Movie Genre Classification: A Multi-Label Approach based on Convolutions through Time

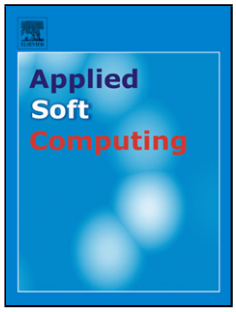
Article *in* Applied Soft Computing · August 2017

DOI: 10.1016/j.asoc.2017.08.029

1 author:

Jônatas Wehrmann
Pontifícia Universidade Católica do Rio Grande do Sul
29 PUBLICATIONS **386** CITATIONS

SEE PROFILE



Title: Movie Genre Classification: A Multi-Label Approach based on Convolutions through Time

Author: Jônatas Wehrmann Rodrigo C. Barros

PII: S1568-4946(17)30511-2
DOI: http://dx.doi.org/doi:10.1016/j.asoc.2017.08.029
Reference: ASOC 4419

To appear in: *Applied Soft Computing*

Received date: 5-1-2017
Revised date: 9-8-2017
Accepted date: 14-8-2017

Please cite this article as: Jônatas Wehrmann, Rodrigo C. Barros, Movie Genre Classification: A Multi-Label Approach based on Convolutions through Time, *<![CDATA[Applied Soft Computing Journal]]>* (2017), http://dx.doi.org/10.1016/j.asoc.2017.08.029

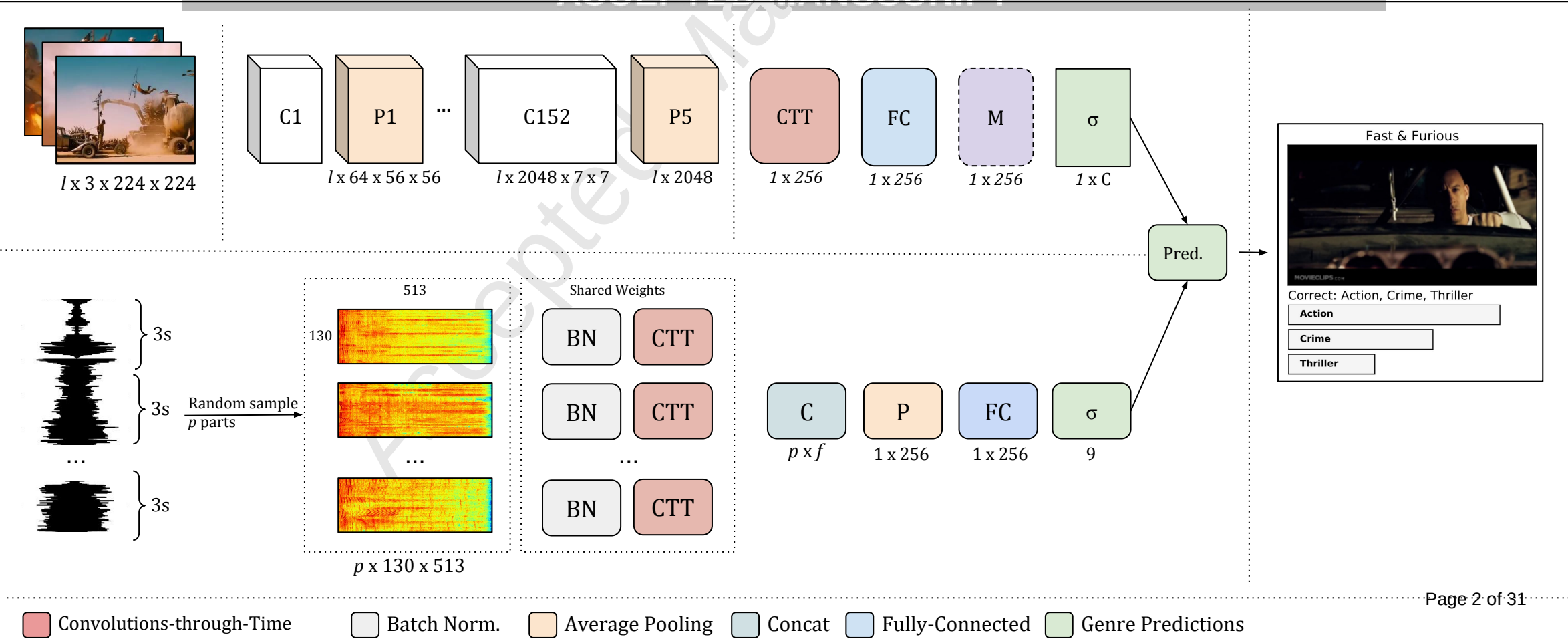
Highlights

Jônatas Wehrmann, Rodrigo C. Barros

June 20, 2017

We present a summary of our findings, as follows:

- Deep features allow better understanding the content of movies.
- CTT modules are capable of learning temporal data, outperforming LSTMs.
- CTT-based methods (but CTT-MMC-S) surpass baseline approaches by large margins.
- CTT-S (audio-based) models outperformed all low-level (audio and video) baselines.
- CTT-based methods outperform LSTMs for all genres albeit being far more simpler.



Movie Genre Classification: A Multi-Label Approach based on Convolutions through Time

Jônatas Wehrmann, Rodrigo C. Barros

Machine Intelligence and Robotics Research Group
Pontifícia Universidade Católica do Rio Grande do Sul
Av. Ipiranga, 6681, 90619-900, Porto Alegre - RS, Brazil

Abstract

The task of labeling movies according to their corresponding genre is a challenging classification problem, having in mind that genre is an immaterial feature that cannot be directly pinpointed in any of the movie frames. Hence, off-the-shelf image classification approaches are not capable of handling this task in a straightforward fashion. Moreover, movies may belong to multiple genres at the same time, making movie genre assignment a typical multi-label classification problem, which is *per se* much more challenging than standard single-label classification. In this paper, we propose a novel deep neural architecture based on convolutional neural networks (ConvNets) for performing multi-label movie-trailer genre classification. It encapsulates an ultra-deep ConvNet with residual connections, and it makes use of a special convolutional layer to extract temporal information from image-based features prior to performing the mapping of movie trailers to genres. We compare the proposed approach with the current state-of-the-art methods for movie classification that employ well-known image descriptors and other low-level handcrafted features. Results show that our method substantially outperforms the state-of-the-art for this task, improving classification performance for all movie genres.

Keywords: movie genre classification, convolutional neural networks, convolutions through time, multi-label classification

1. Introduction

The automatic analysis of video content is an important and challenging Computer Vision (CV) task, with the potential of helping human beings to solve a plethora of problems that are known to be too tedious, too expensive, or too time-consuming for them to solve on their own.

Much research effort has been lately devoted to building machine learning approaches for image classification, in which the goal is to perform single-label classification of images in a domain of a thousand labels [1, 2]. Video-based applications, on the other hand, have shown to be much more challenging, and many traditional and well-established machine learning algorithms have difficulties in properly handling it.

