ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10095502

HIERARCHICAL TRANSFORMER FOR MULTI-LABEL TRAILER GENRE CLASSIFICATION

Zihui Cai Hongwei Ding Xuemeng Wu Mohan Xu Xiaohui Cui *

Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education,
School of Cyber Science and Engineering, Wuhan University

ABSTRACT

Determining the genres of a trailer is a challenging multi-label classification task. Previous studies tend to classify by CNN or RNN. Recently, Transformer based on attention mechanism has achieved better results in many research fields than CNN and RNN. Inspired by these, we propose a Hierarchical Transformer (HT). HT can process both the frame sequence (HT-F) and audio (HT-A) of trailers. Besides, a feature compression module is inserted into HT-F, and audio spectrogram segment is processed by HT-A as a whole, which can effectively reduce the data processed by the second Transformer. In order to reduce the training cost and improve the performance, we load the pre-trained weights from other related fields into some parameters of HT, and utilize the limited resources to train the remaining parameters. Experiments show that our best model outperforms state-of-the-art methods on several comprehensive metrics.

Index Terms— multi-label, trailer genre classification, self-attention, Hierarchical Transformer, transfer learning

1. INTRODUCTION

Today, movie is one of the most popular forms of entertainment for audiences. As a fundamental task, movie trailer genre classification can promote the development of movie retrieval and recommendation, further helping users quickly find the movies they are interested in.

Compared with general object classification, classifying movie trailers is more challenging. There are three reasons. Firstly, genre is an immaterial feature that cannot be directly pinpointed in any of the movie trailer [1]. Secondly, a movie may belong to multiple genres, which means that movie classification is a multi-label classification task [2]. Thirdly, the time length of a trailer is usually 1-3 minutes, a little long, and the timing information is so complicated that it is difficult to capture.

Many previous studies [3-7] are based on traditional machine learning. Most of them first compute low-level features, and then feed the features into some simple classifiers, such as decision tree and SVM. Due to lack high-level features and semantics, these methods usually do not achieve excellent classification performance.

With the development of deep learning, many works based on it [1, 2, 8-12] are tried. Among these works, Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are basic architectures. In [8] and [9], CNN is used to extract high-level features from frames, and these features are fed into SVM to classify. Wehrmann and Barros [1, 10] propose the CTT module based on CNN to learn the temporal information of trailers. Alvarez et al. [11] classify trailers by combining low-level descriptors and high-level features extracted by CNN. Yadav et al. [2] propose a sentiment analysis model based on CNN and RNN, which maps a trailer to 6 emotions and classifies according to the distribution of these emotions. Bi et al. [12] utilize pre-trained I3D model [13] to generate video-level features and fuse them by C3D-LSTM.

Recently, Transformer [14] based on attention mechanism is gradually applied to computer vision [15-18], and some achievements are obtained. Inspired by these researches, we design a hierarchical Transformer for trailer genre classification.

Our main contributions are summarized as follows: (i) Propose a convolution-free Hierarchical Transformer (HT) model to predict the genres of movie trailers, which can process data from two modalities including the frame sequence and audio of trailers. (ii) A feature compression module is inserted into HT-F, and audio spectrogram segment is processed by HT-A as a whole, so that the data to be processed by the second Transformer can be effectively reduced. (iii) Transfer pre-trained weights from other related tasks into HT for reducing the training cost and improving classification performance. (iv) Experiments conducted on the LMTD-9 dataset show that our best model outperforms state-of-the-art methods.

2. HIERARCHICAL TRANSFORMER (HT)

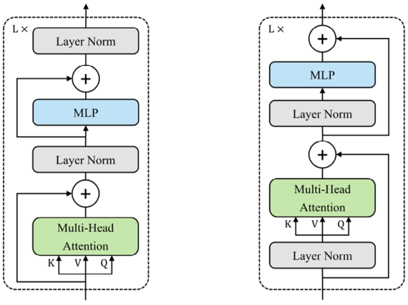
Hierarchical Transformer (HT) can pay attention to the more important information of a trailer and learn the temporal relationship to some extent. Specifically, HT consists of two Transformers, H_1 and H_2. H_1 is utilized to process each token of raw sequence including frame sequence and audio spectrogram segment sequence, for generating a feature sequence. H_2 fuses the feature sequence, so as to predict the movie trailer genres. In HT, Vision Transformer Encoder and Transformer Encoder are the core modules, and position embeddings are added before the sequence is input into Transformer (including H_1 and H_2).

