – Videos with the same TV channels contained on the training library.

- **Not-Trained** – Videos from TV channels not present on the training library.

In the not-trained sub-set of videos, there is a peculiar case of a video that although belonging to a channel from a training library, was acquired in a moment where the corresponding TV logo changed its shape, texture, and position. Therefore, this video was categorized as a new scenario.
and was inserted in the not-trained videos. The video is referenced as “SICN_HD__2HBC” and belongs to the “SIC Noticias” channel.

The testing library is detailed in Appendix B.

5.2.4 Automatic Learning Library

To demonstrate the adaptability of the solution, a TV channel showing low classification accuracy (on first tests with the proposed solution) – namely, the “Discovery Channel” - was chosen. In fact, during programs, this channel presents only one DoG type – the channel logo. Additionally, this logo is quite difficult to detect because it is thin and semi-transparent, as can be seen in Figure 5.1. The assembled library has six videos, each one with a time duration of around one hour, as shown in Table 5.2.

![Figure 5.1 – Discovery Channel TV logo detail.](image)

Table 5.2 – Automatic learning library.

<table>
<thead>
<tr>
<th>Video Sequence</th>
<th>Duration (h:m:s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery_1H_DA</td>
<td>1:12:12</td>
</tr>
<tr>
<td>Discovery_1H_DB</td>
<td>1:01:41</td>
</tr>
<tr>
<td>Discovery_1H_DC</td>
<td>1:01:13</td>
</tr>
<tr>
<td>Discovery_1H_DD</td>
<td>1:02:19</td>
</tr>
<tr>
<td>Discovery_1H_DE</td>
<td>1:13:05</td>
</tr>
<tr>
<td>Discovery_1H_DF</td>
<td>1:01:05</td>
</tr>
</tbody>
</table>

5.3 Performance Assessment Metrics

Since the decision processes involved on the proposed solution are binary - namely, cut/no-cut in the SCD module and regular/non-regular in the CNN classifier – the following metrics typically used for the assessment of binary classifiers where chosen:

- **True Positive Rate (TPR, commonly known as Recall)** – the proportion between events classified correctly as positives and the actual number of positive events

\[
TPR = \frac{TP}{TP + FN}
\]  

(5.1)
• **Precision** – the proportion between events classified correctly as positive and all events classified as positive

\[ \text{Prec} = \frac{TP}{TP + FP} \]  

(5.2)

• **F1-Score (F1)** – the harmonic mean between Precision and TPR

\[ F1 = 2 \times \frac{\text{Prec} \times TPR}{\text{Prec} + TPR} \]  

(5.3)

• **Accuracy (Acc)** – the proportion between true results (true positives and true negatives) and the total number of cases

\[ \text{Acc} = \frac{TP + TN}{TP + TN + FP + FN} \]  

(5.4)

### 5.4 Shot Change Detection Assessment

To access the Shot Change Detection (SCD) module, two analysis were conducted:

- In **Analysis A**, a study of the parameters used in [3] is performed, seeking to found the best values to be used in the proposed solution.
- In **Analysis B**, and with the SCD parameters already established (form Analysis A), the detections of the transitions from regular to non-regular programming, and vice-versa, were assessed.

#### 5.4.1 SCD Analysis A

For analysis A, and as mentioned before, a video library taken from [3] was used. The assessment produces a classification based on the following definitions (in these definitions, a frame is considered as a hard cut if it is the first one following a hard cut):

- **True Positive (TP)** – a video frame was accurately classified as a hard cut;
- **True Negative (TN)** – a video frame was accurately classified as not being a hard cut;
- **False Positive (FP)** – a video frame was incorrectly classified as being a hard cut;
- **False Negative (FN)** – a video frame was incorrectly classified as not being a hard cut.

**SCD Parameters**

The priority in developing the SCD was to avoid false negatives, even if at a cost of increasing false positives. In fact, a wrongly segmented shot, with frames that belong to the two types of classification, regular and non-regular, is much more harmful than an unnecessarily cut to the video. An example of the consequence of a false negative that occurred during the development of the SCD can be seen in Figure 5.2 In this example, is possible to see that the TV logo is above...
a soap opera logo; for the CNN classifier this is a worst-case scenario because it is presented with a shot image that has characteristics from two different classifications, which will may lead to a misclassification.

Figure 5.2 – “Correio da Manhã TV” channel example of a preliminary shot image resulting from a false negative.

The SCD parameters used in [3] are presented in Table 5.3. However, because in [3] several false negatives were detected, and these should be minimized, a new study of the used parameters was conducted.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window Size (WS)</td>
<td>14</td>
</tr>
<tr>
<td>Low Threshold</td>
<td>5000</td>
</tr>
<tr>
<td>High Threshold</td>
<td>70000</td>
</tr>
<tr>
<td>Window Threshold ($H_{Th}$)</td>
<td>8</td>
</tr>
<tr>
<td>Numbers of Histogram Bins</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 5.3 – SCD parameters used in [3].

Parameter selection effects:

1. **Window Size (WS)** – The effect of varying the window size on the video sequence “Rtp1ComPub” can be seen in
2. Figure 5.3. To minimize the number of false negatives, without compromise the other metrics, a window size of 18 was chosen.
3. **Low threshold** – A threshold of 5000, also used in [3], showed a good compromise between all metrics, as can be observed in Figure 5.4 for the “aBolaComPub” - accordingly, this value was kept.
4. **High threshold** – This parameter has no effect in the number of false negatives, and as very small downward effect on the number of false positives when increased, as can be observed in Figure 5.5 for “Rtp1ComPub”. On “hw1” and “aBolaComPub” the effect is
non-existent within the studied interval. Therefore, the final chosen value should be higher than 70000.

5. **Number of histogram bins** – This parameter proved to be the most difficult to study because changing the number of bins has effects on basically every metric and not in a linear manner. The most interesting example is the “Biggs” clip, as can be seen in Figure 5.6. The number of histogram bins should be higher than 16 because at this value a reduction into the number of false negatives was observed in most videos.

![Figure 5.3](image)

**Figure 5.3** – “Rtp1comPub” video Recall, Precision and F1 scores when changing the window size.
In this dissertation, the Low Threshold and High Threshold used in [3] are converted in the $N_F$ parameter, according to:

$$N_F = \frac{High\ Threshold - Low\ Threshold}{2} \quad (5.5)$$

From the previous analysis, the final set of parameters were set and are presented in Table 5.4

<table>
<thead>
<tr>
<th>Window Size (WS)</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalizing Factor ($N_F$)</td>
<td>42000</td>
</tr>
<tr>
<td>Window Threshold ($H_{Tr}$)</td>
<td>8</td>
</tr>
<tr>
<td>Numbers of Histogram Bins</td>
<td>32</td>
</tr>
</tbody>
</table>
With these parameter chosen, a new analysis is performed for comparison purposes. In Table 5.5, it is possible to observe a minor increase in true positives, but also in false positives. However, the number of false negatives is the most important metric to evaluate, and this value has decreased.

Table 5.5 – SCD analysis A results.

<table>
<thead>
<tr>
<th>Video Sequence</th>
<th>Biggs</th>
<th>aBolaComPub</th>
<th>Rtp1comPub</th>
<th>Hw1</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Positives</td>
<td>43</td>
<td>46</td>
<td>61</td>
<td>62</td>
</tr>
<tr>
<td>False Positives</td>
<td>3</td>
<td>6</td>
<td>12</td>
<td>25</td>
</tr>
<tr>
<td>False Negatives</td>
<td>6</td>
<td>3</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>Recall</td>
<td>88%</td>
<td>94%</td>
<td>86%</td>
<td>87%</td>
</tr>
<tr>
<td>Precision</td>
<td>94%</td>
<td>89%</td>
<td>84%</td>
<td>73%</td>
</tr>
<tr>
<td>F1</td>
<td>91%</td>
<td>91%</td>
<td>85%</td>
<td>80%</td>
</tr>
</tbody>
</table>

In conclusion, a trade-off was made between increasing the false positives, which will impact the running time of the SCD in a negative way, and decreasing the false negatives, which could affect negatively the final classification.

