République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A.MIRA-BEJAIA

Faculté des Sciences Exactes
Département d'Informatique

THÈSE

Présentée par

AMRANE Abdesalam

Pour l'obtention du grade de

DOCTEUR EN SCIENCES

Filière : Informatique

Option : Réseaux et Systèmes Distribués

Thème

Multimedia Content Classification in the Cloud

Soutenue le : 26/05/2022 Devant le Jury composé de :

Nom et Prénom	Grade		
Mr BELAID Ahror	Professeur	Univ. de Béjaia	Président
Mr MEZIANE Abdelkrim	Directeur de Recherche	CERIST	Rapporteur
Mr AMROUN Kamal	Professeur	Univ. de Béjaia	Examinateur
Mr CHERFA Yazid	Professeur	Univ. de Blida	Examinateur
Mr MAREDJ Azze-Eddine	MRA	CERIST	Examinateur

Année Universitaire : 2021/2022

Dedicated to my family

ACKNOWLEDGEMENTS

Working on the doctorate was stressful, but I had to succeed. I would like to express my sincere gratitude to my advisors Dr. Abdelkrim Meziane and Dr. Abdelmounaam Rezgui for their qualified support and help with all the advice and practical support that I required over the years.

I owe thanks to everybody else working at the Research Center on Scientific and Technical Information (CERIST), starting by the initiator of the doctoral school Prof. Nadjib Badache, the direction staff, all PhD students, researchers, and Abdelhamid Lebal.

I am also thankful to the University Abderrahmene Mira, Bejaia, Computer Science Department staff, and the School of Information Technology faculty and staff at Illinois State University, USA.

A special recognition to MediaMarketing Group, starting by Mr Youcef Aggoun (DG) and all staff members.

Finally I would like to thank my family and friends especially Rachid Hacene, who supported me during the last years to finish writing this thesis.

TABLE OF CONTENTS

DEDICATION i

ACKNOWLEDGEMENTS ii

LIST OF FIGURES vi

LIST OF TABLES vii

LIST OF ABBREVIATIONS viii

CHAPTER

I. Introduction 1

1.1 Motivation 1
1.2 Problem Statement 2
1.3 Contributions 3
1.4 Thesis Organization 4

II. Background 6

2.1 Multimedia Content Segmentation 6
2.2 Multimedia Content Classification 10
2.2.1 Fingerprint-based Methods: 15
2.2.2 Distance-based Methods: 16
2.2.3 Deep Learning Methods: 16
2.3 Multimedia Processing in Cloud 26
2.3.1 Cloud Architecture 27
2.3.2 Video Analytics in Cloud 29
2.3.3 Chapter summary 31

III. Real time TV Content Analysis for Multimedia Monitoring System 33

3.1 Introduction 33

3.2 Related Work 35
3.3 Proposed System 38
3.3.1 Collecting Block 39
3.3.2 Editing Block 40
3.3.3 Delivering Block 40
3.4 TV News Processing 41
3.4.1 TV Stream Segmentation 41
3.4.2 News Program Identification 42
3.4.3 News Topics Extraction 46
3.4.4 Audio Extraction 48
3.4.5 Audio Transcription 48
3.4.6 Text Region Extraction 48
3.4.7 Text Transcription 49
3.4.8 Text Grouping 49
3.5 Advertisements Detection and Classification 50
3.5.1 Frame Hashing-Based Classification 50
3.5.2 Audio Features Based Classification 51
3.6 Experiments 52
3.6.1 Experiments on national TV channels 52
3.6.2 Experiments on international TV channels 55
3.7 Conclusion 56

IV. A Deep Hybrid Model for Advertisements Detection in Broadcast TV and Radio Content 58

4.1 Introduction 58
4.2 Related work 60
4.3 The Proposed Framework 62
4.4 Processing Block Description 63
4.4.1 Audio Segmentation 63
4.4.2 Features Extraction 64
4.4.3 Advertisements Detection and Classification 64
4.5 Deep Neural Network Models 65
4.6 Experiments 68
4.7 Conclusion 72

V. A Big Data Framework for Video Content Processing 74

5.1 Introduction 74
5.2 Related Work 75
5.3 Architecture of the Big Data Framework for Video Content Processing 76
5.4 Video Collector 78
5.5 Data Processing 78
5.6 Categories Fusion 79

5.7 Experimentation 80
5.8 Conclusion 81

VI. Papers and Author's Contributions 82

6.1 Paper I 82
6.2 Paper II 82
6.3 Paper III 82
6.4 Paper IV 83

VII. Conclusions 84

7.1 Summary 84
7.2 Future Work 85

BIBLIOGRAPHY 86

LIST OF FIGURES

Figure

2.1 Hierarchical structure of video 7
2.2 Overview of TV stream segmentation 8
2.3 Neural network with two hidden layers 17
2.4 CNN topology with the main building blocks 19
2.5 Architecture of AlexNet 20
2.6 Architecture of VGGNet 21
2.7 Inception module used in GoogLeNet 21
2.8 ResNet block illustration 22
2.9 Simple RNN topology - Elman & Jordan 23
2.10 An architecture of autoencoder neural network 26
2.11 Cloud Architecture 27
3.1 Information monitoring process 34
3.2 Overall architecture of the multimedia monitoring system 39
3.3 TV stream processing 41
3.4 Text region extraction 49
4.1 The architecture of an Autoencoder 61
4.2 The block diagram of the proposed TV and Radio Advertisements Monitoring System 63
4.3 Block diagram of advertisements detection and classification 65
4.4 Mean Square Error calculated from the training and test data . . . 69
4.5 Performance accuracy of Advertisements Detection and Classification models 70
4.6 Speed-ups for the three models 71
5.1 Architecture of the Big Data Framework for Video Content Processing 77

LIST OF TABLES

Table

3.1 CNN classifier 51
3.2 TV Stream Analysis (1 hour) with different descriptors 53
3.3 Multicores CPU parallel processing 53
3.4 News program identification 54
3.5 News topics segmentation 54
3.6 News topics segmentation 55
3.7 Comparison with other approaches 56
4.1 ANN classifier model 66
4.2 CNN classifier model 67
4.3 RNN classifier model 67
4.4 Training and test parameters 69
4.5 Precision, Recall, and F-measure achieved over the implemented models 71
4.6 Comparison with other approaches 72
5.1 Execution time for video processing on Spark 80
5.2 Pecision of the video processing framework on Spark 80

LIST OF ABBREVIATIONS

ANN Artificial Neural Network

ASR Automatic Speech Recognition

BRIEF Robust Independent Elementary Features

CNN Convolution Neural Network

CPU Central Processing Unit

DCT Discrete Cosine Transform

DFT Discrete Fourrier Transform

DTW Dynamic Time Warping

EIT Event Information Table

EPG Electronic Program Guide

FAST Spectral Flux

GPU Graphical Processing Unit

HADOOP High Availability Distributed Object Oriented Platform

HDFS Hadoop Distributed File System

HOG Histogram of Oriented Gradients

IAAS Infrastructure As A Service

ICT Information and Communications Technology

IP Inter Program

LBP Local Binary Pattern

LSTM Long Short Term Memory network

MFCC Mel Frequency Cepstral Coefficients

MPEG Moving Picture Experts Group

NLP Natural Language Processing

OC-NN One-Class Neural Network

OCR Optical Character Recognition

OC-SVM One-Class SVM

PAAS Platform As A Service

RDD Resilient Distributed Dataset

RNN Recurrent Neural Network

SAAS Software As A Service

SC Spectral Centroid

SF Spectral Flux

SIFT Scale Invariant Feature Transform

STE Short Time Energy

SURF Speed Up Robust Feature

SVM Support Vector Machine

VGG Visual Geometry Group

ZCR Zero Crossing Rate

Abstract: Information extraction from multimedia content is a challenging task. In this thesis, we present an architecture of multimedia contents classification system that provides different phases to extract semantic information from broadcasted streams, starting with the segmentation process, news topics extraction, and advertisement detection and classification. Next, we give an extension to our framework and describes an audio-based hybrid model for content classification combining different deep neural networks with auto-encoder applied to advertisement detection in TV broadcast. Our models achieve high levels of precision. The last contribution consists of a distributed architecture based on the Kafka and Spark frameworks which offer parallel processing of TV streams, we demonstrate through this work the scalability and robustness of this architecture.

Keywords: Multimedia processing; Parallel processing; Deep learning; TV stream analysis; News identification; Advertisement extraction; Media monitoring

Résumé : L'extraction d'informations à partir de contenus multimédias est une tâche difficile. Dans cette thèse, nous présentons une architecture d'un système pour la classification des contenus multimédia qui fournit différentes phases pour extraire des informations sémantiques des flux diffusés, en commençant par le processus de segmentation, l'extraction de sujets d'actualité et la détection des publicités. Ensuite, on propose une extension de ce système consiste en un modèle hybride basé sur l'audio pour la classification de contenu combinant différents réseaux de neurones profonds avec un auto-encodeur appliqué à la détection de publicité diffusée sur la télévision. Les modèles proposés ont atteint des taux de succès très élevés. La dernière contribution consiste en une architecture distribuée basée sur les frameworks Kafka et Spark qui offrent un traitement parallèle des flux TV, nous démontrons par ce travail l'évolutivité et la scalabilité de cette architecture.

Mots clés : Multimedia processing; Parallel processing; Deep learning; TV stream analysis; News identification; Advertisement extraction; Media monitoring

ملخص : يعد استخراج المعلومات من محتوى الوسائط المتعددة مهمة صعبة. ففي هذه الرسالة، نقدم بنية نظام لتصنيف محتويات الوسائط المتعددة والتي توفر مراحل مختلفة لاستخراج المعلومات الدلالية من تدفقات البث، بدءًا من عملية التجزئة واستخراج الموضوعات الموضعية وإعلانات الكشف. بعد ذلك، يُقترح امتداد هذا النظام ليتألف من نموذج هجين قائم على الصوت لتصنيف المحتوى يجمع بين شبكات عصبية عميقة مختلفة مع مشفر تلقائي مطبق على الكشف عن الإعلانات التي يتم بثها على التلفزيون. لقد حققت النماذج المقترحة معدلات نجاح عالية جدًا. تتكون المساهمة الأخيرة من بنية موزعة تستند إلى أطر عمل كافكا وسبارك التي تقدم معالجة متوازية لتدفقات التلفزيون، ونبين من خلال هذا العمل قابلية التوسع وقابلية التوسع في هذه البنية.

الكلمات الدالة : معالجة الوسائط المتعددة؛ المعالجة المتوازية؛ تعلم عميق؛ تحليل تيار التلفزيون؛ تحديد الأخبار؛ استخراج الإعلانات؛ رصد وسائل الإعلام

CHAPTER I

Introduction

1.1 Motivation

Today, we are living in the multimedia information era where a huge amount of multimedia content is generated every day. The most important provider of multimedia information is the television that broadcasts content constantly. Such massive multimedia data need to be stored, analyzed, and processed for further retrieval. A multimedia content comes in various forms of video, image, audio, and text. The analysis refers to extracting the semantic meaning of a multimedia document. This usually involves segmenting it into semantically meaningful units, classifying each unit into a predefined scene type, and indexing the document for efficient retrieval and browsing [173].

With today's 24/7 news and information cycle broadcasted on TV channels, companies need to monitor everything about their brands, competitors, and industry issues. Now it becomes essential to include broadcast monitoring in an integrated media monitoring system that identifies and classify information in topics related to the users interests. Another kind of information is the advertisement that is a major source of marketing in recent years and considered important to promote the business.

Toward this goal, many systems have been proposed and we can group them into two categories: The first consists of systems that search the closed-caption text embedded in the TV news broadcast signal and deliver the closed-caption text joined with a preview video of any news segments related to the users interests. These systems process only visual features. The second category uses speech-to-text software to transcript the content of news programs and classifies the resulting text. Such a system is an audio-based system and it can also be used to process radio news. Speech-to-text software is more accurate for the English language and less so for the French and Arabic languages.

In this thesis, we will investigate different approaches ranging from content segmentation, image/audio hashing, and similarity distance, to deep learning-based methods in order to employ the most cost-efficient solution related to our local context.

1.2 Problem Statement

In this thesis, we show that there are significant challenges that must be addressed in order to perform the task of multimedia content classification successfully. A TV broadcast is composed of an audio-visual stream and a metadata stream. Metadata provides textual information such as closed-caption [12]. Also, broadcasters embed black frames in the stream to facilitate program segmentation. However, local TV channels don't use metadata or black frames embedding option in broadcasting process.

Our target is to develop a deep learning based models and algorithms that effectively capture streams, detect and classify both news and advertisement segments. We focus on three challenging tasks:

1. Performing a segmentation of audio and video streams in order to identify news or advertisements: In this problem, we are interested in capturing all news segments broadcast from different channels,

2. Extracting and classifying news topics into a set of predefined categories: The goal is to split the news broadcast into topics and extract the headlines from the visual content, this applies to TV channels that do not embed text in the broadcasted stream (MPEG-7),

3. Detecting and classifying advertising segments: We tackle the issue of media monitoring in order to provide a fast audit of all advertising activities for companies and their competitors.

1.3 Contributions

This thesis contributes at different levels of multimedia content classification fields. Our main contributions are listed below:

- Fast and smart object proposals for object detection: Object localization plays an important role in object detection and classification. In the last years, several methods have shifted from sliding windows techniques to object proposals techniques. The latter produces a small set of windows submitted to an object classifier to reduce the computational time. In this paper, we propose a fast unsupervised method that combines the edge feature and saliency map to generate less than hundred bounding boxes from the processed image. Our approach exploits a number of rules based on edges information plus saliency regions to decide if an object is present in a window. We have carried out several experiments to validate our approach on the ImageNet dataset and obtained very promising results.

- Object detection in images based on homogeneous region segmentation: Most popular approaches in the segmentation/object detection tasks use sliding-window or super-pixel labeling methods. The first method suffers from the number of window proposals, whereas the second suffers from the over-segmentation problem. To overcome these limitations, we present two strategies: the first one is a fast algorithm based on the region growing method for segmenting images into homogeneous regions. In the second one, we present a new technique for similar region merging, based on a three similarity measures, and computed using the region adjacency matrix. We have evaluated all of these methods and compared to other state-of-the-art approaches that were applied on the Berkeley image database. The experimentations yielded promising results and would be used for future directions in our work.

- Real time TV Content Analysis for Multimedia Monitoring System: Media monitoring plays an important role in the success of companies by protecting their corporate, staff, and brand reputation. With the emergence of ICT, monitoring has emerged as an inevitable means of covering various fields of media: newspapers, online news, broadcast news (TV, radio), and social networks. Companies need a 360-degree view of media sources in near real-time that reports what's going on. In this work, a multimedia monitoring system has been proposed to manage multimedia information (textual, audio, and video news). The monitoring process is fully described and the automatic processing has been detailed. Also, in order to enhance the video description in our production system, we suggest adding a text recognition tool. The system has been evaluated and given promising results. However, more improvements can be done in the future.

- A Deep Hybrid Model for Advertisement Detection in Broadcast TV and Radio Content: Advertisement detection and classification in electronic media (TV and radio) is an essential part of a media monitoring system and is very useful for companies that work in a competitive environment. Advertisement detection entails a number of difficulties including, unbalanced data, misclassification caused by outliers and variation in loudness levels between TV and radio channels. To overcome these challenges, we propose a Deep Hybrid Model (DHM-ADS) for advertisement detection. We conduct several experiments by combining different methods: deep neural network models (ANN, CNN, and RNN) with dynamic time warping and multi-level deep neural networks such as autoencoders. The evaluation shows that the LSTM autoencoder combined with ANN classifier gives the best result for advertisement detection in TV and radio broadcast.

1.4 Thesis Organization

This thesis is organized as follows. In Chapter 2, we provide an introduction to multimedia content processing techniques, describing segmentation and classification approaches. We focus on reviewing conventional and deep learning methods for data classification. Also, we describe the multimedia processing on the cloud. In Chapter 3, we present a description of the proposed framework for multimedia content classification which is the main framework of our contribution, the architecture is described for each components. In Chapter 4, we extend our studies to deep learning based models for audio content classification. We focus on issues such as outliers detection and unbalanced data. We give solution for each problem. In Chapter 5, we highlight another contribution where we focus on the scalability of our proposed models. We present a distributed deep learning model for a real-time TV and radio monitoring system. In Chapter 6, we list our contributions included in this thesis. In Chapter 7, we give a summary of works done in this thesis followed by a discussion on future work.

CHAPTER II

Background

In this chapter, we introduce multimedia content processing techniques related to our domain. To this end, we provide a state of the art of segmentation and classification methods used for image, audio, and video content. Also, we provide details on cloud architectures for multimedia processing.

2.1 Multimedia Content Segmentation

Multimedia content segmentation is a very important task in computer vision. It consists of breaking up an audio-visual stream into manageable chunks of data, where each chunk shares certain consistent properties [147]. It has become a key technique for semantic content extraction and plays an important role in multimedia content processing [88]. Video content segmentation and categorization can be applied to a number of application areas such as broadcast content indexing, and monitoring broadcast content [118]. In order to perform multimedia content segmentation various low-level audio and/or video based features can be exploited [97].

The main aim of this work is to research an automatic and semantic segmentation solution that extracts news topics and advertisement segments from the multimedia content (TV broadcast). Afterward, the extracted segments will be classified using pre-trained datasets.

A video is a set of temporally ordered images [163] also called frames. The structure of a video consists of frames, shots, and scenes [81]. Figure 2.1 shows the video structure. Image segmentation plays an important role in many applications [44].

It can be formulated as a classification problem of pixels with semantic labels or partitioning of individual objects [109].

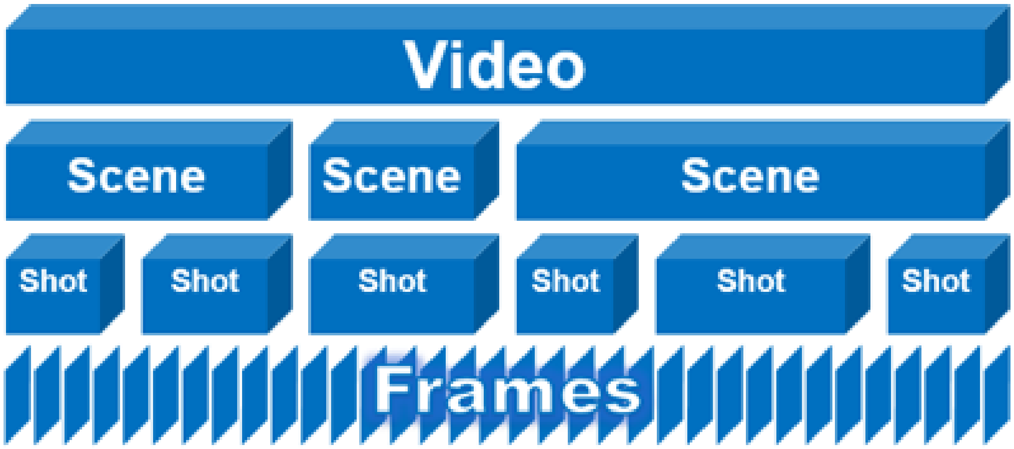


Figure 2.1: Hierarchical structure of video content [146].

A video frame represents the visual perception of an object localized at a specific time [1]. A shot is a set of one or more frames grabbed continually, and these frames symbolize an incessant action in time and space [15]. Shots is considered the elementary units of a video content [112]. A scene is a group of contiguous shots captured from multiple camera angles in one place in a period of time [94].

Automatic video content analysis needs a shot boundary detector used in order to perform a temporal video segmentation. The boundary between two consecutive shots is identified by a hard or soft transition. Hard transition occurs when two successive shots are concatenated directly without special effects. This type of transition is known as a cut or abrupt transition. Soft transition occurs when two shots are combined by utilizing special effects throughout the video production [1].

The previous works concern the usual video segmentation process. However, The TV stream segmentation is different, the video content is composed of a series of heterogeneous programs breaks without markers at the signal level [68]. The segmentation of TV content is called a TV stream macro-segmentation, its objective is to segment the stream into programs. Macro-segmentation algorithms generally rely on detecting inter-programs (IP), which include commercials, trailers, and jingles [83]. Figure 2.2 gives an overview of TV stream segmentation.

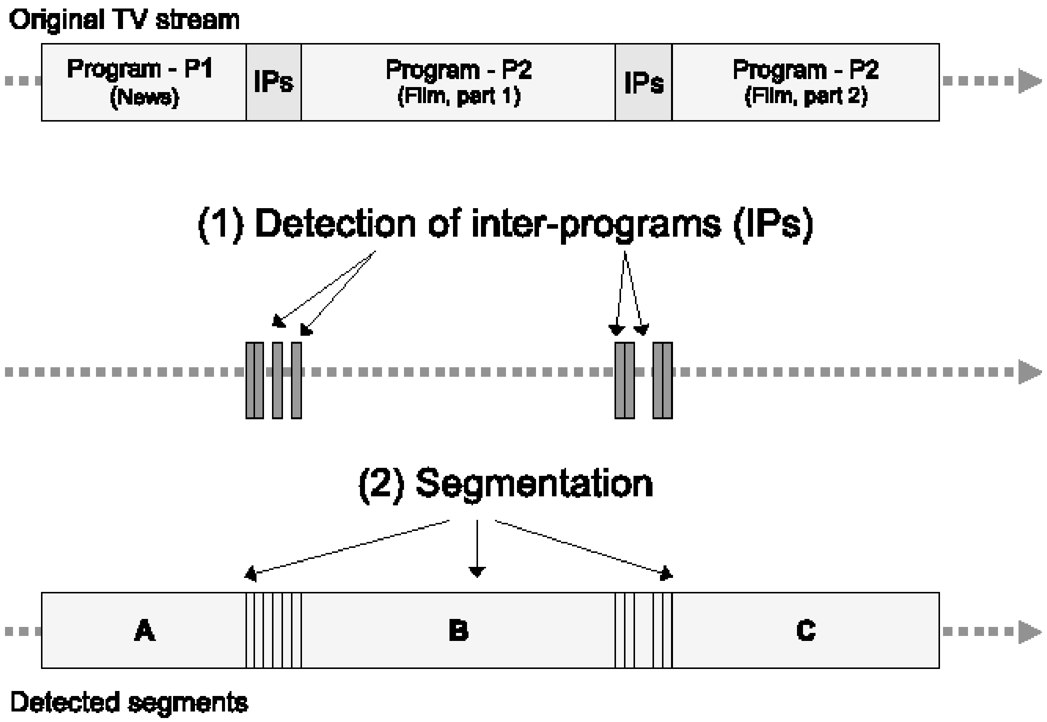


Figure 2.2: Overview of TV stream segmentation [101].

The literature review shows that two main categories of approaches were proposed for tv stream segmentation [68]. The first one is metadata-based and consists in analysis of metadata embeded in TV broadcast such as closed caption, electronic program guide (EPG), and the event information table (EIT). EPG metadata includes information about TV programs such as title, genres, cast, date, and time [144]. The second one is content-based and can be done by searching the program boundaries or detecting breaks between programs.

