International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 4
ISSN: 2088-5334

Coefficient Prediction for Physically-based Cloth Simulation Using Deep learning

Makara Mao [a], Hongly Va [a], Min Hong [b,*]

[a] Department of Software Convergence, Soonchunhyang University, Asan-si, Republic of Korea
[b] Department of Computer Software Engineering, Soonchunhyang University, Asan-si, Republic of Korea
Corresponding author: *mhong@sch.ac.kr

***Abstract*— Physically-based cloth simulation involves modeling cloth as a collection of particles or nodes connected by various types of constraints. These particles interact with each other and the environment, such as gravity or collisions, to accurately simulate the cloth's behavior. One essential component of such simulations is the set of material parameters or coefficients that dictate the cloth's physical properties, such as stiffness and damping. Deep learning-based coefficient prediction in physically-based cloth simulation involves using machine learning techniques, specifically deep neural networks, to predict the material parameters of cloth from its geometric and physical properties. The deep learning model is trained using a dataset of simulated cloth instances, where the material parameters are known. The input to the model is a set of geometric and physical properties of the cloth, such as the dimensions, orientation, and velocity. The output of the model is the set of material parameters that best represent the cloth's behavior under these conditions. This paper proposes a deep learning method for predicting these coefficients using a multi-label video classification approach. The training data is generated from a physics-based simulator, and the method is evaluated on some cloth simulations, such as fabric falling down, fabric with collision, and fabric affected by airflow. The cloth movement dataset is generated from a mass-spring-based simulation. The results show that the transformer model has much higher accuracy than other models. This study provides a promising approach for predicting the coefficients of virtual cloth in physically-based simulations.**

***Keywords*— Deep learning; computer vision; cloth simulation; transformer; LSTM; GRU.**

Manuscript received 12 Dec. 2022; revised 28 Apr. 2023; accepted 15 May 2023. Date of publication 31 Aug. 2023.

I. INTRODUCTION

In recent years, the explosive growth of the metaverse has highlighted the critical role of accurate cloth simulation in creating immersive virtual environments. The metaverse, a virtual shared space increasingly popular in gaming, social media, and other online platforms, allows users to create avatars, explore virtual environments, and interact with other users in real-time [1], [2].

Accurately predicting the coefficient values of the mass-spring system for different fabrics can aid in creating more realistic clothing and avatars that accurately reflect the physical properties of real-world fabrics, including their draping, wrinkling, and movement with the user. This is particularly crucial in the metaverse, where creating realistic avatars and clothing customization options is essential for providing users with an immersive and engaging experience [3], [4].

Simulating the behavior of cloth accurately is a complex and dynamic process that has become increasingly important in computer graphics, particularly in 3D animation, virtual reality, and the metaverse. The mass-spring system is often used to model the behavior of cloth, representing the cloth as a set of interconnected particles connected by springs. The system's behavior is governed by the physical properties of the cloth, including its mass, elasticity, and damping coefficients [5], [6].

Recent advancements in machine learning have provided a new tool for accurately predicting the coefficient values of the mass-spring system. By training machine learning algorithms on large datasets of fabric videos to analyze the motion of the fabric and predict its physical properties, this approach has shown promising results in accurately predicting the physical properties of a range of fabrics, including cotton, silk, and synthetic materials [7], [8]. Another approach to accurately predicting the physical properties of fabrics is to use physical experiments and simulations. This method can be particularly useful when dealing with more complex fabrics that exhibit non-linear or anisotropic properties [9].

Accurately simulating the behavior of cloth in 3D animation can be used to create lifelike and dynamic virtual environments by modeling the physical behavior of cloth. This involves using a combination of physics-based modeling [10] and computer graphics techniques [11], such as algorithms to simulate the behavior of cloth and texturing and shading to create the appearance of different types of fabric [12]. By accurately simulating the behavior of cloth, it is possible to create more lifelike animations that accurately depict the way fabric moves and interacts with other objects in a scene [13], [14].

The accurate prediction of the coefficient values of the mass spring system has numerous potential applications in 3D animation, virtual reality, and the metaverse. For example, it can be used to create more realistic clothing for characters in video games, films, and other types of media, as well as more dynamic and interactive virtual environments. The ability to simulate cloth behavior accurately is essential for creating realistic avatars and clothing customization options, enhancing the overall user experience [15], [16]. Furthermore, accurate cloth simulation can enable new forms of social interaction in virtual spaces, such as virtual fashion shows or other events where cloth behavior plays a significant role [17].

Mass spring systems are a widely used approach to simulate the behavior of cloth in computer graphics. The system consists of a grid of points connected to its neighboring points by springs. The spring coefficients, also known as stiffness constants, are used to model the physical properties of the cloth, such as its elasticity and rigidity. The mass of each point is also considered, as it affects the motion of the cloth under external forces such as gravity and wind [18].

However, tuning the parameters of the mass spring system to produce realistic simulations can be challenging. This is because the optimal values for the stiffness constants and other parameters can vary depending on the physical properties of the cloth and the external factors acting on it. Adjusting these parameters manually can be time-consuming and often requires extensive trial and error [19].

