NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

PROGRAM OF POSTGRADUATE STUDIES

Msc THESIS

Multimodal video classification with deep neural networks

Nikiforos I. Pittaras

Advisor: **Perantonis Stavros**, Research Director, NCSR-D

ATHENS

September 2018

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Πολυτροπική κατηγοριοποίηση βίντεο με βαθιά νευρωνικά δίκτυα

Νικηφόρος Ι. Πιτταράς

Επιβλέπων: **Περαντώνης Σταύρος**, Διευθυντής Έρευνας, ΕΚΕΦΕ-Δ

ΑΘΗΝΑ

Σεπτέμβριος 2018

Msc THESIS

Multimodal video classification with deep neural networks

Nikiforos I. Pittaras
S.N.: 1422

SUPERVISOR: **Perantonis Stavros**, Research Director, NCSR-D

EXAMINATION COMMITTEE: **Perantonis Stavros**, Research Director, NCSR-D
Giannakopoulos Theodoros, Postdoc Researcher, NCSR-D

Examination Date: September 17 2018

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Πολυτροπική κατηγοριοποίηση βίντεο με βαθιά νευρωνικά δίκτυα

Νικηφόρος Ι. Πιτταράς
Α.Μ.: 1422

ΕΠΙΒΛΕΠΩΝ: **Περαντώνης Σταύρος**, Διευθυντής Έρευνας, ΕΚΕΦΕ-Δ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:
Περαντώνης Σταύρος, Διευθυντής Έρευνας, ΕΚΕΦΕ-Δ
Γιαννακόπουλος Θεόδωρος, Μεταδιδ. Ερευνητής, ΕΚΕΦΕ-Δ

Ημερομηνία Εξέτασης: 17 Σεπτεμβρίου 2018

ABSTRACT

The recent abundance of video data, automatic video classification tools have become important components in multiple video machine learning tasks. Given the rich multimodal qualities of video, it offers a variety of information sources that can be utilized to further aid classification. In this study we examine research questions adhering to the effect of the visual, audio and temporal video modalities on video classification. To process the visual and audio modalities, we extract frame and audio spectrogram sequences from random video segments. We adopt a shared deep representation approach for the visual and audio data, using deep features extracted from a fully-connected layer of Alexnet-based DCNN. Regarding multimodal fusion, we examine a variety of early direct-fusion methods, i.e. approaches that aggregate information from the visual and audio modality into a single, multimodal representation. Specifically, we use averaging, concatenation and max pooling. In addition, we attempt to apply sequence bias methods borrowed from image description, which we call input-bias and state-bias fusion. Finally, we perform a late fusion of video-level classification scores, examining linear combination and max pooling of the marginal predictions. Regarding the temporal information present in video data, we examine its contribution by comparing the fully-connected, feed-forward softmax classification layer – which processes input sequence in an aggregation-based manner – to the sequence-aware LSTM model that is sensitive to and able to model temporal input inter-dependencies. We apply these approaches (named FC and LSTM workflows, respectively) both in separate visual and audio modality data and in the multimodal fusion schemes. A set of experimental evaluations are performed on multiple video classification datasets to examine the performance of each research question. The experimental results indicate that the LSTM workflow performs better on visual data, with the FC approach faring better on the audio modality. The relationship between the visual and audio modalities relies on the underlying dataset and annotation, as reflected by the superiority of the audio modality in the audio-inclined Audioset, and its inferior results, compared to the visual modality, in the other datasets in the multimodal experiments. Regarding multimodal fusion approaches, results show that simple late late-video linear combination fusion works best, despite its practical disadvantages with the maximum pooling variant performing close to single-modality baselines. Excluding that, averaging or concatenation of modality encodings works best for the FC and LSTM workflows respectively, while the sequence-bias approaches do not perform as well as in the image description task. We verify the complementarity of the visual and audio modalities, with multimodal techniques outperforming single-modality baselines per dataset, extract guidelines towards achieving it and establish a multimodal DNN baseline per dataset and workflow.

SUBJECT AREA: Machine Learning

KEYWORDS: Machine Learning, Neural Networks, Multimodal, Classification, Deep Learning

ΠΕΡΙΛΗΨΗ

Η πρόσφατη ραγδαία αύξηση και αφθονία των πολυμεσικών δεδομένων καθιστά αναγκαία τη χρήση αυτόματων εργαλείων κατηγοριοποίησης σε σχετικές εφαρμογές μηχανικής μάθησης. Η πλούσια πολυτροπικότητα (multimodality) των τελευταίων παρέχει πλήθος πηγών πληροφορίας προς χρήση και υποβοήθηση της διαδικασίας κατηγοριοποίησης. Στην παρούσα μελέτη εξετάζουμε ερευνητικά ερωτήματα σχετικά με την επιρροή της οπτικής, ακουστικής και χρονικής πληροφορίας ενός βίντεο, στην κατηγοριοποίησή του. Εξάγουμε καρέ και φασματογράμματα, υιοθετώντας μία βαθιά αναπαράσταση βασισμένη στο συνελικτικό νευρωνικό δίκτυο Alexnet και αξιολογούμε πολυτροπικές προσεγγίσεις early fusion μεθόδων, που συγχωνεύουν το οπτικό και το ακουστικό κανάλι σε μία πολυτροπική αναπαράσταση. Επιπλέον, εξετάζονται μέθοδοι προδιάθεσης (bias) οπτικών δεδομένων με τη συγχωνευμένη ακουστική πληροφορία, εμπνευσμένες από τεχνικές περιγραφή εικόνας. Τέλος, εφαρμόζουμε συγχώνευση των σκορ κατηγοριοποίησης σε επίπεδο βίντεο, μέσω γραμμικού συνδυασμού και συγχώνευσης μεγίστου. Για τη χρονική πληροφορία, συγκρίνουμε τη συγχώνευση πληροφορίας (αρχιτεκτονική FC βασισμένη στο νευρωνικό ταξινομητή πλήρους σύνδεσης και της συγχώνευσης softmax) από το επίπεδο των καρέ σε αυτό ολόκληρης της αλληλουχίας, και της αρχιτεκτονικής LSTM, που ενσωματώνει απευθείας χρονικές αλληλοεξαρτήσεις της εισόδου. Εφαρμόζουμε τα δύο μοντέλα σε οπτική και ακουστική πληροφορία, καθώς και στις τεχνικές πολυτροπικής κατηγοριοποίησης. Στη συνέχεια εκτελούμε πειραματική αξιολόγηση σε πολλαπλά σύνολα δεδομένων για να αξιολογήσουμε τις παραπάνω μεθόδους και τα ερευνητικά ερωτήματα. Τα αποτελέσματα δείχνουν πως η LSTM τεχνική υπερτερεί της FC σε οπτικά δεδομένα, ενώ το αντίθετο ισχύει σε δεδομένα φασματογραμμάτων ήχου. Η επιλογή χρήσης της οπτικής ή της ακουστικής πληροφορίας εξαρτάται από το σύνολο δεδομένων και τον τύπο των κλάσεων, όπως φαίνεται από την συγκριτικά καλύτερη απόδοση του ήχου στο Audioset, και την υποδεέστερη απόδοση στα υπόλοιπα σύνολα δεδομένων, στα πολυτροπικά πειράματα. Σχετικά με τις πολυτροπικές τεχνικές, η απλή συγχώνευση σε επίπεδο βίντεο μέσω γραμμικού συνδυασμού δίνει βέλτιστα αποτελέσματα παρά τα πρακτικά μειονεκτήματά της, ενώ η συγχώνευση μεγίστου δίνει έχει απόδοση πολύ κοντά στις μη πολυτροπικές προσεγγίσεις. Η απλή συγχώνευση μέσου όρου και επιθέματος των οπτικοακουστικών δεδομένων δίνει βέλτιστα αποτελέσματα στην FC και LSTM τεχνική αντίστοιχα. Αντίθετα, οι τεχνικές προδιάθεσης αλληλουχιών δεν φαίνεται να εφαρμόζονται με την ίδια επιτυχία που έχουν στην περιγραφή εικόνας. Επιβεβαιώνουμε τη συμπληρωματικότητα τού οπτικού και ακουστικού καναλιού, με τις πολυτροπικές τεχνικές να υπερτερούν των προσεγγίσεων με μία πηγή πληροφορίας, εξάγουμε βασικές κατευθύσεις για επίτευξή αυτής της βελτίωσης, και προσφέρουμε ένα baseline για την απόδοση πολυτροπική τεχνικών, ανά σύνολο δεδομένων που εξετάζουμε.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Μηχανική Μάθηση

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Μηχανική Μάθηση, Νευρωνικά Δίκτυα, Πολυτροπικότητα, Κατηγοριοποίηση, Βαθιά Μάθηση

ACKNOWLEDGEMENTS

I would like to thank my thesis advisors, Dr. S. Perantonis and Dr. T. Giannakopoulos for their invaluable help, guidance and support on the project. In addition, I am grateful to Dr. G. Giannakopoulos who was always available to provide useful feedback on various aspects of the project. Finally, this thesis would not be possible without my parents and C. Themeli, and their steady love and support throughout my studies.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

This document is an investigation of artificial intelligence methods, that attempt to efficiently tackle the video classification task, i.e. assigning meaningful labels to video clips with respect to their content. For example, a video of a music concert could include labels such as “crowd”, “music”, or, obviously, “concert”. The purpose of an automatic video classification system is to be able to correctly produce such output, given a video as input, with no human intervention after its construction. In this study, a set of approaches are investigated that build such systems by taking advantage of three main components commonly found in videos: the visual content, the audio content, and the temporal interdependencies of their parts. For example, video frames of a basketball trajectory are not independent of each other – an airborne ball must have originated in a player’s hands. Likewise, the sound of the crowd cheering in a basketball game is likely to succeed a three-pointer “swish”. Lastly, both of the two preceding audiovisual events in a video are strong evidence of the label “basketball game”. The aim of the study is to examine the effect and contribution of each of the above components or “modalities”, on video classification. Specifically, two avenues of considering the temporal dependencies are investigated, and are applied in visual, audio and “multimodal” (that is, a combination of both audio and visual content) video inputs, along with a consideration of various parameters and architectural technical details of the underlying models. All of these approaches constitute machine learning models, in the sense that the underlying mechanisms required to produce correct classification results are learned automatically from supplied labelled data via a processed called “supervised learning”. To examine if these trained models can manage video classification efficiently, they are applied on new, unseen videos. Using available datasets compiled for video classification benchmarking, we evaluate the above procedures, with the resulting experimental outputs giving rise to conclusions that address the underlying research questions.

This work builds on previous studies on multimodal video classification. It verifies existing findings, examines established research questions under new scenarios and investigates new approaches for modality fusion for the task, all via a large experimental evaluation process and subsequent extraction of interesting, usable results. The thesis was compiled towards the fulfillment of the graduation requirements for the Signal and Information Processing and Learning postgraduate program of the Department of Informatics and Telecommunications, of the National and Kapodistrian University of Athens in Greece.

1. INTRODUCTION

In the first chapter we introduce the multimodal video classification problem, which is the object of study in this thesis. In section 1.1, we explain the motivation behind addressing the video classification in an automated fashion, using advances and acquired knowledge from artificial intelligence and machine learning. We move on to provide a reasoning for adopting a multimodal approach for processing video data with the aforementioned approach, in section 1.1.2. Furthermore, we provide a more precise formulation of the video classification problem in section 1.2, which is the setting by which we will tackle the problem in later chapters in this document. Finally, we close this chapter by first establishing specific research goals which this study will attempt to address, and a description of the structure the thesis document will adhere to.

1.1 Motivation

This chapter provides evidence on the motivation of the problem at hand, namely the purpose and applications of video classification. In addition, we report the two important component of the task. First, its implementation in an automated manner and secondly, its utilization of multiple data modalities.

1.1.1 The need for automatic video classification

A number of fairly recent factors have brought about a tremendous increase in the availability and consumption of multimedia content. The surge of popularity of the Internet, the proliferation of camera-equipped smartphones, IoT and embedded camera systems have contributed to videos participating in this trend [16]. The resulting amount of videos prevents their efficient handling and processing by humans alone. This problem has led to the necessity of developing automated tools in order to aid organization, handling and accessing of the available video content. One way such a system can contribute to the aforementioned operations is to annotate videos with predefined meaningful labels, categories and semantic indexes, corresponding to video content or metadata. These annotations can subsequently be utilized for a variety of tasks such as video retrieval and indexing. These automated systems will thus act as classification machines, performing the task of automatic video classification.

A key factor towards the development of such tools has been the application of artificial intelligence research to the video and multimedia domains. Specifically, the utilization of machine learning practices [4, 64], coinciding with the aforementioned wealth of available data, has enabled the construction of data-driven classification models. While traditional artificial intelligence methods would rely on explicit rules designed by experts, machine learning approaches construct models that learn the characteristics and the building blocks of the desired categorization directly from annotated data. The latter approach can reduce

human intervention and required fine-tuning by a considerable degree.

1.1.2 Video multimodality

In the field of semiotics, a *mode* or *modality* is refers to an information channel and/or encoding scheme utilized to convey meaning in communication [65]. For example, spoken language, a photograph or a sequence of symbols are all modes of communication that can be used to construct a message. A video is a *multimodal* message, i.e. it is an information package that contains multiple modalities. These are, for example, the visual component of the video consisting of the visual stimuli transmitted to the viewer during playback, the audio track (e.g. music, dialogue, ambient effects), as well as metadata that might accompany the file, such as the video title, subtitles, transcript and others. Apart from these straight-forward information sources, one can apply more refined extraction on the video content, arriving at higher level separable information sources in a video. For example, extracting, separating and tracking dialogue and ambient noise could aid audio classification tasks, or differentiating between visual background from objects of interest in a scene would be a helpful approach in object detection.

Unless all modalities contain identical information content, it follows that an intelligent system seeking to reach an informed decision with respect to that video would want to utilize all available information sources, therefore applying an information extraction process to every modality. Combining this information should yield results useful for the task at hand, and if said combination is performed effectively, these results should be qualitatively superior to corresponding outputs when only a single video modality is exploited. Given this intuition, in this study we seek to take advantage of the inherent multimodal nature of video data towards aiding classification, as explained in the following section.

1.2 Problem definition

In this thesis, we examine the video classification problem, considering the aforementioned multimodal nature of video data. Given a set of predefined video labels or classes $C = \{c_1, c_2, \ldots, c_c\}$ and an input video v_i that belongs to some $c_i \in C$, the goal is to discover c_i, given v_i. We approach the problem as a machine learning task. This entails the construction of a classification model which, learning from available training data consisting of video / label ground truth pairs, will be able to output a label c_i for a input video. The output label is a product of a decision, reached by exploiting prior knowledge acquired during the training process. The model architecture we will use is a deep artificial neural network, a popular learning model in the classification literature [108] as well as various other machine learning tasks. Given that a video is a multimodal object, we exploit a number of useful modalities in the following ways. The contribution of the temporal modality is investigating by examining two classification architectures: first, we apply a simple, fully-connected, feed-forward neural layer, followed by a softmax operator to enforce probabilistic results (see section 3.2 for an explanation). This model is followed by

simple aggregation mechanisms, applying simple fusion from sequence-level scores into video-level predictions. On the other hand, we apply a deep neural architecture that accommodates the temporal input dimension (e.g. the video duration, in our case) in a more refined manner, capturing possible sequence inter-dependencies that may exist directly. The intuition to these two approaches is that in cases where the temporal component of the input is important to classification (i.e. dependencies between sequence members exist and are useful), the sequence-based model is expected to produce good results. We will examine all of the above in more detail in section 4.2.

The remaining modalities we exploit are the audio and visual modalities, which are common channels utilized in the video classification literature [116]. The first consists of the video frames, while the second contains the audio portion of the video. Given these two modality-specific views of a video object, we will look into modality fusion schemes that combines this partial information channels to produce a single label prediction for the video. We investigate the multimodal part of this study in section 4.3.

1.3 Thesis Goals

Given the two main components of the study, we arrive at two closely related goals we wish to tackle in this thesis: First, we seek to investigate the capability of sequential neural models versus simple aggregation-based approaches, to handle temporal context for the video classification task, with the former modelling temporal inter-dependencies into its internal representations, while the latter applying simple aggregation mechanisms. Secondly, we explore the contribution of the audio and visual modalities in the video classification task. Specifically, we compare the classification performance achievable with each modality alone, subsequently moving on to test a number of multimodal fusion strategies that combine these two modalities in a variety of ways. We perform both of these comparisons for both types of deep neural models.

We run a set of experiments in order to address the above goals, on multiple, diverse datasets both in respect of content, class set, specific classification domain, content / annotation noise, and many more. These are covered in detail in chapter 5 for more details. By these generic evaluation benchmarks, we strive to establish a general performance baseline one can expect to reach by using multimodal features with the proposed classification models, rather than producing the highest possible performance tuned for each dataset.

1.4 Thesis Structure

This study is structured as follows. In chapter 2 we present related approaches and recent work on video classification. We include methods that enforce a multimodal consideration of the video input, using visual, audio, temporal or multimodal approaches to the learning process.

A description of the proposed method to address our stated goals follows in chapter 4. There, we begin with a introduction to classification, neural networks and deep learning before moving on to the presentation of our proposed workflows, namely the feed-forward and sequential deep neural models in section 4.2, and the multimodal fusion approaches in section 4.3.

What follows is the experimental evaluation. We present this in chapter 5, describing the datasets and implementation details in section 5.1.1 and a set of preliminary experiments in section 5.2. The main experimental setting, results and discussion with respect to the state goals of the study can be found in sections 5.3 and 5.4.

We will conclude by a summary of the contributions of this thesis in chapter 6, along with potential future work that could complement and extend the investigation of this project.

2. RELATED WORK

There has been significant work in video classification, exploiting visual, temporal, audio or textual information, as well as metadata, to arrive at a class prediction. In this section we outline related work in the following categories. First, we outline preprocessing methods applied on video data, as a preliminary step prior to feeding them into a learning model. We move on to examine related work related to each modality relevant to this study, beginning by examining approaches that exploit the visual component of a video, laying out hand-crafted and learned feature extraction methods separately. We continue by considering approaches that exploit audio, followed by an examination of works that apply multimodal fusion methods. We close this section by regarding miscellaneous modalities and approaches for the video classification task. The reader is encouraged to refer to surveys (e.g. [14, 91, 135]) for a more detailed recounting of multimodal video classification.

2.1 Video preprocessing methods

Since video files have duration which can extend to several minutes, partitioning a video to smaller manageable chunks is a common practice.

Shot segmentation [63] is a process that partitions a video into shots. A video shot is a contiguous uninterrupted sequence of frames, usually semantically concise, often taken from a single camera. Shots can vary from abrupt cuts to gradual transitions (e.g. fade in, fade out, dissolve, wipe and others). After clip extraction, the resulting clips can be subsequently processed by aggregation of individual frame information, or processed sequential approaches. Alternatively, keyframe sampling [19] extracts a single or a few frames as representative frames (i.e. *keyframes*) of the video shot.

2.2 Visual-based approaches

Visual-based approaches to video classification analyze the frames of the video for structure and information that is expected aid discrimination. In this sense, they exploit advances in image classification and content-based image retrieval. In the following sections we outline related work on visual features extraction methods.

2.2.1 Handcrafted Feature approaches

Multiple hand-crafted features are devised to generate efficient features for image classification. Global image features take the entirety of the image into account, producing responses related to the overall texture (e.g. Local Binary Pattern [92] and histogram of oriented gradients [20]), the color distribution in the image ([96]) and others.

On the other hand, local features describe local image patches, usually residing around a point of interest (i.e. a “keypoint”) in the image. Keypoints are usually corners in an image, discovered by detectors such as FAST [102], SURF [8], Harris [40], multi-scale difference of Gaussians [81], and others [110, 86]. After keypoint detection, the surrounding area can be represented with a feature vector by a keypoint descriptor such as SIFT [81], SURF [8], KAZE [2], BRIEF [15] and ORB [103]. Other strategies apply the description in color channels in the image [83], local and global features are combined in an aggregation scheme [78, 76], or apply a scale invariance mechanism via a pyramidal multi-resolution scheme [35].

The local feature extraction process is followed by an aggregation scheme, which combines all local features into a feature vector for the entire image. In [114], the image retrieval task is modeled in a bag-of-words scheme, where “visual words” are formed by clustering local features. In [95] the authors use the Fisher Kernel [94] with various normalization functions over SIFT features fed to linear SVMs, while in [6, 21] a local descriptor aggregation scheme is used which seems to increase performance by alleviating noise from the previous approach.

Global color features have been extensively used [100, 99, 28, 127] for video classification, due to their relatively low computational cost, usually combined with features from other modalities.

Local features have been used in machine learning tasks. Due to their increased computational costs, shot segmentation sampling is a common procedure (e.g. selecting a few frames as the clip representative) for concept detection [49].

2.2.2 DNN-feature approaches

While hand-crafted features have been the state of the art until recently, the re-emergence of neural networks, sparked by advances in hardware [108], led to the construction, training and successful application of deep neural models. These networks are capable of achieving an impressive performance boost compared to hand-crafted features in multiple machine learning tasks, as well as producing layer responses that act as deep and rich features [10, 11]. The latter can be used as input for meta-learning stages further improving performance (e.g. fed to logistic regression or SVM classifiers [97]), thus imbuing DNN models with additional value in the machine learning community. There has been therefore an extensive use of DNNs for visual classification lately, incorporating the success of deep convolutional networks in image recognition [66, 113, 124, 41]. In the context of video classification and visual features, these networks are applied in video frames along with an aggregation mechanism so as to pool information from the frame level to the video level. Another approach is to use recurrent networks to capture information regarding the temporal component inherent in video content directly [121, 140, 23]. For a comprehensive review, see [135].

In [121] the authors use an LSTM network as an unsupervised sequence encoder, compressing video frames or DCNN-produced frame representations to a single vector repre-

sentation. The authors report modest improvements, when applying the model to supervised classification on the UCF-51 and HMBD51 datasets.

In [54], the authors modify DCNNs, extending the convolution kernel to the temporal dimension. Their model extracts spatiotemporal features by applying convolution, subsampling and pooling operations to separate channels, i.e. temporally neighbouring contiguous video frames. A final representation is constructed by feature combination of responses from all channels. Experimental results on video action recognition on TRECVID [115] and KTH [109] datasets show that the proposed approach outperforms the state of the art only on the former dataset.

In [58], the authors experiment with different ways to introduce temporal information to DCNNs, namely a single-frame model, a late fusion of frames with a fixed temporal distance and two variants of early frame fusion (namely early and slow fusion, with the latter propagating marginal frame information in a slower manner). Experiments on UCF-101 and Sports-1M datasets show that the the approach is "not particularly sensitive to architectural details of the connectivity in time", and the slow early fusion variant consistently outperforms its competition. They note that the single-frame baseline showcases a very strong performance, which hints to either local motion information not being exceptionally important for classification of these datasets, or a more detailed handling is required.

Donahue et al. [23] apply DCNNs for feature vector generation, feeding collections of frame vectors to an LSTM for video classification. Compared to a single-frame softmax classification with voting aggregation, the experimental results on the UCF-101 dataset show of the sequential LSTM approach faring better in terms of accuracy.

In [140], Ng et al. examine DCNNs with a variety of spatial and temporal pooling approaches, as well as an DCNN - LSTM combination. Their work focuses on processing long video clips (reporting processing up to 120-frame sequences). Experiments on UCF-101 show considerable (approximately 10%) improvement to previous approaches that do not utilize motion information. However, they stress the latter is necessary for benchmarks like the UCF-101 dataset for achieving state of the art results.

2.3 Audio-based approaches

Audio-based approaches isolate the audio content of the video, followed by the extraction of useful audio features for classification. Many studies incorporate handcrafted features on the audio content or its segments. In [80, 79], the authors use a variety of low-level statistical global audio features to distinguish 5 generic television program categories, using HMM classification and ISODATA clustering. In [100, 99], the authors use MFCC features with a GMM for video genre classification. Audio statistics are employed in various works [28, 38], usually alongside additional visual and metadata features. Other approaches study the audio content in the frequency domain, primarily via application of Fourier analysis on the audio signal [3], or use the mpeg-7 feature suite [59] for audio analysis. A number of approaches utilize frequency-based audio information in the format of audio

spectrogram images [5, 29].

