Advertisement Detection in Videos

Alan Wang, Dagar Rehan, Adebowale Adelekan

Introduction

We attempt to solve the problem of advertisement detection in videos, identifying the transitions between ads and movies. Some applications:

- Content Streaming - Seamlessly removing or skipping ads improves the user experience on streaming platforms (making a Ad blocker!).
- Video Understanding - Understanding ad transitions helps further the ability for AI models to build interpretable representations of video media

Data: Sources

71 movies from YouTube (~17651 shots/video)

- 360p resolution
- DRM restrictions on many videos (biased based on movie age)

176 advertisements from Youtube (~28 shots/video)

- 360p resolution.
- Biased towards the most popular ads

Methodology: Shot Segmentation

Bhattacharyya distance: Quantifies the similarity between two probability distributions (e.g. the color histograms of consecutive frames). A large change indicates a new shot:

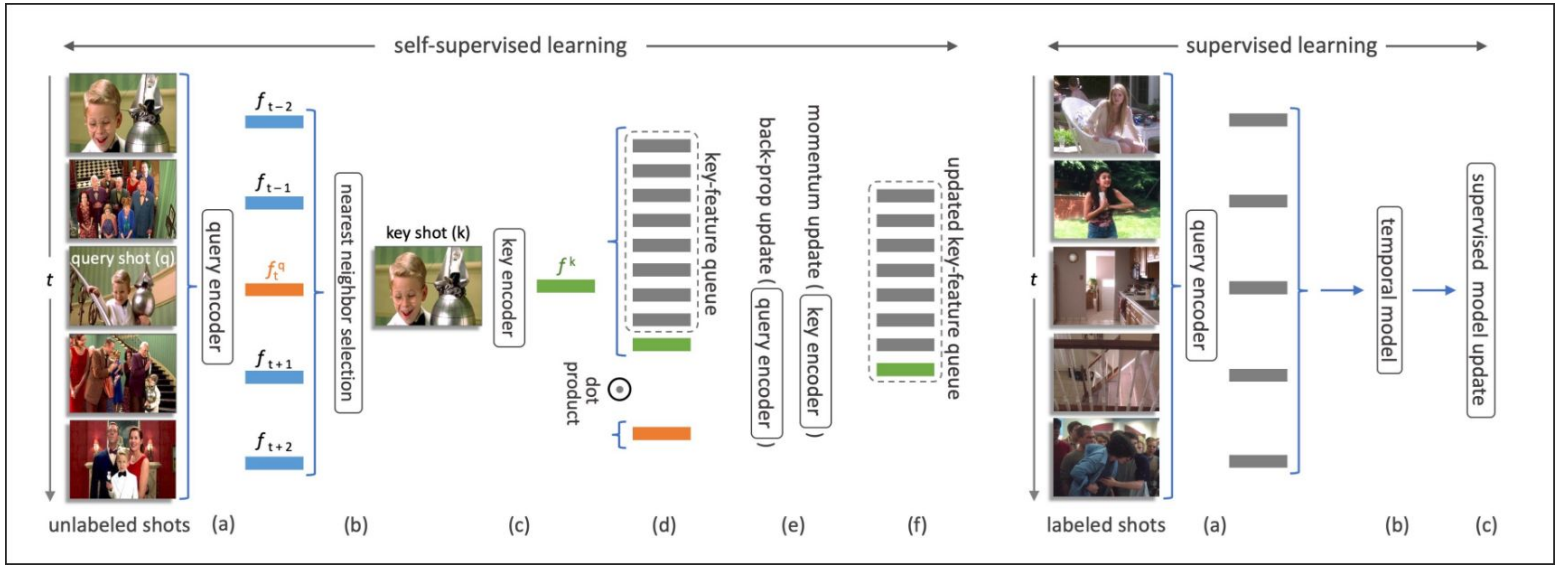
$$BD = \sqrt{(1 - \sum\sqrt{(p_1(i) * p_2(i))})}$$

$p_k(i)$ = Normalized histogram probabilities

Segmented shots serve as the fundamental building block for subsequent parts of our pipeline.

