Multimodal Deep Regression on TikTok Content Success

Louis Wong
Georgia Tech
lwong64@gatech.edu

Mingyao Song
Georgia Tech
msong41@gatech.edu

Jason Xu
Georgia Tech
jxu623@gatech.edu

Ahmed Salih
Georgia Tech
asalih6@gatech.edu

Abstract

Content creators grapple with the challenge of predicting if their time and investments will translate into increased viewership and audience growth, a task made more complex by the hidden algorithms and unpredictable audience interaction of social media platforms. This research's objective is to architect a model that predicts video success, effectively indicating a video's potential virality. By employing advanced convolution techniques for video encoding, leveraging strides in natural language processing models, we're pushing boundaries in deep video content analysis. We construct a powerful multimodal ensemble model for general video content regression, capable of comprehending human-generated content and accurately predicting its nonlinear relationship elements to determine success. Our preliminary results demonstrate the model's effectiveness in predicting video virality, showcasing the potential of our innovative approach.

1. Introduction

The landscape of digital content creation is continuously evolving, presenting content creators with the challenge of predicting the success of their videos in terms of viewership and audience growth. This challenge is further complicated by the opaque algorithms and unpredictable audience interactions on social media platforms like TikTok. To address this issue and empower content creators, various methods and models have been explored for predicting video success and understanding the factors contributing to content virality.

To address the challenges of interpreting and predicting YouTube viewership, Liu et al. [1] proposed a novel Precise Wide-and-Deep Learning model. This model accurately predicts viewership using unstructured video data and established features while providing precise interpretations of feature effects. In the context of TikTok, where content and user preferences are continually evolving, researchers [2] delved into predicting user participation in TikTok challenges. They introduced a novel deep learning model capable of learning and combining latent user and challenge representations from past videos to predict a user's likelihood of participating in a challenge. Salvador et al. [3] studied on human action recognition and explored the use of attention mechanisms to improve the accuracy and efficiency of video recognition models. In their work, they integrated space and time attention mechanisms into the framework of Vision Transformer network structure for feature extraction from video data. To effectively understand and recommend short videos, a multi-modal fusion framework was proposed [4], integrating features from different modalities to capture inherent relationships. Deep neural networks were employed for feature extraction and fusion to accomplish video understanding and recommendation tasks. An approach to describing videos using multi-modal fusion techniques was explored [5]. The research presented a deep neural network that combined visual and textual information at various stages in the network, aiming to learn a joint representation for video description tasks. Additionally, researchers focused on predicting the popularity of images and videos on Instagram [6]. They employed convolutional neural networks and long short-term memory networks to extract spatial and temporal information from images and videos, respectively, and used a regression model to predict popularity. Another work investigated popularity prediction on Instagram using neural networks and regression analysis [7]. The authors explored the predictive power of image composition on Instagram posts by comparing the popularity predictions of neural networks trained on aesthetic value with predictions from regression models using social metadata.

While these work makes significant contributions, research focused on multi-modal fusion encounters the challenge of effectively integrating information from diverse modalities, such as visual and audio data, to achieve a cohesive and informative representation. The success of fusion techniques heavily relies on striking the appropriate balance between these modalities and ensuring their seamless integration.

In multimodal learning, there are two main fusion approaches: early fusion, which concatenates original or extracted features at the input level, and late fusion, which aggregates predictions at the decision level. The performance comparison between early and late fusion is influenced by various factors [8, 9], such as the characteristics of the multimodal data, the complexity of the task, the interdependence between modalities, the quality of the features, the architecture of the network, the size of the dataset, and the availability of labeled data. Therefore, there is no definitive answer regarding which fusion approach is universally superior. Each approach can prove to be more effective in specific scenarios. In this study, we have chosen the late fusion approach. Through careful evaluation and analysis of the aforementioned factors specific to our project, we have determined that late fusion offers distinct advantages. By combining predictions at the decision level, we can leverage the strengths of each modality effectively, allowing us to capture complementary information and improve overall performance

In this study, our goal is to predict video virality using a multimodal ensemble model. To achieve our goal, we collected a diverse dataset of approximately 5,000 videos from TikTok, covering a wide range of hashtag topics, such as Sports, Dance, Entertainment, Comedy, Autos, Fashion, Lifestyle, Pets and Nature, Relationships, Society, Informative, and Music. The dataset includes relevant metadata for each video, such as video IDs, TikTok URLs, and video view counts.

