Ad Transition Detection in Videos Using Shot-Based Encoding and Supervised Learning

Who

Alan Wang (alan_wang2@brown.edu)
Dagar Rehan (dagar_rehan@brown.edu)
Adebowale Adelekan (adebowale_adelekan@brown.edu)

Link to Code

https://github.com/awang343/ad-detection

Introduction

Our project tackles the problem of **ad transition detection in videos**, focusing on identifying transitions between movies and advertisements. This problem is important for enhancing user experiences on streaming platforms by enabling seamless ad skipping, improving automated content editing, and facilitating better ad placement analytics.

We chose this topic because it is a real-world challenge that intersects video processing, machine learning, and user-centered design. The problem is framed as a **classification task**: determining whether a shot transition belongs to one of four categories—ad-to-movie, movie-to-ad, ad-to-ad, or movie-to-movie.

The motivation stemmed from exploring ways to improve streaming services, where detecting ad boundaries can significantly enhance the quality of service. We employed a **contrast learning approach**, inspired by a research paper titled *"ShotCol: A Self-Supervised Approach for Scene Representation Learning,"* and modified the method for the specific context of ad versus movie detection.

Related Work

The primary work we referenced is the Amazon Prime research paper introducing **ShotCol**, which uses a self-supervised learning approach to optimize representations of video shots. The paper described training encoders using shot-similarity but did not provide direct implementations, requiring us to replicate and adapt their methodology.

Other works on ad detection often use rule-based approaches or metadata, but few employ deep learning techniques to analyze transitions. By combining supervised learning with shot-based analysis, our approach represents a novel extension of ShotCol's original idea.

Data

We used two main datasets:

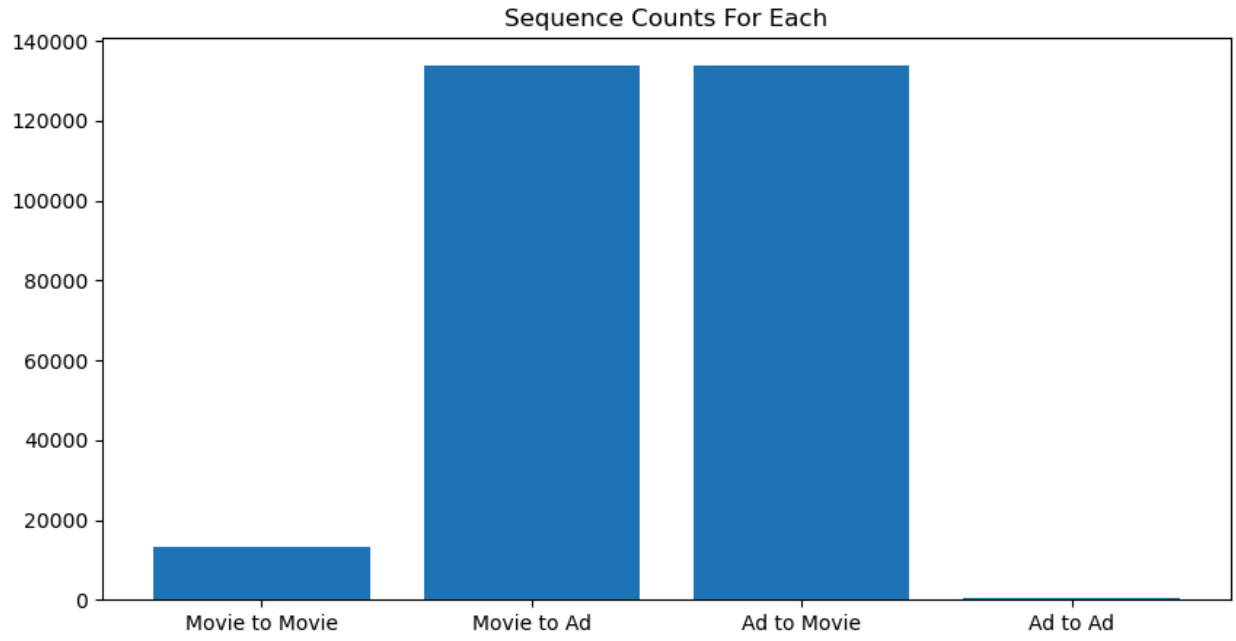
1. **Movies**: A total of 71 free movies were collected from YouTube at **360p resolution** to ensure manageable download and training times. Due to DRM restrictions, genres were randomly selected, and preprocessing involved segmenting each movie into individual shots using a color histogram-based approach. Each movie has an average of 17651 shots.
2. **Advertisements**: A total of 176 advertisements were also collected from YouTube at **360p resolution**, selected randomly to ensure diversity. Each advertisement has an average of 28 shots.

