Movie Genre Classification with Convolutional Neural Networks

Gabriel S. Simões, Jônatas Wehrmann, Rodrigo C. Barros, Duncan D. Ruiz
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, Rio Grande do Sul, Brazil
Emails: {gabriel.simoes.001, jonatas.wehrmann}@acad.pucrs.br, {rodrigo.barros, duncan.ruiz}@pucrs.br

***Abstract*—Automatic pattern recognition from videos is a high-complexity task, and well-established Machine Learning algorithms have difficulties in handling it in an efficient and effective fashion. Convolutional Neural Networks are the state-of-the-art method for supervised image classification, borrowing concepts from image processing in order to ensure some degree of scale and position invariance. They are capable of detecting primary features, which are then combined by subsequent layers of the CNN architecture, resulting in the detection of higher-order complex and relevant novel features. Considering that a video is a set of ordered images in time, we propose in this paper to explore CNNs in the context of movie trailers genre classification. Our contributions are twofold. First, we have developed a novel movie trailers dataset with more than 3500 trailers whose genres are known, and we make it publicly available for the interested reader. Second, we detail a novel classification method that encapsulates a CNN architecture to perform movie trailer genre classification, namely CNN-MoTion, and we compare it with state-of-the-art feature extraction techniques for movie classification such as Gist, CENTRIST, w-CENTRIST, and low-level feature extraction. Results show that our novel method significantly outperforms the current state-of-the-art approaches.**

Keywords*—*convolutional neural networks, video analysis, movie genre classification, machine learning

I. Introduction

Machine Learning (ML) is an area within computer science that is growing and evolving quickly. Probably most of the modern computer-based systems and applications make use of ML at some extent. The successful tasks performed by ML algorithms include a variety of applications such as handwritten digits recognition [1], autonomous driving [2], gene expression classification [3], [4], protein function prediction [5], [6], software metrics estimation [7]–[9], real-time stream sensor analysis [10], and that is just to name a few.

The task of automatically analyzing videos could help humans in solving several problems that are nowadays either too expensive or excessively tedious for them to perform alone. Whereas there are several efficient ML approaches that reach almost 94% of accuracy when classifying images as belonging to one within a thousand of labels [11], video-based applications have shown to be much more challenging. Such a task has a high complexity level, and well-established ML algorithms have difficulties in handling it in an effective and efficient fashion.

Automatic video analysis is a broad concept and offers many research possibilities, such as action recognition, categorization, element recognition, context analysis, and many other tasks. Recent work [12]–[14] address video analysis with Deep Convolutional Neural Networks (CNNs) [15], showing exciting first results and possibly paving the way for many applications to be further explored.

CNNs are the state-of-the-art method for supervised image classification, borrowing concepts from image processing to ensure some degree of scale, position, and distortion invariance. These ideas are based on the detection of primary features that may be located at any part of the image, such as oriented edges, end-points, and corners. These visual features are then combined by the subsequent layers in order to detect novel higher-order features, ultimately mimicking the human learning process.

Each stage of a CNN comprises one or more convolutional layers (often with a pooling step) that are followed by one or more fully-connected layers (as in a standard multi-layer neural network). The CNN architecture is designed to take advantage of the 2D structure of an input image (or other 2D input abstractions such as a two-channel speech signal).

In this paper, we propose a novel method supported by a CNN architecture to classify movie trailers according to their genres, namely CNN-MoTion (Convolutional Neural Networks for Movie Trailer Classification). Our contributions in this paper are as follows. First, we make publicly available a novel movie trailers dataset, which comprises more than 3500 trailers that belong to one of the following four genres: action, comedy, horror, or drama. Second, we present CNN-MoTion in detail, and we empirically demonstrate that it outperforms the state-of-art movie trailer classification techniques Gist [16], CENTRIST [17], w-CENTRIST [18], and low-level features [19].

This paper is organized as follows. Section II introduces the reader to the required background on video analysis through image-processing and machine learning techniques. Section III describes in detail our novel approach for movie trailers genre classification, whereas Section IV presents a thorough experimental analysis for validating our research hypotheses. Finally, we end this paper with our conclusions and suggestions for future work in Section V.

II. Video Analysis

The problem of automatically analyzing videos through image-processing and machine learning approaches has been a much-studied research theme. One of the many possible tasks within video analysis is automatic movie genre classification.

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

In this section, we present a background on the state-of-the-art approaches for solving such a task.

A. Learning from low-level features

Rasheed et al. [19] propose the extraction of low-level features to detect movie genres through the application of the mean-shift classification algorithm [20]. Such features are responsible for describing raw video elements, such as the average shot length, color variance, lighting key, and motion presence, which are computed as follows.

