Baltic J. Modern Computing, Vol. 6 (2018), No. 3, 235-246
https://doi.org/10.22364/bjmc.2018.6.3.02

On-line Television Stream Classification by Genre

Karlis Martins BRIEDIS, Karlis FREIVALDS

Faculty of Computing, University of Latvia, Raina bulv. 19, Riga, LV-1586, Riga, Latvia

k.m.briedis@gmail.com, karlis.freivalds@lu.lv

Abstract. Convolutional neural networks (CNNs) have become the state-of-the-art solution for image classification and other related problems. This paper investigates the use of CNNs' features for on-line television stream classification by genre of the programme. As most existing offline classification solutions propose the use of low level audio-visual video descriptors, this paper compares the precision achieved by simple structure multi-layer perceptrons (MLP) and long short-term memory (LSTM) recurrent neural networks (RNNs) using either low level visual and audial descriptors or activations of InceptionV3 CNN's global pooling layer as features. The best real-time classification accuracy on evaluation data set of 71,6% was achieved by an LSTM RNN of CNN features, supporting the use of CNNs for television genre classification.

Keywords: television genre classification, television stream classification, video classification, neural networks, InceptionV3

1. Introduction

On-line television (TV) genre classification attempts to classify continuous television stream by the genre of broadcasts, labelling programmes as either commercials, movie, series, talk show, or etc., in real-time without any additional information about boundaries of shows.

TV genre classification can be applied to various use cases and a wide range of users, mostly for obtaining statistics of the content. For example, TV channels can label their archives, gather and analyse information about the content of their competitor's channels. Advertisers can recognize their aired commercials and analyse the surrounding TV content, e.g. time distance from the previous programme. In some countries, commercial detection and content recognition is important to national agencies that are supervising TV broadcasters and their compliance with the regulation. For end users the programme classification by genre can provide additional information about the broadcast, a custom genre classifier, matching their individual categorization, can be potentially created.

Several studies have shown promising results by applying machine learning classifiers to various audial and visual video features, but most of the work has been done in classifying whole and separated programme recordings of single genre, which has less use in real life applications. For most users, TV content is received as a continuous stream of video, without distinct boundaries between two different broadcasts, and for real-time classification only left (past) context is available and it should be small enough to be effectively analysed. For the purpose of this work, we

consider context of up to 3 minutes. Performance of both left and two-sided context is tested.

In recent years with the advancement of deep learning, the use of convolutional neural networks has achieved state-of-the-art results on many computer vision tasks, such as image recognition, object-detection, and video classification (Szegedy et al., 2016; Karpathy et al., 2014). The advancements of image recognition are often applied to solve video classification problems by use of similar neural network architecture or model weights obtained from network pretrained on some large image dataset. Use of an already developed convolutional neural network is much easier than careful feature engineering and has been observed to outperform classifiers based on manually extracted visual descriptors, e.g. for sports video classification in (Karpathy et al., 2014).

This paper investigates the use of high-level layer activations of InceptionV3 (Szegedy et al. 2016) CNN, pretrained on ImageNet dataset, for the on-line TV genre classification. Features are used both for single frame classification and as input for a long short-term memory (LSTM) recurrent neural network (RNN) for adding context. Results are compared with classifier utilizing manually extracted audial and visual features.

2. Related work

Traditional TV genre classification methods, using hand-crafted audio-visual features, have been successfully applied to the classification problem. However, this approach requires careful selection and processing of used features. Most of the work has been done on classification of complete and separated programmes, which is slightly different problem than on-line TV genre classification investigated in this paper, as features of whole programme are available and, in every video, only frames of one genre are present.

For the problem of complete and separated programme classification, Yuan et al. (2006) use a 10 dimension feature vector of whole clip (3-10 minutes). It consists of average shot length, cut percentage, average colour difference between every two frames, and ratio of each of four camera motion types: still, pan, zoom, and others. In addition, spatial features of face frames ratio, average brightness, and average colour entropy are used. Hierarchical binary SVMs are deployed to classify extracted features – each vertex discriminates genres yet to be classified in 2 groups. The best average classification accuracy of 86.97% is achieved by global optimal SVM binary tree, which is selected by performing cross-validation for each possible binary tree. It outperforms local optimal SVM binary tree, hierarchical SVM built by K-means, typical 1-vs-1 SVM scheme, and C4.5 decision tree. Though the achieved accuracy might be too optimistic as the dataset used is made by cutting videos into short clips with duration of 3 to 10 minutes and 50% of clips are selected for training, rest for testing, so there is no guarantee in the paper that different fragments of same broadcast are not in different sets.