Our approach involves data preparation and multimodal deep regression modeling. The data preparation process includes scraping video data from TikTok, audio extraction, and meticulous organization of visual tensors for efficient integration into the model. For audio analysis, we leverage the open-source Whisper model to transcribe audio content and obtain audio embeddings that capture the semantic meaning of the audio. The heart of our model lies in the visual embeddings generated through an unsupervised pretraining process using a ConvLSTM Autoencoder. This process encodes the context of the video into compact and informative embeddings that retain essential spatial and temporal features. Subsequently, the visual and audio embeddings are concatenated and fed into a Transformer-based regression model for multimodal analysis. The late fusion technique combines the visual and audio data, enabling the model to learn the semantic and nonlinear relationships that contribute to video creator success. The Transformer model, with its self-attention layers and feed-forward neural networks, captures complex patterns and relationships within the data.

This research can benefit various stakeholders in the social media ecosystem. Content creators, including influencers, advertisers, and artists, can gain valuable insights into the potential success of their videos before investing time and resources. The predictive ability can help creators can optimize their content strategies, increase their audience reach, and potentially experience viral success. This research will empower creators to make data-driven decisions, improve their content's impact, and enhance their overall presence on social media platforms.

In the following sections, we will delve into the specifics of our implemented model, including its training process and the results showcasing its performance.

2. Approach

2.1. Data collection and preparation

The data collection process involved scraping video data from TikTok using a custom-built Selenium scraper running on a Chromium browser. The scraper was designed to randomly collect videos across various hashtag topics, ensuring a diverse representation of content. The selected hashtag topics included Sports, Dance, Entertainment, Comedy and Drama, Autos, Fashion, Lifestyle, Pets and Nature, Relationships, Society, Informative, and Music. This approach aimed to capture a wide range of video content to ensure the model's generalizability. In total, the data collection effort yielded approximately 5,000 videos, each in .mp4 format, resulting in a substantial dataset with a total size of 32.7 GB. To facilitate further analysis and model training, each video was assigned a unique video ID tag for easy reference and organization. Alongside the video files, the dataset also includes JSON data containing relevant metadata for each video, such as the video ID tag and its corresponding TikTok URL. Additionally, the video view count on TikTok was also recorded as an essential metric for evaluating video creator success.

Data preparation is a critical aspect in the training of multimodal deep regression models, encompassing several pivotal steps, such as data loading, preprocessing, and splitting, to achieve optimal outcomes. Initially, audio extraction from video datasets is performed, with the extracted audio files thoughtfully organized into an audio directory. Simultaneously, the visual data undergoes meticulous processing, involving frame skip, shrink scale adjustments, and normalization to enhance consistency and compatibility. The resultant processed visual tensors are then meticulously stored in their respective directories, poised for seamless integration into subsequent training and validation stages, paving the way for an efficient and robust multimodal deep regression model. To maintain a clean and organized workflow, a virtual environment is set up outside the project folder to prevent conflicts with other packages. We leverage appropriate utilities for loading and processing the data, including audio extraction, transcribing, and obtaining embeddings. data transformations such as normalization and frame skipping are applied to enhance the data quality. The dataset is split into training and validation sets to evaluate the model's performance effectively. Data loaders are created to enable efficient batching during the training process, leading to smoother and more effective training of the model.

2.2. Model

The video is first broken down into visual and audio tracks, with our primary focus on the video visual. The video visual will undergo an autoencoder process, utilizing a convolution-based network architecture, ConvLSTM Autoencoder, to unsupervised pre-training from scratch. This process encodes the context of the video into embedding vectors. Subsequently for the audio, we leverage a pretrained model, the open-source Whisper, to create a transcript for the audio, supplementing the project. Afterward, visual embeddings and audio embeddings are respectively extracted from the visual branch and the audio branch, which are then be concatenated and input into a Transformer-based regression model. The aim is for this model to learn the semantic and non-linear relationships needed to predict video creator success metrics, such as video views. Finally, we establish a baseline using a less complex model and compare it with our main implementation to evaluate its success.

2.2.1 Visual Embedding

To generate compact and informative visual embeddings, an unsupervised pretraining method using a ConvLSTM Autoencoder is employed. The Autoencoder architecture consists of ConvLSTM layers, which are a combination of Convolutional and Long Short-Term Memory (LSTM) layers. The ConvLSTM layers are used to process the video frames, capturing both the spatial features (through the convolutional layers) and the temporal dependencies (through the LSTM layers). This allows the Autoencoder to retain important contextual information from the video data while reducing the dimensionality to create compact embeddings.

