The Viral Vision: Computer Vision Models and the Cultural Zeitgeist

JOSEPH ELLIS, DYLAN MCINTOSH, SAMUEL BERNSEN, KAREN NGO, and SPENCER AU

The study proposes a novel computer-vision technique to assess and predict short videos' virality. This study utilizes a dataset of TikTok videos, downloaded based on statistics like view count, likes, shares, and includes audio.

This research allows a closer look into what makes a video viral and offers insight for content creators and marketers to optimize engagements.

1 Introduction

The concept of virality as a tool has been a revolutionary change brought forth by the evolution of the internet. Clicks, likes, and other integer counts representing interactions now have a dollar sign attached. With the current generation so deeply rooted in social media and entertainment channels, it is only natural that companies put so much importance into online presence. Entire jobs now hold responsibilities for maximizing interactions and maintaining them.

This brings us back to the concept of virality, the "golden goose" that everyone is constantly looking for in the entertainment world. With the eruption of short-video formats in TikTok, several creators have shot into fame from low-effort and low-cost productions. In this avenue lies spectacular opportunities for brands of any size to make content that boosts their public knowledge. However, there will always be a degree of randomness in how a video happens to be boosted to the point of virality. For example, a well-planned out and edited video could end up with a couple thousand likes, while a video of a piece of toast falling over could get a hundred thousand. Several factors can be investigated in how a video is processed in the "For You" algorithm.

Our question, which we want to research and answer, is whether a computer vision model could accurately assess and predict the 'virality' of a video. Using powerful tools from Pytorch Video, we can create models that decode video data while maintaining temporal information [1]. Pytorch tools allow us to assess the audial and textual in tandem with the videos, such that we can also use the information of likes, views, and shares [1]. Plans are to add the caption (hashtags, etc.) from the videos, as this information goes into the "For You" algorithm. The benefits of this research could be critical for the average content creator and marketing teams of larger businesses. With a tool that analyzes a video's chances of going 'viral,' brands can branch out and have more confidence in different content ideas. Smaller brands could find the success they have been searching using the model. Finally, individual content creators could use it to reduce some of the stress of maintaining popularity after a hit video.

The model architecture will use the R(2+1)D structure proposed in [2]. Combining 2D and 3D layers allows for efficient temporal and visual information assessment. The dataset used contains viral and unpopular videos from TikTok, given that this is the main source of conversation surrounding viral content. These videos will be split into training, validation, and testing sets, along with proper preprocessing before training. Predicting a video's virality could depend greatly on the context, thus explaining our consideration in model selection. Our model will be evaluated utilizing unseen data from TikTok and other social media sources. The goal is to have a generalizable model that can recognize a viral video from any source.

Authors' address: Joseph Ellis, joellis@chapman.edu; Dylan McIntosh, dmcintosh@chapman.edu; Samuel Bernsen, bernsen@chapman.edu; Karen Ngo, kango@chapman.edu; Spencer Au, spau@chapman.edu.

2 Related Works

With the increased focus on the monetary benefits of popularity, it is only natural that investigations have been carried out on this subject before. Previous studies focused on a simple binary classification of posts labeled as popular or unpopular (based on normalized view counts) [3]. The videos (taken from Facebook) were split frame by frame and analyzed to find moments that contribute to the popularity of the video [3]. This used an intuitive gradient calculation of the popularity score with respect to the output layer of a ResNet50 model [3]. Accuracy results showed that combining convolutional models with attention while leveraging an LSTM's ability to find temporal connections found the best success [3]. A similar method is proposed in this paper, utilizing a weighted combination of video statistics to create a virality score. This score will be a key part in the way our model assesses how popular a video is or not. Additionally, the paper in [3] found that temporal connections are vital for video analysis tasks; thus, this became a major consideration, as stated previously.

We want our model to connect not only the video information but the video itself or, more specifically, the content. Thus, we plan to utilize a model architecture to maintain these details. Another paper proposed a novel CNN architecture titled R(2+1)D that we plan to investigate. Using a dual 3D-2D layer structure, the model can capture motion and spatial information [2]. This architecture was tested on datasets ranging from sports to general human actions. Given that this new combination model outscored past 2D and 3D CNN models, it is hopeful that this formula could work for our proposal. The paper in [2] mentions that future work could look at adjusting the pre-trained architecture used (ResNet) or even different structures, thus are all avenues considered when the model was being developed.

