Integrated Computer-Aided Engineering 31 (2024) 439–453
DOI 10.3233/ICA-240740
IOS Press

Effectiveness of deep learning techniques in TV programs classification: A comparative analysis

Federico Candela[a,*], Angelo Giordano[a], Carmen Francesca Zagaria[b] and Francesco Carlo Morabito[c]
[a]*Dipartimento di Ingegneria dell' Informazione, dell' Infrastrutture e dell' Energia Sostenibile (DIIES) Department, University Mediterranea, Reggio Calabria, Italy*
[b]*Regional Communication Committee Calabria, Reggio Calabria, Italy*
[c]*Dipartimento Ingegneria Civile, Energia, Ambiente e Materiali (DICEAM), University Mediterranea, Reggio Calabria, Italy*

Abstract. In the application areas of streaming, social networks, and video-sharing platforms such as YouTube and Facebook, along with traditional television systems, programs' classification stands as a pivotal effort in multimedia content management. Despite recent advancements, it remains a scientific challenge for researchers. This paper proposes a novel approach for television monitoring systems and the classification of extended video content. In particular, it presents two distinct techniques for program classification. The first one leverages a framework integrating Structural Similarity Index Measurement and Convolutional Neural Network, which pipelines on stacked frames to classify program initiation, conclusion, and contents. Noteworthy, this versatile method can be seamlessly adapted across various systems. The second analyzed framework implies directly processing optical flow. Building upon a shot-boundary detection technique, it incorporates background subtraction to adaptively discern frame alterations. These alterations are subsequently categorized through the integration of a Transformers network, showcasing a potential advancement in program classification methodology. A comprehensive overview of the promising experimental results yielded by the two techniques is reported. The first technique achieved an accuracy of 95%, while the second one surpassed it with an even higher accuracy of 87% on multiclass classification. These results underscore the effectiveness and reliability of the proposed frameworks, and pave the way for a more efficient and precise content management in the ever-evolving landscape of multimedia platforms and streaming services.

Keywords: Deep learning, video classification, pattern recognition, video segmentation, few-shot learning

1. Introduction

In the realm of TV program recognition and content analysis, which includes acronyms, program types, and various data points, identifying relevant information is crucial, particularly when working with large datasets that require classification. This challenge becomes even more complex when optimizing network training in a supervised manner, especially with the introduction of new programs, TV program acronyms, or advertisements. Furthermore, it is essential to recognize that each channel has its unique characteristics and programming lineup. An examination of Italian television programs reveals that they can be broadly classified into two principal genres: fiction and non-fiction. Fiction encompasses TV films, series, miniseries, cartoons, soap operas, and telenovelas. Non-fiction, in contrast, includes pro-grams addressing real-life issues such as news, weather, talk shows, current affairs, popular science, cultural segments, variety and game shows, reality series, advertising, and teleshopping. The Communi-

[*] Corresponding author: Federico Candela, DIIES Department, University Mediterranea, 89124 Reggio Calabria RC, Italy. E-mail: federico.candela@unirc.it.

cations Guarantee Authority acts as the regulatory and supervisory body within the audiovisual communications sector, delegating certain responsibilities to the Regional Communications Committees (Co.Re.Com.). These bodies oversee local audiovisual broadcasts and address any irregularities, such as exceeding program durations, airing unauthorized commercials, broadcasting con-tent inappropriate for all audiences, categorizing de-bates (political, historical, etc.), recognizing program credits or acronyms, and classifying program types according to nationally regulated criteria [1]. Recent researches made significant advances in this area. This paper aims to contribute to this field, in order to aid communication agencies, especially through the deployment of two innovative and comparative methodologies. The first methodology implements a framework that integrates Structural Similarity Index Measure (SSIM) and ResNet50, as proposed in [2]. It analyzes stacked frames to classify the beginning and the end times of programs, along with their content. While versatile, it is limited by the predetermined image size required for SSIM comparison. The second methodology, which is an evolution of the first, is considers the processing of optical flow. This approach relies on a shot-boundary detection technique with background subtraction to pinpoint changes in frames, which are then categorized using a Transformers network. The rest of this document is organized as follows: Section 2 lays out the fundamental theories behind the techniques employed in our frameworks. Sections 3 and 4 evaluate the frameworks and discuss the results, respectively. Finally, Section 5 summarizes the conclusions from our research.

2. State of the art

Some of the most advanced video classification methods are founded on CNNs, which have evolved to include a variety of new methodologies. For instance, the 3D Convolutional Neural Networks (3D CNN) introduced by Tran et al. [3] utilize a three-dimensional kernel to extract features across multiple frames. Karen et al. [4] proposed the Two-Stream CNN, a model comprising two neural networks: one assessing the video's appearance and the other its motion. The appearance stream employs a standard CNN to analyze frames, while the motion stream leverages a 3D CNN to assess optical flow between frames. Wang et al. [5] introduced the Temporal Segment Network, which uses a 2D CNN for spatial analysis of video frames paired with a 1D CNN for temporal sequence analysis. Carreira et al. [6] adapted a 3D CNN, pre-trained on ImageNet, for video analysis by extracting features from frames and their temporal progression. Feichtenhofer et al. [7] combined a 'slow' 3D CNN for spatial analysis with a 'fast' 3D CNN for temporal sequence analysis. The ongoing advancement of these techniques has led to the development of attention-based networks, which concentrate on specific video segments for classification, often used in tandem with architectures like 3D CNNs or LSTMs. Pioneered by Bahdanau et al. [8], attention mechanisms have been applied in video classification by researchers such as Sharma et al. [9] in "Action Recognition Using Visual Attention." A fundamental initial step in video classification is video segmentation, which aims to partition the video stream into manageable segments for indexing [10].

In the domain of TV program recognition and content analysis, recent studies indicate that substantial strides have been achieved, highlighting the significant progress in this field. Yi Cao et al. [11] proposed a model that uses a CNN network to encapsulate the information extracted from video scenes, incorporating a visual attention technique via a separate convolutional neural network. This network generates a visual attention map. However, the model demands significant computational resources, notably for creating the visual attention map, which involves numerous convolutions and scalar products between large tensors. This could render the model computationally inefficient on less robust hardware. Additionally, the reliance on a visual attention map may reduce interpretability, as the criteria for selecting the most relevant video sections for classification aren't explicit. It might necessitate the application of model interpretation methods for a clearer understanding of its operation.

Fangzhao Wu et al. [12] applied a CNN for image analysis and an RNN for text analysis. They also employed multi-task learning to handle various tasks simultaneously and embedding techniques to numerically translate textual TV program descriptions for deep learning application. Nonetheless, potential enhancements could include the adoption of sophisticated data pre-processing, such as natural language processing (NLP), to capture more nuanced information from TV program descriptions, thereby im-proving the analysis quality. Moreover, the images in the study were downscaled to 64x64 pixels, potentially limiting the model's capacity to discern intricate visual details.