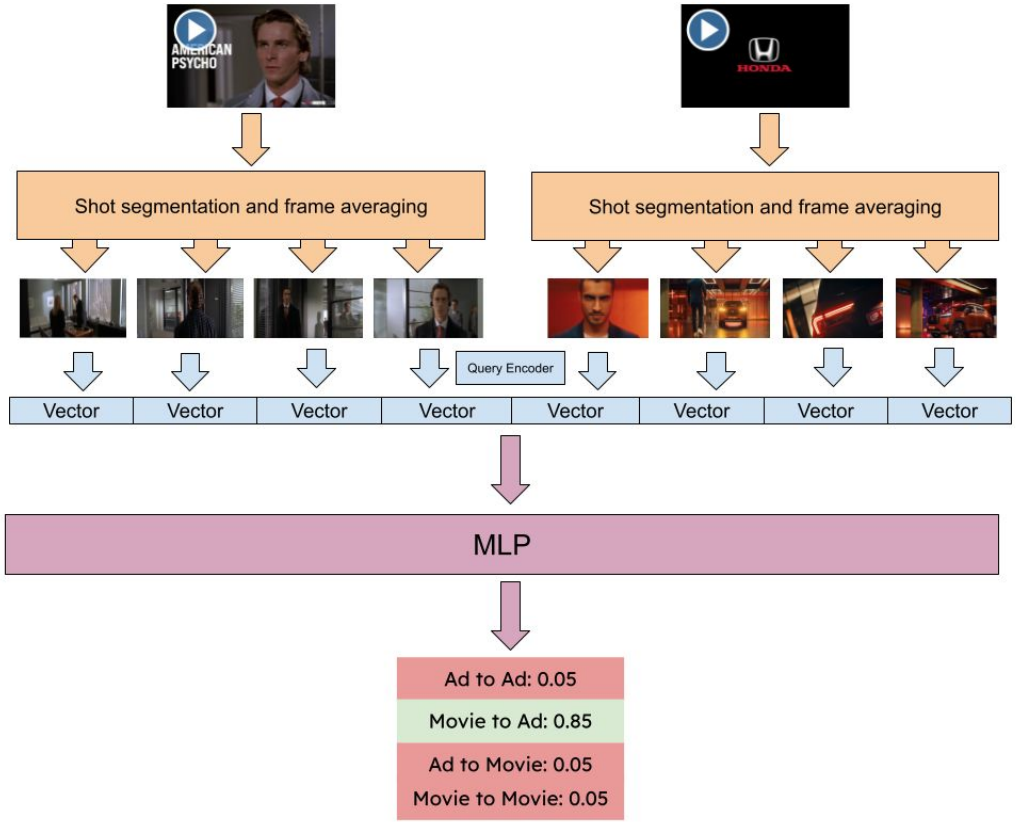
[Bhattacharyya distance: From statistics to application in data science]