5.4.2 SCD Analysis B

In the second SCD analysis, the goal was to assess if the SCD module detects the transitions between regular and non-regular programming (and vice-versa), since the objective of the SCD is to provide the classifier with a group of video frames (shot) belonging to the same class. To build the required Ground Truth information, several videos from the testing library were selected and their content was visually analyzed, resulting in an associated file with the frame numbers of the beginning and end of each non-regular video segment. The chosen videos have different types of contents and different number of regular/non-regular transitions, as shown in Table 5.6.

Results

The assessment was performed with 13 videos, which in total almost reach 30 hours of video, and 110 transitions (see Table 5.6.). Every transition between regular and non-regular programming (and vice-versa) was detected; this can be attributed to the EU legislation stating that “Television advertising and teleshopping shall be quite distinct from other parts of the programme by optical and/or acoustic and/or spatial means”[5]; this results that even with a not perfect hard cut detection algorithm, the separation between regular and non-regular programming is always detected.
Table 5.6 – SCD Assessment with videos from the testing library.

<table>
<thead>
<tr>
<th>Video Sequence</th>
<th>Duration (h:m:s)</th>
<th># of Transitions</th>
<th>Content Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disney_2H_CA</td>
<td>2:14:28</td>
<td>11</td>
<td>Kids Shows</td>
</tr>
<tr>
<td>Eurosport1_HD_CA</td>
<td>1:42:39</td>
<td>16</td>
<td>Sports Events</td>
</tr>
<tr>
<td>HW_HD_2H_CA</td>
<td>2:10:27</td>
<td>4</td>
<td>Movies</td>
</tr>
<tr>
<td>RTP1_HD_2H_CA</td>
<td>2:00:51</td>
<td>8</td>
<td>News</td>
</tr>
<tr>
<td>24Kitchen_HD_CA</td>
<td>0:58:18</td>
<td>4</td>
<td>Cooking Shows</td>
</tr>
<tr>
<td>AXN_HD_2H_CA</td>
<td>02:37:41</td>
<td>13</td>
<td>TV Series</td>
</tr>
<tr>
<td>CMTV_HD_CA</td>
<td>01:17:22</td>
<td>1</td>
<td>News</td>
</tr>
<tr>
<td>Discovery_HD_CA</td>
<td>1:34:29</td>
<td>12</td>
<td>Documentaries</td>
</tr>
<tr>
<td>FOX_HD_2H_CA</td>
<td>2:02:30</td>
<td>10</td>
<td>TV Series</td>
</tr>
<tr>
<td>Nicklodeon_CA</td>
<td>1:03:58</td>
<td>6</td>
<td>Kids Shows</td>
</tr>
<tr>
<td>PortoCanal_HD_CA</td>
<td>1:36:39</td>
<td>2</td>
<td>Talk Shows</td>
</tr>
<tr>
<td>SIC_HD_3H_BA</td>
<td>3:14:35</td>
<td>7</td>
<td>Talk Shows</td>
</tr>
<tr>
<td>SICN_HD_2HBC</td>
<td>1:46:47</td>
<td>2</td>
<td>News</td>
</tr>
<tr>
<td>SICRAD_HD_AB</td>
<td>1:30:44</td>
<td>2</td>
<td>TV Series</td>
</tr>
<tr>
<td>TVI_2H_CA</td>
<td>1:56:57</td>
<td>4</td>
<td>News</td>
</tr>
<tr>
<td>TVI24_BA</td>
<td>1:45:44</td>
<td>8</td>
<td>News</td>
</tr>
</tbody>
</table>

5.5 Initial CNN-Based Classification Assessment

The initial CNN-based classification encompasses the classification process of the shot images using the trained CNN classifies, before applying both the temporal coherency-based reclassification and the automatic learning procedure. Its assessment of produces a classification of the shot images in regular or non-regular shots, and uses the following definitions:

- **True Positive (TP)** – a shot was accurately classified as *regular* programming.
- **True Negative (TN)** – a shot was accurately classified as *non-regular* programming.
- **False Positive (FP)** – a shot was incorrectly classified as *regular* programming.
- **False Negative (FN)** – a shot was incorrectly classified as *non-regular* programming.

The Post-Processing module has different parameters, and it is not possible to evaluate every combination of parameter values. Thus, a set of combinations (or test conditions) were considered – and shown in Table 5.7 – which were selected according to the following reasoning:

- 1 and 2 have Post-Processing parameters similar to the ones used in [3].
- 3 and 4 were used to assess the impact of not applying the edge intersection procedure.
• 5 and 6 were a compromise between the previous testing conditions, with a smaller $N_{SW}$ and a lower number of EFs.

• 7 and 8 assess if the choice of using just the frame edges, instead of the frame luminance or color information, was the correct one. The Post-Processing module for these two testing conditions only performs the selection of a single EF per shot. The selected EF is located in the middle of the shot, followed by the ROIs extraction and stitching. The stitched ROI’s are then delivered to the Automatic Video Shot CNN-Based Classification.

Table 5.7 – Parameters for each test condition.

<table>
<thead>
<tr>
<th>#</th>
<th>Edge Detection Method</th>
<th>The boundary to distinguish between short and long shots ($S_{LTth}$)</th>
<th>Edge Intersection Window Size ($N_{SW}$)</th>
<th>Final Shot Image Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Canny</td>
<td>100</td>
<td>5</td>
<td>Binary</td>
</tr>
<tr>
<td>2</td>
<td>LSD</td>
<td>100</td>
<td>5</td>
<td>Binary</td>
</tr>
<tr>
<td>3</td>
<td>LSD</td>
<td>30</td>
<td>n.a.</td>
<td>Binary</td>
</tr>
<tr>
<td>4</td>
<td>Canny</td>
<td>30</td>
<td>n.a.</td>
<td>Binary</td>
</tr>
<tr>
<td>5</td>
<td>Canny</td>
<td>30</td>
<td>3</td>
<td>Binary</td>
</tr>
<tr>
<td>6</td>
<td>LSD</td>
<td>30</td>
<td>3</td>
<td>Binary</td>
</tr>
<tr>
<td>7</td>
<td>n.a.</td>
<td>No</td>
<td>n.a.</td>
<td>Grayscale</td>
</tr>
<tr>
<td>8</td>
<td>n.a.</td>
<td>No</td>
<td>n.a.</td>
<td>Color</td>
</tr>
</tbody>
</table>

Each test condition implies a new training of the CNN, and a new resulting CNN-based classifier. For every testing conditions the classifier performance was obtained.

To obtain comparable results, in all test conditions the CNN was trained with the same training parameters, depicted in Table 5.8, and with the same training videos. Also, for every training condition, a visual inspection of the training samples was performed, to avoid undesirable samples; this process was extremely time-consuming, since it implied around 140000 images that need to be examined for each training process.

Table 5.8 – CNN training parameters.

<table>
<thead>
<tr>
<th>Training Parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Rate</td>
<td>0.001</td>
</tr>
<tr>
<td># Training Steps</td>
<td>3 456 000</td>
</tr>
</tbody>
</table>

The chosen learning rate is close to the value recommended in [33]. Concerning the training time, it was experimentally verified that less than three days of training did not provide sufficiently accurate classification results, and more than three days did not bring significant improvements to those results. Accordingly, the number of training steps was calculated such that the training procedure, for each test conditions, would not take longer than three days. Since the measured
(with the computer used in this thesis) number of steps per hour was 48000, for three days it resulted in 3 456 000 training steps.

The metric of most importance to this thesis is the classification accuracy. Therefore, for each testing condition the average classification accuracy was computed, considering all videos in the two sub-sets (i.e., already-trained and not-trained) of the testing library; the best accuracy value and the worst accuracy value were also considered. Table 5.9 presents the results.

Table 5.9 – Classification accuracy results.

| #   | Trained | | | | Not-Trained | | | | | |
|-----|---------|-----------------|-----------------|---------|-----------------|-----------------|---------|-----------------|-----------------|
|     | Average Acc | Best Acc Value | Worst Acc Value | Average Acc | Best Acc Value | Worst Acc Value |
| 1   | 93%       | 100%            | 82%             | 81%       | 97%             | 57%             |
| 2   | 94%       | 99%             | 82%             | 75%       | 97%             | 47%             |
| 3   | 87%       | 100%            | 76%             | 75%       | 94%             | 33%             |
| 4   | 89%       | 100%            | 73%             | 69%       | 91%             | 46%             |
| 5   | 96%       | 99%             | 88%             | 69%       | 98%             | 46%             |
| 6   | 94%       | 99%             | 86%             | 73%       | 98%             | 49%             |
| 7   | 93%       | 99%             | 72%             | 57%       | 92%             | 20%             |
| 8   | 93%       | 100%            | 82%             | 56%       | 93%             | 29%             |
| Average | 92%    | 99%             | 80%             | 69%       | 95%             | 41%             |

From Table 5.9, the following remarks should be made:

- As expected, there is a general loss in accuracy between the videos of the already-trained and not-trained sub-sets, ranging (for the average values) from a 12% loss in test 1 to a 37% loss in test 8.