In [75], Lee et al. use a variety of representation methods (PLSA, Gaussian standard and mixture models) on top of MFCC features, classified with SVMs with various distance functions. They report better than chance performance, with respect to average precision.

The authors in [74] use convolutional deep belief networks to learn deep features which they demonstrate have some correspondence to phonemes. They apply their representation to genre and artist classification, reporting similar classification accuracy to MFCC features.

In [43], the authors tackle multiple instance learning by generating deep features from a three-layered NN with 500 hidden units per layer. The features are fed to multiple neural classification models and experiments on the Audioset dataset [32] show that DNNs softmax-based attention works best in terms of mAP, AUC and d-prime scores, compared to DNN and RNN pooling approaches.

2.4 Multimodal fusion approaches

Given the rich multimodal nature of videos, several approaches utilize multiple modalities, followed by a fusion / aggregation scheme to combine all information into a single video prediction. Multimodal approaches can be seen as a special case of multi-view learning [122, 136], where each modality composes a distinct view of the multi-modal object.

In [99] the authors combine low-level visual motion with MFCC feature responses, applied for TV program genre classification. They report optimal results with a weighted average combination (assigning a 0.7 bias to the audio) with respect to ROC performance. In [28], global color features, audio statistics and motion information of segmented objects and the camera are combined for video genre classification. A similar approach is examined in [127] for visual and temporal modalities. Snoek et al. [117] investigates fusion schemes for visual, audio and textual modalities in the video concept detection task. Specifically, early and late fusion strategies are investigated, aggregating information in the feature level and semantic (prediction) level, respectively. Using SVM classifiers, they report late fusion giving improved average precision scores, at the cost of additional learning effort.

In [112] the authors apply a two-stream DCNN architecture to separately process visual and temporal context (captured buy optical flow images [130]), mimicking the human visual ventral and dorsal optical pathways [33]. Experiments on UCF-101 [118] and HMDB-51 [67] datasets show the temporal network (using optical flow) outperforming the spatila network, in terms of accuracy by a 10% absolute score. However, there significant complementarity between the spatial and the temporal networks, best achieved by a meta-learning process of SVM fusion on softmax scores.

In [138], the authors also use a two-stream spatiotemporal approach like in [112]. They use two DCNN architectures for deep feature generation (CNN_M [112] and VGG_19 [113]) over two fully-connected layers. Additionally, they examine NN softmax classifications

versus a meta-learning phase with linear SVM classifiers. Experiments on the UCF-101 and CCV [56] datasets in terms of mAP show that the deeper VGG_19 network performs better, if large amounts of training data are available. Model (average) fusion (different architectures on the visual stream) performs poorly when fusing networks with different performance, and spatio-temporal linear combination fusion (same architecture, different modality / stream) works well for both datasets, with the spatial part having the larger contribution. In addition, the authors conclude that softmax fares better than SVM meta-learning.

Wu et al. [134, 133, 55] utilize video frames, optical flow images and audio spectrograms, each modal stream fed first to deep convolutional networks and then to an LSTM network. Modality predictions are fused with a set of methods, within which an adaptive approach is proposed which uses class relationships as a regularization mechanism. Experiments on UCF-101 and CCV datasets verify multimodal and CNN / LSTM complementarity and show the proposed fusion outperforming other aggregation techniques. Compared to multiple recent studies, the authors surpass the state of the art in terms of classification accuracy. In [85], the authors compare a variety of learnable temporal pooling approaches along with a two-stream audiovisual DNN Model, proposing a context gating aggregation approach that outperforms competition in the recent Youtube-8M Large-Scale Video Understanding challenge [1].

2.5 Other approaches

2.5.1 Temporal features

Apart from audiovisual features, temporal qualities exploit the changes occurring within a video clip. Such shifts can occur either by motion of objects of interest in a scene (e.g. a car drives by, a face moves) or by more drastic changes occuring by camera movement or shot change (e.g. a news graphic appears in a news segment, or a cut occurs in a film). In [61] the authors devise a descriptor to capture spatiotemporal information using on 3-dimensional gradients in the video visual stream. In [111], video “tomographs”, e.g. one-dimensional cross-cuts in the temporal dimension, are extracted, to be used as visual input that spans the temporal duration of the video. Optical flow or image velocity [47, 7] is an estimate of temporal changes in a sequence of frames, approximating the two-dimensional motion field from pattern trajectories in the frames by examining relative positions of pixel intensities. Optical flow has been widely used and achieved state of the art results, with respect to temporal features. In [126], the authors utilize a Hidden Markov Model (HMM) [50] on short video segments in order to model the underlying temporal structure. They evaluate their approach on the event detection task on Trecvid MED data, reporting significant average precision gains over the related work. In [130], the authors use SURF descriptors and dense optical flow trajectories [129] for video action recognition in multiple datasets, while in [53] adopts a decomposition strategy, partitioning motion to dominant and residual parts. In [131] optical flow is used in conjunction with

DCNNs on UCF-101 and HMDB51 datasets, fusing DNN and trajectory responses via a spatiotemporal aggregation and pooling schemes.

Yamato et al. [137] use temporal information in human action recognition in frame sequences, by computing mesh features on thresholded visual data and training a HMM for each class. In [52], the authors compute motion features and project them to a one-dimensional signal which is used to train HMMs. They report in binary TV program genre classification. The authors in [28, 127] exploit motion features of the camera (e.g. panning, zooming and cuts), as well as segmented object motion for TV program categorization. A similar approach is investigated in [99], where flow features are computed by tracking pixel-wise frame difference.

2.5.2 Text-based approaches

Text-based approaches use a variety of text metadata that often accompany a video. For example, dialogue transcripts, subtitles, or hearing-impaired captions that contain a documentation of sound effects in the video. In addition, semantic information like tags and partial categories may be available. This textual information can be subsequently used with a text representation model (e.g. bag-of-words vectors [107]) as discriminatory features for classification. Textual information can be extracted if not available in metadata – for example, Dimitrova et al. use an OCR-based text box detection and understanding [22], where text elements in video frames are mined via an image processing approach.

2.5.3 Domain-specific approaches

Other approaches incorporate existing domain-specific knowledge to aid discrimination. In [89], expert knowledge in sound energy dynamics in film is exploited in horror film binary classification. In addition, face detection and tracking is a popular high level feature. Dimitrova et al. [22] apply a face detection and tracking procedure along with OCR-extracted text, using a HMM model for TV program categorization. Face recognition features are used for news videos classification along with text and multimodal features in [132], using SVMs and GMM learning models.

3. BACKGROUND

i

In this chapter, we take the time to introduce some basic concepts related to this thesis. Specifically, section 3.1 will present the artificial neural network model, its composition, operation and functioning mechanisms. We will move on to describe the classification task in section 3.2, adopting a supervised machine learning perspective, which is the approach followed in the rest of the thesis. We conclude the chapter with a discussion on deep learning and popular models for deep learning approaches, in section 3.3.

3.1 Neural networks

Here we provide an overview of the basic concepts around neural networks. For a more in-depth examination, the interested reader is encouraged to see related surveys and studies [119, 26].

3.1.1 Definition

An artificial neural network is a biologically-inspired learning model. It consists of a number of simple interconnected computational units called *neurons*, designed similar to their biological counterparts from the human nervous system. Each neuron produces an output as a function of three quantities: the first is the input stimulus the neuron receives, the second is the neuron's sensitivity to each component of the input (i.e. the corresponding weights) and the third is an activation threshold (i.e. the bias term). The neural output is filtered via a special activation function. This output can affect its environment by causing a reaction by other neurons in the network, thereby enabling a distributed chained computation.

The fundamental logic behind the artificial neuron is found in the *linear threshold unit*, conceived by the work of Mculloch and Pitts on threshold-based computational models [84]. It can be defined by:

$$y = sgn\left(\sum_{j=1}^{d}(w_j x_j + w_0)\right) = sgn\left(\boldsymbol{w}^T\boldsymbol{x}\right) \tag{3.1}$$

There, binary neurons are excited (i.e. they "fire", producing an output value of $fy = 1$) or are inhibited (produce an output of $y = -1$) as a function of their d-dimensional augmented input vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_d, 1]$, a weight-bias augmented vector $\boldsymbol{w} = [w_1, w_2, \ldots, w_d, w_0]$ and an *activation* sign function $sgn(\cdot)$. The activation function $sgn(\cdot)$ acts as a *non-linearity*-inducing operator, with other common candidates being the logistic function (yielding

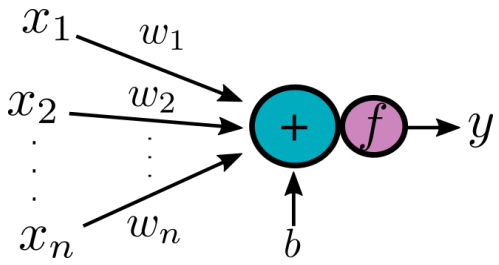

Figure 1: An artificial neuron acting on an input n-dimensional vector x. Each input component is scaled with a quantity w_i and contributes along with the bias term to the linear combination as described in equation 3.1. The final output y is obtained after scaling with an activation function $f(\cdot)$.

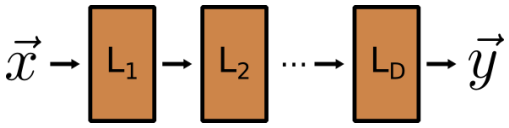

Figure 2: An artificial neural network model, consisting of D layers acting on an input x and producing an output y.

$f(x) = 1/(1 + e^{-x})$, the $tanh(\cdot)$ function or the unit step operator. A visualization of a neuron is shown in figure 1.

3.1.2 The neural network

Neurons processing a single input vector at the same step in the computation chain are said to form a neural layer. The organization of a multitude of layers in a sequential fashion, i.e. connecting the output of the neurons in layer l to the input of neurons in layer $l+1$, forms a computational graph which we call a neural network. See figure 2 for a visualization. Layers between the input and the output layer are called “hidden”, in the sense that they are neither visible from the input nor from the output end of the network. The hidden layers are the ones providing the computational capabilities to the learning model. In fact, a single hidden layer in a neural network can approximate under some simple assumptions [48]. This sequential connection of simple, linear components is what necessitates the inclusion of the non-linearity element mentioned above: non-linearity prevents the neural chains from degenerating into one long linear operation, while simultaneously enabling non-linear, complex functions to be estimated by the network.

3.2 Neural networks for supervised classification

In this work, we examine neural networks as supervised classification models. In the following paragraphs we will define the classification problem, as well as describe the necessary components to train a deep neural model aimed for this task.

3.2.1 The classification problem

Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ a dataset of N elements and $C = \{c_1, c_2, \ldots c_M\}$ a set of predefined M classes (we will use the terms class, label and category interchangeably in this document). A *classification* or *categorization* is the task of mapping each pair $(\boldsymbol{x}_i, c_i)$ to a boolean value $\{T, F\}$, where . A pair $(\boldsymbol{x}_i, c_i)$ mapped to the value T indicates that the data object $\boldsymbol{x}_i$ *belongs to* or *is an instance of* the category c_i. Assuming that the true mapping is computed by a function $f : X \times C \rightarrow \{T, F\}$, the goal of the classification task is to find a function $\hat{f}$ that is as close an approximation of f as possible. If exactly one c_i is set to T for any given $\boldsymbol{x}_i \in X$, as in this study, we speak of single-label classification, as opposed to multi-label classification when the number of categories assigned to an instance exceeds unity.

If we consider X as a data generation source, the latter can be modelled as a conditional distribution $P(\boldsymbol{x}_i|c_j), i \in \{1, \ldots N\}, j \in \{1, \ldots M\}$. Then, the single label classification problem becomes finding a function $\hat{f}$ where for $(\boldsymbol{x}_i, c_i) \in X$, $\hat{f}(\boldsymbol{x}_i, c_i) \rightarrow 1$ and $\hat{f}(\boldsymbol{x}_i, c_j) \rightarrow 0$, $j \neq i$. In other words, $\hat{f}$ represents a probability distribution of the input over the target classes.

3.2.2 Probabilistic neural networks

This is the model adopted by a multilayer probabilistic neural network: let the last layer of the network contain M neurons, thus producing an M-dimensional vector $\boldsymbol{z}$, one for each of the desired classes. Since we want this output to represent a probability distribution over the available classes, we can impose the desired properties of a probability distribution (i.e. each probability value $z_i \in [0, 1]$ and $\sum_{i=1}^{M} z_i = 1$, for $i \in [1, \ldots, M]$. To enforce this structure on the network, the output activations $\boldsymbol{z}$ of the last layer are fed to a softmax layer. The latter applies softmax normalization [123] on the input values.

$$y_i = \frac{e^{z_i}}{\sum_{j \in layer} e^{z_j}} \tag{3.2}$$

As seen in equation 3.2, the softmax function squashes the incoming values to the desired $[0, 1]$ range, and ensures that they sum up to unity.

3.2.3 Training a neural network

In the neural network, *learning* corresponds to modifying the weights and biases such that given an input $\boldsymbol{x}$, the resulting output $\boldsymbol{y}$ matches a desired value $\boldsymbol{t}$ as close as possible. The perceptron algorithm [101] was a first an implementation of such a process, where given a sets of desired input-output examples $(\boldsymbol{x}, \boldsymbol{t})$, a single-layer neural network is able to modify its parameters such that each output $\boldsymbol{y}$ progressively approaches the desired

values $\boldsymbol{t}$ for each training example, thus reducing its prediction error. This process of *training* a model with such a set of given labelled examples is called supervised learning.

In order to apply supervised training on a neural network, we define an objective function to optimize its performance. In cases where the function represents a cost (e.g. the error / misclassification rate), it is called a loss function. The loss function $L(\cdot)$ is an numeric estimate of the categorization performance of the network, with respect to a set of training examples pairs $(\boldsymbol{x}, \boldsymbol{t})$. In single-label classification networks equipped with the softmax layer, the cross-entropy cost function is commonly used. It quantifies the loss as the negative log-probability of the correct class and is depicted in equation 3.3.

$$L = -\sum_{j=1}^{M} t_j log(y_j) \tag{3.3}$$

where $\boldsymbol{y}$ is the network prediction vector and $\boldsymbol{t}$ is a one-hot representation of the desired class (i.e. $t_i = 1$, $t_j = 0$ for $j \neq i$). The farther y_j is from unity, the network is increasingly penalized. Cross entropy has the desirable property of having a large gradient when the correct class is 1 and the network prediction is very close to zero. With a loss function available, we can evaluate the average performance of the network on a set of training examples. Given a current weight configuration, we can deduce how to change the weights on the last hidden layer in order to shift results towards the ground truth pairs $(\boldsymbol{x}, \boldsymbol{t})$ by deriving a gradient vector $\nabla \boldsymbol{L} = \left[\frac{\partial L}{\partial w_1}, \ldots, \frac{\partial L}{\partial w_M}\right]$. However, since we want to train the entire network, when the latter is composed of multiple layers weight inter-dependencies between layers in the computational graph need to be considered. For example, the weights in the i-th layer depend on the weights of all previous layers, since the data fed into it is produced by the chain of computations that consists of the layers in positions $0, 1, \ldots, i-1$. Given $\nabla \boldsymbol{L}$, the backpropagation method [77] can efficiently compute a global gradient $\nabla \boldsymbol{L_G}$ vector via the chain rule. $\nabla \boldsymbol{L_G}$ contains loss gradients with respect to all neural weights and bias in the network [73], by considering the backwards propagation of errors from the last layer to the first. The acquired gradient vector $-\nabla \boldsymbol{L_G}$ indicates a direction in the network parameter space, towards which a maximal error reduction is achievable. Shifting the network configuration effectively towards such a point is the subject of global optimization research. For neural networks, variants of gradient descent are usually used [104].

The training process is repeated, balancing two evaluation criteria; the first is the aforementioned training loss, $L(\cdot)$, which reflects how well the network has learned the training data. To avoid over-training, i.e. to prevent the learned function $\hat{f}$ from fitting noise and too coarse elements of the manifold that represents the probability distribution of the training dataset, we additionally measure performance on test data. Test data are instances not seen during the training phase, but assumed to originate from the same data source, X. Obtaining a model that, having been trained on one dataset, performs well on another, unseen dataset, is called *generalization*, and it is the overall goal in machine learning.

3.3 Deep learning

While training a network with a large number of hidden layers is not a simple task (e.g., since large networks mean costly backpropagation operations), such architectures come with strong advantages. In a multilayer network, the n-th hidden layer can be considered to be a representation of the input on an additional level of abstraction with respect to the $n-1$-th layer, as well as a learned representation of the previous layer itself. This stacked architecture enables the discovery of intricate structures in the data, leading to a paradigm shift: from expert-led feature engineering to automatic methods for pattern recognition and representation learning [11]. This leads to the emergence of ever larger networks and, the number of layers in a neural network being referred to as its "depth", the sub-field of *Deep Learning* [108]. There, it is common practice to build deep networks to tackle difficult, large-scale learning problems. While this practice is computationally expensive, recent hardware advances in GPU technology [9] have kick-started a number of successes in multiple machine learning fields.

3.3.1 Convolutional neural networks

Convolutional neural networks (CNNs) are deep neural models, commonly applied in computer vision tasks. They were inspired by biological studies on the visual cortex of cats, where neural cells with specific properties were observed [51], while also bearing resemblance to the visual mechanism of primate species [27]. One group of cells responded to certain visual primitives such as orientation of edges, while others exhibited a larger degree of spatial invariance responding to certain stimuli within a spatial region of the visual input, named the receptive field. This paradigm was first implemented in an artificial neural network in the Neocognitron model [30], which like modern variants used convolutional and pooling operators. Backpropagation based training was applied for handwritten zip-code character recognition in [71] and handwritten check recognition in [72]. Deep convolutional neural networks (DCNNs) have become extremely popular recently, being used successfully in numerous pattern recognition contests and commercial applications [125, 66, 18, 39].

The main component of a CNN is the convolution layer. For discrete signals $\boldsymbol{x}$ and $\boldsymbol{w}$, convolution is defined as:

$$y[n] = \sum_{k=-\infty}^{\infty} (x[n-k]w[k]) \tag{3.4}$$

where the extension to two-dimensional signals like images is straightforward.

$$y[m,n] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} (x[m-k,n-l]w[k,l])$$

In practice, the convolution summation limits are restricted with respect to the dimensions of the convolution operands. As a result, each neuron in a convolutional layer is associated

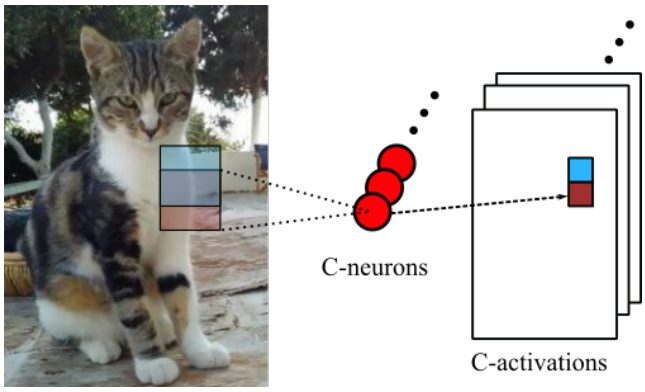

Figure 3: Operation of a convolutional layer. An input image (left) is scanned by convolution kernels (red circles), producing output feature maps (right). The input convolution windows of two adjacent steps are illustrated (red and blue translucent rectangles in the image), which produce corresponding output vales (red and blue squares in the feature map output), for the first convolution kernel.

with weights that form a convolution kernel: a $n \times n$ window, $n = 2k+1$, $k \in \mathbb{N}$, the weights of which the network optimizes. A fixed stride size and padding configuration is also preset for each layer, parameters which also determine the dimensionality of the output. For a visual example of the operation of the convolutional layer on an image, see figure 3.

Common architectures include series of convolutional layers, acting both as learnable feature detectors and dimensionality reduction operators. Further reduction is achieved by pooling layers, which replace their receptive field with a single computed or selected value, according to a specified criterion (e.g. select the pixel with the maximum intensity in max pooling layers, compute the average of the image patch in average pooling layers, and so on). Not surprisingly, an area where DCNNs are widely used is computer vision. The input image is read by the input layer as an array of numeric values on which the convolutional layers learn to recognize ever more abstract visual features. For example, convolution kernels in the first layer may learn to distinguish primitive visual artifacts like oriented edges, simple textures or blobs. The second layer can learn more complex shapes, a function of the activations of the first layer, and so on. We describe a DCNN model is described in section 4.2.1.

3.3.2 Recurrent neural networks

Convolutional networks discussed so far operate in a feed-forward manner; computation flows from the input layer towards the output layer and once the network prediction is computed, the process stops. In contrast, a recurrent neural network (RNN) [24] retains a memory-like internal state which is updated via a feedback mechanism on subsequent *time steps* of an input *sequence*. This enables the network to store information pertaining to the temporal inter-dependencies of the input, rather than considering the content of each input separately at the present time step.

Given a sequence of inputs $\{\boldsymbol{x}_i, \ldots, \boldsymbol{x}_n\}$ and an initial state $\boldsymbol{h}_0$, the RNN computes outputs $\boldsymbol{y}$ and updated state $\boldsymbol{h}_t$ as in equation 3.5:

$$\begin{aligned} \boldsymbol{h}_t &= f\big(\boldsymbol{W}_{xh}\boldsymbol{x}_t + \boldsymbol{W}_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_1\big) \\ \boldsymbol{y}_t &= g\big(\boldsymbol{W}_{hy}\boldsymbol{h}_t + \boldsymbol{b}_2\big) \end{aligned} \tag{3.5}$$

where $\boldsymbol{W}_{xh}, \boldsymbol{W}_{hh}, \boldsymbol{W}_{hy}$ are the connection weights between inputs and state, recurrent state and between state and outputs respectively and $\boldsymbol{b}_i$ are corresponding bias terms. Figure 4 depicts the model in the recursive and unrolled conceptualizations, as well as the multilayer architecture.

In RNN training, the phenomena of vanishing and exploding gradients [45] arise. As names imply, again, these phenomena relate to the behaviour of the gradient magnitude when propagated through the network graph [12]. Specifically, it refers to cases when said magnitude takes extreme values due to the gradient computation procedure being based on the chain rule: on long chains of computation (i.e. long paths in the network graph), gradient values tend to shrink or explode as a result of repeated multiplications of small or large gradient quantities, respectively. These phenomena often occur during RNN training, since unrolled RNN networks include long computational paths which are necessary to capture temporal inter-dependencies of the input sequences.

With respect to gradient explosion, clipping excessive values with respect to a predefined norm threshold [34] is a simple solution that mitigates the problem. Vanishing gradients can be rectified by the use of the LSTM network, which involves an additional memory component, the *cell memory state*, propagated via a separate channel than the RNN state h. This component stores information regarding the process by which the state changes as a function of step, input data and current state. Specifically, a series of operators are applied on the input and state variables via suppression and scaling functions, namely the sigmoid and $tanh(\cdot)$. These are depicted depicted in equations 3.6.

$$\begin{aligned} \boldsymbol{f}_t &= \sigma\left(\boldsymbol{W}_f \boldsymbol{z}_t + \boldsymbol{b}_f\right) \\ \boldsymbol{i}_t &= \sigma\left(\boldsymbol{W}_i \boldsymbol{z}_t + \boldsymbol{b}_i\right) \\ \boldsymbol{c}_t^* &= tanh\left(\boldsymbol{W}_c \boldsymbol{z}_t + \boldsymbol{b}_c\right) \end{aligned} \tag{3.6}$$

$$\tag{3.7}$$

where $\boldsymbol{z}_t$ is the concatenation of $\boldsymbol{x}$ and $\boldsymbol{h}_{t-1}$, $\boldsymbol{W}$ and $\boldsymbol{b}$ terms represent weights and biases, and the $\odot$ symbols represent element-wise vector multiplication. Specifically, the f_t term scales which parts of the old cell state $\boldsymbol{c}_{t-1}$ to forget, $\boldsymbol{i}_t$ selects which values to use in order to update the cell state, using the new candidate values in $\boldsymbol{c}_t^*$. The new hidden state $\boldsymbol{h}_t$ is produced by a similar process. Parts of the new cell state are concealed via $\boldsymbol{o}_t$, which are scaled and set as the $\boldsymbol{h}_t$.