Fortunately, deep learning techniques can be used to automate this process by predicting the optimal values of the mass-spring system coefficients based on the physical properties of the cloth and external factors. This involves training a neural network to recognize patterns and relationships between the input data and the desired output, which is the optimal coefficient value for the mass-spring system [20].

The neural network itself will be designed to take as input the physical properties of the cloth, such as its density, elasticity, and other material properties, as well as any external factors that might influence its behavior, such as gravity, wind, or collisions with other objects. It will then output a set of coefficients for the mass spring system that are optimized for simulating the cloth under these conditions [21].

Once the neural network has been trained, it can be used to predict the optimal coefficient values for new simulations based on the input data. This can greatly simplify the process of tuning the mass spring system for different cloth materials and simulation scenarios, allowing designers to create more realistic and accurate cloth simulations in less time [22]. The use of deep learning techniques to predict the coefficient value of the mass-spring system for cloth simulation has shown promising results. By training a neural network on a large dataset of cloth properties and external factors, it is possible to accurately predict the behavior of cloth in a variety of different scenarios [23].

This paper focuses specifically on fabric materials and presents a specific application that involves capturing fabric from multiple camera angles and extracting frames from videos. The study aims to demonstrate how predictive models can be used on input video data to create an advanced try-on system. The paper includes a comprehensive review of related works in Section 2, addressing issues in the fields of the metaverse, 3D animation, and virtual reality for video prediction. Section 3 outlines the proposed methodology and presents experimental results. Finally, in Section 4, the paper provides conclusions and discusses future work.

II. Materials and Methods

Cloth simulation is an important research area in computer graphics with numerous applications across 3D animation, virtual and augmented reality, and the emerging metaverse. Accurately simulating cloth behavior in virtual environments is challenging, and researchers have developed various techniques, including mass-spring models, finite element methods, particle-based methods, and machine-learning techniques. Despite these advances, many challenges exist, such as simulating cloth at different scales and incorporating real-time interactions. Continued research in cloth simulation is essential for creating more realistic and immersive virtual worlds.

The paper approach by Feng et al. utilizes a deep learning framework to simulate the interaction between clothing and the human body through cloth simulation. It employs a two-stream network architecture that takes the body mesh and clothing mesh as input and predicts the deformation of the clothing mesh resulting from the interaction with the body. The network is trained on a vast dataset of human and clothing meshes with ground truth data on the deformation of the clothing meshes [24].

In recent years, cloth simulation has seen significant advancements, including techniques such as finite element methods [25], which divide the cloth into small elements and solve equations of motion for each element. More sophisticated models that incorporate both stretching and bending stiffness have also been explored for more realistic simulations of cloth behavior [26], [27].

Others have investigated more sophisticated models that incorporate stretching and bending stiffness for more realistic simulations of cloth behavior [26], [27]. Other techniques include particle-based methods that model the cloth as a collection of particles connected by springs and hybrid approaches that combine different techniques to take advantage of their strengths.

Other cloth simulation techniques have also been proposed, such as particle-based methods using particles connected by springs and hybrid approaches combining different methods. For instance, Zhang et al. [28] proposed particle-based methods that utilize connected particles and springs to simulate cloth behavior. Moreover, mass-spring systems, position-based dynamics, and dynamic relaxation can provide a more comprehensive overview of cloth simulation methods. By evaluating various techniques, researchers can better understand their strengths and limitations and make informed choices for specific applications.

Chekir [29] has also introduced a log-Euclidean Fisher vector end-to-end learning approach for video classification, involving analyzing large-scale video data and extracting valuable information from it. Other deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have also been successful in various applications, such as action recognition, object detection, and video summarization.

In addition to progress in cloth simulation techniques, computer graphics have seen significant development in recent years, particularly in the area of realistic rendering techniques that accurately capture the behavior of light in a scene. Researchers, including Pueyo et al. [30], have introduced techniques such as ray tracing, which models the path of light as it interacts with objects in a scene, and radiosity, which simulates how light is diffused and reflected between surfaces. These techniques have found applications in industries such as film, video games, and architecture, enabling more realistic and accurate renderings of virtual environments.

Another important area of research in computer graphics is the development of techniques for modeling and animating complex shapes and materials, such as fabric, hair, and other objects. Researchers have developed a range of techniques that have found applications in different fields, including film, video game development, medical imaging, and scientific visualization, enabling more realistic and visually appealing results [31].

In the context of fabric study materials, Vişan et al. [32] have also explored various techniques for simulating the behavior of fabric. One popular approach is finite element methods, which involve dividing the fabric into small elements and solving equations of motion for each element. Other techniques include particle-based methods, such as mass-spring systems, and optimization-based approaches that minimize the energy of the cloth system while ensuring certain constraints are met.

Overall, cloth simulation and computer graphics have seen significant progress in recent years, driven by advances in computing power and the development of new techniques and algorithms. As the field continues to evolve, we will likely see even more sophisticated and realistic simulations of cloth and other materials, with applications in a wide range of industries.

III. Results and Discussion

A. Method Overview

This section presents an overview of the methods used in this paper and their results. We explored three models: Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and Transformer. The GRU model is particularly useful for sequence modeling tasks, including natural language processing, speech recognition, video classification, and image captioning. The key feature of GRU is its ability to selectively update its hidden state using "gates", which are learned parameters that determine how much of the previous hidden state is used.