During training, the Autoencoder takes video frames as input and tries to reconstruct them at the output. The difference between the original and reconstructed frames is quantified using the Mean Squared Error (MSE) loss function. Through backpropagation, the Autoencoder adjusts its weights and biases to minimize this reconstruction error, thus learning to capture meaningful patterns in the video data. The trained Autoencoder is evaluated thoroughly over multiple epochs to ensure that it learns robust and meaningful embeddings. In each epoch, the Autoencoder is tested on both training and validation data, and the loss values are recorded to monitor the training progress. The embeddings generated by the Autoencoder represent the visual context of the video. These embeddings condense the raw visual data into a more informative and compact representation, which is crucial for downstream visual data analysis.

2.2.2 Audio Embedding

Audio embeddings are obtained using the open-source Whisper model through unsupervised pretraining. Whisper is a powerful automatic speech recognition (ASR) system developed by OpenAI to convert spoken language into text. First, the audio is extracted from the video dataset, and then Whisper transcribes the audio dialog, producing textual transcripts that capture essential information from the spoken content. These textual outputs serve as audio embeddings, effectively encapsulating the semantic meaning and characteristics of the audio. Utilizing the pre-existing Whisper model for audio embedding generation offers several advantages. The Whisper model is already trained on extensive speech data, making it effective in understanding diverse spoken language patterns. This saves the effort and time required to train an ASR system from scratch, making the process more efficient. The resulting audio embeddings play a pivotal role in augmenting audio analysis and comprehension capabilities, facilitating downstream tasks that demand a profound understanding of the audio content.

2.2.3 Transformer-based Ensemble Model

Transformer-based ensemble model is employed to perform multimodal deep regression by combining visual and audio data using late fusion technique. The ensemble model is composed of two main components: visual Transformer model and audio Transformer model. Both models utilize the Transformer architecture, which consists of multiple self-attention layers and feed-forward neural networks. These models take visual and audio embeddings as inputs, respectively, and process them using the Transformer layers to capture complex patterns and relationships within the data. Positional encodings are added to the input data, providing positional information to the Transformer model. The positional encodings are calculated using sine and cosine functions with varying frequencies.

The ensemble model is trained on the combined data using Mean Squared Error (MSE) loss and the Adam optimizer for a specified number of epochs. The trained model is evaluated on the validation set, and the MSE between the predicted values and ground truth values is calculated, providing an indication of the model's performance. The model generates plots to visualize the training and validation loss during the training of the Ensemble Model. Finally, the model inspects and compares the ground truth and predicted values for a random sample from the validation set, and the MSE value between the two sets of values is obtained.

2.3. Loss Function

The model is trained using the mean squared error (MSE) loss function, which serves as a measure of the discrepancy between the model's predicted values and the actual ground truth video creator success metrics. The primary objective is to minimize this discrepancy and achieve a high level of accuracy in predicting the success metrics. The MSE loss function calculates the average squared difference between the predicted values and the actual values. By squaring the differences, it penalizes larger errors more severely, emphasizing the importance of accurate predictions across all data points. MSE loss function is defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Where n is the total number of samples in the dataset. y_i represents the actual value of the success metric for the i^{th} video creator. $\hat{y}_i$ represents the predicted value of the success metric for the i^{th} video creator.

2.4. Implementation

We utilized PyTorch (version xxx) and Python 3.xxx to implement our model. Hyperparameters are listed in Table 1.

3. Experiments and Results

3.1. Experiment Setup

Jason: Due to the size of the dataset, it was cost prohibitive to train in the cloud, so experiments were ran on our personal computers with NVIDIA RTX 2070 Super, 2080 Super, and 3080 GPUs. This caused some issues as our GPUs have limited VRAM capacities. Early attempts at training would fail due to exceeding both RAM and GPU memory. Initially we reduced the batch size, but found that this would greatly increase the training time on an already very large training set. As a result, frame skipping and shrinking were two techniques we chose to reduce the computational and storage requirements while maintaining essential information. We were careful about choosing the degree of shrinkage and number of frames to skip given that too high of either might lead to a loss of important visual details and temporal information. We decided on shrinking by a factor of 8. Although this resulted in significantly reduced frame clarity, subjects within the frames remained identifiable. The number of frames to skip was set to 200, which was the only way to run experiments without each epoch taking a significant amount of time.

A series of additional experiments revealed that the number of frames to skip has a significant impact on the performance of the autoencoder. When the number of frames to skip was reduced from 200 to 100, the validation loss of the autoencoder decreased by approximately 300%. This suggests that skipping fewer frames allows the autoencoder to learn more temporal information, which in turn improves its performance. However, this experiment was computationally expensive, requiring 80 epochs, each of which took an average of 40 minutes to complete. This resulted in a total runtime of over 2 days.

In this section, we provide a comprehensive and detailed account of the experiments conducted to train and evaluate the Convolutional LSTM Autoencoder and Ensemble Model within the context of the multimodal deep regression framework. The methodologies, parameters, and configurations utilized during both the training phases are explained. All the hyper-parameters are listed in Table 1. The training and validation losses are listed in Table 2.