As a result of their sequence-oriented architecture, RNNs have been widely used for applications where data are composed of inter-dependent units, e.g. natural language and speech processing [106, 87, 36, 17], image description and generation [37, 23, 128] and others. We describe a RNN variant (the LSTM model) in section 4.2.3, as one of our proposed workflows.

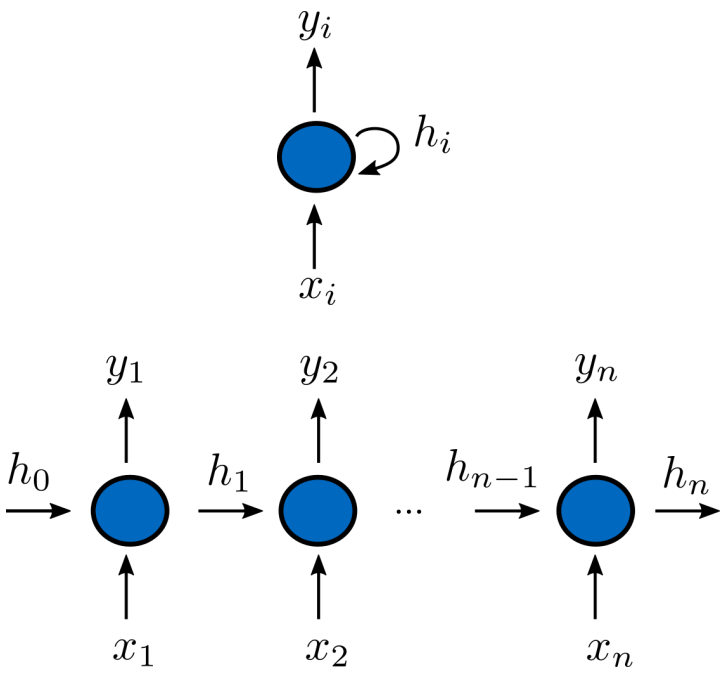

Figure 4: The RNN model. The recursive depiction (top) illustrates the feedback of the internal state h into subsequent timesteps. The unrolled illustration (bottom) shows an equivalent feed-forward model.

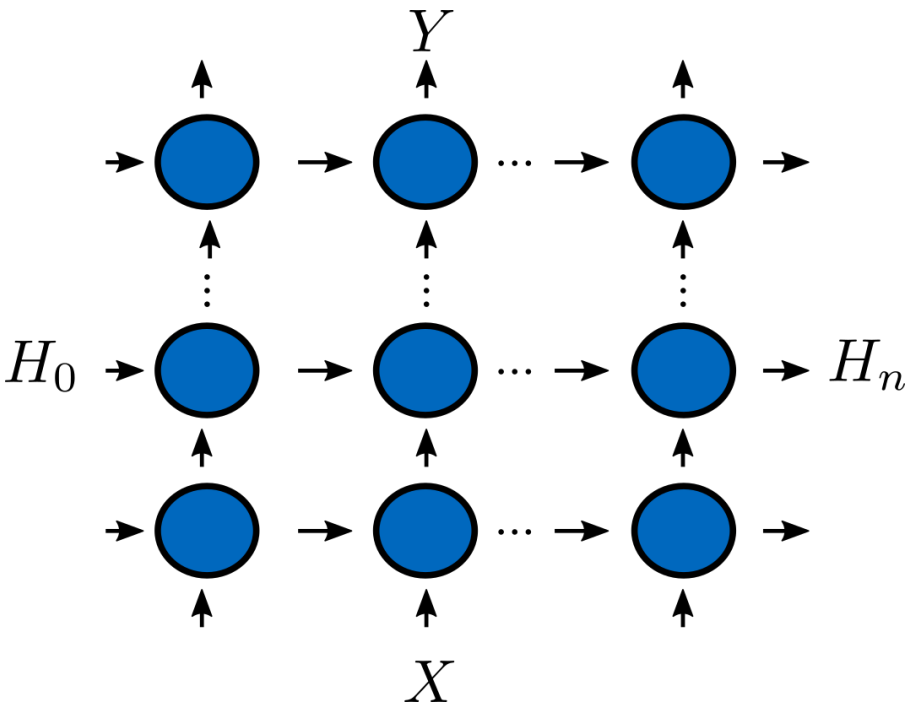

Figure 5: The multilayer RNN model, depicted in an unrolled illustration. X denote the network input sequence, Y the output sequence and H the multi-layer state.

4. PROPOSED METHOD

In this chapter we describe our method for tackling the stated goals. To recall, this study aims to answer the following research questions, with respect to the video classification task:

1. How do sequential neural models fare versus simple, aggregation-based approaches that do not take temporal inter-dependencies of the input into account?

2. What is the effect of the visual and audio video modalities, for each of the above model types (e.g. aggregation-based and sequential)? How can video modalities be combined to aid classification performance?

In the next sections we detail the proposed method, starting with the preprocessing phase, which prepares video data to be fed to the neural models. This consists of video sampling and frame encoding stages, presented in sections 4.1) and 4.2.1 respectively. For the first stage, we adopt a randomized clip and frame sequence extraction approach, producing image sequences that capture modality-specific information. We tackle frame encoding by adopting a DCNN representation extraction scheme, using the Alexnet model to map video frames to numeric feature vectors that can be processed by a computer algorithm.

To address the research questions stated in the beginning of this chapter, we deploy two classification models, namely the fc and LSTM classifiers. These models are used to produce prediction scores for input video parts, which are subsequently combined to correspond to predictions pertaining to the entire video. This is achieved by a variety aggregation methods which we examine, for each classifier. The two classifiers give rise to the homonymous FC and LSTM video classification workflows, each composed of the aforementioned frame encoding and corresponding classification phases.

In section 4.3, we describe the approach adopted to examine the second research question, i.e. the utilization of multiple video modalities to aid video classification. There, we investigate a series of multimodal fusion methods to combine visual and audio data in an end-to-end fashion. In addition, we extend the investigation of the first research question in the multimodal setting, applying the FC and LSTM workflows in each fusion method examined here.

4.1 Data preprocessing

In this section we outline the preprocessing steps undertaken to prepare video data to be fed to our classification pipeline. Firstly, we describe the preprocessing stage for the visual content of the video, followed by the spectrogram extraction approach on the audio content.

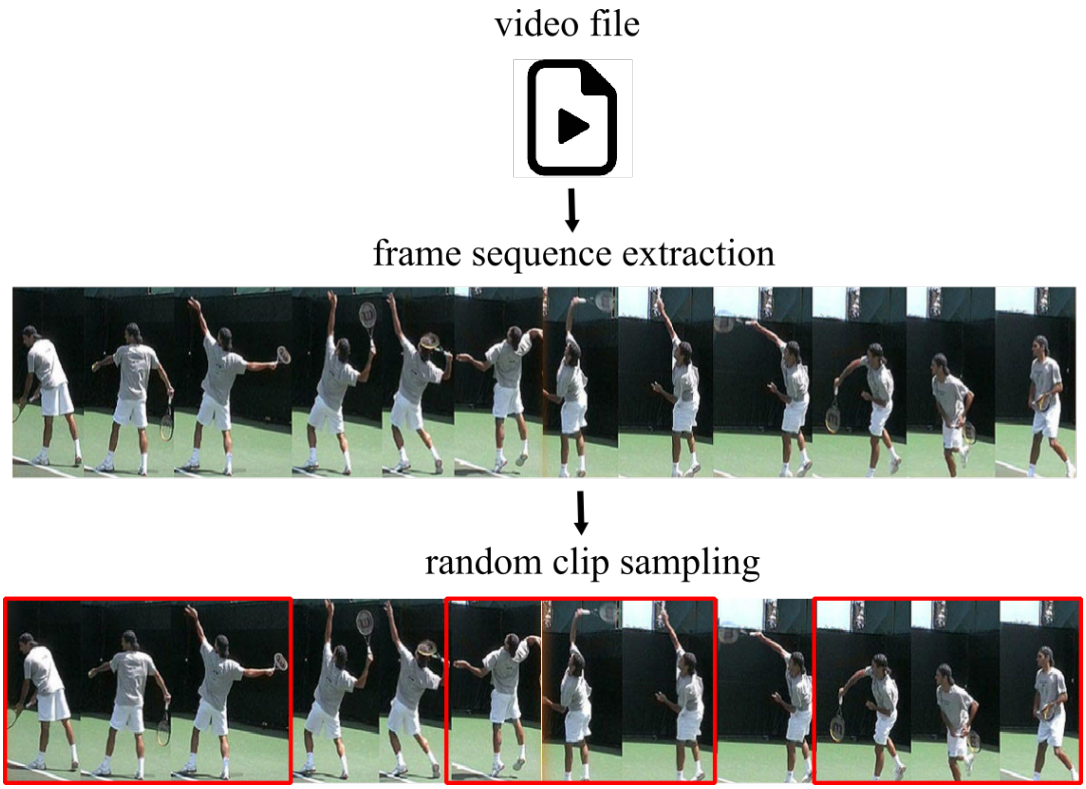

Figure 6: The visual content preprocessing pipeline. The underlying frame sequence is extracted from the video from which random clips – i.e. contiguous frame sequences of constant length – are selected. In the example, red rectangles denote the clip boundaries, which are composed of the 3 enclosed frames.

4.1.1 Visual content processing

The video classification pipeline we use begins with extracting usable data from a video file, in a machine-recognizable format. We extract visual video information with the following process. Given a video file v, we extract the video frames at a reduced rate, namely one frame per second. This serves to both arriving at a temporal resolution adequate for classification and reducing the amount of superfluous frame data for each video, keeping the dataset sizes manageable. At the end of this process, each video is represented by an ordered series of N frames. Given v, we extract K video *clips* $\{c_1, c_2, \dots c_K\}$, each composed of N frames: $c_i = \{f_1, f_2, \dots f_N\}$. A visualization of the clip extraction process for the visual modality is depicted in figure 6. We note that the frame extraction order respects the temporal order of each frame in the video.

For each video, we extract 4 clips, each consisting of 8 frames, amounting to 32 frames per video. Clip boundaries are non overlapping and randomly selected. For videos where the available frames are not enough to incorporate the selected clips per video and frames per clip, we duplicate the first frame of the video to reach the quota mentioned above. If the number of clips itself is insufficient, we randomly duplicate clips from those that were managed to be extracted.

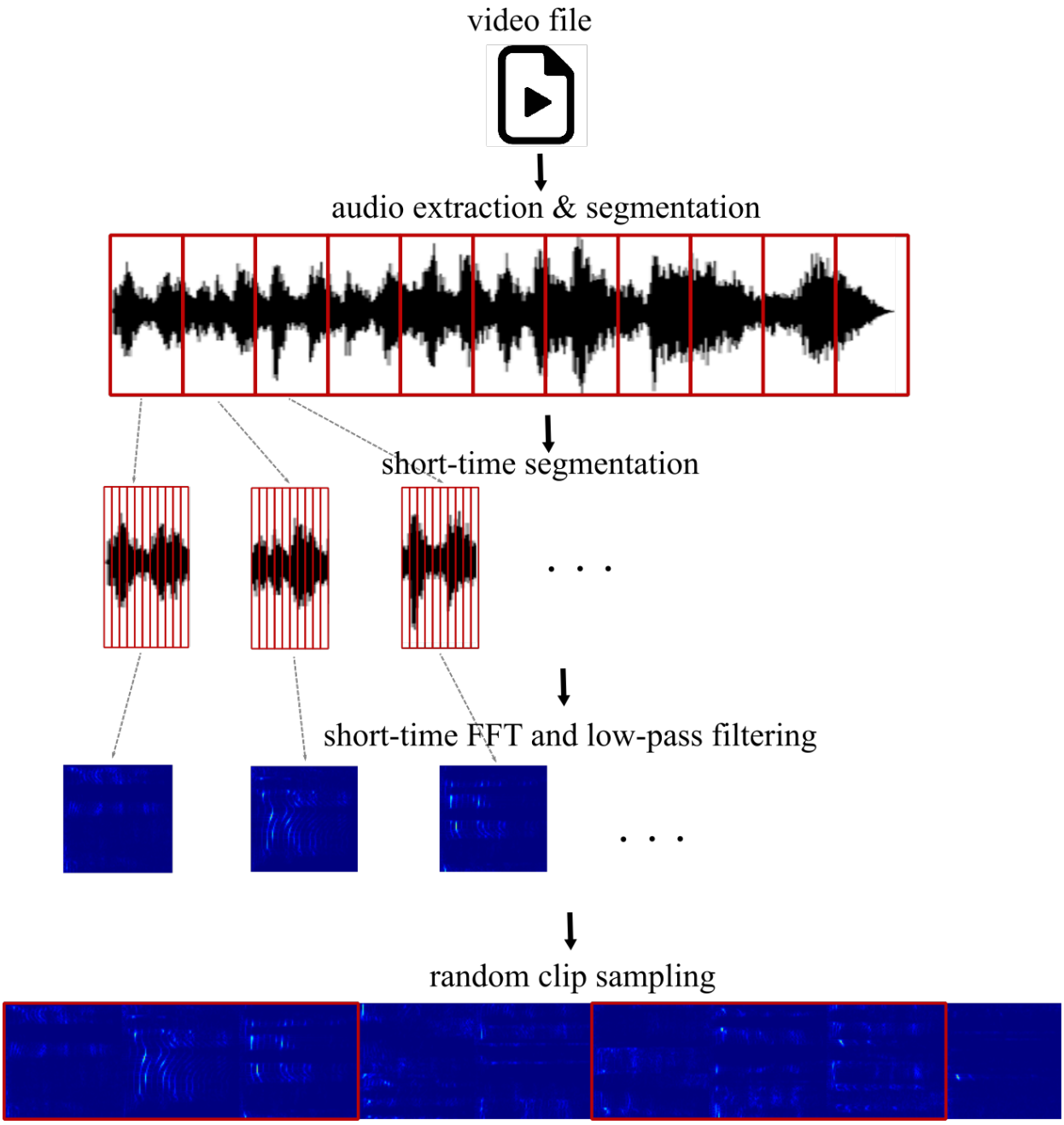

Figure 7: The audio content preprocessing pipeline. The audio track is extracted from the video and is partitioned into 1-second segments. Each such segment is mapped into a spectrogram image, via a further 20ms temporal partitioning, followed by application of short-time FFT and low-pass filtering. After the spectrogram sequence is produced, clip extraction is performed in the same way as in the visual content, showcased in figure 6.

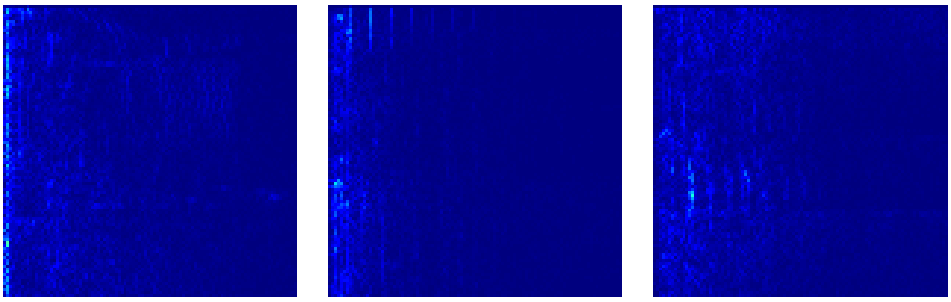

Figure 8: Examples of spectrogram images.

4.1.2 Audio processing

Regarding the audio component, we extract the accompanying audio file from each video and proceed to a spectrogram extraction process. Specifically, we partition the audio to a sequence 1-second audio clips. For each such clip, we apply short-time Fourier transform on 20ms windows with a 10ms overlapping stride. This segmentation procedure produces 99 spectral responses for the 1000 ms clip. After applying a low-pass filtering operation, keeping only the first 100 low-frequency coefficients of each response, we end up with a 99×100 coefficient matrix. The matrices are stored in image format, resulting in a pictorial representation for the audio modality. Clip extraction is applied in a similar manner as in the visual content described in the previous section (4.1.1). A visualization of the audio processing pipeline is available in figure 7, whereas an example of spectrogram images used in this study is illustrated in figure 8.

4.2 Single-modality workflows

In this section, we outline the components we use for the single-modality classification approaches. In section 4.2.1, we describe the frame encoding process, by which input image data are represented by a feature vector. This is accomplished by obtaining the response of an appropriate DCNN layer, specifically the Alexnet architecture.

In the sections that follow we describe the architectures that apply different parsing strategies with respect to the temporal information content of the encoded frame sequence. Specifically, in section 4.2.2 we describe the FC workflow, which classifies the encoded frames via a combination of fully-connected and softmax layers, followed by simple, straightforward temporal aggregation schemes. What follows is section 4.2.3, where the LSTM workflow, utilizing an LSTM recurrent neural model, is applied to process the encoded input frame sequence. This method is designed to effectively keep track of temporal interdependencies in the input, utilizing them for producing “temporally-aware” classification predictions. Finally, we present a set of fusion strategies applied to the aforementioned workflows, by which predictions are aggregated from frame-level predictions (i.e. classification scores with respect to a single frame) to sequence-level ones (e.g. regarding the input video or clip).

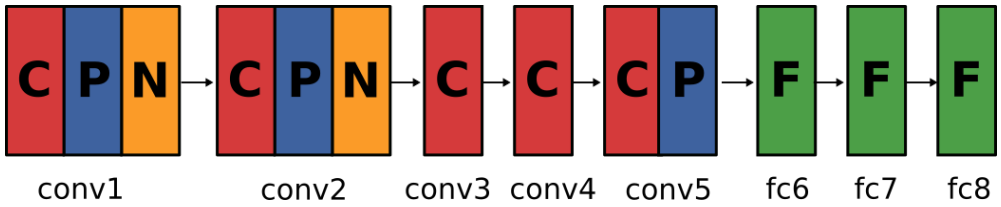

Figure 9: The architecture of the Alexnet DCNN. The letters C, P, N and F stand for convolutional, pooling, normalization and fully-connected layers, respectively.

4.2.1 Frame encoding

In order for our model to process video content, the preprocessing steps described in section 4.1 are applied. This transforms video content to a series of images, representing the former in the selected input format (i.e. video frames for visual content or audio spectrograms for audio content). In order to process these images, we use a popular DCNN model widely used in image classification tasks, namely the Alexnet architecture [66]. The Alexnet model receives 224×224 images with three color channels in RGB format and feeding them through a series of successive processing layers. It contains five instances of convolutional layers and three fully-connected layers, where after each convolutional layer a ReLU operation is applied, introducing additional non-linearity to the computation. Max-pooling operators are applied on the output of the first, second and last convolutional layers for dimensionality reduction and translation invariance of convolutional responses. Local response normalization operations follow the first two convolutional layers, applying a "brightness normalization" transformation on the layer responses which was found by the authors to aid discrimination [66]. While training from scratch initializes the network with random weights, we take advantage of transfer learning approaches [93] and initialize the encoding with pretrained weights on visual data of the ILSVRC 2012 challenge [105]. See figure 9 and table 1 for the detailed architecture and model components.

The Alexnet model is used as a frame encoding operation by mapping input frames to encoded feature vectors. To achieve this, we truncate the network, discarding all components past a selected layer, obtaining the output of the latter as the representation of the network input. Specifically, we experiment with selecting the last, fully-connected layers in the network, i.e. the fc6, fc7 and fc8 layers. The first two layers produce 4096-dimensional feature vectors, since their response consists of the output of 4096 neurons. The final fc8 layer acts the classifier component, when the network is used as an end-to-end classification model, rather than a feature generation process, as is the case here. In the former scenario, fc8 produces 1000-dimensional vectors, since the pretrained model was fitted to the 1000 classes in the ILSVRC competition [105].

4.2.2 The FC workflow

With the vector representation of an input frame in hand, we can move on and produce a classification score for it (and at the same time, the video clip it represents) via feeding it

Table 1: The Alexnet network architecture. "conv", "lrn", "pool" and "fc" stand for convolutional, local response normalization, pooling and fully-connected layers, respectively. $C = \{c_1, c_2, \ldots, c_{|C|}\}$ is the label set for the classification task.

name	type	dimensions	neurons
conv-1	c	$11 \times 11 \times 3$	96
pool-1	p	2×2	-
lrn-1	p	2×2	-
conv-2	c	$5 \times 5 \times 96$	256
pool-2	p	2×2	-
lrn-2	P	2×2	-
conv-3	c	$5 \times 5 \times 256$	384
conv-4	c	$5 \times 5 \times 256$	384
conv-5	c	$5 \times 5 \times 256$	384
lrn-5	p	2×2	-
fc-6	fc	$9216 \times 1 \times 1$	4096
fc-7	fc	$4096 \times 1 \times 1$	4096
fc-8	fc	$4096 \times 1 \times 1$	$\|C\|$

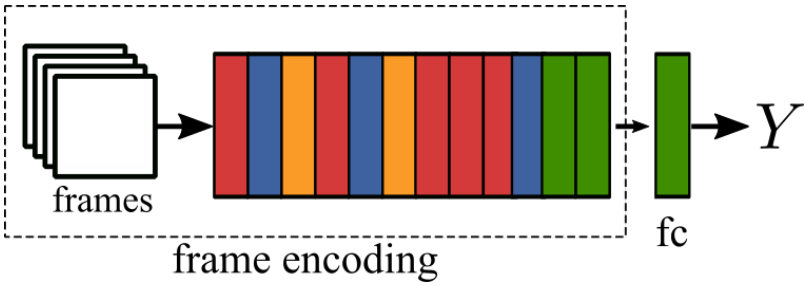

Figure 10: DCNN frame encoding, illustrated for the FC workflow.

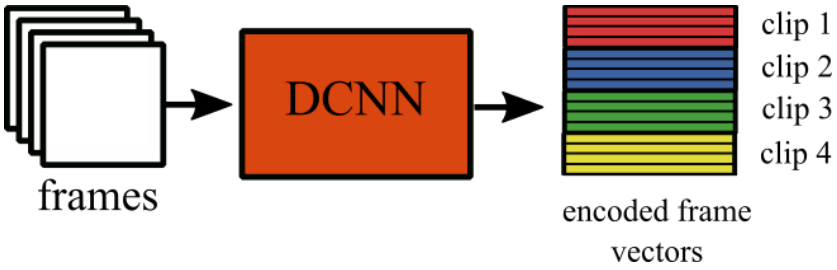

Figure 11: The frame encoding process: a collection of frames (left) is fed through a DCNN model (middle). The responses of a selected fc layer are gathered as frame encodings (right). In this image, there is a single video consisting of 4 clips (color-coded in red, blue, green and yellow), each consisting of 4 frames.

to a classification model. In the FC workflow, we use an fc classifier, i.e. a fully-connected layer neural layer, followed by a softmax operator as explained in section 3.2. To recall, the number of neurons in the fc layer is set to the number of classes in the classification task, producing raw prediction scores per class. Attaching a softmax operator to the aforementioned output squashes the results to a $[0, 1]$ range and enforces all predictions summing to unity, thus producing an output that can be interpreted as a probability density over the classes. In the pretrained network, we discard the last pretrained fully-connected layer (i.e. fc8 in Alexnet) and reinitialize it with random weights, learning the classifier of interest from scratch.