- The results for the test conditions 7 and 8 show that using the frame luminance or color information does not translate into better results; while the values for the already-trained subset are close to the highest values achieved in the other tests, a visible drop in accuracy is visible in the not-trained videos (so, the trained CNN lacks generalization).

- Test conditions 3 and 4 show the lowest average accuracy in the already-trained sub-set, meaning that the absence of the Edge Intersection procedure has a negative impact on the results.

- The test condition 1 presents the highest accuracy values for the not-trained sub-set.

At this point, some testing conditions can be discarded for their poor results:

- 7 and 8 – although with very good results in the already-trained sub-set, they showed the largest drop in accuracy values for the not-trained sub-set.

- 3 – presents a very low (worst case) accuracy value for the not-trained sub-set.
The remaining test conditions show promising values and will be further analyzed in the next section. All testing conditions will be maintained in the following tables, for comparison purposes.

**Running Time Analysis**

While developing the proposed solution for commercials detection, one of the main concerns was that the sum of all modules processing time should be below the video time. There are three main modules analyzed when it comes to running time: SCD, Post-Processing and Automatic Video Shot CNN-Based Classification.

At first a combined analysis of the resulting processing time of the SCD and Post-Processing modules, with either Canny and LSD edge detectors presented in Table 5.10.

<table>
<thead>
<tr>
<th>Video Sequence</th>
<th>Duration (h:m:s)</th>
<th>Computation Time (h:m:s)</th>
<th>Computation Time (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disney_2H_CA</td>
<td>2:14:28</td>
<td>0:57:10</td>
<td>42%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0:52:17</td>
<td>39%</td>
</tr>
<tr>
<td>Eurosport1_HD_CA</td>
<td>1:42:39</td>
<td>0:41:02</td>
<td>40%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0:37:20</td>
<td>36%</td>
</tr>
<tr>
<td>HW_HD_2H_CA</td>
<td>2:10:27</td>
<td>0:40:13</td>
<td>31%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0:41:23</td>
<td>32%</td>
</tr>
<tr>
<td>RTP1_HD_2H_CA</td>
<td>2:00:51</td>
<td>0:53:15</td>
<td>44%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0:48:13</td>
<td>40%</td>
</tr>
</tbody>
</table>

The Automatic Video Shot CNN-Based Classification is the last section of the solution with more than negligible computation cost, therefore an analysis was performed to determine if the maximum time percentage a classification of a full video would take, the highest value was below 10%. Therefore, in a hypothetical situation where the worst case of both the SCD plus the Post-Processing modules, 44% and the, Automatic Video Shot CNN-Based Classification, would add to no more than 54%, thus the proposed solution is capable of running in real-time.

5.6 Temporal Coherency-Based Reclassification

The Temporal Coherency-Based Reclassification involves two procedures: The Anchor Shot based approach and the Simple Isolation Detection and Correction approach. The values of the related parameters were obtained experimentally and are presented in Table 5.11.
Table 5.11 – Temporal coherency-based classification parameter.

<table>
<thead>
<tr>
<th>TempWindowTH</th>
<th>5 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnchorLoConfTH</td>
<td>70%</td>
</tr>
<tr>
<td>AnchorShotDuratTH</td>
<td>10 s</td>
</tr>
</tbody>
</table>

Table 5.12 presents the accuracy results initially obtained for the different test condition (IC) and their variation after applying the Anchor Shot based approach (AS).

Table 5.12 – Anchor shot approach accuracy improvements over initial classification.

<table>
<thead>
<tr>
<th></th>
<th>Average Accuracy</th>
<th>Best Accuracy Values</th>
<th>Worst Accuracy Values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trained</td>
<td>AS</td>
<td>Not Trained</td>
</tr>
<tr>
<td>IC</td>
<td>AS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>93%</td>
<td>2%</td>
<td>81%</td>
</tr>
<tr>
<td>2</td>
<td>94%</td>
<td>1%</td>
<td>75%</td>
</tr>
<tr>
<td>3</td>
<td>87%</td>
<td>2%</td>
<td>75%</td>
</tr>
<tr>
<td>4</td>
<td>89%</td>
<td>3%</td>
<td>69%</td>
</tr>
<tr>
<td>5</td>
<td>96%</td>
<td>1%</td>
<td>69%</td>
</tr>
<tr>
<td>6</td>
<td>94%</td>
<td>1%</td>
<td>73%</td>
</tr>
<tr>
<td>7</td>
<td>93%</td>
<td>1%</td>
<td>57%</td>
</tr>
<tr>
<td>8</td>
<td>93%</td>
<td>1%</td>
<td>56%</td>
</tr>
</tbody>
</table>

From Table 5.12 the following considerations can be made:

- The Anchor Shot approach achieves better results in the not-trained videos than in the already-trained videos. It also rarely reduces the accuracy values: in 48 results, it reduces the accuracy in only 4; if the 7 and 8 testing conditions are removed, the number lowers to 2. The accuracy improvements are in average higher in the worst accuracy values, and this is achieved without penalizing the best accuracy values, which was a major concern when developing this approach.

- The Anchor Shot approach increases the average and worst-case accuracy in all testing conditions, except in 7 and 8.

- When the initial accuracy result is lower than 30% (which happens in the test conditions 7 and 8) the Anchor Shot approach significantly worsens it, meaning this procedure is useless when the initial results are too poor.
Table 5.13 shows the accuracy changes brought by the combined use of the Anchor Shot approach and the Simple Isolation Detection and Correction approach (AS+S), where it is possible to see a general increase in accuracy. Most of the observations already made for the Anchor Shot approach are still valid for the combination of both approaches; the major difference is a small accuracy decrease (worst values) for the not-trained videos.

Table 5.13 – Anchor shot plus single isolation detection combined approaches improvements.

<table>
<thead>
<tr>
<th>#</th>
<th>Average Accuracy</th>
<th>Best Accuracy Value</th>
<th>Worst Accuracy Values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trained AS</td>
<td>AS+S</td>
<td>Trained AS</td>
</tr>
<tr>
<td>1</td>
<td>95%</td>
<td>2%</td>
<td>85%</td>
</tr>
<tr>
<td>2</td>
<td>95%</td>
<td>1%</td>
<td>79%</td>
</tr>
<tr>
<td>3</td>
<td>89%</td>
<td>3%</td>
<td>78%</td>
</tr>
<tr>
<td>4</td>
<td>92%</td>
<td>3%</td>
<td>73%</td>
</tr>
<tr>
<td>5</td>
<td>97%</td>
<td>1%</td>
<td>73%</td>
</tr>
<tr>
<td>6</td>
<td>95%</td>
<td>2%</td>
<td>75%</td>
</tr>
<tr>
<td>7</td>
<td>94%</td>
<td>2%</td>
<td>62%</td>
</tr>
<tr>
<td>8</td>
<td>94%</td>
<td>2%</td>
<td>60%</td>
</tr>
</tbody>
</table>

Table 5.14 presents the final results to facilitate the choice of the two best testing conditions, that will be analyzed in the next (and last) procedure - the Automatic Learning.

The test condition with an overall best result is condition 1, with the highest accuracy value in two categories - average and worst accuracy values for the not-trained videos - being also well positioned for all other categories.

Due to reasons that will be explained in the next section, a testing condition using the LSD Edge Detection should be also maintained for the Automatic Learning procedure - since 3 was already discarded previously, and 6 presents accuracy values that are in general higher than those obtained for condition 2, condition 6 will be kept.
Table 5.14 – Final temporal coherency-based classification accuracy results.