The dataset used in their research was sourced exclusively from the Chinese streaming platform Youku, which may affect the model's applicability to other regions and cultural contexts. In 'Automatic TV Pro-gram Genre Classification Using Deep Convolutional Neural Networks' [13], Hieu Khac et al. utilized a limited dataset of images from diverse origins, which constrained the model's generalizability across different genres. They implemented a VGG16 neural network to extract image visual features. Despite this, the model's ability to represent the semantic content of TV programs, such as dialogue or storylines, remains a limitation. They later used a Support Vector Machine (SVM) classifier to categorize each image by genre. However, the study did not benchmark the model against other genre classification methods, leaving its comparative efficacy undetermined.

3. Methods

3.1. Materials

In this section, we're going to introduce the two methods we have developed for classifying television broadcasts and extended videos. Our goal is to pro-vide an in-depth explanation of the methodologies we've utilized throughout the classification process. We'll delve into the specifics of each method, spot-lighting their distinctive features and the fundamental principles upon which they're based. This thorough analysis will clarify the two separate strategies, set ting the stage for an exhaustive evaluation of their efficiency and their ability to adapt to different scenarios. This level of detailed scrutiny is essential to pinpoint the most appropriate and effective approach for categorizing television content and longer video formats.

We created an initial dataset for the SSIM-CNN framework. We used a first dataset comprising test images to evaluate the SSIM [14]. Originally, the images in our dataset captured the opening and closing acronymous of a sports news program. We have since expanded this dataset to include content from additional programs beyond sports, incorporating various categories from a second dataset created for CNN. To train CNN network, we assembled datasets using image annotations sourced from web search engines and video frame captures from the specified channels. The project's initial phase concentrated on identifying content from sports news. We later expanded our dataset to include a wider range of categories. The training dataset now covers diverse genres: Geo documentaries (826 images), Religious events (769 images), Game shows (525 images), Talk shows (685 images), Sales promotions (470 images).

For our second initiative, the Shot Boundary Detection with Transformers framework, we have developed an enriched dataset of mini videos. These were generated utilizing Shot Boundary Detection techniques and were systematically classified into diverse categories following the A.g.Com program classification guidelines. This comprehensive dataset includes the following segments: Cartoons (559); Cooking (313); Culture (244); Debates (164); Religious (309); Geography (439); Interviews (476); Weather (337); Politics (100); Commercials (604); News Summaries (570); Sports (122); also integrating videos from UCF-101 [15] from specific categories due to data scarcity, particularly Basketball (15), Soccer (8), Tennis (15), Swimming (26), Golf (12), chosen based on the monitoring of the channels and the creation of the dataset itself, Teleshopping (450), and News bulletins (191).

3.2. Similarity structure index measure with convolutional neural network

The proposed architecture utilizes an image comparison system based on SSIM, augmented with a ResNet50 [16]. This novel method focuses on analyzing stacked frames from the target video. Each frame undergoes a detailed comparison against standardized test images obtained from the broadcasting channels of the TV shows in question.

The Structural Similarity Index Measure (SSIM) [17] works as a perceptual tool quantifying image quality degradation by measuring changes in structural information. Unlike most image quality metrics, which typically calculate discrepancies based on pixel value differences like mean squared error, the SSIM index reflects the human visual system's ability to detect structural information within a scene. It excels at discerning the details between a reference image and a comparison image. A metric that mimics this capability generally excels in tasks that require this level of discrimination. The SSIM index evaluates three essential characteristics of an image: Luminance, Contrast, and Structure, as shown in Fig. 1.

Consider a collection of test images; every image is represented as a Matrix I that capture specific moments of detection, such as the beginning or the end of a TV show's acronym. Each image has size $N \cdot N$, and we define with n the n-th image $\{I1, I2, I3, \ldots, In\}$.

A video is essentially a temporal sequence of images, each always represented as a matrix *Mi* where

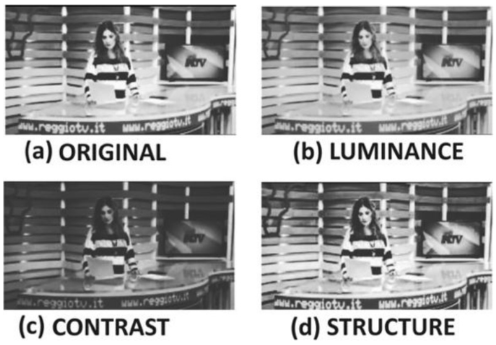


Fig. 1. Comparative visualization of the SSIM metrics: Original, Luminance, Contrast, and Structure.

i indicates the frame number over time. If the video consists of n frames, then we have a set of matrices $\{M1, M2, \ldots, Mn\}$.

The goal is to precisely identify these distinct moments, like the commencement or conclusion of a TV show acronym. Under the assumption of discrete signals, Luminance is determinate by computing the average of all pixel values:

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i; \mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i; \tag{1}$$

The luminance comparison function, denoted as $l\,(x, y)$, relies on μx and μy. In this context, xi represents i-th pixel value of image x, while *yi* denotes the i-th pixel value of image y. The variable

N represents the total number of pixel values. Regarding contrast, it is determined by calculating the standard deviation, i.e. the square root of the variance, across all pixel values:

$$\sigma_x = \left(\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2\right)^{\frac{1}{2}}; \tag{2}$$

$$\sigma_y = \left(\frac{1}{N-1}\sum_{i=1}^{N}(y_i - \mu_y)^2\right)^{\frac{1}{2}}; \tag{3}$$

The contrast denoted as $c(x, y)$, involves comparing σx and σy.

In this case, x and y represent the two images under comparison, and μ is the average of the pixel values. The structural comparison is conducted by dividing the input signal by its standard deviation, which normalizes the result to a standard deviation of one, facilitating a more reliable comparison:

$$N_x = \frac{x - \mu_x}{\sigma_x}; N_y = \frac{y - \mu_y}{\sigma_y}; \tag{4}$$

We define functions that compare two specified images based on these parameters. We refer to the luminance comparison function as follows:

$$l\,(x, y) = \frac{2_{\mu_x \mu_y} + C_1}{\mu_x^2 + \mu_y^2 + C_1}; \tag{5}$$

where C_1 serves as a constant to ensure stability when the denominator drops to zero. C_1 is given by:

$$C_1 = (K, L)^2; \tag{6}$$

where K is a constant and L represents the dynamic range of the pixel values, which is set to 255 because we are analyzing 8-bit images. The contrast comparison function is defined as follows:

$$c\,(x, y) = \frac{2_{\sigma_x \sigma_y} + C_2}{\sigma_x^2 + \sigma_y^2 + C_3}; \tag{7}$$

C_2 shares the same structure as C_1. The structure comparison function is defined as follows:

$$s\,(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x + \sigma_y + C_3}; \tag{8}$$

where σx represent the standard deviation of a given image, and σxy pertains to the covariance of images being compared. Now, we can define the similarity index using:

$$\mathit{SSIM}\,(x, y) = \left[l(x, y)^{\alpha} \cdot c(x, y)^{\beta} \cdot s\,(x, y)^{\gamma}\right]; \tag{9}$$