Montagnuolu and Messina (2007) propose the use of four parallel MLPs, where each MLP classifies different group of features: low-level visual descriptors, structural, cognitive, and aural properties of video. Low-level visual pattern vector contains information about 7 features: all three HSV colour space components, YCbCr colour space brightness, frame difference between every two frames, and textural features described by contrast and directionality. For each feature a 65-bin histogram is

computed, and 10-component Gaussian mixture is modelled. The final feature vector consists of mean, standard deviation, and weight of each Gaussian mixture component and feature. Structural pattern vector consists of average shot length and 65-bin histogram of shot lengths; the cognitive pattern vector consists of average faces per frame, 11-bin histogram of face count in each frame and 9-bin histogram of placements of detected faces in 3x3 grid. The aural pattern vector contains the ratio of seven audio classes – speech, silence, noise, music, pure speaker, speaker plus noise, speaker plus music – and average speech rate. As part of the work, a TV dataset of about 110 hours of recordings was developed. By using 6-fold cross-validation to train MLPs with 1 hidden layer, a classification accuracy of 92% was achieved over seven genres. In their further work (Montagnuolo and Messina, 2009) they extended the research by improving structural and cognitive features, which allowed to increase the accuracy to 94,9% on the same dataset. The later research shows that the use of cognitive features increases the classification accuracy only by 0,8% and that most important components are visual and structural features, achieving respectively 85,6% and 83,3% when used alone.

The same dataset was deployed in further research by Ekenel and Semela (2013), and Kim et al. (2013). Ekenel and Semela used several visual – HSV colour histogram, colour moments, autocorrelogram, cooccurrence texture, wavelet texture grid, and edge histogram – and audial – Mel-frequency cepstral coefficients, fundamental frequency, signal energy, and zero crossing rate – features in addition to cognitive features as suggested in (Montagnuolo and Messina, 2007). Authors trained a separate binary SVM classifier for each feature and genre, and used the average across all SVMs, achieving almost perfect 99,2% classification accuracy.

Kim, Georgiou, and Narayanan use only audio features for classification purpose by deploying an acoustic topic model, originally designed to capture contextual information embedded within audio segments. Latent Dirichlet allocation with 64 topics is used on acoustic words derived from 13 mel frequency cepstral coefficients (MFCCs) quantized to 2048 acoustic words by use of *Linde-Buzo-Gray Vector Quantization*. After unsupervised acoustic topic modelling is done, probability distribution over latent acoustic topics is then used as input feature vector for SVM classifier. The used method yielded 94.3% accuracy for classification of whole programmes, 73% and 82% accuracy for on-line classification with context of 1 and 6 seconds, respectively. Results are compared with Gaussian mixture model with the same number of topics over MFCCs, which yielded slightly better performance when using segments under 1 second but underperformed for longer segments.

By analysing the accuracy achieved by other authors, which is reported to be as high as 99,2% on dataset of more than 110 hours of TV recordings, we can conclude that TV genre classification of complete programmes is a solved problem, but further research can be done in on-line TV genre classification. Also, additional research can be done in development of simpler classification methods that do not require extensive feature selection and engineering, such as the use of activations of CNNs.

3. Approach

In this work we propose several neural network-based classifiers for on-line TV stream classification by genre. To investigate the use of convolutional neural network's features for our problem, InceptionV3 model pretrained on ImageNet is used for single frame classification and for feature extraction for an LSTM classifier. To compare the accuracy

achieved by CNN-based solutions with traditional TV genre classification methods, we collect a set of hand-crafted audio-visual features and classify them using models of similar architecture to CNN-based solutions.

For the training of all models, *Keras* 2.1 (Chollet, 2015) library with *Tensorflow* backend is used. For all newly created layers, *Keras* default parameters (e.g. initialization of weights) was used throughout experiments, unless stated otherwise. For all models, categorical cross entropy loss function is used. All fully connected layers are using *ReLU* activation function, except last fully connected layers, which use *softmax* activation.