<table>
<thead>
<tr>
<th>#</th>
<th>Anchor Shot + Simple Isolated Detection Accuracy Results</th>
<th>Trained</th>
<th>Not Trained</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Average Acc</td>
<td>Best Acc Values</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>97%</td>
<td>99%</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>96%</td>
<td>100%</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>93%</td>
<td>99%</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>95%</td>
<td>100%</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>98%</td>
<td>100%</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>97%</td>
<td>100%</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>97%</td>
<td>100%</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>96%</td>
<td>100%</td>
</tr>
</tbody>
</table>

5.7 Automatic Learning Assessment

The Automatic Learning module performs two tasks: detection of a new scenario to learn from (Classification Consistency Analysis), and selection of new shot images to retrain the CNN whenever a new scenario is detected (Representative Image Selection).

Classification Consistency Analysis

For the correct detection of a new scenario the AverageClassificationChange values of the not-trained videos sub-set should have higher values than for the already-trained sub-set. In this case, a correctly selected ClassChangeTH threshold would allow a correct separation between the two sub-sets - if this threshold is set too high, it will make the algorithm miss new scenarios; on the contrary, if the threshold is set too low, the algorithm will start the learning process without being necessary, which may cause the CNN classifier to become specialized in one TV channel and forget features from other TV channels.

The results of the Classification Consistency Analysis for the test condition 1 are presented in Table 5.15, where the videos are ordered accordingly to their average class change. It is word to mention that since the Automatic Learning is directly applied to the CNN classifier output, the accuracy results presented in Table 5.15 are the initial ones.

The results presented in Table 5.15 did not meet the expectations - although most of the already-trained videos have a lower number of classification changes than the observed changes for the not-trained videos, there are some exceptions:

- **AXN** and **FOX** have a higher value when both channels belong to the already-trained sub-set, which will result in unnecessary re-training of the CNN classifier.

- **Eurosport 1** and **RTP 1** have higher average change values than the not-trained videos.
Table 5.15 – Testing condition 1 classification consistency analysis results, ordered by average classification change.

<table>
<thead>
<tr>
<th>Video Sequence</th>
<th>TV Channel</th>
<th>Initial Accuracy (%)</th>
<th>Average Classification Change</th>
<th>Not Trained</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMTV_HD_CA</td>
<td>CM TV</td>
<td>100%</td>
<td>0,01</td>
<td></td>
</tr>
<tr>
<td>SICN_HD_1H_DC</td>
<td>SIC Noticias</td>
<td>98%</td>
<td>0,025</td>
<td></td>
</tr>
<tr>
<td>TVI24_BA</td>
<td>TVI 24</td>
<td>97%</td>
<td>0,043</td>
<td></td>
</tr>
<tr>
<td>Disney_2H_CA</td>
<td>Disney</td>
<td>97%</td>
<td>0,05</td>
<td></td>
</tr>
<tr>
<td>Panda_1H_CA</td>
<td>Panda</td>
<td>96%</td>
<td>0,048</td>
<td></td>
</tr>
<tr>
<td>SIC_HD_3H_BA</td>
<td>SIC</td>
<td>95%</td>
<td>0,07</td>
<td></td>
</tr>
<tr>
<td>Eurosport1_HD_CA</td>
<td>Eurosport</td>
<td>95%</td>
<td>0,057</td>
<td></td>
</tr>
<tr>
<td>CMTVHDA_ads</td>
<td>CM TV</td>
<td>96%</td>
<td>0,069</td>
<td></td>
</tr>
<tr>
<td>SicNotHDBA_moreads</td>
<td>SIC Noticias</td>
<td>93%</td>
<td>0,08</td>
<td></td>
</tr>
<tr>
<td>AXN_WHITE_HD_1H_DA</td>
<td>AXN White</td>
<td>97%</td>
<td>0,054</td>
<td>x</td>
</tr>
<tr>
<td>HW_HD_2H_CA</td>
<td>Hollywood</td>
<td>96%</td>
<td>0,067</td>
<td></td>
</tr>
<tr>
<td>SIC_HD_1H_BB</td>
<td>SIC</td>
<td>92%</td>
<td>0,094</td>
<td></td>
</tr>
<tr>
<td>TVI_2H_CA</td>
<td>TVI</td>
<td>93%</td>
<td>0,096</td>
<td></td>
</tr>
<tr>
<td>SICRAD_HD_AB</td>
<td>SIC Radical</td>
<td>92%</td>
<td>0,11</td>
<td>x</td>
</tr>
<tr>
<td>RTP1_HD_2H_CA</td>
<td>RTP 1</td>
<td>92%</td>
<td>0,109</td>
<td></td>
</tr>
<tr>
<td>PortoCanal_HD_CA</td>
<td>Porto Canal</td>
<td>93%</td>
<td>0,142</td>
<td>x</td>
</tr>
<tr>
<td>SICN_HD_2HBC</td>
<td>SIC Noticias</td>
<td>81%</td>
<td>0,173</td>
<td>x</td>
</tr>
<tr>
<td>Eurosport1HDA_ads</td>
<td>Eurosport 1</td>
<td>84%</td>
<td>0,175</td>
<td></td>
</tr>
<tr>
<td>SPORTTV+_AA</td>
<td>Sport TV</td>
<td>89%</td>
<td>0,132</td>
<td>x</td>
</tr>
<tr>
<td>FOX_HD_2H_CA</td>
<td>FOX</td>
<td>89%</td>
<td>0,186</td>
<td></td>
</tr>
<tr>
<td>NATGEO_HD_1H_DA</td>
<td>National Geographic</td>
<td>83%</td>
<td>0,192</td>
<td>x</td>
</tr>
<tr>
<td>AXN_HD_2H_CA</td>
<td>AXN</td>
<td>82%</td>
<td>0,206</td>
<td></td>
</tr>
<tr>
<td>SICMulher_HD_1H_DA</td>
<td>SIC Mulher</td>
<td>79%</td>
<td>0,242</td>
<td>x</td>
</tr>
<tr>
<td>24Kitchen_HD_CA</td>
<td>24 Kitchen</td>
<td>81%</td>
<td>0,247</td>
<td>x</td>
</tr>
<tr>
<td>Discovery_HD_CA</td>
<td>Discovery</td>
<td>62%</td>
<td>0,245</td>
<td>x</td>
</tr>
<tr>
<td>Nicklodeon_CA</td>
<td>Nickelodeon</td>
<td>57%</td>
<td>0,318</td>
<td>x</td>
</tr>
</tbody>
</table>

As an additional observation, the AXN White channel, although not used in the training set, has a logo very similar to the AXN channel logo; therefore, both should have low average classification change results – however, they occupy quite distinct positions on the table and, furthermore, the not-trained channel is located above the trained channel. Accordingly, even for a limited number of test videos it is not possible to set a proper ClassChangeTH threshold for the test condition 1.
Table 5.16 presents the classification consistency analysis results from the remaining test condition, 6. In this case, a clear separation between already-trained videos and not-trained videos can be seen. Therefore, this test condition was selected as the final one and the \textit{ClassChangeTH} threshold was set to the value 0.159.

Table 5.16 – Testing condition 6 classification consistency analysis results, ordered by average classification change.