2.1. Encoder modules

Fig. 1 depicts the structure of Encoder modules including Transformer Encoder [14] and Vision Transformer Encoder [15]. In fact, Vision Transformer Encoder is a variant of Transformer Encoder, and they are formed by stacking L identical Encoder layers. Each layer of them contains multi-head self-attention (MSA), multi-layer perceptron (MLP), two residual connections [19] and two layer normalization (LN) operations [20]. However, there are some differences between them. Firstly, post-norm is adopted in Transformer Encoder Layer, whereas pre-norm in Vision Transformer Encoder Layer [21]. Secondly, MLP of Transformer Encoder Lay-

* Corresponding author.

er utilizes ReLU activation function shown in Equation (1) and contains Dropout layers [22] with p = 0.1, while MLP of Vision Transformer Encoder Layer utilizes GELU activation function [23] and does not contain Dropout layers.



(a) Transformer Encoder [14] (b) Vision Transformer Encoder [15]

Fig. 1. Encoder modules.

The core component of whether a given layer is a Transformer Encoder layer or a Vision Transformer Encoder layer lies in the MSA proposed in [14]. Actually, MSA is formed of k SA operations depicted in Equation (2), where $z \in \mathbb{R}^{n \times d}$ is an input sequence, and $\boldsymbol{W_Q}$, $\boldsymbol{W_K}$, $\boldsymbol{W_V}$ are three learnable $d \times d_k$ matrices. d and d_k represent the feature dimension, and $k \times d_k = d$.

$$\mathrm{ReLU}(x) = \max(0, x), \quad \mathrm{sigmoid}(x) = \frac{1}{1+e^{-x}} \tag{1}$$

$$\mathrm{SA}(z) = \mathrm{softmax}\left(\frac{(z\boldsymbol{W_Q}) \cdot (z\boldsymbol{W_k})^T}{\sqrt{d_k}}\right) \cdot (z\boldsymbol{W_v}) \tag{2}$$

2.2. HT for frame sequence (HT-F)

Based on the two Encoder modules in Section 2.1, we design a **H**ierarchical **T**ransformer for processing the **f**rame sequence (HT-F) from trailers as shown in Fig. 2. According to [21], pre-norm is more efficient for training than post-norm if the model goes deeper. We set the weights of H1 by transfer learning, which let H1 go deeper, so Vision Transformer Encoder is utilized to build H1. The weights of H2 are obtained by training from scratch and train data is limited, which makes H2 go shallow, so Transformer Encoder is utilized to build H2.

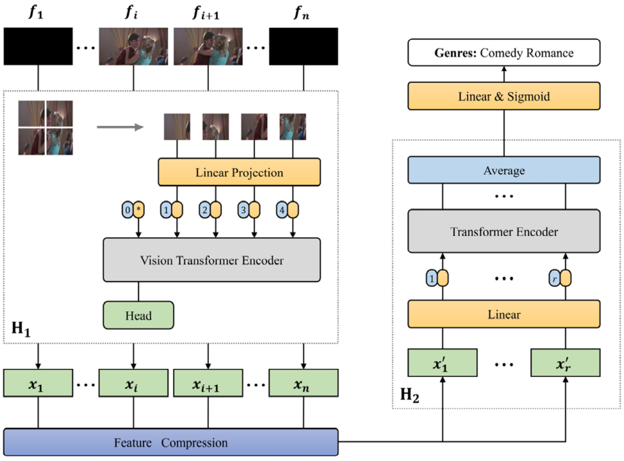


Fig. 2. Hierarchical Transformer for video frames (HT-F).

As the first Transformer, H1 is in charge of extracting the features of each frame. It first cuts a frame into multiple non-overlapping small patches, and then flatten the patches for mapping to a 1D sequence of token embeddings. We prepend a learnable embedding to the sequence, and add standard learnable 1D position embedding to each token of the sequence. Next, the sequence is input to Vision Transformer Encoder for generating an output sequence, the first token of which is treated as frame representation.