Recall the way a video is handled in the preprocessing pipeline (see section 4.1), sampling a number of clips from the video, each consisting of N frames. Given a video clip, its frames are thus encoded into feature vectors $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_N\}$, as shown in figure 11. Given this collection of frame encodings, the FC workflow maps each $\boldsymbol{e}_i$ into a prediction vector $\boldsymbol{p} \in R^{|C|}$, with p_j denoting the confidence of the model that the i-th frame belongs to the j-th class. Since the task of interest is *video* classification however, we need a way of aggregating frame-wise predictions into clip-wise and, further, video-wise confidence scores. To produce these levels of prediction, we adopt score fusion approaches. Regarding aggregation to clip-wise predictions, we examine two methods in conjunction with taking the average of marginal scores into an aggregate. Firstly, early fusion computes the arithmetic mean of frame representations, aggregating all encodings to a single clip encoding vector. The clip vector thus represents the entire clip and can be fed to the FC classifier to obtain a prediction score for it. Conversely, late fusion produces prediction scores for each frame encoding separately, fusing the marginal classification outputs into one aggregate clip prediction. The early and late fusion methods are depicted in figure 12.

Early and late fusion deals with aggregation from frames to clips. For video-level aggregation, we simply compute the average predictions across clips.

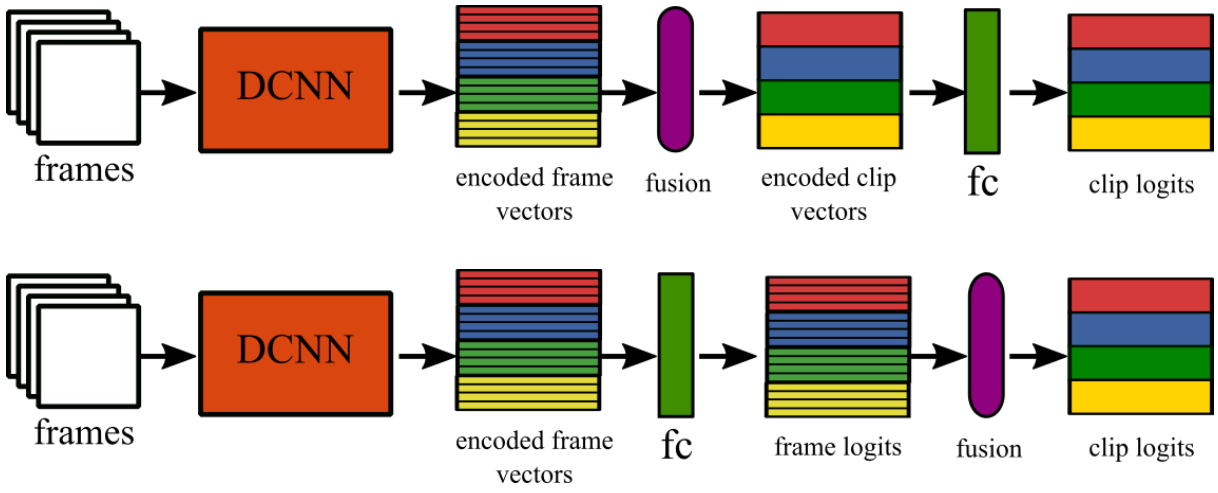

Figure 12: Early (top) and late (bottom) frame fusion approaches. Early fusion acts on encoded frame vectors, fusing frame encodings into a single clip encoding vector, which is subsequently classified by the fc layer. The late fusion approach classifies each frame encoding individually, fusing frame predictions into a single clip prediction. Color-coding follows the conventions in figure 11.

4.2.3 The LSTM workflow

The LSTM workflow utilizes, as the name implies, a Long Short-Term Memory (LSTM) model [46] for video classification. As discussed in chapter 3, an LSTM is a deep recurrent neural model that augments recurrent neural networks (RNNs) [24], with components that attempt to remedy some fundamental problems encountered during training. As discussed in section 3.3.2 RNNs are deep neural networks equipped with a feedback memory parameter h, called the RNN state.

Given an N-frame video clip encoded into vector representations $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_N\}$, we feed the latter of frames through the LSTM network. After N computation steps, the network has produced outputs $Y = \{\boldsymbol{y}_1, \boldsymbol{y}_2 \ldots, \boldsymbol{y}_N\}$ and a final hidden output state $\boldsymbol{h}_N$. Each $\boldsymbol{y}_i$ is a vector in $\mathbb{R}^L$, with $\boldsymbol{y}_i(j)$ representing the confidence of the model that the i-th frame belongs to the j-th class, given the input $\boldsymbol{x}_i$ and all preceding hidden states $\{\boldsymbol{h}_0, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_{i-2}, \boldsymbol{h}_{i-1}\}$. We note that since both the output $\boldsymbol{y}_i$ and state $\boldsymbol{h}_t$ are computed as a function of $\boldsymbol{h}_{i-1}$ and $\boldsymbol{e}_i$, it follows that the order of inputs in a sequence is crucial; earlier frames having a cumulative contribution on the network, affecting all subsequent hidden states and outputs. The workflow is illustrated in figure 13.

Given the multiple outputs of the LSTM model, we examine multiple ways of extracting a single prediction for the input frame sequence. Firstly, we select the prediction at the last time step ($last$ strategy). This is the built-in mechanism of temporal aggregation in the LSTM, since useful prior information should be encoded in the hidden state, and should thus adequately influence the last network output. Additionally, we consider taking the average of all predictions (avg strategy). Furthermore, we examine treating the hidden state as the prediction vector ($state$ strategy); this approach treats the LSTM as a prediction

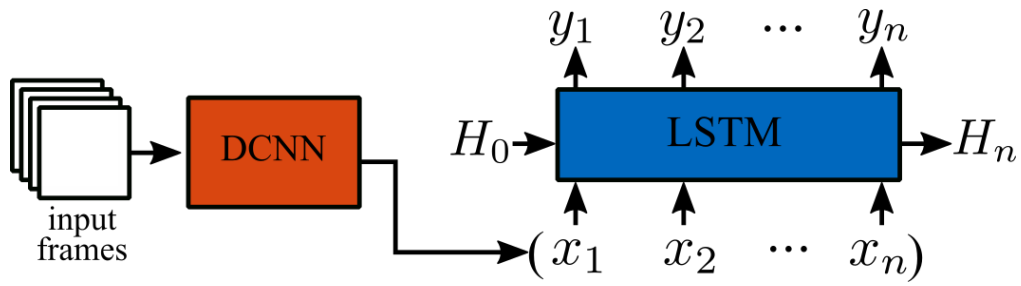

Figure 13: The LSTM workflow. DCNN-encoded input frames x_i are fed to the LSTM, which produces outputs y_i. H_0 and H_n denote the initial and final state vectors.

encoder, representing the input sequence as a model vector in its internal state. This is accomplished by setting the number of state neurons to the number of desired classes and considering the final state vector $\boldsymbol{h}_N$ as the classification response for the input frame sequence, ignoring the network output y. Finally, video-level fusion is performed in the same way as the FC workflow, i.e. by computing the arithmetic mean of clip predictions.

4.3 Multimodal workflows

In this section we describe the workflows used to examine the performance of the multimodal setting for video classification, i.e. the utilization of more than one modalities to reach a video prediction. In this work, we use the visual and audio modalities from video data, both are extracted in the form of images as described in section 4.1. We examine two categories of multimodal fusion, with each candidate method per category applied to both the FC and LSTM workflows described in section 4.2. Firstly, "direct" fusion approaches are examined in section 4.3.1, with straightforward techniques of combining data from different modalities being explored. Secondly, in section 4.3.2 we test fusion methods inspired from image description approaches, where modality-specific information is presented as a bias in the input sequence.

4.3.1 Direct data fusion

In this section, we outline straightforward modality fusion approaches, bearing resemblance to the work of [138] but applied on audio-visual content rather than spatial and temporal streams. We examine two methods for data combination, given the visual and audio data sources that provide frames, as discussed in section 4.1. It should be noted that, although beneficial, the visual and audio frame / spectrogram sequence need not be aligned, i.e. the two sequences need not correspond to the same temporal window in the source video. First, avg fusion combines modality data by computing the arithmetic mean at the frame encoding level, i.e. after images have been mapped to a vector representation (see section 4.2.1). Specifically, given visual and audio input clips $c_v = \{\boldsymbol{e}_{v1}, \boldsymbol{e}_{v2}, \ldots, \boldsymbol{e}_{vN}\}$, $c_a = \{\boldsymbol{e}_{a1}, \boldsymbol{e}_{a2}, \ldots, \boldsymbol{e}_{aN}\}$, avg fusion produces the multimodal clip $c_m = \{\boldsymbol{e}_{m1}, \boldsymbol{e}_{m2}, \ldots, \boldsymbol{e}_{mN}\}$,

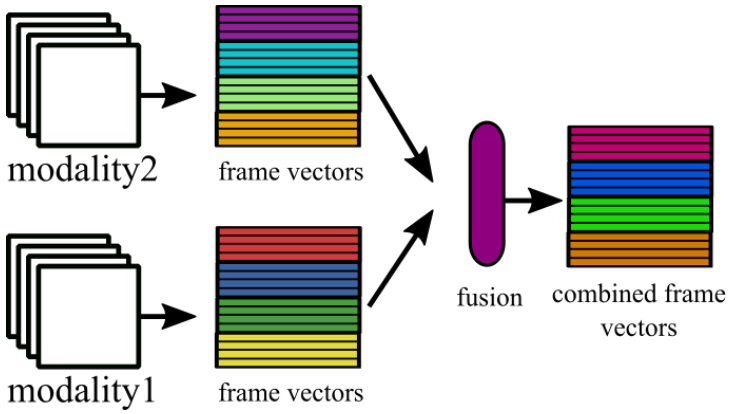

Figure 14: The process of avg and max multimodal fusion. Given the vector representations of two collections of images corresponding to different modalities of the same object, avg fusion combines the two modalities by averaging the encoding values, while max produces coordinate-wise maxima. Either way, the resulting vector has the same dimension as the input visual / audio vectors. Color coding follows the conventions as in figure 12.

where $\boldsymbol{e}_{mi} = \frac{\boldsymbol{e}_{vi}+\boldsymbol{e}_{ai}}{2}$. This process implies that both modalities have the same number of clips for a video, and said clips are composed of the same number of frames. Similar to avg, max fusion computes coordinate-wise maxima from two clips, i.e. $\boldsymbol{e}_{mi} = \max(\boldsymbol{e}_{vi}, \boldsymbol{e}_{ai})$. The process for avg or max multimodal fusion is illustrated in figure 14. Furthermore, $concat$ fusion concatenates the vectors representations from each modality, i.e. produces clip with vectors $\boldsymbol{e}_{mi} = [\boldsymbol{e}_{vi}^T, \boldsymbol{e}_{ai}^T]^T$, where $\boldsymbol{x}^T$ is the transpose of $\boldsymbol{x}$. This fusion approach is shown in figure 15.

After direct data fusion, the fused clip c_m can be treated as if it originates from a single modality. It can thus be processed by either of the single-modality workflows outlined in section 4.2.

4.3.2 Sequence bias fusion

In this section, we investigate fusion methods inspired from neural sequence models in the image description task, where given an input image, the goal is to produce a caption that best describes the visual content. Given the sequential attributes of an image caption and the importance of word order in it, the use of recurrent deep neural models (see section 4.2.3) is a common established approach that yields good results for the task [23, 128, 57, 82].

We construct the first fusion method borrowing from the image description approach in [128]. There, images are encoded into vectors in a similar manner as explained in section 4.2.1 while caption words are also mapped to vectors via learned word embeddings [88]. The authors provide the visual content vector as the input at the first time step, by which to impose a conditioning mechanism on the network, i.e. a preliminary step that introduces

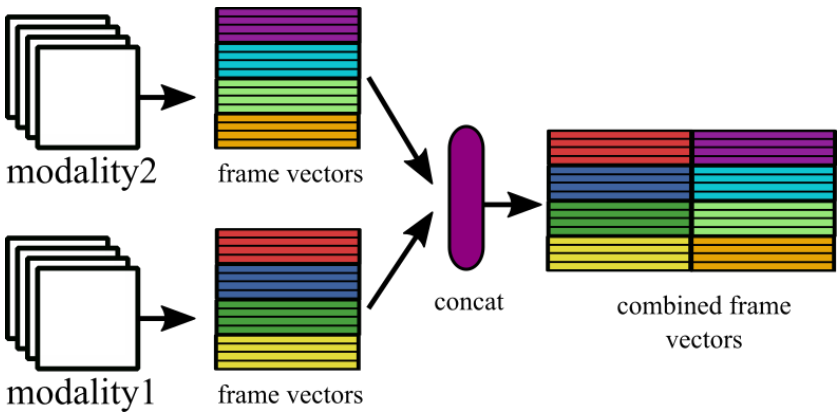

Figure 15: The process of concat multimodal fusion. Given the vector representations of two collections of images corresponding to different modalities of the same object, concat fusion combines the two modalities by concatenating the encoding vectors, with the visual modality leading. Color coding follows the conventions as in figure 12.

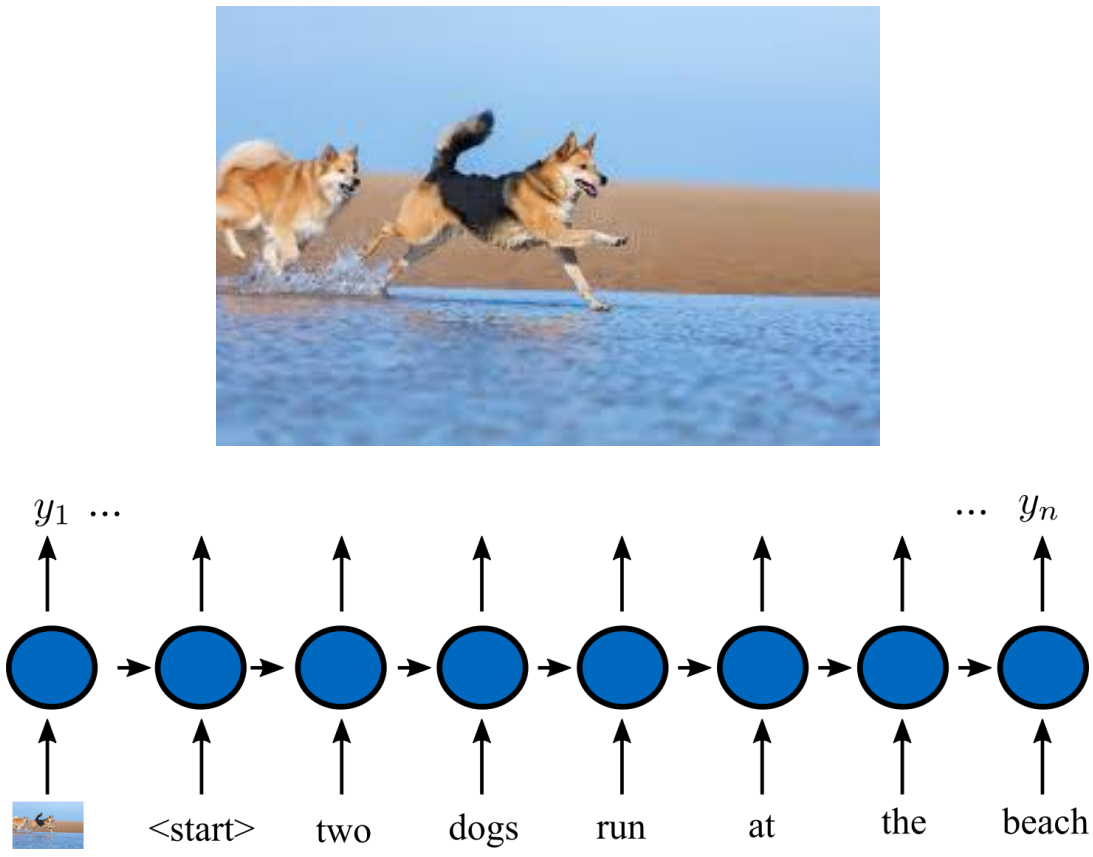

Figure 16: Example of handling an image description task with a recurrent model. Given the input image (top), information pertaining to it is supplied in the first input position in the recurrent model (bottom). It is followed by caption information tokenized to words. The <start> item is a special token that denotes the beginning of the caption word sequence.

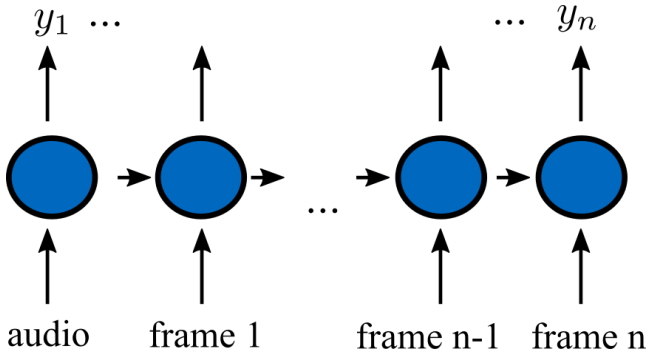

Figure 17: The input-bias method for introducing audio context in the visual frame sequence of an RNN. Aggregated audio information (the "auxiliary" channel) is supplied as the first input element, followed by the visual encoded frames of the "main" modality.

a bias. Given the significant effect of the first frame in the sequence (see section 4.2.3) information extracted by the first input can carry on via the hidden state, biasing the model to interpret the word sequence that follows. The mechanism is depicted in figure 16.

In a similar manner, we apply the same logic for the visual and audio modalities in this work, with the *input-bias* multimodal fusion method. We label main and auxiliary information modalities in the multimodal data. The auxiliary modality information is aggregated and introduced as the input bias, like the image vector in image description. Likewise, the main modality is analogous to the caption words in the image description analogue, the contents of which define the sequence fed to the recurrent model after the bias. We select the visual and audio component as the main and auxiliary channels respectively. Given the auxiliary audio sequence of a video, we apply average fusion to the encoded frame vectors, producing a single vector for the entire clip. The audio clip vector is then introduced as the input bias in the LSTM, followed by the visual sequence. See figure 17 for a depiction of the procedure. This process can be alternatively viewed as imbuing the frame sequence with an additional, special frame at the start. It is apparent that the audio and visual frame encodings need to be mapped to the same vector space for this type of fusion to work, i.e. the marginal modality vectors have to be of the same dimensionality. The *input-bias* method is depicted in figure 18.

The second fusion method also introduces the auxiliary modality as a bias to the main data sequence, with a few important differences. Firstly, instead of inserting the bias vector at the first input position in the sequence, we feed it as the initial state for an LSTM model, i.e. a *state-bias*. This initialization aims to condition the internal state of the network directly, rather than providing an introductory input bias and letting the network produce a good first state (after the consumption of the bias, i.e. $\boldsymbol{h}_1$) on its own. This approach assumes that the encoded bias information will hold patterns that the network will be able to interpret as a meaningful initial hidden state seed. Secondly, this method is applicable only to RNN-like models which is why we apply it only in the LSTM workflow. A visualization is depicted in figure 19.

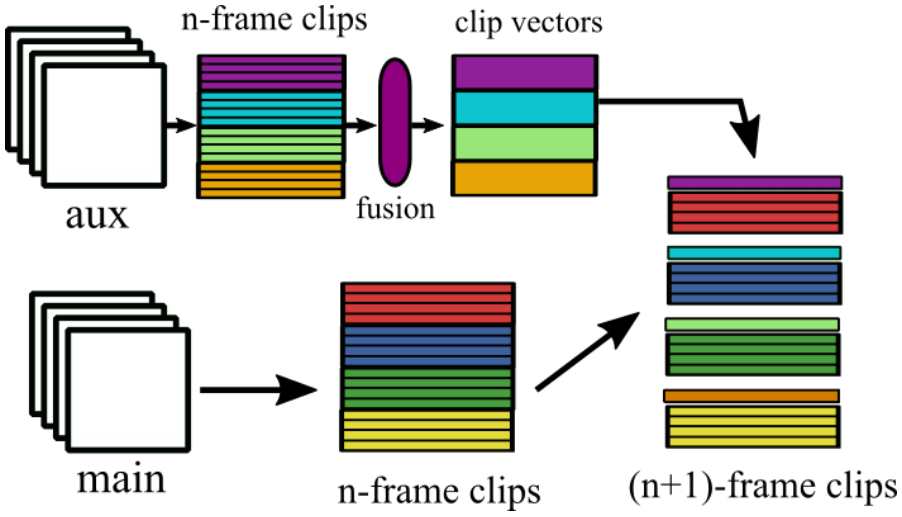

Figure 18: The input-bias multimodal fusion method. Frame vector data from the auxiliary modality (top diagram) are fused into clip vectors and inserted as the first vector in the corresponding main modality sequence.

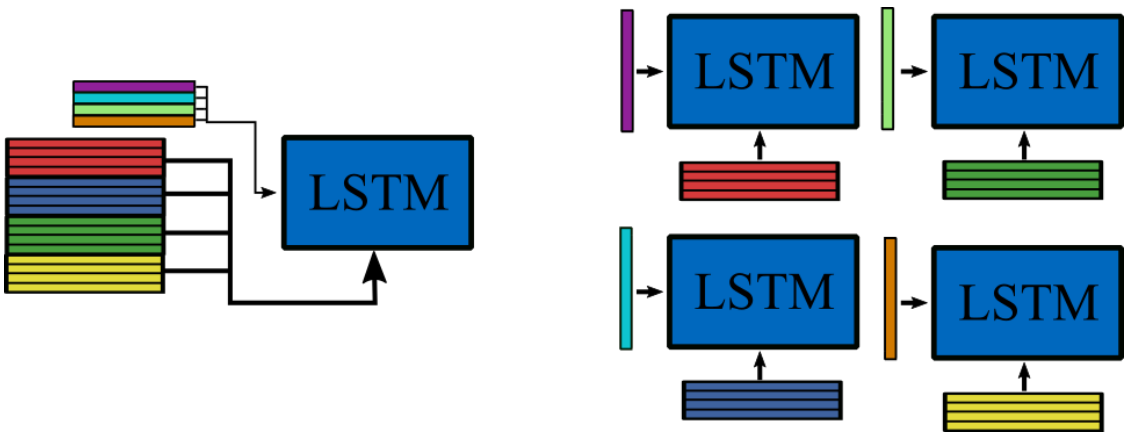

Figure 19: The state-bias multimodal fusion method. In the left, the input sequences are fed to the LSTM input and the state vectors are set as the initial LSTM state. In the image to the right, the same process is shown at a per-clip basis. Encoding vector colors denote which clips the vectors belong to.

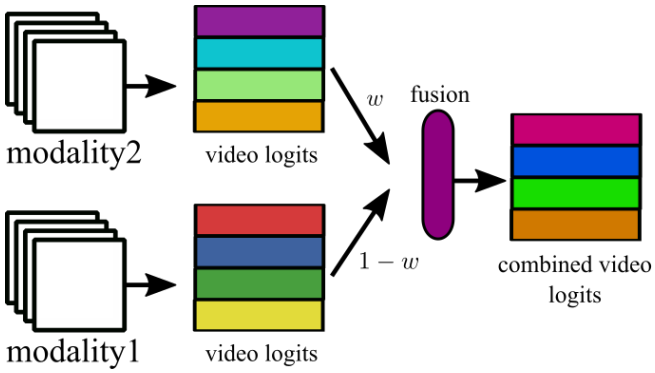

Figure 20: Multimodal late fusion.

4.3.3 Late video-level fusion

Finally, we examine late fusion of prediction scores from entire video items. While the previous multimodal fusion methods combine frame encodings from different modalities and subsequently apply a classification process with workflows from sections 4.2.2 and 4.2.3, *video-level late fusion* directly aggregates classification results, working on video-level rather than clip-level scores. The aggregation is examined in two aggregation schemes. First, we compute a linear combination of the marginal modality scores, setting complementary visual and audio modality weights, i.e. $w \in [0, 1]$ and $1 - w$, respectively. We experiment by varying w in its range with a step of 0.1. See figure 20 for a visualization of this fusion method. Secondly, we examine max pooling aggregation, in the same way as the max multimodal fusion method on the frame encoding level as described in section 4.3. For the remainder of this study, We will refer to this fusion process as late-video fusion.