3.1. Classifier based on manually extracted features

To build a classifier of manually extracted features, various low level audial and visual descriptors used by other authors for TV genre classification, TV commercial detection, and music classification were collected. Then a set of audial and low level visual features was chosen, building up a 486-dimmension feature vector. Most features are using only information of 1 video frame and audio frame of respective length (40ms). Some features are also using information extracted from previous frame. All used descriptors can be divided in three categories – colour, edge, and audial features. Later these features are used to train an MLP classifier of TV genre.

3.1.1. Colour features

RGB histogram difference is calculated as absolute difference between normalized 64-bin RGB histograms of two consequent frames. Histograms are obtained by quantizing 24-bit pixel values to 6-bit values, using only 2 most significant bits for each colour channel.

HSV histogram difference is calculated similarly to RGB histogram difference, but a 256-bin histogram is obtained for each frame by using 4, 2, and 2 bits of hue (H), saturation (S), and value (V) components, respectively.

Brightness difference, average brightness and standard deviation, and dark pixel ratio are all calculated from Y component of YCbCr colour space. Brightness difference is expressed as absolute difference between Y value histograms, quantized to 128 bins and normalized to sum of 1. Average brightness and standard deviation are calculated from all Y values of each pixel. Dark pixel ratio expresses ratio of pixels in the frame with Y value less than 25% of maximum.

Frame difference (FD) is similar but simpler feature than histogram difference. It is calculated as mean absolute difference of each pixel values between two consequent frames. Both colour (mean absolute difference of each RGB colour component) and grayscale (mean absolute difference of averages of RGB colour components) difference is used in this work. FD shows the true similarity of frames, whereas histogram difference can be little, if two frames have similar colours in different positions.

3.1.2. Edge features

Features derived from edges are often good descriptors of action as proposed by Lienhart et al. (1997). For the edge detection the Canny edge detection algorithm (Canny, 1987)

is applied to each frame with 3x3 Sobel operator and lower hysteresis threshold of 40%, upper threshold of 80%.

Edge change ratio (ECR) represents structural changes between two frames and for frame *n* is defined as:

$$ECR_n = \max\left(\frac{X_n^{in}}{\sigma_n}, \quad \frac{X_{n-1}^{out}}{\sigma_{n-1}}\right)$$

where σ_n is count of edge pixels in frame *n*, X_n^{in} – count of entering edge pixels (pixels that are edges in frame *n*, but where not edges in frame *n-1*), and $\frac{X_{n-1}^{out}}{\sigma_{n-1}}$ – count of exiting edge pixels of frame *n-1*.

Edge stable ratio (ESR) is ratio between preserved and total number of edges between two adjacent frames:

$$ESR_n = \frac{|X_{n-1} \cap X_n|}{|X_{n-1} \cup X_n|}$$

where X_n are edge pixels of frame *n* and $|X_n|$ count of such pixels.

Edge-based contrast (EBC) represents ratio between strong and weak edges of the frame. We calculate it as:

$$EBC_n = \frac{S_n - W_n - 1}{S_n + W_n + 1}$$

where S_n is count of strong and W_n – count of weak edges in frame *n*. Edges with gradient magnitude above upper hysteresis threshold are considered to be strong, below it – weak.

Static region distribution can be used to detect presence of logo and other static regions that could help detection of TV genre, e.g. news bar. To calculate static pixels, we propose the use of momentum value for each pixel of frame – if pixel belongs to edge, momentum is increased, otherwise it is decreases. We define momentum *p* for *n-th* frame (x, y) pixel as:

$$p_n(x,y) = min(p_n^{inc}(x,y), \quad p_n^{dec}(x,y))$$

$$p_n^{inc}(x,y) = E(x,y) * 0{,}01 + p_{n-1}^{inc}(x,y) * 0{,}99$$

$$p_n^{dec}(x,y) = E(x,y) * 0{,}99 + p_{n-1}^{dec}(x,y) * 0{,}01$$

$$E(x,y) = \begin{cases} 1, & if\ (x,y)\ is\ edge\ pixel \\ 0, & otherwise \end{cases}$$

After calculating momentum for all pixels of *n-th* frame, adaptive thresholding of $max\left(\max_{x,y}\left(p_n(x,y)\right);\ 0{,}8\right) * 0{,}8$ was used. Then, frame is divided into 4 equal columns and 3 equal rows and average static pixel count is calculated for each block, resulting to final 12-dimension vector of static region distribution.

3.1.3. Audial features

Volume can be useful for detection of silences, which are sometimes used between scenes. It is defined as average of absolute values of all samples in given audio fragment.