In order to look natural and realistic, many adjacent frames in a video are usually extremely similar or even identical. Considering this, we sample a frame every 12 frames, so as to obtain a frame sequence $\boldsymbol{F} = [f_1, f_2, ..., f_n]$. However, $\boldsymbol{X} = [x_1, x_2, ..., x_n]$ generated by H1 contains many similar features because there are still many similar frames are also preserved in $\boldsymbol{F}$. In order to combine these similar features and reduce data redundancy, we insert a feature compression module shown in Algorithm 1 between H1 and H2. In this algorithm, a fixed threshold t and cosine similarity between those adjacent features are utilized to judge whether these features need to be combined by max pooling. Finally, a new feature sequence $\boldsymbol{X'} = [x'_1, x'_2, ..., x'_r]$ is generated, and the sequence length is reduced from n to r.

Algorithm 1: Features Compression

Input: $\boldsymbol{X} = [x_1, x_2, ..., x_n]$, a fixed threshold t

Output: $\boldsymbol{X'} = [x'_1, x'_2, ..., x'_r]$, where $r < n$

```
$j = 1, x'_1 = x_1$
for $i = 2$ to $n$ :
    $s = (x_i^T \cdot x'_j) / (\|x_i\| \cdot \|x'_j\|)$    # Calculate cosine similarity
    if $s > t$ :                                            # Compare with threshold
        $x'_j = \mathrm{maxpool}([x_i; x'_j])$              # Combine features
    else:
        $j = j + 1$                                         # Start next combination
        $x'_j = x_i$
```

And then, $\boldsymbol{X'}$ is input into H2. To reduce computing cost, we add a linear layer at the beginning of H2 to reduce the dimension of each token in $\boldsymbol{X'}$. And then, its output is fed into Transformer Encoder after adding position embeddings, which generate a feature sequence of length r. A vector is the final output of H2 by averaging the sequence. Next, the vector is fully connected to c neurons, which correspond to c genres. To get the predicted probability that a trailer belongs to each of these c genres, sigmoid function described in Equation (1) is utilized for activation. Since different trailers are not the same in duration, the length of the obtained feature sequence $\boldsymbol{X'}$ varies. Therefore, we utilize fixed position embeddings proposed in [14], which is shown in Equation (3), where $P_{i,2j}$ and $P_{i,2j+1}$ represent the elements of even and odd positions in the i-th position embedding, respectively.

$$P_{i,2j} = \sin\left(\frac{i}{10000^{2j/d}}\right), \; P_{i,2j+1} = \cos\left(\frac{i}{10000^{2j/d}}\right), \; i = 1, 2, ..., r \tag{3}$$

2.3. HT for audio (HT-A)

Fig. 3 describes the **H**ierarchical **T**ransformer for processing **a**udio fbank features (HT-A). Similar to HT-F, it contains two Transformers, H_1 and H_2, but does not contain the feature compression module.

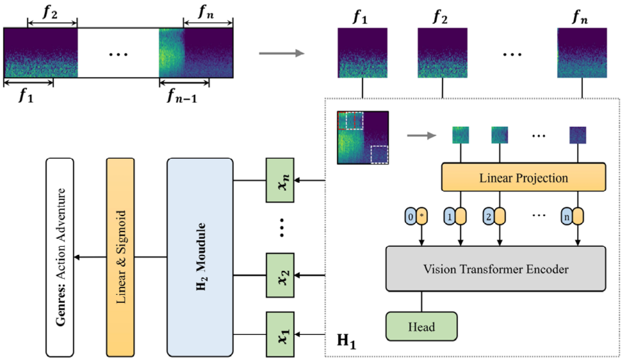


Fig. 3. Hierarchical Transformer for audio (HT-A).

Firstly, we convert a trailer audio into a sequence of 128-dimensional log Mel filterbank (fbank) features computed with a 25ms Hamming window every 10ms, which results in $100t \times 128$ spectrogram for subsequent processing, where t is the duration of audio. If each 10ms feature is input into H_1, H_2 will need to process a sequence up to $100t$, which is difficult to calculate for Transformer with a quadratic complexity. With this in mind, we allocate some work to H_1, making it process 1024 consecutive 10ms instead of single 10ms. Therefore, we cut the spectrogram into multiple segments, and each segment corresponds to 10.24 seconds of the audio. Considering the coherence, we adopt the overlapping cutting, which makes the tail of the i-th segment f_i overlaps with the head of the i+1-th segment f_{i+1}. Finally, a spectrogram segment sequence $\boldsymbol{F} = [f_1, f_2, ..., f_n]$ are obtained.