The resulting dataset includes thousands of shot transitions, with labels indicating their type (ad-to-movie, movie-to-ad, etc.). Preprocessing involved:

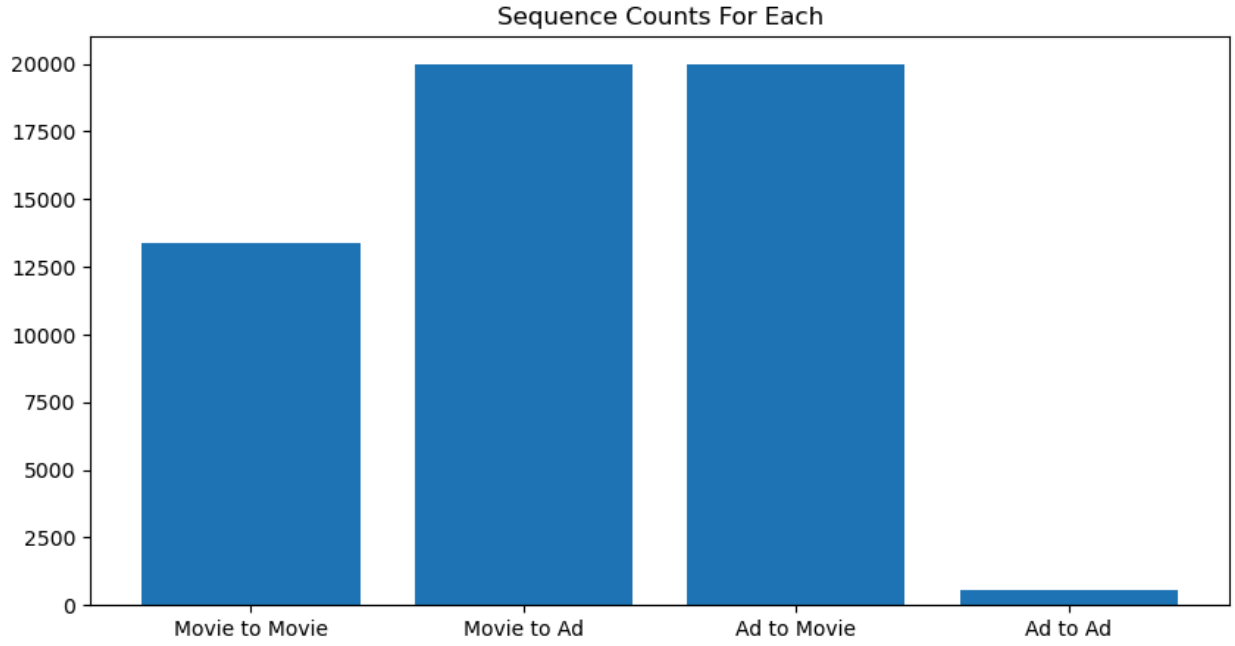
- **Shot segmentation**: Using Bhattacharyya distance on color histograms to detect scene changes. We were able to produce
- **Feature representation**: We had to ensure that all shots had the same numpy shape to pass into our downstream models. We did this by averaging frame data for each shot as input for the model.
- **ShotCol-enhanced Encoding**: We used the ShotCol method to refine the ResNet 50 model and produce a more specific encoder for our task.