Zero-crossing rate (ZCR) represents the rate at which amplitude of signal passes through zero. Together with silence, it can be used to discriminate between voiced and unvoiced speech (Bachu et al., 2010). It is defined as:

$$ZCR_n = \frac{1}{2 * (M_n - 1)} * \left(\sum_{m=1}^{M_n} |sign(n_m) - sign(n_{m-1})|\right)$$

where M_n is sample count for from n and n_k – k-th audio ample of frame n.

Short-time energy (STE) is used as additional feature to help discriminate between voice and music. It is used as:

$$STE_n = \log(\int_0^{\omega_0} |F(\omega)|^2 d\omega)$$

where $|F(\omega)|^2$ is energy at frequency ω and ω_0 – half of used sampling rate.

Spectral flux (SF) describes spectral difference between every two audio frames. In (Barbedo and Lopes, 2007) it is defined as:

$$SF_n = \sum_{k=1}^{K} (\log F_n(k) - \log F_{n-1}(k))^2$$

where $F_n(k)$ is k-th DFT coefficient for n-th audio frame.

Mel Frequency Cepstral Coefficients (MFCCs) is perception-orientated audio representation format, successfully applied for speech and music recognition. We extract MFCCs using "python_speech_features" library (Lyons, 2013) and use first 13 coefficients.

3.1.4. Feature classifier

To be able to better compare results with CNN-based solutions, we use a naïve approach of classifying raw extracted features using an MLP, without any additional modelling. In attempts to reduce processing time, we deployed random hyperparameter optimization for three types of networks:

1) Extracting features at 1fps and using every feature vector
2) Extracting features at 25fps, but using only every 25th feature vector
3) Extracting features at 25fps and using every feature vector

Unfortunately, solutions (1) and (2), which would be able to give boost in processing time, yielded the maximum of 59% validation accuracy compared to 69% of (3), therefore only (3) was furtherly investigated. The final used MLP configuration is:

- Input layer (486 input neurons);
- Fully connected layer with 2048 neurons (input dropout – 15%);
- Fully connected layer with neuron for each genre (input dropout – 50%).

Experiments with LSTM were not successful – as 25fps features were used, only context of few seconds of video could be used to achieve real-time classification speed (for LSTM with 128 units and sequence length of 1 minute, evaluation of each second took more than 3 seconds on NVIDIA Tesla K80, which cannot be used for on-line classification). Also, for all three types of networks, median filter yielded better accuracy than any deployed LSTM, therefore RNN solution was not considered for manually extracted features.

3.2. CNN feature-based classifiers

Modern image classification CNNs have proven to be highly successful at different image classification problems, even outperforming accuracy achieved by humans. Models inspired by architectures of well performing image classification CNN's, such as VGG and InceptionV3, have been successfully applied for video classification. Karpathy

et al. (2014) were able achieve 82,4% top 5 video classification accuracy on dataset of 1 million YouTube videos belonging to 487 classes (Sports-1M dataset) by using architecture similar to the ImageNet challenge winning *AlexNet* model (Krizhevsky, Sutskever, and Hinton, 2012), extended in time by different techniques of fusion of temporal features. By re-training last 3 layers of their best-performing classifier, they were able to achieve 68% classification accuracy on UCF-101 dataset (80% for classification of just sports classes). Carreira and Zisserman (2017) also used existing InceptionV1 architecture, but inflated all filter and pooling layers to 3 dimensions, and, by training separate models for both RGB and optical flow streams, achieved 93,4% accuracy on UCF-101 data set.

As deep modern CNNs requires a very large dataset and can take weeks to train, transfer learning is often used to adapt models trained on ImageNet or other large datasets to other use cases. As we expect the CNN to learn generic features, e.g. edges, at the bottom layers of network and higher-level features on top layers, it is possible to use activations of these layers as features of image. Ng et al. (2015) uses pretrained AlexNet and GoogLeNet (InceptionV1) to extract features and classify videos using different temporal feature pooling approaches and LSTM. The highest top 5 classification accuracy on Sports-1M dataset of 90.8% was achieved by convolutional