Recent work on video analysis [3, 4, 5, 6] have shown that a great many problems may be handled by Deep Convolutional Neural Networks (ConvNets) [7], which are the most popular method within the so-called *Deep Learning* paradigm [8]. These initial results are exciting and pave the way for many novel applications to be fully explored. Indeed, ConvNets are nowadays the state-of-the-art for many CV tasks (e.g., supervised image classification, localization and detection, and semantic segmentation), borrowing concepts from neuroscience to create a mathematical mechanism capable of assigning meaning to visual content. They consist of multiple layers of small sets of neurons that process portions of the input data (receptive fields), tiling the outputs so that their input regions overlap. The hierarchy of concepts that are sequentially learnt allows complex mappings from input to desired output. The well-known backpropagation algorithm is employed for training the multiple layers of neurons by backpropagating the gradients of a loss function with respect to the network's weights[1].

In this paper, we investigate the use of ConvNets for a very particular video

[1] We refer the interested reader to the book by Goodfellow et al. [8] for a great introduction to deep neural networks

analysis problem, movie genre classification, in which the goal is to automatically classify movies according to their genre (e.g., action, horror, drama, comedy) based on the content of their corresponding movie trailer. Movie genre classification is a much more challenging task than object classification and detection or scene recognition. One of the main challenges lies in the fact that the classes to be predicted are not physically present within any region of the movie frames. Genres are intangible immaterial features that cannot be identified within a frame or even within a sequence of movie frames like an object can. Furthermore, movie trailers have a much more diverse content than video-based applications like the analysis of surveillance cameras, and movie trailers are much longer than typical clips found in action recognition datasets such as UCF [9]. Finally, movie genre classification is a multi-label problem, i.e., each movie may be labeled as belonging to multiple genres at the same time[2]. Multi-label classification has been considerably addressed in structured data scenarios [12, 13, 14], though there is a gap in the literature regarding work that deal with image and video analysis problems under the perspective of multi-label classification, specially considering the more recent deep learning studies.

In order to address the above-mentioned issues, we propose a novel neural architecture that encapsulates an ultra-deep residual ConvNet that comprises a special type of convolutional layer hereafter known as *convolution-through-time* (CTT) module, which is positioned in the top of the deep architecture. The CTT module allows the mapping of the features extracted from a sequence of frames into intangible genres. More importantly, the CTT module is directly plugged within the top of the ConvNet, allowing gradients to be propagated into the entire network, making the whole system a single end-to-end architecture.

This paper is organized as follows. Section 2 presents related work in the field of movie genre classification. Section 3 describes our proposed approach

[2]For more details on multi-label classification, please refer to two comprehensive surveys on the subject: [10, 11]

in detail, whereas Sections 4 and 5 present the experimental analysis that was conducted for validating our hypotheses. Finally, we end this paper with our conclusions and suggestions for future work in Section 6.

2. Related Work

In this section we describe the related work for the movie genre classification task. To the best of our knowledge, the best methods for genre classification were designed exclusively for single-label classification. Nevertheless, in real world scenarios, a movie rarely belongs exclusively to a single genre.

Rasheed et al. [15] propose the extraction of low-level features to detect movie genres through the application of the mean-shift classification algorithm [16]. Such features are responsible for describing raw video elements, such as the average shot length, color variance, lighting key, and motion presence. One of the important elements for low-level feature extraction from movies is the *shot detection* algorithm. A scene boundary is found when the inter-frame similarity is low. Frame similarity is computed via histogram intersection in the HSV color space. A scene boundary is set at the local minima of the inter-frame similarity smoothed function. Each scene is then represented by a single static frame known as the *keyframe*, which is the central frame from the scene. *Color variance* is another low-level feature that seems to play an important role at movie genre classification. For instance, comedies often present a higher color variance than horror movies. To calculate such a feature one must convert the keyframes into the CIE *Luv* space. A covariance matrix is generated and its determinant represents the trailer's total color variance. The *lighting key* feature for a given frame is extracted from the HSV color space by computing the mean (μ) and standard deviation (σ) of the pixel values. A frame with high-key lighting is a consequence of high μ and σ values. Conversely, a low-key frame is a consequence of low values from both μ and σ. The *motion content* feature represents the amount of action in a movie, i.e., the number of active pixels with time. For calculating such a feature, one must approximate the

partial derivatives of the frames in both the spatial and temporal dimensions. When motion is global (e.g. due to camera movement), all pixels tend to move to the same direction. Local motion causes pixel values to move to different regions. To perform this analysis, the directions of the pixels' gray levels are summarized into a 7-bin histogram. The majority of pixels are static and hence always fall into the first bin, whereas the remaining non-static pixels are defined as active. The overall motion of a scene is the ratio of active pixels per total amount of pixels.

Huang et al. [17] propose a rule-based movie genre classifier using only two features: scene transition visual effect and lighting key. The method was evaluated on a 44-trailer test set. However, the experimental methodology is not clear enough for allowing results to be reproduced.

Jain [18] claims to make use of a neural network with 21 inputs (7 low-level visual features and 14 audio features) for classifying single-label movie trailers into 5 movie genres, namely action, comedy, horror, drama and music. Since no hidden layers were used, what the author is actually training is a variant of logistic regression. The model is evaluated on a test set with 16 movies, which is surely not enough for proper validation of hypotheses.

Another approach for movie genre classification makes use of well-known image descriptors to compute high-level features for each keyframe. The work of Zhou et al. [19] employ the image descriptors Gist [20], CENTRIST [21], and w-CENTRIST to extract high-level features from frames and then perform movie genre classification via the k-NN algorithm. The Gist descriptor tries to encode semantic information like naturalness, openness, roughness, expansion, and ruggedness that represent the dominant spatial structure of a scene [20]. CENTRIST [21] is an image descriptor that applies a spatial pyramid at different levels, breaking the image into smaller patches. This process enables the detection of both local and global information. Finally, w-CENTRIST [19] modifies CENTRIST by taking into account colour information, neither present in Gist nor in CENTRIST. A global multi-dimensional histogram is then built for each trailer using a bag-of-features, where each dimension encodes a part

of the trailer. In its final step, each trailer in the test set is processed by the k-NN algorithm that computes its neighbours according to the χ^2 histogram similarity measure.

Huang and Wang [22] propose a hybrid approach that combines both low-level visual features and audio information, resulting in a total of 277 features. They make use of the well-known *jAudio* tool [23] to extract audio features such as audio intensity (measured in terms of the the RMS amplitude), timbre (based on different structures of amplitude spectrum), and rhythm. They extract more than 200 audio features via *jAudio*, including the well-known Mel-Frequency Cepstral Coefficients (MFCCs). During the classification process, they make use of the self-adaptive harmony search (SAHS) algorithm in order to search for the optimal subset of features for each of the one-vs-one SVMs that are designed to classify 223 movie trailers from the Apple website.