1) Shot detection: For detecting when a novel shot happens within a video, one must compare every frame to its adjacent neighbor. A scene boundary is found when the inter-frame similarity is low. Frame similarity is computed via histogram intersection in the HSV (Hue, Saturation, Value) color space. Each histogram comprises 16 bins: eight for hue, four for saturation, and four for value. Eq. (1) details the histogram intersection computation.

$$s(i) = \sum_{j \in allbins} min(H_i(j), H_{i-1}(j)) \tag{1}$$

Such an approach works well for detecting abrupt scene changes. However, the algorithm fails when soft transitions occur. For fixing that issue, we can iteratively smooth $s(i)$ with a Gaussian kernel, as proposed by [21] and adapted by [19]. This process is presented in Eq. (2) , where t is the iteration number, $\lambda = 0.1$ and $k = 0.1$.

$$S^{t+1}(i) = S^t(i) + \lambda[_{CE} \cdot \nabla_E S^t(i) +_{CW} \cdot \nabla_W S^t(i)] \tag{2}$$

$$\nabla_E S(i) \equiv S(i+1) - S(i) \tag{3}$$

$$\nabla_W S(i) \equiv S(i-1) - S(i) \tag{4}$$

$$C_E^t = g(|\nabla_E S^t(i)|) \tag{5}$$

$$C_W^t = g(|\nabla_W S^t(i)|) \tag{6}$$

$$g(x) = e^{-(\frac{x}{k})^2} \tag{7}$$

A scene boundary is set at the local minima of the inter-frame similarity smoothed function $s(i)$. Each scene is represented by a single static frame known as the *keyframe*, which is the central frame from the scene.

2) Color Variance: Color variance seems to play an important role at the 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. Then, a covariance matrix must be generated, as presented in Eq. (8).

$$p_{cov} = \begin{bmatrix} \sigma_L^2 & \sigma_{Lu}^2 & \sigma_{Lv}^2 \\ \sigma_{Lu}^2 & \sigma_u^2 & \sigma_{Luv}^2 \\ \sigma_{Lv}^2 & \sigma_{uv}^2 & \sigma_v^2 \end{bmatrix} \tag{8}$$

The determinant of p_{cov}, also known as the generalized variance, represents the movie trailer's total color variance.

3) Lighting Key: There are two main lighting categories: high-key lighting and low-key lighting. The first one concerns of abundant bright color levels and less contrast between dark and light. In the latter, usually darker tones are predominant and there is a high contrast ratio.

$$\zeta_i = \mu_i \cdot \sigma_i \tag{9}$$

The lighting key feature for frame i (ζ_i) is computed as shown in Eq. (9). It is better 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.

4) Motion Content: The motion content feature represents the action in a movie, i.e., the amount of active pixels with time. This analysis must be done for every scene/shot with all frames. A practical way to compute this feature is based on the structural tensor theory (Γ). Given the x, y spatial dimensions, and the temporal dimension t, one can compute Γ following Eq. (10).

$$\Gamma = \begin{bmatrix} J_{xx} & J_{xt} \\ J_{xt} & J_{tt} \end{bmatrix} = \begin{bmatrix} \sum_w H_x^2 & \sum_w H_x H_t \\ \sum_w H_x H_t & \sum_w H_t^2 \end{bmatrix} \tag{10}$$

where H_x and H_t are the partial derivatives of the frames in either the x spatial dimension or the t temporal dimension. To calculate the gradients we convolve the frames both spatially and temporally by means of the *Sobel* filter. To find the direction of the gray levels we need to compute θs for every pixel, as presented in Eq. (11).

$$\theta = \frac{1}{2} tan^{-1} \frac{2J_{xt}}{J_{xx} - J_{tt}} \tag{11}$$

When all θ values are constant, it means there is no motion on the scene. When motion is global (e.g. 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, we use the directions of the pixels' gray levels summarized into a 7-bin histogram. The majority of pixels are static and hence always fall into the first bin. 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.

B. Learning from high-level features

A second approach for movie genre classification makes use of well-known image descriptors to compute high-level features for each keyframe. These image descriptors form a trailer holistic representation, though they are not capable of capturing motion information that varies with time.

The work of Zhou et al. [18] make use of the image descriptors Gist [16], CENTRIST [17], 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 [16]. Census

Transform Histogram (CENTRIST) [17] is an image descriptor that produces good results for environmental classification. It is often employed to recognize places and scenes. To generate the CENTRIST image, one must apply a spatial pyramid at different levels, breaking the image into smaller patches. This process enables the detection of both local and global information. Each patch is processed through the Census Transform which compares the pixels with its neighbors. This step produces an 8-bit vector replacing the current pixel. Afterwards, it is appended to the final vector containing all values from the patches. Finally, w-CENTRIST [18] modifies CENTRIST by taking into account color information, neither present in Gist nor in CENTRIST. For such, it makes use of the W color space, which is derived from the opponent color space [22] (see Eq. (12)).

$$\begin{bmatrix} O1 \\ O2 \\ O3 \end{bmatrix} = \begin{bmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} W1 \\ W2 \end{bmatrix} = \begin{bmatrix} \frac{O1}{O3} \\ \frac{O2}{O3} \end{bmatrix} \tag{13}$$

It is often the case that the image descriptors output is employed to build a bag-of-visual-words (BOVW) via the well-known k-means clustering algorithm [16]–[18]. The final centroids generated by k-means are known as codewords. Hence, each trailer keyframe is assigned to one cluster represented by a codeword, and then a global multi-dimensional histogram is built for each trailer, where each dimension encodes a part of the trailer (e.g., if $t = 3$ there is the first, second, and the third part of the trailer encoded as a histogram). In its final step, each trailer in the test set is processed by the k-NN algorithm that computes its neighbors according to the χ^2 histogram distance.