Among the metadata-based category of methods those who use the EPG metadata [96, 119]. However, the content of EPG is often erroneous and not up to date [76]. More recent works use closed caption text to identify the story boundaries such as in [96, 42] that proposed a deep neural network. Although, the closed caption text suffers due to dynamic scheduling of TV programs [168, 76]. In cases of non-availability of closed caption text, automatic speech recognition (ASR) is used to transcribe the speech associated with video shots [75]. Another constraint on the use of metadata is that laws in most Asian or African countries do not mandate such inclusion of metadata with broadcast [29]. On the other hand, researchers have focused on content-based methods for broadcast segmentation [184].

Content-based methods analyze the audio and video signals in order to discover the high-level structure of the stream [102]. Video segmentation is also called TV broadcast structuring, which consists in automatically determining the boundaries of programs [144]. Berrani & al. [12] classified these methods into two classes: inter-programs based methods and reference database based methods.

Inter-programs based methods use advertisements, trailers, monochrome frames, and scene breaks to determine the boundaries of each program. Pinquier and Andre-Obrecht [122], detect and locate one or many jingles to structure the audio dataflow in program broadcasts. Manson and Berrani [102] have proposed an automatic system for TV broadcast structuring. It is based on studying repeated sequences in the TV stream in order to segment it. Segments are then classified using an inductive logic programming-based technique that makes use of the temporal relationships between segments. Metadata are finally used to label and extract programs using simple overlapping-based criteria.

Reference database based methods use a set of audio/video sequences stored in a database and exploited during the matching process or training phase for predicting of the boundaries. El-Khoury et al. [39] proposed a generic method that chunk any audiovisual content into homogeneous segments and experimented it for shot boundary detection. Zlitni and Mahdi [185] proposed a visual grammar approach for the identification of TV programs. However, the benchmark needs to be up-to-date. In [61], Hmayda et al. proposed an automatic approach to determine the TV stream using a sparse auto-encoder as an extractor of characteristics corresponding to visual jingles for training the classifier of a video category.

In the real world context, some national TV channels don't follow the specification of the domestic video programme delivery. For that, the segmentation algorithms should be adapted to the specificity of each country. Among the recent proposed work, we can mention that published by Raghvendra and Prithwijit [76]. They proposed a modular and scalable framework and software architecture for the broadcast segmentation system for deployment on a computation cluster. This involves scheduler based recording module and broadcast segmentation module. The software was deployed on a cluster of nine desktops and one workstation. The system was used for round the clock processing of three Indian English news channels. The experimentation shows the feasibility of the proposed system for the task of broadcast segmentation.

2.2 Multimedia Content Classification

Multimedia content (Video) classification concentrates on automatically labeling video clips based on their semantic contents like human actions or complex events [165]. Conventional multimedia content classification methods either use text, audio, or image features [128]. However, most methods are based on visual video features, either used alone, or in combination with text or audio features [16]. For the task of multimedia content classification, mostly of research has the intent of classifying an entire video, others authors have focused on classifying segments of video especially for TV stream [182]. The experiments attempt to classify programs by gender, advertisements by advertiser, and distinguish between different news segments within a news broadcast (news topics extraction).

In [16], authors divided the video classification approaches on four groups: text-based approaches, audio-based approaches, visual-based approaches, and multimodal approaches. The text-based approaches exploit two sources of information. the first one identifies text objects viewable in video and use the optical character recognition (OCR) to convert these objects to usable text [57]. The second one uses the speech recognition methods to transcript the speech into text, this generally correspond to the closed captions or subtitles found on bottom of screen in a TV stream.

Audio-based approaches isolate the audio content of the video, followed by the extraction of audio features for classification. They use either time or frequency domain features [33]. Both time and frequency domains are studied and developed for audio fingerprinting tasks. The time domain features are as follows [4]:

- Zero-crossing rate (ZCR), is the number of signal amplitude sign changes per frame. Unvoiced speech has a low volume but a high ZCR [151]. It can be determined by the following formula:

$$ZCR_t = \frac{1}{2}\sum_{n=1}^{N} |sign(x[n]) - sign(x[n-1])| \tag{2.1a}$$

  Where the function is 1 for positive arguments and 0 for negative arguments and is the time domain signal for frame t.

- Short-time energy (STE), is the energy of a short voice signal segment [172] used to estimate the loudness. STE is defined as follows:

$$STE_n = \sum_{n=-\infty}^{\infty} [x(m)w(n-m)]^2 \tag{2.2a}$$

  Where w(n) represent the windowing function and n is the shift in number of samples at which we want to determine the short time energy.

The frequency domain is based on energy distribution across frequency components. The frequency centroid approximates brightness, and is the midpoint of the spectral energy distribution [33]. An audio signal in the time domain can be transformed to the frequency domain using the Fourier transform. Features capture temporal and spectral characteristics of the audio signal and discriminate acoustically different sound types [153]. The following subsections describe the well known audio descriptors.

- Mel-frequency Cepstral Coefficients (MFCC) is derived from the human perceptual system and successfully applied for speech and music recognition. It consists of filtering acoustic signal by non-linear mel-scale triangular filters and computing the Discrete Cosine Transform (DCT) [65]. Steps involved in MFCC are Pre-emphasis, Framing, Windowing, Fast Fourier transform, Mel frequency filter bank, and computing DCT. We can approximate the Mel frequency f by the following formula:

$$Mel(f) = 2595 * \log_{10}(1 + f/700) \tag{2.3a}$$

- Spectral Flux (SF) is a measure of how quickly the magnitude of the power spectrum is changing. It's used for speech or music discrimination in automatic audio classification systems [166]. Spectral flux is defined as the Euclidean distance between successive spectral frames, It can be computed by the following formula:

$$SF(_t) = \left(\sum_k |X(t,k) - X(t-1,k)|^2 \right)^{1/2} \tag{2.4a}$$

Where X(t,k) represents the magnitude of bin number k of frame t.

- Spectral Centroid (SC) is defined as a point in the signal's spectrum with dominant frequencies. A higher centroid values correspond to "brighter" textures with high frequencies [151].

  $$SC_t = \frac{\sum_{n=1}^{N} X_t[n] * n}{\sum_{n=1}^{N} X_t[n]} \tag{2.5a}$$

  $X_t[n]$ is the magnitude of the Fourier transform at frame t and frequency bin n.

- Spectral Roll-Off (SRO) is the measure of skewness of the signals frequency spectrum. It is used to distinguish voiced from unvoiced speech and music [151]. The roll-off R_t is defined as the frequency below which 85% of the accumulated magnitudes of the spectrum is concentrated. The formula is as follows:

  $$\sum_{n=1}^{R_t} |X_t[n]| \leq 0.85 \sum_{n=1}^{N/2} |X_t[n]| \tag{2.6a}$$

For the visual-based approaches, classification is done by analyzing the frames within a video. The features can be extracted from the video or images [80]. Features are designed global or local. Global features are computed from the entirety frame, such as Local Binary Pattern LBP) [114] and Histogram of Oriented Gradients (HOG) [32]. Local features describe the points of interest called keypoints, they are discovered by detectors such as FAST [132], SURF [11], Harris [56], etc. These keypoints will be represented with a feature vector by a descriptor such as SIFT [98], BRIEF [19], and others. Extracted features are then encoded to produce a global descriptors which is able to aggregate all features with bag of words [181] that are used to train a support vector machines classifier to label the video.

LBP is computed as follows: for each pixel c, the 8 neighbours of the center pixel are compared with the pixel c and the neighbours p are assigned a value 1 if p ≥ c.

The equation 2.7 is given in [114]. This process is applied across the whole image.

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{2.7a}$$

Where g_c is the gray value of pixel center and g_p the gray values of the circularly symmetric neighborhood, s is the sign value (1 or 0).

HOG is a global feature and can be used to describe the structure of the object such as people, vehicle, etc. Five steps are required to compute the HOG feature:

1. Preprocess the image by resizing and color normalization
2. Compute the gradient vector of every pixel, as well as its magnitude and direction
3. Divide the image into many 8x8 pixel cells. In each cell, the magnitude values of these 64 cells are binned and cumulatively added into 9 buckets of unsigned direction
4. Slide a 2x2 cells block across the image. For each block, 4 histograms of 4 cells are concatenated into one-dimensional vector of 36 values and then normalized to have an unit weight
5. Concatenate all the bloc vectors

FAST is a corner detection method, which could be used to extract feature points and later used to track and map objects in many computer vision tasks. The detection process[1] is as follows:

1. Select a pixel p in the image which is to be identified as an interest point or not. Let its intensity be I_p
2. Select appropriate threshold value t
3. Consider a circle of 16 pixels around the pixel under test

[1] https://docs.opencv.org/master/df/d0c/tutorial_py_fast.html

4. The pixel p is a corner if there exists a set of n contiguous pixels in the circle (of 16 pixels) which are all brighter than $I_p + t$, or all darker than $I_p - t$

SIFT is an algorithm used to detect and describe local features in images[2]. It locates certain keypoints and then furnishes them with quantitative information called descriptors. It can be used for object recognition. The descriptors are supposed to be invariant against various transformations such as scale and rotation. The algorithm can be decomposed into five steps[3]:

1. Scale-space peak selection: Potential location for finding features
2. Keypoint Localization: Accurately locating the feature keypoints
3. Orientation Assignment: Assigning orientation to keypoints
4. Keypoint descriptor: Describing the keypoints as a high dimensional vector
5. Keypoint Matching

Each modality of features has their advantages and disadvantages. To overcome weaknesses of each, researchers proposed multimodal-based approaches. In [126], Qi et al. use audio, visual, and textual features to classify news streams into genres of news stories. Audio and visual features are utilized to segment and group video shots into scenes. Text is extracted through closed captions using OCR technique. Another work by [160], classify news video into one of ten categories using text features extracted by speech recognition software and combined with the audio features represented by the MFCC combined with others audio features, all these audio features are extracted from one second clips. The classification was done using an SVM classifier.

There are several studies that have attempted multimedia content classification using different modalities of signals. These methods can be classified into three categories: fingerprint-based methods, distance-based methods, and deep learning methods. We review briefly these methods as follows:

[2] https://en.wikipedia.org/wiki/Scale-invariant_feature_transform

[3] https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3

2.2.1 Fingerprint-based Methods:

Fingerprinting is known as perceptual hashing or content-based media identification, is receiving increased attention [73]. Deriving compact signatures from complex multimedia objects is an essential step in Multimedia Information Retrieval. Fingerprinting can extract information from the audio signal at different abstraction levels, from low level descriptors to higher level descriptors [21]. Perceptual hashing techniques can be used for both image and audio identification.

Automatic identification of ads content within TV streams is a classification problem and can be resolved using a compact signature and a distance metric. Generally, this technique is used to detect repeated objects such as identify the presence or absence of an ads content [60]. Two signatures can be used: audio fingerprint [14, 141] and perceptual hashing [154]. The audio signatures are used to efficiently process the audio stream (Radio) and audio-visual streams (TV). This approach assumes that repeated objects contain both same visual and same audio information [90]. Other works use the video frame information and the audio change information to resolve this issue such as proposed in [37].

Perceptual hashing use a cryptographic function that can be categorized into unkeyed hash functions and keyed hash functions [106]. An unkeyed hash function H generates a hash value h from an arbitrary input x. A keyed hash function generates a hash value h from an arbitrary input x and a secret key k. Keyed hash functions are also called message authentication codes. Perceptual hash functions can be used for many applications such as image spam detection, searching the internet for copyright violations or maintaining databases of illegal content, etc.

Cano [20], in his dissertation which deals with the issue of content based audio search, he has proposed several audio fingerprinting frameworks related to the usage modes such as identification of recordings, integrity verification (detection alteration of data), Watermarking support (verify the authenticity), and content-based audio retrieval and processing. He concludes that the different audio fingerprinting systems can be explained with respect to a general fingerprinting framework.

Wu and Satoh [164] proposed a fast and unsupervised system based on exact-duplicate matching detected and localized TV commercials in a video stream, clustered the exact duplicates, and detected duplicate exact-duplicate clusters across video streams. The used algorithm is based on a new bag-of-fingerprints model. It is robust against decoding errors. Testing using ten-hour, one-month, and five-year video streams demonstrated the effectiveness and efficiency of this system.

Vega et al. [154] performed a benchmarking process of the four most know, available, and free perceptual image hashing algorithms to determine their efficiency. The evaluation results suggested that employing just a sample frame to identify a video is possible, due that the obtained success rate was upper than 98.22% using perceptual hashing (PHASH) with hamming distance of 6.

2.2.2 Distance-based Methods:

The simplest way to measure similarity between audio sequences is to use a distance such as the euclidean distance [10], mean distance [87], edit distance [27], and dynamic time warping [79]. To compare two perceptual hashes appropriate measures must be used. The most used are the hamming distance [55], the bit error rate [169], and the peak of cross correlation [155].

The hamming distance measures the difference of two strings that can be binary coded numbers, numbers, or alphabets. a XOR operation can be used to calculate the hamming distance for binary coded numbers. The bit error rate is defined as the rate at which errors occur in a transmission system and can be directly translated into the number of errors that occur in a string of stated number of bits [156]. The peak of cross correlation correspond to a function that determines how much the two signals are shifted each others.

Most time series data mining algorithms require similarity comparisons as a subroutine [127], and despite consideration of dozens alternatives, there is increasing evidence that the dynamic time warping is the best measure in most domains [35].

2.2.3 Deep Learning Methods:

Several studies have shown promising results by applying machine learning classifiers to various audio and visual features [17]. Recent advances in deep learning for image [135] and speech [51] domains have motivated techniques to learn robust video feature representations to effectively exploit abundant multimodal clues in video data [165]. In this subsection we review three categories of deep learning algorithms for video/multimedia classification that are supervised, semi-supervised and unsupervised deep learning.

Neural Networks are multi-layer networks of neurons consisting of nodes which are used for classification and prediction of data provided as input to the network. There is an input layer, one or many hidden layers and an output layer[139]. All the layers have nodes and each node has a weight which is considered while processing information from one layer to the next layer. Each neuron includes a bias and an activation function. Figure 2.3 shows a simple neural network with five inputs, 5 outputs, and two hidden layers of neurons[4].

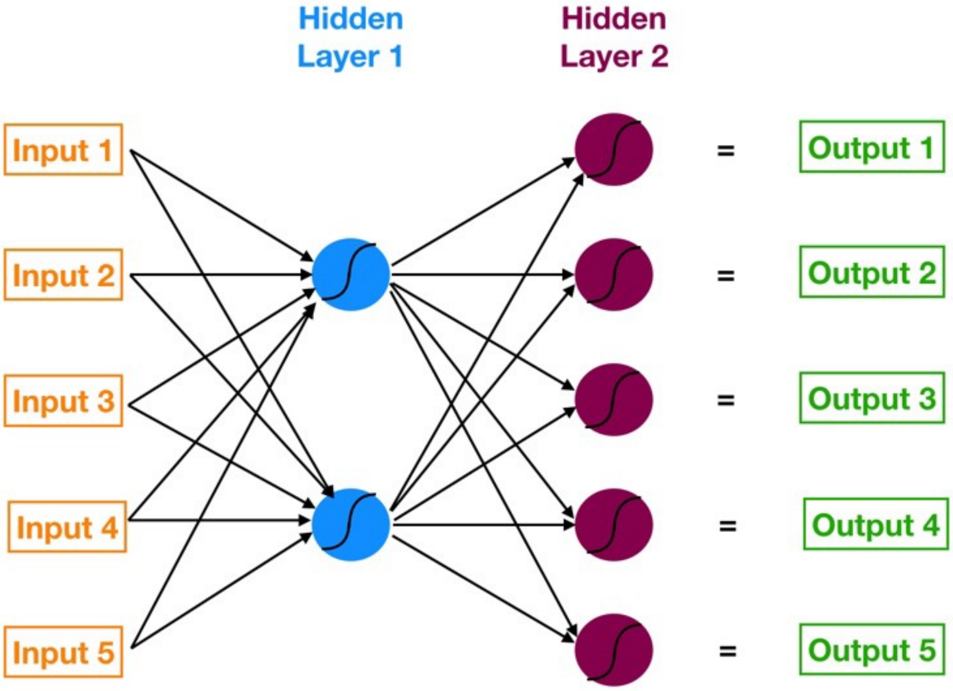


Figure 2.3: Neural network with two hidden layers.

In Neural networks, activation functions are a crucial component. They determine the output of a deep learning model, its accuracy, and also the computational efficiency of training a model. Generally, activation functions can be divided into linear and non-linear [128]. The most popular activation functions are Sigmoid, TanH, ReLU, and Softmax.

[4] https://towardsdatascience.com/understanding-neural-networks-19020b758230

The sigmoid function is a non-linear activation function in the field of neural network. It projects the input into the range of 0 to 1 [18]. The mathematical expression of this function is as follows:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.8a}$$

Where x is the input and f is the output, and e is the standard exponential. The hyperbolic tangent (TanH) activation function is also like logistic sigmoid but the output is from the range of -1 to 1. It's mainly used for binary classification, and we can compute it by the following equation:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{2.9a}$$

Where x is the input and f is the output, and e is the standard exponential. The rectified linear units function (ReLU) is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It's defined by the following function:

$$f(x) = max(0, x) \tag{2.10a}$$

Where x is the input and f is the output. The softmax function is the best choice for a multi-class problems [18]. It calculates the probability distribution over N different events. The mathematical formulation of this function is as follows:

$$F(e^{X_i}) = \frac{e^{X_i}}{\sum_{j=1}^{K} e^{X_j}}, i = 1, 2, ..K \tag{2.11a}$$

Where X is the input vector to the softmax function, e is the standard exponential function, and K is the number of classes in the multi-class classifier.

The activation functions have the capability to improve the learning of the patterns in data thereby automating the process of features detection and justifying their use in the hidden layers of the neural networks, and usefulness for classification purposes across domains [113].

2.2.3.1 Supervised Learning

With deep learning models, a video clip is processed as a collection of frames, and then for each frame, feature representation could be derived by running a feed-forward pass till a certain fully-connected layer with state-of-the-art deep models pre-trained on different datasets such as ImageNet [34, 84], VGGNet [140], GoogLeNet [148], and ResNet [58].

A convolutional neural network (CNN) is a class of deep neural models, commonly applied in computer vision tasks such as image recognition, object-detection, and video classification [149, 77]. CNN model operate in a feed-forward manner, computation flows from the input layer to the output layer. Mona Ramadan [128] presented the topology of a CNN with the main building blocks like shown in Figure 2.4.

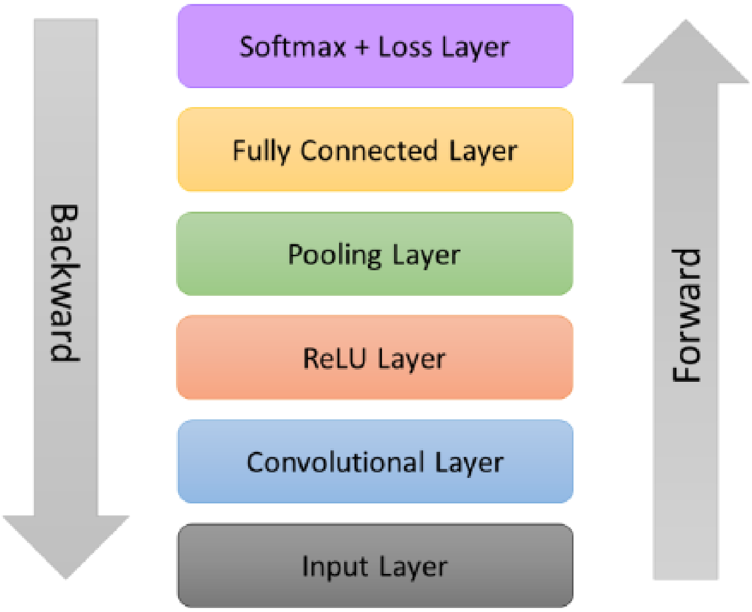


Figure 2.4: CNN topology with the main building blocks.

AlexNet is a CNN model designed by Alex Krizhevsky and competed in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) on September 2012. The model is shown in Figure 2.5, it consists of five convolutional layers, some of which are followed by max-pooling layers, and two globally connected layers with a final 1000-way softmax. On test data, the model achieved top-1 and top-5 error rates of 39.7% and 18.9% which is considerably better than the previous state-of-the-art results [84].

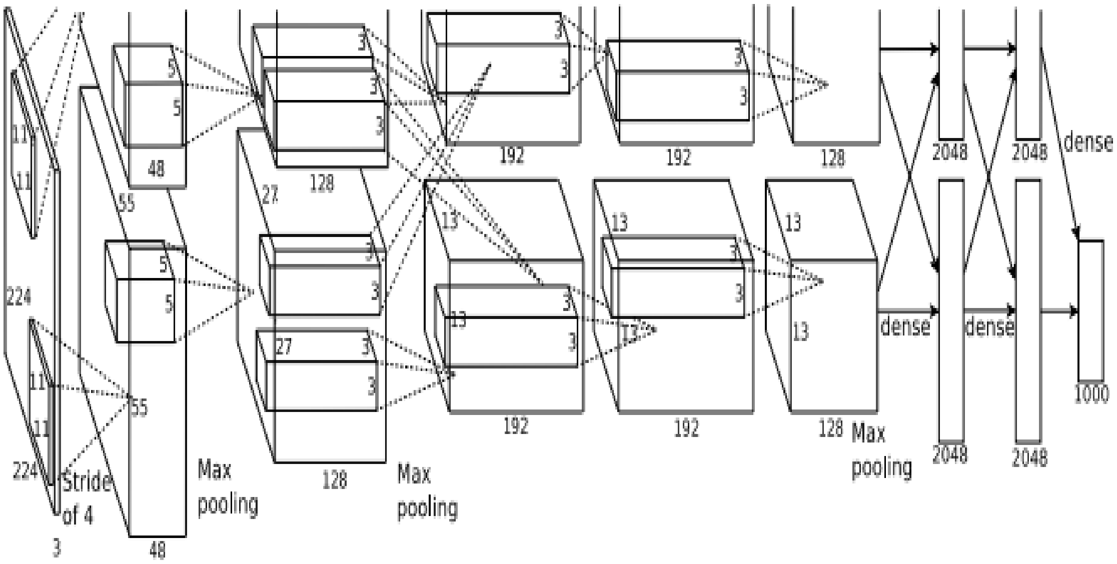


Figure 2.5: Architecture of AlexNet.

VGGNet is another CNN model proposed by Karen Simonyan and Andrew Zisserman from University of Oxford (Visual Geometry Group) in 2014 [140]. The macroarchitecture of VGGNet can be seen in Figure 2.6. Performance achieved at a single test scale are top-1 error of 25.5% and top-5 error of 8.0%. At multiple test scales, VGG got a top-1 error of 24.8% and a top-5 error of 7.5%. It achieved second place in the 2014 ImageNet competition with its top-5 error of 7.3%.