The parameters $\gamma > 0$, $\beta > 0$, and $\alpha > 0$ are utilized to adjust the relative prominence of the three components. By setting $\alpha = \beta = \gamma = 1$, and assigning $C_3 = C_2/2$, we obtain the following expression:

$$\mathit{SSIM}\,(x, y) = \frac{(2\mu_x \mu_y + C_1)\left(2_{\sigma_{xy}} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)}; \tag{10}$$

It is advantageous to use the SSIM index locally rather than globally for assessing the image quality. Instead of applying the metrics globally, it is more effective to apply them regionally for higher accuracy. Facing the challenge of comparing images extracted from videos, referred to as *Mn*, with test images, labeled In. It's crucial to note that while the Mn video frames include specific date and time information at the time of their recording, displayed on the edges of the image, the *In* test images have been saved with fixed date and time stamps, corresponding to the original video from which they were extracted. This temporal discrepancy between the test images and the video frames can vary, especially if the video frames are from recordings made on different days. This difference in date and time information can lead to errors in comparisons based on the

SSIM index, a metric used to assess image similarity. To address this issue, we have developed and applied a mask of $H \times L$ dimensions to both the Mn video frames and the *In* test images. The use of this mask allows us to exclude the date and time information from the comparison process, thereby improving the accuracy in evaluating the similarity between the images. This leads to a further reduction in image size. Regarding classification, we feed the video's SSIM index between an old and a new scene change duration, resulting in a floating-point value SSIM $(x, y) = fn$, where the index n represents the n-th comparison between the Test image *In*, and the n-th frame in the *Mn* video.

Using a standard parameter threshold t:

$$f_n > t; \tag{11}$$

By setting t to a high value, we can accurately identify when an image of interest represents a TV theme acronym. Once the image of interest is identified, CNN with ResNet50 initiates the classification of general content, including sports, human activities, sales, products, talk shows, debates, and others. For training the network, we converted the images to grayscale and employed the technique of transfer learning [18]. This involved pre-training the network on ImageNet [19]. Splitting the model into a head and body, training only the head while freezing the body. In our training data, we incorporate random rotations, zooms, shifts, shears, and flips to augment the dataset. We employed a stacked frame recognition technique: to achieve temporal classifications of scenes, we implemented a moving average prediction, by considering the frames per second of the video.

As we have represented the frames in a video as an M matrix, we define:

$$Y = \sum_{i=1}^{n} Mi; \tag{12}$$

The ResNet50 makes predictions on each frame, assigning a classification percentage to every nth frame. *Mn*, we write the prediction function as:

$$P\left(Mn\right) = pn; \tag{13}$$

here *pn* represents the probability assigned to the nth frame. We only consider the highest probabilities and can define a subset of these probabilities. Let's assume we want to consider the top k probabilities, where $k \leqslant N$ and N is the total number of frames being considered. We order the probabilities in descending order and take the first k: *{p(1), p(2), . . . , p(k)}* where $p(1) \geqslant p(2) \geqslant \ldots \geqslant p(k)$. Let $p(j)$ denote the jth-highest probability, after ordering all probabilities in descending order.

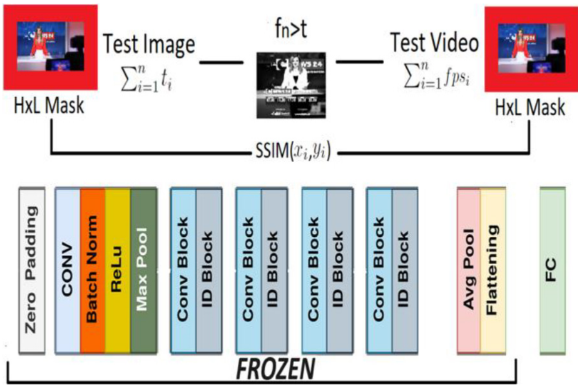


Fig. 2. Architecture of SSIM with CNN.

We now calculate the average of the top k probabilities as follows:

$$mean = \frac{1}{k} \sum_{j=1}^{k} p\left(j\right); \tag{14}$$

This value represents the overall mean probability based on the frames with the highest classification confidence of the neural network. This approach is used to evaluate the performance of the network on video segments.

3.3. Shot boundary detection with transformer

The proposed framework operates based on optical flow. In this scenario, the video to be analyzed is divided into subsequences.

Given a video as Y, let us define Y as a sequence of frames $Y = \{M1, M2, \ldots, Mn\}$ where *Mn* represents the i-th frame of the video. With the shot boundary detection technique, we divide the video Y into n sub videos, $y1, y2, \ldots, yn$, where each *yk* represents a video segment with a distinct semantic event. This subdivision can be expressed as: $Y = \bigcup_{j=1}^{k} yj$, where $yj = \{M_{aj}, M_{aj+1}, \ldots, M_b\}$ defining *aj* e *bj* as variables that represent the indices of the frames defining the start and end of each sub video *yj* with $1 \leqslant aj \leqslant bj \leqslant n$ e $aj + 1 = bi + 1$ for every i from 1 to $k - 1$. This ensures that each frame of Y belongs to exactly one sub video *yj*. Each subsequence will be the input for the Transformers Network [20].

The video segmentation is realized with a shot boundary detection [21], that generates a binary mask $My(i, l)$, where i represents the vertical coordinate of the pixel. Then, l denotes the horizontal coordinate of the pixel. Representing the area of the image occupied by the foreground object. This mask is produced by applying an adaptive threshold to the difference map between the current frame $Iy(i, l)$ and the background model $By(i, l)$. The adaptive threshold is determined as the mean plus a constant multiplied by the standard deviation. It is specifically dependent on the standard deviation of the difference map $My(i, l) = 1$ if:

$$|l_y\left(i, l\right) - B_y\left(i, l\right)| > \mu + k\sigma; \tag{15}$$

and 0 otherwise, where μ represents the mean of the difference map, highlighting the disparity between the current frame and the background model. σ denotes the standard deviation of the difference map, while k is a multiplicative constant used to compute the adaptive threshold for generating the binary mask.

In this study, we introduce a novel video classification model that harnesses the capabilities of Transformers, a category of neural networks renowned for their proficiency in handling sequential data. The structure of our model consists of several critical components. Initially, video subsequences yj, are pre-processed by segmenting them into frames, then sub-sampled to create sequences. These sequences are subsequently processed by a DenseNet121 [22], a pre-trained convolutional neural network, to extract prominent features. The top layers of the DenseNet are excluded to maintain its expertise in capturing detailed spatial information.

Features of each frame (X $\in R^{NxD}$) are arranged into a sequence (S $\in R^{TxNxD}$), like the patch-based method used in Vision Transformers. This sequence, enhanced with positional embeddings (PE $\in R^{TxD}$), is processed by a single layer of the Transformer. The output of the Transformer, Z, is provided by:

$$Z = \mathit{Transformer}\left(S + \mathit{PE}\right); \tag{16}$$

where S represents the sequence of features extracted from each frame of the video, organized to reflect the spatial and temporal structure of the original sequence of frames.