Simões et al. [24] propose ConvNet for extracting visual high-level features from movie frames. The ConvNet is trained at frame level in the LMTD-4 dataset. They use the extracted features to find scene clusters in order to build semantic histograms of the trailer scenes. Such histograms are concatenated with a pool of predictions and MFCC information, generating a novel representation for the entire movie trailer. Finally, they use an SVM to predict movie genres for each movie trailer. Similarly, Wehrmann et al. [25] propose the use of five neural networks (4 ConvNets and 1 MLP) to learn different aspects from the movie trailers. In their approach, 3 GoogleNets [1], 1 C3D [5], and 1 MLP form an ensemble of networks. The generated predictions from the networks are employed in different voting schemes at scene and movie-levels. An SVM is employed to perform the final genre classification by using several class predictions as features.

Both [25, 24] discussed in this section train or fine-tune ConvNets by using the single-label softmax loss function. Adapting both methods for a multi-label scenario is not trivial. For instance, using a multi-label loss function for training an entire ConNet may prevent the convergence of the approaches due to the

vanishing gradient problem, or may affect negatively the performance of the proposed methods.

3. CTT-MMC

In this paper, we propose to substantially extend the work of Wehrmann et al. [25] by developing an approach called **C**onvolution-**T**hrough-**T**ime for **M**ulti-label **M**ovie genre **C**lassification (CTT-MMC), which is a deep neural network architecture that is designed to take advantage of features from the movie trailer frames across time. More specifically, a CTT module is employed so convolutions can be used to learn spatio-temporal feature relationships within the entire movie trailer. The CTT module draws inspiration from the work of [26, 27, 28] in Natural Language Processing (NLP). In this work we use CTT for learning both image and audio information, in a two-stream based strategy [29].

3.1. CTT Module

The idea behind the CTT module is to use convolutional filters for learning temporal relations instead of using Recurrent Networks such as LSTMs. The main advantage of using CTT modules is that it is much faster and less error-prone than LSTMs. In addition, backpropagating gradients through a single convolutional layer is much easier than across recurrent temporal loopings that often cause saturation of the gradients.

Formally, let $X \in \mathbb{R}^{l \times m}$ be a given feature set that stores m-dimensional feature vectors for l timestamps. The first component within CTT is a convolutional layer with n filters of size $f \times m$, where f is the receptive field size and m the feature vector length. Common values for f used in ConvNets are $\{3, 5, 7\}$, but often using $f = 3$ leads to reasonable results. Convolutions are not zero-padded and we set the stride to 1, generating an output volume of $n \times 1 \times (l - f + 1)$. The next component within CTT is a max-pooling-over-time operation, which selects the most relevant features across time. Such an

operation is performed by using filters of size $(l-f+1)\times 1$, resulting in a volume $\mathbf{x} \in \mathbb{R}^{n\times 1\times 1}$, which can be used as a novel feature representation. Finally, let $\psi(X)$ be the computation performed by the convolutional and pooling layers in the CTT module, then $\psi(X) = \mathbf{x}$. Figure 1 shows how the CTT module works.

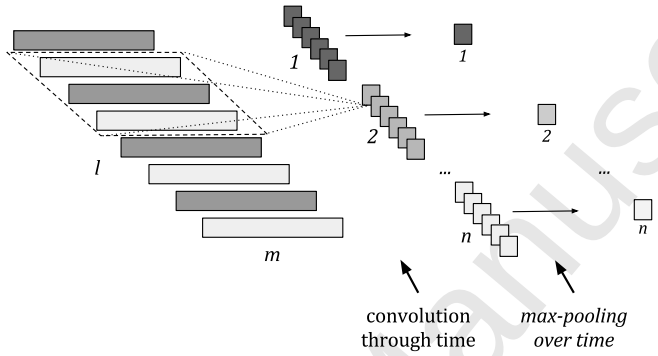


Figure 1: CTT module. Dashed lines represent a convolutional filter being applied over the temporal dimension in a movie trailer with l frames. Here, the filter size is set to $3 \times m$, since it convolves three adjacent frames. Each frame is represented by $m = 2048$ features. After the max-pooling over time operator, the resulting trailer representation is a $\mathbb{R}^n$ vector.

3.2. Learning Frame-based Information

Understanding the content of frames and their temporal relationships is fundamental for movie genre classification. Note that related work based on low-level video features are not capable of semantically recognizing the content of a movie. Our proposed approach takes into account high-level frame-based information extracted by deep neural networks. The high-level movie trailer features are learnt by an ultra-deep 152-layer ConvNet with residual connections [2]. It is pre-trained in both ImageNet [30] and Places365 [31]. ImageNet is the well-known image dataset that comprises 14 million images divided in 21,000 classes, being widely used for CV tasks such as object classification and detection. The residual network is pre-trained over 1.2 million images from the ILSVRC 2012 subset of ImageNet. In addition, it was also pre-trained over Places365, which is a scene-centric dataset that contains roughly 1.8 million

images from 365 classes. Hence, our ultra-deep ConvNet is suited to learn features regarding both objects and environmental aspects.

Formally, a movie trailer $\mathcal{T} \supset \{T_1, T_2, ..., T_t\}$ consists of t keyframes of dimension $w \times h \times c$, where w is the width, h is the height and c is the number of channels. Let $\phi(T_i) = \mathbf{k}_i$ denote the forward pass of the i^{th} keyframe T_i through the pre-trained ultra-deep ConvNet, generating feature vector $\mathbf{k}_i$ with 2048 features that represent the respective scene. CTT-MMC generates a temporal representation based on the visual aspects of the entire movie trailer, V, by stacking all $\mathbf{k}_i$ vertically, i.e., $V \in \mathbb{R}^{t \times 2048}$. This novel temporal representation is then fed to the CTT module so it can extract temporal relationships among trailer scenes.

The post-convolutional visual movie trailer representation is given by $\mathbf{v}$. Although such representation can be mapped directly to the classes, we also investigate the use of an additional fully-connected (FC) layer and a Maxout [32] activation layer. Hence, the prediction vector for all genres is calculated by $\zeta(\mathbf{v})$, where ζ is a (non-) linear operation over $\mathbf{v}$. The resulting prediction scores undergo the logistic sigmoid activation, i.e., $\mathcal{P} = \sigma(\zeta(\mathbf{v}))$, so they are turned into a probability vector $\mathcal{P} \in \mathbb{R}^C$, where C is the number of genres.

We experiment over three main variations of architecture within CTT-MMC for image-based learning, namely:

- CTT-MMC-A: it uses a single FC layer that linearly maps features to the classes;
- CTT-MMC-B: it uses two FC layers at the end to increase the non-linearity of the model;
- CTT-MMC-C: it uses a Maxout layer before the class prediction. A Maxout layer [32] is able to approximate any activation function, and performs (non-)linear classification by using two distinct weight matrices: Θ_1 and Θ_2. The activation of the Maxout layer is given by the larger of the inner products: $max(\Theta_1^T\mathbf{x}, \Theta_2^T\mathbf{x})$.