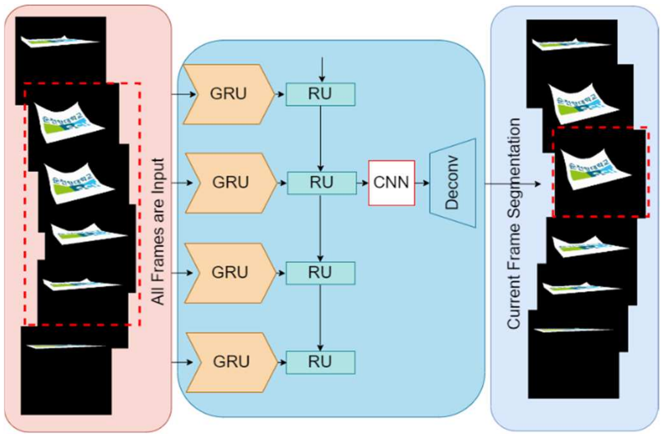

Fig. 1 The overview of the proposed Gated Recurrent Units (GRUs) method and how it is visualized through unrolling the recurrent part of the network.

The working flow of a GRU model can be portioned into various stages. To start with, the input data to the GRU is a sequence of vectors, like frames in a video. Next, an embedding layer is used to convert the input vectors to a fixed-size vector representation. At the beginning of the sequence, the hidden state of the GRU is initialized to a fixed-size vector of zeros. For each input sequence element, the GRU computes a new hidden state by combining the current input vector with the previous hidden state.

This computation involves several steps, including the use of sigmoid functions to determine how much of the previous hidden state should be preserved or forgotten, as well as the generation of a candidate hidden state through the application of a hyperbolic tangent function. The candidate hidden state is then combined with the output of the update gate to generate the new hidden state. Finally, the output of the GRU is typically the hidden state corresponding to the last element of the input sequence, which can be used for downstream tasks such as classification or generation.

Long Short-Term Memory (LSTM) is another type of recurrent neural network (RNN) that can be used for video classification tasks, similar to GRU models. The working flow of an LSTM model for video classification can be described as follows:

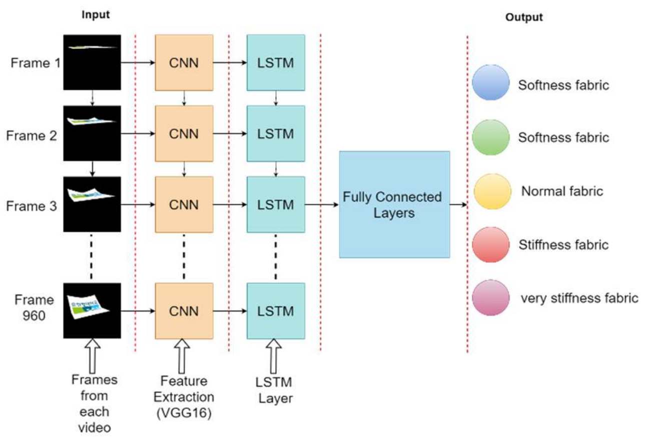

Fig. 2 The system architecture consists of a CNN VGG16 model to extract high-level features from video frames. These features are then fed into a LSTM model that classify classes.

The process of using LSTM for video analysis involves several steps. Firstly, the input data consists of a sequence of video frames, each represented by a 3D tensor of pixel values. Secondly, preprocessing may be required to normalize size, color, and brightness and extract relevant features using techniques such as optical flow analysis or convolution neural networks (CNNs). Thirdly, the processed frames are passed through a CNN-based feature extractor that outputs a fixed-size feature vector for each frame.

Fourthly, a time-distributed layer applies the same set of weights to each frame in the sequence, allowing the model to learn different weights for different time steps. The feature vectors are then fed into an LSTM model, which updates its memory cell and hidden state at each time step using the input feature vector, previous memory state, and the outputs of three gates (input, forget, and output).

The final output of the LSTM model is typically a probability distribution over different video categories, obtained by passing the last hidden state through a dense layer with soft-max activation. Lastly, the model is trained using a labeled dataset of video sequences, with the loss calculated using cross-entropy between the predicted and true labels.

A transformer is a powerful neural network architecture that was originally proposed for natural language processing tasks but has also been adapted for computer vision tasks, including video classification. The working flow of a transformer model for video classification can be described as follows:

The process of using a transformer for video analysis involves several steps. Firstly, the input data consists of a sequence of video frames, where a 3D tensor of pixel values represents each frame. Secondly, the video frames may need to be preprocessed to normalize their size, color, or brightness and extract relevant features using optical flow analysis or CNNs. Thirdly, the preprocessed frames are fed into a CNN-based feature extractor that outputs a fixed-size feature vector for each frame.

Fourthly, the feature vectors are combined with a sequence of positional encoding vectors that provide information about the position of each frame in the sequence. Fifthly, the combined feature and positional encoding vectors are fed into a stack of transformer layers, performing multi-head self-attention and feedforward computations. The multi-head self-attention operation allows each frame to attend to all other frames in the sequence, weighted by their relevance and similarity, while the feedforward operation applies a set of fully connected layers to each frame independently, allowing the model to learn non-linear relationships between features.