In this study, the marginal visual and audio modalities selected to be combined with $late-video$ fusion, are the best-performing single-modality workflow for each modality. This scheme is simple but its practical application has some disadvantages. First, picking the best performing workflow imposes an implied prior selection procedure. While we include such a process as a part of our single-modality investigation, its execution in a generic multimodal classification setting is at best impractical and cumbersome. Secondly, the late, video-level combination of predictions of two different models requires two different and separate training runs to produce the distinct visual and audio models, rather than handling multimodal content in an end-to-end multimodal classification model.

5. EXPERIMENTS

This chapter describes the experimental evaluation of our proposed method for the task of multimodal video classification. It is structured as follows. First, we outline the datasets we use in the experiments in section 5.1.1. Secondly, in section 5.2 we run a series of preliminary experiments in order to determine model meta-parameters for the classification architectures described in section 4.2, namely the FC and LSTM workflows. We move on and apply the preliminary findings on the models of each workflow, and evaluate the latter on the single-modality setting 5.3 and the multimodal one 5.4. Finally, we discuss the obtained results in section 5.3.2.

5.1 Datasets and implementation

5.1.1 Datasets

We use a number of datasets to evaluate our methods, presented below.

1. The UCF-101 [118] human action recognition dataset consists of 13320 YouTube [1] videos spanning 101 categories. These categories can be grouped in 5 broad supersets, namely Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments and Sports. Video quality varies, both in terms of visual and audio content quality. We divide the videos into the train / test sets using the respective partitions used in [23], resulting in 9537 and 3783 videos for training and testing, respectively. Since our multimodal workflows utilize the visual and audio components of videos, we discard a number of the 101 classes for which no videos with audio content exist. After this filtering we end up with 51 classes for which 6837 videos with audio content exist, corresponding to 4839 and 1944 train / test videos. We call this modified dataset UCF-51. We apply a further refinement process from UCF-51, restricting the dataset to 10 classes with the least samples, arriving at UCF-10 consisting of 1097 videos, split to 783 and 314 train / test videos. We apply UCF-10 for a set of preliminary experiments to determine workflow model parameters, described in 5.2.

2. The KTH dataset [109] is a human actions dataset, consisting of 6 classes. It consists of 599 videos, split to 389 and 210 train / test videos. In KTH, videos are simple, with no occlusion and with the actor centered in simple scenes. In addition, the videos lack both color and audio, so we use this dataset only for the first stated goal of this study (i.e. handling of the temporal content) and not for the audiovisual multimodal workflows.

3. Audioset [32] consists of videos from YouTube, forming a hierarchical ontology of audio-centered events of 632 classes. The dataset was manually downloaded via

[1] `https://www.youtube.com`

the provided video URLs, with the following two filtering preprocessing steps. First, since we deal with single-label classification, we restricted the video classes to leaf classes in the ontology tree, restricting classes to non-abstract, specific events. Secondly, we keep classes that are annotated with a high quality, with respect to the provided class-wise annotation quality index [2]. Specifically, we kept a quality value of 1.0, resulting in 43 retained classes. We downloaded the respective videos from YouTube via the provided urls with the youtube-dl tool [3], where we discovered that not all videos were available via the listed urls. Downloading resulted in 2602 train and 2618 test videos, with notable imbalance among the class samples.

4. The Columbia Consumer Video (CCV) dataset [56] consists of user videos from YouTube, which were manually downloaded from provided video URLs. It consists of 20 diverse classes (e.g. human activities, scenes, objects) and provides pretrained hand-crafted audiovisual features (SIFT, STIP, MFCC features). As with Audioset, the videos are provided via a list of YouTube video identifiers, which we downloaded with the youtube-dl tool and from which multiple videos were either missing or not accessible. In the end, we managed tot obtain 2708 train and 2759 test videos.

The datasets and their description are summarized in table 2, along with information about the minimum and maximum available samples per class, as a result of the download and filtering steps above. In summary, we collect a variety of diverse datasets in terms of video classification task (e.g. action, event, scene, object recognition), number of classes, video quality (e.g. image / sound artifacts and noise, audio relevance to the ground truth class), classification difficulty (e.g. per-class variability in, e.g., video angle, pose, actors, environment and background), completeness (e.g. manually and partially downloaded data versus complete list of provided data).

5.1.2 Implementation

We used `python3` and `tensorflow` 1.4 [4] [25] to construct a complete video classification suite, including dataset preprocessing, serialization, classification, large-scale experiment execution and monitoring tools. The code is available on GitHub [5]. We used the "Caffe reference" implementation [6] for the DCNN Alexnet model. We used tensorflow's `TFRecord` format for data serialization and the `ffmpeg` utility to extract visual frames and mp3 audio files from a given video. Regarding spectrogram extraction, we used the `pyAudioAnalysis` [7] library. Experiments ran on an Ubuntu 16.04 system, with a Tesla K40 GPU.

[2] `https://research.google.com/audioset/dataset/index.html`
[3] `https://rg3.github.io/youtube-dl/`
[4] `https://www.tensorflow.org`
[5] `https://github.com/npit/video-learning-tf`
[6] `http://caffe.berkeleyvision.org/model_zoo.html`
[7] `https://github.com/tyiannak/pyAudioAnalysis`

Table 2: The datasets used in the experimental evaluation.

dataset	classes	video instances		min / max samples per class		notes
		train	test	train	test	
UCF-51 [118]	51	4893	1944	76 / 120	28 / 48	human activity classes poor / irrelevant audio
UCF-10 [118]	10	783	314	76 / 83	28 / 34	human activity classes poor / irrelevant audio
KTH [109]	6	389	210	64 / 65	35 / 35	human activity classes no color or audio
Audioset [32]	43	2602	2618	8 / 226	19 / 170	audio-centric classes imbalanced
CCV [56]	20	2708	2759	41 / 287	43 / 295	diverse classes

5.2 Preliminary experiments

Prior to our main experimental evaluation, we perform a set of preliminary experiments to determine network parameters for each workflow. We use the visual data of the UCF-10 dataset for this task, training with cross-entropy loss and mini-batch stochastic gradient descent [13]. We empirically tuned the training meta-parameters to the following values. We use a mini-batch size of 5 videos (i.e. 160 frames) and a learning rate of 0.1, exponentially decreasing 100 times on the course of training. We train with an early stopping criterion where the model with the best test performance is retained within an overall training process up to 10 epochs. We evaluate classification with the accuracy measure. For the UCF-10 dataset, the majority and chance accuracy baselines are 10.8% and 10.0%, respectively.

5.2.1 FC workflow

First, we select meta-parameters for the FC workflow. To this end, we execute video classification runs with early and late fusion. In addition, we vary the frame encoding layer to one of the fully-connected layers of the encoder DCNN (i.e. the Alexnet network) to either the fc6 or the fc7 layer. Given the identical layer types for frame encoding and classifier in the FC workflow (i.e. a fully-connected layer), we omit the fc8 layer from candidate frame encoding layers. We obtain the results displayed in table3. We observe that we get a better performance with late fusion rather than to early fusion, regardless of the encoding layer by which we produce frame vector embeddings. In addition, selecting the fc7 layer for the encodings outperforms the fc6 choice by a large margin, in both early and late fusion. In light of these results, we select the fc7 encoding layer and late frame fusion for the FC workflow, for the main experiments in sections 5.3 and 5.4.

Table 3: Results of the preliminary experiments for the FC workflow, in terms of classification accuracy (higher is better), varying the frame encoding layer and the frame fusion strategy. Bold values denote row-wise maxima.

fusion strategy	late		early	
encoding layer	fc6	fc7	fc6	fc7
accuracy	0.76115	**0.79618**	0.652866	0.79299

5.2.2 LSTM workflow

For the LSTM workflow, we need to avoid full grid search due to the large number of parameters and possible values. Instead, we set a baseline configuration with sensible values from related literature on training LSTM networks. To find optimal values, we vary one parameter at a time, run a set of experiments for each possible values and select the best performers for the final setting. The baseline configuration consists of avg LSTM output fusion, the fc7 Alexnet layer for frame encoding and a 3-layer LSTM network with a hidden state of 200 neurons. To prevent overfitting, we apply a dropout mechanism [120] at the LSTM output set to a 0.5 probability threshold.

Given this baseline configuration, we vary subsets of parameters and examine performance across different settings, keeping the rest at their baseline values. In table 4, we investigate the effect of the number of LSTM layers and the fusion method (i.e. avg, $last$ or $state$ LSTM fusion), with respect to classification accuracy.

Table 4: Fusion results of the preliminary experiments for the LSTM workflow, in terms of classification accuracy (higher is better), varying the number of LSTM layers and the output fusion strategy. Bold values indicate row-wise maxima.

number of layers	1	2	3
	avg fusion		
accuracy	0.09554	**0.84076**	0.76752
	last fusion		
accuracy	0.09554	**0.8121**	0.78025
	state fusion		
accuracy	0.09554	0.76115	**0.7707**

We can observe the following from the results:

- A single hidden layer completely fails to train the network, resulting in an accuracy score close to to random chance (i.e. 1 in $|C| = 10$, where C the number of labels for UCF-10). Given the improved scores of the 2-layer and 3-layer LSTMs for all fusion methods, the most likely scenario is that a single layer LSTM is incapable of capturing discriminating factors in the sequence in this experimental setting, thereby severely underfitting the model.
- A 2-layer model outperforms a 3-layer setup for avg and $last$ fusion approaches,

possibly due to introduced overfit by the additional layer. This, however, is not the case with $state$ fusion, where the 3-layer model performs better.

- Comparison between the fusion methods is not clear, with avg fusion outperforming $last$ with a 2-layer LSTM, and vice versa for a 3-layer model. It is noteworthy that averaging the LSTM outputs outperforms simply taking the last output of the network, which indicates that temporal inter-dependencies of the preceding inputs were not fully and / or correctly captured in the hidden state representation. This thus leads to improved performance when explicitly taking into account past steps, than trusting the network to encode them into the last output prediction vector. $state$ fusion outperforms avg for a 3-layer LSTM, while for a 2-layer model it is surpassed by the other fusion methods.

- The best performance is obtained with avg fusion with a 2-layer network, with an accuracy of 84%. This is thus the fusion method and number of network layers we will select for the main experimental evaluations.

Moving on, in table 5 we investigate the effect of the LSTM hidden layer size, e.g. the number of memory neurons in the network. We vary the size to values in $\{200, 500, 1000, 2000\}$, observing optimal accuracy results for a hidden layer with 500 neurons. Doubling the size to 1000 reduces performance by approximately 2%, while dropping it to 200 neurons incurs a deterioration of almost 7%. For a 2000-sized hidden layer the training process fails to converge, degenerating to chance-level classification performance.

Table 5: Hidden layer size results of the preliminary experiments for the LSTM workflow, in terms of classification accuracy (higher is better). Bold values indicate row-wise maxima.

hidden layer size	200	500	1000	2000
accuracy	0.75796	**0.82166**	0.80255	0.09554

Regarding frame encoding for the LSTM, we vary the selected DCNN layer to all available fully-connected components, namely fc6, fc7 and fc8. In contrast to the FC workflow, we choose to include the pretrained fc8 classification layer in the encoding candidates, since the LSTM network applies a classification process qualitatively different than fully-connected classification in the FC workflow. Results in table 6 illustrate that the fc6 layer fares best with respect to classification accuracy by a significant margin. The fc7 performs worst, while the vastly shorter (10-dimensional, since the class set for UCF-10 is 10) fc8 layer performs at about 10% reduced performance than fc6.

Table 6: Encoding layer results of the preliminary experiments for the LSTM workflow, in terms of classification accuracy (higher is better). Bold values indicate row-wise maxima.

encoding layer	fc6	fc7	fc8
accuracy	**0.82484**	0.69745	0.72611

In light of the above findings, we set the LSTM network in the LSTM workflow to the following configuration: fc6 as the DCNN encoding layer, avg LSTM output fusion and 2 hidden layers with 500 neurons.

5.2.3 Summary

With the completion of the preliminary experiments, we have arrived on a configuration for both the FC and LSTM workflow. We present the parameter values in table 7.

Table 7: Tuned parameters for the FC (top) and the LSTM (bottom) workflow, arrived at by the preliminary experiments set.

FC workflow	
encoding layer	fc7
frame fusion	$late$
LSTM workflow	
encoding layer	fc6
lstm fusion	avg
lstm layers	2
hidden state size	500

We use architectures with these parameters for all instances of the two deep neural workflows realized in the main experiments of the sections that follow, for all modalities and multimodal configurations. It is thus important to emphasize that the limitations imposed and assumptions undertaken during this preliminary phase heavily affects the succeeding evaluations. Specifically, the search was incomplete for the LSTM workflow due to the intractability of full grid search for its parameters. In addition, we used the visual modality of the UCF-10 dataset to fit the classifiers, but the models will use the resulting configuration to handle not only visual data, but audio and multimodal input as well. This may hinder the performance of the LSTM workflow for audio data, as observed in the subsequent experiments and in section 5.3.2, specifically.

5.3 Single-modality experiments

In this section we describe the experiments with which we evaluate our proposed workflows on a single-modality setting, in an attempt to address the first goal of this study, as described in section 1.3. To recall, we seek to evaluate sequential and aggregation-based feedforward neural models, on their ability to utilize the temporal component in a video and the extent to which this contribution has an effect on video classification. In this study, the goal is tackled by comparing the LSTM and FC workflows, which represent the two aforementioned model categories, in the video classification task. We outline the experimental setup and the obtained results in section 5.3, followed by a discussion of the findings in section 5.3.1.

5.3.1 Results

In order to evaluate each proposed workflow, we setup video classification experiments for the datasets outlined in 5.1.1. We extract the visual and audio modality data and apply the FC and LSTM workflow on each one. Each model is trained with the same parameters for training, clip and frame extraction as in the preliminary experiments, which, under an empirical evaluation, were found adequate for the experiments in this section as well. To accommodate the larger datasets, we increase the mini-batch size to 20 videos, resulting in batches consisting of 640 frames with the frame extraction technique described in section 4.2.1. For each dataset examined, we compute the chance and majority classifier baselines. The first selects on of the available classes at random, while the second selects the majority class in the dataset samples.

5.3.1.1 KTH

In figure 21 we illustrate classification results for the KTH dataset, evaluating only visual video content, since the videos in the dataset do not contain audio. The green and purple bar stroke (edge) color denotes FC and LSTM workflow runs, respectively, for this figure and subsequent ones. In addition, dashed and dotted grey horizontal lines denote chance and majority classification performance baselines. We see that for KTH classification, despite ignoring frame inter-dependencies in the video sequence, the FC workflow outperforms the LSTM one by a relative 8.2% increase. This may due to the visual content in KTH videos being simple and lacking variation, as stated in 5.1.1, rendering the content simple enough for the fully-connected classifier to handle, and at the same time, for the LSTM classifier to overfit. Finally, the worst-performing proposed method introduces a four-fold increase on the highest baseline accuracy.

5.3.1.2 UCF-51

In figure 22, results for single-modality runs on the UCF-51 dataset are presented. Red and blue bars show visual and audio content respectively, a convention retained for the rest of this document. We can make a number of observations regarding the results:

- Visual content runs outperform audio content runs for both workflows, by more than double accuracy scores.
- The FC workflow outperforms the LSTM workflow by 5.2% with respect to audio content.
- The LSTM workflow outperforms the FC workflow with respect to visual content by a factor of 12%.
- The worst-performing proposed method achieves an approximately ten-fold increase on the naive classification baselines.

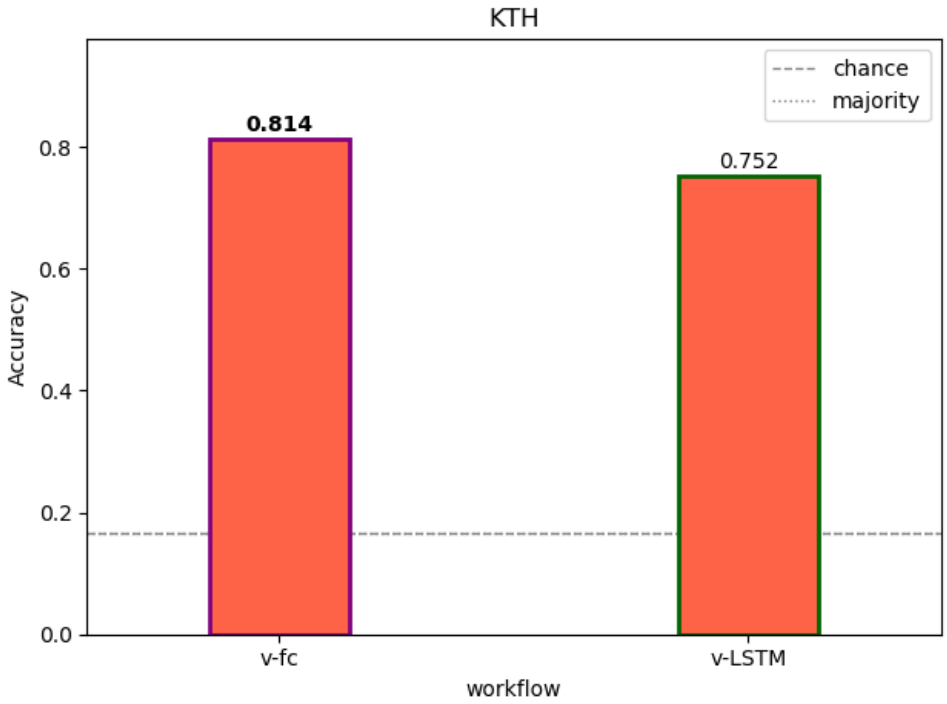

Figure 21: Single-modality results for the KTH dataset. Bars with a green edge color denote FC workflow runs, where purple bar edge colors denote LSTM workflow runs.

5.3.1.3 Audioset

Audioset results for our single-modality experiments are illustrated in figure 23, where we can make the following remarks:

- The FC workflow on audio data emerges as the best performer in the dataset, with an accuracy score of 34.1%, a 27.7% better score than the second best performer, i.e. the the LSTM applied on visual video content with 31%. We thus get no definitive results for this dataset, with no modality and workflow being consistently better than their competition.

- Regarding audio content, the FC workflow outperforms the LSTM workflow by a significant relative factor of 23.1%.

- For the visual modality the LSTM workflow fares better than the FC workflow by 16.1%.

- The worst-performing method increases the baseline performance by a factor of three.

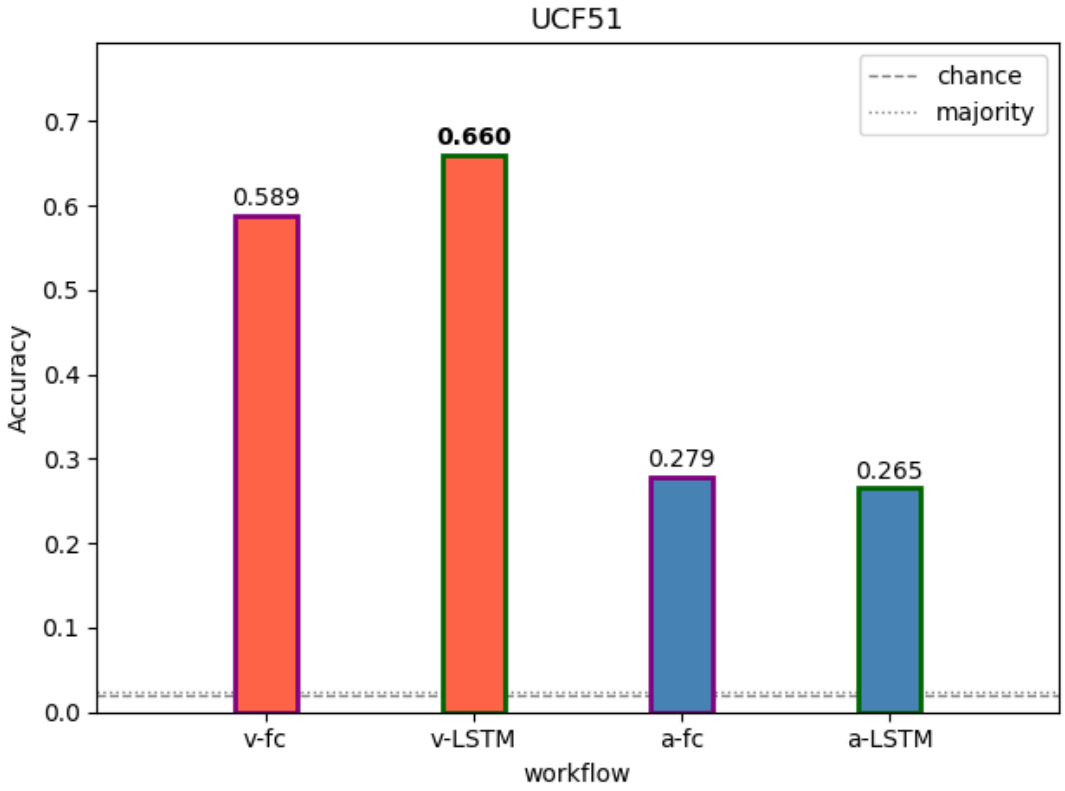

Figure 22: Single-modality results for the UCF-51 dataset. Red and blue bars represent visual and audio modality runs, respectively.

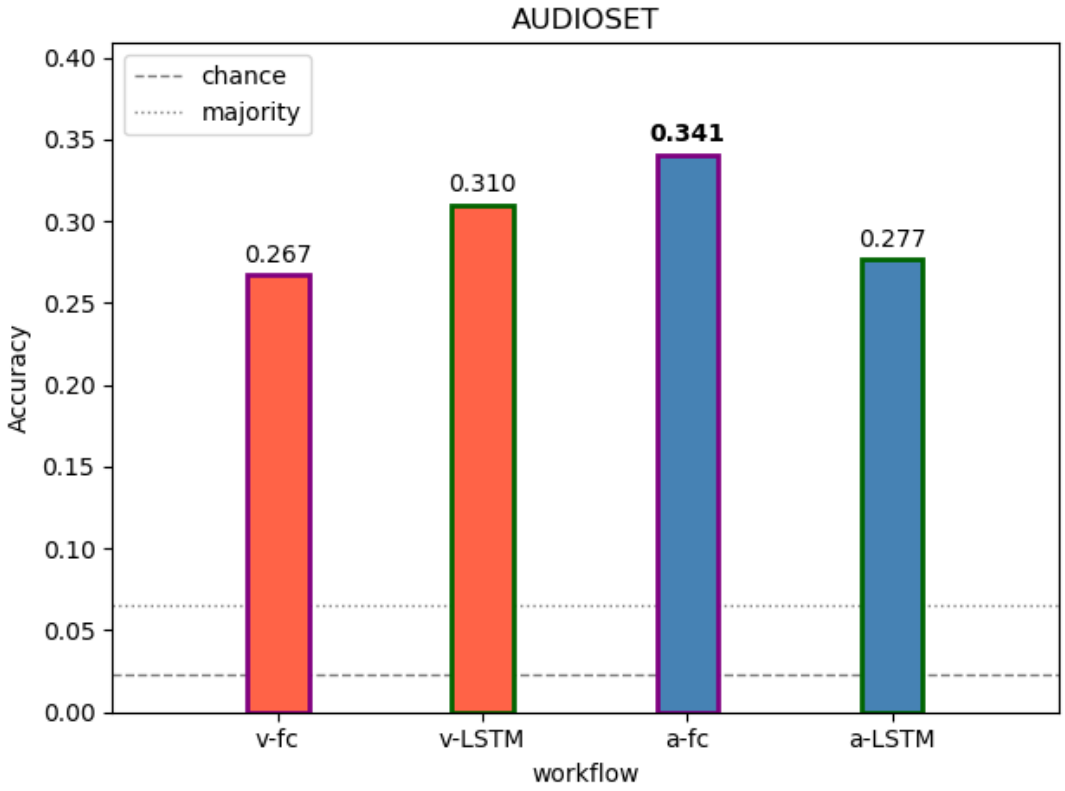

Figure 23: Single-modality results for the Audioset dataset.