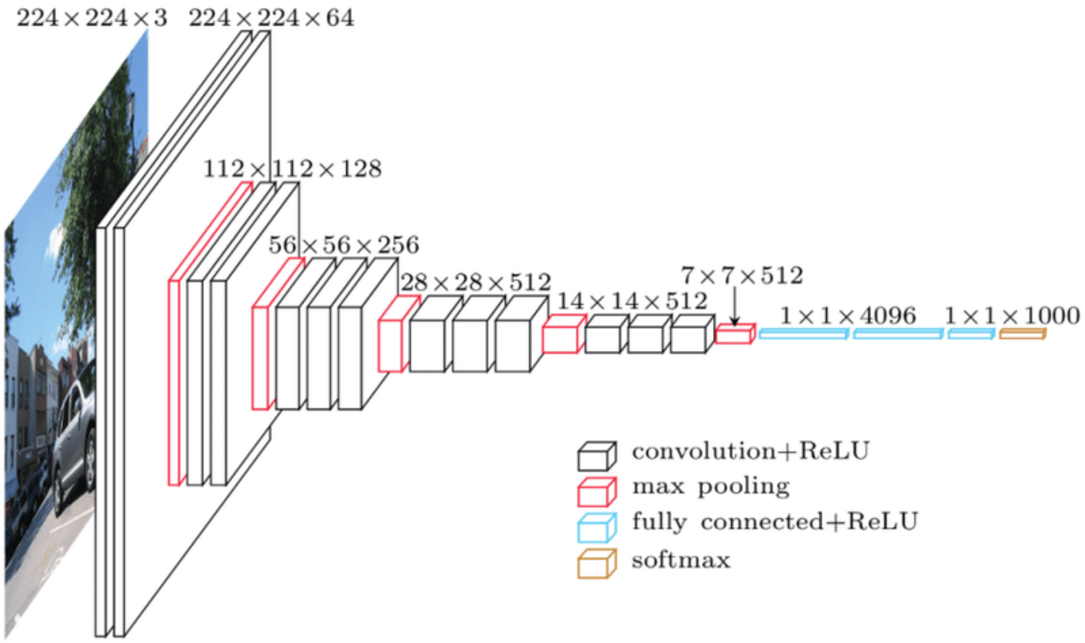
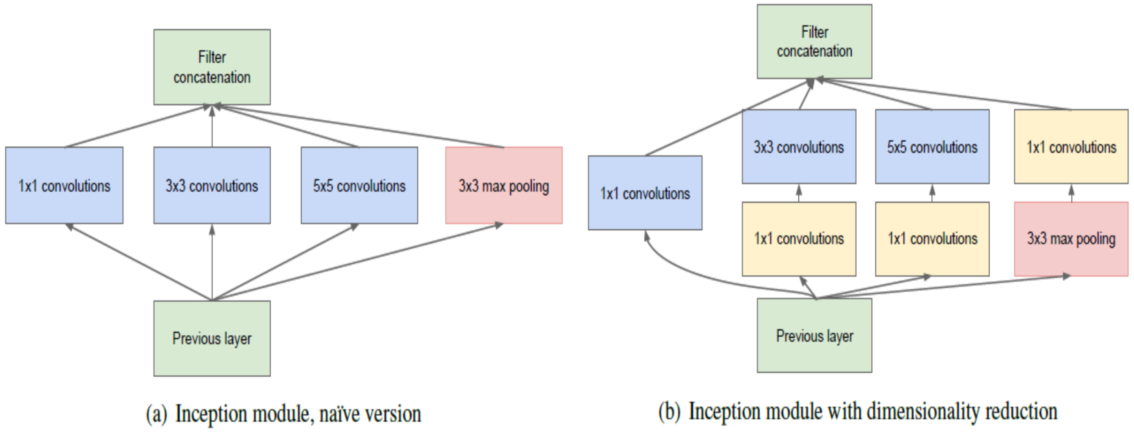


Figure 2.6: Architecture of VGGNet.

GoogLeNet from Google and is composed of 22 layers deep network and uses 9 inception modules, Szegedy et al. [148] introduced the Inception module inspired from Hebbian principle (neurons that fire together wire together). A schematic view of the inception module [148] is depicted in Figure 2.6. It was the winner of the ILSVRC 2014 competition and achieved a top-5 error rate of 6.67% in classification task.



(a) Inception module, naïve version

(b) Inception module with dimensionality reduction

Figure 2.7: Inception module used in GoogLeNet.

ResNet (residual network) proposed by Kaiming He et al. [58] from Microsoft

Research, they introduced the concept called Residual Network to solve the problem of the exploding gradient. The ResNet architecture is composed of a stack residual blocks where every residual block has two 3x3 convolutional layers, Figure 2.8 shows the residual block[5]. The Network consists of up to 152 layers by learning the residual representation functions instead of learning the signal representation. It was the winner of ILSVRC 2015 in image classification, detection, and localization.

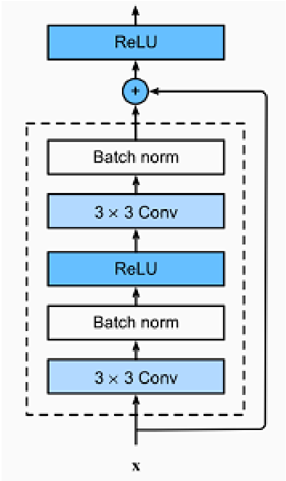


Figure 2.8: ResNet block illustration.

There are some other variants of CNN models that have been introduced. Although instead of CNN, recurrent neural network (RNN) becomes a much more suitable choice for times series [64]. Some of specially designed RNN cells, like Long-Short Term Memory (LSTM) [62], have achieved impressive performance gains on several natural language processing (NLP) tasks and on movie genre classification [40]. There are two popular topology of RNN, Elman and Jordan networks (see the following figure 2.9). In the Elman network, the hidden layer feeds a state layer of context nodes that retain memory of past inputs. A single set of context nodes exists that maintains memory of the prior hidden layer result. For the Jordan network, It's different in that instead of maintaining history of the hidden layer they store the output layer into the state layer [100].

[5] http://d2l.ai/chapter_convolutional-modern/resnet.html

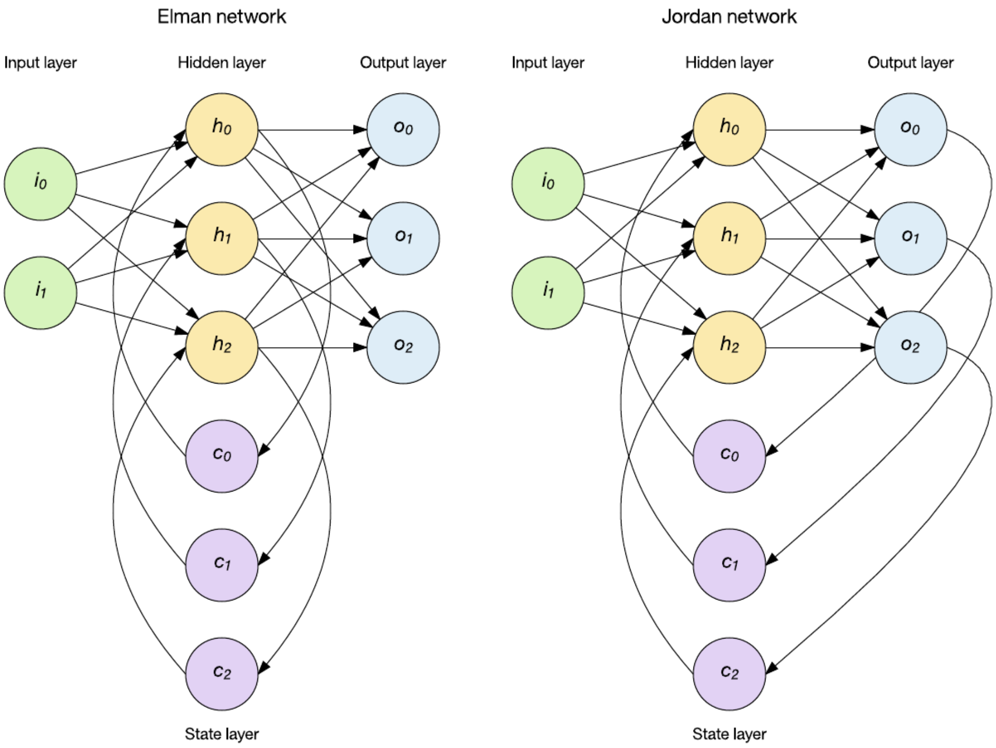


Figure 2.9: Simple RNN topology - Elman & Jordan.

The two networks can be trained through standard back-propagation, and each has been applied to sequence recognition and natural language processing [100].

2.2.3.2 Semi-Supervised Learning

Semi-supervised learning is a method that combines both labeled and unlabeled data to train the network [111]. Recently, semi-supervised methods focused on finding latent representations of the input data for features extraction. A prominent example of this is the autoencoder. Autoencoders attempt to find a lower-dimensional representation of the input space without sacrificing substantial amounts of information [152]. The most popular autoencoder introduced by Vincent et al. [157] is the denoising autoencoder that is trained on noisy versions of the input data, penalizing the reconstruction error of the reconstructions against the noiseless originals.

Jing et al. [71] proposed a semi-supervised learning approach for video classification using CNN. This approach trains network from a small number of labeled examples and exploits two regulatory signals from unlabeled data. The first signal is the pseudo-labels of unlabeled examples computed from the confidences of the CNN being trained. The other is the normalized probabilities, as predicted by an image classifier CNN, that captures the information about appearances of the interesting objects in the video. Authors show that, under the supervision of these guiding signals from unlabeled examples, a video classification CNN can achieve impressive performances utilizing a small fraction of annotated examples.

Facebook AI team is developing a new model training technique called semi-weak supervision that is a new way to combine the merits of two different training methods: semi-supervised learning and weakly supervised learning. It opens the door to creating more accurate, efficient production classification models by using a teacher-student model training paradigm and billion-scale weakly supervised data sets [66].

2.2.3.3 Unsupervised Learning

In supervised learning methods, scaling up a high number of classes requires an insurmountable annotation efforts. Therefore the use of unsupervised learning is a promising way to overcome this issue. Srivastava et al. [145] proposed an encoder-decoder based on LSTM to learn feature representations in an unsupervised way and pre-trained on YouTube data without manual labels, and then fine-tuned on standard benchmarks to recognize actions. They conclude that their model give any significant improvements in terms of classification accuracy after finetuning, however they did give slightly lower prediction error.

In [131], authors proposed a multi-view unsupervised deep learning methodology for novelty-based highlight detection. The method jointly analyses both game footage and social signals such as the players facial expressions and speech, and shows promising results for generating highlights on streams of popular games such as Player Unknown's Battlegrounds.

2.2.3.4 Imbalanced Data

In a classification task with supervised learning, researchers may be facing a problem of imbalanced classes in the training data. It means that some classes have a lower number of samples than others. Effective classification with imbalanced data is an important area of research [72]. There are two main methods addressing class imbalance include undersampling and oversampling, they modify the training distributions in order to decrease the level of imbalance.

Undersampling method discards observations (data) from some class, reducing the total amount of information that the model has to learn from. Unfortunately, there is a high possibility that the deleted data may contain important information about the predictive class. Oversampling method increase the number of observations which are just copies of existing samples. The oversampling may cause an overfitting of the training data.

A variety of intelligent sampling methods for deep learning models have been developed in an attempt to balance these trade-offs. Hensman and Masko [59] show that oversampling is a viable way to counter the impact of imbalances in the training data. Lee et al. [85] incorporate transfer learning by pre-training CNN with class-normalized data and fine-tuning with original data, and showed superior classification accuracy. Pouyanfar et al. [124] introduce a new dynamic sampling technique that adjusts the class distribution of the training samples according to class-wise performance.

2.2.3.5 Outliers detection

Deep learning-based models are used to overcome some of the limitations of the hand-crafted features [108], but in the context of electronic media, they suffer from the presence of outliers and that leads to misclassification. A common problem that researchers face when analyzing real-world datasets is determining which instances of the processed data stand out as being dissimilar to the trained data. Such instances are known as outliers or anomalies.

Chalapathy et al. [24] proposed a one-class neural network (OC-NN) model to detect anomalies in complex datasets. To train their network, they use a loss function inferred from a one-class SVM (OC-SVM) that was proposed by Scholkopf & Smola [137]. The autoencoder is a fundamental deep learning approach to anomaly detection [107]. A typical autoencoder network includes two phases: an encoder that transforms the input data into a lower dimensional representation and a decoder, that tries to reconstruct the original input data [53].

The autoencoder is trained by minimizing the reconstruction error between input and output data [78]. Figure 2.10 shows an autoencoder neuronal network [167].

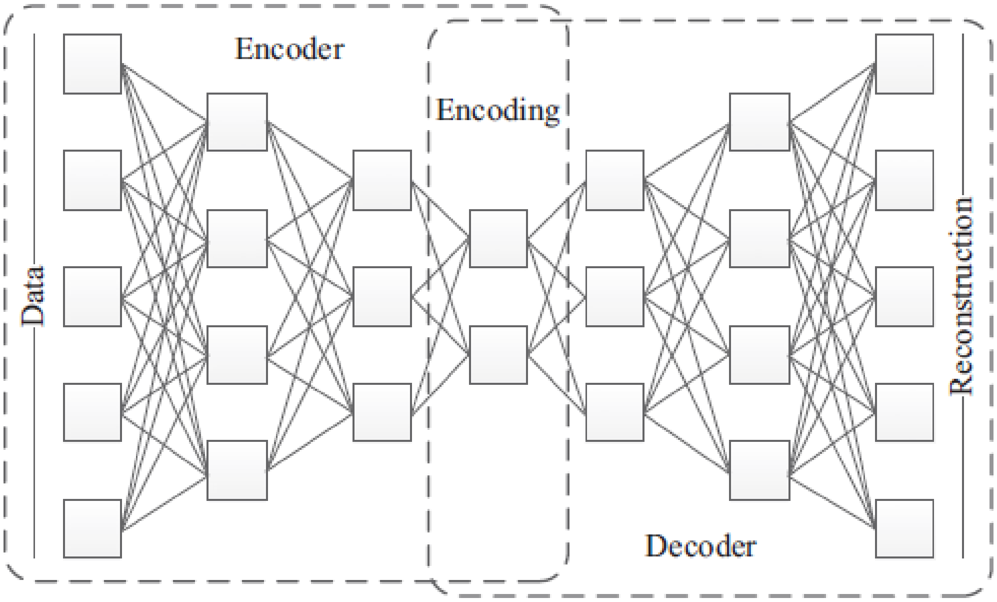


Figure 2.10: An architecture of autoencoder neural network.

Mathematically an autoencoder can be represented by the following functions [78]:

$$f_1 : X \rightarrow \hat{X} \tag{2.12a}$$

$$f2 : min(X, \hat{X}) \tag{2.12b}$$

Where X is the input data, $\hat{X}$ is the reconstructed data, f_1 denotes the prediction $\hat{X}$, and f_2 denotes the minimized loss function.

2.3 Multimedia Processing in Cloud

The amount of multimedia data delivered by media such as television is increasing every day. This considerable size of video data poses significant challenges for video management and mining systems that require powerful machines to deal with large-scale video data [5]. An efficient solution is necessary to store and analyze this large volume of video data for extracting information. Designing such a system and providing its required computing power are challenging using traditional computing paradigms [115]. Recently, researchers have been attracted by the efficient services provided by the cloud frameworks.

2.3.1 Cloud Architecture

In this sub-section, we present an overview of the cloud architecture and cloud services. Figure 2.11 shows the cloud computing reference architecture designed by the NIST[6] agency. It defines five major actors: cloud consumer, cloud provider, cloud carrier, cloud auditor and cloud broker.

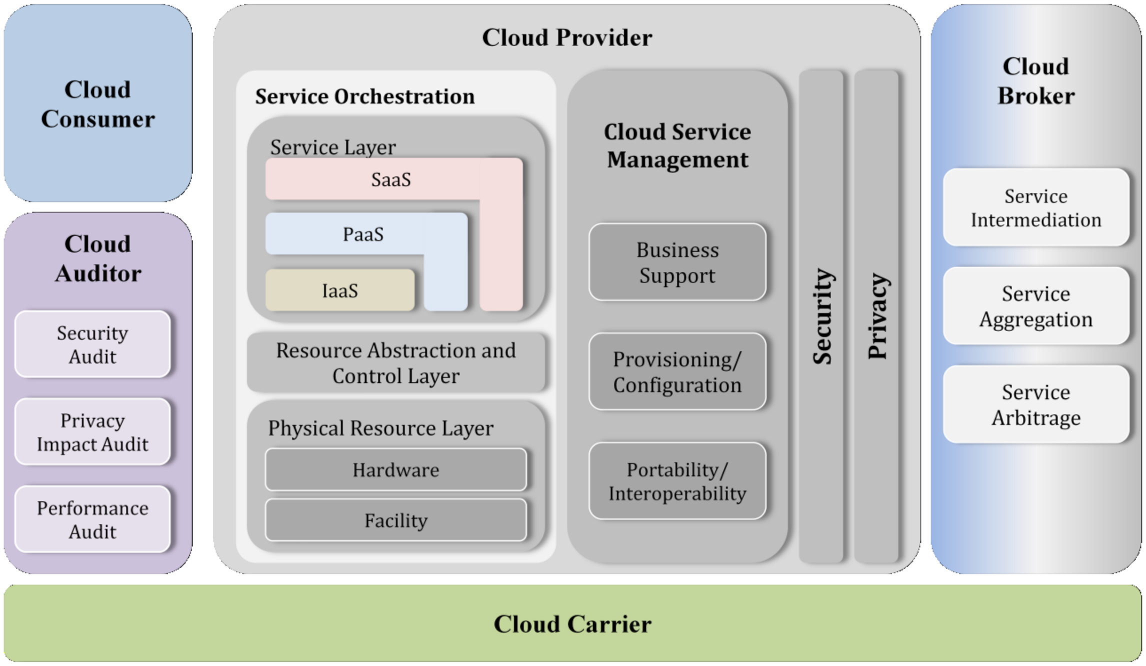


Figure 2.11: Cloud architecture [95].

Where each actor is an entity (a person or an organization) that participates in a transaction or process and/or performs tasks in cloud computing [95]. The five actors of cloud computing are defined as follows:

- Cloud Consumer: A person or organization that maintains a business relationship with, and uses service from, Cloud Providers.
- Cloud Provider: A person, organization, or entity responsible for making a service available to interested parties.

[6] National Institute of Standards and Technology, a US Federal government agency responsible for developing technology standards and guidelines

- Cloud Auditor: A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.

- Cloud Broker: An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers.

- Cloud Carrier: An intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers.

The cloud model defined by the NIST agency is composed of five essential characteristics, which are:

- On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

- Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

- Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

- Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

- Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability10 at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, active user accounts). Resource usage can be monitored, controlled, audited, and reported, providing transparency for both the provider and consumer of the utilized service.

There are three models of cloud computing: public, private, and hybrid clouds. Public cloud models offer their services to all users over the internet. Private cloud models are dedicated to one organization or business, and often have much more specific security controls than a public cloud. Hybrid cloud models are a blend of public and private clouds. This is a more complex cloud model in that the organization must manage their ressources. Cloud computing services fall into three main categories: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS).

IaaS provides virtual infrastructure as well as actual hardware for managing the storage, virtual machines, and virtual network over the Internet. In PaaS, The cloud provider manages and delivers programming languages, frameworks, libraries, services, and tools for customers to create and deploy applications. SaaS provides a complete software solution on a pay-as-you-go basis [115].

2.3.2 Video Analytics in Cloud

There are many techniques proposed for video analytics in cloud infrastructure, we review some of them:

Tariq Abdullah et al. [2] designed a framework for stream processing in clouds capable of detecting vehicles from the recorded video streams. The framework was developed on a GPU cluster with two GPUs and the results showed performance gain of 14 times when compared with one human operator doing the same analysis. Also, a CPU implementation of the framework yielded 4 times of analysis time improvement.

Ashiq Anjum et al. [7] proposed a cloud based video analytics framework for scalable and robust analysis of video streams. The framework empowers an operator by automating the object detection and classification process from recorded video streams. An operator only specifies an analysis criteria and duration of video streams to analyse. The streams are then fetched from a cloud storage, decoded and analysed on the cloud. The framework executes compute intensive parts of the analysis to GPU powered servers in the cloud.

Yaseen et al. [174] proposed a cloud-based video analytics system that use convolutional neural networks whose parameters are optimally tuned for accurate classification of objects from video streams. The system learns features from large amounts of input data by performing training in parallel on a multi-node cluster. The proposed system proved to be robust to classification errors with an accuracy and precision of 97% and 96% respectively. Several factors contributed to achieve high accuracy such as optimal selection of learning rate, regularization, normalization and optimization algorithms.

Lin Feng-Cheng et al. [93] proposed a cloud-based face video retrieval system with deep learning. A dataset is collected and pre-processed. Blurry images are removed, and face alignment is implemented on the remaining images. The dataset is constructed and used to pre-train the CNN models (VGGFace, ArcFace, and FaceNet) for face recognition. The results of these three models are compared and the most efficient one was choosed to develop the system.

Daniel Pop [123] investigated how cloud computing paradigm impacted the field of machine learning. He gives a review of popular statistics tools and libraries deployed in the cloud. After, he listed existing tools that allow users to create a Hadoop cluster in the cloud and run jobs on it. Also, he presented libraries of distributed implementations for machine learning algorithms. Lastly, he gives a survey of machine learning as Software-as-a-Service. Daniel Pop synthesized the literature as follows:

- Existing programming paradigms for expressing large-scale parallelism such as MapReduce and the Message Passing Interface are defacto choices for implementing machine learning algorithms.

- Machine Learning in distributed environments come in different approaches, offering viable and cost effective alternatives to traditional machine learning and statistical applications, which are not focused on distributed environments.

- Existing solutions target either experienced, skilled computer scientists, mathematicians, statisticians or novice users who are happy with no (or few) possibilities to tune the algorithms. Ens-user support and guidance is largely missing from existing distributed machine learning solutions.

In [28], Tse-Shih Chen et al. proposed a Platform-as-a-Service architecture to conduct large scale video analysis in more systematic and efficient ways. They presented the construction and implementation steps of a real-world video analysis service that have successfully integrated over 25 thousands of surveillance devices throughout the city and have proven that their solution can dramatically reduce human resource needed for video scanning.

Recently, many applications such as surveillance and traffic control that process a huge amounts of data generated by cameras have migrated to edge computing for his ability to perform real-time analysis of videos. Jianguo Chen et al. [26] proposed a distributed intelligent video surveillance system that uses deep learning algorithms and deployed it in an edge computing environment. The system can migrate computing workloads from the network center to network edges to reduce huge network communication overhead and provide low-latency and accurate video analysis solutions. Johan Barthélemy et al. [8] designed a framework based on edge-computing that collect, store and access the data. It uses computer vision and deep neural networks to track in real-time multi-modal transportation while ensuring citizens' privacy.

2.3.3 Chapter summary

Multimedia content processing includes segmentation and classification of large video data captured from broadcast stream. In this chapter, the existing works related to video and TV stream segmentation are reviewed. The taxonomy of existing methods for video content classification have been explored, starting from conventional methods to deep learning methods which have been detailed such as supervised, semi-supervised and unsupervised methods. Also some issues related to the deep learning models have been pointed like the problem of imbalanced data and the presence of outliers in the data.

Processing large amount of data involves the use of a platform for big-data analysis. The suitable environment is the cloud computing, it provides a distributed resources and use efficient methods in the processing of data. Some works have been proposed for video analytics on cloud computing have been studied and the last are related to the edge computing that overcome some issues encountered in using the cloud computing.

In the next sections, we will give a preview of our main results and state the contributions of the thesis.