<table>
<thead>
<tr>
<th>Video Sequence</th>
<th>Inicial Accuracy(%)</th>
<th>Average Classification Change</th>
<th>Not Trained</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW_HD_2H_CA</td>
<td>99%</td>
<td>0,013</td>
<td></td>
</tr>
<tr>
<td>SICN_HD_1H_DC</td>
<td>98%</td>
<td>0,02</td>
<td></td>
</tr>
<tr>
<td>AXN_WHITE_HD_1H_DA</td>
<td>98%</td>
<td>0,032</td>
<td>x</td>
</tr>
<tr>
<td>Eurosport1_HD_CA</td>
<td>97%</td>
<td>0,035</td>
<td></td>
</tr>
<tr>
<td>TVI24_BA</td>
<td>98%</td>
<td>0,035</td>
<td></td>
</tr>
<tr>
<td>SicNotHDAB_moreads</td>
<td>97%</td>
<td>0,04</td>
<td></td>
</tr>
<tr>
<td>Panda_1H_CA</td>
<td>97%</td>
<td>0,044</td>
<td></td>
</tr>
<tr>
<td>CMV_HD_CA</td>
<td>97%</td>
<td>0,047</td>
<td></td>
</tr>
<tr>
<td>CMVHDAA_ads</td>
<td>97%</td>
<td>0,048</td>
<td></td>
</tr>
<tr>
<td>SIC_HD_3H_BA</td>
<td>95%</td>
<td>0,071</td>
<td></td>
</tr>
<tr>
<td>FOX_HD_2H_CA</td>
<td>95%</td>
<td>0,086</td>
<td></td>
</tr>
<tr>
<td>SIC_HD_1H_BB</td>
<td>93%</td>
<td>0,1</td>
<td></td>
</tr>
<tr>
<td>TVI_2H_CA</td>
<td>92%</td>
<td>0,11</td>
<td></td>
</tr>
<tr>
<td>Disney_2H_CA</td>
<td>93%</td>
<td>0,117</td>
<td></td>
</tr>
<tr>
<td>Eurosport1HDAA_ads</td>
<td>86%</td>
<td>0,133</td>
<td></td>
</tr>
<tr>
<td>AXN_HD_2H_CA</td>
<td>86%</td>
<td>0,146</td>
<td></td>
</tr>
<tr>
<td>RTP1_HD_2H_CA</td>
<td>87%</td>
<td>0,149</td>
<td></td>
</tr>
<tr>
<td>SICRAD_HD_AB</td>
<td>88%</td>
<td>0,189</td>
<td>x</td>
</tr>
<tr>
<td>SPORTTV+_AA</td>
<td>86%</td>
<td>0,201</td>
<td>x</td>
</tr>
<tr>
<td>PortoCanal_HD_CA</td>
<td>84%</td>
<td>0,21</td>
<td>x</td>
</tr>
<tr>
<td>24Kitchen_HD_CA</td>
<td>84%</td>
<td>0,229</td>
<td>x</td>
</tr>
<tr>
<td>NATGEO_HD_1H_DA</td>
<td>72%</td>
<td>0,251</td>
<td>x</td>
</tr>
<tr>
<td>SICN_HD_2HBC</td>
<td>65%</td>
<td>0,255</td>
<td>x</td>
</tr>
<tr>
<td>Discovery_HD_CA</td>
<td>50%</td>
<td>0,256</td>
<td>x</td>
</tr>
<tr>
<td>Nicklodeon_CA</td>
<td>49%</td>
<td>0,298</td>
<td>x</td>
</tr>
<tr>
<td>SICMulher_HD_1H_DA</td>
<td>55%</td>
<td>0,351</td>
<td>x</td>
</tr>
</tbody>
</table>

Representative Image Selection

The Automatic Learning module has a process to select the new shot images that will be used to retrain the CNN classifier. After the detection of a new scenario, the corresponding shot images along with their classifications (from both the CNN classifier and the Temporal Coherency-Based
Reclassification) are used for the selection. This process has the parameters values presented in Table 5.17, which were obtained experimentally.

Due to the high sensitivity of the SCD – that was optimized for a low false negatives rate – the minimum shot duration (AutoDurationTH) for a shot image being selected for retraining had to be set to a quite low value. The CNN retraining has a lower learning rate and a lower number of training step than those of the main training phase, to achieve a balance between learning the new scenario and to not forget the previously trained TV channels.

To demonstrate that the proposed solution can adapt to new scenarios, the test video “Discovery_HD_CA” was selected from the automatic learning library. The initial classification accuracy results for this video are refereed in Table 5.18 by “Attempt #1”. Two different videos from the automatic learning library – “Discovery_1H_DB” and “Discovery_1H_DC” – were than applied to the classifier, and both resulted in a retraining process. Following each retraining, the same “Discovery_HD_CA” sequence was again classified, and the resulting accuracy values are referred in Table 5.17 by Attempt #2 and Attempt #3”. Attempt #3 does not occur because the ClassChangeTH is above the Average Classification Change obtained for “Discovery_1H_DC”. As can be seen from this table, the accuracy results show a clear improvement. Also, the number of selected images highly increases from the first attempt to the second one - as the CNN learns the new scenario, the levels of confidence in each classification rises, which explains the increased number of selected images.

Table 5.17 – Automatic Learning module parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoLevelTH</td>
<td>80%</td>
</tr>
<tr>
<td>AutoDurationTH</td>
<td>1.7 s</td>
</tr>
</tbody>
</table>

Table 5.18 – Automatic learning results for the “Discovery_HD_CA” video.

<table>
<thead>
<tr>
<th># Attempt</th>
<th>Initial Acc</th>
<th>AS+S Acc</th>
<th>Ingested Video</th>
<th>Average Classification Change</th>
<th># of Selected Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50%</td>
<td>55%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>71%</td>
<td>75%</td>
<td>Discovery_1H_DB</td>
<td>0.238</td>
<td>96</td>
</tr>
<tr>
<td>3</td>
<td>86%</td>
<td>92%</td>
<td>Discovery_1H_DC</td>
<td>0.187</td>
<td>847</td>
</tr>
<tr>
<td>4</td>
<td>n.a</td>
<td>n.a</td>
<td>Discovery_1H_DD</td>
<td>0.144</td>
<td>n.a</td>
</tr>
</tbody>
</table>

This shows the ability of the proposed solution to adapt itself when confronted to a scenario it was not trained for.
The impact of the CNN relearning process on the ability to classify the initial TV channels is summarized in Table 5.19, for the testing library videos. The full results are presented in Table 5.20.

Table 5.19 – After learning Discovery channel summarized results on the testing library.

<table>
<thead>
<tr>
<th>CNN Classification</th>
<th>Anchor Shot + Single Isolation Detection Combined Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average change</td>
<td>Highest positive change</td>
</tr>
<tr>
<td>2%</td>
<td>42%</td>
</tr>
</tbody>
</table>

The relearning seems to affect a group of channels in a negative way, while other channels seem to benefit from it. Although the overall effect is not very significative, the minimum resulting accuracy value was 55%, which is above the minimum accuracy value before applying the automatic relearning procedure.

In a real application case, the user should maintain a new TV channel tuned during a certain period, allowing the system to learn it - when the user changes to a new TV channel, the system can warn him when it is ready to function properly, this action being dictated by the system itself, more specifically the Classification Consistency Analysis part of the system.

As a final remark, it should be emphasized that the conceived automatic learning, although being one of the main contributions of the proposed solution for the commercial detection, is just a first step towards a truly automatic learning system - much more tests are required to fully support its feasibility.
### Table 5.20 – Accuracy results per video sequence, after each procedure.