The final output of the transformer model is typically a probability distribution over different video categories, which can be obtained by passing the output of the last transformer layer through a dense layer with soft-max activation. Lastly, the model is trained using a labeled dataset of video sequences, with the loss calculated using cross-entropy between the predicted and true labels.

The transformer model is better than GRU and LSTM models in terms of accuracy and execution time due to some key reasons. This is because of some important reasons. The transformer model utilizes a self-attention mechanism that allows it to attend to all positions in the input sequence simultaneously, resulting in more effective long-range dependency capture compared to GRU and LSTM models, which only attend to previous positions. Furthermore, the self-attention mechanism is highly parallelizable, resulting in faster input sequence processing times compared to other models. Another advantage of the transformer model is its lack of recurrent connections, which makes it easier to train and less susceptible to vanishing gradients.

The transformer model also utilizes positional encoding to provide crucial information about the input sequence's order, which is essential in certain tasks. And another one, the Transformer model, is more parameter efficient than GRU and LSTM models, which require fewer parameters to achieve similar accuracy. Transformer model is an appealing option for various applications, especially in resource-constrained settings.

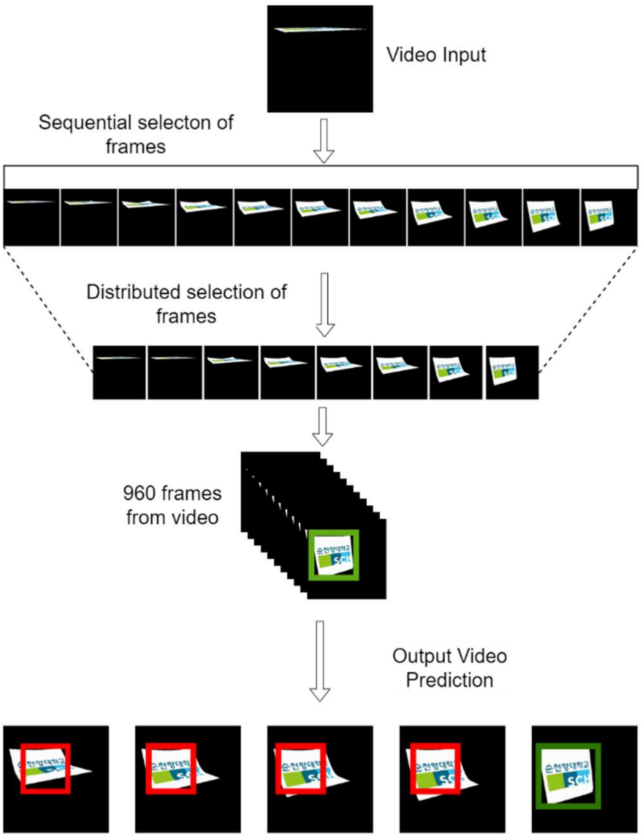

Fig. 3 CNN-Transformer architecture for video classification.

B. *The Dataset for Experiments*

This study explores the classification of fabrics generated from computer simulations, specifically fabric falling down, fabric with collision, and fabric affected by airflow. Depth images of these fabrics were used as input for a deep learning model to predict their fabric class. The fabric classes were divided into five categories: softness fabric, very softness fabric, normal fabric, stiffness fabric, and very stiffness fabric. Each category contained approximately 80 videos, with each video having a duration of 30 seconds and consisting of 960 frames. In total, 1,330 videos were used in this study. The video description is shown in Table 1. The videos in this dataset depict fabric in free fall, exhibiting unique movements corresponding to each coefficient as it resists the pull of gravity.

TABLE I
DATASET FOR EXPERIMENT IN VIDEO CLASSIFICATION

Dataset	Length (seconds)	Video	Class	Frame
Fabric falling down	30	450	5	432,000
Fabric with collision	30	415	5	398,400
Fabric affected by airflow	30	465	5	446,400

Fig. 4 The video animation depicts the movement of fabric falling down in the video clip.

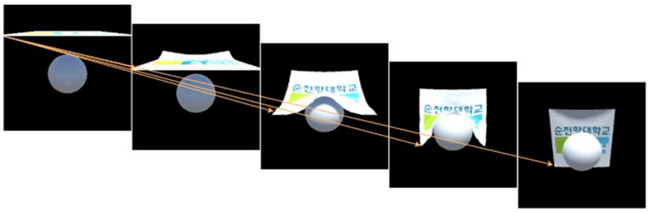

Fig. 5 The video animation depicts the movement of fabric with a collision in the video clip.

Fig. 6 The video animation depicts the movement of fabric affected by airflow in the video clip.

However, as the dataset is generated through a computer simulation, it poses a challenge since many videos feature identical objects and camera angles, even though the fabric motion varies in direction (e.g., Fig. 4, Fig. 5, Fig. 6). Fig. 7 shows the simulation video of virtual cloths which are tested with mass-spring system. For the very softness fabric, we assigned the coefficients for mass-spring system as 200 for Ks value, 100 for Kd vakue.