Since this R(2+1)D model was only utilized for action datasets, we planned to test our plan utilizing another visual model, such as a visual transformer. Transformers have found, in many cases, much more success at sequential modeling than models like LSTM [4]. A Memory-Augmented Recurrent Transformer (MART) was able to find success captioning videos using an external memory module [4]. The utilization of a memory module could be a novel method for us to make use of, adjusting a transformer of our own to better remember sequential connections. Thus, our method adjusted to account for comparison between different model architectures and to better understand how these structures can understand temporal information.

3 Data & Methods

3.1 Data Collection

The dataset contains 7522 TikTok videos with several metrics regarding the video's viral statistics. This dataset was created from web scraping TikTok, not using the official API. This means that we encountered many issues with collection since web scraping policies are strict. These issues range from download corruption to methodically pulling data slowly. It was collected in the context of a Guest account, initially grabbing 25 videos from the For You Page. The videos were then recursively scraped from the initial 25 videos by continuously looping to randomly select a video present in the dataset and scraping 5 related videos. This led to videos of many different strata of viral success in the dataset to encourage generalizability. The recursive nature of this scraping process leads to much variability in the demographics that the videos target to get a general idea of virality. Metrics recorded include likes, views, comments, and shares.

3.2 Data EDA

An exploration of the dataset was conducted to understand how to best adjust model architecture and create a virality feature. Below are plots relating to the metadata of the videos collected.

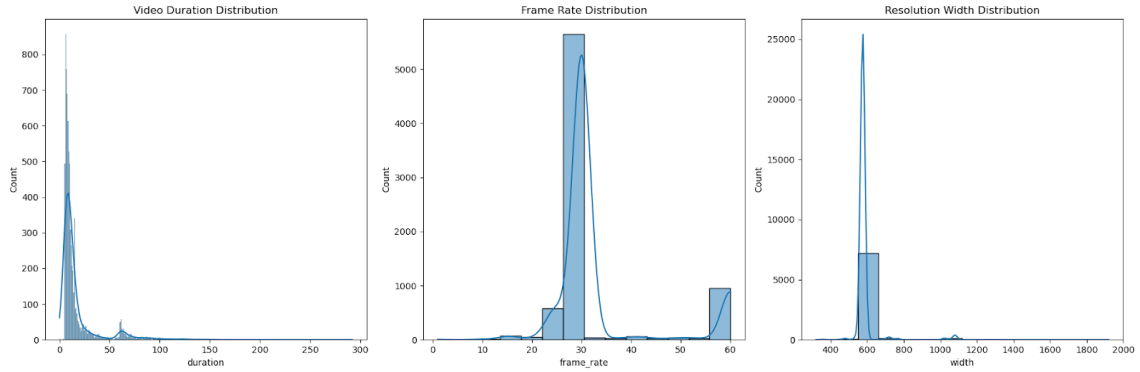

Fig. 1. Video Duration Distribution

Figure 1, displaying video duration, indicates that a majority of the videos are under 30 seconds, with a slight spike around 60 second videos. The frame rate of the videos was much more varied than expected. There were videos with as little as 1 frame per second, which is poor data to include in the training. There was also slight variability in the resolution of videos. These plots led us to remove any videos with under 100 frames from the dataset, which ended up being 20 videos. This was done since videos with that little amount of frames would harm the learning of the model and would be near incomprehensible.

Metric	Mean	Std	25th Percentil	50th Percentile	75th Percentile	Max
Comments	2,773	20,689	145	527	1,743	1,600,000
Likes	308,553	996,870	23,500	78,600	258,675	46,300,000
Plays	2,519,753	10,187,573	281,125	803,750	2,300,000	684,000,000
Shares	10,363	50,299	135	903	4,732	1,900,000

Fig. 2. User Engagement Metrics Distribution

An exploration of the user engagement metrics (Figure 2) was conducted to indicate how we capture the idea of virality from this dataset. Table 1, describing the description of the metrics, is below along with the log distribution plots.