One challenge faced was the distribution of the different sequence types. Our goal was to classify scene transitions as “Movie to Movie”, “Movie to Ad”, “Ad to Movie”, and “Ad to Ad”. The

distribution of the labels in our sequence data was as follows:



We generated our sequences by taking every group of seq_length shots from each movie and each ad, splicing together all possible combinations. Due to basic rules of combinatorics, we were grossly oversampling movie to ad transitions. We decided to somewhat alleviate this problem by randomly sampling a smaller subset of “Movie to Ad” and “Ad to Movie” for our training and testing. We brought the number of “Movie to Ad” and “Ad to Movie” shots down to 20000 with random sampling.

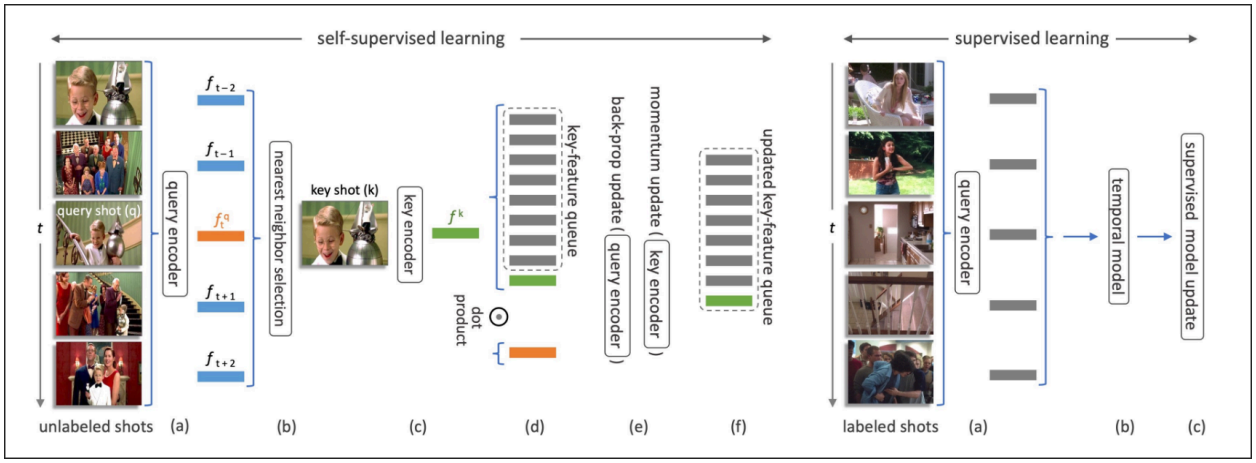


Unfortunately, due to the difficulty of collecting ads, we found it difficult to generate many “Ad to Ad” sequences. This was a limitation of our project.

Methodology

The foundation of our methodology lies in combining supervised pre-text with supervised classification to detect transitions between ads and movies. Our approach consists of two main components: the encoder for feature extraction and the classification model for transition prediction.

The first step involved implementing a **shot-contrastive learning encoder** inspired by the ShotCol framework described in the Amazon research paper. We fine-tuned a ResNet-50 model to act as the encoder, tasked with extracting features from video shots. This component used a contrastive loss function to train the encoder by comparing representations of similar and dissimilar shots. The primary goal was to optimize the encoder to produce distinctive feature embeddings for ad shots versus movie shots. More specifically, we took each movie shot and found its “nearest neighbor” in the temporal window around that shot using our query encoder. This was our positive key. Our negative keys were random shots from ads. We structured the shots this way because we didn’t just want overall movie shots to be similar to each other. We specifically wanted movie shots that were temporally close to be represented by the encoder similarly. This more specific task is more useful for the downstream task, which takes in coherent and consecutive sequences of shots, and not just random movie and ad shots.