Figure 2 shows the training behaviour of the three variations of CTT-MMC with default hyper-parameters (as described in Section 4). All models of this section are optimized in the LMTD-9 training set and evaluated in its validation set. CTT-MMC-A seems to show signs of underfitting, whereas CTT-MMC-B seems to provide a more stable training. When analyzing the $AU(\overline{PRC})$ performance of all variations, we can see that CTT-MMC-C presents the best overall results in validation data.

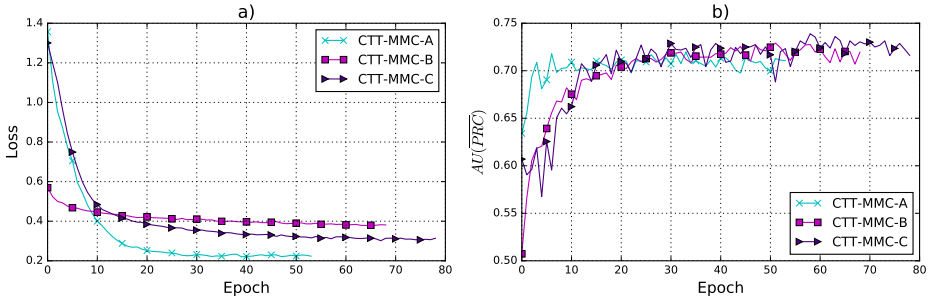


Figure 2: Training behaviour of three variations of CTT-MMC. Values are of $AU(\overline{PRC})$. a) training loss across epochs; and b) validation performance (average area under the precision-recall curve) across epochs.

In a nutshell, for learning frame-based information, the CTT module in CTT-MMC convolves the frame-based features across time, allowing for temporal relationships to be naturally learnt by the network. After convolving these features, a max-pooling-over-time operation is employed to extract the most representative features from the trailer. Such an operation results in a feature vector that encodes information from the whole movie trailer, enabling the proper mapping from trailer to genre.

Figure 3 shows the architecture of CTT-MMC. Note that the architecture is totally connected and allows for end-to-end learning, i.e., the gradients can be backpropagated through the entire scheme, though in this paper, we do not modify the weights of the pre-trained ultra-deep residual ConvNet. We do not update the ConvNet weights due to two reasons: i) to save both training time and computational resources; and ii) to avoid ruining the previously trained

feature extractor.

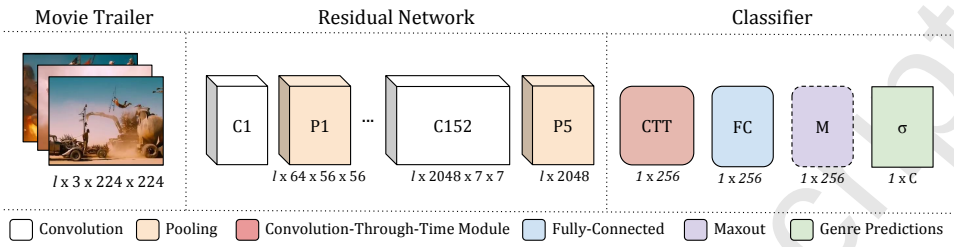


Figure 3: CTT-MMC architecture.

For training the models, CTT-MMC optimizes a multi-label loss function – the binary cross-entropy for multiple classes, given by

$$\mathcal{L}(\hat{y}, y) = -\sum_{i=1}^{C} [y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)] \quad (1)$$

where, C is the number of classes (i.e, number of movie genres), y_i is the actual class, and $\hat{y}_i$ is the probability predicted by CTT-MMC that the movie trailer belongs to the i^{th} class. Note that y is a binary vector of length C and $\hat{y} \in \mathbb{R}^C$.

3.3. Learning Audio-based Features

Audio elements such as music and sound effects are often used to build the movie's atmosphere. For instance, movie directors often use upbeat music for comedies and lower tones for characterizing thrilling and tense situations. Therefore, we argue that audio information is particularly meaningful for movie genre classification.

Following recent work in the area of audio classification and speech recognition [33, 34], we explore the use of spectrograms for learning audio information. For that, we also employ a CTT module for learning movie genre information from the spectrograms. We first convert fixed non-overlapping 3-second audio signals of $44100hz$ into spectrograms, which are time-frequency representations of the audio signals. Each audio signal is now represented by 130×513 matrices, where 130 denotes the temporal dimension, and 513 the frequency domain. The intensity values represent the amplitude of the signals.

We project the spectrograms in the RGB space for better visualization. In addition, we calculate the average spectrogram for normalizing instances during training. The reason for choosing a 3-second window is because it generates features with an acceptable dimensionality for feeding the CTT module – larger audio snippets would lead to an undesirably-large dimensional space, while smaller snippets would lead to loss of information.

Note that audio from a given movie trailer is represented by a sequences of fixed-length time-frequency matrices. Thus, let $\mathcal{S} \supset \{S_1, S_2, S_3, ..., S_s\}$ be the normalized spectrogram sequence, and $S_j \in \mathbb{I}^{130\times513}$ be the j^{th} spectrogram matrix in the sequence. It is reasonable to assume that CTT is well-suited for learning from such matrices, given its capability of understanding temporal relations.

For training audio-based models we draw inspiration from [35, 25] that perform segment-based learning. We first instantiate a CTT module, denoted by ψ_A, with the following parameters: $\{n = 256, f = 3, l = 130, m = 513\}$. This instance is suited for extracting information from fixed-length spectrograms. Movie trailers contain an average of 50 3-second audio clips. For providing a better representation of the entire audio, we split $\mathcal{S}$ it into p parts and randomly sample one spectrogram per part. This strategy allows backpropagating stronger and better gradient signals to optimize weights responsible for *reading* spectrograms regardless of its temporal position.

Outputs of ψ_A for the p parts are then concatenated and average-pooled with filters of size $p \times 1$ for generating a consensual representation of the entire audio signal. Since pooling operations are totally differentiable, it is easy to backpropagate gradients generated by each part p for providing a conjoint weight update for ψ_A. The average pooling also helps in regularizing the model even when using large values of p, i.e., $p > 7$. With this particular strategy, the count of trainable parameters in the audio-based models is not affected by the choice of p.

Figure 4 depicts how CTT was used for learning audio-based movie trailer information. Note that, in this particular model, we discovered that it is very

important to use a batch normalization layer [36] before the CTT-module, since it helps with regularization and in keeping activation values within a healthy range across batches.