C. Audio-based classification

The work described in [23] employs both audio and video features extracted from the entire movie trailer. The authors extracted 277 features (75 from video and 202 from audio) and trained several SVMs with the one-vs-one approach, performing feature selection for selecting the best features for identifying each genre pair. This approach requires $\frac{C_n(C_n-1)}{2}$ SVMs, where C_n is the number of classes. For a 7-class problem, it demands 21 classifiers, each trained with different features. First the authors define subset of features with the Self-Adaptive Harmony Search (SAHS) algorithm, and then they compute relative correlations for choosing the best subset for a given genre-pair. A good feature subset should contain: independent variables – Eq. (14), reduced intra-subset correlation – Eq. (15), and high correlation with the target class – Eq. (16), improving the overall relative correlation – Eq. (17).

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) log \frac{p(x,y)}{p_1(x)p_2(y)} \tag{14}$$

$$RI(S) = \frac{1}{C(|S|,2)} \sum_{i=1}^{|S|} \sum_{j=i+1}^{|S|} I(x_i, x_j) \tag{15}$$

$$RT(S,y) = \frac{1}{|S|} \sum_{i=1}^{|S|} I(x_i, y) \tag{16}$$

$$RC(S,y) = \frac{k \times RT(S,y)}{\sqrt{k + k \times (k-1) \times RI(S)}} \tag{17}$$

III. CNN-MoTion

In this section we detail our novel approach for movie genre classification, namely CNN-MoTion: Convolutional Neural Networks for Movie Trailer Classification. We first present the data preprocessing and augmentation step in Section III-A, and then the CNN architecture that was built to extract features from the movie trailers in Section III-B. Finally, we explain our post-processing learning step, which is needed for converting the frame-based classification given by the CNN into the ultimate movie trailer classification (Section III-C).

A. Preprocessing and Data Augmentation

In order to meet the input requirements for the CNN, a sequence of steps were taken to clean and augment the data. Cleaning the data is an important preprocessing step since the raw movie trailers contain data that are irrelevant for the purpose of genre identification, such as black borders used to compensate different video sizes and aspect ratios. In addition, when using models with thousands of parameters, such as CNNs, data augmentation is important to avoid overfitting. In a nutshell, we employ techniques to artificially enlarge the original data, generating label-preserving transformations with a low computational cost [24].

Since the input for a 2D-CNN is a set of images represented by 3D matrices ($width \times height \times RGB$), the preprocessing steps are applied for each frame of each movie trailer. First, the cleaning process is applied to clean and transform the frames to the format required by the CNN. It comprises the following steps: isolate the content area by removing black borders and downsizing the image to a fixed dimension ($width \times 256$), keeping its original aspect ratio. Second, the augmentation process is employed to artificially extract images (of size 224×224) from each frame ($width \times 224$). It generates 5 overlapping patches from different image regions (one patch per corner and one for the center). Alongside a horizontal flipping, this approach allow us to enlarge the original data tenfold. Figure 1 depicts the preprocessing and data augmentation process.

Finally, the preprocessed and augmented frames are then used as training set. The augmentation step is not performed for testing the model. Thus, for predicting purposes we use only the original frames without the black borders.

B. CNN Architecture

Convolutional Neural Networks (CNNs) [25] are a powerful class of models with impressive results in image recognition problems. Indeed, many studies have applied and improved these models for traditional image-based classification [24], [26], [27]. Notwithstanding, video-based classification has proven to be a much more challenging task.

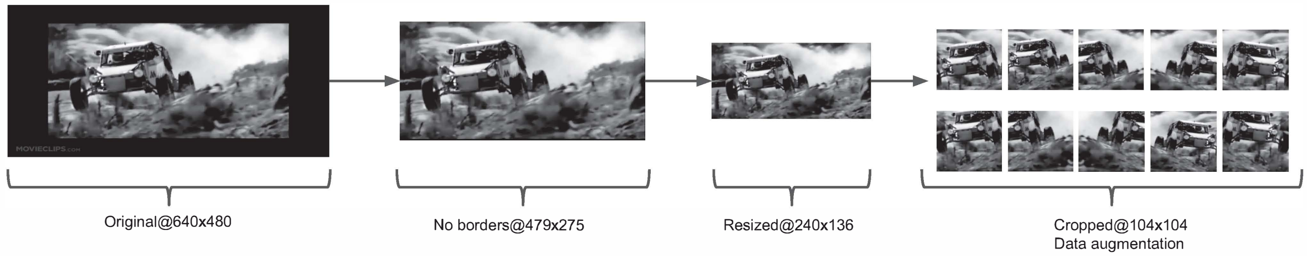


Fig. 1: Data preprocessing and augmentation in CNN-MoTion.