Metric	Mean	Median	Max
Likes	12,345	2,345	123,456
Views	123,456	34,567	1,234,567
Comments	1,234	123	12,345
Shares	1,234	123	12,345

Table 1. Description of User Engagement Metrics

This shows how extreme the dataset is with there being a massive difference between the 75th percentile and the max value for all of the metrics. That is normal though since viral videos are inherently outliers. This makes modeling virality tricky since the scale is so vast.

3.3 Virality Engineering

The extreme range of user engagement metrics led to testing two different approaches for this problem. We engineered a continuous virality as well as a binary virality. This allows us to explore the problem from both a regression and a simplified binary classification approach.

*3.3.1 Continuous Virality*The engagement metrics utilized were likes, views, comments, and shares to define virality as a continuous variable. The end goal of the continuous method was to have a virality score between [0, 1] in order to use a sigmoid prediction in the model. Much attention was needed to handle the scaling of the score into this range since outliers are so intense in this dataset. The metrics were transformed to get the log of each variable, and then scaled using RobustScaler, which is designed to handle outliers well. These transformed variables were used in a linear combination to create a raw virality score. The linear combination is defined as such:

$$\text{Virality Score} = 0.1 \times \text{commentCount} + 0.4 \times \text{likesCount} + 0.3 \times \text{playCount} + 0.2 \times \text{shareCount} \quad (1)$$

This combination was chosen for general knowledge of social media algorithms, and could be improved upon using empirical based methods. This raw virality score was then scaled between 0 and 1 using a min-max scaler. The distribution of virality scores can be seen in Figure 3 below.

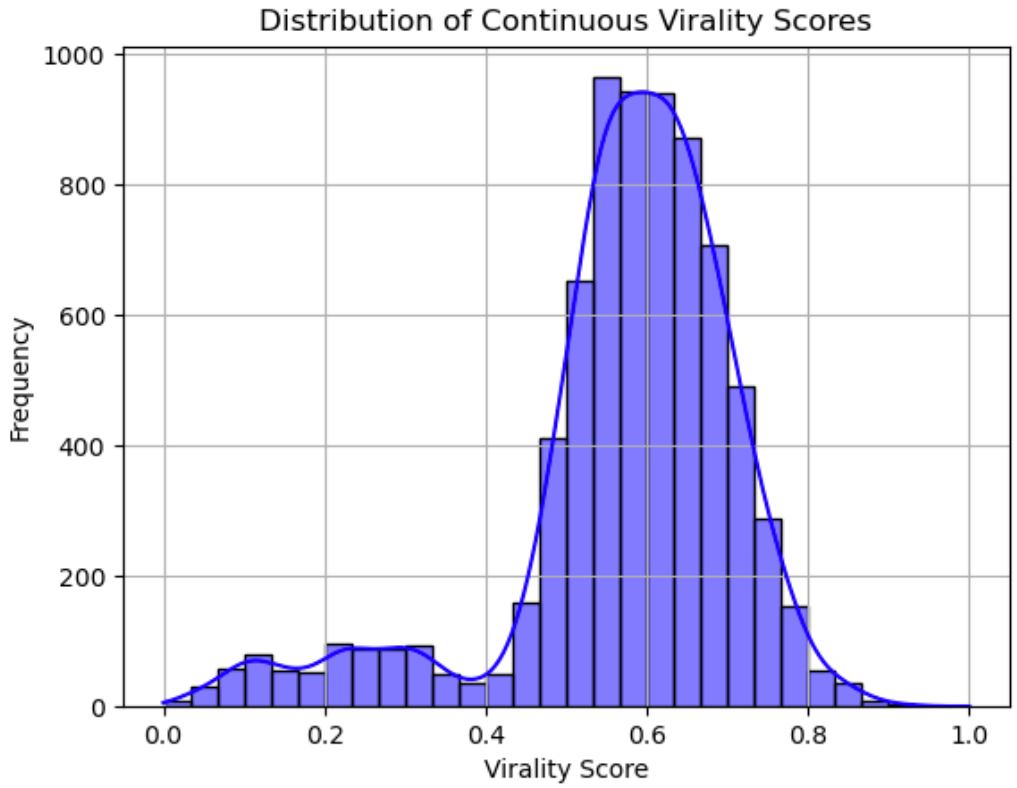

Fig. 3. Distribution of Virality Scores

The distribution above shows that this method leads to a very small amount of extremely viral scores, with quite a few low virality scores. There are improvements to make to this method, but this should suffice for determining the feasibility of the task.