<table>
<thead>
<tr>
<th>Video Sequence</th>
<th>IC Acc (%)</th>
<th>After AS+S Acc (%)</th>
<th>After Automatic Learning Acc (%)</th>
<th>After Automatic Learning + (AS+S) Acc (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>24Kitchen_HD_CA</td>
<td>84%</td>
<td>92%</td>
<td>75%</td>
<td>81%</td>
</tr>
<tr>
<td>AXN_HD_2H_CA</td>
<td>86%</td>
<td>92%</td>
<td>83%</td>
<td>86%</td>
</tr>
<tr>
<td>AXN_WHITE_HD_1H_DA</td>
<td>98%</td>
<td>99%</td>
<td>81%</td>
<td>82%</td>
</tr>
<tr>
<td>CMTV_HD_CA</td>
<td>97%</td>
<td>98%</td>
<td>99%</td>
<td>99%</td>
</tr>
<tr>
<td>CMTVHDAA_ads</td>
<td>97%</td>
<td>98%</td>
<td>92%</td>
<td>93%</td>
</tr>
<tr>
<td>Discovery_HD_CA</td>
<td>50%</td>
<td>55%</td>
<td>88%</td>
<td>93%</td>
</tr>
<tr>
<td>Disney_2H_CA</td>
<td>93%</td>
<td>97%</td>
<td>97%</td>
<td>96%</td>
</tr>
<tr>
<td>Eurosport1_HD_CA</td>
<td>97%</td>
<td>97%</td>
<td>94%</td>
<td>95%</td>
</tr>
<tr>
<td>Eurosport1HDAA_ads</td>
<td>86%</td>
<td>100%</td>
<td>87%</td>
<td>97%</td>
</tr>
<tr>
<td>FOX_HD_2H_CA</td>
<td>95%</td>
<td>98%</td>
<td>87%</td>
<td>88%</td>
</tr>
<tr>
<td>HW_HD_2H_CA</td>
<td>99%</td>
<td>100%</td>
<td>80%</td>
<td>82%</td>
</tr>
<tr>
<td>NATGEO_HD_1H_DA</td>
<td>72%</td>
<td>76%</td>
<td>61%</td>
<td>72%</td>
</tr>
<tr>
<td>Nicklodeon_CA</td>
<td>49%</td>
<td>54%</td>
<td>91%</td>
<td>92%</td>
</tr>
<tr>
<td>Panda_1H_CA</td>
<td>97%</td>
<td>98%</td>
<td>93%</td>
<td>96%</td>
</tr>
<tr>
<td>PortoCanal_HD_CA</td>
<td>84%</td>
<td>89%</td>
<td>94%</td>
<td>96%</td>
</tr>
<tr>
<td>RTP1_HD_2H_CA</td>
<td>87%</td>
<td>95%</td>
<td>96%</td>
<td>97%</td>
</tr>
<tr>
<td>SIC_HD_1H_BB</td>
<td>93%</td>
<td>95%</td>
<td>87%</td>
<td>92%</td>
</tr>
<tr>
<td>SIC_HD_3H_BA</td>
<td>95%</td>
<td>97%</td>
<td>91%</td>
<td>94%</td>
</tr>
<tr>
<td>SICMulher_HD_1H_DA</td>
<td>55%</td>
<td>68%</td>
<td>72%</td>
<td>77%</td>
</tr>
<tr>
<td>SICN_HD_2HBC</td>
<td>65%</td>
<td>73%</td>
<td>92%</td>
<td>95%</td>
</tr>
<tr>
<td>SICN_HD_1H_DC</td>
<td>98%</td>
<td>99%</td>
<td>96%</td>
<td>98%</td>
</tr>
<tr>
<td>SicNotHDAB_moreads</td>
<td>97%</td>
<td>100%</td>
<td>88%</td>
<td>94%</td>
</tr>
<tr>
<td>SICRAD_HD_AB</td>
<td>88%</td>
<td>93%</td>
<td>97%</td>
<td>98%</td>
</tr>
<tr>
<td>SPORTTV+_AA</td>
<td>86%</td>
<td>91%</td>
<td>91%</td>
<td>96%</td>
</tr>
<tr>
<td>TVI_2H_CA</td>
<td>92%</td>
<td>96%</td>
<td>92%</td>
<td>95%</td>
</tr>
<tr>
<td>TVI24_BA</td>
<td>98%</td>
<td>98%</td>
<td>95%</td>
<td>97%</td>
</tr>
</tbody>
</table>
6. Summary and Future Work

In this final chapter, a summary of all the work developed in this thesis and highlighting the conclusions is presented in section 6.1 and areas to further develop in future works in section 6.2.

6.1 Summary and Conclusions

The main objective of this Thesis was to develop a system that can detect TV commercials in present day TV broadcasted content.

In chapter 1 an introduction to TV broadcast and the importance TV commercials still play in the world of advertisement, e.g. the Super Bowl commercials are cited as the pinnacle of marketing spending up to this day; and how TV commercials are very important to the TV stations, as their main source of revenue, which funds their various activities from News segments to entertainment programs. A structure of the Thesis is also presented with a small description of every chapter.

Chapter 2 begins with the legal framework that regulates TV commercials maximum consecutive time per hour, followed by a description of the various exploitable characteristics of TV commercials, some of which will be utilized by this thesis solution; some relevant examples of algorithms that exploit the aforementioned characteristics are also described. The algorithms are separated into two categories, knowledge-based or repetition-based, with an emphasis in the knowledge-based ones because of the methods utilized to exploit the aforementioned characteristics, giving an insight in how to exploit them.

An overview of TV logo detection-based systems is presented in chapter 3. The most significant to this thesis is the first one presented [3][4], that served as a starting point of this thesis algorithm, a solution based on the TV logo appearance only when outside commercial blocks, the system relies on the persistence of said logo to determine the video segment classification, after a description of the main sections,; the bigger shortcomings are pointed out to inform these thesis system design. The following described systems have different features detection that is then utilized in conjunction with a machine learning model, taking advantage of the robustness of this models. These systems features selection before deployment of the machine learning models served as an inspiration to the SCD and Post-Processing section of this thesis.

In chapter 4, a TV channel characterization with an emphasis on the elements the system was designed to capture and preserve in the SCD and Post-Processing sections, especially the ones visible on regular programming is performed. The developed system is described in detail, section by section starting with how the video input is segmented, in the SCD section; followed by the steps required to achieve a single image that accurately represents the collection of frames within a shot while preserving the on-screen elements the CNN classifier will need to correctly classify, resulting in the final shot image, Post-Processing section. The Automatic Video Shot CNN-Based Classification that uses a CNN classifier chosen for its faster computation time and high
classification accuracy. Also, the guidelines for picking the correct training dataset are laid out, ending with a small analysis on the effects certain elements have on the level of confidence given by the CNN classifier when assessing a shot image. The last two sections were designed to assist the Commercial detection by the CNN, Temporal Coherency-Based Classification corrects the classifications given by the CNN utilizing the knowledge presented in chapter 2; the Automatic Learning section was thought out because the CNN by itself cannot adapt to scenarios it was trained for, therefore the new scenario detection subsequently new training images selection was envisioned to address these scenarios.

Chapter 5 contains everything related to the performance assessment of the Global Solution. The video libraries captured from various TV channels to train and test the various sections, followed by the performance assessment of the various sections of this thesis solution, starting with the SCD. The SCD assessment is divided into two analysis: in analysis A the goal is to determine how well the SCD detects hard-cuts, the normal separation between commercials; in analysis B most of the Testing library is used to analyze if the SCD detects all transitions between regular and non-regular programming, or vice-versa, the SCD is proven to detect all transitions; these were performed with a careful selection of the SCD parameters that resulted in an SCD that is overly sensitive. The CNN classifier is then assessed with the testing library with eight testing scenarios, consisting of various combinations of SDC and Post-Processing parameters, for every testing conditions a CNN is trained with selected images extracted from the Training library with the same SCD and Post-Processing parameters; the obtained results show promise in the deployment of a pre-trained CNN in the TV commercials detection, the results are improved with the temporal correlation based approaches resulting in accuracy values on average of 95% for the already trained subset of videos. For the untrained scenarios it is proven that the system can detect when it is performing poorly and also with the most difficult example, the Discovery Channel it is demonstrated that the system can learn and improve its result from 55% to 92% accuracy.

The Global Solution shows two main strengths; the use of a pre-trained CNN, when the papers reviewed in this thesis choose to utilize their own machine learning models when in the present day various pre-trained architectures are available for usage; these pre-trained models have compared to the average researcher, in their initial training phase access to an incredibly large amount of computational power, which can be used in more specific uses as proved by these thesis results.

The ability for the Global Solution to detect when it is performing poorly and to acquire the training data it needs to return the classifications to their normal high level of accuracy it’s the most important part of the Global Solution. Additionally, the accuracy improvements achieved with the Anchor Shot and Simple Isolation Detection approaches are interesting for a very low computational cost. The use of the four corners of the video frame at all times gives more robustness to the Global Solution by enabling the CNN classifier to learn more elements than the TV channel logo it gives a larger redundancy against hard to identify cases because of background diversity and/or poor edge detection.
6.2 Future Work

The final results obtained validate the approach taken in this thesis for a system capable of distinguishing between regular and non-regular programming, there are various areas that could be improved with further work, namely:

- **Larger dataset** – although the dataset used in this thesis amounts to more than 150 hours of recorded video from a large variety of TV channels, the dataset can always be expanded with more hours per channel and more importantly more TV channels, perhaps even from different countries. All TV channels captured are from Portuguese TV channels by including other countries in the dataset it would make the Global Solution a classification system with broader appeal.

- **A new SCD method** – the SCD section used in this thesis overly sensitive to small luminance changes, a more complex method could more easily avoid false negatives and still detect all transactions between commercials and regular programming.

- **Pre-trained CNN architectures** – the used CNN architecture, Inception in its third iteration [26], was utilized for its advancements when compared with the more commonly used AlexNet [31] and ease of use, however a comparison between AlexNet and other more recent CNN architectures like Inception fourth iteration [35], VGG Net [36], DenseNet [37], ResNet [38], could reveal interesting results.