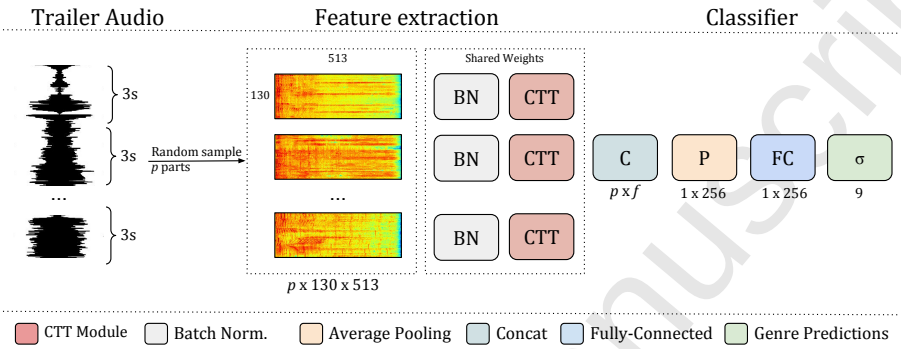


Figure 4: CTT-MMC architecture for audio-based learning, namely CTT-MMC-S. A CTT module is responsible for learning audio information from normalized spectrograms randomly sampled from p parts of a movie trailer. Features are then averaged for a consensual representation of the entire audio signal.

For providing some reassurance regarding the impact of hyper-parameter p, we trained models with $p \in \{3, 5, 7, 9, 11\}$. For monitoring the training phase, we use only one prediction from each instance with p segments. Figure 5 depicts the training behavior for each configuration. Note that using $p \in \{3, 5\}$ results in a more challenging optimization. In these cases, values of training loss are consistently higher than for training with more segments per instance, e.g, $p \in \{7, 9, 11\}$. Indeed, larger p yields to more stable results across training.

For generating the final predictions we vary the number of samples used per instance in order to marginalize the error. Table 1 depicts validation results when using different p segments and varying the number of instances sampled for testing. We found that in 4 out of 5 cases, using 10 samples leads to better results. Moreover, the best value that was found is $p = 11$ and 10 samples, which will be used in the test set experimental analysis.

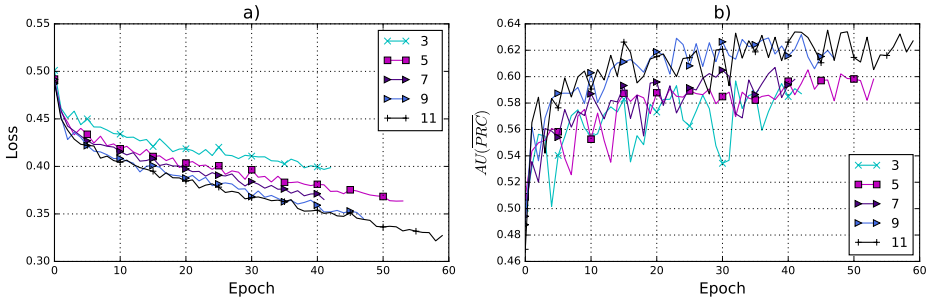


Figure 5: Training behaviour of audio-based models varying number of parts p used as input. a) training loss across epochs; and b) validation performance (average area under the precision-recall curve) across epochs.

Table 1: Impact of p for training and number of samples for testing. Values are of $AU(\overline{PRC})$ in validation data.

Value of segments p	5 samples	10 samples
3	0.635	0.648
5	0.644	0.651
7	0.639	0.651
9	0.655	0.649
11	**0.659**	**0.663**

3.4. Two-Nets

We train the frame and audio-based networks independently. The final genre prediction for a given movie trailer is generated by a fusion of predictions from both networks. Such a fusion is performed through a simple weighted average where the visual network importance is empirically set to 60% and the audio network to 40%. Such a definition has been set by analyzing validation results varying the following ratios: {(1.0, 0.0), (0.95, 0.05), (0.90, 0.10), ..., (0.0, 1.0)}.

4. Experimental Setup

In this section, we detail the dataset that is used in our experiments (Section 4.1), as well as the baseline algorithms (Section 4.2) that we use to compare with CTT-MMC, the evaluation criteria that we employ to measure

predictive performance (Section 4.3), and the hyper-parameter settings for CTT-MMC (Section 4.4).

4.1. Dataset

For validating the performance of CTT-MMC, we performed several experiments over the recently-introduced movie trailer dataset LMTD (Labelled Movie Trailer Dataset) [25, 24]. As far as we know, LMTD is the largest movie trailer dataset publicly made available. Its full extend comprises about 10,000 movie trailers from 22 genres, which were assigned according to the IMDB meta-data. In addition, LMTD comprises roughly 400 hours of video and 30 million frames. For comparison purposes, note that the action recognition dataset UCF-101 [9] comprises around 27 hours of video (it is 15 times smaller than LMTD).

The LMTD movie trailers have the following characteristics: i) high variability of video features, such as aspect ratio, image quality, and total length; ii) a wide range of movie release years (e.g, 1920 to 2016); and iii) an unbalanced class distribution that is representative of the real-world (drama and comedy movies are much more common than documentaries or musicals).

We make use of LMTD's novel multi-label subset, henceforth called LMTD-9 (Table 2). As the name suggests, it comprises movie trailers from 9 movie genres, namely action, adventure, comedy, crime, drama, horror, romance, sci-fi, and thriller. These genres were selected as a consequence of limiting each class to have at least 10% of the total number of instances in LMTD. In LMTD-9, each instance is assigned to at least 1 and at the most 3 genres. In addition, to mitigate problems with outliers, LMTD-9 discards movie trailers with more than 6500 frames and with a release year older than 1980. It is randomly divided in training, validation and test sets as presented in Table 2. LMTD-9 is the largest subset for multi-label movie genre classification to date. It comprises 4007 movie trailers, which is roughly half of all movie trailers in LMTD. The dataset used in [22], in turn, comprises only 223 trailers, being 18 times smaller than LMTD-9.

Table 2: LMTD-9 dataset.

Genre	Training	Validation	Test
Action	611	78	164
Adventure	432	51	108
Comedy	1109	148	301
Crime	477	59	121
Drama	1437	192	394
Horror	324	33	78
Romance	468	59	122
SciFi	229	26	57
Thriller	502	61	129
1 genre	884 (30.90%)	124 (33.15%)	251 (32.47%)
2 genres	1226 (42.85%)	167 (44.65%)	340 (43.98%)
3 genres	751 (26.25%)	83 (22.20%)	181 (23.46%)
Label cardinality	1.95	1.89	1.91
Label density	0.22	0.21	0.21
Total Movies	2861	374	772

4.2. Baseline Algorithms

To the best of our knowledge, this is the first work to analyze the movie genre classification problem under the perspective of multi-label classification. Hence, we compare CTT-MMC to the state-of-the-art approaches for single-label classification, namely video low-level features (VLLF) [15] and audio-visual features (AV) [22]. Note that those methods are not naturally suited for multi-label classification. Hence, for the sake of fairness, we employ the features described in each baseline in a one-vs-all SVM scheme. Considering that SVMs are not probabilistic models, we normalize the scores transforming them into probabilities following the method proposed in [37]. All baseline hyper-parameters were optimized by grid-searching: $c \in \{0.1, 1, 10, 100, 1000\}$, kernel $\in \{linear, gaussian\}$, and $\gamma \in \{0.1, 0.2, 0.3, ..., 1.0\}$.