The mass-spring system (MSS) is a commonly used technique in computer graphics for simulating cloth and other deformable objects. In a MSS, the cloth is represented as a collection of particles connected by springs. Each particle has a mass, and the springs between the particles have spring constants (Ks) and damping coefficients (Kd).

The Ks determines the stiffness of the spring, and the Kd controls how quickly the spring's energy dissipates. These parameters are important because they affect the behavior of the cloth simulation. If the Ks value is too high, the cloth will be too stiff and not move realistically. On the other hand, if the Ks value is too low, the cloth will be too soft and may appear to collapse. Similarly, if the Kd value is too high, the cloth will move too quickly and appear unstable, while if the Kd value is too low, the cloth may oscillate excessively and not settle.

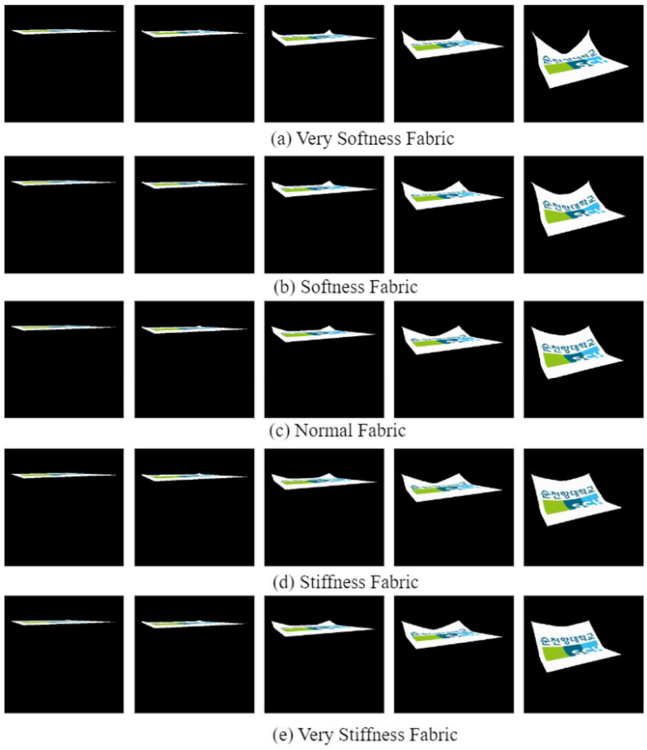

Fig. 7 Displays frames of the agent interacting with the fabric in five challenging tasks where it learns to manipulate specific verticals to their target positions.

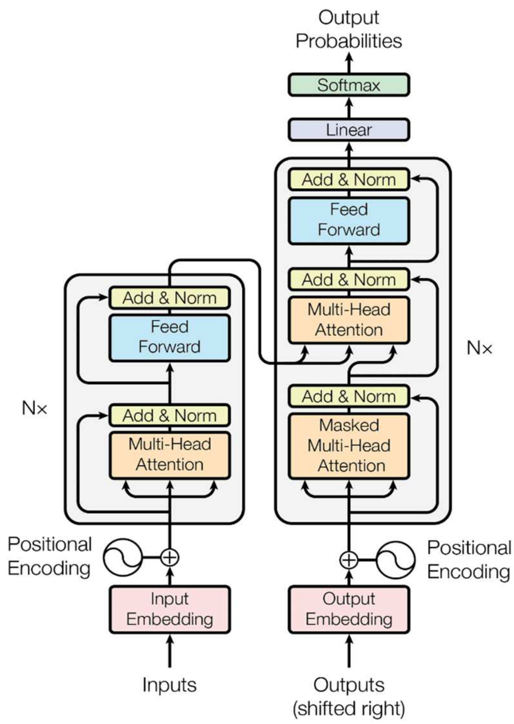

Fig. 8 The proposed transformer architecture follows an encoder-decoder structure [33].

The first layer, "PositionalEmbedding", is responsible for adding positional information to the input embeddings of the transformer. The positional information is necessary for the Transformer model to operate on sequences of inputs since the model has no inherent notion of sequence order. The layer achieves this by creating an "Embedding" layer with "sequence_length" as the input dimension and "output_dim" as the output dimension. The output dimension of the Embedding layer is equal to the embedding dimension of the input tensor, which is typically a fixed value for a given model. The layer then generates a sequence of position embeddings using the "range" function and adds them to the input tensor. The resulting tensor has the same shape as the input tensor, with each positional embedding element added to the input tensor's corresponding element.

The second layer, "TransformerEncoder", is one of the building blocks of the Transformer model. It consists of multi-head self-attention, a feedforward network, and residual connections with layer normalization. The layer takes an input tensor and an optional mask as input and applies the attention and feedforward layers to the input. The attention layer computes the self-attention of the input tensor, allowing the model to weigh the importance of different parts of the input sequence. The feedforward network then applies a non-linear transformation to the attention output, producing a new representation of the input tensor. The output of the feedforward layer is added to the input tensor using residual connections, and layer normalization is applied before returning the output. The "TransformerEncoder" layer also includes dropout layers to regularize the model and reduce overfitting.