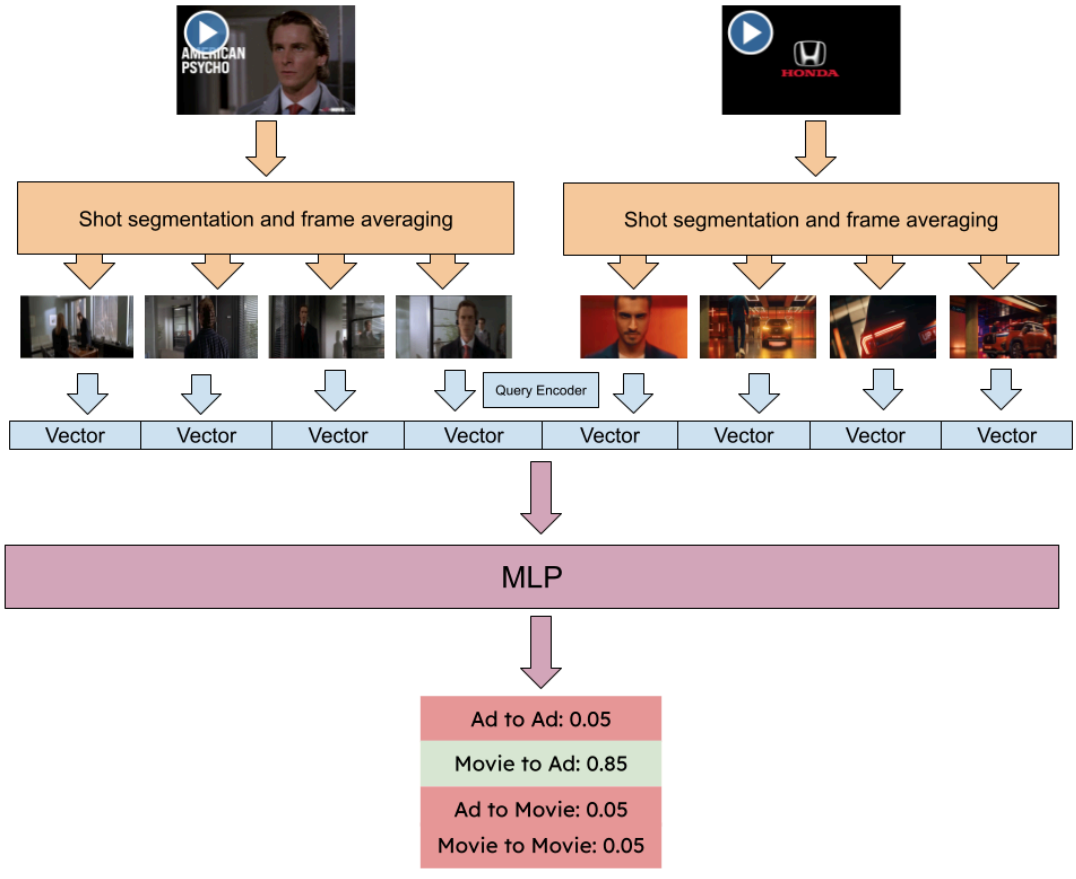


The above picture is taken from the Amazon paper we used as reference. We reimplemented the model, except instead of using previous key shots as negative keys, we used randomly shuffled shots from ads.

After obtaining feature embeddings from the encoder, we fed these representations into a **multi-layer perceptron (MLP)** for supervised classification. The MLP was trained to predict one of four transition classes: ad-to-movie, movie-to-ad, ad-to-ad, or movie-to-movie. To build the dataset for this supervised step, we labeled transitions from the segmented video data and

paired consecutive shot embeddings with their corresponding transition type. Finally, we investigated alternative methods for shot representation. Initially, we averaged all the frames in a shot to produce a single frame representation, but we hypothesized that using key frames (e.g., the first, middle, or last frame) might better capture essential information. Ultimately, we did not find too big of a difference between the methods.