CHAPTER III

Real time TV Content Analysis for Multimedia Monitoring System

In this chapter, we present the multimedia content classification framework that we have designed to process the TV stream and can be used for media monitoring. The purpose of this framework is to tackle both news extraction and advertisements identification. The chapter is organized as follows: First, we introduce the media monitoring domain and related work. Then, we give a detailed description of each block that compose our multimedia monitoring system.

3.1 Introduction

Media monitoring is the process of reading, watching or listening to the editorial content of media sources on a continuing basis [30]. Media monitoring systems are forcing themselves to automate more and more the process of collecting and analyzing news information, while some solutions combine the automatic information gathering from one side and the experts analysis in the field of information and communication sciences in the other. We are mainly interested in relieving experts as much as possible of the automated tasks that are part of the multimedia watch process, namely: monitoring, collecting and disseminating information. The process of media monitoring (news and advertisements) is depicted in Figure 3.1.

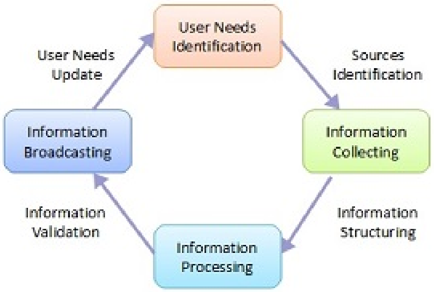


Figure 3.1: Information monitoring process

Through media monitoring, companies aim to find information about competitors and specific issues relevant to their operation. Whereas the old-fashioned press clipping services required 2 to 3 weeks to deliver clips, online media monitoring services deliver clips overnight as a standard service and usually offer near real time delivery [30]. The set of clips are summarized and delivered by e-mail in text or HTML format accompanied with PDF files generated from clips and sent via FTP. This enables executives in companies to stay up-to-date with a comprehensive overview of their reputation.

The providers of information operate on various media of communication: newspapers, TV, radio, web, and social networks. The number of media sources is still growing which makes it necessary to have a platform that monitors, processes, and distributes daily press reviews for companies according to their areas of interest. Multimedia data daily generated by television channels is of the order of gigabytes which urges us to exploit the power of the Big Data frameworks for storing, processing, and analyzing these data.

The media monitoring system provides companies near real-time reports on all aspects of compliance and competitor activity. Video advertisements (TV commercials) have become an indispensable tool for marketing. Advertisement detection is a classification problem with various training datasets where every class may have a different audio length which varies between 3 and 60 seconds. Some recent research has started to focus on the classification of imbalanced data since real-world data is often skewed [13]. Deep learning-based models are used to overcome some of the limitations of the hand-crafted features [108].

In this work, we present a global architecture of our Multimedia Monitoring System (MMS), detailing some development of specific tools for collecting and processing multimedia news extracted from the television streams of local channels. They include distributed streams acquisition and storage, stream segmentation, content analysis, Arabic text recognition, and selective broadcast of the press review to customers. Monitoring news includes newspaper, online news, broadcast news, and social media. We focus on processing ten public and private TV channels where news content is extracted automatically and merged by subject, then analyzed and summarized by specialized users before the step of delivery to the customers.

3.2 Related Work

In the literature, the problem of extracting and analyzing information from television stream is processed in three stages which are: stream structuring into programs, news program identification, and news topics segmentation. The television stream structuring also called “program extraction” requires a macro-segmentation of the TV stream. Macro-segmentation algorithms generally rely on detecting interprograms (IP) which include commercials, trailers, and jingles as in [83]. Pinquier and Andre-Obrecht, detect and locate one or many jingles to structure the audio dataflow in program broadcasts [122]. The authors of [102] proposed an automatic system for TV broadcast structuring. It is based on studying repeated sequences in the TV stream in order to segment it. Segments are then classified using an inductive logic programming-based technique that makes use of the temporal relationships between segments. Metadata are finally used to label and extract programs using simple overlapping-based criteria. Ramires et al. [129] proposed an approach that uses only audio features and centers on the detection of short silences that exist at the boundaries between programs and advertisements.

Wang et al. [159] proposed a multimodal representation of individual programs by using program-oriented informative images, key frames, and textual keywords in a summarization manner for program segmentation in broadcast video streams. Unsupervised TV program structuring is another work proposed in [3], the idea is to automatically recover the original structure of the program by finding the start time of each part composing it. It is based on the detection of separators which are short audio/visual sequences that delimit the different parts of a program.

Several works were proposed for news program identification, among them Zlitni et al. [184] suggested an approach based on video grammar to identify the programs in TV stream and deduce their internal structure. Li et al. [?] proposed a program segmentation system in news broadcast and have used the online electronic program guides (EPG) and closed caption text in TV news. In a recent work, Kannao et al. [74] proposed a two-stage approach to classify news video segments. First, broadcast video shots are classified with multiple labels based on a set of audiovisual features. Second, sequences of these shot features are modeled to detect news programs. Another motivation of program identification is TV commercial detection, as the advertisers spend much money, it is necessary to verify their commercials are broadcasted as contracted. A system for TV commercial monitoring is desired [37].

The topics segmentation for news program has been studied by many researchers such as in [184], [177]. It has processed in two techniques: based on audio transcription methods and based on text detection methods. Text detection methods in the literature can be grouped into texture-based, connected component-based and hybrid methods. Texture-based algorithms scan the image using generally multi-scale sliding windows to extract different texture proprieties and classify image areas as text or non-text based on texture-like features [177].

Many multimedia content analysis systems often require real-time processing and scalability to deliver and store videos. To achieve scalability, programmers need to use distributed programming frameworks such as Hadoop/MapReduce. [82] proposed a high-speed and scalable video server system using PC-cluster for video content delivery. Regarding on development tools, [48] proposed PyCASP, a Python-based content analysis parallelization framework. That is designed using a systematic, pattern-oriented approach with the goal of making it modular, comprehensive and applicable to a wide range of multimedia content analysis applications.

In general, TV broadcast processing can be performed using either the metadata associated with the stream or by directly analyzing the audiovisual stream [12]. Metadata stream like closed caption, electronic program guide, event information table and teletext provides specific textual information that describes the audiovisual streams. Manson et al. [101] proposed a metadata-based system that automatically structures TV stream, the system first detects inter-programs as repeated sequences in the broadcasted stream, and then deduce the boundaries of programs. In audiovisual-based systems, some solutions have been proposed that use speech transcription to extract information like proposed by [45].

The embedded text in videos is one of the most relevant sources of high-level semantic information [176] that are used in videos indexing and searching. In [179], Zhang proposed a novel unsupervised method to detect and localize the text objects occurring in image and video documents based on a text model, character features and a tracking method to track text event in video documents. Optical Character Recognition systems (OCR) have been widely developed in past years for different languages and addressing specially scanned documents, also used for TV news video indexing.

Smith et al. [142] described the development of a system for learning of semantic concepts in broadcast video and report on experimental results showing effective automatic detection of semantic concepts in the domain of broadcast news video. Among the works dealing with semantics, [6] proposed a multimedia retrieval system that uses low level information and textual information in indexing process. However semantic concepts alones are not useful for executives in client companies, they needs to be informed by only the most important news that have been analyzed and summarized by experts in communication science. There are another category of works in broadcast news that concerns person identification, such as proposed by Rouvier et al. [133] and who uses scene understanding in order to improve unsupervised people identification.

Escalada et al. [41] proposed an automatic indexing and aggregation web based system (NewsClipping) for the news domain that manages multimedia information without any manual annotation, but their application use metadata associated to the TV stream for indexing content. Some online news service such as CyberAlert[1] monitor the closed caption text of TV stations and deliver a text file of all clips. Y. Aggoun proposed a media monitoring process and reporting for decision-makers [175].

Finally, we conclude that most of the cited methods are based on metadata provided by the broadcasters. In our context, local TV channels don't use metadata and

[1] www.cyberalert.com

black frames in the broadcasting process. Our contributions are the following:

- A new method for TV stream segmentation,
- A classifier for news program identification
- A new process fo news topics segmentation
- A deep learning model for advertisements classification

Next, we will describe a global architecture of a multimedia monitoring system, and experiment different modules.

3.3 Proposed System

The media monitoring systems must provide 360-degree view of media sources in real-time and coverage 24/7 support to the clients. Based on the companies needs and according to the context of local media in Algeria, we propose a semi-automatic Multimedia Monitoring System (MMS) that processes newspaper, online news, broadcast news, and social networks. It combines the thoroughness of automated news clipping with the accuracy and judgment of human readers. The system must deliver daily alerts to companies, reports and periodically synthesis of all media, news have to be extracted, stored, summarized and prepared for delivery. Our proposed system is described in Figure 3.2.

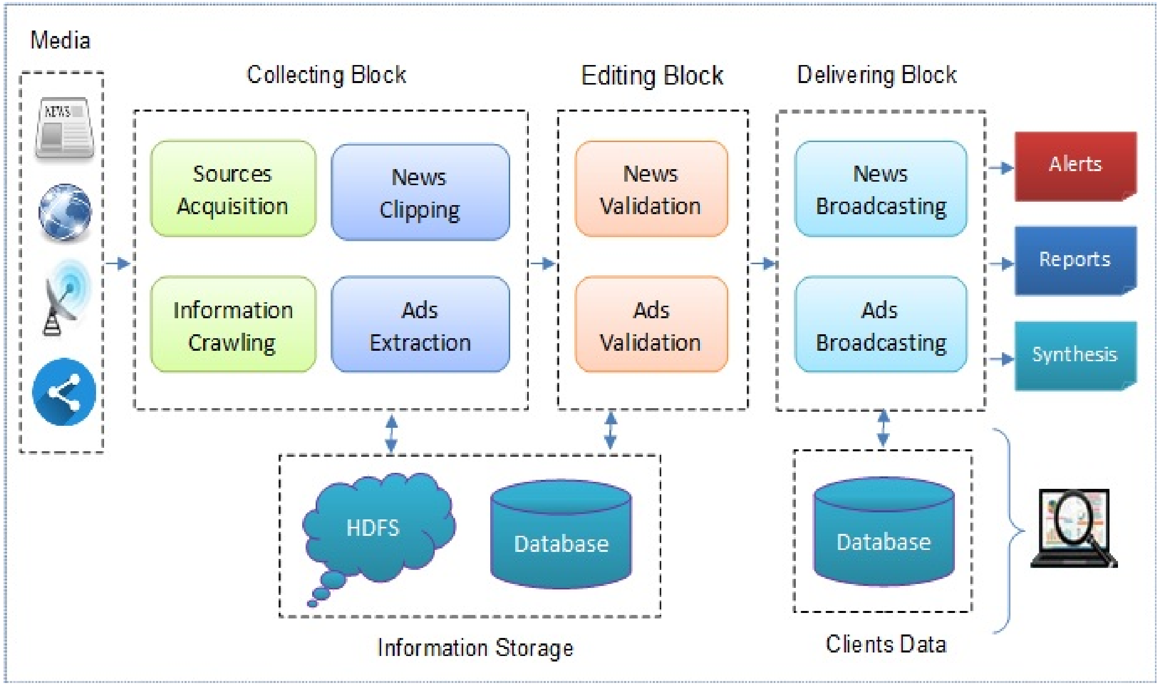


Figure 3.2: Overall architecture of the multimedia monitoring system

The media monitoring system is composed of three functional blocks, grouped in consideration of the user's categories: Collecting Block (CB), Editing Block (EB) and Delivering Block (DB). Each block exposes some functionality of the proposed system that we will detail later. Our work is mainly focused in TV news processing. Our system can be used by the following users: readers, journalists, editor-in-chief and broadcasting officer. Companies can access to the news portals on the Web and perform their search in the database.

3.3.1 Collecting Block

The collecting block consists of four modules: sources acquisition, information tracking, news clipping and advertisements extraction. Some tasks are done automatically and others are performed manually, we give an overview of these modules. The sources acquisition strategy varies according to the media to be captured: print news are digitized and stored in HDFS[2] or in NAS[3]; the broadcast contents from TV and radio channels are captured, stored by DVB-S card in database (metadata is not

[2] Hadoop Distributed File System
[3] Network Attached Storage

provided). Information crawling module is used for data capture from online news and social media, online news are downloaded using RSS feeds and social media are crawled and stored in database.

News clipping is a manual method invented by libraries used for newspaper processing; in our system we adopted the same method using the new ICTs. Users can produce a digital copy of news associated by some data (title, author, media source, time or page number and date of publication) from different media sources. In the past, users listen/watch audiovisual content and store segments of news. For this we introduce audiovisual processing algorithms to automate this task. News received from the crawling module are classified by users in order to keep only the news that concern their customers, the rest is deleted. The topics of news from different sources are fused.

To identify the advertisement segments in the video file and to define the boundary (start and end) for each instance of ads, segmentation can be performed using a sliding window, a peaks detector, or a silence detector. In our proposed system, the segmentation is done by splitting the audio content on silence boundaries that correspond to the low signal energy (values less than a threshold).

3.3.2 Editing Block

The editing block is composed of news indexing and news validation modules, they are done only by experienced journalists. The indexing module offers the possibility to edit, annotate and summarize each news item, also validation is a confirmation that the news is important to be delivered to companies. Every multimedia news record ingested in the system and validated by editor-in-chief must be sent to companies. In the same way, the advertisements extracted automatically should be validated before the delivery process.

3.3.3 Delivering Block

Like news media services, the system will send daily news alerts and reports via e-mail with articles containing all news and advertisements related to client needs. Also, a summary should be sent every month. News in reports are classified by date, media source and degree of importance. A broadcasting team can customize reports and assure that customers will receive exactly the relevant news to their preferences. Features can be media type, domain, location, language, etc. Customers are classified into two categories (executives and staff). The delivery is done automatically and planned in three stages (8am, 1pm, and 4pm). However, alerts can be sent in real time.

3.4 TV News Processing

In addition to the design of the MMS and in order to enhance our system, we integrate automatic functionalities that capture continuous TV streams and save them to one hour long mp4 files using the RTSP protocol/H.264 codec, segment streams in programs, identify news program and extract text from video frames. The details of the TV news processing steps are described in Figure 3.3 and detailed below.

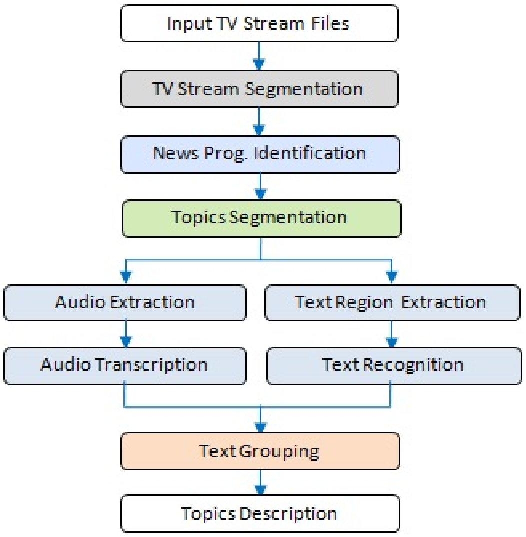


Figure 3.3: TV stream processing

3.4.1 TV Stream Segmentation

The first stage of news processing is to segment the TV streams in programs (called inter-segmentation) using visual features for channels that include monochrome frames or using audio features for others, because textual metadata are not provided by TV channels. One of the most popular solutions to solve this problem for the first category of channels is to use a simple monochrome frame detector that is based on standard deviation (SD) computation. Each frame with SD less than a threshold T_{sd} is considered a monochrome frame and indicates the start or the end of a program. The result of this stage is a set of temporal values Sp = T_1, T_2, …, T_n.

Concerning the second category of channels, it is necessary to analyze the stream using audio features and compare them with a dataset of the audio signatures of TV programs. More details will be given in the next section. The pseudo code presented in Algorithm 1 shows the steps involved in the TV stream segmentation process.

Algorithm 1 Segment the TV stream and extract the news programs

```
procedure ProcessVideofile($f$)
    $video \leftarrow OpenVideo(f)$
    $start \leftarrow GetVideo_info(video,'start')$
    $end \leftarrow GetVideoInfo(video,'duration')$
    $cores \leftarrow CpuCount()$
    $jobs \leftarrow []$
                                                        ▷ fork all jobs
    while $start < end$ do
        $step \leftarrow min(start + end/cores, end)$
        $proc \leftarrow NewsDetection(f, start, step)$
        $jobs.append(proc)$
        $start \leftarrow step + 1$
    end while
    $segments \leftarrow []$
                                                        ▷ join all jobs
    for $proc$ in $jobs$ do
        $result \leftarrow proc.join()$
        $segments.append(result)$
    end for
                                                        ▷ save segments
    for $item$ in $segments$ do
        $segment\_file \leftarrow GetTmpfile()$
        $seg\_video \leftarrow GetSubVideo(video, item[0], item[1])$
        $SaveVideo(seg\_video, segment\_file)$
    end for
end procedure
```

3.4.2 News Program Identification

The automatic identification of news program within TV streams is a classification problem and can be resolved using discriminative features and distance metrics. One of the best discriminative features is the video segment called “generic” that is a marker used to identify the start or the end of a particular program such as news program (Special programs are characterized by an audible and visual illustration of a specific generic). For that, we construct an audio dictionary composed of a set of generic video segments from different TV channels and represented by an MFCC[4] audio descriptor and a HOG[5] descriptor computed for each frame.

Most often, however, cepstral parameters are required and these are indicated by setting the target kind to `MFCC` standing for Mel-Frequency Cepstral Coefficients (MFCCs). These are calculated from the log filterbank amplitudes $\{m_j\}$ using the Discrete Cosine Transform. The MFCC feature extraction steps are described bellow:

- Frame the signal into short frames (windowing),
- Apply Discrete Fourier Transform (DFT) to each window (Eq. 3.1),
- Take the log of amplitude spectrum,
- Mel-scaling and smoothing,
- Discrete Cosine Transform (DCT) (Eq. 3.2),
- Obtain MFCC features.

The formulas of DFT and DCT are:

$$X_k = \sum_{j=0}^{N-1} x_j e^{-i2\pi jk/N} \tag{3.1}$$

$$X_k = \sum_{j=0}^{N-1} x_j \cos[\frac{\pi}{N}(j+1/2)k] \tag{3.2}$$

Where N represents the number of points of the audio signal, $0 \leq k < N$ and $i = \sqrt{-1}$.

[4] Mel Frequency Cepstral Coefficients

[5] Histogram of Oriented Gradient

Afterward, we compare each segment of video extracted from a program at time T$_i${Sp} using the DTW[6] distance (Eq. 3.3). If the distance is less than a threshold value T$_{Generic}$, we check if the visual similarity between segment frames and dictionary frames (using Euclidian distance of HOG vectors). If they are similar, we consider that this segment corresponds to a news program. Algorithm 2 shows the steps of the news identification process.

The formal definition [91] of the DTW distance is given as:

$$DTW(X,Y) = d(x_i, y_j) + \min \begin{cases} DTW(X, Y[2:m]), \\ DTW(X[2:n], Y), \\ DTW(X[2:n], Y[2:m]) \end{cases} \tag{3.3}$$

Where $X(x_1, x_2, ...x_n)$ and $Y(y_1, y_2, ...y_m)$ are two series of points and $d(x_i, y_j)$ indicates the distance between points x_i and y_j.

[6] Dynamic Time Warping

Algorithm 2 Process video segments in order to detect news programs

```
function NewsDetection($file$, $from$, $to$)
    $video \leftarrow OpenVideo(file, from, to)$
    $s \leftarrow 0$                                        ▷ start of segment
    $e \leftarrow to$                                       ▷ length of segment
    $w \leftarrow 3$                                        ▷ window = 3 secondes
    $l \leftarrow -1$                                       ▷ last position in segment
    $dataset\_waves \leftarrow OpenDataset('generics.txt')$
    $results \leftarrow []$
    while $s < e$ do
        $audio \leftarrow GetAudio(segment, s, min(s + w, end))$        ▷ extract audio
        $mfcc \leftarrow GetMFCC(audio)$                    ▷ compute MFCC features
        $dist \leftarrow 999999$
                                   ▷ compute distance between generic sequences
        for $item$ in $dataset\_waves$ do
            $dist \leftarrow DTW(item, mfcc)$
            if $dist < 90$ then
                break
            end if
        end for
 ▷ save and group consecutives positions/slide window over segment with w step
        if $dist < 90$ then
            if $(l = -1)$ or $(s - results[l, 1]) > 1)$ then
                $results.append([s, s, item.class])$
                $l \leftarrow l + 1$
            else
                $results[l, 1] \leftarrow s$
            end if
            $s \leftarrow s + w$
        else
            $s \leftarrow s + w/2$
        end if
    end while
    return $(results)$
end function
```

In our work, we have used the MFCC implementation of McFee et al. [104] and then the DTW implementation of Pierre Rouanet[7] In the same logic, we can also use other features like "jingle" that serve to do intra-segmentation and classification of news in topics (national, international, sport, business,...). The next stage is to process the news program.

[7] https://libraries.io/github/pierre-rouanet/dtw

3.4.3 News Topics Extraction

For each news program extracted in the previous stages, we proceed to extract news topics by the following steps :

1. Identify anchorpersons from each frame
   a. Extract all faces from each frame
   b. Group all similar faces together
   c. Select the group that has the most occurrences (anchorperson)
   d. Associate the co-anchorperson with the group (if there is one)
2. Extract video segments that occurs between two consecutive appearances of the anchor persons
3. Save extracted video segments
4. Describe the segments (topics) by one of the following methods:
   a. Audio Transcription
   b. Text recognition

We have defined some criteria to select faces in video frames such as:

- Face size: The size of the presenter face has a proportional value to the size of the whole image.
- Face distance: Inspired by Flores & al. [43], we estimate the distance between the camera and the human head by computing the distance between the right eye and the left eye.
- Face motion: Both anchorperson and camera are motionless. Based on that, we eliminate some moving faces shots.

The pseudo code to do this task is described in Algorithm 3.

Algorithm 3 News topics segmentation

```
function TopicsSegmentation($file$, $from$, $to$)
    $video \leftarrow OpenVideo(file, from, to)$
    $s \leftarrow 0$                                              ▷ start of segment
    $e \leftarrow to$                                             ▷ length of segment
    $fc \leftarrow [-1, -1, 0, 0, 0, 0]$
    $prev\_face \leftarrow []$
    $results \leftarrow []$
    while $s < e$ do
        $img \leftarrow GetFrame(video, s)$                       ▷ extract image
        $faces \leftarrow DetectFaces(img)$                       ▷ detect all faces
        for $face$ in $faces$ do
            if $prev\_face$=[] then
                $prev\_face \leftarrow face$
            end if
            $cam\_dist \leftarrow face.righteye - face.lefteye$   ▷ check criteria
            $face\_size \leftarrow face.w * face.h * 100.0/(video.size[0] * video.size[1])$
            if  ($cam\_dist > 25$) and ($cam\_dist < 55$) and ($face\_size$¿1) and
              then
                if $fc[0]{=}{-1}$ then
                    $fc \leftarrow [s, s, 0, face.x, face.y, face.w, face.h]$
                else if $(s - fc[1]) > 1$ then
                    $results.append(fc)$
                    $fc \leftarrow [s, s, 0, face.x, face.y, face.w, face.h]$
                else                                              ▷ check face motion
                    $face\_dist \leftarrow sqrt(pow(face.x - prev\_face.x, 2)+$
                        $pow(face.y - prev\_face.y, 2))$
                    if $face\_dist < 60$ then
                        $fc[1] \leftarrow s$
                    else
                        $results.append(fc)$
                        $fc \leftarrow [s, s, 0, face.x, face.y, face.w, face.h]$
                    end if
                end if
            end if
            $prev\_face \leftarrow face$
        end for
        $s \leftarrow s + 1$
    end while
    if $(fc[1] - fc[0]) > 0$ then
        $results.append(fc)$
    end if
    return ($results$)
end function
```

The method used for identifying anchor persons is as follows:

1. Create a dictionary of extracted faces
2. Construct a frequency occurrence matrix based on faces
3. Select the face (anchorperson) that has the max of occurrence
4. Select the face (co-anchorperson) that has the max of occurrence and which is visible simultaneously which the anchorperson

Each occurrence of a face represents an appearance on TV studio (successive frames containing the same face). For an anchorperson, it indicates a set of intervention that we will use it to delimit news reports (topics).