Table 8: Single-modality collective results for all datasets. Bold values indicate dataset-wise maxima, while underlines indicate modality-wise maxima.

dataset	visual		audio		baseline	
	FC	LSTM	FC	LSTM	chance	majority
KTH	**0.814**	0.7523	N/A	N/A	0.1667	0.1667
UCF-51	0.5889	**0.6604**	0.2788	0.2654	0.0196	0.025
Audioset	0.267	0.309	**0.340**	0.276	0.023	0.064
CCV	0.653	**0.657**	0.367	0.342	0.05	0.106

Table 9: Hand-crafted single-modality results for the CCV dataset. Since the features provided correspond to a single video, only the FC workflow is applicable for classification.

visual		audio	baseline	
FC-SIFT	FC-STIP	FC-MFCC	chance	majority
0.491	0.390	0.304	0.05	0.106

5.3.1.4 CCV

Finally, we present the results for the CCV dataset in figure 24. Here, we additionally include performance of pretrained audiovisual features which were provided with the dataset. These features are handcrafted visual descriptors (SIFT [81], STIP [69]) and the MFCC descriptor [31] for audio. From the experimental results we can arrive at the following observations:

- Regarding hand-crafted features, the established SIFT descriptor outperforms STIP features by a factor of 25.5%. The proposed workflows outperform the handcrafted features, by a significant margin (with respective min / max relative improvements of 12.5% and 68%).
- The visual modality outperforms the audio modality, for all workflows and handcrafted features. For the proposed workflows, the relative performance difference is 77.7% and 92.1%, for the FC and LSTM workflows, respectively.
- The visual LSTM is the best performer, very closely followed by the visual FC with a 0.6% relative performance difference.
- Regarding audio, the FC workflow outperforms LSTM by a factor of about 7.6%.
- The worst performing proposed approaches introduces a more than double improvement on baseline classification.

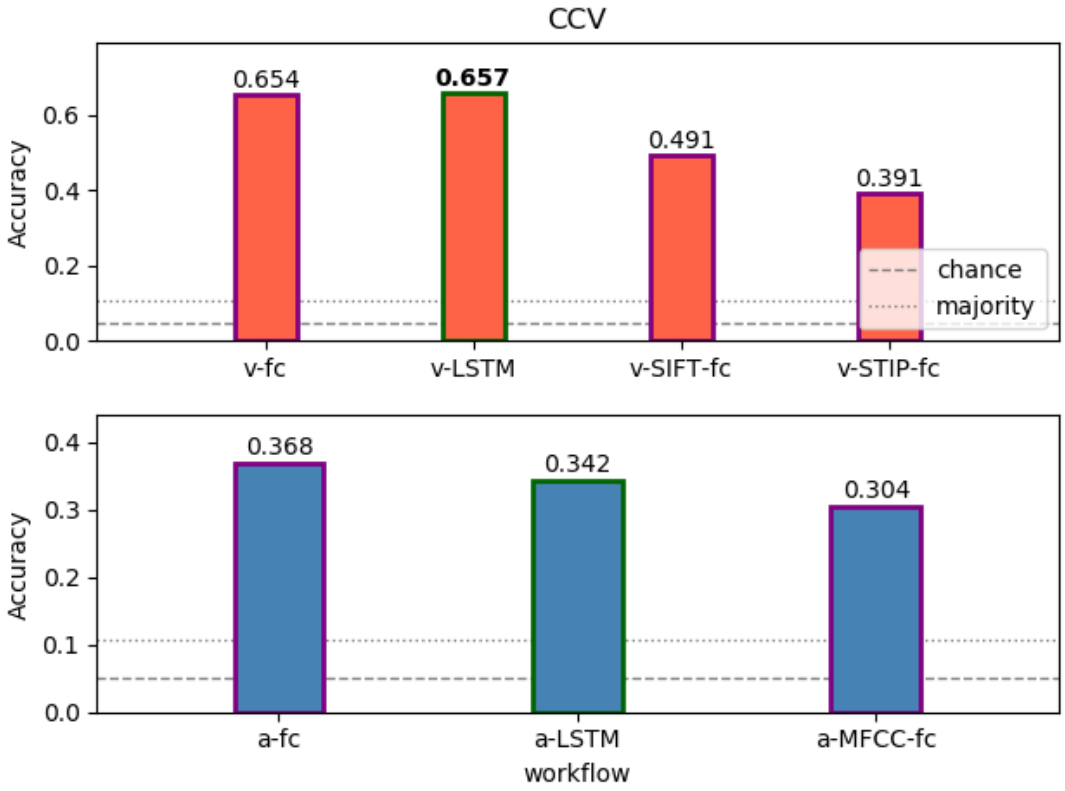

Figure 24: Single-modality results for the CCV dataset.

5.3.2 Discussion

Here we attempt to summarize the findings of the single-modality experiments and connect them to the goals of this study. The experimental results of the proposed methods are collectively illustrated in tables 8, with table 9 containing the results of the handcrafted features for the CCV dataset. Additionally, we illustrate relative comparison charts per modality and workflow (figure 26 and 25). Applied on the proposed methods of the thesis, the experiments described in this section can provide evidence on the suitability of the feedforward FC workflow versus the recurrent LSTM workflow that considers input interdependencies. In addition, the aforementioned suitability is investigated in a multimodal setting, i.e. for the visual and audio modalities, all with respect to the video classification task. Thus, in light of the experimental results for each dataset, we can arrive at the following conclusions:

1. The visual modality outperforms the audio modality everywhere but Audioset. This can be explained by the ontology and content of the latter, where both the videos and event classes are audio-related. In addition, the dataset was filtered to high-quality samples, limiting the occurrence of videos with unrelated audio (e.g. music, narration, overlain audio effects, as explained in section 5.1.1 which further enhances the significance of audio and reduces its potential to introduce harmful noise. It should be noted, however, that the performance difference between modalities in these cases is nowhere near similar. In figure 26, we depict the performance relationship of the two modalities for each dataset in our experiments, with respect of average and maximum achieved scores per dataset. For the cases where the visual modality outperforms the audio one, it does so with a mean relative accuracy difference of approximately 136% and 78%, for max modality performances in UCF-51 and CCV, respectively. However, the audio modality outperforms the visual one by approximately 10%, for Audioset. Similar workflow relationships can be observed when considering average modality performance per dataset, rather than the maximum. This hints at the visual modality being an important discriminatory information channel in video classification, as well as verifying the primacy of the visual modality in human perception, categorization and semantic segmentation.

2. The pretrained weights of the frame encoding Alexnet DCNN manage to provide a good initialization point not only for the visual modality, but for spectrogram image encoding as well. The latter can be illustrated by performance in Audioset, where the audio modality runs outperform the visual one. Had the representation been inadequate to capture spectrogram information, the audio modality runs would behave consistently poorly in all datasets.

3. In figure 25 we present the relationship of the examined workflows, both per utilized modality and dataset. We can see that the LSTM workflow outperforms the FC workflow on visual content for all datasets we examined, except in KTH. As expressed in the previous section, this may be attributable to the wealth of motion-related information in the visual modality, which can be effectively captured by the LSTM model –

the simplicity of KTH data causing the LSTM model to overfit on irrelevant video features. However, the FC workflow in turn outperforms the LSTM workflow on audio content, for all datasets examined. A possible explanation for this is that temporal input dependencies in audio spectrogram images are qualitatively different from visual motion cues, and they cannot be efficiently captured by the LSTM classifier with the current configuration. Another possible culprit could be the Alexnet vector encoding being inadequate for capturing spectrogram temporal inter-dependencies. Since the encoding approach yields good results with the FC workflow however, a more probable avenue for the efficient application of the LSTM workflow could be a modification of the training parameters or, as emphasized in the preliminary experiments summary (see section 5.2.3) the classifier's architecture itself, so as to bring about a better fit of the model on the encoded spectrogram input sequences.

4. The handcrafted features examined are outperformed by the proposed FC and LSTM workflows in both modalities. This verifies the superior expressive capabilities of DNN-based deep distributed features. This observation concerns the CCV dataset, which is the only dataset among the ones examined that provided handcrafted audiovisual features.

5.4 Multimodal experiments

In this section we tackle the second goal of this thesis (see 1.3), i.e. an examination of various multimodal approaches for the video classification task. While in the previous section we examined the proposed workflows for each modality separately, here we apply the proposed audiovisual fusion methods described in section 4.3. As in the previous section, we use data from the datasets described in section 5.1.1. We exclude the KTH dataset due to its lack of have audio content. In section 5.4.1 we describe the experimental setting and results per dataset, followed by a summary of the results in section 5.4.2.

5.4.1 Results

In the sections below we present the results of the multimodal experiments.

5.4.1.1 UCF-51

In figure 27, we present late-video fusion results for the UCF-51 dataset, combining the best-performing single-modality runs for the same dataset, outlined in the previous section. We adopt the following conventions regarding late-video fusion figures: We depict single-modality run performance baselines in dotted red and blue lines, for the visual and audio modalities respectively. The continuous light blue curve is the linear combination result, with the horizontal axis showing the visual modality weight w (the audio modality

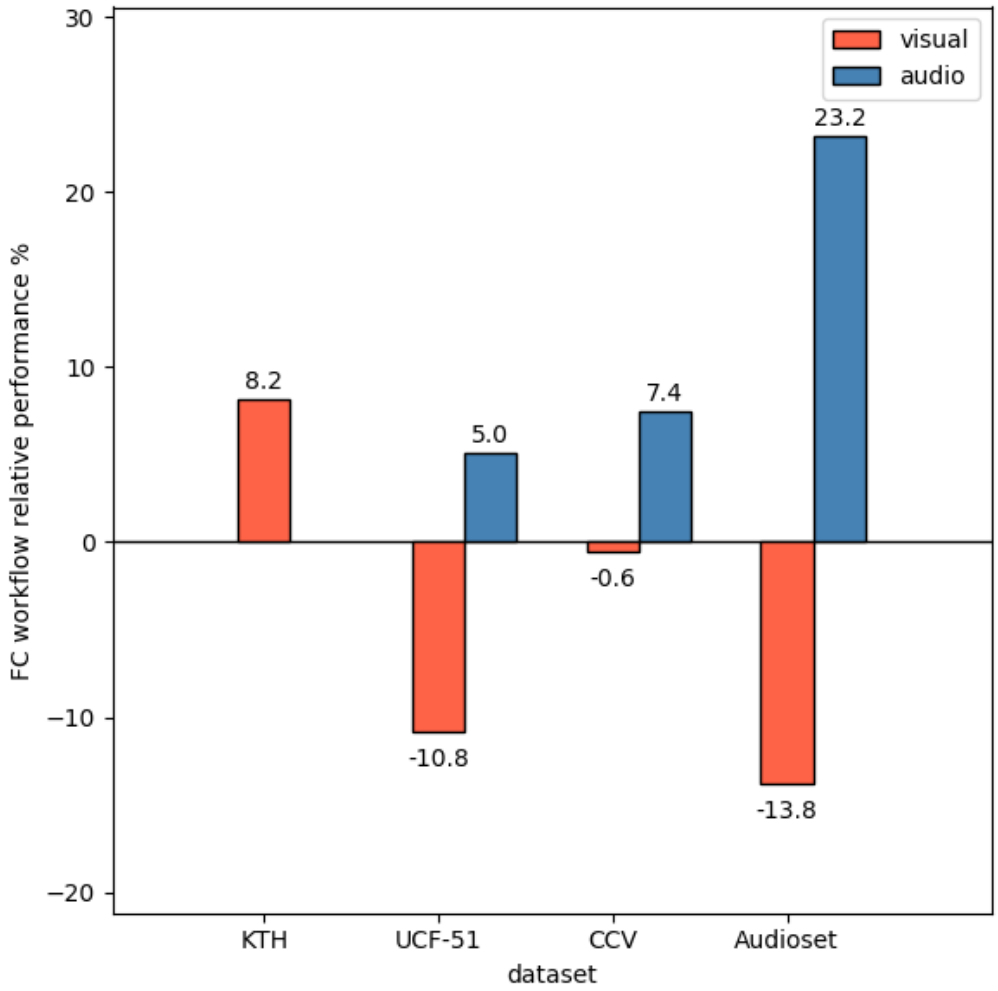

Figure 25: Relative performance comparison of the FC workflow to the LSTM workflow for the visual and audio modalities, per dataset. Values represent percentages, and higher values indicate the FC workflow outperforming LSTM workflow by a larger margin.

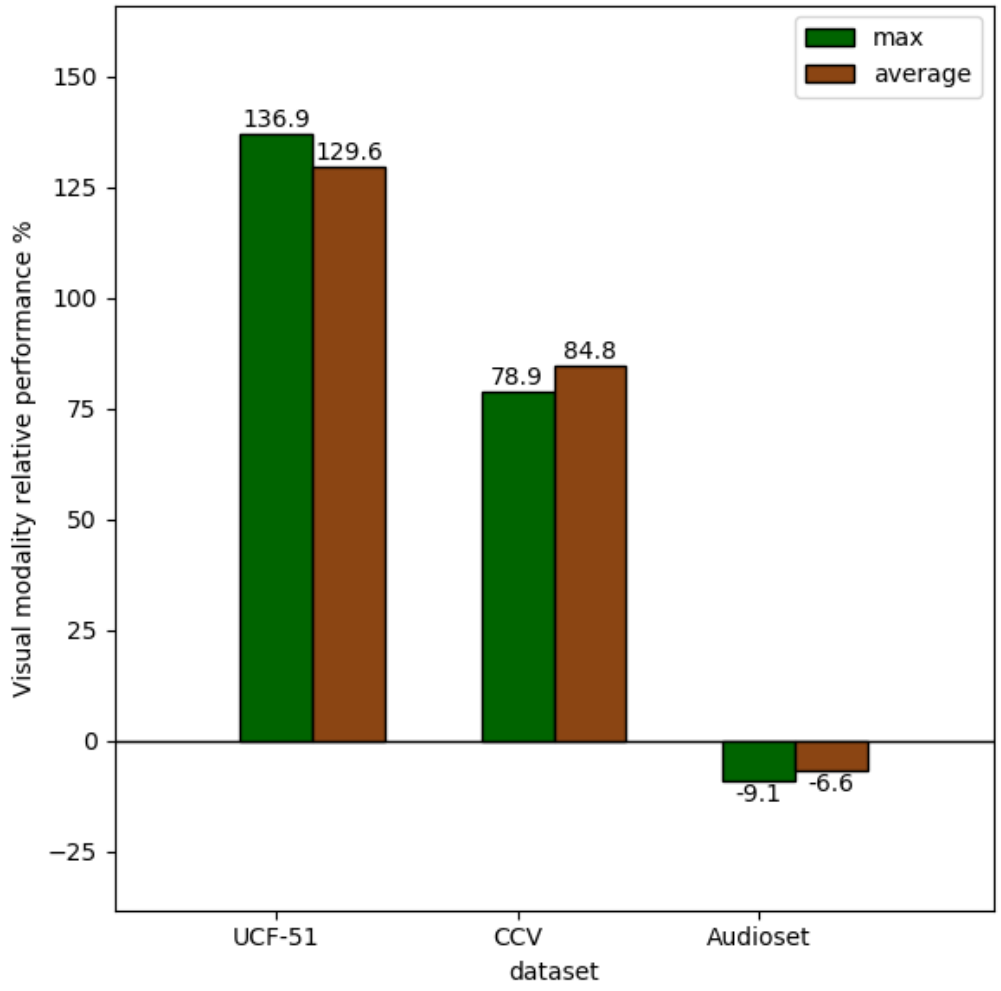

Figure 26: Relative performance comparison of the visual to the audio modality, per dataset. The comparison is performed via average and maximum modality scores per dataset, and values represent percentages. Higher values indicate the visual modality outperforming the audio modality by a larger margin.

always has a corresponding weight $1 - w$ and is omitted – see section 4.3.3 for details) at a 0.1 increment step. The best combination performance is shown with a vertical gray line to the visual weight value and the actual accuracy score overlaid on the performance curve. Finally, the continuous light green line represents the maximum late-video fusion result.

In addition, in figure 28 we present the multimodal fusion results, with the following illustration conventions: The multimodal FC and LSTM workflow fusion methods performance are shown in bars, following the bar and baseline classifier color conventions as in the single-modality figures (i.e. purple and green bar stroke color for FC and LSTM workflows, dashed and dotted grey horizontal lines for the chance and majority classification baselines, respectively). In addition, we note the performance of the best performing visual and audio single-modality runs for the dataset in red and blue horizontal dashed lines, respectively. Finally, we mark the performance of the best late-video fusion performance in a horizontal yellow dashed line.

We can make the following observations out of these two sets of results.

- In the multimodal fusion workflows, LSTM outperforms FC in all shared fusion methods (i.e. agv, $ibias$, max and $concat$).
- All LSTM fusion methods manage to outperform single-modality baselines, except $sbias$ fusion. On the other hand, FC fusion can not exceed the best single-modality visual run (with a relative 3.78% lower accuracy). All multimodal fusion methods outperform the best audio single-modality run.
- For FC fusion, the avg method performs best, followed by $concat$, $ibias$ and max, with each method having relative neighbouring performance differences of 2.75%, 1.95% and 2.71% respectively.
- Regarding LSTM fusion, the $concat$ method appears to fare best, followed by avg, max, $ibias$ and $sbias$ aggregation. Their relative performance difference is 0.84%, 1.43%, 3.72% and 3.22%, in the aforementioned order, respectively.
- For late-video fusion, we acquire a best linear combination performance of 0.72119 with visual weight 0.8, while maximum fusion performs significantly worse. The large visual weight in the combination is not surprising, given the performance different of the visual modality over the audio modality noted for UCF-51 in section 5.3.
- late-video fusion outperforms FC fusion, LSTM fusion and single-modality runs, achieving relative increase of 13.5% and 1.12% over the next best approaches in FC and LSTM fusion.

5.4.1.2 Audioset

For Audioset, late-video and multimodal-fusion results are illustrated in figures 29 and 30, respectively. Reviewing the classification results, we can arrive on the following remarks.

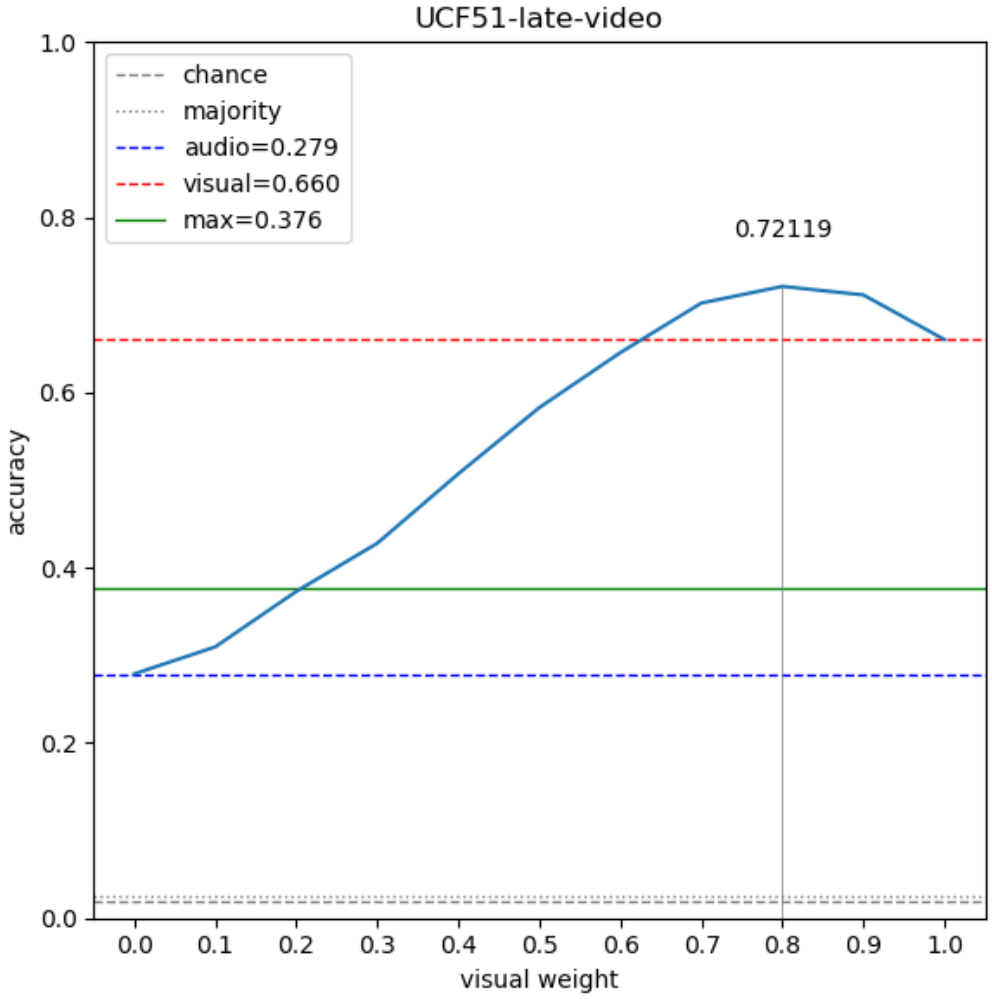

Figure 27: late-video multimodal fusion results for the UCF-51 dataset. The dotted red and blue lines depict the marginal visual and audio single-modality runs respectively, while the continuous light blue curve illustrates their linear combination. The vertical grey line marks the maximum performance visual weight of the combination (0.8, in this case).

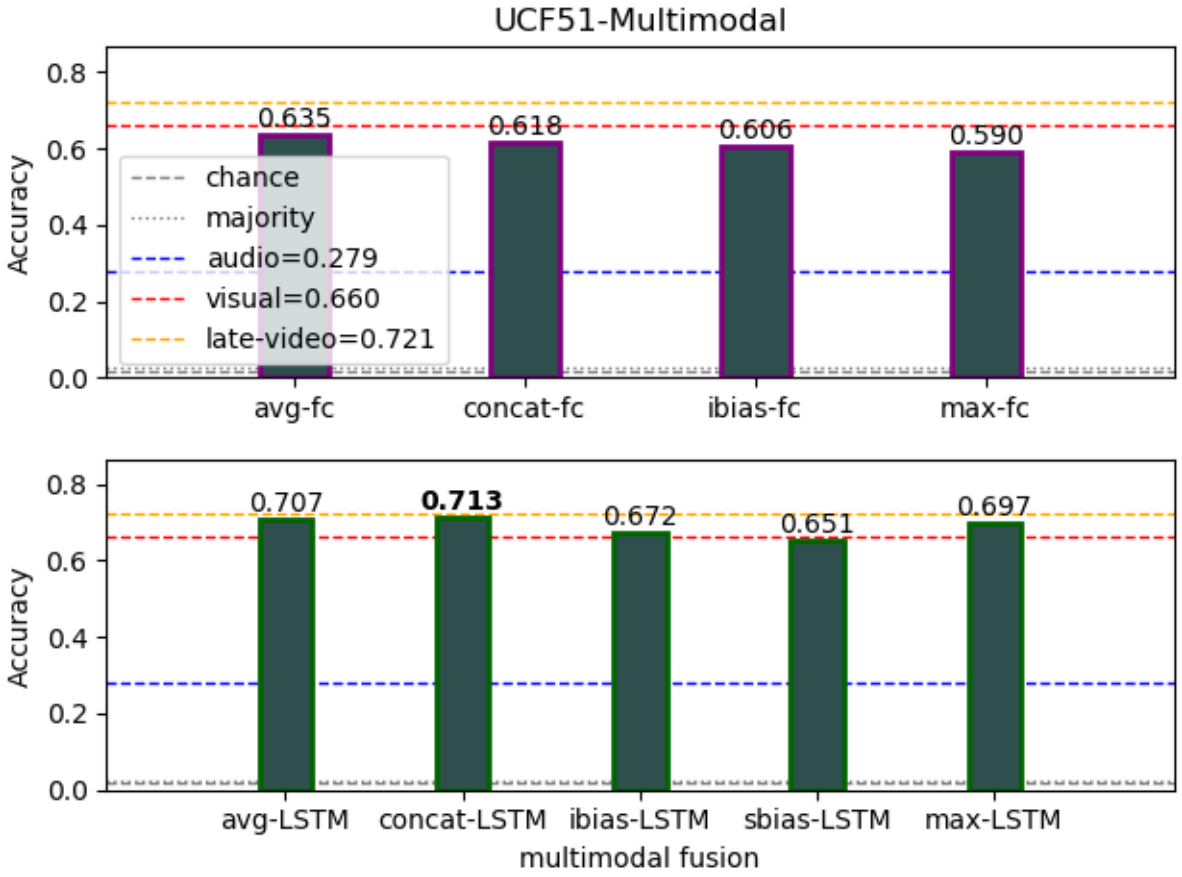

Figure 28: Multimodal fusion results for the UCF-51 dataset. The shorthands “avg”, “ibias”, “concat” and “sbias” refer to average, input-bias, concatenation and state-bias fusion, respectively.