The third layer, "MultiHeadAttention", is a sub-layer used in the TransformerEncoder layer. It consists of multiple parallel self-attention operations, or "heads", which allow the model to attend to different parts of the input sequence in parallel. The input tensor is transformed into three different tensors, which are then used to compute the self-attention scores for each head. The resulting attention scores are combined across all heads to produce a single attention matrix, which is used to compute a weighted sum of the input tensor. The resulting output is passed through a linear layer to obtain the final attention output.

The fourth layer, "FeedForward", is also a sub-layer used in the TransformerEncoder layer. It consists of a two-layer feedforward neural network, which applies a non-linear transformation to the input tensor. The output of the first linear layer is passed through a non-linear activation function, such as ReLU, and then through a second linear layer. The resulting output is then added to the input tensor using residual connections, and layer normalization is applied before returning the output.

The Transformer model is a powerful deep-learning architecture consisting of multiple layers of building blocks that enable it to learn complex representations of input sequences and make accurate predictions. While originally designed for natural language processing, the Transformer model has been applied to other domains, such as computer graphics and deep learning for video classification. In computer graphics, the Transformer model has proven effective for tasks like image synthesis and captioning, thanks to its ability to capture long-term dependencies and relationships between different parts of the input sequence. In video classification, the model has been used for tasks like action recognition and video captioning, which improve the accuracy of classification models and make them more robust to variations in video content. However, it is important to note that the Transformer model may not always be the best choice for every task. Depending on the nature of the input data, the complexity of the task, and available computing resources, other deep learning models like CNNs or RNNs may be more suitable. Ultimately, the choice of model depends on careful consideration of these factors.

C. Results

In our experiment, we compare three models (GRU, LSTM, and Transformer); finally, the transformer model obtains the highest accuracy compared to another model mentioned in this paper. Transformers achieve more than 78.57% accuracy with 1,330 video datasets. For training, our models are pre-trained on VGG16. Unless otherwise specified, we use 32-image input clips to fine-tune our models. These clips are created by randomly cutting 64 consecutive frames from the original full-length video and then omitting each frame. The spatial size is 224 x 224 pixels. we use window10 (64 bit), CPU Core™ i9, and RAM 32 GB with the execution of python version 3.9.12. The detailed, tested environment configuration is shown in Table 2.

TABLE II
TESTBED ENVIRONMENT CONFIGURATION

Software/Platform	Value
Window 10	64-bit
Programming Language/tool	Python/Jupyter notebook
CPU	Intel Core ™ i9
RAM	32 GB
Batch size	32
Validation split	0.03
Learning rate	0.0001
Epochs	250
Dataset	1,330 videos

Table 3 shows the classification performance results of three methods: accuracy, precision, recall, and f1-score. Results show the transformer model achieved accuracy 78% better than other methods, and precision, recall, and f1-score are higher than LSTM and GRU models.

TABLE III
CLASSIFICATION ACCURACY, PRECISION, RECALL, AND F1-SCORE COMPARISON WITH TRANSFORMER, LSTM, GRU MODEL

Method	Accuracy	Precision	Recall	F1-Score
Transformer	0.78	0.75	0.75	0.75
LSTM	0.60	0.59	0.59	0.59
GRU	0.20	0.04	0.20	0.20

Table 4 compares classification performance for three types of fabrics using the transfer model: fabric falling down, fabric colliding, and fabric affected by airflow. The results reveal that the fabric falling down achieved the highest accuracy of 78.57%, outperforming the other fabric types in precision, recall, and f1-score.

TABLE IV
CLASSIFICATION ACCURACY, PRECISION, RECALL, AND F1-SCORE COMPARISON USING TRANSFORMER MODEL

Dataset type	Accuracy	Precision	Recall	F1-Score
Fabric falling down	78.57	0.75	0.75	0.75
Fabric with collision	75.06	0.70	0.70	0.70
Fabric affected by airflow	63.02	0.69	0.69	0.69

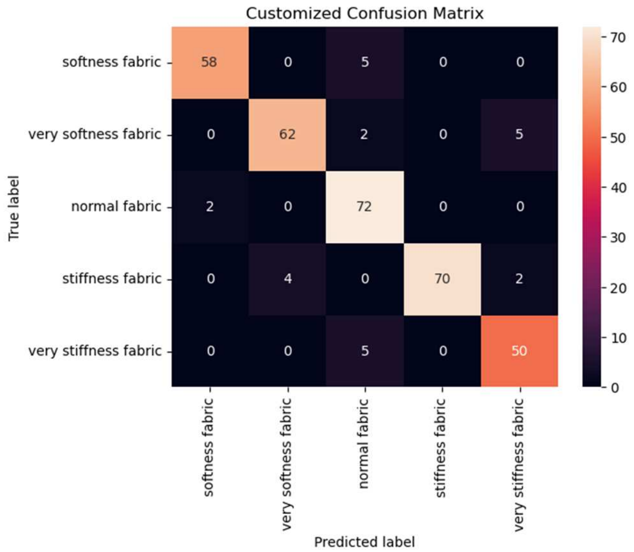

Fig. 9 The confusion matrix analysis of the video classification dataset reveals the results.

IV. Conclusion

In this study, we utilized a transformer model with positional embedding and transformer encoder for video classification and achieved an impressive accuracy of 78.57% on a dataset of 1,330 videos categorized into five classes. Our approach offered significant advantages over LSTM and GRU models, which process video inputs sequentially and can be less accurate and inefficient when dealing with similar datasets.