CNNs are neural networks that make use of convolutions instead of regular matrix multiplications in at least one of their layers [28]. Convolution is a mathematical operation over 2 functions resulting in a modified version of these functions. Eq. (18) defines a convolution, where b_{ij} is the bias for a given feature map, m indexes over the set of feature maps in the $(i-1)^{th}$ layer connected to the current feature map, w_{ijk}^{pq} is the value at the position (p,q) of the kernel connected to the k^{th} feature map, and P_i and Q_i are the height and width of the kernel, respectively.

$$v_{ij}^{xy} = relu\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1} w_{ijm}^{pq} v_{(i-1)m}^{(x+p)(y+q)}\right) \quad (18)$$

The intuition regarding the application of CNNs for video-based classification emerges from the basic video components: frames. Movie trailers can be seen as a set of sorted frames, and each frame (image) will probably express a given genre regarding the particular shot that it represents. However, note that it is perfectly possible that some frames are more important in defining the actual movie genre than others, and also that there may be frames that are genre-free or *generic* regardless of the movie at hand. In a standard CNN architecture, each video frame consists of a single instance labeled according to the genre of the entire video. Note that the relationship among frames is not taken into account when considering a 2D-CNN, which means the network will evaluate each frame separately, since it does not convolve into the temporal dimension.

Each stage of a CNN comprises one or more convolutional layers (often with a pooling step) and is followed by one or more fully connected layers (as in a standard multi-layer neural network). The CNN architecture is designed to take advantage of the 2D structure of an input image or any other 2D input abstraction that is developed according to the application at hand.

Based on Simonyan [29], the CNN-MoTion architecture is defined as follows: $C(16,3)-C(16,3)-P(2)-C(32,3)-C(32,3)-P(2)-C(64,3)-C(64,3)-P(2)-C(128,3)-C(128,3)-P(2)-C(128,3)-C(128,3)-P(2)-FC(2048)-FC(2048)$, where C(#filters, filters' size) denotes a convolution, P denotes a pooling layer (size 2×2), and FC(#nodes) denotes a Fully-Connected architecture. All convolutional layers are succeeded by ReLU non-linear normalization. The network input consists of a $224\times 224\times 3$ trailer frame. The result is an output array with 4 probabilities: one for each movie genre.

For optimizing the network, we employ the Stochastic Gradient Descent (SGD) with mini-batch of 128 instances, learning rate of 1×10^{-3}, *momentum* of 0.9 and weight decay of 1×10^{-5}. A separate validation set was employed to select the best training model. The network makes use of the well-known Cross-Entropy loss function.

For decreasing the computational cost of passing all trailer frames to the CNN, we decided to split each trailer movie into m keyframes (scene representatives) by employing the shot-detection algorithm described in [19], plus an additional percentage of frames from every shot according to its length (10% of the frames in a scene), resulting in a total of n frames per movie (variable size according to the number and length of shots per trailer). Our training set consists of 1,2 million images (post data augmentation), belonging to around 350 movie trailers.

CNN-MoTion offers different strategies for performing the final genre prediction based on the four probabilities provided by the CNN per test frame i, ρ_{ig}, $g \in \mathcal{G} = \{action, comedy, drama, horror\}$:

- CNN-MoTion-S: classifies each movie trailer according to the maximum weighted genre probability, as described in Eq. (19).
- CNN-MoTion-P: employs a SVM post-processing learning step to fine-tune the classification (see Section III-C for details). The following frameworks (denoted by subscript letters) indicate which features are used during the post-processing learning step.
  - CNN-MoTion-P_P: only the CNN weighted predictions (Eq. (20)) are used in the post-processing step.
  - CNN-MoTion-P_{APV}: MFCC audio features are used as features alongside the CNN weighted predictions (Eq. (20)) and low-level video features.
  - CNN-MoTion-P_H: the frequencies of elements in a scene histogram are used as features during the post-processing step.
  - CNN-MoTion-P_{AHP}: MFCC audio features, scene-histograms, and weighted predictions are used during the post-processing step.

$$\mathcal{P}_t = \underset{g}{\operatorname{argmax}}[\rho_g] \tag{19}$$

$$\rho_g = \frac{\sum_{i \in \mathcal{F}} p_{ig}}{\sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{G}} p_{ij}} \tag{20}$$

C. Post-Processing Learning Step

The CNN output in CNN-MoTion can be seen somehow as a genre histogram of the entire movie trailer. We know that even for action and horror movies it is quite common the presence of *generic* frames, like people talking, landscape shots, etc. Action movies may not have the majority of frames expressing the "action" genre, considering that maybe the majority of frames could be *generic*.