- **New CNN** – with the data optimizations deployed in this thesis, designing a simpler more focused CNN architecture, only trained with shot images, would be feasible, by utilizing a high-level API such as Keras [39], a new CNN from scratch could be deployed and tested against more generalized CNN’s like the one mentioned above.

- **Recurrent Neural Network** – Recurrent Neural Networks (RNNs) are CNNs with at least one feedback loop connection, this enables the CNN to learn temporal patterns, which could be used in TV commercials detection, where the TV channels display temporal loops of regular and non-regular programming. In this thesis a section was created to address the lack of internal memory the CNN has, this could perhaps be resolved with the deployment of an RNN.
References


3, p. iii-541-4.


A. Basic Concepts of Convolutional Neural Networks (CNN)

A.1 CNN Composition

A traditional CNN consists of five main layers [40]:

1. Input Layer
2. Convolution Layer
3. Non-linearity (Rectified Linear Unit Layer)
4. Pooling (Sub-sampling) Layer
5. Classification – Fully Connected Layer

Input Layer

The CNN receives an image, and extracts the numerical values for every pixel, from 0 to 255 with three to one channels, for respectively, images in color or in grayscale. In this thesis, color was never used to reduce computation time, and most tests were run with binary images further reducing computation time. The input shot image is therefore represented by a two-dimensional matrix.

Convolution Layer

The objective of using convolutions is the extraction of features from the input shot image. Features are small portions of the original image, they match frequent elements of the original image, a general example is presented in Figure A.1.

![Figure A.1 – Features matching [41]](image)

Convolutions have the benefit of preserving the relative spatial relationships between the pixels, depending on the size of the convolution kernel, $n \times n$ or filter in CNN terminology, therefore capturing different features.
The filters are learned during training, allowing the CNN to adapt itself to various areas of image recognition and classification.

The convolution layer performs the following steps, for each channel of the original image:

1. Receive the two-dimensional matrix (per channel) from the original shot image.
2. The filter starts multiplying for every element between the two matrices, adding up the multiplications output, to obtain the final single number of the final matrix (left).
3. The filter slides over a certain number of positions, strides, and repeats until the final matrix is completed.

The final matrix is called a feature map, it has some parameters:

- **Depth** – It corresponds to the number of filters that were used in the convolution step, each one optimally detecting different features.
- **Stride** – The number of pixels the filter moves between each multiplication with the original matrix. The bigger the stride the less information is collected, on the contrary, it reduces computational cost.
- **Zero-Padding** – To avoid mismatching between the filter and original matrix sizes, a common practice is to pad the border of the original matrix with zeros which allows applying the filter to border elements that otherwise would be lost.

**Rectified Linear Unit Layer**

Rectified Linear Unit, (RELU) is a simple function that is applied in every pixel of the feature map, and simply replaces every negative value with zero, resulting in a rectified feature map, it does not change the size of the input.

\[ f(x) = \max(0, x) \]  
(A.1)

This is the activation method used in AlexNet [31] and on the Inception architecture [32].
This layer purpose to introduce non-linearity to the CNN, adding non-linearity to the CNN is crucial, because if non-linearity was not introduced, the CNN output would be a linear transformation of the input and real-world problems are not linear. To make input data non-linear an activation function must be used, because the other layers convolution and pooling, perform linear operations.

**Pooling layer**

The purpose of the Pooling layer is the reduction of the size of each feature map (subsampling), reducing the computational cost while retaining the most relevant information. It is performed on each feature map independently.

Two Types of pooling are used in the Inception architecture:

- **Max Polling** – operate on a window retaining only the highest value element, this pooling method is more common, an example is depicted in Figure A.3

  ![Max pool with a 2 x 2 filters and a stride of 2](image)

  *Figure A.3 – Max pooling operation on a Left: Original matrix with Right: Output matrix [40]*

- **Average Polling** – computing the average value, of the elements inside a window, keeping only the average value per window.

**Classification - Fully Connected Layer**

The main characteristic of a fully connected layer is every node of one layer, is connected to all the nodes of the adjacent layers.

Fully connected layers take the results from the output of the previous layer and translate them into votes. Fully connected layers treat inputs as a single list, every value gets its own vote in the decision, in the case of this thesis, a binary classification, the votes are either on regular or non-regular. When making decision votes are not equal, some values count more for one classification than the others, this means they have different weights or connection strengths between each value and each class, the weights, and strengths of the connections are tuned during learning.

The fully connected layer also adds non-linear combinations of the previously acquired features, from the convolution and pooling layers. The sum of the output from the fully connected layer is one, this is assured using a SoftMax function that squashes the output, to values between one and zero.
A.2 CNN Learning Process

The learning process of the CNN is summarized in the following items:

1. Initialization – All filters and weights are initiated with randomized values
2. First image – A training image is taken as input, goes through the all the layers, giving an output probability for each classification. The results will be very divergent from the intended classification because the network was initialized with random variables.
3. Error computation – The total error is computed at the output layer, with (A.2) where \( n \) is the number of classes.

\[
Total\ Error = \sum_{i=1}^{n} (desired\ probability - output\ probability)^2
\]

(A.2)

4. Backpropagation – consists of computing the gradient of the error of all weights in the network, gradient descent to update all filters values/weights and parameters values to minimize the error of the outputs.
   - The weights are altered accordingly to their effect on the total error.
   - If the same image was again put through the CNN the error would be reduced, meaning it would be closer to the desired classification, the network learned to better classify this image.

The network during learning can adapt the matrix of the filters and the connection weights. Parameters that do not change during training are the filter sizes and the number of filters.

Repetition – the steps one to four are repeated for all the training images.
### B. Training and Testing Video Library

#### B.1 Training Library

Table B.1 – Training Library.