*3.3.2 Binary Virality*The engagement metrics utilized were likes, views, comments, and shares to define virality as a binary variable. The approach was done by calculating thresholds for each engagement metric at the 50th percentile. Videos with all engagement metrics being greater than the 50th percentile were deemed viral, and all other videos were deemed non-viral (1 is viral, 0 is non-viral). The distribution of the labels is seen in Figure 4 below.

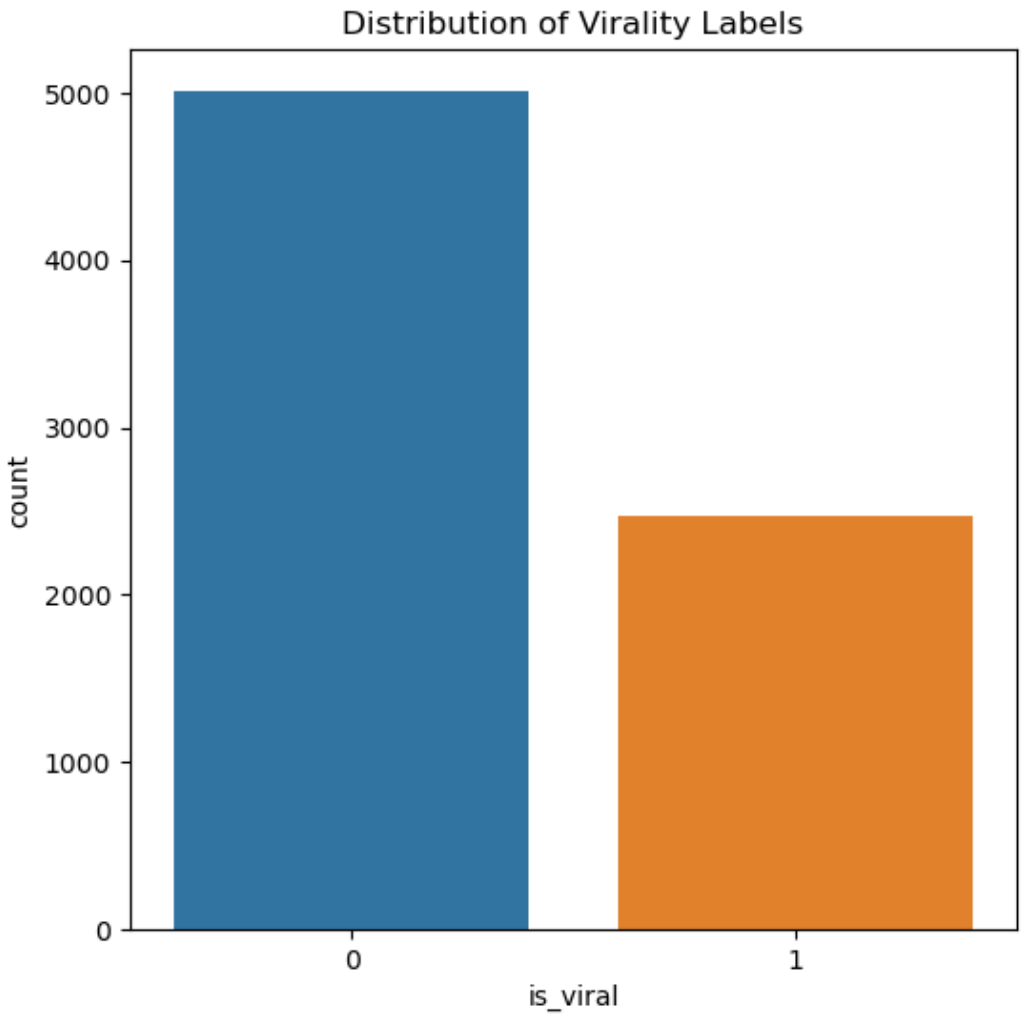

Fig. 4. Distribution of Binary Virality Labels

The intent was to retain the intensity of virality while mitigating a heavy class imbalance. This intention led to the distribution above with double the amount of non-viral videos compared to viral videos.