Each element of S is a feature vector describing a frame or frame segment of the video. PE represents positional embeddings, which are added to the S sequence to provide the Transformer with information about the temporal position of each frame within the sequence. This layer is designed to learn spatial and temporal dependencies among the features, providing a proficient solution for the analysis of video data. Moreover, the model makes use of a GlobalMaxPooling1D operation to effectively refine spatial information:

$$(\text{Zpooled} = \text{GlobalMaxPooling1D(Z)}); \tag{17}$$

and this is complemented by a dropout layer to reduce the risk of overfitting.

The Transformer features a single attention head, and projects the embeddings through a dense layer with a dimensionality of 4 ($F \in R^{Tx4}$), thereby enhancing the model's learning capabilities:

$$F = \mathit{Dense}\left(Z_{pooled}\right); \tag{18}$$

in essence, F represents the final processing of the input data through the Transformer model, where, after leveraging the spatial and temporal learning capabilities of the single attention head, the features are synthesized into a four-dimensional vector for each timestep. This condensed output, F, embodies the understanding gleaned by the model and is poised for deployment in decision-making stages, such as classification or advanced interpretation of patterns in video data.

4. Experimental verification

In this Section, we examine the experiments conducted on the two proposed frameworks. The experiments were executed on a dedicated system with the following specifications: an Intel(R) Xeon(R) Gold 6126 CPU at 2.6 GHz, 64 KiB of BIOS, 64 GiB DIMM DDR4 system memory, and 2 × GV100GL [Tesla V100 PCIe 32 GB]. The frameworks were developed using Python and the Keras library with TensorFlow backend. Video classification tests were performed for both frameworks on the same datasets. Specifically, we considered LaC as a local channel, and we considered additional channels such as RTV, TeleSpazio, TenTv while also analyzing two 24-hour video recordings.

4.1. Performance of the proposed system

To evaluate how well the system operates, we use P to represent a favorable outcome, and N to symbolize an unfavorable one. Here's how we classify the results: TP refers to the count of scenes accurately recognized in a video, FP is used for the count of scenes recognized in a video but labeled incorrectly, TN is the count for scenes that were misidentified in a video, and FN stands for the scenes in a video that went undetected or for any irregularities found. We paid more attention to the 2 transformer and shot boundary methodology.

The framework's performance was carefully assessed by utilizing:

$$\mathit{Precision} = \frac{TP}{TP + FP}; \tag{19}$$

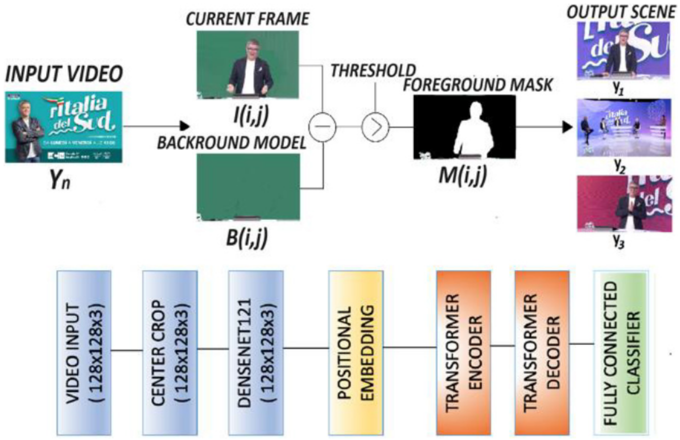


Fig. 3. Architecture of the shot boundary detection with transformer.

Table 1
Processed classification SSIM and CNN-ResNet50 output

Time	Label	Probability
00:00:11	LAC_SPORT_TV_THEME	98.28%
00:00:24	Football	99.94%
00:00:34	TV news	97.33%
00:00:55	Football	98.60%
00:00:57	Football	98.76%
00:00:59	TV news	99.00%
00:01:06	Football	99.48%

$$Recall = \frac{TP}{TP + FN}; \tag{20}$$

$$F\ score = 2 * \frac{Precision * Recall}{Precision + Recall}; \tag{21}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}; \tag{22}$$

Accurately describing the results obtained.

4.2. *Evaluation of similarity structure index measure with ResNet50*

The results of the classification are shown in Table 1, which serves as an example of the results derived from the framework classification process. In this table, we can examine the first row that details the performance of SSIM, then the classification done on each framework by ResNet50. We divided the dataset, allocating 80% for training and 20% for validation. Figure 4 illustrates the training loss and accuracy of the network, employing cross-entropy to measure the difference between the model's predictions and the actual labels throughout the training period. The network underwent training over 20, 50, 100, and 120 epochs.

Graph (a) – 20 epochs: The training loss decreases rapidly, indicating that the model is learning from the dataset effectively. Both the training and validation accuracy improve quickly, and appear to stabilize by the 20th epoch. There's a small gap between training and validation loss, suggesting minor overfitting.

Graph (b) – 50 epochs: This graph extends to 50 epochs and shows a continued decrease in training loss. The training and validation accuracy both rise and then plateau, indicating that the model may not be gaining significant improvements from additional epochs. There's a consistent gap between training and validation loss, but it does not appear to be widening significantly, which is positive.

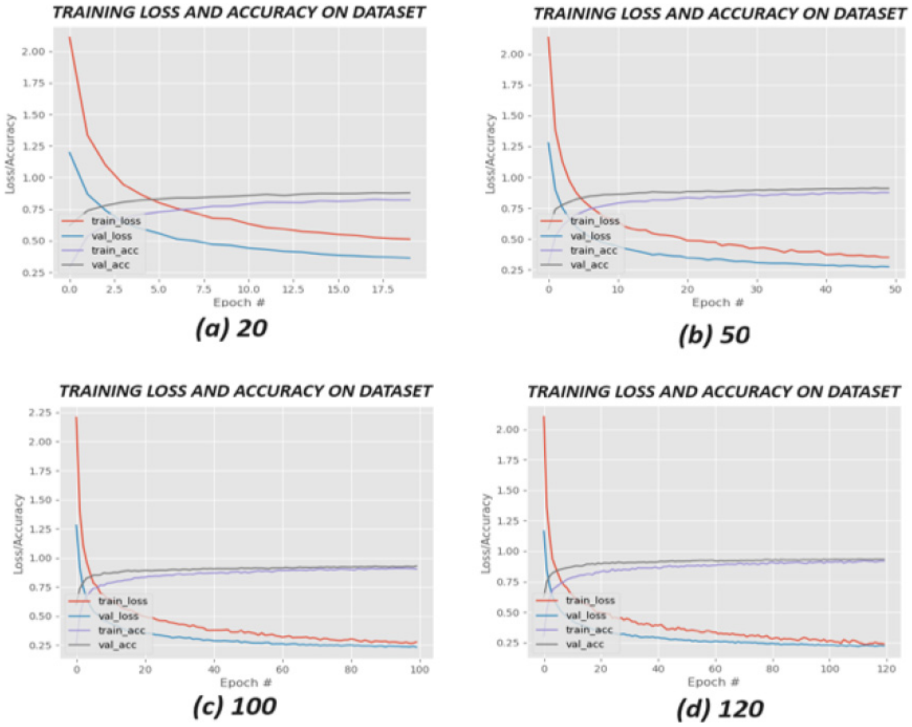
Graph (c) – 100 epochs: Here, over 100 epochs, the training loss continues to decrease but at a much slower rate. The accuracy seems to have plateaued. The gap between the training and validation loss appears slightly larger compared to the 50 epochs graph, which may indicate overfitting as the model continues to learn specifics about the training data that do not generalize to the validation data.