The transformer architecture we employed processes inputs in parallel, resulting in faster execution times and superior accuracy, particularly in scenarios where context plays a crucial role in classification. The transformer model is a versatile and effective tool for video classification and can also be used in other research areas, such as natural language processing and computer vision.

Our results suggest that the transformer model holds great promise for improving the accuracy and efficiency of video classification. Furthermore, future studies can expand the scope of our research by increasing the dataset size and the number of video categories, which will help further evaluate our approach's performance.

Overall, our research highlights the importance of exploring innovative techniques for video classification, and our approach showcases the potential of transformer models as a powerful tool for advancing research in this field. As video content continues to proliferate in today's digital age, we believe this study will contribute to developing more sophisticated and accurate video classification models that can benefit a wide range of applications, from surveillance systems to social media platforms.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No.: NRF-2022R1I1A3069371), funded by the Ministry of Education; was funded by BK21 FOUR (Fostering Outstanding Universities for Research) (No.:5199990914048).

References

[1] M. Bhatia, N. Manini, F. Biasioli, and L. Cappellin, "Calculated rate coefficients between CI-MS reagent ions and organosulfur compounds causing food taints and off-flavours," *International Journal of Mass Spectrometry*, vol. 478, p. 116860, Aug. 2022, doi: 10.1016/j.ijms.2022.116860.

[2] S. M. Alshammari, J.-L. Le Garrec, and J. B. A. Mitchell, "Electron collisions with dimethylamine and trimethylamine molecules and molecular ions," *International Journal of Mass Spectrometry*, vol. 474, p. 116802, Apr. 2022, doi: 10.1016/j.ijms.2022.116802.

[3] J. S. Bhanot et al., "Adaptation and operation of a quadrupole/time-of-flight tandem mass spectrometer for high mass ion/ion reaction studies," *International Journal of Mass Spectrometry*, vol. 478, p. 116874, Aug. 2022, doi: 10.1016/j.ijms.2022.116874.

[4] N. Sirigiri et al., "Factors controlling the physical properties of an organic ionic plastic crystal," *Materials Today Physics*, vol. 22, p. 100603, Jan. 2022. doi: 10.1016/j.mtphys.2022.100603.

[5] Z. Ballard, C. Brown, A. M. Madni, and A. Ozcan, "Machine learning and computation-enabled intelligent sensor design," *Nature News*, Jun. 2021. doi: 10.1038/s42256-021-00360-9.

[6] Zimmerling, Clemens, Daniel Trippe, Benedikt Fengler, and Luise Kärger. "An approach for rapid prediction of textile draping results for variable composite component geometries using deep neural networks," *In AIP conference proceedings*, vol. 2113, no. 1, p. 020007. Jul. 2019. doi: 10.1063/1.5112512.

[7] J. Jiang, B. Sheng, P. Li, L. Ma, X. Tong, and E. Wu, "Real-time hair simulation with heptadiagonal decomposition on mass spring system," *Graphical Models*, vol. 111, p. 101077, Sep. 2020. doi: 10.1016/j.gmod.2020.101077.

[8] J. Flotyński, K. Walczak, and M. Krzyszkowski, "Composing customized web 3D animations with semantic queries," *Graphical Models*, vol. 107, p. 101052, Jan. 2020. doi: 10.1016/j.gmod.2019.101052.

[9] C.-Y. Liu, L.-J. Chiou, C.-C. Li, and X.-W. Ye, "Analysis of Beijing Tianjin Hebei regional credit system from the perspective of big data credit reporting," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 300–308, Feb. 2019, doi: 10.1016/j.jvcir.2019.01.018.

[10] N. Agarwal and S. Chauhan, "Amplifying Employability Skills to Create Co-Working Space for Human and Cobots in the E-Commerce Industry," *Procedia Computer Science*, vol. 214, pp. 1040–1048, 2022, doi: 10.1016/j.procs.2022.11.275.

[11] B. Gaudenzi, L. Mola, and C. Rossignoli, "Hitting or missing the target: Resources and capabilities for alternative e-commerce pathways in the fashion industry," *Industrial Marketing Management*, vol. 93, pp. 124–136, Feb. 2021, doi: 10.1016/j.indmarman.2020.12.016.

[12] A. Pandey and U. C. Pati, "Image mosaicing: A deeper insight," *Image and Vision Computing*, vol. 89, pp. 236–257, Aug. 2019, doi: 10.1016/j.imavis.2019.07.002.

[13] N. K. S. Behera, P. K. Sa, S. Bakshi, and R. P. Padhy, "Person re-identification: A taxonomic survey and the path ahead," *Image and Vision Computing*, vol. 122, p. 104432, Mar. 2022, doi: 10.1016/j.imavis.2022.104432.

[14] N. Moniz and L. Torgo, "A review on web content popularity prediction: Issues and open challenges," *Online Social Networks and Media*, vol. 12, pp. 1–20, Jul. 2019, doi: 10.1016/j.osnem.2019.05.002.