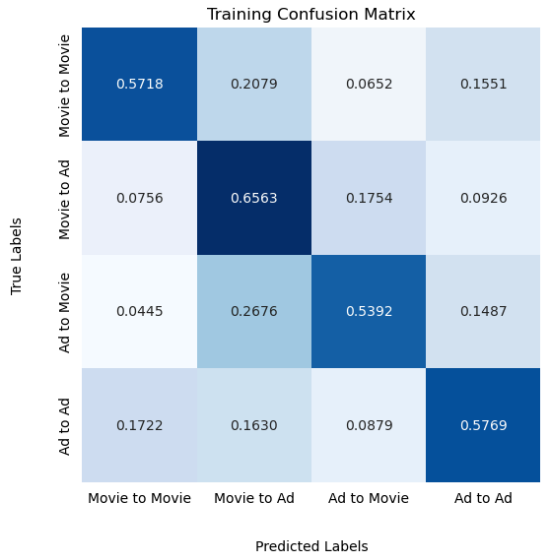
Throughout the process, we iteratively tuned hyperparameters, including learning rates, batch sizes, and contrastive loss weights, to optimize model performance. We also experimented with different thresholds for the Bhattacharyya distance during shot segmentation to ensure high-quality inputs for the encoder. By combining supervised pre-text, supervised fine-tuning, and temporal modeling, our methodology aimed to address the challenge of ad detection comprehensively.

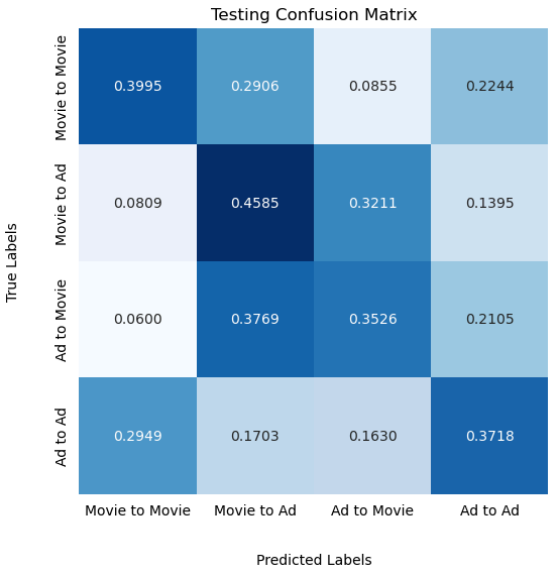


The above figure is a diagram of our training pipeline for the MLP section. Note that the query encoder was pre trained using the ShotCol pipeline.