CNN-MoTion's more naïve classification approach, CNN-MoTion-S, does not account for this fact. For instance, if the CNN associates these *generic* frames to a given genre, such as drama or comedy, the final classification of an action movie trailer most probably will not be accurate. Our idea, thus, for the CNN-MoTion-P version is to train an SVM classifier with the four CNN predictions ρ_g produced by the best training model. The novel training set for this post-processing learning step will thus contain 4 features (likelihood of each genre provided by the CNN) for each of the training set instances.

Since our dataset is labeled at the movie level, there exists a semantic gap between the frame and movie levels, which can lead the CNN to be much less accurate than it is in problems such as object recognition from images. To address this issue, one could try many different things, such as: i) manually providing labels for all dataset scenes, a tedious and laborious task; ii) automatically providing labels to scenes using a pre-trained scene recognition model [30], and then performing some kind of mapping between sets of scene labels and movie genres; iii) automatically providing labels for scenes with an unsupervised learning algorithm, and also performing some kind of mapping between these novel labels and movie genres.

In CNN-MoTion, we decided to implement the unsupervised approach which represent each movie trailer as a bag of visual features. For each frame, we extracted the output from the last convolutional layer, which is a 2048 feature vector. Then, we calculate the average of these values per scene, going from frame-level analysis to scene-level analysis. This step generates a 2048-long vector that represents each movie trailer scene. Afterwards, we perform k-means clustering to automatically find scene categories. For finding the proper value of k (number of clusters), we run k-means on the training set and test the approach on validation data, varying the value of k 2 and 150. Once we have defined the clusters, CNN-MoTion assigns each trailer scene to a “category" (Figure 2), building a scene-level histogram, where the histogram bins are the clusters. Finally, the k-bin histogram (H) act as potentially novel features for the post-classification step. Since we are dealing with a high-dimensional feature space, the initialization has an important impact on the clusters that are found. Thus, CNN-MoTion runs k-means 20 times for finding the best prototype initialization. All feature values are normalized in the $[0, 1]$ range.

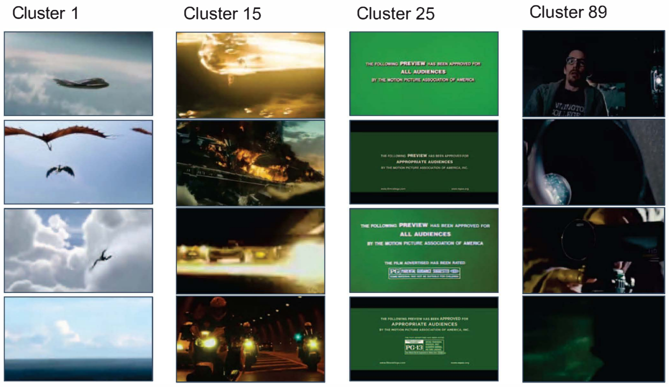


Fig. 2: Scene-level histogram found with k-Means.

So far, CNN-MoTion has several features at disposal: i) scene histograms provided by the unsupervised clustering algorithm (H, for short); ii) weighted genre predictions for each trailer (P) (Eq. (20); iii) low-level video features from [19] (V_1) and from [23] (V_2); and iv) audio features [23] (A). For evaluating the discriminative power of these feature subsets, we performed SVM classification with and without feature selection based on the information gain criterion [31], [32]. Table I presents details on the subsets of features.

TABLE I: Feature sets summary.

Feature Set	Abrev.	Vector size
Video Low Level Features [19]	V_1	4
Video Low Level Features [23]	V_2	75
Audio Features [23]	A	202
MFCC (5 statistics) [23], [33]	MFCC	52
CENTRIST scene histogram [19]	CENTRIST	100
Gist scene histogram [19]	GIST	100
w-CENTRIST scene histogram [19]	w-CENTRIST	100
Weighted CNN Predictions [Ours]	P	4
CNN scene histograms [Ours]	H	89*

In Table II we provide the comparison results regarding the available feature subsets. Feature vectors P and H, which were extracted by CNN-MoTion, have shown to be more discriminative than the other feature subsets. Comparing only the image and video-based features, we see that our approach is 15% better in terms of accuracy terms than the second place subset of features (V_2). The features extracted by the CNN based on the trailer frames are only behind of the audio A set, which contains 277 audio features.

Since the scene histogram feature vector is dependent of the unsupervised learning algorithm, its size is equal to the number of clusters k that was defined. The best k found by CNN-MoTion is 89. The MFCC descriptor is in the A feature vector, however we decided to create a separate set since it is a strong audio classification baseline [34], [35].

IV. Experimental Analysis

For validating the performance of CNN-MoTion, we performed several experiments over a novel movie trailer dataset that we have developed and make it now publicly available

TABLE II: Comparing the discriminative power of the feature subsets.