[15] M. M. Lazar and L. Wan, "Remediatisation, media interdiscursivity and ideological ambivalence in online news reports on sexual assault," *Discourse, Context & Media*, vol. 48, p. 100620, Aug. 2022. doi: 10.1016/j.dcm.2022.100620.

[16] C. Ilbury, "Discourses of social media amongst youth: An ethnographic perspective," *Discourse, Context & Media*, vol. 48, p. 100625, Aug. 2022. doi: 10.1016/j.dcm.2022.100625.

[17] T. Yıldız, "The most effective element in conceptualization is social interaction, not source or modality: A new model of the conceptual development in children," *Learning, Culture and Social Interaction*, vol. 24, p. 100377, Mar. 2020. doi: 10.1016/j.lcsi.2019.100377.

[18] X. Tong, K. Myszkowski, and J. Huang, 'Foreword to the special section on the international conference on computer-aided design and computer graphics (CAD/Graphics) 2019," *Computers & Graphics*, vol. 86, pp. A5–A6, Feb. 2020 doi: 10.1016/j.cag.2019.12.002.

[19] Q. Jing, K. Sasa, C. Chen, Y. Yin, H. Yasukawa, and D. Terada, "Analysis of ship maneuvering difficulties under severe weather based on onboard measurements and realistic simulation of ocean

environment," *Ocean Engineering*, vol. 221, p. 108524, Feb. 2021. doi: 10.1016/j.oceaneng.2020.108524.

[20] L. Accorsi, A. Lodi, and D. Vigo, "Guidelines for the computational testing of machine learning approaches to vehicle routing problems," *Operations Research Letters*, vol. 50, no. 2, pp. 229–234, Mar. 2022. doi: 10.1016/j.orl.2022.01.018.

[21] N. Sirigiri et al., "Factors controlling the physical properties of an organic ionic plastic crystal," *Materials Today Physics*, vol. 22, p. 100603, Jan. 2022. doi: 10.1016/j.mtphys.2022.100603.

[22] V. Terziyan, D. Malyk, M. Golovianko, and V. Branytskyi, "Hyper-flexible Convolutional Neural Networks based on Generalized Lehmer and Power Means," Neural Networks, vol. 155, pp. 177–203, Nov. 2022. Doi: 10.1016/j.neunet.2022.08.017

[23] S. Nag, M. Basu, S. Sanyal, A. Banerjee, and D. Ghosh, "On the application of deep learning and multifractal techniques to classify emotions and instruments using Indian Classical Music," *Physica A: Statistical Mechanics and its Applications,* vol. 597, p. 127261, Jul. 2022.doi: 10.1016/j.physa.2022.127261

[24] Y. Feng, Y. Yang, W. Deng, H. Chen, and T. Ran, "SyntaLinker-Hybrid: A deep learning approach for target specific drug design," *Artificial Intelligence in the Life Sciences*, vol. 2, p. 100035, Dec. 2022, doi: 10.1016/j.ailsci.2022.100035.

[25] R. Codina and Ö. Türk, "Modal analysis of elastic vibrations of incompressible materials using a pressure-stabilized finite element method*," Finite Elements in Analysis and Design*, vol. 206, p. 103760, Sep. 2022. doi: 10.1016/j.finel.2022.103760.

[26] Z.-X. Zou, S.-S. Huang, T.-J. Mu, and Y.-P. Wang, "ObjectFusion: Accurate object-level SLAM with neural object priors," *Graphical Models*, vol. 123, p. 101165, Sep. 2022, doi: 10.1016/j.gmod.2022.101165.

[27] H. Wang, B. Ma, J. Cao, X. Liu, and H. Huang, "Deep functional maps for simultaneously computing direct and symmetric correspondences of 3D shapes," *Graphical Models*, vol. 123, p. 101163, Sep. 2022, doi: 10.1016/j.gmod.2022.101163.

[28] Y.-W. Zhang, B.-B. Qin, Y. Chen, Z. Ji, and C. Zhang, "Portrait relief generation from 3D Object," *Graphical Models*, vol. 102, pp. 10–18, Mar. 2019, doi: 10.1016/j.gmod.2019.01.002.

[29] A. Chekir, "A deep architecture for log-Euclidean Fisher vector end-to-end learning with application to 3D point cloud classification," *Graphical Models*, vol. 123, p. 101164, Sep. 2022. doi: 10.1016/j.gmod.2022.101164.

[30] O. Pueyo, X. Pueyo, and G. Patow, "An overview of generalization techniques for street networks," *Graphical Models*, vol. 106, p. 101049, Nov. 2019. doi: 10.1016/j.gmod.2019.101049.

[31] D. Mahapatra, A. Poellinger, and M. Reyes, "Interpretability-Guided Inductive Bias For Deep Learning Based Medical Image," *Medical Image Analysis*, vol. 81, p. 102551, Oct. 2022. doi: 10.1016/j.media.2022.102551.

[32] C. Vişan et al., "Automated circuit sizing with multi-objective optimization based on differential evolution and Bayesian inference," *Knowledge-Based Systems*, vol. 258, p. 109987, Dec. 2022. doi: 10.1016/j.knosys.2022.109987.

[33] Vaswani, Ashish, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. “Attention is All you Need.” *NIPS,* 2017.