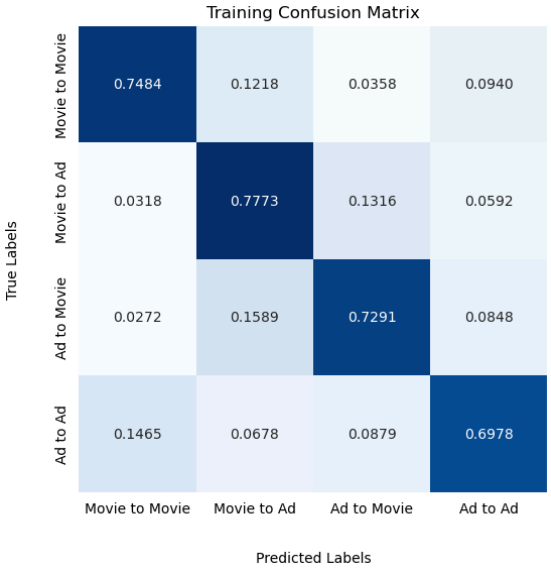
Results

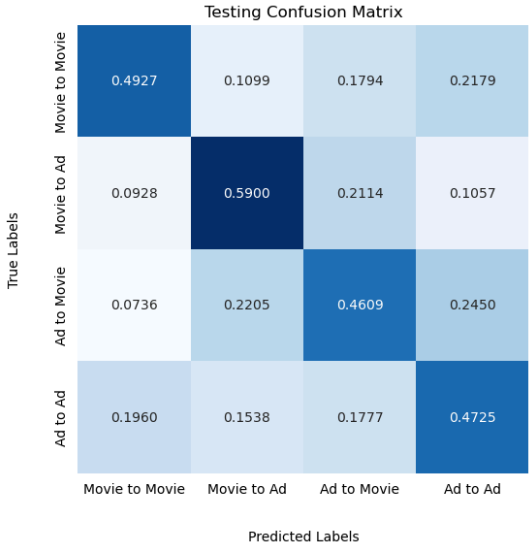
RestNet-50 Encoder Results:

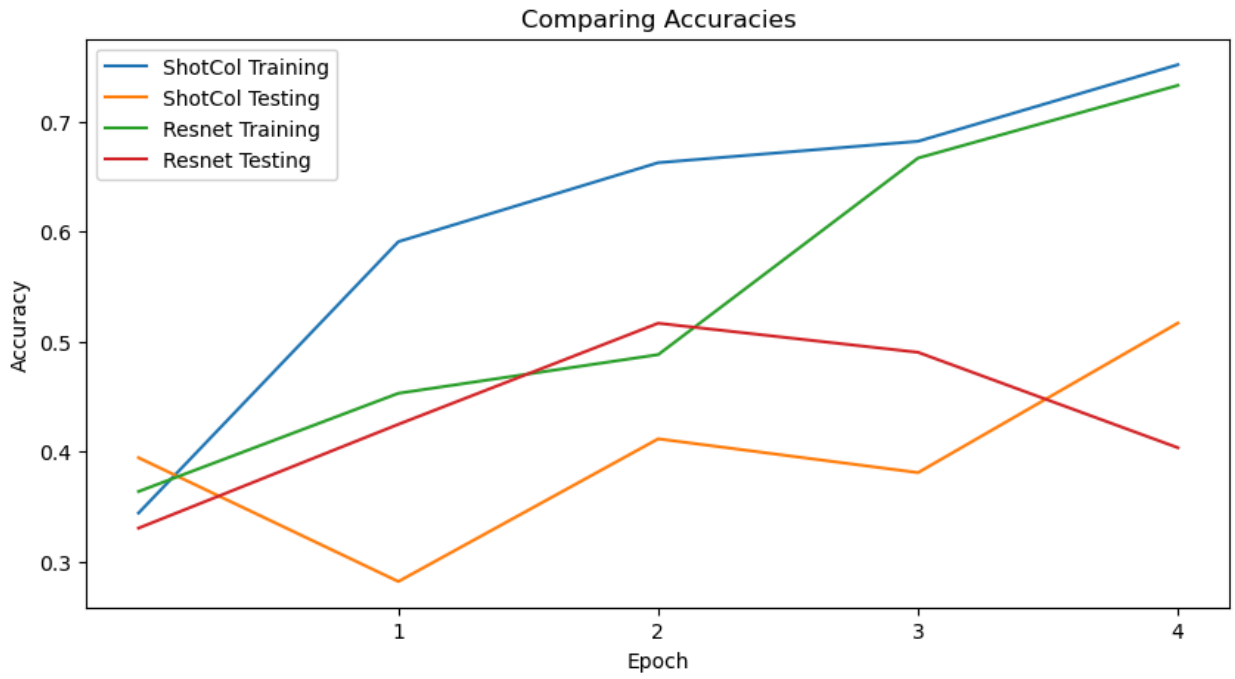




ShotCol Encoder Results:







In the general case, our model demonstrates higher accuracy across all transition types when compared to the baseline ResNet-50. This improvement reflects the effectiveness of our methodology, which incorporates targeted feature extraction using ShotCol and fine-tuning for ad/movie transitions, addressing the limitations of the base model's generic feature representations for ResNet-50 and the ShotCol approach.