For a stronger baseline, we include in our analysis an LSTM, which is a standard algorithm for sequence analysis. The LSTM is trained using the same data that feeds the CTT-based models. We limit the temporal iterations in 120 steps, cutting the final steps or zero-padding when the video is shorter than that.

This strategy allows training the LSTMs with mini-batches, which accelerates the training phase. The LSTM comprises 256 hidden units and a dropout of 50% in the final layers. It contains a similar amount of parameters than the CTT-based models, i.e., whereas the Maxout network (CTT-MMC-C) contains $1.8M$ parameters, the LSTM contains $2.3M$ parameters.

We also provide as a baseline the results that would be expected from random classification. Such an evaluation is important due to the existence of unbalanced classes. To generate that baseline, we sample random probabilities for the test set from a uniform distribution. These random probabilities are then used to compute all evaluation measures described in Section 4.3.

4.3. Evaluation Measures

The outputs of CTT-MMC for each class are probability values, and the same is true for the baseline algorithms. We follow the trend of multi-label classification research in which we avoid choosing thresholds by employing precision-recall curves (PR-curves) as the evaluation criterion for comparing the different approaches. For generating a PR-curve for a given classification method, one must select a predefined number of thresholds within [0, 1] to be applied over the outputs of each method, finally generating several precision and recall points in the PR plane. The interpolation of these points [38] constitute a PR-curve, and the quantitative criterion one analyzes is the area under such a curve ($AU(\overline{PRC})$).

We employ the following derived measures: $AU(\overline{PRC})$ (micro average), $\overline{AU(PRC)}$ (macro average), and $AU(\overline{PRC})_w$ (weighted average). Each of these measures point to different aspects regarding each method's performance. For instance, $\overline{AU(PRC)}$ measure is calculated by averaging the areas of all classes, which causes less-frequent classes to have more influence in the results. $AU(\overline{PRC})$ is calculated by using all labels globally, providing information regarding the entire dataset, which makes high-frequency classes to have greater influence in the results. Finally, $AU(\overline{PRC})_w$ is calculated by averaging the area under precision-recall curve per genre, weighting instances according to the class

frequencies.

In addition, we also employ the Ranking Loss, which is an evaluation measure that expresses the number of times that irrelevant labels are ranked higher than relevant labels [39].

4.4. Hyper-Parameters Settings

For training CTT-MMC, we chosen our algorithms and hyper-parameters based on [40]. We use Stochastic Gradient Descent (SGD) with mini-batches of 64 instances, 256 neurons in the FC layers, learning rate of 1×10^{-3}, weight decay regularization $\gamma = 1 \times 10^{-4}$, dropout of 0.5 and the Adam rule for parameter update. In the test phase, we use the 10-crop strategy described in [1]. In order to provide a fair comparison with the baselines, we have not optimized the hyper-parameters nor performed any kind of feature selection. We train the CTT module for a maximum of 200 epochs, early-stopping when the predictive performance in the validation set does not improve for 20 consecutive epochs. In average, our models reach convergence within 70 epochs.

5. Results and Discussion

In this section, we present the experimental results in the test set comparing the predictive performance of the following algorithms: CTT-MMC-(A/B/C), CTT-MMC with spectrograms only (CTT-MMC-S) for audio-based evaluation, two-nets (CTT-MMC-TN), VLLF [15], AV [22], LSTM, and random classification. Recall that CTT-MMC-A maps directly the features extracted by CTT to the classes, while CTT-MMC-B has one fully-connected layer before the classes to improve non-linearity, and CTT-MMC-C has a Maxout hidden layer, which is capable of approximating any activation function. Finally, CTT-MMC-S is trained with spectrograms only, for evaluating the capability of an audio-based network.

5.1. Quantitative Analysis

In Table 3, we report the predictive performance of all methods in the test set. Note that all variations of CTT-MMC outperform all baselines by a large margin (values in bold). Our best approach, namely CTT-MMC-TN, in terms of $\overline{AU(PRC)}$ performs around 24% better than the LSTM, which is the strongest baseline. When compared to AV (the best approach based on low-level features), CTT-MMC-TN is 42% better. This difference is almost twice as large than the improvement provided by AV when compared do VLLF.

Table 3: Results comparing CTT-MMC with the state-of-the-art methods and with random classification.

Method	$\overline{AU(PRC)}$	$AU(\overline{PRC})$	$AU(\overline{PRC})_w$	Ranking Loss
Random	0.206	0.204	0.294	0.466
VLLF	0.278	0.476	0.386	0.253
AV	0.455	0.599	0.567	0.159
LSTM	0.520	0.640	0.590	0.150
CTT-MMC-S	0.485	0.642	0.599	0.183
CTT-MMC-A	0.618	0.712	0.683	0.109
CTT-MMC-B	0.599	0.704	0.680	0.121
CTT-MMC-C	0.624	0.722	0.697	0.108
CTT-MMC-TN	**0.646**	**0.742**	**0.724**	**0.099**

The values of $\overline{AU(PRC)}$ show that CTT-MMC presents a solid performance even for trailers of rare classes (e.g., sci-fi and horror). In this particular scenario, the VLLF method presents virtually random results (0.278 vs 0.206) and AV's performance decreases in 32% when compared to the $AU(\overline{PRC})$ values. CTT-MMC 's decrease in performance is about of 16%, which shows it is much more consistent when predicting rare genres. We believe that CTT-MMC's ability of learning semantic features helps it to better discriminate those genres.

By analyzing the values of Ranking Loss, we reach to the same conclusion: CTT-MMC-TN is the preferred method. Overall, methods based on deep features have shown to perform much better than low-level based approaches. Note that CTT-based architectures trained over frame-level features present the best performance across all evaluation measures.

Recall that $AU(\overline{PRC})_w$ values give greater weight for high-frequency classes (e.g, drama and comedy). This fact explains the improvement in performance of the random classifier (0.294). In this scenario, CTT-MMC outperforms VLLF by 87% and AV by 28%. Also, we observe that the difference in $AU(\overline{PRC})_w$ between CTT-MMC and VLLF is proportional to the difference between AV and random classification.

Table 4 shows values of AUPRC for all genres. For short, within our methods we selected the best method for image and audio-based learning, as well as CTT-MMC-TN, which is a combination of both. We also compare the two strongest baselines and the random classifier. We highlight that the Spectrogram network's performance (CTT-MMC-S) is quite competitive with the LSTM.

Table 4: Per-genre AUPRC results of CTT-MMC, the state-of-the-art methods, and random classification.