- With respect to multimodal fusion workflows, the FC workflow outperforms LSTM in all shared fusion methods.
- All FC fusion methods outperform single-modality baselines, while LSTM fusion exceeds best single-modality performance only with the $input - bias$ fusion method.
- For FC fusion, the avg method performs best, followed by $ibias$, $concat$ and at a very low accuracy, max. Relative neighbouring performance differences of the methods lie at 1.04%, 1.58% and 96.8% respectively.
- Regarding LSTM fusion, the best method is $ibias$, followed by the $concat$, $sbias$, avg and max methods. Relative performance differences are 31.07%, 8.52%, 4.03% and 10.2%, in the aforementioned order, respectively.
- Regarding late-video fusion, an optimal combination performance of 0.46409 is obtained with a visual weight 0.6. It is noteworthy that despite the audio modality outperforming the visual modality for Audioset, the best combination results are obtained with a larger weight for the visual component. Maximum late-video fusion performs close above the single-modality scores, at an accuracy of 0.351.
- late-video fusion outperforms FC fusion, LSTM fusion and single-modality runs. The performance increase over the best proposed multimodal fusion runs are 19.58% and 26.34%, respectively.

5.4.1.3 CCV

Finally, we study results for the CCV dataset. Examining figures 31 and 32, we can arrive at the following observations:

- With respect to multimodal fusion workflows, the FC workflow outperforms LSTM in all shared fusion methods.
- All LSTM fusion methods outperform the single-modality visual baseline, except from $sbias$ aggregation. No FC fusion method manages the previous, however. Both workflows exceed the best audio single-modality run with all aggregation methods.
- LSTM fusion outperforms the FC approach for all shared fusion methods.
- For FC fusion, the avg method performs best, followed by $ibias$, $concat$ and max. The relative performance difference between the aggregation methods is 0.62% for the first two pairs, and 0.1% for the last.
- Regarding LSTM fusion, the best methods are $ibias$ and $concat$, followed by the avg, max and $sbias$ aggregation. The relative performance differences are 0.89%, 0.44% and 4.85%.

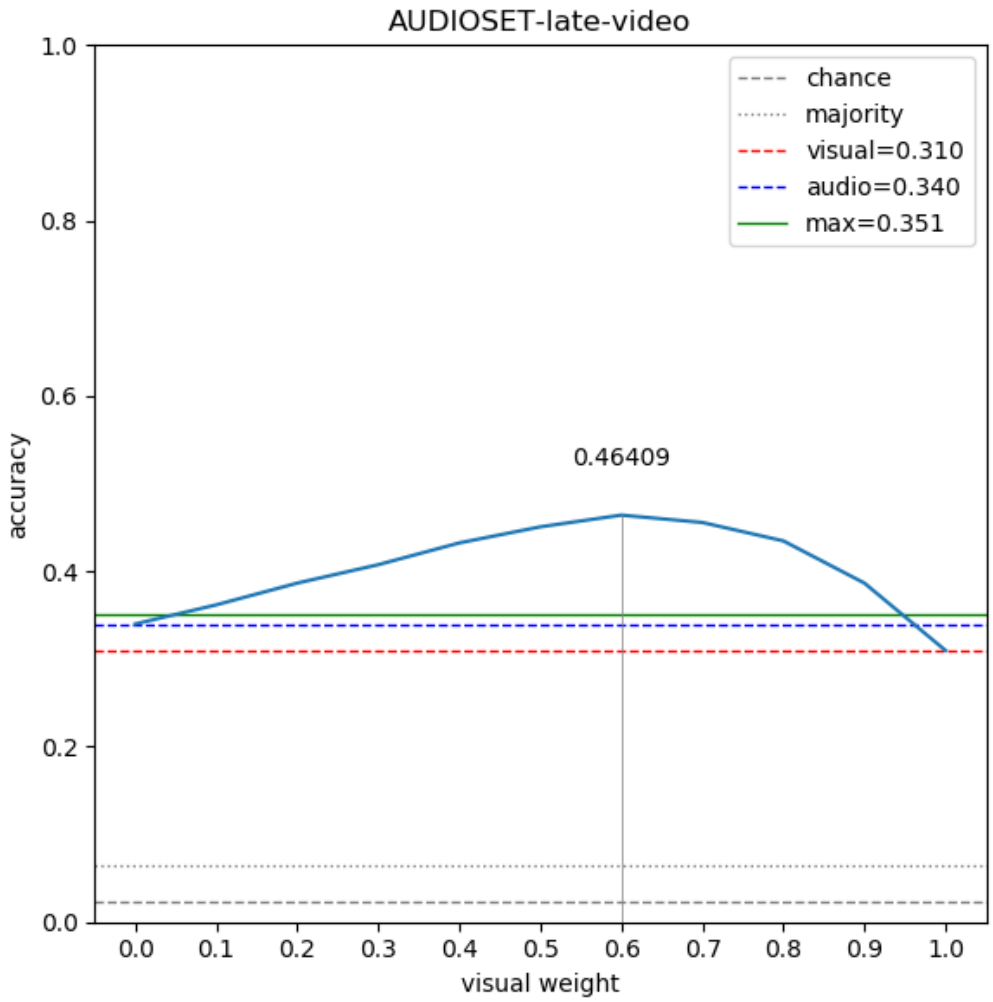

Figure 29: late-video multimodal fusion results for Audioset.

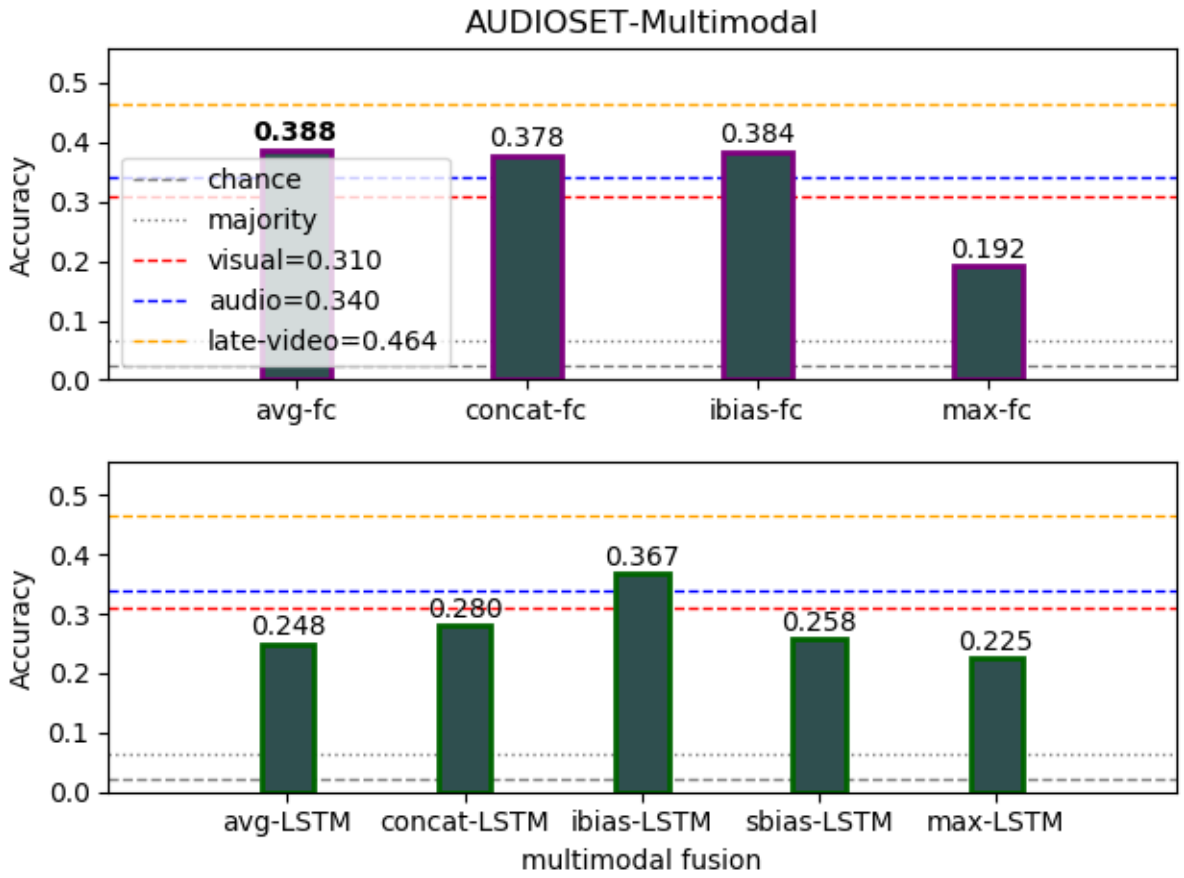

Figure 30: Multimodal fusion results for Audioset.

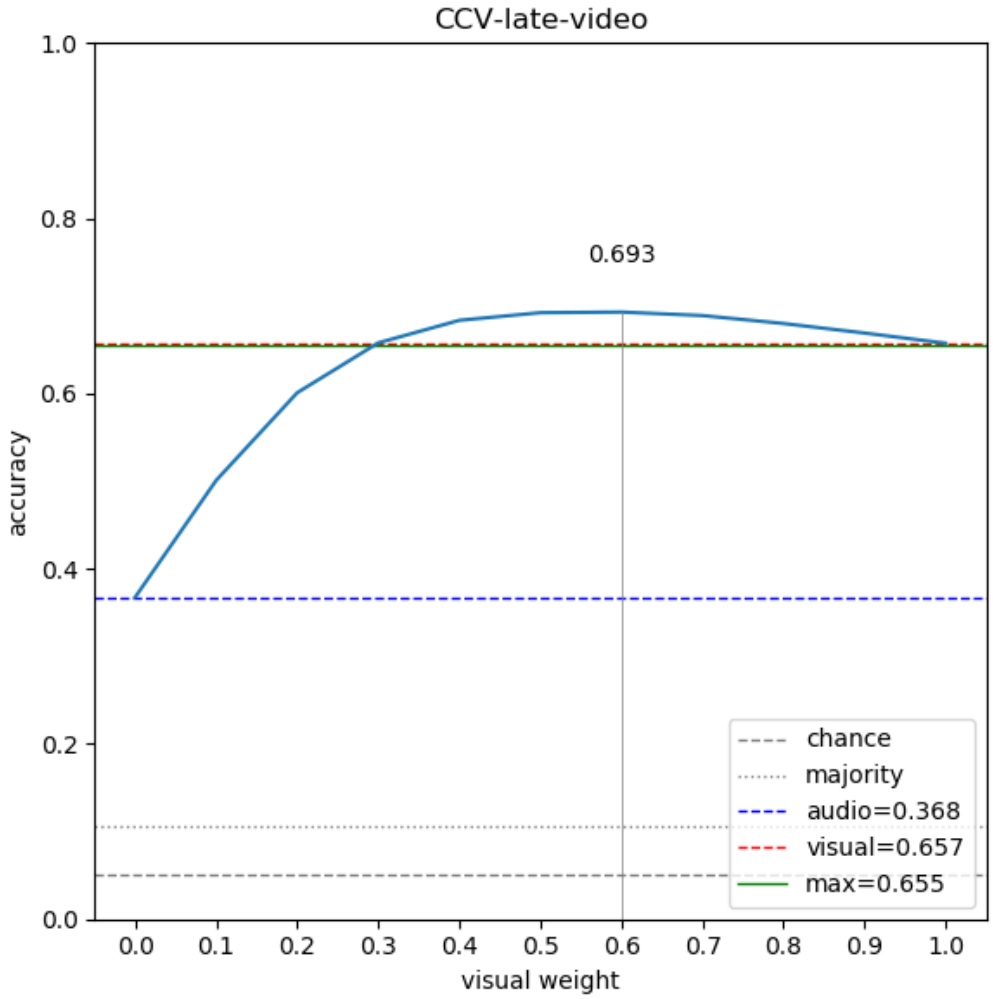

Figure 31: late-video multimodal fusion results for the CCV dataset.

- late-video fusion produces a best result of 0.693 at a linear combination with a visual weight of 0.6, while maximum fusion performs very close to the visual baseline, at 0.655 accuracy.
- late-video fusion outperforms FC fusion, LSTM fusion and single-modality runs, achieving 7.77% and 2.06% relative accuracy increases over the best proposed multimodal fusion runs, respectively.

5.4.2 Discussion

At this point we summarize the findings of the multimodal experiments, extending the investigation of the single-modality experiments to modality aggregation approaches for video classification and (multimodal classification in general). With respect to the thesis goals, we condense the experimental conclusions in order to address the two stated

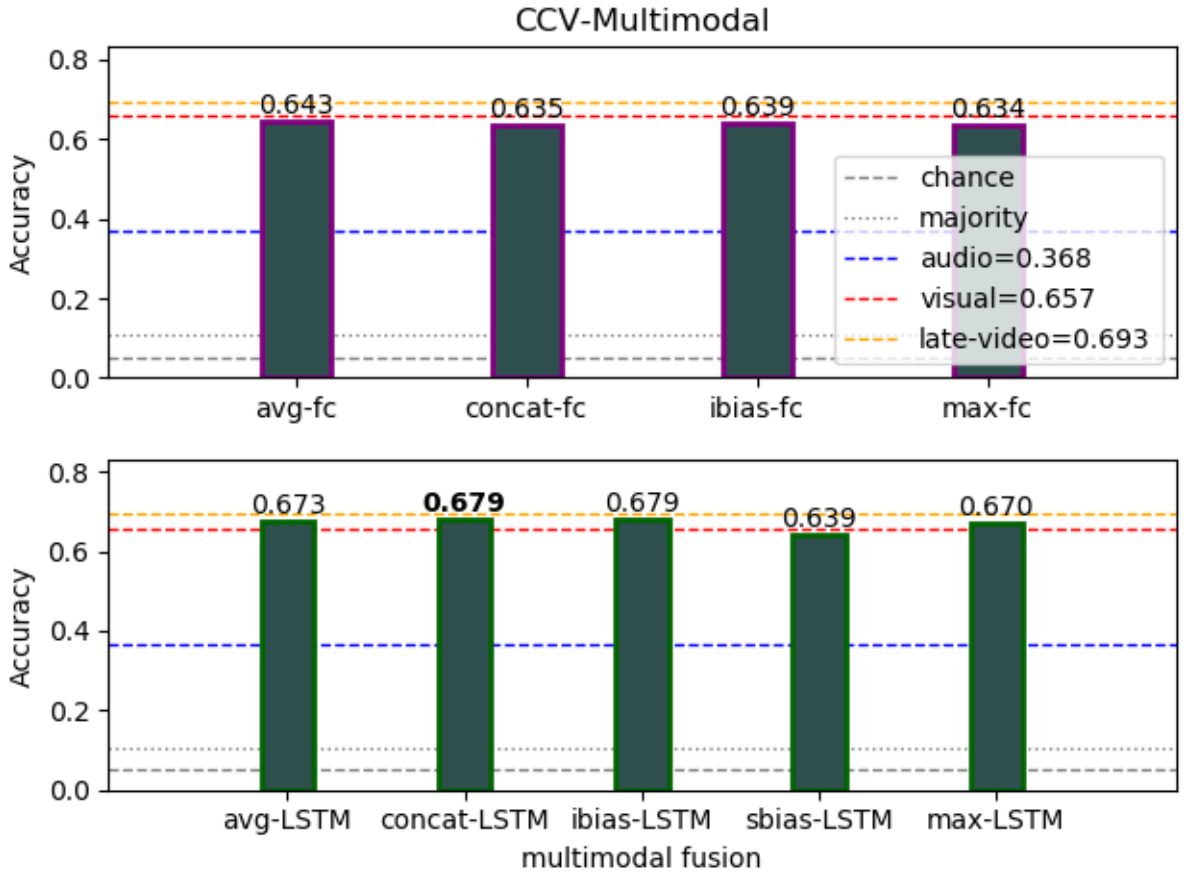

Figure 32: Multimodal fusion results for the CCV dataset.

Table 10: Multimodal fusion collective results for all datasets. Regarding the multimodal workflow-based runs (i.e. FC or LSTM runs, excluding late-video fusion), bold values indicate dataset-wise maxima, while underlined ones represent workflow-wise maximum values, for the given dataset. For late-video runs, LC denotes linear combination late-video fusion runs, with the optimal weight included in parentheses. In addition, max stands for maximum late-video fusion and we highlight late-video maximum values with italics.

dataset	FC				LSTM					late-video	
	avg	$concat$	$ibias$	max	avg	$concat$	$ibias$	max	$sbias$	LC (w)	max
UCF-51	0.635	0.618	0.606	0.590	0.707	**0.713**	0.672	0.697	0.65	*0.721* (0.8)	0.376
Audioset	**0.388**	0.378	0.384	0.192	0.248	0.28	0.367	0.225	0.258	*0.464* (0.6)	0.351
CCV	0.643	0.635	0.639	0.634	0.673	**0.679**	**0.679**	0.67	0.639	*0.693* (0.6)	0.655

Table 11: Multimodal fusion method average ranks, with respect to each dataset, the multimodal fusion workflow and overall. Bold values indicate row-wise minima (with respect to the rank reference), while underlined values indicate group and column-wise minima (with respect to the both fusion method and the reference type)

rank reference	fusion methods				
datasets	avg	$concat$	$ibias$	max	$sbias$
UCF51	**3.0**	**3.0**	5.0	5.0	4.0
CCV	**3.0**	3.5	3.5	5.5	5.0
Audioset	3.0	3.0	**2.0**	7.5	5.0
total	**3.0**	3.167	3.5	6.0	4.667
workflows	avg	$concat$	$ibias$	max	$sbias$
LSTM	3.0	**1.333**	2.333	4.0	4.667
FC	**3.0**	5.0	4.667	8.0	N/A
total	**3.0**	3.15	3.5	6.0	4.667

questions of the former: first, the comparison of feedforward (i.e. the FC workflow) and recurrent (i.e. the LSTM workflow) deep neural models, with respect to their ability to capture the temporal video components, and secondly, the investigation of the effect of multimodal approaches, for the video classification task.

We make the following observations, given the collected results of the multimodal experiments displayed in table 10, the extracted average rank information of the multimodal fusion methods used, in table 11 and a chart of the relative performance of the multimodal fusion methods per proposed workflow in figure 33.

- The FC fails to surpass single-modality baselines for all datasets except Audioset, which is the only dataset where it outperforms the LSTM workflow. This can be explained by the affinity of the FC workflow for the audio modality and the audio-oriented data of Audioset, as explained in section 5.3.2.

- Regarding fusion methods for the FC workflow, the avg aggregation method is the top performer on average, followed by $ibias$ and $concat$. max fusion performs worst, for all datasets.

- The LSTM workflow outperforms the FC workflow, for all datasets but Audioset, as explained above.

- The LSTM workflow surpasses the single-modality best performing runs in every dataset with some aggregation method. However, no aggregation method can be chosen does so consistently.

- The LSTM workflow performs best, on average, when paired with the $concat$ fusion method, which is top performer in 2 out of 3 datasets examined. The rest of the methods in order of performance are $ibias$, avg, max and finally the $sbias$ method.

- A comparison of fusion method performance across workflows in figure 33 indicates that the LSTM workflow outperforms the FC workflow for every fusion method, in every dataset except Audioset. This reinforces the findings of the single-modality experiments, over the LSTM classifier's ability to capture temporal context in the input that contributes to video classification. This conclusion does not hold for Audioset, a possible reason being the characteristics of the dataset, as explained in 5.3.2.
- In total, simple frame averaging via avg fusion emerges as the best method on average, outperforming concatenation via $concat$ fusion. The large feature vector dimensionality of the latter possibly requires a more complex model and / or additional training, the former of which is reflected by the improved performance on the much more expressive LSTM model, when compared to the fc classification of the FC workflow. The sequence bias introduction approach via $ibias$ does not seem to provide a better fusion approach, although performing better on the sequence-oriented LSTM workflow. max fusion does not produce good results, indicating that marginal modality information should be combined, rather than discarded. Finally, the $sbias$ method is the worst performing fusion approach. This can probability be explained by the low dimensionality of the state vector (i.e. set to a number of neurons the number of desired classes), which, although it enabled the formation of model vector representation, it seemed to prevent the LSTM from retaining sequence-related information effectively.
- Regarding late-video fusion, the linear combination approach consistently outperforms the FC and LSTM multimodal workflows. We can identify a preference for an increased bias towards the visual modality (even for Audioset) with all optimal visual weights exceeding 0.5. However, no unique optimal weight pair can be deduced that works best across all datasets. A naive result can be obtained by averaging the accuracy scores across datasets and collecting the weights of the best aggregate accuracy. Doing this, we acquire a visual / audio weight pair of $(0.7, 0.3)$.

5.5 Comparison to the state of the art

In this section we present a brief comparison to related work on video classification, for the selected datasets. This comparison is not entirely valid and straightforward, since we use reduced class sets for some of these datasets for the reasons outlined in section 5.1.1. In addition, our experiments focus on answering the questions stated in 1.3, rather than aiming to surpassing the state of the art for each of the examined datasets and corresponding specific classification task. This results in an overall reduced performance per dataset. Despite these observations, we include the comparative results below, for completeness.