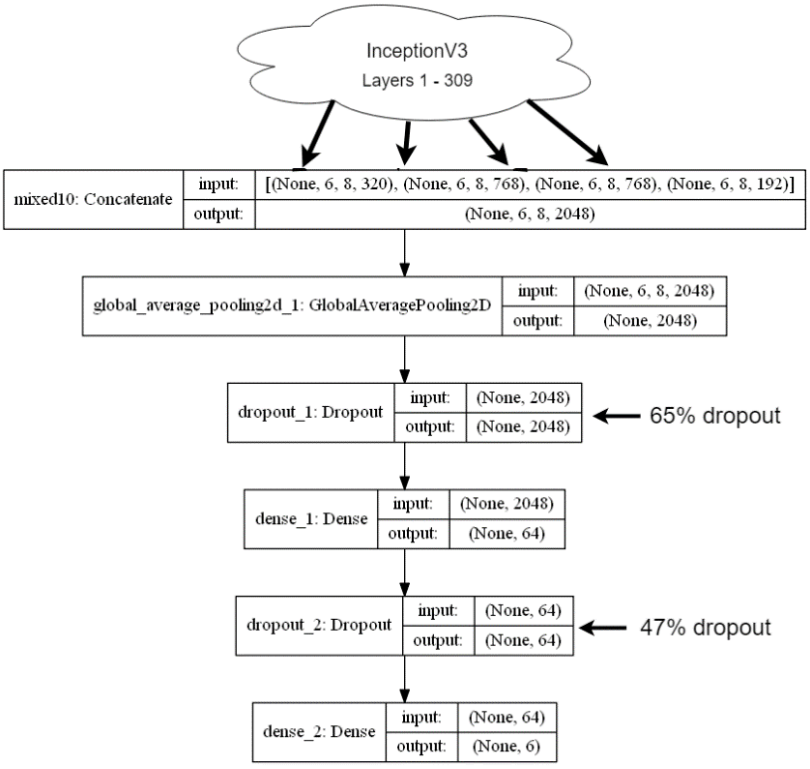

Fig. 1. Structure of the deployed single frame classifier.
First dimension of the input and the output shape denotes the batch size.

pooling approach, where max-pooling over final convolutional layer across the video's frames is performed, followed by LSTM solution with accuracy of 90,5%, significantly improving the accuracy achieved by Karpathy et al. The highest accuracy on UCF-101 dataset of 88,6% was achieved by an LSTM network.

For purposes of this work we chose to use LSTM to add context, as use of convolutional pooling loses information about the sequence of frames, which can be crucial for our application of real-time classification where different parts of fragment can belong to different classes.

After attempts to use the InceptionV3's default configuration of fully connected layers showed signs of overfitting, extensive hyperparameter random search was used. The final chosen structure of model is shown in Fig. 1.

First, only newly created layers (dense_1 and dense_2) were trained, until the network loss had not decreased for 20 epochs. Then a fine tuning of last 164 layers (last 5 inception blocks) was applied with SGD optimizer and small learning rate of 1E-6 for 200 epochs.

For LSTM model we initially tested classification of middle or last frame of sequence, but only last frame solution is furtherly considered as it can be used for real-time classification and performed slightly better. To increase training performance of network, we first extracted activations of global average 2D pooling layer (see Fig. 1) for each 25^{th} frame of dataset videos, obtaining 2048-dimension feature vector. We found that for our case this model works best with input sequence length of 150 frames (2,5 minutes). The final structure can be described as:

- Input layer with shape (150, 2048);
- Fully connected layer with 128 neurons (input dropout – 46%);
- LSTM layer with 512 output units (input dropout – 48%);
- Fully connected layer with neuron for each genre.

4. Experimental results

To empirically evaluate proposed on-line TV genre classification methods, we collected more than 28 hours of TV recordings of two public Latvian television channels – LTV1 and LTV7. The initially accumulated data set had too few different broadcasts, therefore the desired K-fold cross-validation could not be used. We used a holdout validation with training set of 70% and validations set of 30%. Recordings were split on per-broadcast basis and we filtered genres to leave only ones that had ratio of train set between 60% and 80%. As the final solution for each method was chosen based on accuracy on validation set, an additional evaluation set was required. The final content distribution of each set is shown Table 1. Data set consists of 6 genres with 32% of programmes being series, commercials – 25%, lifestyle shows – 13%, news – 12%, documentary movies – 11% and kids programmes – 7%. Evaluation data set is proportionally big (about 34% of used broadcasts) as it was recorded more than 2 months after broadcasts of train and validation sets. The long interval between recordings also reduces the probability of containing similar commercials and shows to broadcasts in train and validation sets.