	CTT-MMC-C	CTT-MMC-S	CTT-MMC-TN	LSTM	AV	Random
Action	0.813	0.669	**0.835**	0.687	0.601	0.158
Adventure	**0.720**	0.397	0.672	0.573	0.461	0.131
Comedy	0.853	0.834	**0.870**	0.792	0.838	0.512
Crime	0.505	0.339	**0.547**	0.421	0.215	0.140
Drama	0.791	0.764	**0.841**	0.740	0.792	0.435
Horror	0.609	0.390	**0.667**	0.478	0.423	0.088
Romance	0.432	0.362	**0.456**	0.313	0.256	0.129
SciFi	0.398	0.226	**0.401**	0.237	0.161	0.063
Thriller	0.496	0.386	**0.522**	0.437	0.308	0.196

We present in Figure 6 the precision-recall curves generated for all genres regarding the trailers from the test set. We compare the curves among all methods in order to provide some intuition about each method's learning capabilities. We notice that the features extracted by our deep learning architecture provide large improvement in predictive performance for all movie genres, though we highlight the more *subjective* ones. For instance, the *crime* genre is considerably more subjective than *action* or *sci-fi*, and neither visual low-level features nor audio features were helpful for classifying such a genre. In

this case, both VLLF and AV presented nearly-random results. CTT-MMC, on the other hand, was capable of correctly predicting most of the *crime*, *sci-fi* and *romance* trailers. We believe that CTT-MMC's ability on finding objects within the trailer scenes is greatly responsible for achieving such good results in these more subjective genres. Furthermore, the good results when classifying *adventure* and *action* genres are probably due to the Places-based learnt features.

Horror is the only movie genre for which CTT-MMC-TN presents a large improvement when compared to CTT-MMC, demonstrating the importance of audio for learning features of this particular genre.

We present a summary of our findings, as follows:

- Deep visual features provide large improvements when compared to low-level baselines. Such features allows better understanding the content of movies, which explains the performance of LSTM and CTT-based methods for classifying genres such as *adventure*, *crime*, *romance*, and *thriller*.

- All CTT-based methods outperform the LSTM in all genres, albeit being lighter (fewer learnable parameters) and faster. This is quite surprising since LSTMs are traditional networks for understanding temporal relationships, being the current state-of-the-art for several sequence-learning tasks.

- CTT audio-based models outperformed all low-level baselines and also proved to be competitive with the LSTM.

- Audio information seems to significantly improve classification performance for the horror genre when jointly used with frame-based models.

5.2. Qualitative Analysis

Figure 7 depicts predictions generated by CTT-MMC-TN. It correctly predicted all genres for several movies, including *Fast & Furious*, *Independence*

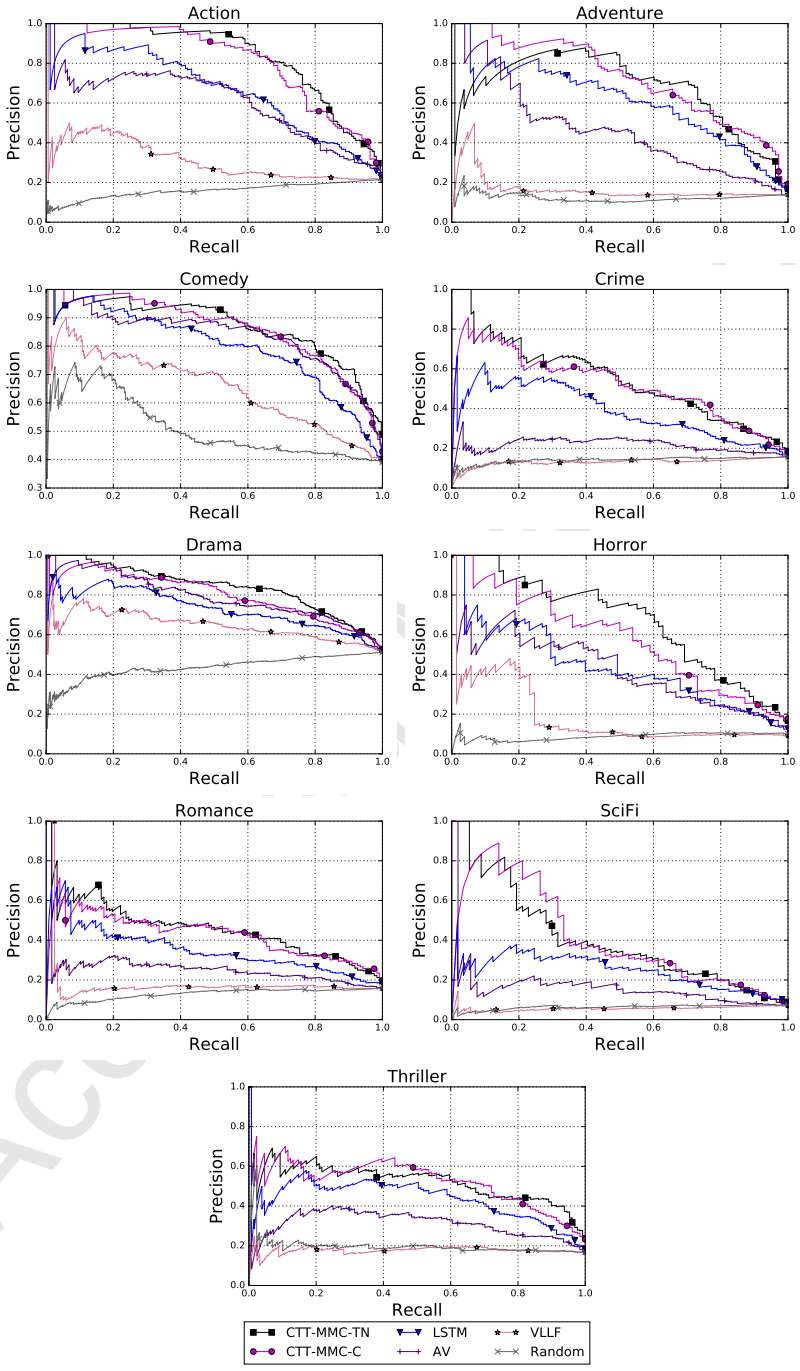


Figure 6: Precision-recall curves for all movie genres.

Day, *Rush Hour*, and *Sicario*. These predictions include more subjective and complex genres such as *crime*, *comedy* and *thriller*. Predictions of *Tammy* show that CTT-MMC-TN is also capable of identifying the correct number of genres. Nonetheless, CTT-MMC-TN also makes some mistakes, as it can be seen in the predictions for the following movies: (1) *Phantom of the Opera*, where it recognizes the *Thriller* genre; and (2) *The Perfect Man*, which depicts drama as the second most probable genre, whereas the ground truth contains only *Comedy* and *Romance*.

6. Conclusion

This paper proposed CTT-MMC, a novel method for multi-label movie genre classification. It comprises an ultra-deep neural network with residual connections that was pre-trained in two large datasets, and also a convolution-through-time module that makes use of a convolutional layer and of a max-pooling-over-time layer for encoding temporal information. We have shown that CTT-MMC comfortably outperforms the current approaches for all genres in a large movie trailer dataset, establishing itself as the novel state-of-the-art for multi-label movie genre classification.