In classification experiments on the KTH dataset, the authors in [70] achieve a performance close to 90% using HoG features in a bagging configuration. More recent approaches adopt CNN features, such as the work in [98], where DCNN spatiotemporal

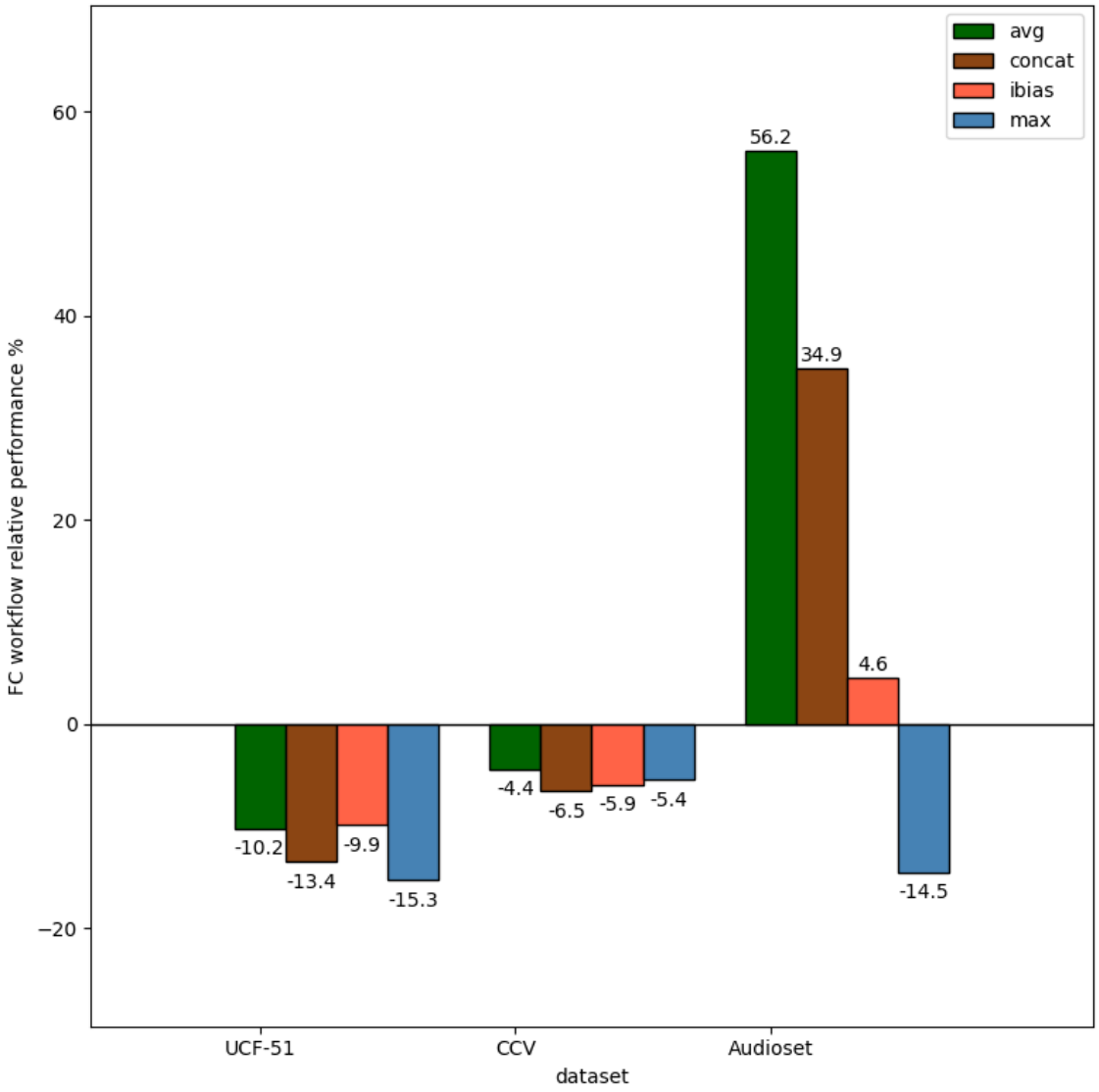

Figure 33: Relative performance comparison of the FC workflow to the LSTM workflow for each of the multimodal fusion methods, per dataset. Values represent percentages, and higher values indicate the FC workflow outperforming LSTM workflow by a larger margin.

features in a pyramidal hierarchy manage to push the state of the art closer to 95%. For the UCF-101 dataset, a significant performance increase is achievable via the inclusion of temporal features in the multimodal setting and by using deep models for feature generation. Specifically, accuracy scores above 87% can be approached, while multiple model fusion techniques result in a performances around 92% [134, 55]. Very recent approaches approach an accuracy of 95% [133].

For CCV, we already reported the superiority of the deep learned features on visual and audio content, when compared to the provided visual and audio handcrafted features. Recent optimal results using SIFT features are obtainable by Fisher vector aggregation, which is the approach adopted [90], producing results with an accuracy of 71.7%. Given deep approaches, the same trends of using temporal features (e.g. image flow and motion trajectories) as well as deeper convolutional models than Alexnet are popular in recent works. For example, a classification performance of 84% can be achieved with audio, video and temporal features, using a regularized context-analysis scheme in [55].

Regarding Audioset, the deep audio features provided with the dataset (extracted with a VGG model in the same way as in [42] and pretrained on YouTube-8M dataset [1]) reach an accuracy of 46% when evaluated with the FC workflow and a mAP 0.31 as reported in [42]. Other approaches use deep DCNNs with multi-label training, taking advantage of the class ontology hierarchy of the dataset [68], reaching a mAP of 0.21 or using an attention mechanism in [62, 139], reporting a mAP of 0.327 and 0.36, respectively.

Possible extensions to the work of this study are outlined in the following section (6.3) and provide a number of ways and generic guidelines of approaching the aforementioned performance improvements.

6. CONCLUSIONS AND FUTURE WORK

This section concludes the thesis by offering a summary of the goals, the proposed models and the conclusions that can be drawn from the experimental results. In section 6.1, we provide the aforementioned summary, providing a brief description of the problem tackled in the thesis, the stated goals and the proposed approaches to reach them. In section 6.2, we layout the main findings extracted from the experimental evaluation of the proposed methods. In section 6.3 we present a number of ways to extend the investigation performed in this work in various directions, in light of the technical details, method assumptions and acquired results.

6.1 Summary

In this study, we examined the multimodal video classification task, entailing the automated prediction of video labels relevant to the content present in the video. Given the inherent multimodal nature of the latter, this prediction process was designed to take into account different modalities, namely the visual, audio and temporal video data streams. We handle the audio and visual data directly, by extracting spectrogram and video frame sequences respectively, feeding them to a deep neural classifier. On the other hand, we consider the temporal context indirectly, by examining classification models with varying sensitivity to the temporal inter-dependencies of the input sequence items. Given these aspects of our classification pipeline, we set the following research goals for this study, with respect to video classification:

1. How do feedforward, aggregation-based models perform, compared to sequence-based models that consider temporal inter-dependencies? How does this relationship change for the visual and audio modalities?

2. What is the performance of the aforementioned models in the multimodal, audiovisual setting? How can modality data be combined to improve video classification score?

The first goal was examined by instantiating two video classification workflows representative of the described approaches: the feedforward FC and the sequence-aware LSTM workflow. Each approach was applied on multiple, diverse datasets after a set of preliminary experiments with which architecture meta-parameters were fine-tuned. Regarding the second goal, we applied these workflows with three general strategies of modality fusion. These combined the visual and audio data representation directly (“direct” data fusion), introduced a audio bias on the visual information sequence (sequence-bias techniques) or applied a late fusion on video-level prediction scores (late-video fusion). Both of these goals can be condensed into a formulation of a mutlimodal video classification baseline configuration, illustrating generic estimated of expected performance on each dataset, with a deep multimodal classification pipeline applicable to any video classification task.

6.2 Conclusions

Given the experimental evaluation results, we arrived at a number of conclusions per stated goal. For the comparative performance of the feedforward FC workflow and the recurrent LSTM, the latter approach is the more suitable choice for visual data in general, albeit the model can overfit very simple datasets. On the other hand, the FC workflow performs consistently better on audio content when represented by spectrogram images. Furthermore, while the relative significance of the visual and audio modalities for video classification depends on the underlying dataset and corresponding annotation, we found the visual modality to be the significant information channel, significantly outperforming the audio modality in virtually all datasets examined. The exception to this is the audio-inclined Audioset, where the tables are turned but with a far lesser performance difference. Finally, we verified the superiority of DNN-based learned representations with respect to handcrafted features, for the CCV dataset, where handcrafted features were provided.

Furthermore, we examined audiovisual fusion approaches for multimodal video classification, each examined with the aforementioned network architectures. Despite practical disadvantages, the late-video linear combination fusion approach produced the best multimodal fusion results, while, conversely, the max-pooled variant performs much worse, close to single-modality baselines. With respect to the other approaches, we identified the suitability of the avg and $concat$ fusion methods for the FC and LSTM workflows respectively and in general, amongst all fusion methods on average. Max-pooling modality fusion performed poorly here as well as in the late-video case. In addition, we concluded that the sequence-bias fusion methods examined – i.e. $ibias$ and $sbias$ – are not as effectively applicable in audiovisual video classification, as in the image description task. In addition, we verified the complementary relationship between the visual and audio modality, with the majority of multimodal approaches outperforming the best single-modality runs.

For a detailed discussion on the conclusions above, see sections 5.3.2 and 5.4.2. In general, the acquired results can be interpreted as a baseline performance, achievable by the utilization of the audio and visual modalities with each the proposed classification models presented here. Specialization to more complex models, more persistent, dataset-wise fine-tuning and approaches pertaining to each specific video classification task (e.g. human action recognition, event detection, e.t.c.) can improve these accuracy scores further. Possible avenues towards this are presented in the next section.

6.3 Future work

There is a number of ways this work can be extended. In the future we would like to utilize additional video modalities – such as directly utilizing temporal content, video text metadata or detected high-level objects in the video (e.g. faces or segmentation information) – and investigate their combination in the proposed workflows. This entails an extension of the dual-modality settings operated in this work (i.e. the sequence-bias approaches, which assume a “main” and “auxiliary” modality), towards incorporating multiple informa-

tion channels in the video. The “main” and “auxiliary” channels on audiovisual classification could be swapped, with the audio content considered the primary modality. This could be applied in datasets where the audio modality is the dominant one, such as Audioset. Furthermore, the best-performing late-video LC fusion could be modified in order to address its disadvantages. Namely, instead of training two separate models, the marginal modality models could be combined into a two-stream model with the streams combined via a learnable weight w. Regarding the clip extraction process, instead of random selection of frame sequences the clip extraction could be implemented in a sequential moving window in the video. In this scenario, temporal inter-dependencies could be exploited on the clip level – in addition to the frame level – with temporal fusion strategies being examined on this level as well. Regarding the frame encoding layer, the performance of model vector representations (like fc8 layer of the Alexnet DCNN) could be explored. As observed in the preliminary experiments in this study, these features could provide a good balance between classification performance and computational cost, since they produce low-dimensional but rich vectors. Furthermore, additional sequence fusion schemes could be investigated (e.g. RNN-based encoder fusion [44]), as well as alternative sequential deep neural models such as GRU [17]. Finally, deeper neural models could be used to encode image data, borrowing from the state of the art and recent advances in image recognition (e.g. state of the art approaches such as in [124, 41]), as well as more sophisticated optimization approaches than mini-batch Stochastic Gradient Descent, such as the Adam [60] or Adadelta [141] optimizers.

ABBREVIATIONS - ACRONYMS

NN	Neural Network
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DCNN	Deep Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
FC	Fully Connected
GRU	Gated Recurrent Unit
SGD	Stochastic Gradient Descent

TERMINOLOGY TRANSLATION

Neural Network	Νευρωνικό δίκτυο
Convolution	Συνέλιξη
Recurrence	Αναδρομή
Multimodal	Πολυτροπικός
Classification	Κατηγοριοποίηση
Gradient	Κλίση (συνάρτησης)
Aggregation	Συγχώνευση / συνδυασμός
Sequence	Σειρά / αλληλουχία

REFERENCES

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 31, 82

[2] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. Kaze features. In *European Conference on Computer Vision*, pages 214–227. Springer, 2012. 28

[3] Jonathan Allen. Short term spectral analysis, synthesis, and modification by discrete fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(3):235–238, 1977. 29

[4] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2009. 23

[5] Richard A Altes. Detection, estimation, and classification with spectrograms. *The Journal of the Acoustical Society of America*, 67(4):1232–1246, 1980. 30

[6] Relja Arandjelovic and Andrew Zisserman. All about vlad. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013. 28

[7] John L Barron, David J Fleet, and Steven S Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994. 31

[8] 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. 28

[9] Ron Bekkerman, Mikhail Bilenko, and John Langford. *Scaling up machine learning: Parallel and distributed approaches*. Cambridge University Press, 2011. 37

[10] Y. Bengio. Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning*, 2(1):1–127, nov 2009. 28

[11] Yoshua Bengio. Deep Learning of Representations for Unsupervised and Transfer Learning. In *JMLR: Workshop and Conference Proceedings*, volume 7, pages 1–20, jun 2011. 28, 37

[12] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. 39

[13] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010. 57

[14] Darin Brezeale and Diane J Cook. Automatic video classification: A survey of the literature. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(3):416–430, 2008. 27

[15] 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. 28

[16] Xu Cheng, Cameron Dale, and Jiangchuan Liu. Statistics and social network of youtube videos. In *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on*, pages 229–238. IEEE, 2008. 23

[17] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 39, 85

[18] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011. 37

[19] Costas Cotsaces, Nikos Nikolaidis, and Ioannis Pitas. Video shot boundary detection and condensed representation: a review. *IEEE signal processing magazine*, 23(2):28–37, 2006. 27

[20] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 27

[21] Jonathan Delhumeau, Philippe-Henri Gosselin, Hervé Jégou, and Patrick Pérez. Revisiting the vlad image representation. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 653–656. ACM, 2013. 28

[22] Nevenka Dimitrova, Lalitha Agnihotri, and Gang Wei. Video classification based on hmm using text and faces. In *Signal Processing Conference, 2000 10th European*, pages 1–4. IEEE, 2000. 32

[23] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. 28, 29, 39, 50, 55

[24] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 38, 48

[25] Martín Abadi et al. Dean, Tucker, Yu, and TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 56

[26] Laurene V Fausett et al. *Fundamentals of neural networks: architectures, algorithms, and applications*, volume 3. Prentice-Hall Englewood Cliffs, 1994. 33

[27] Daniel J Felleman and DC Essen Van. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991. 37

[28] Stefan Fischer, Rainer Lienhart, and Wolfgang Effelsberg. Automatic recognition of film genres. *Technical reports*, 95, 1995. 28, 29, 30, 32

[29] James L Flanagan. *Speech analysis synthesis and perception*, volume 3. Springer Science & Business Media, 2013. 30

[30] Kunihiko Fukushima. Neural network model for a mechanism of pattern recognition unaffected by shift in position- neocognitron. *Electron. & Commun. Japan*, 62(10):11–18, 1979. 37

[31] Sadaoki Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2):254–272, 1981. 65

[32] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017. 30, 55, 57

[33] Melvyn A Goodale and A David Milner. Separate visual pathways for perception and action. *Trends in neurosciences*, 15(1):20–25, 1992. 30

[34] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016. 39

[35] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1458–1465. IEEE, 2005. 28

[36] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013. 39

[37] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015. 39

[38] Guodong Guo and Stan Z Li. Content-based audio classification and retrieval by support vector machines. *IEEE transactions on Neural Networks*, 14(1):209–215, 2003. 29

[39] Raia Hadsell, Pierre Sermanet, Jan Ben, Ayse Erkan, Marco Scoffier, Koray Kavukcuoglu, Urs Muller, and Yann LeCun. Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics*, 26(2):120–144, 2009. 37

[40] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 28

[41] 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. 28, 85

[42] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017. 82

[43] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 131–135. IEEE, 2017. 30

[44] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006. 85

[45] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998. 39

[46] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 48

[47] Berthold Horn, Berthold Klaus, and Paul Horn. *Robot vision*. MIT press, 1986. 31

[48] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991. 34

[49] Weiming Hu, Nianhua Xie, Li Li, Xianglin Zeng, and Stephen Maybank. A survey on visual content-based video indexing and retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):797–819, 2011. 28

[50] Xuedong D Huang, Yasuo Ariki, and Mervyn A Jack. Hidden markov models for speech recognition. 1990. 31

[51] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, 1962. 37

[52] Giridharan Iyengar and Andrew B Lippman. Models for automatic classification of video sequences. In *Storage and Retrieval for Image and Video Databases VI*, volume 3312, pages 216–228. International Society for Optics and Photonics, 1997. 32

[53] Mihir Jain, Herve Jegou, and Patrick Bouthemy. Better exploiting motion for better action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2555–2562, 2013. 31

[54] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013. 29

[55] Yu-Gang Jiang, Zuxuan Wu, Jinhui Tang, Zechao Li, Xiangyang Xue, and Shi-Fu Chang. Modeling multimodal clues in a hybrid deep learning framework for video classification. *IEEE Transactions on Multimedia*, 2018. 31, 82

[56] Yu-Gang Jiang, Guangnan Ye, Shih-Fu Chang, Daniel Ellis, and Alexander C Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, page 29. ACM, 2011. 31, 56, 57

[57] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 50

[58] 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. 29

[59] Hyoung-Gook Kim, Nicolas Moreau, and Thomas Sikora. *MPEG-7 audio and beyond: Audio content indexing and retrieval*. John Wiley & Sons, 2006. 29

[60] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 85

[61] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008. 31

[62] Qiuqiang Kong, Yong Xu, Wenwu Wang, and Mark D Plumbley. Audio set classification with attention model: A probabilistic perspective. *arXiv preprint arXiv:1711.00927*, 2017. 82

[63] Irena Koprinska and Sergio Carrato. Temporal video segmentation: A survey. *Signal processing: Image communication*, 16(5):477–500, 2001. 27

[64] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24, 2007. 23

[65] Gunther Kress. *Multimodality: A social semiotic approach to contemporary communication*. Routledge, 2009. 24

[66] 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. 28, 37, 45

[67] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563. IEEE, 2011. 30

[68] Anurag Kumar, Maksim Khadkevich, and Christian Fugen. Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes. *arXiv preprint arXiv:1711.01369*, 2017. 82

[69] Ivan Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3):107–123, 2005. 65

[70] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 80

[71] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 37

[72] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 37

[73] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 1998. 36

[74] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, pages 1096–1104, 2009. 30

[75] Keansub Lee and Daniel PW Ellis. Audio-based semantic concept classification for consumer video. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1406–1416, 2010. 30

[76] Kart-Leong Lim and Hamed Kiani Galoogahi. Shape classification using local and global features. In *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, pages 115–120. IEEE, 2010. 28

[77] Seppo Linnainmaa. The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors. *Master's Thesis (in Finnish), Univ. Helsinki*, pages 6–7, 1970. 36

[78] Dimitri A Lisin, Marwan A Mattar, Matthew B Blaschko, Erik G Learned-Miller, and Mark C Benfield. Combining local and global image features for object class recognition. In *Computer vision and pattern recognition-workshops, 2005. CVPR workshops. IEEE Computer society conference on*, pages 47–47. IEEE, 2005. 28

[79] Zhu Liu, Jincheng Huang, and Yao Wang. Classification tv programs based on audio information using hidden markov model. In *Multimedia Signal Processing, 1998 IEEE Second Workshop on*, pages 27–32. IEEE, 1998. 29

[80] Zhu Liu, Yao Wang, and Tsuhan Chen. Audio feature extraction and analysis for scene segmentation and classification. *Journal of VLSI signal processing systems for signal, image and video technology*, 20(1-2):61–79, 1998. 29

[81] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 28, 65

[82] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 6, page 2, 2017. 50

[83] Foteini Markatopoulou, Nikiforos Pittaras, Olga Papadopoulou, Vasileios Mezaris, and Ioannis Patras. A study on the use of a binary local descriptor and color extensions of local descriptors for video concept detection. In *International Conference on Multimedia Modeling*, pages 282–293. Springer, 2015. 28

[84] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943. 33

[85] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 31

[86] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *European conference on computer vision*, pages 128–142. Springer, 2002. 28

[87] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010. 39

[88] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. 50

[89] Simon Moncrieff, Svetha Venkatesh, and Chitra Dorai. Horror film genre typing and scene labeling via audio analysis. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 2, pages II–193. IEEE, 2003. 32

[90] Markus Nagel, Thomas Mensink, Cees GM Snoek, et al. Event fisher vectors: Robust encoding visual diversity of visual streams. In *BMVC*, volume 2, page 6, 2015. 82

[91] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011. 27

[92] 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. 27

[93] Sinno Jialin Pan, Qiang Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010. 45

[94] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007. 28

[95] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *European conference on computer vision*, pages 143–156. Springer, 2010. 28

[96] Matti Pietikäinen, Topi Mäenpää, and Jaakko Viertola. Color texture classification with color histograms and local binary patterns. In *Workshop on Texture Analysis in Machine Vision*, pages 109–112. Citeseer, 2002. 27

[97] Nikiforos Pittaras, Foteini Markatopoulou, Vasileios Mezaris, and Ioannis Patras. Comparison of fine-tuning and extension strategies for deep convolutional neural networks. In *International Conference on Multimedia Modeling*, pages 102–114. Springer, 2017. 28

[98] Mahdyar Ravanbakhsh, Hossein Mousavi, Mohammad Rastegari, Vittorio Murino, and Larry S Davis. Action recognition with image based cnn features. *arXiv preprint arXiv:1512.03980*, 2015. 80

[99] Matthew Roach, John Mason, and Li-Qun Xu. Video genre verification using both acoustic and visual modes. In *Multimedia Signal Processing, 2002 IEEE Workshop on*, pages 157–160. IEEE, 2002. 28, 29, 30, 32

[100] Matthew Roach and John S Mason. Classification of video genre using audio. In *Seventh European Conference on Speech Communication and Technology*, 2001. 28, 29

[101] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958. 35

[102] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006. 28

[103] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011. 28

[104] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 36

[105] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 45

[106] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*, 2014. 39

[107] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988. 32

[108] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015. 24, 28, 37

[109] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004. 29, 55, 57

[110] Jianbo Shi and Carlo Tomasi. Good features to track. Technical report, Cornell University, 1993. 28

[111] Panagiotis Sidiropoulos, Vasileios Mezaris, and Ioannis Kompatsiaris. Enhancing video concept detection with the use of tomographs. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 3991–3995. IEEE, 2013. 31

[112] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 30

[113] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 28, 30

[114] Josef Sivic and Andrew Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):591–606, 2009. 28

[115] Alan F Smeaton, Paul Over, and Wessel Kraaij. Trecvid: Evaluating the effectiveness of information retrieval tasks on digital video. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 652–655. ACM, 2004. 29

[116] Cees GM Snoek and Marcel Worring. Multimodal video indexing: A review of the state-of-the-art. *Multimedia tools and applications*, 25(1):5–35, 2005. 25

[117] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402. ACM, 2005. 30

[118] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 30, 55, 57

[119] Donald F Specht. Probabilistic neural networks. *Neural networks*, 3(1):109–118, 1990. 33

[120] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 58

[121] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015. 28

[122] Shiliang Sun. A survey of multi-view machine learning. *Neural Computing and Applications*, 23(7-8):2031–2038, 2013. 30

[123] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*. MIT press, 1998. 35

[124] 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. 28, 85

[125] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014. 37

[126] Kevin Tang, Li Fei-Fei, and Daphne Koller. Learning latent temporal structure for complex event detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1250–1257. IEEE, 2012. 31

[127] Ba Tu Truong and Chitra Dorai. Automatic genre identification for content-based video categorization. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pages 230–233. IEEE, 2000. 28, 30, 32

[128] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015. 39, 50

[129] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013. 31

[130] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. 30, 31

[131] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4305–4314, 2015. 31

[132] Peng Wang, Rui Cai, and Shi-Qiang Yang. A hybrid approach to news video classification multimodal features. In *Information, communications and signal processing, 2003 and fourth pacific rim conference on multimedia. Proceedings of the 2003 joint conference of the fourth international conference on*, volume 2, pages 787–791. IEEE, 2003. 32

[133] Zuxuan Wu, Yu-Gang Jiang, Xi Wang, Hao Ye, and Xiangyang Xue. Multi-stream multi-class fusion of deep networks for video classification. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 791–800. ACM, 2016. 31, 82

[134] Zuxuan Wu, Yu-Gang Jiang, Xi Wang, Hao Ye, Xiangyang Xue, and Jun Wang. Fusing multi-stream deep networks for video classification. *arXiv preprint arXiv:1509.06086*, 2015. 31, 82

[135] Zuxuan Wu, Ting Yao, Yanwei Fu, and Yu-Gang Jiang. Deep learning for video classification and captioning. *arXiv preprint arXiv:1609.06782*, 2016. 27, 28

[136] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013. 30

[137] Junji Yamato, Jun Ohya, and Kenichiro Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 379–385. IEEE, 1992. 32

[138] Hao Ye, Zuxuan Wu, Rui-Wei Zhao, Xi Wang, Yu-Gang Jiang, and Xiangyang Xue. Evaluating two-stream cnn for video classification. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 435–442. ACM, 2015. 30, 49

[139] Changsong Yu, Karim Said Barsim, Qiuqiang Kong, and Bin Yang. Multi-level attention model for weakly supervised audio classification. *arXiv preprint arXiv:1803.02353*, 2018. 82

[140] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015. 28, 29

[141] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 85