In H_1, an overlapping method is utilized to cut the spectrogram segment, which means that there are overlapping parts between some patches. Since H_2 only focuses on processing the feature sequence and does not care about the raw information, its structure is exactly the same as it in HT-F. Finally, the output of H_2 is fully connected to c neurons and sigmoid function depicted in Equation (1) is utilized for activation, so as to get the predicted probability.

2.4. Combine HT-F and HT-A (HT-FA)

The frames and audio from a trailer are two different modalities, and there is usually information complementarity between different modalities [24]. Therefore, we combine HT-F and HT-A, creating a model taking both frames and audio into account, HT-FA. Similar to [10], We train HT-F and HT-A independently, and then feed the frame sequence and audio from the same trailer into them independently to get predicted probabilities P_f and P_a, as shown in Equation (4). Finally, we utilize a simple weighted average shown in Equation (5) to get the final predicted probability P_{fa} that is the output of HT-FA. We set the value of w to each of {0.00, 0.01, 0.02, …, 0.99, 1.00} in turn, and analyze the results on validation dataset. According to the results, we choose 0.76, which means that 76% of the predicted probability in HT-FA is derived from HT-F, and 24% is derived from HT-A.

$$P_f = \text{HT-F}(X_f), \quad P_a = \text{HT-A}(X_a), \quad P_f \text{ and } P_a \in [0,1]^c \tag{4}$$

$$P_{fa} = w \cdot P_f + (1-w) \cdot P_a, \quad w \in [0,1] \tag{5}$$

3. EXPERIMENTS

3.1. Dataset and Evaluation Metrics

We do experiments on LMTD-9 dataset [18]. LMTD-9 is a publicly available movie trailer dataset containing 9 movie genres (action, adventure, comedy, crime, drama, horror, romance, sci-fi, and thriller) and over 4000 movie trailers.

Similar to previous works [1, 10, 11, 12], we use the area under a precision-recall curve (AU(PRC)) and Ranking Loss [25] to evaluate our models. Since trailer genre classification is a multi-label classification task, we need to average the AU(PRC)s from all genres. Based on different averaging methods, there are three different metrics, respectively AU(PRC)$_i$ (micro average), AU(PRC)$_a$ (macro average) and AU(PRC)$_w$ (weighted average).

3.2. Baselines

Five competitive baseline algorithms including CTT-MMC-C [1], CTT-MMC-S [10], CTT-MMC-TN [10], DF [11] and VRFN2-64-10 [12] are utilized to compare with HT models. Among them, CTT-MMC-C, CTT-MMC-S, and CTT-MMC-TN are based on CTT module [1], DF combines low-level visual descriptors and high-level frame features, and VRFN2-64-10 is based on 3D CNN. In terms of classification performance, DF excels in some genres, while CTT-MMC-TN and VRFN2-64-10 achieve the best comprehensive metrics previously.

3.3. Implementation Details

We design H_1 of HT-F as the ViT-H/14 model in [15] with the pre-trained weights on ImageNet-1K [26] and SWAG [27], and design H_1 of HT-A as the AST model (overlap size = 6) [28] with the pre-trained weights on AudioSet [29]. In H_2, we reduce the output features of H_1 to 160 dimensions by a linear transformation, and then feed them into a Transformer Encoder Module with 8 heads and 6 layers. According to previous experience, we set the hidden size of MLP to 4d.

$$\ell(y, \hat{y}) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{c}\left[y_{i,j}\log(\hat{y}_{i,j}) + (1-y_{i,j})\log(1-\hat{y}_{i,j})\right] \tag{6}$$

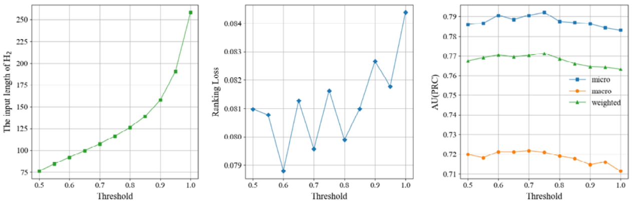
During the training phase, the parameters of H_1 are frozen, and H_2 are learned from scratch by optimizing the binary cross-entropy loss function shown in Equation (6). For HT-F, we train 80 epochs with batch size of 8, learning rate of 1×10^{-3} and weight decay of 1×10^{-3}. For HT-A, we train 50 epochs with batch size of 16, learning rate of 5×10^{-3} and weight decay of 1×10^{-3}. Finally, we select the model with the maximum sum of three AU(PRC)s on the validation dataset among these results.