Methodology: ShotCoL Pretext Task

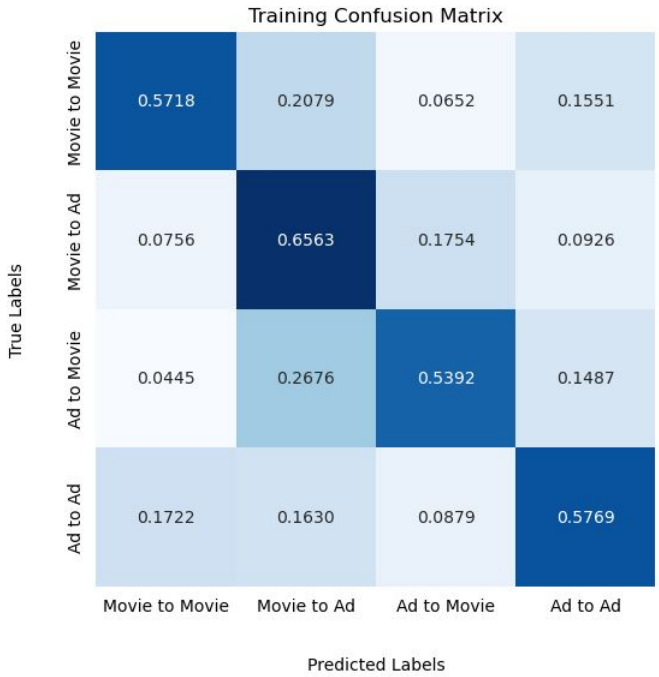


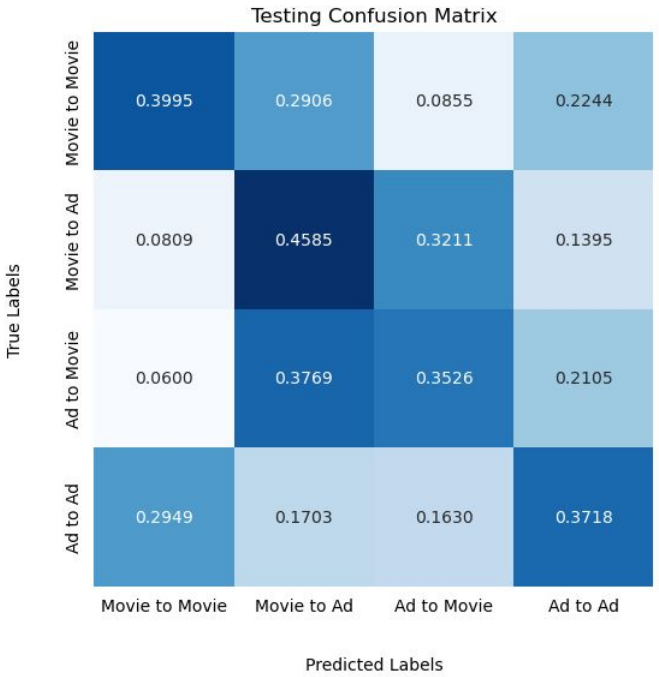
[Shot contrastive self-supervised learning for scene boundary detection]