Table 1. Duration of genres in used dataset

Genre	Train, min	Validation, min	Evaluation, min	Total, min
Commercial	163	68	127	357
Lifestyle	71	25	95	191
Series	250	98	109	457
Kids	24	12	59	95
Documentary	69	35	44	148
News	91	38	46	175
Total	**668**	**275**	**480**	**1423**

The testing of trained models on validation set yielded 69% percent classification accuracy of classifier based on manually extracted feature, 60,6% of single frame CNN model and 78,8% for the LSTM of CNN features.

As all methods are technically classifying only one frame of video, we can make use of additional context, by applying filter to classification predictions. Different types of filters – median of labels, mean of prediction probabilities, Gaussian blur on prediction probabilities – and context lengths were tested on validation data set for both left and two-sided contexts. Results of different filters for LSTM of CNN features are shown in Table 2. In case of the LSTM, a slight improvement from initial 78,8% classification accuracy is observed – approximately 3% increase for both left and both-sided contexts. Similarly, an increase of 6,3% and 10,3% was achieved for classifier of manually extracted features and single frame CNN classifier, respectively. As difference between left-sided and both-sided contexts was not significant, we chose left-sided context because it allows real-time classification.

Table 2. Additional filter applied to predictions of LSTM of CNN features

Type	1 minute	2 minutes	3 minutes
Past context			
Mean	**81,1%**	78,5%	75,9%
Median	74,1%	74,1%	74,1%
Gaussian	79,5%	79,1%	79,0%
Two-sided context			
Mean	81,5%	81,8%	81,2%
Median	78,2%	78,2%	78,8%
Gaussian	80,7%	81,6%	**81,9%**

After additional context filter and length for each method was chosen (median filters with length of 3 minutes for single frame classifiers and 1 minute for LSTM), we tested our trained models on evaluation data set, with final results shown in Table 3. Both single frame classifiers showed very close results, with manually extracted feature MLP being slightly better on validation set but worse on evaluation set. The best solution for

all cases was achieved by LSTM of CNN features, significantly outperforming other solutions by 10 to 20 per cents. The achieved results are worse than the achieved real-time classification accuracy of 82% in (Kim, Georgiou, and Narayanan 2013) but it is hard to objectively compare results achieved between works, because different datasets with different genres were deployed.

Table 3. Results of on-line classification on evaluation set (validation – accuracy on validation set after applying filter, base and final – accuracy before and after applying mean filter on evaluation set, respectively)

	Validation	Base	Final
Single frame manual features	72,3%	50,5%	54,7%
Single frame CNN	68,4%	52,5%	56,8%
LSTM of CNN features	**81,1%**	**69,1%**	**71,6%**

To be able to better compare our results with other authors, who use complete and separated broadcasts, we deployed naïve solution by averaging genre probabilities given by our trained models to predict the genre of whole programme. Comparison with other works is shown Table 4. Our error rate is significantly higher than error rates of others, but as our models were not built for whole programme classification and used data set is less than 40% of second smallest compared data set, our results can be considered to be good. By analysing confusion matrix of LSTM of CNN features for whole programme classification (Table 5) we can see that commercials and series are classified with 100% accuracy, but none of documentaries are classified correctly. Large difference between classes could be caused by uneven distribution of genres in our train set.

Table 4. Comparison of achieved whole programme classification accuracy between our work and pervious works

	Method	Dataset size, minutes	Classification accuracy
(Yuan et al., 2006)	Hierarchical SVMs	3600	86,97%
(Montagnuolo and Messina, 2007)	MLPs	6692	92%
(Montagnuolo and Messina, 2009)	MLPs	6692	94,9%
(Ekenel and Semela, 2013)	Binary SVM for each feature and genre	6692	99,2%
(Kim, Georgiou, and Narayanan, 2013)	ATM + SVM	6692	94,3%
Single frame, manual features	**MLP + Mean**	**1423**	**65,5%**
Single frame, CNN features	**CNN + Mean**	**1423**	**67,9%**
LSTM of CNN features	**LSTM + Mean**	**1423**	**77,0%**

Table 5. Confusion matrix of whole programme classification with LSTM of CNN features (average precision – 77%)

	Series	Commercials	Lifestyle	Kids	Documentary	News
Series	**100%**	0%	0%	0%	0%	0%
Commercials	0%	**100%**	0%	0%	0%	0%
Lifestyle	12%	31%	**41%**	0%	16%	0%
Kids	26%	20%	9%	**31%**	14%	0%
Documentary	9%	0%	74%	0%	**0%**	17%
News	0%	14%	0%	0%	0%	**86%**