3.1.1 Convolutional LSTM Autoencoder

The initial phase of the experiments focuses on training the Convolutional LSTM Autoencoder, a crucial component responsible for learning a compact and meaningful representation of the input video data. The Autoencoder follows an unsupervised learning approach, with the primary objective of reconstructing the original input from the learned latent space representation. The following key parameters and settings are employed during this training phase:

The 'video_pack_20' dataset is chosen for training the Convolutional LSTM Autoencoder. Before training, the video data undergoes preprocessing steps to prepare it for the model. The 'Frame Skip' parameter is utilized to determine how many frames to skip during the preprocessing stage, effectively reducing the temporal depth of the video. Additionally, the 'Shrink' parameter is applied to scale down the resolution of each frame, resulting in a reduced (Height x Width) dimension. Notably, 'Normalization' is disabled in this experiment, indicating that pixel values are not normalized to a specific range. Furthermore, the 'Pad All' parameter is set to False, implying that padding is not applied to all tensors to match the maximum depth.

The Convolutional LSTM Autoencoder is trained using the Mean Squared Error (MSE) loss function, which calculates the mean of the squared differences between the predicted and ground-truth frames. The training process spans three epochs, during which the model learns to reconstruct the input video frames effectively. The 'Hidden Size' of the Convolutional LSTM is set to 64, determining the size of the hidden layer in the model.

Throughout the training iterations, the losses on both the training and validation sets are monitored and recorded. These loss values provide valuable insights into the performance and generalization capabilities of the Autoencoder. Additionally, the total number of trainable parameters in the Convolutional LSTM Autoencoder is calculated and reported, amounting to 1,041,859.

3.1.2 Ensemble Model

The second part of the experiments involves training the Ensemble Model, a more complex architecture that combines the outputs from the Visual Transformer and the Audio Transformer with the learned visual and audio embeddings from the pre-trained Convolutional LSTM Autoencoder. The Ensemble Model is designed to effectively fuse information from both visual and audio modalities and predict the output values. The following parameters and settings are applied during this training phase:

Similar to the Autoencoder training, the 'video_pack_20' dataset is utilized in this phase. However, the Ensemble Model requires additional data preparation for the audio modality. The 'extract_audio' function is called to extract audio from the video dataset and save it in .wav format. Subsequently, the 'extract_embeddings' function is used to transcribe the audio dialog and extract Low-Level Modulation Spectrogram (LLMs) embeddings from the audio files. These LLMs embeddings are essential for training the Audio Transformer within the Ensemble Model.

The Ensemble Model is trained using the Mean Squared Error (MSE) loss function, similar to the Autoencoder. The training process spans three epochs, allowing the model to learn from the data effectively.

The Ensemble Model consists of two transformer-based sub-models: the Visual Transformer and the Audio Transformer. Both sub-models share common hyper-parameters, including the number of attention heads, hidden dimension, and the number of transformer layers. Specifically, the 'Number of Attention Heads' is set to 8, providing the models with multiple attention mechanisms to focus on relevant visual and audio features. The 'Hidden Dimension' is set to 256, representing the size of the hidden layer in each transformer. Finally, the 'Number of Transformer Layers' is set to 6, determining the depth and complexity of the transformer-based architectures.

Throughout training, the losses on both the training and validation sets are recorded to assess the Ensemble Model's performance. Similar to the Autoencoder, the total number of trainable parameters in the Ensemble Model is calculated and reported, which amounts to a substantial 2,228,625,922.

3.2. Results

The results obtained from the experiments on the Convolutional LSTM Autoencoder and Ensemble Model are presented and discussed below.

3.2.1 Convolutional LSTM Autoencoder Results

After training the Convolutional LSTM Autoencoder, the losses on the training and validation sets are analyzed and visualized on a graph. The graph demonstrates the trend of decreasing losses on both the training and validation sets over the three epochs. This indicates that the Autoencoder effectively learns to reconstruct the input video frames and captures essential visual features in the learned latent space representation. The decreasing loss values affirm that the Autoencoder has learned to produce accurate reconstructions of the original video frames.

To further evaluate the quality of the Autoencoder's reconstructions, a random sample inspection is performed on the validation set. The actual video frames and their corresponding reconstructed versions are visually compared. The results illustrate that the Autoencoder successfully preserves critical details and spatial structures in the frames, indicating its proficiency in reconstructing visual information.