Graph (d) – 120 epochs: Extending the training to 120 epochs, the loss and accuracy trends seem consistent with the 100 epochs graph. There's a noticeable gap between the training and validation loss, which may suggest that the model isn't likely to benefit from further training on the same data without adjustments or regularization to reduce overfitting.

The best results obtained for 120 epochs shown in Table 2 are discussing the results, Geo and Religious categories have high precision, recall, and F1 scores, all around 0.94 to 0.98, indicating that the model performs very well in these categories, with a balanced ability

Table 2
Performance of the Network CNN-ResNet50

Class	Precision (%)	Recall (%)	F1_Score (%)	Support (%)
Geo	0.98	0.98	0.98	206
Religious	0.95	0.94	0.94	192
Game_show	0.94	0.95	0.93	137
Talk_show	0.93	0.95	0.94	190
Sales_promotion	0.94	0.92	0.94	118
Accuracy			**0.95**	843
macroavg	0.94	0.94	0.94	843
Weighted avg	0.95	0.95	0.95	843



(a) 20

(b) 50

(c) 100

(d) 120

Fig. 4. Comparison of learning curves for training and validation loss and accuracy on a dataset, with incremental epochs of 20, 50, 100, and 120.

to identify relevant cases (precision) and to identify all actual cases (recall).

Game show, Talk show, and Sales promotion categories have slightly lower but still robust performance metrics, ranging from 0.92 to 0.95, which implies that the model is generally reliable in these classifications as well.

The accuracy of 0.95 suggests that the model correctly classifies 95% of the overall data, which is quite high for most applications.

Both the macro average and weighted average scores across precision, recall, and F1 are consistent at 0.94 and 0.95 respectively. The macro average treats all classes equally, while the weighted average takes the support (the number of true instances for each label) into account. High values in both suggest that the model's performance is uniformly strong across all classes and that the model is not biased towards more frequently occurring classes.

The support for each class varies, with 'Geo' having the highest number of instances (206) and 'Sales promotion' the least (118). Despite these differences, the model's performance is steady across classes.

In conclusion, the model demonstrates excellent and consistent performance across different categories with no significant signs of bias towards frequent categories.

4.3. *Evaluation of shot boundary detection with transformers*

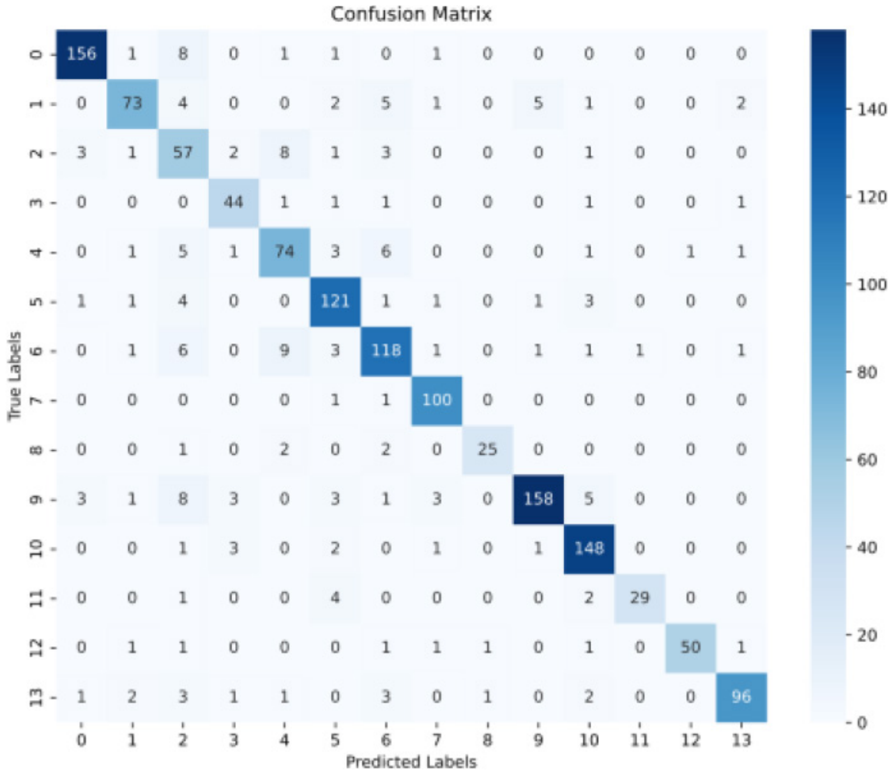
In our work, we pay special attention to the classification results obtained with this technique.

Table 3
Performance of the network shot boundary detection with transformer

Class	Precision (%)	Recall (%)	F1_Score (%)	Support (%)
50EPOCHS				
Cartoons	0.99	0.83	0.91	168
Cooking	0.92	0.71	0.80	93
Culture	0.71	0.58	0.64	76
Debates	0.75	0.90	0.81	49
Religious	0.72	0.80	0.76	93
Geography	0.81	0.95	0.88	133
Interviews	0.74	0.81	0.77	142
Weather	0.97	0.94	0.96	102
Politics	1.00	0.43	0.60	30
Commercials	0.81	0.89	0.85	185
News summaries	0.87	0.98	0.92	156
Sports	1.00	0.94	0.97	36
Teleshopping	0.81	0.88	0.84	57
News bulletins	0.90	0.77	0.83	110
Accuracy	**–**	**–**	**0.84**	**1430**
Macro avg	0.86	0.82	0.82	1430
Weighted avg	0.85	0.84	0.84	1430
100EPOCHS				
Cartoons	0.95	0.93	0.94	168
Cooking	0.89	0.78	0.83	93
Culture	0.58	0.75	0.65	76
Debates	0.81	0.90	0.85	49
Religious	0.77	0.80	0.78	93
Geography	0.85	0.91	0.88	133
Interviews	0.83	0.83	0.83	142
Weather	0.92	0.98	0.95	102
Politics	0.93	0.83	0.88	30
Commercials	0.95	0.85	0.90	185
News summaries	0.89	0.95	0.92	156
Sports	0.97	0.81	0.88	36
Teleshopping	0.98	0.88	0.93	57
News bulletins	0.94	0.87	0.91	110
Accuracy	**–**	**–**	**0.87**	**1430**
Macro avg	0.88	0.86	0.87	1430
Weighted avg	0.88	0.87	0.88	1430
150EPOCHS				
Cartoons	0.94	0.97	0.95	168
Cooking	0.90	0.74	0.81	93
Culture	0.71	0.54	0.61	76
Debates	0.60	0.92	0.73	49
Religious	0.77	0.84	0.80	93
Geography	0.93	0.81	0.87	133
Interviews	0.77	0.89	0.83	142
Weather	0.92	0.97	0.94	102
Politics	0.92	0.80	0.86	30
Commercials	0.91	0.87	0.89	185
News summaries	0.94	0.96	0.95	156
Sports	0.97	0.83	0.90	36
Teleshopping	0.83	0.91	0.87	57
News bulletins	0.91	0.85	0.88	110
Accuracy	**–**	**–**	**0.87**	**1430**
Macro avg	0.86	0.85	0.85	1430
Weighted avg	0.87	0.87	0.87	1430
200EPOCHS				
Cartoons	0.99	0.89	0.94	168
Cooking	0.84	0.69	0.76	93
Culture	0.80	0.58	0.67	76
Debates	0.88	0.90	0.89	49
Religious	0.88	0.66	0.75	93