From the confusion matrices, we can observe that our model outperforms the base ResNet-50 model in distinguishing the “Ad to Movie” transition. This improvement suggests that the combination of ShotCol learning with our shot-based approach enhances the model's ability to learn the unique features associated with this specific transition type. We hypothesize that this could be due to our model picking up on the text that some Ads have at the very end to be able to distinguish between this and other transitions.

The base ResNet-50 model often struggles to distinguish between "Movie to Ad" and "Ad to Movie" transitions, as evidenced by its frequent misclassifications in these categories. We believe this confusion arises because the generic ResNet-50 lacks the task-specific optimization necessary to differentiate the subtle contextual and visual cues that define these transitions. For example, both transitions may share overlapping visual features such as abrupt changes in color, scene composition, or pacing, making it difficult for a generic model to separate them effectively. Our implementation significantly improves upon this weakness by employing contrastive learning during the pre-task phase. The improved performance of our model demonstrates that the contrastive learning approach enables it to capture transition-specific features, such as distinct changes in visual tone or content structure, which are critical for accurate classification. This targeted learning process reduces ambiguity in the model's

decision-making, resulting in a notable decrease in misclassifications for these transition types compared to the baseline ResNet-50.

However, one persistent challenge is the low accuracy in identifying the “Ad to Ad” transition, which consistently lags behind other transition types. This weakness can be attributed to the way our dataset was constructed. Specifically, the dataset includes fewer examples of consecutive advertisements due to the inherent distribution of ads and movies in the source videos. Even with preprocessing steps, such as capping transition frequencies and balancing data where possible, the scarcity of “Ad to Ad” examples limits the model’s ability to learn these transitions effectively.

We hypothesize that this issue arises from two factors:

1. **Dataset Imbalance**: The lack of sufficient training samples for “Ad to Ad” transitions leads to poor generalization for this class.
2. **Feature Overlap**: Ads often share visual and stylistic similarities, such as repeated branding, similar color schemes, or consistent pacing, making it harder for the model to distinguish transitions within this category compared to transitions involving movies.

Our results demonstrate that the proposed model outperforms the base ResNet-50 in classifying ad and movie transitions, showcasing its effectiveness in tackling the specific challenges of this problem domain. Unlike the generic ResNet-50 model, which lacks optimization for the task of video transitions, our approach incorporates targeted learning and tailored preprocessing techniques that enhance its ability to capture shot-level distinctions.

Challenges

- **Implementing ShotCol**: The lack of code in the Amazon paper meant recreating the methodology from scratch, requiring significant debugging.
    - We realized early on that our problems were in some ways fundamentally different from the Amazon paper. They were trying to identify the boundary between scenes while we were trying to identify the boundary between ads. This meant we spent a lot of time figuring out how to feed movie and ad shots instead of just shots from different scenes while still retaining the main benefits of the model suggested by the paper.
- **Data Limitations**: DRM restrictions constrained the variety of movies, potentially limiting the generalizability of the model. For example, we were only able to train on older movies. This could have created biases that make our model not extensible to more modern movies.
- **Pipeline Complexity**: Constructing two pipelines and integrating them was time-intensive.
- **Compute Limitations**: Due to our limited compute resources we could not test more advanced MLP implementations or play around too much with our model.

- **Ad to Ad transitions**: Our dataset was imbalanced as it did not have too many Ad to Ad transitions. This could be fixed with collecting more ads.

Ethics

What broader societal issues are relevant to your chosen problem space?

Advertisement detection tools help users by giving them greater control over their viewing experience. By enabling features such as ad-skipping or automated removal of ads, these technologies respect user privacy/autonomy and help improve the quality of entertainment consumption. Ad detection tools question the ethics of advertising as a default content consumption model. Some argue that users should have the right to opt-out of ads, particularly invasive or poorly targeted ones. However, this challenges the implicit agreement between free access to content and the viewing of ads, creating a debate over fairness and user choice.