3.4.4 Audio Extraction

For every topics segmented in the previous steps, the audio content will be extracted and saved in a temporary file in order to do a transcription of content and get the corresponding text.

3.4.5 Audio Transcription

Several online frameworks offer the audio transcription service for non commercial use by using their API. In our context, developing an audio transcription tool is another challenge that only large companies can effectively deal with like Google and IBM. For this, we have opted to use and automatic speech recognition (ASR) software to transcribe the speech associated with a news video.

3.4.6 Text Region Extraction

The automatic extraction and recognition of text embedded in news videos provides an efficient approach to annotate TV news content. One or several rows of text appear in the screen bottom (in the scope of ¼ of screen height) and express the meaning of news presented. Text region extraction steps are shown in Figure 3.4.

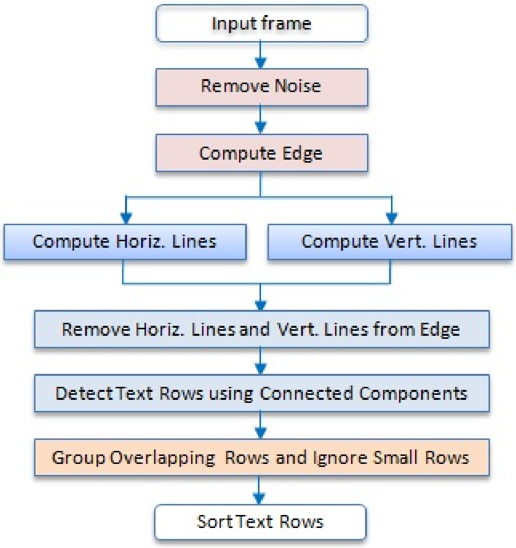

Input frame
Remove Noise
Compute Edge
Compute Horiz. Lines
Compute Vert. Lines
Remove Horiz. Lines and Vert. Lines from Edge
Detect Text Rows using Connected Components
Group Overlapping Rows and Ignore Small Rows
Sort Text Rows

Figure 3.4: Text region extraction

The final result of this step is a set of rectangular regions bounding text lines that will be submitted to the transcription stage.

3.4.7 Text Transcription

The text regions extracted are binarized and transmitted to the OCR (Optical Character Recognition) engine that processes each text region and returns equivalent textual information. There are a few OCR systems capable of recognizing Arabic text, namely: Automatic Reader produced by the Sakhr[8] Software Company; FineReader[9] produced by the ABBYY Company and Tesseract[10] produced originally by Hewlett-Packard and available from Google. We are interested in Tesseract because it is an open-source OCR.

3.4.8 Text Grouping

In the final post-processing of news processing, we use combined speech transcription and image understanding technology to represent the TV news by a bag of words that will be exploited for describing the content and the news topics.

[8] http://www.sakhr.com
[9] http://www.abbyy.com
[10] http://code.google.com/p/tesseract-ocr

3.5 Advertisements Detection and Classification

The first stage of ads processing is to extract frames from TV stream. This functionality is used during the training and testing phase. All extracted frames will be used for matching and detecting of the presence or absence of ads. In the training phase, we extract all frames from different ads and construct a dictionary composed of the fingerprint of frames. We extract only one frame per second in the test phase because in one second the contents of the frames are often similar to each other.

3.5.1 Frame Hashing-Based Classification

The automatic identification of ads content within TV streams is a classification problem and can be resolved using discriminative features and distance metric. One of the fastest and best discriminative features is the use of a fingerprint algorithm like the perceptual hashing used to identify the presence or the absence of ads.

In order to decide if a frame is a part of an ad or not, we use a simple search in a dictionary. This technique is called direct hash lookup. According to the training dataset, we can extract the category of each ad using a simple textual list containing the id and the name of ad. The pseudo code presented in Algorithm 4 shows the steps involved in the last stage of processing streams.

Algorithm 4 Ads detection and Classification in TV streams

```
procedure ProcessFrame($f$)
    $list\_ads \leftarrow \{\}$
    $dict \leftarrow \{dictionaryofhashedframesofads\}$
    $buffer \leftarrow \{setofframesinbuffer\}$
    while $bufferisnotempty$ do
        $frame \leftarrow ReadFramefromBuffer(buffer)$
        $roi \leftarrow CropFrameCenter(frame)$
        $hash \leftarrow GetPerceptualHash(roi)$
        if $hash$ in $dict$ then
            $list\_ads \leftarrow list\_ads + dict(hash)$
        end if
    end while
    return ($list\_ads$)
end procedure
```

3.5.2 Audio Features Based Classification

Based on the audio part of advertisements, we should extract a perceptual digest of this content. One of the technics is the audio fingerprint. The calculations of MFCC begin with a downsampling of the original signal to a reduced frequency. MFCC feature is calculated using filters with an order of 20, using a window length 25 msec and overlap length 10 msec. Then, the MFCC vector is sent to a pre-trained CNN classifier for dimensionality reduction. The architecture of the used CNN is described in Table 3.1.

Layers	Filters/units	Filter sizes	Outputs
Input			1000x1
Convolution	16	9	992x16
Maxpool	16		62x16
Dropout	0.1		62x16
Convolution	32	3	60x32
Maxpool	4		15x32
Dropout	0.1		15x32
Convolution	64	3	13x64
Maxpool	4		3x64
Dropout	0.15		3x64
Convolution	256	3	256
Global Maxpool			256
Dropout	0.2		256
Dense			128
Dense			50

Table 3.1: CNN classifier

Given a video file, the first step is video segmentation based on audio content. It allows to obtain a set of small segments that probably contain advertisements. In our proposed system, we use a classifier based on a deep learning model. The convolutional layer uses filters to extract local patterns from input data and outputs feature maps. Then, the pooling layer reduces the dimensions of the feature maps. The max-pooling applies a max filter over a specified window size. The dropout layer is used to avoid overfitting, whereas the dense layer is used for classification. The first dense layer is used for feature extraction and the second one is used as an output for the classifier. The input to the model consists of a vector of MFCCs corresponding to a segment of audio signal, and the output layer contains the probabilities of predicted classes.

A common problem in analyzing continuous streams is how to determine which instances of the processed data stand out as being dissimilar to the trained data. Such instances are known as outliers or anomalies. For that, we propose to align the predicted class with the trained data of the same class using the dynamic time warping distance (DTW). The predicted class is accepted if the distance DTW is less than a threshold. The alignment serves to reduce the misclassification produced by the classifier and reduce the cost of computation compared to the traditional DTW technique used alone.

3.6 Experiments

For the evaluation of our proposed system, two experiments are conducted. In our first experiment, we build a dataset of tv streams from five national TV channels. The second experiment concerned international TV channels, it was conducted to achieve comparative results with similar works.

3.6.1 Experiments on national TV channels

The data used in our experiments has been collected by MediaMarketing[11] company specialized in news monitoring, dataset is composed of five different national channels in Algeria that spoke arabic, french or english. The majority of channels are recorded at a resolution of 720x480 pixels. We have took a sample with 24h times recording for each channel, the recording streams are divided into one-hour portions.

For the experiments, we have created a dataset composed of the generics of the news programs as well as the jingles of advertisements relating to each TV channels and which are represented by the MFCC features and using the library LibROSA, an open-source Python package for music and audio analysis. In order to compute the similarity between the dataset and the TV stream we have used the DTW dis-

[11] www.mediamarketing-dz.com

tance (best threshold = 90), we divided the speech signal into short frames, e.g. 3 seconds segments, and processed each frame as a single unit. We have used a classical evaluation process; we compare the manual annotation provided by MediaMarketing (ground truth) with results obtained by our system.

To choose a discriminative and efficient descriptor used in the learning part, we tested the MFCC (audio), HOG (visual) and MFCC+HOG (Table 3.2). Based on our tests, we have chosen the MFCC descriptor alone and, to increase the relevance, we proceeded to the improvement of our segmentation algorithm by grouping the marks that are close to each other. In order to reduce execution time, we performed experiments through parallel processing. We have tested a processor with 4 cores and a processor with 12 cores (Table 3.3).

Features	Relevance	Execution time
MFCC	93%	360s
HOG	95%	840s
MFCC + HOG	98%	960s

Table 3.2: TV Stream Analysis (1 hour) with different descriptors

Processors	Execution time
1 core	1500s
4 cores	840s
12 cores	360s

Table 3.3: Multicores CPU parallel processing

In Topics extraction level, we tested different models of face detection and description like MTCNN, DLIB, OpenFace. In light of our tests, we used DLIB library to do face detection because it is the most accurate in our task.

To evaluate the performance of the proposed system, the main blocs of the system are compared with the ground truth, which are:

- News program detection
- News topics extraction

We have used the precision, recall and F-measure metrics (Eq. 3.4, 3.5 & 3.6) that are described as follows:

- Precision (P) is defined as the number of hits (corrects) over the number of hits plus the number of false alarms.

$$P = \frac{\#hits}{\#hits + \#false\ alarms} \tag{3.4}$$

- Recall (R) is defined as the number of hits over the number of hits plus the number of misses.

$$R = \frac{\#hits}{\#hits + \#misses} \tag{3.5}$$

- F-score (F) is defined as the harmonic mean of the precision and the recall, the formula is:

$$F - score = 2.\frac{P.R}{P+R} \tag{3.6}$$

The experimentation results of each bloc are shown in tables 3.4, and 3.5.

National TV channels	P	R	F
A3	0.93	0.90	0.91
CA	0.91	0.94	0.92
Ennahar	0.90	0.89	0.89
Dzair News	0.95	0.93	0.94
Echorouk	0.96	0.95	0.95
All TV channels	**0.93**	**0.92**	**0.92**

Table 3.4: News program identification

National TV channels	P	R	F
A3	0.88	0.84	0.81
CA	0.85	0.90	0.87
Ennahar	0.90	0.93	0.91
Dzair News	0.92	0.91	0.91
Echorouk	0.89	0.92	0.90
All TV channels	**0.88**	**0.90**	**0.88**

Table 3.5: News topics segmentation

For the advertisements classification model, we exploited 50 classes of advertisements to train our classifier, and we fixed the threshold value (DTW) to 20. We obtained an accuracy for training and testing equal to 1.0 and a loss value equal to 0.001. The performance of our model is as follows: Precision = **0.92**, Recall = **0.73**, and F-measure = **0.81**

The obtained results shows high precision of our proposed approach, but there are still improvements to be done in text extraction from the video frames. The best results are observed in news program detection and advertisements classification. We can say that we can generalize the proposed process on other TV channels and the result will also be relevant. To obtain an accurate comparison with other works, we will do other experiments that we describe in the next section.

3.6.2 Experiments on international TV channels

After experimenting with our approach on national TV channels and to make an accurate comparison with other works, we evaluate the anchorperson detection method on a varied dataset composed of TV news from five international channels namely TF1, France24, M6, LCI, and CNews. This dataset is used by Dumont et al. [38], Zlitni et al. [184], Kannao et al. [75], and Hmayda et al. [61]

We evaluate the performance using the precision, recall, and F-measure metrics. The comparative results are detailed in the following tables:

International TV channels	P	R	F
TF1	0.92	0.95	0.93
LCI	0.78	0.90	0.83
France24	0.94	0.96	0.94
CNews	0.85	0.94	0.89
M6	0.96	0.95	0.95
All TV channels	**0.89**	**0.94**	**0.90**

Table 3.6: News topics segmentation

Approaches / Measures	P	R	F
Dumont et al. (2012)	0.76	0.89	0.80
Zlitni et al. (2016)	0.78	0.90	0.81
Kannao et al. (2019)	0.79	0.92	0.83
Hmayda et al. (2020)	0.86	**0.96**	**0.91**
Our approach (2021)	**0.89**	0.94	0.90

Table 3.7: Comparison with other approaches

We observe from tables (3.5, 3.6, 3.7) that, the proposed method for TV news segmentation gives better performance in processing national channels and international as well. On the other hand, our method has a better result in term of precision than other approaches. In terms of recall and F-measure, we obtain an acceptable results compared to results obtained by Hmayda et al. From these results, we also showed that our proposal can be used for processing different kind of channels. These results can be justified by the fact that:

- The previous works (Hmayda et al., Zlitni et al., ...) use an hierarchical clustering of faces shot, then the cluster which contains the largest number of faces is considered an anchorperson.
- However, our approach use a frequency occurrence matrix based on faces similarities grouping method. Where, we have exploited a frequency occurrence matrix to identify the face that has the maximum number of occurrences as well as the faces that appear at the same time as this one. Each occurrence represents an appearance in the TV studio during the news program (successive frames containing the same face).

3.7 Conclusion

In this paper, we have proposed a global architecture of our multimedia monitoring system, that processes newspaper, online news, broadcast news and social media. This system is a service for a targeted audiences such as large corporations and state institutions. It offers a digital storage of all news clips, faster delivery via e-mail and guaranteed near zero missed news clips. We have tried to automate some tasks of the system's workflow processes, the results obtained are satisfactory, we intend to extend this work on the treatment of radio channels and we wish to parallelize some treatment with Spark framework or implement our algorithms on a different multi-core GPU.

However, we hope to do exhaustive comparison with others methods using a public datasets such as ALIF (A Dataset for Arabic Embedded Text Recognition in TV Broadcast). In future work we plan to do an exhaustive analysis of TV News for the identification and segmentation of topics. Among the future work, we will investigate three tasks which are:

- News story categorisation using NLP processing, each topics must be classified in one of the categories: politics, economics, health, sports, ...),
- Alignment of segmented news topics with textual news collected by the RSS Feeds,
- Design of an autoencoder for outliers detection in advertisements classification.

CHAPTER IV

A Deep Hybrid Model for Advertisements Detection in Broadcast TV and Radio Content

In this chapter, we introduce a deep hybrid model for advertisement detection in broadcast TV and radio content that is a part of our global architecture previously presented but now full detailed and well experimented using different deep neural networks. We investigated and compared different audio features extraction and three machine learning algorithms.

4.1 Introduction

TV and radio monitoring systems are a type of the media monitoring systems (MMS). TV and radio monitoring is defined as the process of reading, watching, or listening to the editorial content of media sources continuously [30]. The MMS provides companies near real-time reports on all aspects of compliance and competitor activity. Video advertisements (TV commercials) have become an indispensable tool for marketing. Companies not only invest heavily in advertising, but several companies generate revenue from advertisements [143]. The TV and radio advertisement market was valued at over 214 billion dollars in 2008 [22] and at over 563 billion dollars in November 2019[1]. Through media monitoring, companies find information about competitors and specific issues relevant to their business domain. Reports are delivered by e-mail to enable executives in client companies to keep them up-to-date with a comprehensive overview of their reputation.

[1] https://www.statista.com/statistics/236943/global-advertising-spending/

In the literature, existing algorithms for the detection of advertising can be grouped into two main categories: Those which use explicit prior knowledge of a known set of advertisements and identify them using fingerprinting methods, and those which rely on heuristics as advertising indicators [129]. Other researches have focused on using visual features for segmentation, classification, and summarization. In one of them, [6] proposed a multimedia retrieval system that uses low-level information and textual information in the indexing process. Recently, researchers have begun to realize that audio characteristics are equally, if not more, important when it comes to understanding the semantic content of a video [161]. Consequently, to process the broadcast content of both TV and radio, audio features can be used with potentially high accuracy. Advertisement detection is a classification problem with variable data training sizes where every class may have different audio length, which vary between 3 and 60 seconds. Some recent research has started to focus on the classification of imbalanced data since real-world data is often skewed [13]. Deep learning-based models are used to overcome some of the limitations of the hand-crafted features [108], but in the context of electronic media (data stream), they suffer from the presence of outliers and that leads to misclassification. Another issue we should be concerned with is the variation in loudness found on processing different media that does not conform to the international standards for loudness measurement by ITU, ATSC, and EBU [86]. Currently, there are several solutions for the automatic detection and identification of TV and radio advertisement that are based on typical characteristics which are grouped in intrinsic and extrinsic characteristics [46], that are used in the automatic detection process. However, these solutions cannot be directly applied to many of the existing TV and radio broadcasters that not use valid characteristics (black frames, silence, presence/absence of channel logo, . . .).

In this work, we present a novel architecture of Media Advertisements Detection System that is based on a Deep Hybrid Model (DHM-ADS), detailing some algorithms for processing TV and radio streams of local channels. These algorithms include stream acquisition and storage, stream analysis, and advertisements identification and classification.

The next sections of this chapter are organized as follows: In section 2, we present a brief review of media advertisements detection approaches. In section 3, we present the architecture of the proposed system for the TV/radio advertisements monitoring system with details of its processing blocks. In section 4, we describe our experimentations varried on real world data and evaluates and discuss the performance of our proposals. Finally, we conclude the chapter with some perspectives.

4.2 Related work

Media advertisements monitoring systems are used to detect and extract the advertisement in multiple sources of broadcasted streams. Existing approaches in the literature can be classified into two main categories: knowledge-based detection and repetition-based detection [36] [164]. The first one uses the a priori knowledge like black frames or the absence of logos to identify the TV advertisements. The second one observes the notion of duplication of shots, but it requires a large computational.

Broadcast stream analysis requires a preprocessing steps. Some works on TV stream propose a macro-segmentation of the stream. Macro-segmentation algorithms generally rely on detecting inter-programs (IP), which include commercials, trailers, jingles, and credits [83]. Pinquier and Andre-Obrecht, detect and locate one or many jingles to structure the audio dataflow in program broadcasts [122]. [129] proposed an approach that uses only audio features and centers on the detection of short silences that exist at the boundaries between programs and advertisements.

The performance of any machine learning algorithm depends on the features on which the training and testing are done [138]. Hence feature extraction is one of the most vital parts of a machine learning process. Some methods have been proposed for audio classification are based on the time-domain that includes: zero crossing rate and root mean square [116], and entropy of energy [121]. Other methods are based on frequency-domain that includes: spectral centroid, spectral spread, spectral entropy [110], mel-frequency cepstral coefficients (MFCC) [9], and Spectral features (spectral centroid, spectral bandwidth, spectral roll-off) [134].

Several works were proposed in the literature for advertisements detection and extraction [46]. They suggest exploiting the presence or absence of the broadcaster (channel) logo on the screen. However, in our context, the local channels always publish their logo. In [89], the authors proposed a system based on exact-duplicate matching that detects and localizes TV commercials in a video stream, clusters the exact duplicates, and detects duplicate exact-duplicate clusters across video streams, but the processing is done in batch mode. The authors of [37] proposed a multimodal (visual, audio, and text) commercial video digest scheme to segment individual commercials and carry out a semantic content analysis within a detected commercial segment from TV streams, the disadvantage of this approach lies in its high computational cost. Unlike many other types of data used with machine learning, audio data consists of time series which are usually quite large [63]. Similarity measures on time-series have emphasized the need for elastic methods that align the pairs of time series. Neural networks have been successfully applied to do sequences alignment. NeuralWarp [50] is a model that predicts whether or not to align frames of the sequences. Dynamic time warping (DTW) is a robust similarity measure of time series, but, it has a high computational complexity [92].

A common problem that researchers face when analyzing real-world datasets is determining which instances of the processed data stand out as being dissimilar to the trained data. Such instances are known as outliers or anomalies. [24] proposed a one-class neural network (OC-NN) model to detect anomalies in complex datasets. To train their network, they use a loss function inferred from a one-class SVM (OC-SVM) that was proposed by [137]. The autoencoder is a fundamental deep learning approach to anomaly detection [107]. A typical autoencoder network includes two phases: an encoder that transforms the input data into a lower dimensional representation and a decoder, that tries to reconstruct the original input data [53]. Figure 4.1 shows a generic model of the architecture of the Autoencoder.

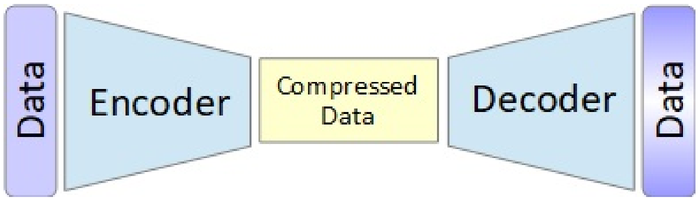


Figure 4.1: The architecture of an Autoencoder

The objective of an Autoencoder is to minimize the loss function l [25], a typical loss function is the mean squared error (MSE) that is as follows:

$$l_{MSE}(u, v) = ||u, v||_2^2 \tag{4.1}$$

A powerful multimedia content analysis system often requires real-time processing and scalability to deliver and store videos. To achieve scalability, programmers need to use multiprocessing techniques on CPU [52] or distributed programming frameworks such as Hadoop/MapReduce. [82] proposed a high-speed and scalable video server system using a PC-cluster for video content delivery. In another work, [170] proposed a scalable content-based analysis of images in web archives with TensorFlow and the archives Unleashed toolkit, The authors evaluate their model on CPU and GPU and show that processing time was greatly reduced. Multimedia big data often entails considerably more resources in terms of the acquisition, storage, transmission, presentation, and processing, including, for instance, the need for GPU processing and parallel, distributed software [183]. Through this work, we will experiment with different deep neural network models with two validation approaches of audio classification.

4.3 The Proposed Framework

Media monitoring systems must provide a 360-degree view of media sources in realtime and 24/7 coverage for competing companies. Based on their needs, we propose a TV/Radio Advertisements Monitoring System that detects and classifies the broadcasted advertisements. To this end, we investigate and compare different audio features extraction and different machine learning algorithms. The system described in Figure 4.2 is composed of three functional blocks grouped in consideration of the user's categories: Collecting Block (CB), Processing Block (PB), and Delivering Block (DB). Each block exposes some functionality of the proposed system that we will detail later. Also we focus our work primarily on advertisements processing.