Method	Accuracy	Accuracy with Feat. Selection
w-CENTRIST + k-NN	29.50%	N/A
CENTRIST + k-NN	29.97%	N/A
V1 + k-NN	33.14%	N/A
Gist + k-NN	34.21%	N/A
V1 + SVM	35.31%	31.56%
V2 + SVM	43.44%	46.88%
MFCC + SVM	59.69%	65.31%
A + SVM	59.38%	**66.21%**
P [Ours] + SVM	58.44%	60.63%
H [Ours] + SVM	**62.19%**	63.44%

(see Section IV-A). In Section IV-B, we detail the baseline algorithms that we compare with CNN-MoTion. These algorithms are currently the state-of-the-art approaches in the movie genre classification domain. Finally, in Section IV-C we present the results of this empirical analysis and a discussion on our findings.

A. Dataset

To validate the hypothesis that movie trailer genres can be properly identified by CNN-MoTion, we need a labeled movie trailer dataset. Zhou et. al. [18] describe their own movie trailer data, though it is not made publicly available for the research community. Another curious fact is that 54% of the trailers in their dataset belong at the same time to three out of the four genres (the same four genres evaluated in here), and their reported accuracy values consider a correct classification whenever their approach classify the movie trailer as belonging to any of the labeled genres, which means movies with 3 genres have a 75% probability of being correctly classified simply by chance.

We have developed a novel movie trailers dataset hereby called LMTD (Labeled Movie Trailer Data), which comprises more than 3500 trailers whose genres are known, and we make it publicly available for the interested reader[1]. The ≈ 3500 movie trailers are distributed over 22 different genres. For creating LMTD, we selected trailers with runtimes between 60 and 200 seconds and release data after 1974.

In this paper, to avoid the problems identified in the work of Zhou et. al. [18], we have selected a subset of 1067 movie trailers from the LMTD, as presented in Table III, where each trailer belongs to one of 4 disjoint genres (action, comedy, drama, or horror). Note that this subset is a consequence of 1) restricting to 4 genres among the 22 existing ones, and 2) selecting all disjoint movie trailer from the 4 selected genres. This subset is called LMTD-4. The training, validation, and test sets were chosen randomly among the available trailers.

To collect the movie trailers, we developed a script to download them from a licensed Youtube channel. They are stored as *.mp4* files. Genres were automatically collected from IMDB (Internet Movie Database) and properly assigned to each collected trailer. Every movie is associated with at most 3 genres, which are identified in the respective filenames.

TABLE III: LMTD-4 dataset.

Genre	Training	Validation	Test	Total
Action	97	90	98	285
Comedy	98	93	91	282
Drama	100	84	101	285
Horror	86	65	64	215

B. Baselines and Parameters

To validate our results we compare CNN-MoTion with the state-of-the-art methods in movie genre classification, namely Gist [16], CENTRIST [17], w-CENTRIST [18], low-level feature extraction [19], and one-vs-one SVMs [23].

Note that the low-level features extraction approach presented by [19] makes use of a strategy that cannot be directly compared to the other methods. Therefore, we employ the same strategy of the work by Zhou et al. [18] for classifying the low-level features, which is to perform k-NN classification with a varying number of neighbors k.

For training the 6 one-vs-one SVMs as described in [23], we used the RBF (Radial Basis Function) kernel. Both C and γ parameters were defined by the best performance on the validation set. All values were normalized between -1 and 1. For the sake of simplicity, we performed feature selection with the information gain as heuristic instead of the SAHS plus cross-correlation as described in [23].

The remaining baselines are CENTRIST, w-CENTRIST, and Gist. For these approaches, we set the same parameters as defined in [18], namely: BOVW of 200 codewords, 100 bin histogram with $t = 3$, and k-NN with $k = 5$.

C. Results and Discussion

Table IV presents the results of our experimental analysis. Some of the baseline methods reach an accuracy of $\approx 30\%$. Considering that classification by chance would achieve $\approx 25\%$ of accuracy, one can easily verify the high-complexity of analyzing hundreds of movies and automatically discovering their genre through image-based approaches.

TABLE IV: Results of the experimental analysis.

Method	Accuracy	Accuracy with Feat. Selection
w-CENTRIST [18]	29.50%	N/A
CENTRIST [18]	29.97%	N/A
Gist [18]	34.21%	N/A
One-vs-One SVM [23]	47.50%	66.87%
CNN-MoTion-S	49.06%	N/A
CNN-MoTion-P$_{APV}$	60.94%	72.19%
CNN-MoTion-P$_{AHP}$	**65.31%**	**73.75%**

Table IV shows that CNN-MoTion outperforms all image/video based approaches, and that is true even for our simplest approach namely CNN-MoTion-S. Comparing models based on both audio and video features, our best result is around 7% superior than the state-of-the-art methods. These results indicate that the CNN feature extraction process is indeed better than then image/video features used in [23]. In addition, they show that even though our training method is

[1] In our research group website, http://www.inf.pucrs.br/gpin, go to the downloads section.