<table>
<thead>
<tr>
<th>Name of the video</th>
<th>Channel</th>
<th>Duration</th>
<th>Logo(s) notes</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXN_HD_AC_long_limpo_</td>
<td>AXN HD</td>
<td>1:06:16</td>
<td>Opaque, static with complex contours, event-specific extras</td>
<td>TV series, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>AXN_HD_long_AA_limpo_</td>
<td>AXN HD</td>
<td>5:45:04</td>
<td>Opaque, static with complex contours, event specific extras</td>
<td>TV series, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>CMTV_HD_AB</td>
<td>Correio da Manhã TV HD</td>
<td>4:59:21</td>
<td>Opaque, static, with clock</td>
<td>News, Talk Show, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>Disney_AA_longo</td>
<td>Disney Channel</td>
<td>4:19:55</td>
<td>Static, with transparency, complex contours, program specific logo</td>
<td>Kids Shows, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>Eurosport1_HD_4h_BA_limpo_</td>
<td>Eurosport 1 HD</td>
<td>3:51:46</td>
<td>Static, with transparency, complex contours, event-specific extra</td>
<td>Sports Events, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>Eurosport1_HD_4h_BB</td>
<td>Eurosport 1 HD</td>
<td>3:58:44</td>
<td>Static, with transparency, complex contours, event-specific extra</td>
<td>Sports Events, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>Eurosport1_HD_long_BA_</td>
<td>Eurosport 1 HD</td>
<td>5:14:48</td>
<td>Static, with transparency, complex contours, event-specific extra</td>
<td>Sports Events, Broadcaster Self-Promotion, Talk Show, Ads</td>
</tr>
<tr>
<td>FOX_HD_long_AD</td>
<td>Fox HD</td>
<td>7:10:22</td>
<td>Opaque, static</td>
<td>TV series, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>FOX_HD_long_AE</td>
<td>Fox HD</td>
<td>4:19:41</td>
<td>Opaque, static</td>
<td>TV series, Movies, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>HW_HD_AE_Long</td>
<td>Hollywood HD</td>
<td>6:30:54</td>
<td>Opaque, static</td>
<td>Movies, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>Panda_AA_limpo_</td>
<td>Panda</td>
<td>00:46:45</td>
<td>Opaque, static</td>
<td>Kids Shows, Broadcaster Self-Promotion</td>
</tr>
<tr>
<td>File Name</td>
<td>Broadcaster</td>
<td>Time</td>
<td>Description</td>
<td>Promotions</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Panda_Long_AB_limpo</td>
<td>Panda</td>
<td>6:46:45</td>
<td>Opaque, static</td>
<td>Kids Shows, Self-Promotion, Ads</td>
</tr>
<tr>
<td>RTP1_HD_AA_long</td>
<td>RTP1 HD</td>
<td>8:50:59</td>
<td>Static, with transparency, program specific logo</td>
<td>News, Self-Promotion, Talk shows, Ads</td>
</tr>
<tr>
<td>SIC_HD_AC</td>
<td>SIC HD</td>
<td>1:00:03</td>
<td>Dynamic texture, event-specific extra</td>
<td>News, Self-Promotion, Ads</td>
</tr>
<tr>
<td>SIC_HD_AE</td>
<td>SIC HD</td>
<td>4:27:18</td>
<td>Dynamic texture, event-specific extra</td>
<td>Talk Show, Football Game, Self-Promotion, Ads</td>
</tr>
<tr>
<td>SICN_HD_AC_long</td>
<td>SIC Noticias HD</td>
<td>4:21:29</td>
<td>Static, with transparency and clock</td>
<td>Discussion panels, News, Self-Promotion, Ads</td>
</tr>
<tr>
<td>TVI_Long_AA__</td>
<td>TVI</td>
<td>4:34:21</td>
<td>Dynamic texture, clock (during news)</td>
<td>Talk show, Self-Promotion, News, Ads</td>
</tr>
<tr>
<td>TVI_Long_AB_limpo</td>
<td>TVI</td>
<td>6:36:39</td>
<td>Dynamic texture, program specific logo, clock (during news)</td>
<td>Soap opera, Self-Promotion, News, Ads</td>
</tr>
<tr>
<td>TVI_Long_AC_limpo</td>
<td>TVI</td>
<td>5:18:26</td>
<td>Dynamic texture, program specific logo</td>
<td>Soap opera, TV series, Self-Promotion, Ads</td>
</tr>
<tr>
<td>Tvi24_4h_AA_limpo__</td>
<td>TVI 24</td>
<td>3:57:46</td>
<td>Opaque, static with clock. Logo present as a watermark on non-regular programming</td>
<td>News, Discussion Panels, Self-Promotion, Ads</td>
</tr>
<tr>
<td>Tvi24_Long_AC_limpo__</td>
<td>TVI 24</td>
<td>4:20:48</td>
<td>Opaque, static with clock. Logo present as a watermark on non-regular programming</td>
<td>News, Discussion panels, Self-Promotion, Ads</td>
</tr>
<tr>
<td>Tvi24_Long_AD</td>
<td>TVI 24</td>
<td>6:53:02</td>
<td>Opaque, static with a clock. Logo present as a watermark on non-regular programming</td>
<td>News, Self-Promotion, Ads, Infomercials</td>
</tr>
</tbody>
</table>
### B.2 Testing Library

**Testing Library – Already-Trained subset**

Table B.2 – Testing Library Already-Trained subset.

<table>
<thead>
<tr>
<th>Name of the video</th>
<th>Channel</th>
<th>Duration</th>
<th>Type of TV logo</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXN_HD_2H_CA</td>
<td>AXN</td>
<td>02:37:41</td>
<td>Opaque, Static, event-specific logo</td>
<td>TV series, Movies, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>CMTV_HD_CA</td>
<td>Correio da Manhã TV</td>
<td>01:17:22</td>
<td>Opaque, static, with clock</td>
<td>News, ads</td>
</tr>
<tr>
<td>CMTVHDAA_ads</td>
<td>Correio da Manhã TV</td>
<td>0:26:21</td>
<td>Opaque, static, with clock</td>
<td>News, ads</td>
</tr>
<tr>
<td>Disney_2H_CA</td>
<td>Disney Channel</td>
<td>2:14:28</td>
<td>Static, with transparency, complex contours, program specific logo</td>
<td>Kids, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>Eurosport1_HD_CA</td>
<td>Eurosport 1</td>
<td>1:42:39</td>
<td>Static, with transparency, complex contours, event-specific extra</td>
<td>Sports Events, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>Eurosport1HDAA_ads</td>
<td>Eurosport 1</td>
<td>0:34:29</td>
<td>Static, with transparency, complex contours</td>
<td>Sports Events, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>FOX_HD_2H_CA_</td>
<td>Fox</td>
<td>2:02:30</td>
<td>Opaque, static</td>
<td>TV series, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>HW_HD_2H_CA</td>
<td>Hollywood</td>
<td>2:10:27</td>
<td>Opaque, static</td>
<td>Movies, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>Panda_1H_CA</td>
<td>Panda</td>
<td>0:45:02</td>
<td>Opaque, static</td>
<td>Kids Shows</td>
</tr>
<tr>
<td>RTP1_HD_2H_CA</td>
<td>RTP 1</td>
<td>2:00:51</td>
<td>Opaque, static</td>
<td></td>
</tr>
<tr>
<td>SIC_HD_1H_BB</td>
<td>SIC</td>
<td>0:59:41</td>
<td>Static, dynamic texture, event-specific extra</td>
<td>Talk Show, Football Game, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>SIC_HD_3H_BA</td>
<td>SIC</td>
<td>3:14:35</td>
<td>Static, dynamic texture</td>
<td>News, Soap Opera, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>SicNotHDAB_moreads</td>
<td>SIC Noticias</td>
<td>0:18:19</td>
<td>Static, with transparency and clock</td>
<td>News, Ads</td>
</tr>
<tr>
<td>SICN_HD_1H_DC</td>
<td>SIC Noticias</td>
<td>1:07:35</td>
<td>Static form (with sometimes hard to determine limits), dynamic texture with clock below</td>
<td>News, Ads, Broadcaster Self-Promotion</td>
</tr>
<tr>
<td>TVI_2H_CA</td>
<td>TVI</td>
<td>1:56:57</td>
<td>Static form, dynamic texture with clock below</td>
<td>News, Ads, Broadcaster Self-Promotion</td>
</tr>
<tr>
<td>Name of the video</td>
<td>Channel</td>
<td>Duration</td>
<td>Type of TV logo</td>
<td>Observations</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------</td>
<td>----------</td>
<td>--------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>24Kitchen_HD_CA</td>
<td>24 Kitchen</td>
<td>0:58:18</td>
<td>Opaque, Static, Program-specific logos</td>
<td>TV series, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>AXN_WHITE_HD_1H_DA_</td>
<td>AXN White</td>
<td>1:19:05</td>
<td>Static form, opaque texture, great similarities with the AXN HD channel logo</td>
<td>TV series, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>NATGEO_HD_1H_DA_</td>
<td>National Geographic Channel</td>
<td>1:00:59</td>
<td>Static form and opaque texture</td>
<td>Documentaries, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>SICRAD_HD_AB</td>
<td>SIC Radical</td>
<td>1:30:44</td>
<td>Static form and opaque texture</td>
<td>TV Series, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>SPORTTV+_AA</td>
<td>Sport TV +</td>
<td>0:38:53</td>
<td>Static form and opaque texture</td>
<td>Sport Events, News, Ads</td>
</tr>
<tr>
<td>Discovery_HD_CA</td>
<td>Discovery Channel</td>
<td>1:34:29</td>
<td>Static form, semitransparent texture</td>
<td>Documentaries, Broadcaster Self-Promotion, Ads</td>
</tr>
<tr>
<td>Nicklodeon_CA</td>
<td>Nickelodeon</td>
<td>1:03:58</td>
<td>Opaque, static</td>
<td>Kids Shows, Ads</td>
</tr>
<tr>
<td>SICN_HD__2HBC</td>
<td>SIC Noticias</td>
<td>1:46:47</td>
<td>Animated, with transparency and clock, alternate form</td>
<td>News, Ads</td>
</tr>
<tr>
<td>PortoCanal_HD_CA</td>
<td>Porto Canal</td>
<td>1:36:39</td>
<td>Opaque, animated with two alternating logos</td>
<td>News, Talk Shows, Ads</td>
</tr>
<tr>
<td>SICMulher_HD_1H_DA_</td>
<td>SIC Mulher</td>
<td>1:13:33</td>
<td>Static form and opaque texture</td>
<td>TV Series, Broadcaster Self-Promotion, Ads</td>
</tr>
</tbody>
</table>