3.4. Results

Fig. 4 depicts the impact of different thresholds in the feature compression module on the feature compression ratio and metrics. According to it, HT-F with compression mechanism reduces the input length of H_2, and makes the Ranking Loss lower and AU(PRC)s slight higher than without it (threshold = 1.0), which demonstrates that the compression module can reduce redundant features without compromising performance. After comprehensively considering, we choose the model with a threshold of 0.6 as our final HT-F model.

Table 1. Results comparing HT with five competitive baselines.

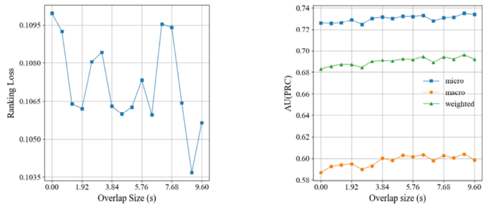
	CTT-MMC -C [1]	CTT-MMC -S [10]	CTT-MMC -TN [10]	DF [11]	VRFN2-64 -10 [12]	HT-A	HT-F	HT-FA
Action	0.813	0.669	0.835	**0.852**	0.79	0.764	0.834	0.849
Adventure	0.720	0.397	0.672	0.752	0.71	0.658	0.762	**0.797**
Comedy	0.853	0.834	0.870	0.871	0.89	0.904	0.892	**0.915**
Crime	0.505	0.339	0.547	0.628	0.52	0.442	**0.651**	0.642
Drama	0.791	0.764	0.841	0.641	0.83	0.816	0.842	**0.853**
Horror	0.609	0.390	0.667	0.424	0.68	0.728	0.806	**0.826**
Romance	0.432	0.362	0.456	0.468	0.46	0.377	0.590	**0.594**
SciFi	0.398	0.226	0.401	0.192	0.44	0.289	**0.577**	0.572
Thriller	0.496	0.386	0.522	0.520	0.49	0.447	0.538	**0.560**
Ranking Loss	0.108	0.183	0.099	-	-	0.106	0.079	**0.070**
AU(PRC)$_i$	0.722	0.642	0.742	-	0.747	0.732	0.790	**0.807**
AU(PRC)$_a$	0.624	0.485	0.646	0.594	0.645	0.603	0.721	**0.734**
AU(PRC)$_w$	0.697	0.599	0.724	0.665	0.721	0.692	0.770	**0.785**



(a) The input length of H_2 - Threshold (b) Ranking Loss - Threshold (c) AU(PRC) - Threshold

Fig. 4. Impact of different thresholds in the feature compression module of HT-F for (a) The input length of H_2. (b) Ranking Loss. (c) three AU(PRC)s.

Fig. 5 depicts the impact of the overlap size when cutting the spectrogram on the performance of HT-A. It shows that overlapping cuts generally has lower Ranking Loss and higher AU(PRC) than non-overlapping cuts (Overlap Size = 0), which demonstrates that overlapping cutting can improve the performance of HT-A to some extent. After comprehensively considering, we choose the model with overlap size equal to 5.12 seconds as our final HT-A model.



(a) Ranking Loss – Overlap size (b) AU(PRC) – Overlap size

Fig. 5. Impact of the Overlap Size of spectrogram for (a) Ranking Loss. (b) three AU(PRC)s.

Table 1 presents per-genre AU(PRC) and comprehensive results including three AU(PRC)s and Ranking Loss. Both CTT-MMC-S and HT-A are audio-based models, and HT-A outperforms CTT-MMC-S on all metrics. Compared with all baselines, HT-F also achieves an outstanding classification performance. HT-FA has the highest AU(PRC) on 6 genres, and the AU(PRC) of other genres including Action, Crime and SciFi is second only to the best. To be specific, the AU(PRC) values of Horror, Romance and SciFi are improved over 10% compared with baseline algorithms. Moreover, the improvement of Horror is nearly 15%. Comprehensively, HT-FA outperforms the best baseline by 0.029 reduction in Ranking Loss, 6.0% improvement in AU(PRC)$_i$, 8.8% improvement in AU(PRC)$_a$ and 6.1% improvement in AU(PRC)$_w$. Besides, HT-FA outperforms HT-F and HT-A on 4 evaluation metrics. In other words, HT-FA is the best among these 7 models.