Table 3, continued

Class	Precision (%)	Recall (%)	F1_Score (%)	Support (%)
Geography	0.70	0.97	0.82	133
Interviews	0.76	0.89	0.82	142
Weather	0.82	0.98	0.89	102
Politics	1.00	0.53	0.70	30
Commercials	0.91	0.84	0.87	185
News summaries	0.98	0.92	0.95	156
Sports	0.97	0.86	0.91	36
Teleshopping	0.84	0.93	0.88	57
News bulletins	0.78	0.91	0.84	110
Accuracy	–	–	**0.85**	**1430**
Macro avg	0.87	0.83	0.84	1430
Weighted avg	0.86	0.85	0.85	1430



	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	156	1	8	0	1	1	0	1	0	0	0	0	0	0
1	0	73	4	0	0	2	5	1	0	5	1	0	0	2
2	3	1	57	2	8	1	3	0	0	0	1	0	0	0
3	0	0	0	44	1	1	1	0	0	0	1	0	0	1
4	0	1	5	1	74	3	6	0	0	0	1	0	1	1
5	1	1	4	0	0	121	1	1	0	1	3	0	0	0
6	0	1	6	0	9	3	118	1	0	1	1	1	0	1
7	0	0	0	0	0	1	1	100	0	0	0	0	0	0
8	0	0	1	0	2	0	2	0	25	0	0	0	0	0
9	3	1	8	3	0	3	1	3	0	158	5	0	0	0
10	0	0	1	3	0	2	0	1	0	1	148	0	0	0
11	0	0	1	0	0	4	0	0	0	0	2	29	0	0
12	0	1	1	0	0	0	1	1	1	0	1	0	50	1
13	1	2	3	1	1	0	3	0	1	0	2	0	0	96

Fig. 5. Confusion matrix on 100 epochs.

We allocated 80% of the dataset for training and the remaining 20% for testing. We conducted experiments across the different numbers of epochs at 50, 100, 150, and 200 epochs, as reported in Table 3. The best results were achieved after 100 epochs, especially when analyzing the different categories, in conjunction with the corresponding confusion matrix as illustrated in Fig. 5.

Our classification model demonstrates good results, particularly in the Cartoons and Weather categories, achieving accuracies of 0.95 and 0.92, respectively. The model's precision in classified cartoons is confirmed by the confusion matrix, which shows 156 correct classifications out of 168 items, with very few false positives and negatives. Similarly, in the Weather category, the model correctly classified 100 out of 102 items.

For Commercials and News Summaries, the model showed high accuracy, with 158 and 148 correct classifications out of 185 and 156 items, respectively. In Sports, despite a very high accuracy of 0.97, the precision is lower at 0.81, suggesting some confusion with other categories; however, the confusion matrix reveals 29 correct classifications out of 36. Teleshopping exhibits the best performance with near-perfect accuracy of 0.98 and 50 correct classifications out of 57, despite

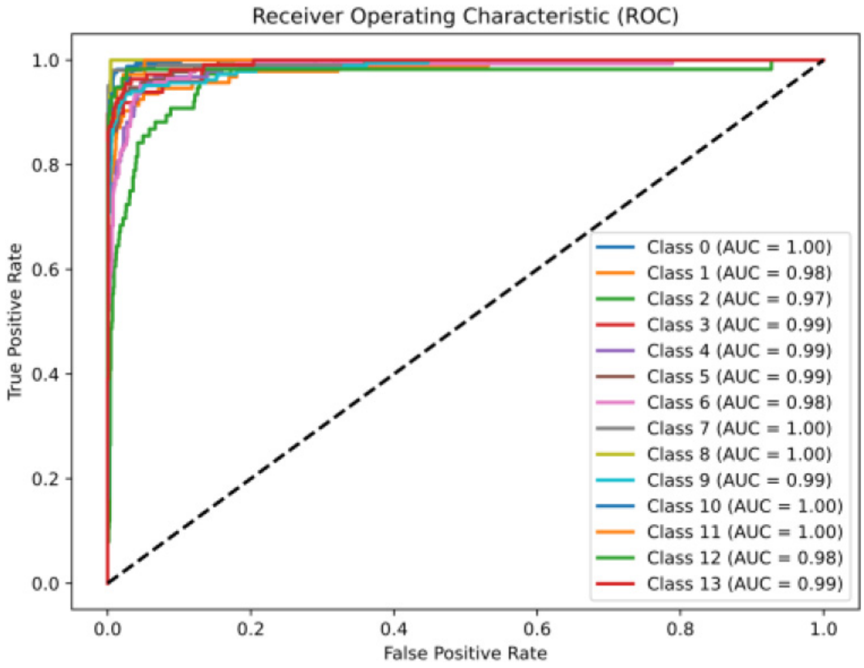


Fig. 6. ROC graphics on 100 epochs.

Table 4
Comparison of shot boundary detection and transformer with other methodologies

Methodologies	Accuracy
3dCNN [23]	90.2%
CNN+RNN [24]	80.2%
PAC+CNN [25]	89.3%
CNN+MLP [26]	93.7%
DNN [27]	53%
LogRegression [28]	82%
SSIM+CNN	**95%**
S.B.+Transf.	**96%**

a moderate amount of misclassification indicated by the confusion matrix.

The model's overall performance is robust with an accuracy of 0.87 across 1430 items. The macro averages for accuracy and precision, which calculate the average performance of the model for each category separately and then average these results, are 0.87 and 0.88, respectively. This indicates balanced performance across categories, ensuring that each category is given equal importance regardless of its size. Meanwhile, the weighted average, considering the number of items per category, confirms good overall performance. These results underscore the effectiveness of the model as a classification tool across a broad spectrum of categories.

Figure 6 shows the ROC (Receiver Operating Characteristic) curves over 100 epochs. These curves chart the model's classification efficacy across 13 distinct classes by plotting the true positive rate (TPR) against the false positive rate (FPR) for various threshold settings.