5. Conclusions and future work

In this paper the use of convolutional neural network features for the on-line television stream classification by genre was investigated. Particularly, we compared classification performance of classifiers using hand-crafted audio-visual features and classifiers using features extracted from InceptionV3 CNN. Our experiments show, that CNN feature-based solution outperforms hand-crafted features-based solution, which means that further work on television genre using CNN features is viable. While performing significantly worse than reviewed whole programme classifiers, our solution was able to achieve 71,6% precision on our evaluation data set for on-line validation and 77% for whole programme validation. Data set of about 24 hours of Latvian public television channels was created during the work and is available upon request for use in additional research.

For additional future work, a larger data set of television recordings must be gathered and use of CNN features should be investigated deeper, e.g. by using activations of different layers.

Acknowledgements

Authors would like to thank the University of Latvia, Faculty of Computing for providing GPU computing resources for the training of classifiers. The research was partly developed under the University of Latvia contract no. AAP2016/B032 “Innovative information technologies”.

References

Bachu R.G., Kopparthi S., Adapa B., Barkana B.D. (2010). Voiced/Unvoiced Decision for Speech Signals Based on Zero-Crossing Rate and Energy. In *Advanced Techniques in Computing Sciences and Software Engineering*, 279–282. Springer, Dordrecht. doi:10.1007/978-90-481-3660-5_47.

Barbedo J.G.A., Lopes A. (2007). Automatic Genre Classification of Musical Signals. *EURASIP Journal on Advances in Signal Processing* 64960 (1). Springer International Publishing. doi:10.1155/2007/64960.

Canny J. (1987). A Computational Approach to Edge Detection. In *Readings in Computer Vision*, edited by Martin A. Fischler and Oscar Firschein, 184–203. Elsevier. doi:10.1016/B978-0-08-051581-6.50024-6.

Carreira J., Zisserman A. (2017). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4724–4733. Honolulu, HI, USA: IEEE. doi:10.1109/CVPR.2017.502.

Chollet F. (2015). Keras, available at https://keras.io/

Ekenel H.K., Semela T. (2013). Multimodal Genre Classification of TV Programs and YouTube Videos. *Multimedia Tools and Applications* 63 (2). Springer US: 547–567. doi:10.1007/s11042-011-0923-x.

Karpathy A., Toderici G., Shetty S., Leung T., Sukthankar R, Fei-Fei L. (2014). Large-Scale Video Classification with Convolutional Neural Networks. In *CVPR '14 Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, 1725–1732. IEEE Computer Society. doi:10.1109/CVPR.2014.223.

Kim S., Georgiou P., Narayanan S. (2013). On-Line Genre Classification of TV Programs Using Audio Content. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 798–802. IEEE.

Krizhevsky A., Sutskever I., Hinton G.E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, 1097–1105.

Lienhart R., Kuhmunch C., Effelsberg W. (1997). On the Detection and Recognition of Television Commercials. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, 509–516. Ottawa, Ontario, Canada: IEEE. doi:10.1109/MMCS.1997.609763.

Lyons J. (2013). Python_speech_features, available at https://github.com/jameslyons/python_speech_features.

Montagnuolo M., Messina A. (2007). TV Genre Classification Using Multimodal Information and Multilayer Perceptrons. In *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*, 730–741. Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-74782-6_63.

Montagnuolo M., Messina A. (2009). Parallel Neural Networks for Multimodal Video Genre Classification. *Multimedia Tools and Applications* 41 (1). Springer US: 125–159. doi:10.1007/s11042-008-0222-3.

Ng J.Y.H., Hausknecht M.J., Vijayanarasimhan S., Vinyals O., Monga R., Toderici G. (2015). Beyond Short Snippets: Deep Networks for Video Classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4694–4702. IEEE. doi:10.1109/CVPR.2015.7299101.

Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. (2016). Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826. IEEE.

Yuan X., Lai W., Mei T., Hua X.S., Wu X.Q., Li S. (2006). Automatic Video Genre Categorization Using Hierarchical SVM. In *2006 International Conference on Image Processing*, 2905–2908. IEEE. doi:10.1109/ICIP.2006.313037.

Received June 4, 2018, accepted July 9, 2018