Through an extensive empirical analysis, we have shown that deep visual features provide large improvements when compared to low-level baselines, helping to better describe the content of movies from difficult genres such as *adventure*, *crime*, *romance*, and *thriller*. We have also shown that our method outperforms an LSTM in all movie genres while being lighter (fewer learnable parameters) and faster. Finally, we have shown that audio-based versions of our method are also competitive with LSTMs, significantly improving the classification performance of the horror genre when jointly used with frame-based models.

The main limitation of this work relies in the use of only nine common movie genres. For future work, we intent to continue improving the LMTD dataset by including new trailers and providing a version with rare genres as

Figure 7: Genre predictions by CTT-MMC-TN for six movies from the test set. Horizontal bars indicate the per-class predicted probabilities.

well. In addition, we plan to integrate motion information in CTT-MMC but guaranteeing that it remains an end-to-end learning architecture.

7. Acknowledgments

We would like to thank the Brazilian research agencies CAPES, CNPq, and FAPERGS for funding this work. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research. We also thank Google and Motorola for financially supporting both authors.

References

[1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, arXiv preprint arXiv:1409.4842.

[2] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, arXiv preprint arXiv:1512.03385.

[3] S. Ji, W. Xu, M. Yang, K. Yu, 3d convolutional neural networks for human action recognition, Pattern Analysis and Machine Intelligence, IEEE Transactions on 35 (1) (2013) 221–231.

[4] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 1725–1732.

[5] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: 2015 IEEE International Conference on Computer Vision (ICCV), IEEE, 2015, pp. 4489–4497.

[6] J. Wehrmann, G. S. Simões, R. C. Barros, V. F. Cavalcante, Adult content detection in videos with convolutional and recurrent neural networks, Neurocomputing in press. doi:http://dx.doi.org/10.1016/j.neucom.2017.07.012.

[7] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[8] I. J. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press, 2016.

[9] K. Soomro, A. R. Zamir, M. Shah, Ucf101: A dataset of 101 human actions classes from videos in the wild, arXiv preprint arXiv:1212.0402.

[10] M.-L. Zhang, Z.-H. Zhou, A review on multi-label learning algorithms, IEEE transactions on knowledge and data engineering 26 (8) (2014) 1819–1837.

[11] G. Tsoumakas, I. Katakis, Multi-label classification: An overview, International Journal of Data Warehousing and Mining 3 (3).

[12] R. Cerri, R. C. Barros, A. C. P. L. F. de Carvalho, Hierarchical classification of gene ontology-based protein functions with neural networks, in: 2015 International Joint Conference on Neural Networks (IJCNN), 2015, pp. 1–8. doi:10.1109/IJCNN.2015.7280474.

[13] R. Cerri, R. C. Barros, A. C. P. L. F. de Carvalho, Y. Jin, Reduction Strategies for Hierarchical Multi-Label Classification in Protein Function Prediction, BMC Bioinformatics 17 (2016) 373–389.

[14] J. Wehrmann, R. C. Barros, R. Cerri, Hierarchical multi-label classification with chained neural networks, in: ACM Symposium on Applied Computing, ACM, 2017.

[15] Z. Rasheed, Y. Sheikh, M. Shah, On the use of computable features for film classification, Circuits and Systems for Video Technology, IEEE Transactions on 15 (1) (2005) 52–64.

[16] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619.

[17] Huang, Hui-yu and Shih, Weir-sheng and Hsu, Wen-hsing, A film classifier based on low-level visual features, Journal of Multimedia 3 (2008) 465–468.

[18] S. K. Jain, Movies Genres Classifier using Neural Network, in: Proceedings of the International Symposium on Computer and Information Sciences, 2009, pp. 610–615.

[19] H. Zhou, T. Hermans, A. V. Karandikar, J. M. Rehg, Movie genre classification via scene categorization, in: Proceedings of the international conference on Multimedia, ACM, 2010, pp. 747–750.

[20] A. Oliva, A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, International journal of computer vision 42 (3) (2001) 145–175.

[21] J. Wu, J. M. Rehg, Where am i: Place instance and category recognition using spatial pact, in: CVPR, 2008, pp. 1–8.

[22] Y.-F. Huang, S.-H. Wang, Movie genre classification using svm with audio and video features., in: R. Huang, A. A. Ghorbani, G. Pasi, T. Yamaguchi, N. Y. Yen, B. Jin (Eds.), AMT, Vol. 7669 of Lecture Notes in Computer Science, Springer, 2012, pp. 1–10.

[23] D. McEnnis, C. McKay, I. Fujinaga, P. Depalle, jaudio: An feature extraction library., in: ISMIR, 2005.

[24] G. Simões, J. Wehrmann, R. C. Barros, D. D. Ruiz, Movie genre classification with convolutional neural networks, in: International Joint Conference on Neural Networks, IEEE, 2016, p. 8.

[25] J. Wehrmann, R. C. Barros, G. Simões, T. S. Paula, D. D. Ruiz, (Deep) Learning from Frames, in: Brazilian Conference on Intelligent Systems, 2016, p. 6.

[26] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882.

[27] J. Wehrmann, W. Becker, H. E. Cagnini, R. C. Barros, A character-based convolutional neural network for language-agnostic twitter sentiment analysis, in: Neural Networks (IJCNN), 2017 International Joint Conference on, IEEE, 2017, pp. 2384–2391.

[28] J. Wehrmann, A. Mattjie, R. C. Barros, Order embeddings and character-level convolutions for multimodal alignment, CoRR abs/1706.00999.

[29] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in neural information processing systems, 2014, pp. 568–576.

[30] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[31] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, Learning deep features for scene recognition using places database, in: Advances in neural information processing systems, 2014, pp. 487–495.

[32] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, Y. Bengio, Maxout networks., ICML (3) 28 (2013) 1319–1327.

[33] H. Lee, P. Pham, Y. Largman, A. Y. Ng, Unsupervised feature learning for audio classification using convolutional deep belief networks, in: Advances in neural information processing systems, 2009, pp. 1096–1104.

[34] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, D. Yu, Convolutional neural networks for speech recognition, IEEE/ACM Transactions on audio, speech, and language processing 22 (10) (2014) 1533–1545.

[35] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks: towards good practices for deep action recognition, in: European Conference on Computer Vision, Springer, 2016, pp. 20–36.

[36] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167.

[37] T.-F. Wu, C.-J. Lin, R. C. Weng, Probability estimates for multi-class classification by pairwise coupling, Journal of Machine Learning Research 5 (Aug) (2004) 975–1005.

[38] J. Davis, M. Goadrich, The relationship between precision-recall and roc curves, in: ICML, 2006.

[39] G. Tsoumakas, I. Katakis, I. Vlahavas, Mining multi-label data, in: Data mining and knowledge discovery handbook, Springer, 2009, pp. 667–685.

[40] D. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.