The key observations from the ROC curves include:

- Perfect Classification (AUC = 1.00): Classes 0, 7, 8, 10, and 11 achieved an AUC (Area Under the Curve) of 1.00, signifying flawless classification with an absence of both false positives and negatives. The ROC curves for these classes perfectly align with the ROC space's left and top edges, denoting 100% sensitivity and specificity.
- Near-Perfect Classification (AUC $\geqslant$ 0.98): Classes 1, 2, 3, 4, 5, 6, 9, 12, and 13 are characterized by near-perfect classification, with AUC values between 0.98 and 0.99. Positioned close to the top left corner, these curves reflect the model's high true positive rate alongside a minimal false positive rate for the classes.
- Consistency Across Classes: The high AUC values' uniformity across all classes indicates a robust model with consistent performance, reliably pinpointing true positives while concurrently keeping false positives to a minimum.
- Distinct Classes with No Overlapping Curves: The absence of overlapping curves implies clear distinction between classes, highlighting the model's effective differentiation capabilities.

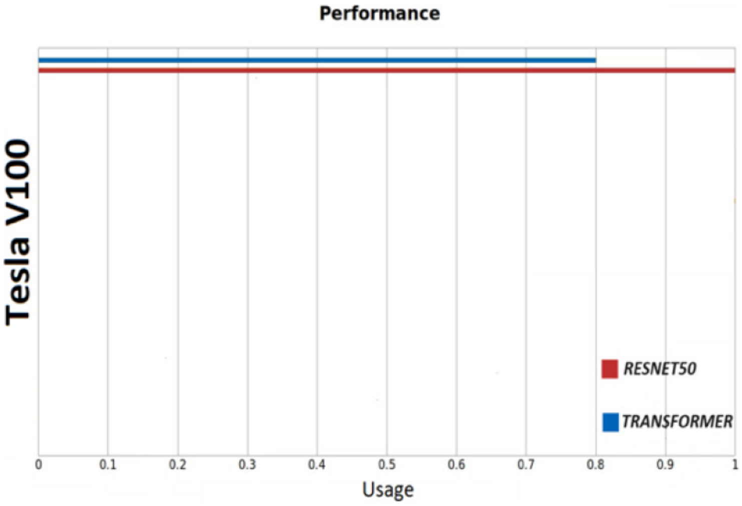


Fig. 7. GPU consumer during training of the CNN and Transformer.

The dashed line represents the baseline of random guessing (AUC = 0.50), with all class curves significantly outperforming this benchmark. This demonstrates that the model's predictions are substantially superior to those made by the chance.

4.4. Experimental results discussion

In this section, we provide a comparative analysis against existing research. The initial SSIM framework, when combined with a CNN, excels at quickly identifying specific TV program opening (or closing) sequence. Notably, it obviates the need for additional training when a channel updates its opening sequence; a simple test image input suffices. ResNet50 consistently shows proficiency in recognizing broader categories. However, it does come with a caveat: the alignment of the test image's dimensions with the video's frames per second (fps) is crucial. Conversely, the second framework, which utilizes the Transformers network for Shot Boundary Detection, adopts a more general approach to opening (or closing) sequence. Rather than focusing on individual opening sequence, it includes 'spots' that cover advertisements, aiming to universalize the model, thereby eliminating the need for retraining. While SSIM with ResNet50 sometimes struggles with accurately marking the beginning of TV News specifically when a journalist introduces a new report the results are generally reliable. Additionally, Fig. 7 details the GPU's usage during training. The accompanying graph reveals that the ResNet50 network requires more power than the Transformers network. Nonetheless, it achieved remarkable results in the 120-epoch training phase, boasting an impressive 95% accuracy rate.

4.5. Comparative analysis

For comparison with existing technologies, it is essential to highlight that our frameworks show promising results when benchmarked against ongoing research efforts. This underscores the potential and effectiveness of our approach amidst the current technological advancements in the field. In Table 4, we juxtapose our findings with those from recent studies in this area. For instance, in [23], the authors leverage the UCF101 dataset to classify a variety of human actions or activities in videos. Notably, their study reveals that our proposed methodologies forego the need for optical flow extraction, enhancing efficiency in terms of execution speed. Our approach utilizes a dual-stream data setup, one stream for visual inputs and another for motion, ensuring a robust representation of spatiotemporal data. It should be noted, though, that training deep neural networks like three-dimensional CNNs demands extensive data and computational power, culminating in a top accuracy of 90.2% for our Two-stream 3D network.

In [24], the researchers introduce a hybrid model that combines a Convolutional Neural Network (CNN) with a Recurrent Neural Network (RNN) to discern video content types, classifying them into categories such as 'Animation,' 'Gaming,' 'Natural Content,' 'Flat Content,' and so on. They propose a novel technique for classifying only key frames, thus curtailing processing time without significantly affecting performance. Using specific classes from the COIN dataset, they selected 1,000 images for training and testing, yielding an accuracy of 80.27%. The model's efficacy was assessed on low-power hardware, which imposed limitations on processing capacity and necessitated the use of a smaller dataset sample.

In [25], the focus is on scene change detection within videos, using PCA in the context of identifying scene transitions. This involves extracting frames from videos and compiling them into a dataset categorized by types of content, such as journalistic reports and sports, with an accuracy of 89.3%. ResNet50 was deployed for classifying transition and non-transition frames within the training classes.

Lastly [26], presents a framework detailing the use of audio features to differentiate between types of television programming like news, sports, and entertainment. Audio data is converted into spectrograms, visual representations of frequency and time within the audio signal, which then serve as inputs for a Convolutional Neural Network (CNN) trained on Audio Set and tested on a tailored BBC dataset, coupled with a Multilayer Perceptron classifier on the backend. The CNN assesses the likelihood of specific sound events within the recording, achieving a commendable accuracy of approximately 93.7%. However, the spectrogram representation might not capture the entire spectrum of relevant audio information in television programs. Despite the inclusion of broadcasts from various genre categories, there's a possibility that some genres are overrepresented relative to others. In [27], the focus is on classifying violent content in videos using deep neural networks (DNNs) trained on the VSD2014 benchmark, which differentiates between violence and non-violence. The highest accuracy achieved was 53% with a network consisting of 21 hidden layers, implemented on a MacBook Pro. The experimental findings suggest that all the various architectures of hidden layers and nodes explored did not surpass 57% accuracy, warranting further research.

In [28], the authors explore and compare different methodologies for the challenging task of classifying television programs. Logistic Regression emerged as the most effective, boasting an 82% accuracy for newly classified content. This method has proven its merit, particularly in scenarios involving brief documents and a limited number of training samples. The principal limitation identified in the study is that despite certain enhancements, incorporating semantic information from Wikipedia did not significantly improve the accuracy of television program classification. In [29] contribute to the ongoing research discourse, as presented at the the European Conference on Advances in Databases and Information Systems in 2023. It introduces various methodologies for the classification of television programming.