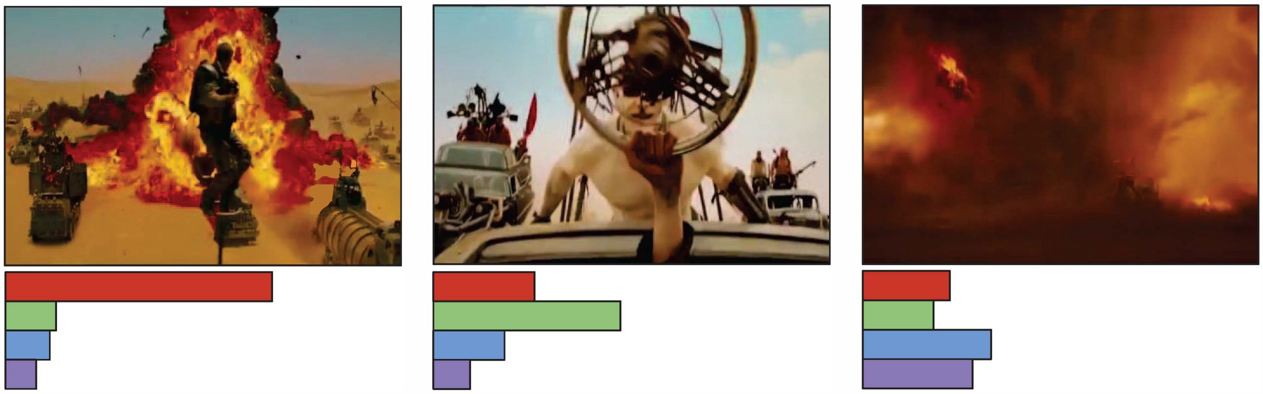


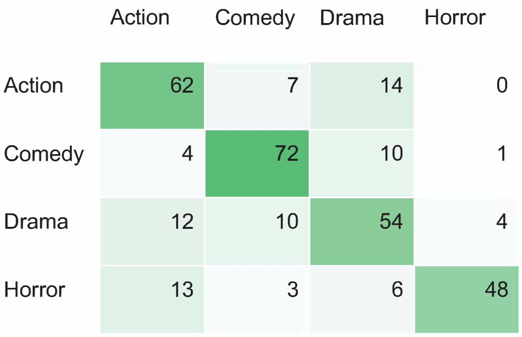
Fig. 3: Frames predicted by CNN-MoTion. The bar graph under the frames presents the respective genre probability. Orange stands for action, green for comedy, blue for drama, and purple for horror.

based on weakly-labeled data, the Convolutional Neural Nets could learn discriminative semantic features.

We noticed after a careful analysis of the per-frame results that CNN-MoTion-S tends to classify "dubious" frames (frames that we would classify as *generic*) as belonging to the comedy genre, which is a genre particularly hard to extract features from. *CNN-MoTion-P_{APV}*, which is the version that performs the post-processing learning step with the aid of low-level and audio features, seems to have properly addressed this issue, since it was capable of correctly classifying action trailers that were previously being classified as comedies/dramas. We also noticed that the color variance feature indeed helped to correctly classify several horror movies. The average shot length and the MFCC features proved to be helpful for discriminating all genres that were explored in our experiment.

By further analysing the CNN-MoTion predictions we observe that our model seems to have learned robust features. The network could find relations not directly indicated in the dataset. For instance, in Figure 3 one can see sampled frames from *Mad Max: Fury Road*. All these frames are annotated as action in our dataset. The network have predicted the “action" genre for the first frame (68% probability), comedy for the second (47%), and for the third there is a draw between drama (32%) and horror (28%). One can observe that all these predictions are plausible: the first frame clearly indicates action, considering it shows an explosion; however, both the second and the third frames do not highlight any particular genre.

Based on the accuracy measure, the best variation of CNN-MoTion is CNN-MoTion-P$_{AHP}$, with 73.45%. For more performance details, see the confusion matrix in Figure 4. Note that our method has a specificity $(TN/(TN+FP))$ of 97.41% for predicting horror movies. For comedies, its specificity is of 89.13%. However, the major issues are: action movies being predicted as drama, which occurs 14 times; and horror and drama movies being predicted as action trailers.



	Action	Comedy	Drama	Horror
Action	62	7	14	0
Comedy	4	72	10	1
Drama	12	10	54	4
Horror	13	3	6	48

Fig. 4: Confusion matrix of the CNN-MoTion-P$_{AHP}$ method. Columns indicate the predictions and rows the real classes.

V. Conclusions

Automatic video analysis is a complex problem yet to be effectively handled by machine learning algorithms. The specialized literature presents different techniques for video classification, most of them based on the extraction of sophisticated image descriptors and further execution of traditional learning algorithms. In the last couple of years, many efforts have been employed for improving the automatic analysis of videos, specially with the rise of the so-called deep learning algorithms.

The problem investigated in this paper was the automatic movie genre classification, which is far from presenting solutions as good as those provided in several image classification contests. For addressing such a problem, we proposed a novel deep-learning based strategy named CNN-MoTion – Convolutional Neural Networks for Movie Trailer Classification. In our experiments, we have compared our novel approach with the current state-of-the-art techniques, namely Gist [16], CENTRIST [17], w-CENTRIST [18], and low-level feature extraction [19], [23]. Our results clearly indicate that CNN-MoTion has the edge in the movie genre classification problem, which confirms the current trend in which deep learning approaches are becoming the state-of-the-art in many media-based applications (image, audio, and video recognition).