3.4 Video Preprocessing

Video preprocessing is more involved than traditional image preprocessing for deep learning since the temporal dimension adds more complexity. Videos were resized to 112x112 width and height and normalized using means and standard deviations noted in [1]. These preprocessing steps match the model's original preprocessing steps for the Kinetics400 dataset [1]. Random clips were extracted from each video of 32 frames long. The videos were also padded with zeroes in the temporal dimension if an extracted clip was less than the set length. The information order in the tensor inputted into the model required exact specifications; thus, permutations were required. If every batch were not processed in this exact way, training would not work.

4 Model Architectures

Initially, we found multiple video understanding models to choose from. V-JEPA from Meta Research seemed promising because it is a state-of-the-art video understanding model with high prediction capabilities. However, using it proved difficult in our environment for various reasons. ResNet 3D (R3D) stood out as a top candidate, but in our investigation of the model trained on the same dataset, we found a more powerful and more computationally efficient model, R(2+1)D. Our final model uses the R(2+1)D backbone developed for data with both spatial and temporal dimensions. This backbone works well to capture spatiotemporal dynamics in videos, which is why we see it working well for our task. Since this approach works well with the task of action recognition, we hoped to slightly modify the architecture to predict virality/popularity. The R(2+1)D architecture performs 3D convolution by applying convolution twice, once for the spatial dimension and once for the temporal dimension. This slight adaptation to R3D significantly improves computational efficiency and reduces the number of learnable parameters. The architects of R(2+1)D include average pooling after all (2+1) convolutional layers to down-sample. R(2+1)D was inspired by ResNet, so it utilizes many skip connections which preserve information from previous layers. This helps mitigate the vanishing gradient problem encountered in deeper architectures.

The vanilla version has densely connected layers at the end for the task of action recognition. It was originally trained on the Kinetics400 dataset with 400 possible classes for human actions. In both solutions to our problem, we only want 1 output node. This corresponds to the virality score (regression) or the binary is-viral / is-not-viral determination (classification). After empirical testing, we finalized the model with three dense layers after the average pooling step. Simply including one or two dense layers could not capture meaningful relationships between features in the feature map space. The first dense layer is connected to the average pooling layer and has 512 nodes. ReLU activation and a 50% dropout were added for regularization. Next, we feed the information to a 128 node layer, again with the ReLU activation and 50% dropout. The final layer is one node. For classification, we apply a sigmoid activation function which forces the output to be between 0 and 1. For regression, we apply ReLU to keep all values positive.

5 Training

5.1 Regression

As discussed in the section on model architecture, the final layer contains a sigmoid activation function to ensure all outputs remain between 0 and 1. Thus, the model architecture remains identical between the training for continuous and binary virality. The only difference between the two methods is the labels pulled from the dataset alongside the video data.

As is the case for regression tasks, MSE loss is used as the criterion to be utilized for weight updates each epoch. Our optimizer was Adam (learning rate of 0.001) alongside a learning rate scheduler with a step size of 7. We trained both models for 20 epochs but as you will see from the training results, early stopping often activated around 5 to 6 epochs in.