However, there are the ethical implications of disrupting advertising revenue models, which sustain many free content platforms such as YouTube. If ad blocking became prevalent this could impact the business of YouTube and its creators who depend on ad revenue for their livelihoods.

Advertising is a primary revenue stream for many free content platforms, such as YouTube, and creators depend on this income to sustain their work. Widespread adoption of ad-blocking technologies could destabilize this ecosystem, forcing platforms to explore alternative monetization models, such as subscription fees or sponsored content. This shift could limit access to free content, largely affecting users with fewer financial resources.

Small creators and marginalized groups may be particularly vulnerable. Unlike large corporations, independent creators rely heavily on ad revenue, and a decline in monetization options could exacerbate inequality in content creation, reducing the diversity of voices online.

What is your dataset? Are there any concerns about how it was collected, or labeled? Is it representative? What kind of underlying historical or societal biases might it contain?

Our dataset consists of two primary components: **movies** and **advertisements** collected from YouTube. The movies and ads were free-to-access content downloaded at 360p resolution. While we limited our data collection to free and publicly accessible content, copyright issues may still arise since the dataset could contain new movies which are still protected by copyright. Copyright is a huge aspect of this problem as collecting diverse data so that our model generalizes steps into the gray area of copyright law as we are not distributing the videos but just training on them. This legality is still being fought over in court by companies like OpenAI and New York TImes.

Division of Labor

Alan: ShotCol and data statistics/results analysis in Jupyter Notebook

Dagar: Shot segmentation, MLP, Data collection
Ade: Dataloader setup and model training
All: Slides design, final project report

There was a lot of communication for each of these parts, and pair programming. However, the person actually writing each part is outlined above.

Reflection

How do you feel your project ultimately turned out? How did you do relative to your base/target/stretch goals?

The project was overall kind of mid. I think we managed to hit our target, and produce something that trained and did better than our baseline of pure Resnet. However, there are some extensions of our project that we didn't manage to get to. For example, we wanted to use something more complicated than the MLP for shot classification. However, setting up the data loader ended up taking *a lot* of time. This is because we had to think a lot about how we wanted to feed the shots into our model and what that implied for model design.

Did your model work out the way you expected it to?

Overall, our model confirmed our hypotheses and the results produced by the Amazon paper. Using ShotCol as a pretext task does indeed increase performance in video understanding and segmentation.

How did your approach change over time? What kind of pivots did you make, if any? Would you have done differently if you could do your project over again?

We had to make a lot of compromises along the way. For example we wanted to make a deep learning model for shot segmentation, then we pivoted to trying an already implemented shot segmenter, to finally building our own. We had to make compromises on how much data we collected as a lot of movies were protected by DRM leading to comparatively less movies collected. Furthermore we thought of using MovieNet but later realized that they do not provide source movie files (e.g. MP4) so we pivoted to building our own dataset. If we had the choice to pick our project again we might have just done a different project entirely as this entire process was a pain. On the other hand if we had to redo this project then we would have spent our time more wisely and tried to build the LSTM.

What do you think you can further improve on if you had more time?

We could have used an LSTM to incorporate wider context of the movie when trying to detect ads. There would have been a tradeoff in model complexity and throughput, but we think it would have produced better accuracy. Overall, we think the hidden state in the LSTM could have contained information about past shots. We could even have two LSTMs. One LSTM could

keep track of the movie state. When the model reaches an ad, another LSTM starts and keeps track of state while inside an ad. The two models would then run in parallel to determine where the ad ends.

What are your biggest takeaways from this project/what did you learn?

The biggest takeaway from this project was that modifying and implementing a novel architecture is not an easy task. This type of project requires great communication between all team members and consistent effort put in by all of us from start to finish. We learned that making a model that focuses on video based input is incredibly difficult as it requires a lot of resources in terms of effort, communication, data collection, and more.

References

Bhattacharyya distance: From statistics to application in data science

Shot contrastive self-supervised learning for scene boundary detection