3.2.2 Ensemble Model Results

Following the training of the Ensemble Model, the losses on the training and validation sets are monitored and plotted on a graph. The graph displays the trend of decreasing losses on both the training and validation sets over the three epochs. This indicates that the Ensemble Model effectively fuses information from both visual and audio modalities, enabling it to make predictions for the output values based on multimodal inputs.

To further analyze the Ensemble Model's predictions, a scatter plot is generated, displaying test values and the corresponding predicted values. The plot shows the correlation between the actual test values and the model's predictions. It allows for an initial assessment of the Ensemble Model's performance in predicting output values.

The performance of the Ensemble Model is further evaluated using the Mean Squared Error (MSE) metric on the validation set. The MSE value provides insights into the overall accuracy of the Ensemble Model's predictions. It helps assess how closely the predicted values align with the ground-truth values and indicates the quality of the Ensemble Model's predictions.

In brief, the experiments entail the training and evaluation of both the Convolutional LSTM Autoencoder and the Ensemble Model within the multimodal deep regression framework. The Autoencoder demonstrates proficiency in reconstructing visual frames, while the Ensemble Model effectively fuses information from multiple modalities. Further analysis and optimization may be required to fine-tune the Ensemble Model's architecture and training process, which could potentially improve the model's predictive capabilities and overall performance.

4. Conclusion and Future Improvements

In conclusion, the experiments involved training and evaluating the Convolutional LSTM Autoencoder and Ensemble Model within the multimodal deep regression framework. While the Autoencoder demonstrated proficiency in reconstructing visual frames and the Ensemble

Model successfully fused information from multiple modalities, the overall results were not as strong as desired. The models were able to produce outputs, but the predictive capabilities were limited, and the accuracy of the predictions could be improved.

It is essential to acknowledge that the complexity of the multimodal deep regression task, with the integration of both visual and audio data, poses inherent challenges. The limited performance could be attributed to factors such as the training cost, performance of the personal computer, dataset's size, hyper-parameter configurations, and the model architectures. Further exploration and optimization of hyper-parameters, network architectures, and training strategies might yield more robust and accurate predictions.

In future work, more extensive and diverse datasets could be explored to enhance the models' ability to generalize across various scenarios. Additionally, fine-tuning the hyper-parameters and experimenting with different model architectures could lead to substantial improvements in predictive performance. Techniques like transfer learning and pre-training on larger datasets for the Autoencoder and Transformers might also contribute to better feature representation and overall performance.

Despite the current limitations, the experiments lay a solid foundation for future advancements in multimodal deep regression tasks. As the field of deep learning continues to evolve, these preliminary results provide valuable insights and directions for further research and development.

It is important to note that the focus of this work was to introduce and experiment with the multimodal deep regression framework. With further refinements and enhancements, such a framework holds great potential for diverse applications, including audio-visual recognition, video analysis, and multimodal data processing.

While the results may not have met the perfect performance levels, the experiments showcased the feasibility of the multimodal deep regression approach in handling complex tasks involving visual and audio data. The work sets the stage for future investigations and improvements in this exciting and challenging area of research. By continuing to explore advanced techniques and methodologies, the potential for more accurate and robust predictions in multimodal data analysis can be realized.

5. Work Division

Contributions of each group member can be found in Table 3.

References

[1] Jiaheng Xie, Yidong Chai, and Xiao Liu. ”Unbox the Black-Box: Predict and Interpret YouTube Viewership Using Deep Learning.” Journal of Management Information Systems, 2023, 541-579.

[2] Lynnette Hui Xian Ng, John Yeh Han Tan, Darryl Jing Heng Tan, Roy Ka-Wei Lee. Will you dance to the challenge? Predicting user participation of TikTok challenges. In Proceedings of the 2021 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM'21), The Hague, 2021, 356–360. New York: ACM.

[3] Qiliang Chen, Hasiqidalatu Tang, and Jiaxin Cai. ”Human Action Recognition Based on Vision Transformer and L2 Regularization.” In Proceedings of the 2022 11th International Conference on Computing and Pattern Recognition (ICCPR '22), 2022, 224-228.

[4] Daya Guo, Jiangshui Hong, Binli Luo, Qirui Yan, Zhangming Niu. ”Multi-modal representation learning for short video understanding and recommendation.” In 2019 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 2019, 687-690.

[5] Jin Q., Chen J., Chen S., Xiong Y., and Hauptmann A. ”Describing videos using multi-modal fusion.” In ACM MM, 2016.

[6] Massimiliano Viola, Luca Brunelli, and Gian Antonio Susto. ”Instagram Images and Videos Popularity Prediction: a Deep Learning-Based Approach.” Università degli Studi di Padova, Padova, IT.

[7] Qian C., Tang J., Penza M., and Ferri C. ”Instagram Popularity Prediction via Neural Networks and Regression Analysis,” 2017, 2561-2570.