The collecting block captures streams of different channels and stores them into a Hadoop Distributed File System (HDFS). The continuous stream is segmented and saved by one hour long that offers management facilities and avoid big volume processing. Each file is processed by the processing block. The processing block is com-

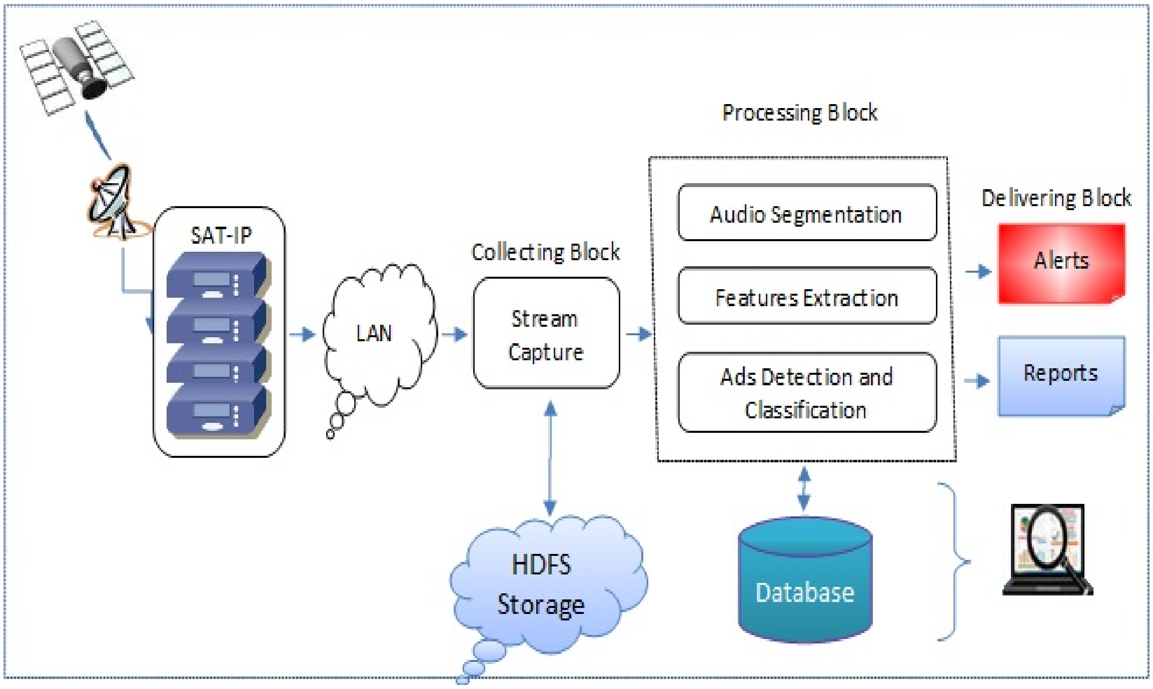


Figure 4.2: The block diagram of the proposed TV and Radio Advertisements Monitoring System

posed of three modules: Audio Segmentation, Features Extraction, Advertisements Detection and Classification. All tasks are done automatically.
We can summarize the advertisements classification steps as follows:

- Reduce noise,
- Segment the audio file,
- Classify segments,
- Save advertisements labels and time.

Details of these modules are given in the next section.

4.4 Processing Block Description

4.4.1 Audio Segmentation

Given an audio file, the first step is audio segmentation. It allows to obtain the delineation of a continuous audio stream into acoustically homogeneous regions [23]. To identify the advertisement segments in the audio file and to define the boundary (start and end) for each instance of advertisements, segmentation can be performed using a sliding window, a peaks detector, or a silence detector. The drawback of the sliding window is that audio matching with a continuous stream is hard. Also, peaks detector gives too many points and generates noise in the matching step. Consequently, to do the best segmentation, we split the audio content on silence boundaries that correspond to the low signal energy (values less than a threshold ($T_{Silence}$)). The split function returns only segments that have duration at least 3s and less than 60s (normal duration of advertisements). The other segments are ignored by the following modules.

4.4.2 Features Extraction

The extraction of features is a very important part of data analysis. It is required for classification and prediction algorithms (machine learning). Before feature extraction, the data should be preprocessed to render it usable for predicting the class of a sample and help distinguish different classes. There are many audio transformations introduced in the audio data before feature extraction, including audio re-sampling and audio normalization. An other type of normalization can be performed on extracted features such as the technique proposed by Prabhavalkar et al. [125] where the authors used automatic gain control to normalize the signal level. In our prototype, we will experiment with different features combined with MFCC and evaluate the accuracy rate of each combination.

4.4.3 Advertisements Detection and Classification

The core module of our framework provides advertisements detection and classification. At this level, we are faced with two challenges : the unbalanced data in the training step, and the presence of outliers in the testing step. To overcome these problems, we use early and late filter jointly with different deep neural networks such as: Artificial Neural Network (ANN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN). This can improves the classification accuracy and remove outliers.

In the late filter, the predicted class is aligned with the trained data of the same class using the dynamic time warping (DTW) distance, the predicted class is accepted if the distance DTW is less than a threshold (T_{DTW}). The filtering serves to reduce the misclassification produced by the classifier and reduce the cost of computation compared to the traditional DTW technique used alone. For the early filter, we train an autoencoder with all known classes and use it for outliers detection. An input data submitted to the autoencoder is considered as an outlier if the loss value calculated using predicted signal is greater than a threshold (T_{MSE}). Only normal data (non outliers) will be sent to the deep neural network for classification.

According to the training dataset, we can extract the name of each advertisement using a simple list containing the id and name of the ad. Figure 4.3 shows a general block diagram for advertisements detection and classification system using late or early filters.

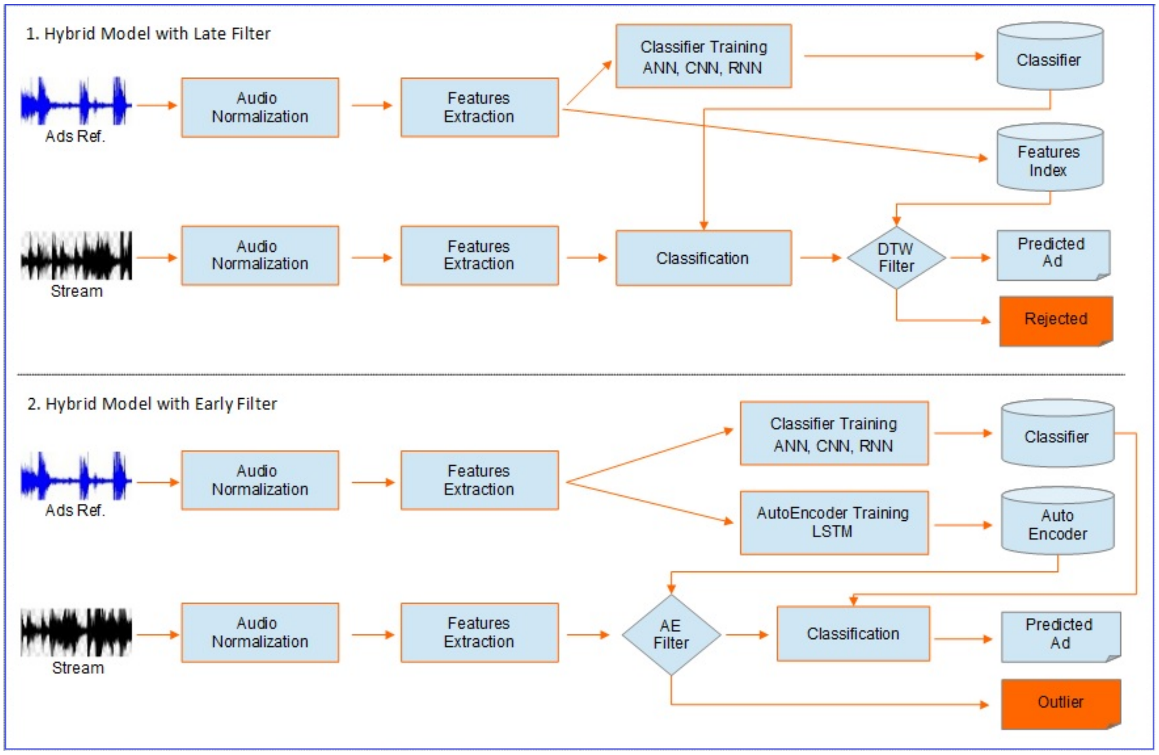


Figure 4.3: Block diagram of advertisements detection and classification

4.5 Deep Neural Network Models

Across the time series domain, ANNs, CNNs, and RNNs have been successfully employed to solve many audio classification problems. Thus, we used designed models based on them and compared them to choose the best one that will be used in production. In our proposed Deep Neural Networks (DNN), we use a BatchNormalization hidden layer for data normalization, with Dense and Dropout hidden layers in the ANN model. Also for the CNN model, we add Convolution, Dense, Maxpolling, and Flatten hidden layers. For the RNN model, we use the Long Short-Term Memory Networks (LSTM) layers. The input to the DNN consists of a combination of MFCC with some features (zero crossing rate, root mean square, ...) computed over 25ms of an audio signal with a frame size of 2s. The Softmax output layer contains the probabilities of predicted classes. We use a Rectified Linear Unit (ReLU) and hyperbolic tangent (TanH) activation functions.

For the Autoencoder model we use LSTM hidden layers with TanH activation function. The input data is the framed signal and the output is the predicted signal after compression processing. The following Tables (4.1, 4.2, and 4.3) describe the architecture of the ANN, CNN, and RNN classifiers used for the detection of advertisements.

Layers	**Filters/units**	**Functions**
Input	20x32	
BatchNormalization	20x32	
Flatten	640	
Dense	256	TanH
Dropout	0.5	
Dense	256	ReLU
Dropout	0.5	
Dense	50	Softmax

Table 4.1: ANN classifier model

Layers	**Filters/units**	**Functions**
Input	20x32	
BatchNormalization	20x32	
Convolution	16	ReLU
Maxpool	2	
Dropout	0.1	
Convolution	16	ReLU
Maxpool	2	
Dropout	0.1	
Global Maxpool		
Flatten		
Dense	50	Softmax

Table 4.2: CNN classifier model

Layers	**Filters/units**	**Functions**
Input	20x32	
BatchNormalization	20x32	
LSTM	64	ReLU
Dense	64	TanH
Maxpool	2	
LSTM	32	ReLU
Dense	32	TanH
Maxpool	2	
Flatten		
Dense	50	Softmax

Table 4.3: RNN classifier model

Our dataset of advertisements has some level of class imbalance, each class having a different number of examples. In order to handle the imbalanced dataset, we need to apply class balancing techniques. We apply the weight balancing technique [120], since under or oversampling data is not suitable for advertisements data.

4.6 Experiments

To evaluate our proposed framework we conducted different experiments on the dataset collected by MediaMarketing[2]. This dataset consists of eight different TV and Radio channels. The streams are recorded on 24h times recording for each channel, the recording streams are divided into one-hour segments. We randomly split the dataset into 80% training and 20% testing disjoint partitions. To choose the discriminative descriptors in the training part, two combinations of features are extracted:

- MFCC and Root Mean Square,
- MFCC, Chromagram, Contrast, Tonal Centroid, and Root Mean Square.

We exploited 50 classes of advertisements to train our classifiers and autoencoder, and the duration of advertisements varies between 3s to 60s. The dataset was annotated manually by MediaMarketing. Also, our framework uses three thresholds that are defined by experiments as follows :

- $T_{Silence}$: We fixed this value to 0.00003. This threshold is used to indicate if the signal is a silence,
- T_{DTW} : the best value in our experiments is equal to 20,
- T_{MSE} : this threshold is fixed by calculating the mean square error on the training process. After training the autoencoder at 200 epochs with 23792 parameters, we got the value of TMSE equal to 0.06

The use of autoencoder expects to do dimensionality reduction, so we need to preserve the initial data and reduce the loss function. Figure 4.4 presents a chart that shows the loss error.

[2] https://www.mediamarketing-dz.com

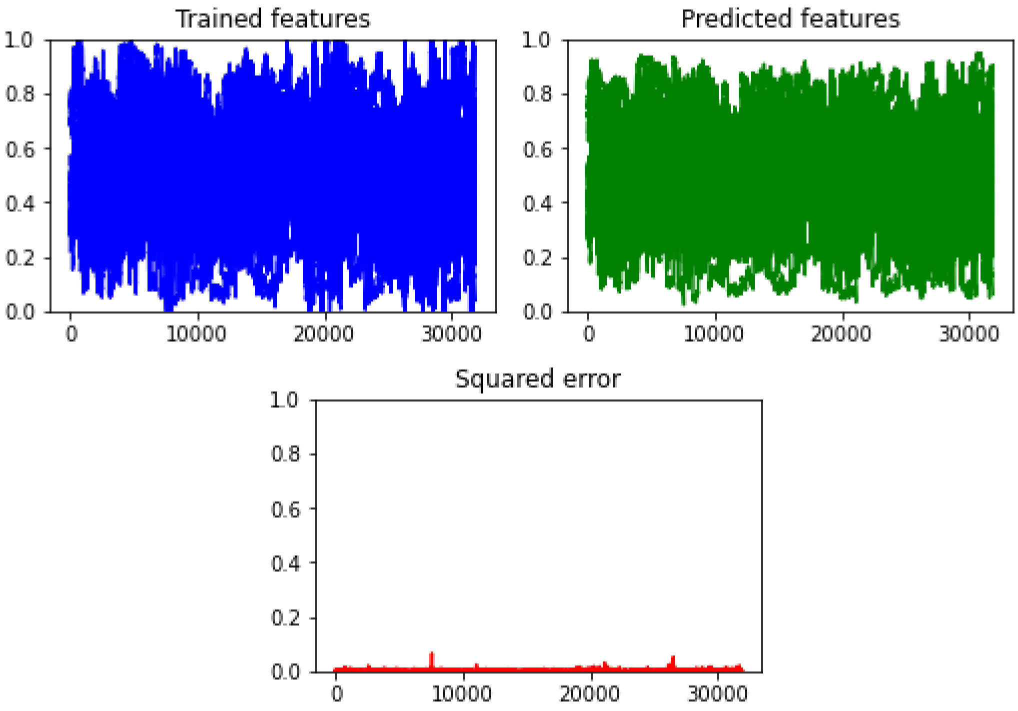


Figure 4.4: Mean Square Error calculated from the training and test data

In order to reduce the time execution, we only keep predicted class if it has probability greater than 0.99. The size of the training dataset is 25519 items and the test dataset is 6380 items. The details of the training and testing steps are shown in Table 4.4.

Performance / Models	ANN	CNN	RNN
Total params	83073	68457	37585
Training accuracy	1.00	1.00	1.00
Test accuracy	1.00	1.00	1.00
Training loss	0.001	0.001	0.001
Test loss	0.001	0.005	0.002

Table 4.4: Training and test parameters

In Figure 4.5 we show the accuracy of each proposed models.

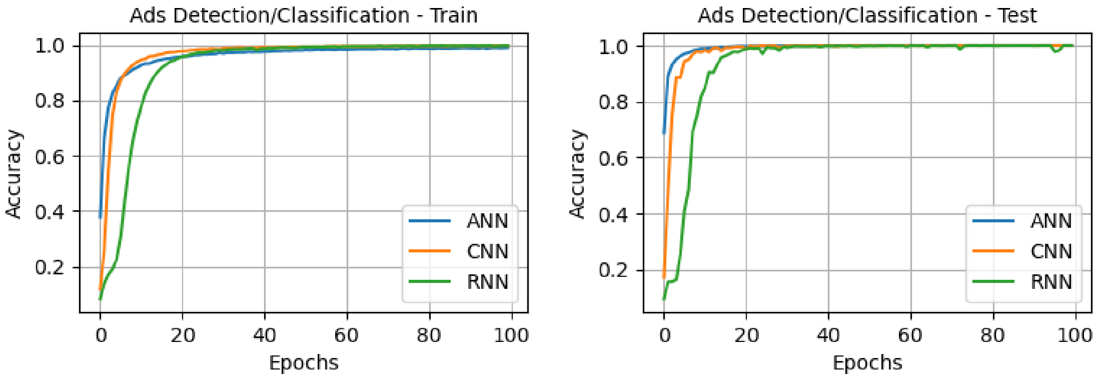


Figure 4.5: Performance accuracy of Advertisements Detection and Classification models

The results in Figure 4.5 shows that the classification-trained models give high accuracy. Overall, the results are satisfying both using MFCC and Root Mean Square or MFCC combined with other features. We finally choose MFCC and Root Mean Square for the rest of our experiments. We now turn our attention to evaluating our models on real-word conditions using previously saved continuous streams. Performance is assessed in terms of Precision (P), Recall (R), and F-measure (F). Precision computes the proportion of all detected sounds that are of the correct class. Recall, by contrast, computes the proportion of detected sound events out of the total number of sound events. F-measure is a combination of precision and recall [105][150]. These measures are calculated as follows:

$$P = TP/(TP + FP) \tag{4.2}$$

$$R = TP/(TP + FN) \tag{4.3}$$

$$F = (2 * P * R)/(P + R) \tag{4.4}$$

Where, TP denotes the number True Positives, FP the number of False Positives, and FN the number of False Negatives. We present the experimental results of proposed models in Table 4.5.

Models / Measures	P	R	F
ANN-DTW	**0.93**	0.70	0.80
ANN-Autoencoder	0.92	**0.84**	**0.87**
CNN-DTW	0.80	0.65	0.72
CNN-Autoencoder	0.87	0.71	0.78
RNN-DTW	0.84	0.76	0.80
RNN-Autoencoder	0.90	0.82	0.86

Table 4.5: Precision, Recall, and F-measure achieved over the implemented models

In our experiments we varied the number of CPU cores for parallel processing and observed how the performance changed for each model. From Figure 4.6 we can see that the performance is high when the number of CPU cores increases.

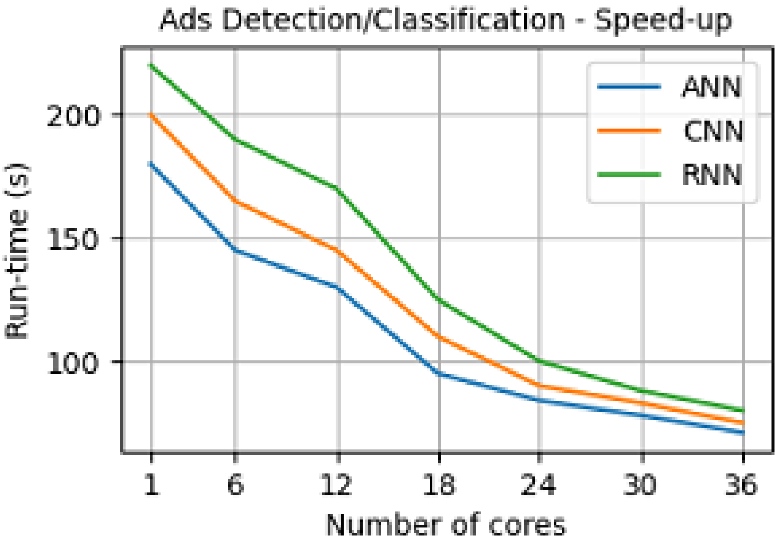


Figure 4.6: Speed-ups for the three models

This observation is surprising as we expected temporal models to outperform static models (ANN) because the audio content is time-series data.

Experimental results demonstrated that the ANN-Autoencoder hybrid model significantly improved the detection performance. Also, ANN combined with the conventional dynamic time warping ANN-DTW is still an efficient method and gives acceptable results, but the recall is low compared to the other models and it's not recommended for the scalability. Fortunately, the autoencoder models provide a potential way to achieve outliers detection although the weak loss. So, we will explore more autoencoder models to improve outliers detection.

To obtain an accurate comparison with other works, we evaluate the performance or our approach with DejaVu-Shazam [162], and Ad-Net [108] approaches. The Shazam approach is based on peaks detection. However, Ad-Net uses a convolutional neural network model and is based on audio spectrogram representation. Our approach is based on MFCC features.
The experimental results are as follows:

Approaches / Measures	P	R	F
DejaVu-Shazam (2014)	0.91	0.80	0.85
Ad-Net (2018)	0.89	**0.85**	0.86
Our approach (2021)	**0.92**	0.84	**0.87**

Table 4.6: Comparison with other approaches

From these experimental results, we observed that DejaVu take a long time to generate fingerprints of the audio file and it generates a huge number of hashes. The MFCCs are relatively inexpensive to calculate and we found that our approach outperformed the Ad-Net approach in the experimental tests. Based on these observations, we can consider that our model ANN-Autoencoder has more advantages in performance and running time. The Shazam algorithm has been created especially for identifying cover audios from other millions of songs. However, when applied to the advertisements detection task, the system is less effective due to the perceptual similarity judgments of songs were sometimes correlated.

4.7 Conclusion

In this work, we have proposed a global architecture of TV and radio advertisements monitoring system based on deep learning models, that captures streams, detect and identify advertisements broadcasted on different channels at different times. The system provides a service for a targeted audience such as large companies and state institutions. We have proposed a silence-based segmentation algorithm that splits an audio stream. Also, an autoencoder was implemented for outliers detection and, three classifiers (ANN, CNN, RNN) were implemented, compared, and evaluated. Since the obtained results are satisfactory, and the comparative results prove the performance of our approach.

The novelty in our work is that our approach is efficient for both television and radio channels and exploits the power of deep learning to detect efficiently the advertisements, we intend to extend this work to novel methods such as an unsupervised model which is very important for the detection of unknown advertisements. We also plan to parallelize and implement our algorithms on a different multi-core GPU for scalability and high performance. Lastly, we are considering the integration of other media such as online TV and Youtube in the analysis processes.

CHAPTER V

A Big Data Framework for Video Content Processing

For efficient processing of large scale TV video contents, there is a need for a scalable and distributed system. In this chapter, we address the problem of TV advertisements identification based on a distributed architecture. It consists to generate the dataset and identification of categories in video streams.

5.1 Introduction

Managing and analyzing TV stream of many channels in near real-time is a challenge. Recent works discussed multimedia processing with Big data platforms like Hadoop, Spark and others. Big data technologies, such as Hadoop or Spark echo system, are software platforms designed for distributed computing to process, analyze, and extract the valuable insights from large datasets in a scalable and reliable way. The cloud is preferably appropriate to offer the big data computation power required for the processing of these large datasets [117]. In [5], authors provide an extensive study on intelligent video big data in the cloud. First, we define basic terminologies and establish the relation between video big data analytics and cloud computing. Several studies have shown that Spark can significantly outperform Hadoop for a broad range of applications [103]. Given the performance offered by Spark, we investigate how this platform can be used to perform video analysis and data extraction.

5.2 Related Work

Different big data frameworks such as Apache Spark, Apache Storm, and Apache Hadoop have been widely used to perform massive data processing on computer clusters [136]. Apache Hadoop has been applied in large text data and graph data mining. Many achievements have been made in the research of video processing based on MapReduce [47]. Spark is the new solution for massive data processing. Whereas Hadoop reads and writes files to HDFS, Spark processes data in RAM using a concept called Resilient Distributed Dataset (RDD). Another framework for stream processing is Apache Kafka that can be used for real-time video processing.

Wang et al. [171] proposed a parallel video data analysis framework based on Spark, and examined the power of the MapReduce framework on different multimedia data mining applications such as video event detection and near-duplicate video retrieval. Zhang et al. [180] introduced a cloud-based architecture that can provide both real-time processing and offline batch data analysis of large-scale videos. This architecture is based on both Apache Kafka and Storm for real-time processing. Zhang et al. [178] presented an online video surveillance framework that includes the distributed Kafka message queue and Spark Streaming. Lv et al. [99] introduced a Spark based solution for near-duplicate video detection, and employed three feature descriptors: Scale Invariant Feature Transform (SIFT), Local Maximal Occurrence (LOMO), and Color Name (CN). In which for SIFT and CN, , Bag-of-Visual-Words (BoVW) is introduced to characterize video features.