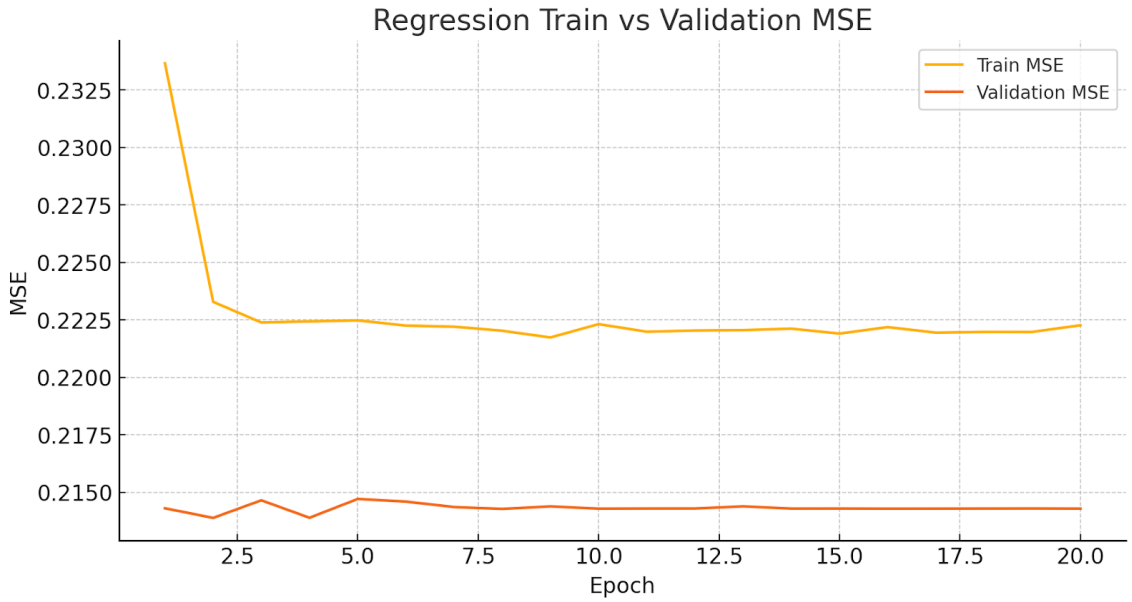

Fig. 5. Training and Validation MSE

Figure 5 displays the comparison in MSE between the training and validation sets. The model immediately plateaus around epoch 2 and does not get better or worse throughout (with the best training MSE being around 0.214). Various differences in layers, preprocessing, and model architectures were attempted to fix this training, however it either got worse or did not change.

5.2 Classification

As mentioned in 5.1, the only difference for the binary virality change is using the label, 'is_viral.' We assumed this training would be an improvement over the last method, given the simplification of a binary classification task compared to regression. Additionally, the hypothesis was that due to outliers the normalization for virality scores might have been affected too much by outliers, thus making that more difficult to predict. Unfortunately, as seen in Figure 6, the plateau happened around the same epoch as regression, with the best training loss being around 0.66. A BCE loss of 0.66 means that the model is practically guessing, which is an inference confirmed in the testing results for classification.