[8] Snoek, C. G. M., Worring, M., and Smeulders, A. W. M. ”Early versus late fusion in semantic video analysis.” In Proceedings of the Annual ACM International Conference on Multimedia, Singapore, 2005, 399-402.

[9] Gadzicki, K.; Khamsehashari, R.; Zetzsche, C. Early vs late fusion in multimodal convolutional neural networks. In Proceedings of the 2020 IEEE 23rd International Conference on Information Fusion (FUSION), Rustenburg, South Africa, 2020, 1–6.

ConvLSTMAutoencoder	Transformer Visual and Audio	Ensemble Model
Learning Rate: 1e-4 Epochs: 10 Hidden Size: 64	Number of Attention hHeads: 8 Number of Layers: 6 Hidden Size: 256	Learning Rate: 1e-3 Epochs: 3 Audio Transformer: True

Table 1. Hyperparameters in this project.

Model	Training Loss	Validation Loss
ConvLSTMAutoencoder	ABCDE	ABCDE
Ensemble Model	ABCDE	ABCDE

Table 2. Performance of models.

Student Name	Contributed Aspects	Details
Louis Wong	ABCDE	ABCDE
Mingyao Song	ABCDE	ABCDE
Jason Xu	ABCDE	ABCDE
Ahmed Salih	ABCDE	ABCDE

Table 3. Contributions of team members.