Recently, a distributed deep learning framework called BigDL [31] is introduced, which is implemented on top of Apache Spark and allows users to develop deep learning applications. BigDL support different learning algorithms such as Neon, Caffe, TensorFlow, Torch, and Theano. It can efficiently scale out to perform data analytics using Spark. Hamilton et al. [54] proposed MMLSpark which provides a distributed image processing library that integrates OpenCV with Spark and combines deep learning library Cognitive Toolkit, with Apache Spark. Also, they integrate the popular image processing library OpenCV with Spark.

In paper [70], authors proposed and evaluated cloud services based on Hadoop and Spark for high resolution video streams in order to perform line detection using

Canny edge detection followed by Hough transform. The results demonstrate the effectiveness of parallel implementation of computer vision algorithms to achieve good scalability for real-world applications.

In literature, some researchers exploited distributed computing technology with GPU for the development of large-scale video retrieval systems. Wang et al. [158] proposed a novel MapReduce framework for near-duplicate video retrieval for large-scale multimedia data processing by joining the computing power of GPU's and MapReduce model to speed up the video processing. In the same way, Rathore et al. [130] proposed a model which integrate parallel and distributed environment of Hadoop ecosystem with GPU and Spark to make it more powerful and real-time in terms of processing. Also, they implemented a MapReduce equivalent algorithm for efficient data processing using GPUs.

In addition, numerous commercial solutions have already deployed the cloud-based distributed video processing system. Google Vision API [49] offers a Video Data management and retrieval framework. They also provide APIs for video processing and can recognize over 20,000 objects, places, and actions in stored and streaming video. IBM Intelligent Video Analytics [67] is another cloud service which provides batch video data processing and real-time video stream processing.

5.3 Architecture of the Big Data Framework for Video Content Processing

In this section, we take a more detailed look at our proposed big data based architecture, we look at the Hadoop cluster. Next, we introduce the data transfer between nodes. Finally, we take a look at GPU units used for fast computation.

We propose an approach for fast and parallel video analysis and processing framework using Apache Spark and Kafka. The cluster helps us to handle large-scale of video data and reduce the processing time. The three main components of our system are the video stream collector, the video processing, and the video content classification. The specification is shown in Figure 5.1.

Kafka is a distributed messaging system developed for the purpose of the collection and delivery of large volumes of data with high throughput and low latency. It is

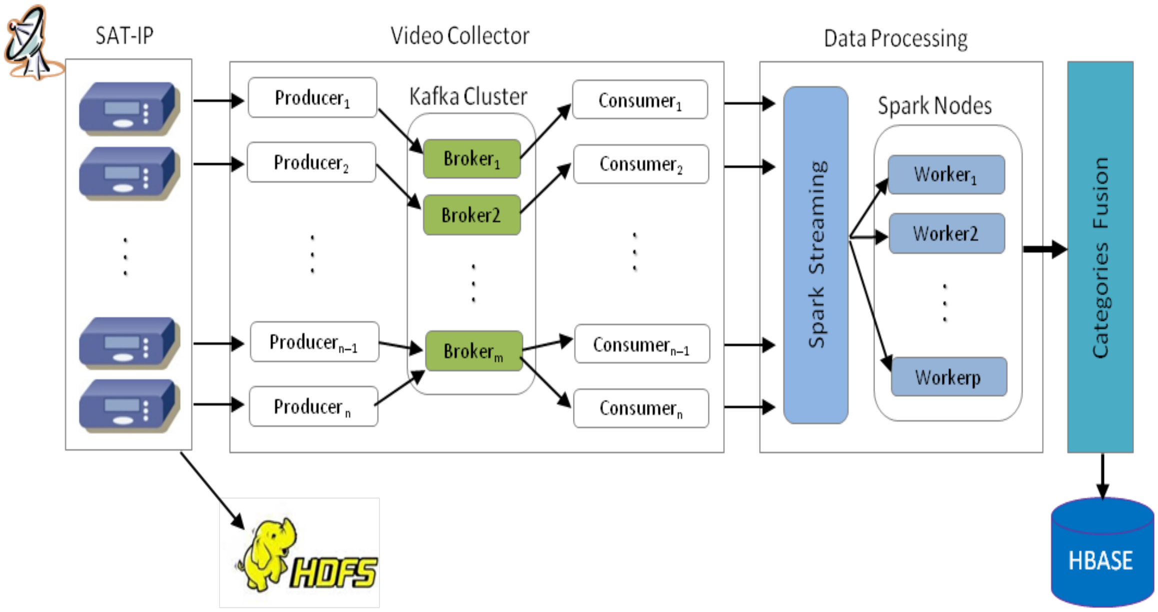


Figure 5.1: Architecture of the Big Data Framework for Video Content Processing

executable as a cluster on multiple servers called Kafka Cluster, and that stores streams consisting of keys, values, and timestamps in categories called topics. There are two major types of messaging models. The first is a push-type model, in which the transmitting side starts transferring data. The second is a pulltype model, in which the transfer is started by the receiving side sending a data request. Kafka consists of producers, brokers, and consumers [69]. For each TV channel we create a topic in the Kafka cluster.

Spark was developed at the University of California at Berkley, and it is a distributed processing framework that stores and processes large-scale data. MapReduce is a specialized distributed batch processing framework for processing methods used in Apache Hadoop. On the other hand, Spark increases the execution speed of the entire process by speeding up the input/output by storing the data in memory. In Spark, data are handled as a DataFrame because a DataFrame can easily be processed in Spark SQL. The proposed framework includes three components which are:

- Video Collector,
- Data Processing,
- Categories Fusion.

In the next, we will review in detail these components.

5.4 Video Collector

The video collector uses a cluster of SAT-IP receivers that provide broadcated TV content. The collector reads the stream from each channel and convert the video content into a series of video frames. Each TV channel can have different specifications such as the resolution, or number of frames per second. The collector uses the FFMPEG video-processing library to convert a video stream into frames. Each frame is resized to a specific resolution (e.g. 300x300). Afterwards, we compute for each frame the hash code, which will be represented in JSON format and pushed to a kafka queue.

Given a video file, we publish the content to a specified Kafka topic using the following algorithm:

Algorithm 5 Publish video content to a Kafka topic

```
procedure PublishVideoFile($file$, $topic$)
    $video \leftarrow OpenVideo(file)$
    $frame\_no \leftarrow 1$
    $count \leftarrow Size(video)$
    $producer \leftarrow KafkaProducer$
    while $frame\_no \leq count$ do
        if $frame\_no\%5$ then
            $frame \leftarrow GetFrame(video, frame\_no)$
            $hash \leftarrow GetHash(frame)$
            $producer.send(topic, hash, frame\_no)$
        end if
        $frame\_no \leftarrow frame\_no + 1$
    end while
    return
end procedure
```

5.5 Data Processing

We introduce a distributed data processing architecture that provides data processing on top of the Apache Spark framework. The Data Processing component is mainly in charge of identify the categoryies of each frame hash received from the spark streaming component. Spark Streaming receives input data streams and divides the data into batches, which are then processed by the Spark nodes (workers). each worker node is responsible to find the category of each frame hash, then the result is pushed to the Categories Fusion component. the corresponding algorithm is as follow:

Algorithm 6 Process data by Spark cluster

```
procedure ProcessDataStream
    $sc \leftarrow SparkContext$
    $consumer \leftarrow KafkaConsumer$
    $df \leftarrow sc.dataframe(hashes, categories)$
    $queue \leftarrow []$
    for $topic, hash, frame\_no$ in $consumer$ do
        if $hash$ in $df$ then
            $time \leftarrow frame\_no/24$
            $queue.push(topic, category, time)$
        end if
    end for
    return
end procedure
```

5.6 Categories Fusion

Once categories of advertisements are found, this step focuses on grouping the consecutive category in one instance. Then, all results will be saved in the database. The strucuture of the final table will contain the topic name (TV), the category of advertisement, and the broadcast time. We can resume the steps of this module in the following pseudo algorithm:

Algorithm 7 Categories Fusion

```
procedure CategoriesFusion
    $cat\_queue \leftarrow sort(queue[topic, category, time])$
    $cat\_list \leftarrow []$
    for $topic, category, time$ in $cat\_queue$ do
        if $(topic, category)$ not in $cat\_list$ then          ▷ long elapsed time
            $final\_list.add(topic, category, time)$
        end if
    end for
    $return(cat\_list)$
end procedure
```

5.7 Experimentation

To evaluate our proposed architecture on large scale, it is necessary to have the appropriate environment such as a Big data infrastructure dedicated for data analytics. To demonstrate the feasibility of our proposed architecture, we performed the experiments on Colab[1] server provided by Google. The host system is a server equiped by 2vCPU, and 12 GB RAM. Each server has two 6-core Intel Xeon processors running at 2.3 Ghz. We have installed Apache Spark in local mode (no cluster). We compared performance of the framework between CPU based node and GPU (NVIDIA) based node. The size of the processed video file is 1 hour long (3600 seconds), and has a frame rate of 25 frames per second. The following tables presents the obtained results.

Configuration	@1/2 frame	@1/4 frame	@1/6 frame	@1/8 frame
CPU (1 node)	300s	230s	120s	100s
GPU (1 node)	110s	80s	65s	40s
Precision	0.94	0.93	0.85	0.78

Table 5.1: Execution time for video processing on Spark

Configuration	@1/2 frame	@1/4 frame	@1/6 frame	@1/8 frame
CPU/GPU	0.94	0.93	0.85	0.78

Table 5.2: Pecision of the video processing framework on Spark

The execution time is improved by reducing the number of processed frames per second as we see in the table 5.1. But, we can't preserve the precision rate of our framework which is indicated on the table in table 5.2. For that reason, we choose parameter that provide the best results in execution time and precision as well. According to the results, we choose to process 1 frame from every 4 consecutive frames. Through these experiments, we can confirm that the use of Big data frameworks is more interesting especially when we exploit the power of GPU.

[1] https://colab.research.google.com/

5.8 Conclusion

In this chapter, we have proposed a robust and distributed framework based on Apache Kafka and Apache Spark with the intention to identify and classify advertisements in TV video stream. We then implemented the proposed framework on Google Colaboratory (Colab) which is done using PySpark framework. By experiments, we demonstrate the success of using the Big data frameworks especially using GPUs. The proposed system can be a candidate solution for large-scale video analytics over a multi-node environment including Apache Spark and Kafka cluster. Where we can measure the resource usage and performance scalability against different sizes of clusters.

CHAPTER VI

Papers and Author's Contributions

6.1 Paper I

Abdesalam Amrane, Abdelkrim Meziane, Noue el Houda Boulekrinat, and Ali Atik: Fast and smart object proposals for object detection, 6th. International Symposium ISKO-Maghreb, Al-Hoceima, Morocco, May 11-13, 2017. (Proceedings).

6.2 Paper II

Abdesalam Amrane, Abdelkrim Meziane, and Nour El Houda Boulkrinat: Object Detection in Images Based on Homogeneous Region Segmentation. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer, Cham, 2018.
doi:10.1007/978-3-319-92058-0_31

6.3 Paper III

Abdesalam, Amrane, Abdelkrim Meziane, Abdelmounaam Rezgui, and Abdelhamid Lebal: A Deep Hybrid Model for Advertisement Detection in Broadcast TV and Radio Content. International Journal of Computational Vision and Robotics. 2021.
doi:10.1504/ijcvr.2021.10041276

6.4 Paper IV

Abdesalam, Amrane, Abdelkrim Meziane, Nour El Houda Boulkrinat, and Ali Atik: A unified system for processing news and detecting advertisements in TV streams. International Journal of Informatics and Applied Mathematics. vol. 4, no. 2, pp. 17-34, Dec. 2021.
doi:10.53508/ijiam.1020112

CHAPTER VII

Conclusions

7.1 Summary

In this thesis, we have developed a TV stream analysis framework based on deep learning models and algorithms. The process covers videos storage, analysis, and information extraction such as news clips, and advertisements reporting. The framework was built component by component, We started by focusing on TV stream segmentation problem, a new method was proposed in TV programs segmentation especially for the news program. Next, we have proposed a macro-segmentation of the TV news content in order to extract the topics, this method consists to identify the sets of frames that the news presenter is located, and the closed captions was extracted and converted to text using OCR tool.

In the second work, we tackled the advertisements identification using deep neural networks models, three classifiers: ANN, CNN, and RNN were implemented, compared, and evaluated. Also, an autoencoder was implemented for outliers detection. Our approach is based on audio features that allows to use it for both TV and radio channels, a dataset composed of 50 classes was made for training phase. Our approach successfully classifies advertisements with a classification accuracy of 100%. However, It remains some improvements to identifies closets advertisements.

Our final contribution concerns the problem of TV content classification based on distributed deep learning models. It consists to perform the training and testing models to achieve a reduced run-time. We look at the use of Hadoop/Spark cluster. Next, we introduce the data transfert between nodes. Finally, we take a look at GPU units used for fast computation.

7.2 Future Work

Although we have proposed several contributions in the TV content analysis domain, some points remain unresolved and we put them in perspective. All experiments was done on a local datasets built from the national media channels, we hope to do exhaustive comparison with others methods using a public datasets such as ALIF. Also, we plan to do an exhaustive analysis of TV News for the identification and segmentation of topics. We will investigate the news story categorisation problem using NLP techniques, when each topics must be classified in one of the categories: politic, economic, health, sport, ...).

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Sadiq H Abdulhussain, Abd Rahman Ramli, M Iqbal Saripan, Basheera M Mahmmod, Syed Abdul Rahman Al-Haddad, Wissam A Jassim, et al. Methods and challenges in shot boundary detection: a review. *Entropy*, 20(4):214, 2018.

[2] Tariq Abdullah, Ashiq Anjum, M Fahim Tariq, Yusuf Baltaci, and Nikos Antonopoulos. Traffic monitoring using video analytics in clouds. In *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, pages 39–48. IEEE, 2014.

[3] Alina Elma Abduraman. *Unsupervised TV program structuring*. PhD thesis, Télécom ParisTech, 2013.

[4] Abdullah I Al-Shoshan. Speech and music classification and separation: a review. *Journal of King Saud University-Engineering Sciences*, 19(1):95–132, 2006.

[5] Aftab Alam, Irfan Ullah, and Young-Koo Lee. Video big data analytics in the cloud: A reference architecture, survey, opportunities, and open research issues. *IEEE Access*, 8:152377–152422, 2020.

[6] Abdesalam Amrane, Hakima Mellah, Rachid Aliradi, and Youssef Amghar. Semantic indexing of multimedia content using textual and visual information. *International journal of advanced media and communication*, 5(2-3):182–194, 2014.

[7] Ashiq Anjum, Tariq Abdullah, Muhammad Tariq, Yusuf Baltaci, and Nick Antonopoulos. Video stream analysis in clouds: An object detection and classification framework for high performance video analytics. *IEEE Transactions on Cloud Computing*, 2016.

[8] Johan Barthélemy, Nicolas Verstaevel, Hugh Forehead, and Pascal Perez. Edge-computing video analytics for real-time traffic monitoring in a smart city. *Sensors*, 19(9):2048, 2019.

[9] Mark A Bartsch and Gregory H Wakefield. Audio thumbnailing of popular music using chroma-based representations. *IEEE Transactions on multimedia*, 7(1):96–104, 2005.

[10] Gustavo EAPA Batista, Xiaoyue Wang, and Eamonn J Keogh. A complexity-invariant distance measure for time series. In *Proceedings of the 2011 SIAM international conference on data mining*, pages 699–710. SIAM, 2011.

[11] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[12] Sid-Ahmed Berrani, Patrick Lechat, and Gaël Manson. Tv broadcast macro-segmentation: Metadata-based vs. content-based approaches. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 325–332, 2007.

[13] Cigdem Beyan and Robert Fisher. Classifying imbalanced data sets using similarity based hierarchical decomposition. *Pattern Recognition*, 48(5):1653–1672, 2015.

[14] Sergiy Bilobrov and Iouri Poutivski. Commercial detection based on audio fingerprinting, February 9 2016. US Patent 9,258,604.

[15] Murat Birinci and Serkan Kiranyaz. A perceptual scheme for fully automatic video shot boundary detection. *signal processing: image communication*, 29(3):410–423, 2014.

[16] Darin Brezeale and Diane J. Cook. Automatic video classification: A survey of the literature. *EEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(3):416–430, 2008.

[17] Karlis Martins Briedis and Karlis Freivalds. On-line television stream classification by genre. *Baltic Journal of Modern Computing*, 6(3):235–246, 2018.

[18] Jinzhou Cai. Large-scale multi-label video classification, 2018.

[19] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[20] Pedro Cano. *Content-based audio search: from fingerprinting to semantic audio retrieval.* PhD thesis, Citeseer, 2006.

[21] Pedro Cano, Martin Kaltenbrunner, Fabien Gouyon, and Eloi Batlle. On the use of fastmap for audio information retrieval and browsing. 2002.

[22] Patrick Cardinal, Vishwa Gupta, and Gilles Boulianne. Content-based advertisement detection. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[23] Diego Castán, Alfonso Ortega, Antonio Miguel, and Eduardo Lleida. Audio segmentation-by-classification approach based on factor analysis in broadcast news domain. *EURASIP Journal on Audio, Speech, and Music Processing*, 2014(1):34, 2014.

[24] R Chalapathy, AK Menon, and S Chawla. Anomaly detection using one-class neural networks. arxiv 2018. *arXiv preprint arXiv:1802.06360*.

[25] David Charte, Francisco Charte, Salvador García, María J del Jesus, and Francisco Herrera. A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. *Information Fusion*, 44:78–96, 2018.

[26] Jianguo Chen, Kenli Li, Qingying Deng, Keqin Li, and S Yu Philip. Distributed deep learning model for intelligent video surveillance systems with edge computing. *IEEE Transactions on Industrial Informatics*, 2019.

[27] Lei Chen, M Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502, 2005.

[28] Tse-Shih Chen, Ming-Fen Lin, Tzi-cker Chieuh, Cheng-Hsin Chang, and Wei-Heng Tai. An intelligent surveillance video analysis service in cloud environment. In *2015 International Carnahan Conference on Security Technology (ICCST)*, pages 1–6. IEEE, 2015.

[29] W Comcowich. The importance of tv news monitoring and how to do it. 2016. https://glean.info/the-importance-of-tv-news-monitoring-and-how-to-do-it/. Accessed: August 2020.

[30] William J Comcowich. Media monitoring: The complete guide. *Cyber alert Inc, White paper*, 2010.

[31] Jason Jinquan Dai, Yiheng Wang, Xin Qiu, Ding Ding, Yao Zhang, Yanzhang Wang, Xianyan Jia, Cherry Li Zhang, Yan Wan, Zhichao Li, et al. Bigdl: A distributed deep learning framework for big data. In *Proceedings of the ACM Symposium on Cloud Computing*, pages 50–60, 2019.

[32] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[33] Amal Dandashi and Jihad Mohamad Alja'am. Video classification methods: Multimodal techniques. In *Recent Trends in Computer Applications*, pages 33–51. Springer, 2018.

[34] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[35] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

[36] Ina Döhring and Rainer Lienhart. Mining tv broadcasts for recurring video sequences. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 1–8, 2009.

[37] Ling-Yu Duan, Jinqiao Wang, Yantao Zheng, Jesse S Jin, Hanqing Lu, and Changsheng Xu. Segmentation, categorization, and identification of commercial clips from tv streams using multimodal analysis. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 201–210, 2006.

[38] Emilie Dumont and Georges Quénot. Automatic story segmentation for tv news video using multiple modalities. *International journal of digital multimedia broadcasting*, 2012, 2012.

[39] Elie El-Khoury, Christine Sénac, and Philippe Joly. Unsupervised segmentation methods of tv contents. *International journal of digital multimedia broadcasting*, 2010, 2010.

[40] Ali Mert Ertugrul and Pinar Karagoz. Movie genre classification from plot summaries using bidirectional lstm. In *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, pages 248–251. IEEE, 2018.

[41] J Gregorio Escalada, Helenca Duxans, David Conejero, Albert Asensio, and David Salinas. Newsclipping: An automatic multimedia news clipping application. In *2010 International Workshop on Content Based Multimedia Indexing (CBMI)*, pages 1–6. IEEE, 2010.

[42] Wei Feng, Xuecheng Nie, Yujun Zhang, Zhi-Qiang Liu, and Jianwu Dang. Story co-segmentation of chinese broadcast news using weakly-supervised semantic similarity. *Neurocomputing*, 355:121–133, 2019.

[43] Arturo Flores, Eric Christiansen, David Kriegman, and Serge Belongie. Camera distance from face images. In *International Symposium on Visual Computing*, pages 513–522. Springer, 2013.

[44] David A Forsyth and Jean Ponce. *Computer vision: a modern approach.* Prentice Hall Professional Technical Reference, 2002.

[45] Hiranmay Ghosh, Sunil Kumar Kopparapu, Tanushyam Chattopadhyay, Ashish Khare, Sujal Subhash Wattamwar, Amarendra Gorai, and Meghna Pandharipande. Multimodal indexing of multilingual news video. *International Journal of Digital Multimedia Broadcasting*, 2010, 2010.

[46] Alexandre Gomes, Maria Paula Queluz, and Fernando Pereira. Automatic detection of tv commercial blocks: A new approach based on digital on-screen graphics classification. In *2017 11th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pages 1–6. IEEE, 2017.

[47] Herman M Gomes, João M de Carvalho, Luciana R Veloso, Adalberto G Teixeira, B de O Tarciso Filho, Abner MC de Araújo, Tong Zhang, Lisandro Trarbach, and Fabio Machado. Mapreduce vocabulary tree: An approach for large scale image indexing and search in the cloud. In *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*, pages 170–173. IEEE, 2016.

[48] Ekaterina Gonina, Gerald Friedland, Eric Battenberg, Penporn Koanantakool, Michael Driscoll, Evangelos Georganas, and Kurt Keutzer. Scalable multimedia content analysis on parallel platforms using python. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 10(2):1–22, 2014.

[49] Google. Google vision api. 2021. https://cloud.google.com/video-intelligence/. Accessed: September 2021.

[50] Josif Grabocka and Lars Schmidt-Thieme. Neuralwarp: Time-series similarity with warping networks. *arXiv preprint arXiv:1812.08306*, 2018.

[51] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[52] Antonio Greco, Nicolai Petkov, Alessia Saggese, and Mario Vento. Aren: A deep learning approach for sound event recognition using a brain inspired representation. *IEEE Transactions on Information Forensics and Security*, 15:3610–3624, 2020.

[53] Wenzhong Guo, Jinyu Cai, and Shiping Wang. Unsupervised discriminative feature representation via adversarial auto-encoder. *Applied Intelligence*, 50(4):1155–1171, 2020.

[54] Mark Hamilton, Sudarshan Raghunathan, Akshaya Annavajhala, Danil Kirsanov, Eduardo Leon, Eli Barzilay, Ilya Matiach, Joe Davison, Maureen Busch, Miruna Oprescu, et al. Flexible and scalable deep learning with mmlspark. In *International Conference on Predictive Applications and APIs*, pages 11–22. PMLR, 2018.

[55] Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

[56] Christopher G Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[57] Alexander Hauptmann, Rong Yan, Yanjun Qi, Rong Jin, Michael G Christel, Mark Derthick, Ming-yu Chen, Robert Baron, W-H Lin, and Tobun D Ng. Video classification and retrieval with the informedia digital video library system. *Baltic Journal of Modern Computing*, 2002.