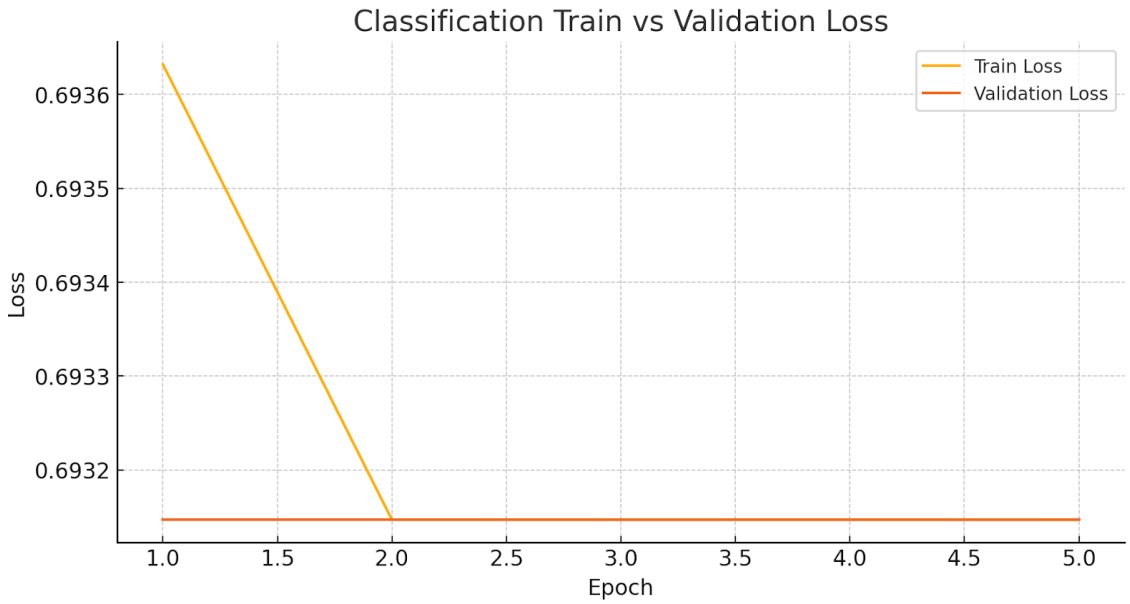

Fig. 6. Training and Validation Loss for Classification

6 Results

6.1 Regression

The continuous virality score prediction process allowed us to utilize the '3-Best 3-Worst Videos' to gain a better understanding of how the model is predicting so incorrectly.

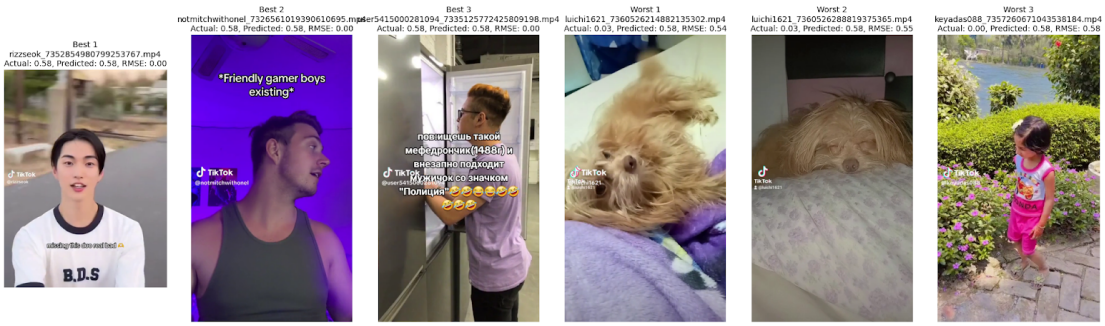

Fig. 7. 3-Best 3-Worst Videos Prediction

Figure 7 displays this, showing that the model is predicting the exact same value, 0.58, for every single video. This occurrence is presumably the reason for the model's extremely poor values for MSE, RMSE, and MAE seen in Table 2.

Metric	Value
MSE	0.0208
RMSE	0.1443
MAE	0.1008

Table 2. Regression Results

While the MSE does not appear to be poor, based on the RMSE the model is quite inaccurate considering the values are normalized between 0 and 1.

6.2 Classification

The values for the binary value testing came out even worse than the regression model, seen in Table 3. An accuracy of 0.6572 is about as good as the model guessing whether or not a video is viral or not. The F1-score and recall reveal more information about these predictions, showing that the model is likely predicting all videos as viral, thus getting none of the non-viral predictions correct. This inference is cemented by the confusion matrix shown in Figure 8. Attempts were made to find a more accurate decision threshold, however this never changed the results.