Such improvements are the results of both our attention-based hierarchical Transformer structure and weights transfer from other tasks. Besides, our feature compression module and overlapping cutting for the spectrogram also play an important role.

4. CONCLUSION AND FUTURE WORK

In this paper, we propose a HT framework formed of two Transformers (H_1 and H_2) for multi-label trailer genre classification, including HT-F, HT-A and HT-FA models. H_1 is utilized to process each token of raw sequence including frame sequence and audio spectrogram segment sequence, for generating a feature sequence. And H_2 fuses the feature sequence. We insert a feature compression module into HT-F for reducing redundant features, and obtain the weights of H_1 by transfer learning, which reduces the training cost and improve the performance. The experiments show that our best model (HT-FA) achieves improvements on several comprehensive metrics compared with state-of-the-art methods.

In the future, we will consider more other complex movie genres or try to apply the HT framework to other video classification tasks and audio classification tasks.

5. ACKNOWLEDGEMENTS

The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University. This research was supported by the AI-based infrared target detection and recognition program of Yunnan Province Science and Technology Department (No.2022KZ00125), and Wuhan University Specific Fund for Major School-level International Initiatives.

6. REFERENCES

[1] J. Wehrmann and R. C. Barros, “Convolutions through time for multi-label movie genre classification,” in Proceedings of the Symposium on Applied Computing, 2017, pp. 114–119.

[2] A. Yadav and D. K. Vishwakarma, “A unified framework of deep networks for genre classification using movie trailer,” *Applied Soft Computing*, vol. 96, p. 106624, 2020.

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

[4] H.-Y. Huang, W.-S. Shih, and W.-H. Hsu, “A film classifier based on low-level visual features,” in *2007 IEEE 9th workshop on multimedia signal processing*, 2007, pp. 465–468.

[5] S. K. Jain and R. Jadon, “Movies genres classifier using neural network,” in *2009 24th International Symposium on Computer and Information Sciences*, 2009, pp. 575–580.

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

[7] Y.-F. Huang and S.-H. Wang, “Movie genre classification using svm with audio and video features,” in *International Conference on Active Media Technology*, 2012, pp. 1–10.

[8] G. S. Simões, J. Wehrmann, R. C. Barros, and D. D. Ruiz, “Movie genre classification with convolutional neural networks,” in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 259–266.

[9] J. Wehrmann, R. C. Barros, G. S. Simões, T. S. Paula, and D. D. Ruiz, “(Deep) learning from frames,” in *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, 2016, pp. 1–6.

[10] J. Wehrmann and R. C. Barros, “Movie genre classification: A multi-label approach based on convolutions through time,” *Applied Soft Computing*, vol. 61, pp. 973–982, 2017.

[11] F. Alvarez, F. Sanchez, G. Hernandez-Peñaloza, D. Jimenez, J. M. Menéndez, and G. Cisneros, “On the influence of low-level visual features in film classification,” *PloS one*, vol. 14, no. 2, p. e0211406, 2019.

[12] T. Bi, D. Jarnikov, and J. Lukkien, “Video representation fusion network for multi-label movie genre classification,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 9386–9391.

[13] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[14] A. Vaswani et al., “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[15] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[16] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10012–10022.

[17] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, “Vivit: A video vision transformer,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6836–6846.

[18] Z. Liu et al., “Video swin transformer,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3202–3211.

[19] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[21] Q. Wang et al., “Learning deep transformer models for machine translation,” *arXiv preprint arXiv:1906.01787*, 2019.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] D. Hendrycks and K. Gimpel, “Gaussian error linear units (gelus),” *arXiv preprint arXiv:1606.08415*, 2016.

[24] D. Ramachandram and G. W. Taylor, “Deep multimodal learning: A survey on recent advances and trends,” *IEEE signal processing magazine*, vol. 34, no. 6, pp. 96–108, 2017.

[25] G. Tsoumakas, I. Katakis, and I. Vlahavas, “Mining multi-label data,” *Data mining and knowledge discovery handbook*, pp. 667–685, 2009.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[27] M. Singh et al., “Revisiting Weakly Supervised Pre-Training of Visual Perception Models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 804–814.

[28] Y. Gong, Y.-A. Chung, and J. Glass, “Ast: Audio spectrogram transformer,” *arXiv preprint arXiv:2104.01778*, 2021.

[29] J. F. Gemmeke et al., “Audio set: An ontology and human-labeled dataset for audio events,” in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2017, pp. 776–780.