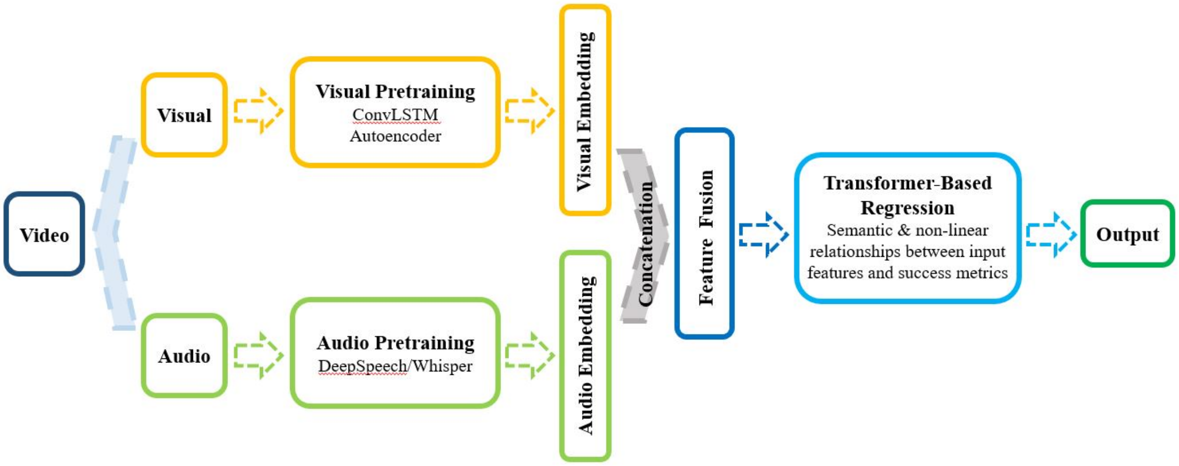

Figure 1. Illustration of the model

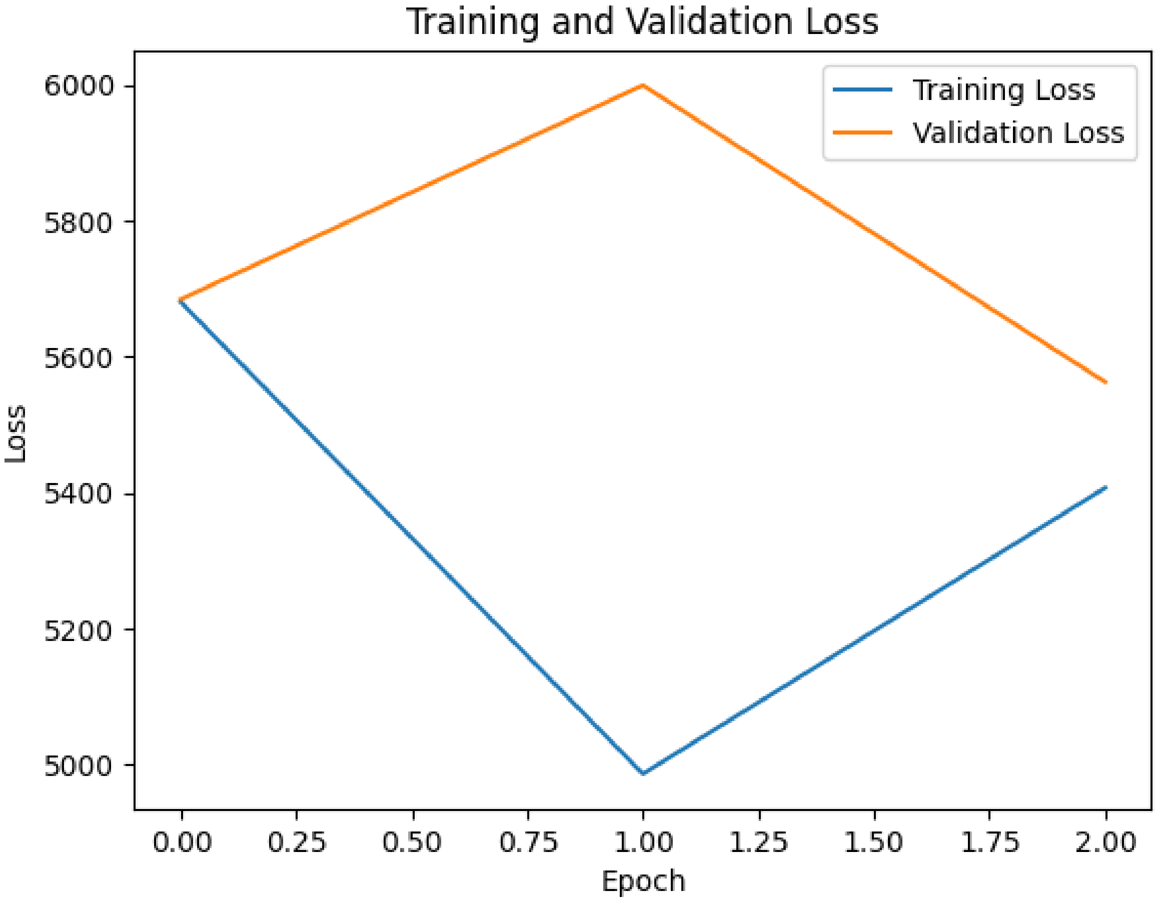

Figure 2. ConvLSTM Autoencoder: Training and validation loss