Metric	Value
Accuracy	0.6572
F1-Score	0
Recall	0
ROC-AUC	0.4908

Table 3. Classification Results

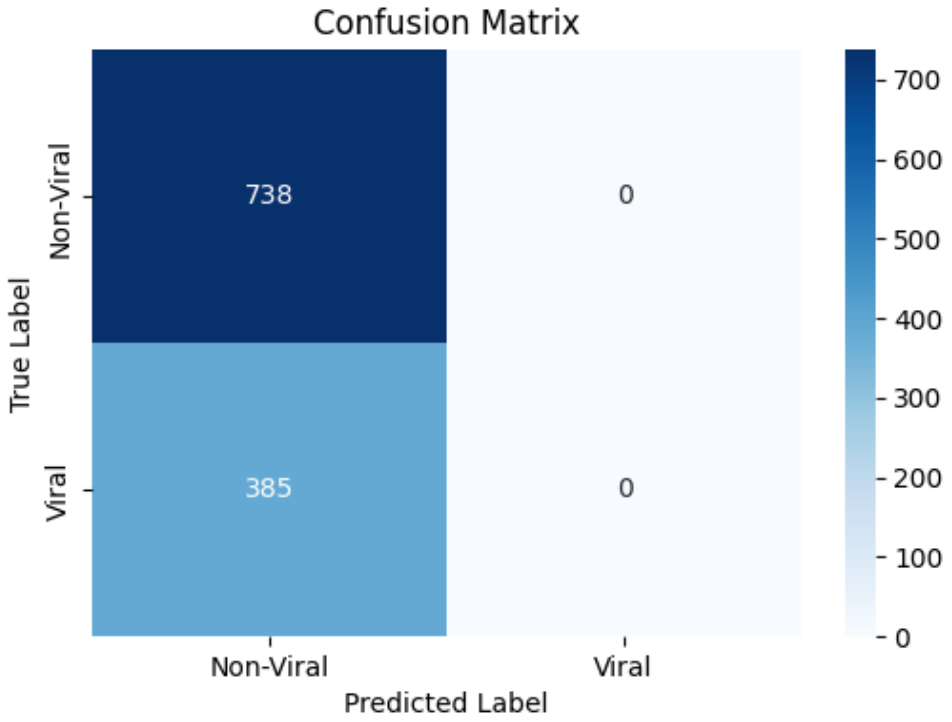

Fig. 8. Confusion Matrix

7 Conclusion

Seen by the '3-Best 3-Worst' for the regression version and the confusion matrix for the classification version, both models had an issue where they only predicted one value. Throughout all the variations in architecture, layers, and other hyper-parameter settings this prediction error remained an issue. Although the binary virality score is much simpler, the model remained unable to discern between viral and not-viral. It can be presumed that the content of the videos is simply too different and variable for a model to learn to predict its virality. Looking deeper into the videos, you could see two with the same content (for example, a woman lip syncing a song) and one getting viral and the other not. It was expected that the model was going to have difficulty getting a good accuracy, given that it is almost impossible for human subjects to predict virality.

A point of virality that would make it such a useful tool for prediction is that it is so unpredictable. The ability to gain confidence in a post's ability to gain extremely viral interactions would be an extremely lucrative program. Unfortunately, without more information about the videos or better visual models, this task will likely continue to be impossible.

8 Future Work

There is much work needed to be done in order to solve the problem of virality prediction. This project could be taken down many avenues of improvement, touching on dataset related changes or model architecture improvements.

The most obvious step in the right direction would be to collect much more data. Since virality is such a complex subject, a lot of data is needed for a model to be able to predict it. With a much larger dataset, videos could be split into categories. A different model could be trained on each category to get a more exact understanding of virality for each demographic since TikTok has such a wide array of video types.

Another area for improvement is the engineering of virality. There are so many different approaches to defining virality into a number. A more optimal scaling method could be thought out, or different thresholds could be determined. The previously mentioned improvement of more data collected would allow for a thorough analysis of the landscape of internet virality to numerically define it more appropriately.

Adjustments to the model architecture are the next area to discuss. There may be architectures better suited for this problem such as SWIN transformers or VJEPPA. Attempts with various architectures are needed to identify the most successful approach. Incorporating a form of audio embedding into these future models has major potential. The current model does not utilize auditory information in its predictions. We believe that audio plays an important role in a video's viral performance on the internet.

These are just a few methods for improving upon this implementation of virality prediction. Employing all of these techniques may lead to fruitful results.

References

[1] Facebook Research. *PyTorchVideo*. https://github.com/facebookresearch/pytorchvideo/tree/main. Accessed: 2024-04-15. 2024.

[2] Du Tran et al. "A Closer Look at Spatiotemporal Convolutions for Action Recognition". In: *Journal of Machine Learning Research* 19.1 (2018), pp. 1–35.

[3] Adam Bielski and Tomasz Trzcinski. "Pay Attention to Virality: Understanding Popularity of Social Media Videos with the Attention Mechanism". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA: IEEE, June 2018.

[4] Jie Lei et al. "MART: Memory-Augmented Recurrent Transformer for Coherent Video Paragraph Captioning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020.