5. Conclusion

In conclusion, this article underscores the pivotal importance of program classification within the ever-evolving landscape of multimedia content. It acknowledges the persistent challenges faced by researchers in this field. Two methods of classification are proposed. The first method integrates the Structural Similarity Index (SSIM) with a custom-designed Convolutional Neural Network (CNN) specifically for overlapping frames while this method is versatile across different systems; it does come with the constraint of needing a predefined sample image size for SSIM comparison. In contrast, the second approach proposes the use of the optical flow to achieve remarkable precision and wide range applicability for various program types. A thoughtful examination of the limitations and the potential future developments of these techniques is carried out. It suggests the adoption of more sophisticated deep learning strategies and the inclusion of additional data sources to increase classification accuracy. Moreover, it proposes that investigating the integration of semantic comprehension could be a compelling direction for future research.

Overall, these promising results indicate opportunities for further enhancement in program classification, a process particularly relevant for television monitoring systems and the sorting of substantial video archives.

The manuscript offers a detailed presentation of the proposed methods and their empirical results. It also highlights the complexities of program classification, considering the variety of formats, genres, and production styles, and the ever-growing volume of daily content production. This underscores the urgent need for developing sophisticated and flexible automated classification techniques to improve the efficiency of television monitoring systems.

Future work should focus on ensuring these methods are seamlessly integrated into the dynamic media environment. A critical goal is to expand the dataset significantly, particularly for national broadcasters.

Additionally, the second proposed method opens an exciting path for specialization. This involves investigating binary classification training with varied weights, an approach that could fine-tune the precision of specific categories during further assessments. A future prospect worth considering is the integration of a Neural Dynamic Classification (NDC) algorithm [30]. This algorithm could be useful for classifying for television programs. With content continuously being updated, program features may vary considerably, whereas the broader categories generally stay more stable. Thus, applying an algorithm like NDC might offer an effective means to manage this variability. The dynamic classification enabled by the NDC algorithm goes beyond just static features. It also considers how these characteristics may change over time or in reaction to certain changes. This is especially relevant when the associations between features and classes are subject to shifts or dynamic influences, as often seen with the evolving nature of television content.

Moreover, it would be wise to evaluate the efficacy of an NDC algorithm specifically for television program classification. Such an approach could provide a flexible and robust solution to the unique challenges posed by the fluid nature of television content and its inherent properties.

Replace “TV” with “television” in the sentence discussing the unique challenges posed by the fluid nature of TV content. (Page 5, Line 768)

Methods like those described in [31] utilize a combination of techniques, including the strategic addition and subtraction of neurons, to optimize the neural network architecture. The aim is to develop a suite of high-performing neural networks that can dynamically and adaptively process complex data. This could be advantageous, particularly with large datasets, such as those encountering in television program classification.

References

[1] Agcom. 2008. Available from: https://wwwagcom.it/documents/10179/539063/Allegato+12-11-2008+13.

[2] Candela F, Morabito FC, Zagaria CF. Television programs classification via deep learning approach using SSMI-CNN. In: Proceedings of the Second International Conference on Applied Intelligence and Informatics (AII 2022). 2022 Sep 1–3; 293-307. Cham, Switzerland. Springer. 2023.

[3] Tran D, Bourdev L, Fergus R, Torresani L, Paluri M. Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. 2015; 4489-4497.

[4] Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems. 2014; 27.

[5] Wang L, Xiong Y, Wang Z, Qiao Y. Temporal segment networks: Towards good practices for deep action recognition. In: Proceedings of the European Conference on Computer Vision (ECCV). 2016.

[6] Carreira J, Zisserman A. Quo vadis, action recognition? A new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017; 4724-4733.

[7] Feichtenhofer C, Fan H, Malik J, He K. SlowFast networks for video recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019; 6202-6211.

[8] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv1409.0473.

[9] Sharma S, Kiros R, Salakhutdinov R. Action recognition using visual attention. arXiv preprint arXiv1511.04119.

[10] Hu W. IEEE transactions on systems, man, and cybernetics – part c: Applications and reviews. 2011; 41(6): 729-743.

[11] Cao Y, Qiu M, Feng W, Li J. Scene-based TV program classification with visual attention mechanism. In: 2019 IEEE International Conference on Multimedia and Expo (ICME). 2019; 640-645. doi: 10.1109/ICME.2019.00230.

[12] Wu F, Zuo L, Chen S, Tang Y. TV program classification with multi-modality features and multi-task learning. In: 2020 IEEE International Conference on Multimedia and Expo (ICME). 2020; 1-6. doi: 10.1109/ICME46284.2020.9102761.

[13] Le HK, Moon S. Automatic TV program genre classification using deep convolutional neural networks. In: Proceedings of the 16th International Conference on Control, Automation, Robotics and Vision (ICARCV). 2019; 133-138. IEEE.

[14] Candela F. SSIM_PROGRAM_CLASSIFICATION. Available from: https://githubcom/itsCandels/SSIM_PROGRAM_CLASSIFICATION-main.

[15] Soomro K, Zamir AR, Shah M. UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv1212.0402.

[16] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016; 770-778.

[17] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing. 2004; 13(4): 600-612.

[18] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25 (NIPS 2012). 2012; 1097-1105.

[19] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. In: Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE. 2009; 248-255.

[20] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in Neural Information Processing Systems. 2017; 30.

[21] Jadhav DA, Sharma Y, Arora PS. Adaptive background subtraction models for shot detection. In: Advances in Signal and Data Processing: Select Proceedings of ICSDP 2019. Springer Singapore. 2021.

[22] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017; 4700-4708.

[23] Diba A, Pazandeh AM, Van Gool L. Efficient two-stream motion and appearance 3D CNNs for video classification. arXiv preprint arXiv1608.08851.

[24] Patil P, Saitwal K, Kamat P, Kulkarni A. Video content classification using deep learning. arXiv preprint arXiv2111.13813.

[25] Chakraborty D, Chiracharit W, Chamnongthai K. Video shot boundary detection using principal component analysis (PCA) and deep learning. In: 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). IEEE. 2021; 272-275.

[26] Pham L, Tran D, Nguyen D, Phan S. An audio-based deep learning framework for BBC television program classification. In: 2021 29th European Signal Processing Conference

(EUSIPCO). IEEE. 2021.

[27] Ali A, Senan N. Violence video classification performance using deep neural networks. In: Recent Advances on Soft Computing and Data Mining: Proceedings of the Third International Conference on Soft Computing and Data Mining (SCDM 2018). Springer International Publishing. 2018; 91-100.

[28] Narducci F, Musto C, Semeraro G, Lops P, de Gemmis M. TV-program retrieval and classification: A comparison of approaches based on machine learning. Information Systems Frontiers. 2018; 20: 1157-1171.

[29] Candela F. Deep learning techniques for television broadcast recognition. In: European Conference on Advances in Databases and Information Systems. Cham: Springer Nature Switzerland. 2023.

[30] Rafiei MH, Adeli H. A new neural dynamic classification algorithm. IEEE Transactions on Neural Networks and Learning Systems. 2017; 28(12): 3074-3083. doi: 10.1109/TNNLS.2017.2682102.

[31] Alam KMR, Siddique N, Adeli H, Rafiei MH, Gauthier L, Takabi D. Self-supervised learning for electroencephalography. IEEE Transactions on Neural Networks and Learning Systems. 2023.