Methodology: Transition Classification Model

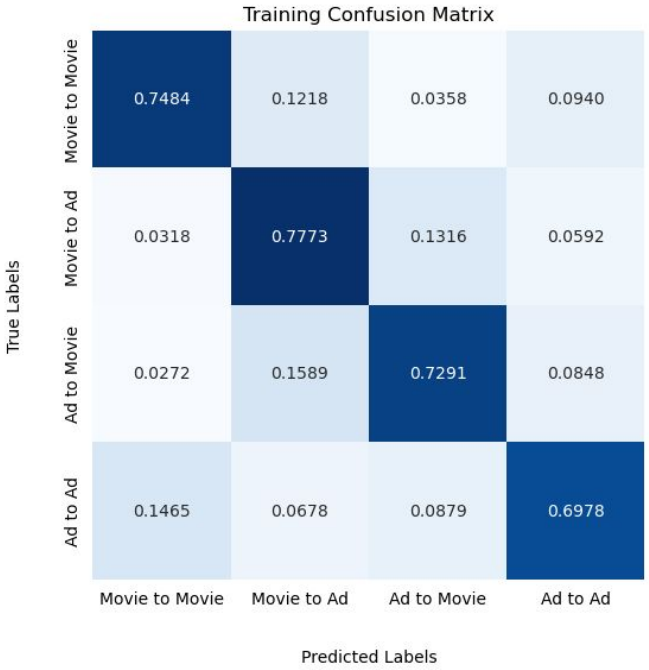


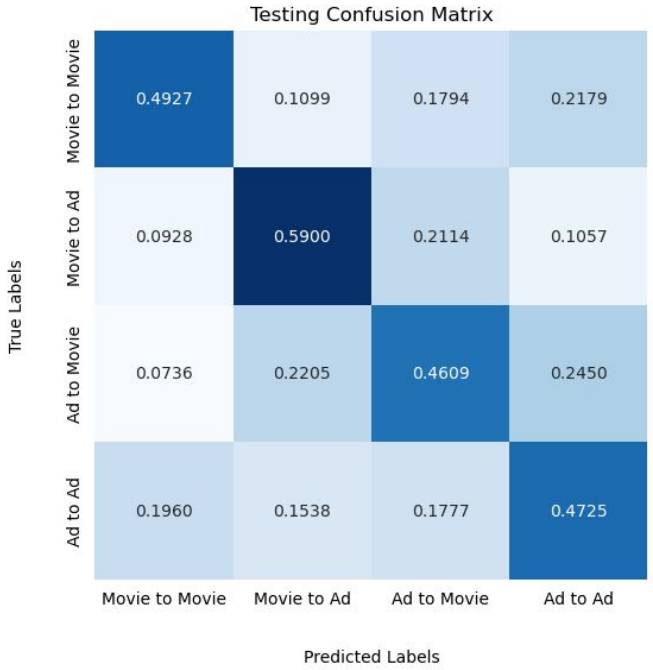
Results: Base ResNet50



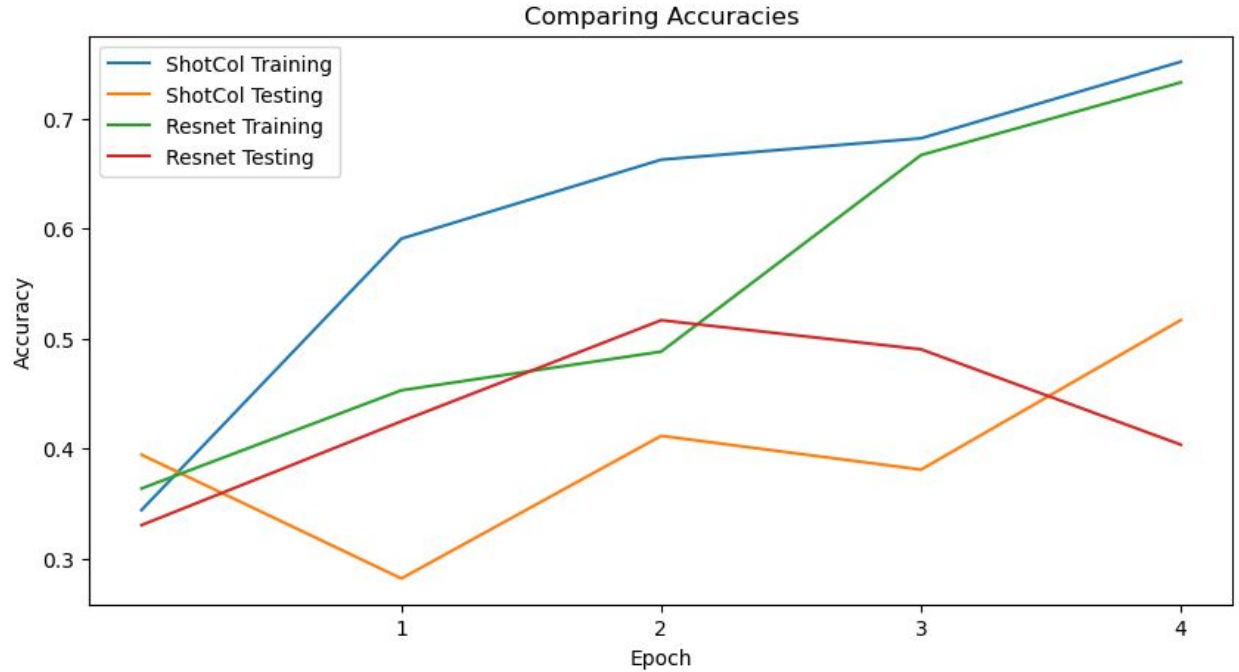


Results: Our Model





Accuracy Over Time



Discussion: Lessons Learned

- Preprocessing video data takes a long time
- Increasing batch size helps a lot with smoothing out model performance increments, especially when you have very diverse data (like movies and ads)
- Making the data loaders to handle video for ShotCol and supervised learning is a major obstacle
- Compute required to train model is very expensive in terms of money and time

Discussion: Further Work

- Use something more complex than an MLP for the classification task
  - Use an LSTM model instead of a direct MLP, which would pick up movie context better
  - Include audio
  - Gather more movies and ads
  - Experiment with different encoder
  - Make shot segmentation parallelizable