Even though our results show a strong improvement over the state-of-the-art methods, we are confident there is much to be done until movie genre classification is a solved research problem. For instance, we believe we could enhance the results by incorporating convolutions over the time dimension in order to avoid the *generic frame* issue that was detected during the experimental analysis. Hence, for future work we intend to develop a novel 3D-CNN architecture [12] for CNN-MoTion in order to capture the relationship among frames within the same trailer.

Acknowledgements

We would like to thank the Brazilian research agency CAPES for funding this research. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

[1] B. B. Le Cun, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*. Citeseer, 1990.

[2] U. Dogan, J. Edelbrunner, and I. Iossifidis, "Autonomous driving: A comparison of machine learning techniques by means of the prediction of lane change behavior," in *2011 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec 2011, pp. 1837–1843.

[3] R. C. Barros, M. P. Basgalupp, A. C. P. L. F. de Carvalho, and A. A. Freitas, "Automatic design of decision-tree algorithms with evolutionary algorithms," *Evol. Comput.*, vol. 21, no. 4, pp. 659–684, Nov. 2013. [Online]. Available: http://dx.doi.org/10.1162/EVCO_a_00101

[4] R. C. Barros, M. P. Basgalupp, A. A. Freitas, and A. C. P. L. F. de Carvalho, "Evolutionary design of decision-tree algorithms tailored to microarray gene expression data sets," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 6, pp. 873–892, Dec 2014.

[5] R. Cerri, R. C. Barros, and A. C. P. L. F. de Carvalho, "A genetic algorithm for hierarchical multi-label classification," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, ser. SAC '12. New York, NY, USA: ACM, 2012, pp. 250–255.

[6] ——, "Hierarchical multi-label classification using local neural networks," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 39 – 56, 2014.

[7] R. C. Barros, D. D. Ruiz, N. N. Tenorio, M. P. Basgalupp, and K. Becker, "Issues on estimating software metrics in a large software operation," in *32nd Annual IEEE Software Engineering Workshop (SEW 2008)*, Oct 2008, pp. 152–160.

[8] M. P. Basgalupp, R. C. Barros, and D. D. Ruiz, "Predicting software maintenance effort through evolutionary-based decision trees," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, ser. SAC '12. New York, NY, USA: ACM, 2012, pp. 1209–1214.

[9] M. P. Basgalupp, R. C. Barros, T. S. da Silva, and A. C. P. L. F. de Carvalho, "Software effort prediction: A hyper-heuristic decision-tree based approach," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, ser. SAC '13. New York, NY, USA: ACM, 2013, pp. 1109–1116.

[10] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. P. L. F. d. Carvalho, and J. a. Gama, "Data stream clustering: A survey," *ACM Comput. Surv.*, vol. 46, no. 1, pp. 13:1–13:31, Jul. 2013.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *arXiv preprint arXiv:1409.4842*, 2014.

[12] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 1, pp. 221–231, 2013.

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

[14] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *Human Behavior Understanding*. Springer, 2011, pp. 29–39.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[16] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[17] J. Wu and J. M. Rehg, "Where am i: Place instance and category recognition using spatial pact," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

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

[19] Z. Rasheed, Y. Sheikh, and M. Shah, "On the use of computable features for film classification," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 1, pp. 52–64, 2005.

[20] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[21] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 7, pp. 629–639, 1990.

[22] K. E. Van De Sande, T. Gevers, and C. G. Snoek, "Evaluating color descriptors for object and scene recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1582–1596, 2010.

[23] Y.-F. Huang and S.-H. Wang, "Movie genre classification using svm with audio and video features," in *Active Media Technology*. Springer, 2012, pp. 1–10.

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

[25] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*. IEEE, 2010, pp. 253–256.

[26] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1915–1929, 2013.

[27] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in neural information processing systems*, 2012, pp. 2843–2851.

[28] Y. Bengio, I. J. Goodfellow, and A. Courville, "Deep learning," 2015, book in preparation for MIT Press. [Online]. Available: http://www.iro.umontreal.ca/~bengioy/dlbook

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[30] J. Johnson, A. Karpathy, and F. Li, "Densecap: Fully convolutional localization networks for dense captioning," *CoRR*, vol. abs/1511.07571, 2015. [Online]. Available: http://arxiv.org/abs/1511.07571

[31] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[32] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[33] S. K. Jain, "Movies Genres Classifier using Neural Network," pp. 610–615, 2009.

[34] V. Tiwari, "Mfcc and its applications in speaker recognition," *International Journal on Emerging Technologies*, vol. 1, no. 1, pp. 19–22, 2010.

[35] M. R. Hasan, M. Jamil, and M. G. R. M. S. Rahman, "Speaker identification using mel frequency cepstral coefficients," *variations*, vol. 1, p. 4, 2004.