[58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[59] Paulina Hensman and David Masko. The impact of imbalanced training data for convolutional neural networks. *Degree Project in Computer Science, KTH Royal Institute of Technology*, 2015.

[60] Cormac Herley. Argos: Automatically extracting repeating objects from multimedia streams. *IEEE Transactions on multimedia*, 8(1):115–129, 2006.

[61] Mounira Hmayda, Ridha Ejbali, and Mourad Zaied. Classification program and story boundaries segmentation in tv news broadcast videos via deep convolutional neural network. *Journal of Computer Science*, 2020.

[62] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[63] Helge Homburg, Ingo Mierswa, Bülent Möller, Katharina Morik, and Michael Wurst. A benchmark dataset for audio classification and clustering. In *ISMIR*, volume 2005, pages 528–31, 2005.

[64] Hao Hu. Leaning robust sequence features via dynamic temporal pattern discovery. 2019.

[65] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, and Raj Reddy. *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice hall PTR, 2001.

[66] Yalniz I. Zeki, Jégou Hervé, and Mahajan Dhruv. Billion-scale semi-supervised learning for state-of-the-art image and video classification. 2019. https://ai.facebook.com/blog/billion-scale-semi-supervised-learning/.
Accessed: September 2020.

[67] IBM. Ibm intelligent video analytics. 2021. https://www.ibm.com/cloud/.
Accessed: September 2021.

[68] Zein Al Abidin Ibrahim. Tv stream table of content: a new level in the hierarchical video representation. *J. Comput. Sci. Appl*, 7(1):1–9, 2019.

[69] Ayae Ichinose, Atsuko Takefusa, Hidemoto Nakada, and Masato Oguchi. A study of a video analysis framework using kafka and spark streaming. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2396–2401. IEEE, 2017.

[70] Bilal Iqbal, Waheed Iqbal, Nazar Khan, Arif Mahmood, and Abdelkarim Erradi. Canny edge detection and hough transform for high resolution video streams using hadoop and spark. *Cluster Computing*, 23(1):397–408, 2020.

[71] Longlong Jing, Toufiq Parag, Zhe Wu, Yingli Tian, and Hongcheng Wang. Videossl: Semi-supervised learning for video classification. *arXiv preprint arXiv:2003.00197*, 2020.

[72] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.

[73] Ton Kalker, Jaap Haitsma, and Job C Oostveen. Issues with digital watermarking and perceptual hashing. In *Multimedia Systems and Applications IV*, volume 4518, pages 189–197. International Society for Optics and Photonics, 2001.

[74] Raghvendra Kannao, Durgaprasad Dandi, Swamy Yellapu, and Prithwijit Guha. News program detection in tv broadcast videos. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 546–550, 2016.

[75] Raghvendra Kannao and Prithwijit Guha. Segmenting with style: detecting program and story boundaries in tv news broadcast videos. *Multimedia Tools and Applications*, 78(22):31925–31957, 2019.

[76] Raghvendra Kannao and Prithwijit Guha. A system for semantic segmentation of tv news broadcast videos. *Multimedia Tools and Applications*, 79(9):6191–6225, 2020.

[77] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[78] Lukas Kaupp, Ulrich Beez, Jens Hülsmann, and Bernhard G Humm. Outlier detection in temporal spatial log data using autoencoder for industry 4.0. In *International Conference on Engineering Applications of Neural Networks*, pages 55–65. Springer, 2019.

[79] Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and information systems*, 7(3):358–386, 2005.

[80] Khushboo Khurana and MB Chandak. Study of various video annotation techniques. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(1):909–914, 2013.

[81] Jang-Hui Kim and Dae-Seong Kang. Design of an object-based video retrieval system using sca and invariant moments. In *International Conference on Future Generation Communication and Networking*, pages 509–516. Springer, 2009.

[82] Hiroyuki Kimiyama, Mitsuru Maruyama, Masayuki Kobayashi, Masao Sakai, and Satria Mandala. An uhd video handling system using a scalable server over an ip network. *International Journal of Advanced Media and Communication*, 7(1):1–19, 2017.

[83] Yiannis Kompatsiaris, Bernard Merialdo, and Shiguo Lian. *TV content analysis: Techniques and applications*. CRC Press, 2012.

[84] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[85] Hansang Lee, Minseok Park, and Junmo Kim. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In *2016 IEEE international conference on image processing (ICIP)*, pages 3713–3717. IEEE, 2016.

[86] SangWoon Lee, BongJin Baek, and CheongGhil Kim. A study on audio levels and loudness standard for digital broadcasting program. In *2014 International Conference on IT Convergence and Security (ICITCS)*, pages 1–3. IEEE, 2014.

[87] Seok-Lyong Lee, Seok-Ju Chun, Deok-Hwan Kim, Ju-Hong Lee, and Chin-Wan Chung. Similarity search for multidimensional data sequences. In *Proceedings of 16th International Conference on Data Engineering (Cat. No. 00CB37073)*, pages 599–608. IEEE, 2000.

[88] Hongliang Li and King Ngi Ngan. Image/video segmentation: Current status, trends, and challenges. In *Video segmentation and its applications*, pages 1–23. Springer, 2011.

[89] Hongzhi Li, Brendan Jou, Jospeh G Ellis, Daniel Morozoff, and Shih-Fu Chang. News rover: Exploring topical structures and serendipity in heterogeneous multimedia news. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 449–450, 2013.

[90] Yijun Li and Suhuai Luo. A tv commercial detection system. In *International Conference on Web Information Systems and Mining*, pages 35–43. Springer, 2011.

[91] Yingmin Li, Zheqian Wu, and Huiguo Chen. A grid-based method to represent the covariance structure for earthquake ground motion. *Mathematical Problems in Engineering*, 2012, 2012.

[92] Zhengxin Li. Exact indexing of time series under dynamic time warping. *arXiv preprint arXiv:2002.04187*, 2020.

[93] Feng-Cheng Lin, Huu-Huy Ngo, and Chyi-Ren Dow. A cloud-based face video retrieval system with deep learning. *The Journal of Supercomputing*, pages 1–21, 2020.

[94] Cailiang Liu, Dong Wang, Jun Zhu, and Bo Zhang. Learning a contextual multi-thread model for movie/tv scene segmentation. *IEEE transactions on multimedia*, 15(4):884–897, 2013.

[95] Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger, and Dawn Leaf. Nist cloud computing reference architecture. *NIST special publication*, 500(2011):1–28, 2011.

[96] Zhu Liu and Yuan Wang. Tv news story segmentation using deep neural network. In *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–4. IEEE, 2018.

[97] Nicolas Louis. *Indexation cross-média vidéo/son des contenus multimédia numériques*. PhD thesis, Bordeaux 1, 2006.

[98] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[99] Jinna Lv, Bin Wu, Shuai Yang, Bingjing Jia, and Peigang Qiu. Efficient large scale near-duplicate video detection base on spark. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 957–962. IEEE, 2016.

[100] Jones M. Tim. Recurrent neural networks deep dive. 2017. https://developer.ibm.com/articles/cc-cognitive-recurrent-neural-networks/. Accessed: August 2020.

[101] Gaël Manson and Sid-Ahmed Berrani. Content-based video segment reunification for tv program extraction. In *2009 11th IEEE International Symposium on Multimedia*, pages 57–64. IEEE, 2009.

[102] Gaël Manson and Sid-Ahmed Berrani. Automatic tv broadcast structuring. *International journal of digital multimedia broadcasting*, 2010, 2010.

[103] Ilias Mavridis and Helen Karatza. Performance evaluation of cloud-based log file analysis with apache hadoop and apache spark. *Journal of Systems and Software*, 125:133–151, 2017.

[104] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25, 2015.

[105] Ian McLoughlin, Haomin Zhang, Zhipeng Xie, Yan Song, Wei Xiao, and Huy Phan. Continuous robust sound event classification using time-frequency features and deep learning. *PloS one*, 12(9):e0182309, 2017.

[106] Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 2018.

[107] Nicholas Merrill and Azim Eskandarian. Modified autoencoder training and scoring for robust unsupervised anomaly detection in deep learning. *IEEE Access*, 2020.

[108] Shervin Minaee, Imed Bouazizi, Prakash Kolan, and Hossein Najafzadeh. Adnet: Audio-visual convolutional neural network for advertisement detection in videos. *arXiv preprint arXiv:1806.08612*, 2018.

[109] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *arXiv preprint arXiv:2001.05566*, 2020.

[110] Hemant Misra, Shajith Ikbal, Hervé Bourlard, and Hynek Hermansky. Spectral entropy based feature for robust asr. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–193. IEEE, 2004.

[111] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[112] Jan Nesvadba. *Segmentation sémantique des contenus audio-visuels*. PhD thesis, Bordeaux 1, 2007.

[113] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*, 2018.

[114] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[115] Rezvan Pakdel. *Cloud-based machine learning architecture for big data analysis*. PhD thesis, University College Cork, 2019.

[116] Costas Panagiotakis and Georgios Tziritas. A speech/music discriminator based on rms and zero-crossings. *IEEE Transactions on multimedia*, 7(1):155–166, 2005.

[117] Zikopoulos Paul, Krishnan Parasuraman, Thomas Deutsch, James Giles, and David Corrigan. Harness the power of big data the ibm big data platform. *McGraw Hill Professional*, 2012.

[118] Jose Pio Pereira, Sunil Suresh Kulkarni, Oleksiy Bolgarov, Prashant Ramanathan, Shashank Merchant, and Mihailo Stojancic. Tv content segmentation, categorization and identification and time-aligned applications, November 29 2016. US Patent 9,510,044.

[119] Rade Petrovic, Babak Tehranchi, Kanaan Jemili, Joseph M Winograd, and Dean Angelico. Media monitoring, management and information system, May 9 2017. US Patent 9,648,282.

[120] Stjepan Picek, Annelie Heuser, Alan Jovic, Shivam Bhasin, and Francesco Regazzoni. The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(1):1–29, 2019.

[121] Aggelos Pikrakis, Theodoros Giannakopoulos, and Sergios Theodoridis. Gunshot detection in audio streams from movies by means of dynamic programming and bayesian networks. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 21–24. IEEE, 2008.

[122] Julien Pinquier and Régine André-Obrecht. Jingle detection and identification in audio documents. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages iv–iv. IEEE, 2004.

[123] Daniel Pop. Machine learning and cloud computing: Survey of distributed and saas solutions. *arXiv preprint arXiv:1603.08767*, 2016.

[124] Samira Pouyanfar, Yudong Tao, Anup Mohan, Haiman Tian, Ahmed S Kaseb, Kent Gauen, Ryan Dailey, Sarah Aghajanzadeh, Yung-Hsiang Lu, Shu-Ching Chen, et al. Dynamic sampling in convolutional neural networks for imbalanced data classification. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, pages 112–117. IEEE, 2018.

[125] Rohit Prabhavalkar, Raziel Alvarez, Carolina Parada, Preetum Nakkiran, and Tara N Sainath. Automatic gain control and multi-style training for robust small-footprint keyword spotting with deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4704–4708. IEEE, 2015.

[126] Wei Qi, Lie Gu, Hao Jiang, Xiang-Rong Chen, and Hong-Jiang Zhang. Integrating visual, audio and text analysis for news video. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 3, pages 520–523. IEEE, 2000.

[127] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 262–270, 2012.

[128] Mona Ramadan. *Video Analysis by Deep Learning.* PhD thesis, University of Pittsburgh, 2019.

[129] António Ramires, Diogo Cocharro, and Matthew EP Davies. An audio-only method for advertisement detection in broadcast television content. *arXiv preprint arXiv:1811.02411*, 2018.

[130] M Mazhar Rathore, Hojae Son, Awais Ahmad, Anand Paul, and Gwanggil Jeon. Real-time big data stream processing using gpu with spark over hadoop ecosystem. *International Journal of Parallel Programming*, 46(3):630–646, 2018.

[131] Charles Ringer and Mihalis A Nicolaou. Deep unsupervised multi-view detection of video game stream highlights. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*, pages 1–6, 2018.

[132] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[133] Mickael Rouvier, Benoit Favre, Meriem Bendris, Delphine Charlet, and Geraldine Damnati. Scene understanding for identifying persons in tv shows: beyond face authentication. In *2014 12th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–6. IEEE, 2014.

[134] Gowrisree Rudraraju, ShubhaDeepti Palreddy, Baswaraj Mamidgi, Narayana Rao Sripada, Y Padma Sai, Naveen Kumar Vodnala, and Sai Praveen Haranath. Cough sound analysis and objective correlation with spirometry and clinical diagnosis. *Informatics in Medicine Unlocked*, 19:100319, 2020.

[135] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[136] Eihab Saatialsoruji. *Improving the Performance of Video Processing on Hadoop Clusters*. PhD thesis, Carleton University, 2020.

[137] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Adaptive Computation and Machine Learning series, 2018.

[138] Garima Sharma, Kartikeyan Umapathy, and Sridhar Krishnan. Trends in audio signal feature extraction methods. *Applied Acoustics*, 158:107020, 2020.

[139] Sagar Sharma. Activation functions in neural networks. *Towards Data Science*, 6, 2017.

[140] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[141] Malcolm Slaney and Andres Hernandez Schafhauser. Audio fingerprint for content identification, February 3 2015. US Patent 8,949,872.

[142] John R Smith, Murray Campbell, Milind Naphade, Apostol Natsev, and Jelena Tesic. Learning and classification of semantic concepts in broadcast video. In *Proceedings of the International Conference of Intelligence Analysis*. Citeseer, 2005.

[143] Krishna Somandepalli, Victor Martinez, Naveen Kumar, and Shrikanth Narayanan. Multimodal representation of advertisements using segment-level autoencoders. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pages 418–422, 2018.

[144] Celso L de Souza, Flávio Luis Cardeal Pádua, Cristiano Fraga Guimarães Nunes, Guilherme Tavares de Assis, and Giani David Silva. A unified approach to content-based indexing and retrieval of digital videos from television archives. 2014.

[145] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015.

[146] IBM Research Editorial Staff. Video scene detection using optimal sequential grouping. 2018. https://www.ibm.com/blogs/research/2018/09/video-scene-detection/. Accessed: September 2020.

[147] Hari Sundaram and Shih-Fu Chang. *Segmentation, structure detection and summarization of multimedia sequences*. Columbia University, 2002.

[148] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[149] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[150] Alaa Tharwat. Classification assessment methods. *Applied Computing and Informatics*, 2020.

[151] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

[152] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, 2020.

[153] Jozef Vavrek, Peter Fecil'ak, Jozef Juhár, and Anton Čižmár. Classification of broadcast news audio data employing binary decision architecture. *Computing and Informatics*, 36(4):857–886, 2017.

[154] Francisco Vega, José Medina, Daniel Mendoza, Víctor Saquicela, and Mauricio Espinoza. A robust video identification framework using perceptual image hashing. In *2017 XLIII Latin American Computer Conference (CLEI)*, pages 1–10. IEEE, 2017.

[155] Ramarathnam Venkatesan, S-M Koon, Mariusz H Jakubowski, and Pierre Moulin. Robust image hashing. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 3, pages 664–666. IEEE, 2000.

[156] Vivek Singh Verma and Rajib Kumar Jha. An overview on digital image watermarking.

[157] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[158] Hanli Wang, Fengkuangtian Zhu, Bo Xiao, Lei Wang, and Yu-Gang Jiang. Gpu-based mapreduce for large-scale near-duplicate video retrieval. *Multimedia Tools and Applications*, 74(23):10515–10534, 2015.

[159] Jinqiao Wang, Lingyu Duan, Qingshan Liu, Hanqing Lu, and Jesse S Jin. A multimodal scheme for program segmentation and representation in broadcast video streams. *IEEE Transactions on Multimedia*, 10(3):393–408, 2008.

[160] Peng Wang, Rui Cai, and Shi-Qiang Yang. A hybrid approach to news video classification multimodal features. In *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*, volume 2, pages 787–791. IEEE, 2003.

[161] Yao Wang, Zhu Liu, and Jin-Cheng Huang. Multimedia content analysis-using both audio and visual clues. *IEEE signal processing magazine*, 17(6):12–36, 2000.

[162] worldveil. Audio fingerprinting and recognition in python. 2013. https://github.com/worldveil/dejavu.
Accessed: September 2021.

[163] Marcel Worring. Lecture notes: Multimedia information systems.

[164] Xiaomeng Wu and Shin'ichi Satoh. Ultrahigh-speed tv commercial detection, extraction, and matching. *IEEE transactions on circuits and systems for video technology*, 23(6):1054–1069, 2013.

[165] Zuxuam Wu, Ting Yao, Yanwei Fu, and Yu-Gang Jiang. Deep learning for video classification and captioning. In *Frontiers of multimedia research*, pages 3–29. 2017.

[166] Lei Xie, Zhong-Hua Fu, Wei Feng, and Yong Luo. Pitch-density-based features and an svm binary tree approach for multi-class audio classification in broadcast news. *Multimedia systems*, 17(2):101–112, 2011.

[167] Yihui Xiong and Renguang Zuo. Recognition of geochemical anomalies using a deep autoencoder network. *Computers & Geosciences*, 86:75–82, 2016.

[168] Su Xu, Bailan Feng, Zhineng Chen, and Bo Xu. A general framework of video segmentation to logical unit based on conditional random fields. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 247–254, 2013.

[169] Bian Yang, Fan Gu, and Xiamu Niu. Block mean value based image perceptual hashing. In *2006 International Conference on Intelligent Information Hiding and Multimedia*, pages 167–172. IEEE, 2006.

[170] Hsiu-Wei Yang, Linqing Liu, Ian Milligan, Nick Ruest, and Jimmy Lin. Scalable content-based analysis of images in web archives with tensorflow and the archives unleashed toolkit. In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 436–437. IEEE, 2019.

[171] Shuai Yang and Bin Wu. Large scale video data analysis based on spark. In *2015 International Conference on Cloud Computing and Big Data (CCBD)*, pages 209–212. IEEE, 2015.

[172] Xiaoling Yang, Baohua Tan, Jiehua Ding, Jinye Zhang, and Jiaoli Gong. Comparative study on voice activity detection algorithm. In *2010 International Conference on Electrical and Control Engineering*, pages 599–602. IEEE, 2010.

[173] Wang Yao, Liu Zhu, and Huang Jin-Cheng. Multimedia Content Analysis. 2000.

[174] Muhammad Usman Yaseen, Ashiq Anjum, Mohsen Farid, and Nick Antonopoulos. Cloud-based video analytics using convolutional neural networks. *Software: Practice and Experience*, 49(4):565–583, 2019.

[175] Aggoun Youcef. Procédé de traitement, de présentation et de visualisation synoptiques de l'information pour les décideurs, 2011. Algeria Patent.

[176] Sonia Yousfi, Sid-Ahmed Berrani, and Christophe Garcia. Deep learning and recurrent connectionist-based approaches for arabic text recognition in videos. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1026–1030. IEEE, 2015.

[177] Oussama Zayene, Mathias Seuret, Sameh M Touj, Jean Hennebert, Rolf Ingold, and Najoua E Ben Amara. Text detection in arabic news video based on swt operator and convolutional auto-encoders. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 13–18. IEEE, 2016.

[178] Haitao Zhang, Jin Yan, and Yue Kou. Efficient online surveillance video processing based on spark framework. In *International Conference on Big Data Computing and Communications*, pages 309–318. Springer, 2016.

[179] Jing Zhang. *Extraction of text objects in image and video documents*. University of South Florida, 2012.

[180] Weishan Zhang, Liang Xu, Pengcheng Duan, Wenjuan Gong, Qinghua Lu, and Su Yang. A video cloud platform combing online and offline cloud computing technologies. *Personal and Ubiquitous Computing*, 19(7):1099–1110, 2015.

[181] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52, 2010.

[182] Weiyu Zhu, Candemir Toklu, and Shin-Ping Liou. Automatic news video segmentation and categorization based on closed-captioned text. In *ICME*, page 829–832, 2001.

[183] Wenwu Zhu, Peng Cui, Zhi Wang, and Gang Hua. Multimedia big data computing. *IEEE multimedia*, 22(3):96–c3, 2015.

[184] Tarek Zlitni, Bassem Bouaziz, and Walid Mahdi. Automatic topics segmentation for tv news video using prior knowledge. *Multimedia Tools and Applications*, 75(10):5645–5672, 2016.

[185] Tarek Zlitni and Walid Mahdi. A visual grammar approach for tv program identification. *arXiv preprint arXiv:1301.2200*, 2013.

Abstract: Information extraction from multimedia content is a challenging task. In this thesis, we present an architecture of multimedia contents classification system that provides different phases to extract semantic information from broadcasted streams, starting with the segmentation process, news topics extraction, and advertisement detection and classification. Next, we give an extension to our framework and describes an audio-based hybrid model for content classification combining different deep neural networks with auto-encoder applied to advertisement detection in TV broadcast. Our models achieve high levels of precision. The last contribution consists of a distributed architecture based on the Kafka and Spark frameworks which offer parallel processing of TV streams, we demonstrate through this work the scalability and robustness of this architecture.
Keywords: Multimedia processing; Parallel processing; Deep learning; TV stream analysis; News identification; Advertisement extraction; Media monitoring

Résumé : L'extraction d'informations à partir de contenus multimédias est une tâche difficile. Dans cette thèse, nous présentons une architecture d'un système pour la classification des contenus multimédia qui fournit différentes phases pour extraire des informations sémantiques des flux diffusés, en commençant par le processus de segmentation, l'extraction de sujets d'actualité et la détection des publicités. Ensuite, on propose une extension de ce système consiste en un modèle hybride basé sur l'audio pour la classification de contenu combinant différents réseaux de neurones profonds avec un auto-encodeur appliqué à la détection de publicité diffusée sur la télévision. Les modèles proposés ont atteint des taux de succès très élevés. La dernière contribution consiste en une architecture distribuée basée sur les frameworks Kafka et Spark qui offrent un traitement parallèle des flux TV, nous démontrons par ce travail l'évolutivité et la scalabilité de cette architecture.
Mots clés : Multimedia processing; Parallel processing; Deep learning; TV stream analysis; News identification; Advertisement extraction; Media monitoring

ملخص : يعد استخراج المعلومات من محتوى الوسائط المتعددة مهمة صعبة. ففي هذه الرسالة، نقدم بنية نظام لتصنيف محتويات الوسائط المتعددة والتي توفر مراحل مختلفة لاستخراج المعلومات الدلالية من تدفقات البث، بدءًا من عملية التجزئة واستخراج الموضوعات الموضعية وإعلانات الكشف. بعد ذلك، يُقترح امتداد هذا النظام ليتألف من نموذج هجين قائم على الصوت لتصنيف المحتوى يجمع بين شبكات عصبية عميقة مختلفة مع مشفر تلقائي مطبق على الكشف عن الإعلانات التي يتم بثها على التلفزيون. لقد حققت النماذج المقترحة معدلات نجاح عالية جدًا. تتكون المساهمة الأخيرة من بنية موزعة تستند إلى أطر عمل كافكا وسبارك التي تقدم معالجة متوازية لتدفقات التلفزيون، ونبين من خلال هذا العمل قابلية التوسع وقابلية التوسع في هذه البنية.
الكلمات الدالة : معالجة الوسائط المتعددة؛ المعالجة المتوازية؛ تعلم عميق؛ تحليل تيار التلفزيون؛ تحديد الأخبار؛ استخراج الإعلانات؛ رصد وسائل الإعلام