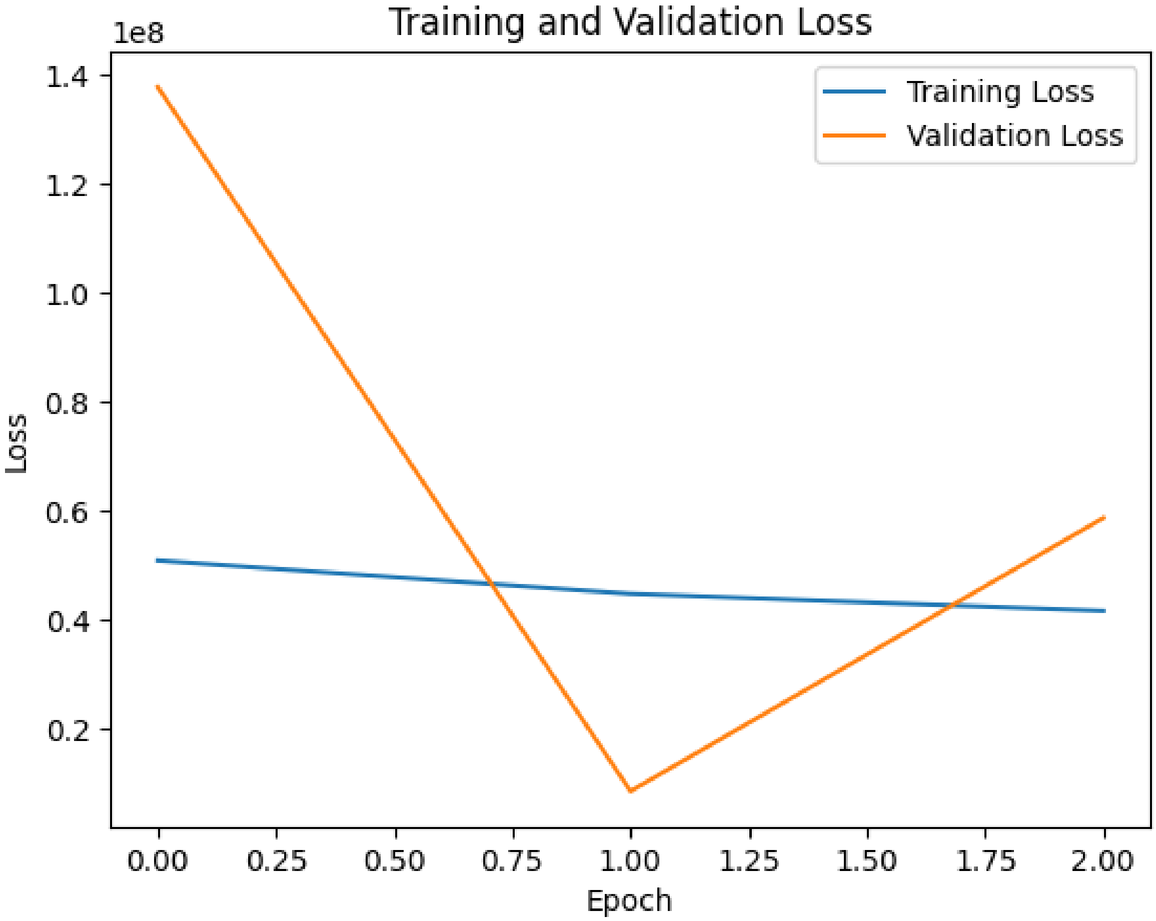

Figure 3. Ensemble model: Training and validation loss

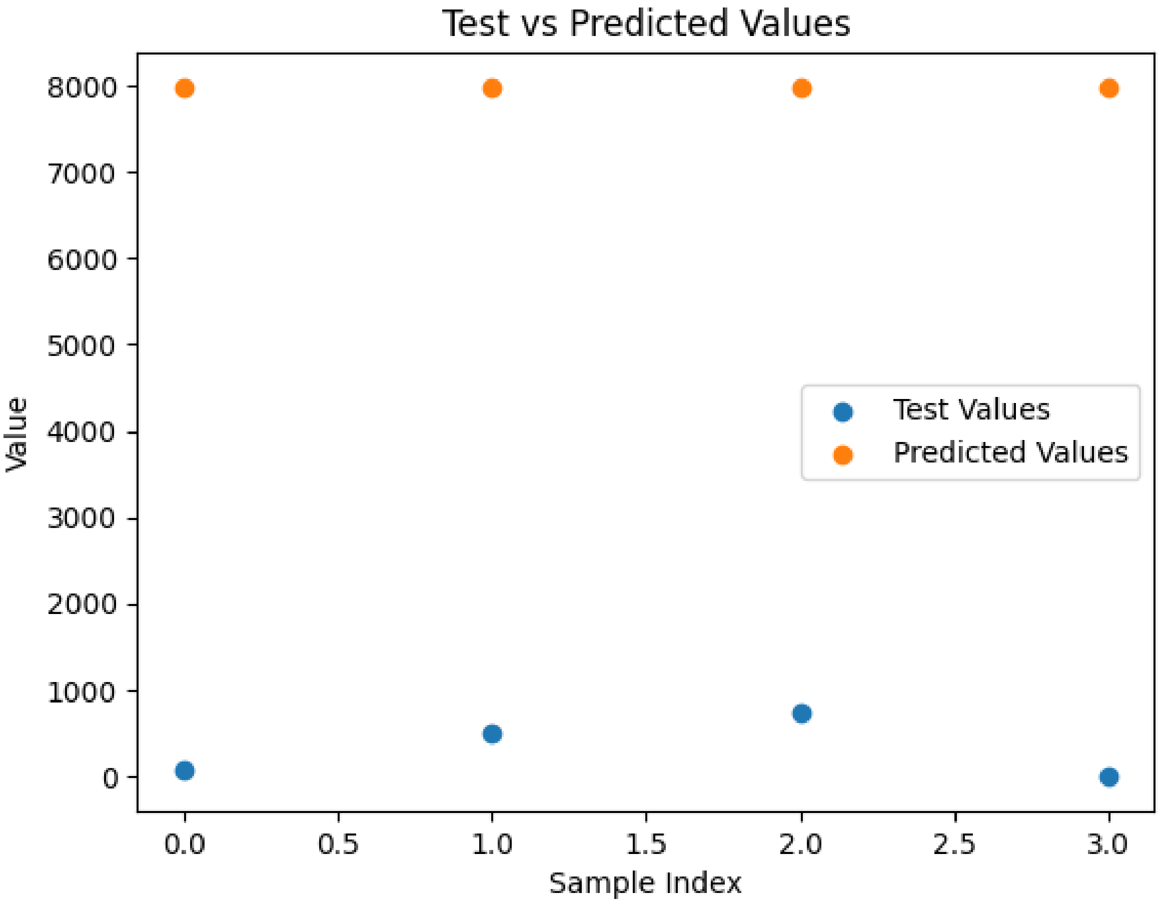

Figure 4. Test vs Predicted Values