Recommendation Engine

Basic Recommendation Engines

- Text-based meta data analysis
    - Grouping based on keywords

- Collaborative Filtering
    - User-User Filtering
    - Item-Item Filtering

- Time-based Recommendation Engines
    - Removes low impact parameters for characteristics that more drastically impact users' moods.

Advanced Recommendation Engines

- Visual Content Based Recommendation Engines
    - Poster object detection helps tag & cluster images (Illustration2Vec Model)

Objective: To build a trailer-based recommendation engine that clusters & recommends based on top-k nearest neighbours (Essentially developing a Video2Vec Model)

Data Scraping

- Iterating through the imdb movie dataset & downloading .mp4 files of trailers.

Data Generation

- Splitting of the .mp4 files into K Iframes (based on the size of trailer) for inputting into model.

Model

Frame-based encoding & concatenating

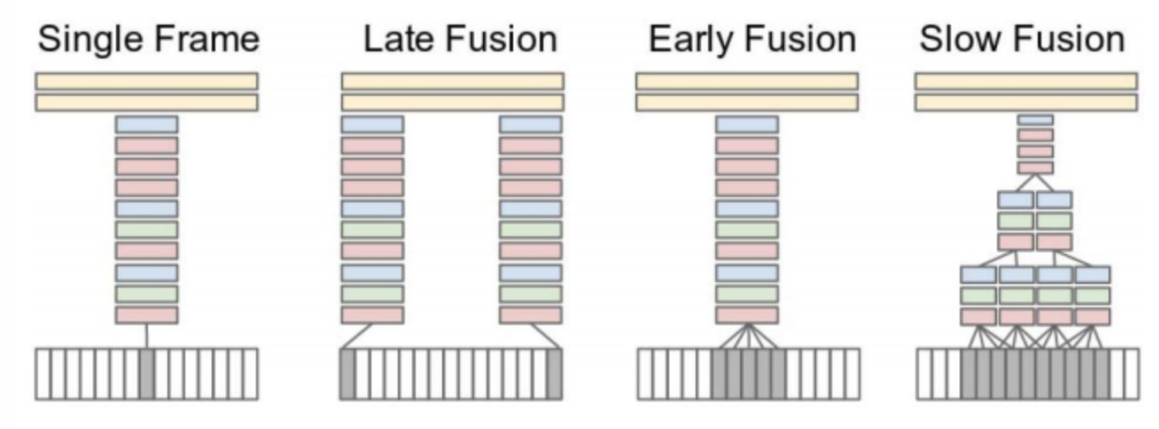
- Using autoencoders, encode each frame of the trailer (into size n encodings)
- Concatenate all frames' encodings into an (n * k) sized encoding
- Find max length of all encodings, and concatenate 0s to end of each encoding to make all encodings this max length.

Object Detection & Temporal Data Extraction

- Using an object detection model, extract lists of objects for each frame of the trailer.
- Feed the hidden layer encodings into a RNN for each frame, essentially converting the problem into a many-to-one model where the RNN will predict a final encoding.
- Map all these final encodings & perform Approximate Nearest Neighbour (ANN) search

Multiresolution CNNs

- 2 separate inputs are fed to separate convolutional layers which fuse together after 2 isolated sequences of Conv-MaxPool-BatchNorm, saving computation time for Convolutional Layers.
- In addition to the speed up, it also reports a small improvement over a Single-Frame model which takes in the original 178 x 178 frames.
- All videos are 0 padded at the ends such that all inputs are of the same length.
- Predictions (inputs into the CNN) are on crops of the videos and aggregated at the end.
- Types of Models
    - Single Frame
        - Aggregates predictions across single frames/images.
    - Late Fusion
        - Combines frames by concatenating the first and last frame in the clip.
    - Early Fusion
        - Takes a larger contiguous segment from the clip
    - Slow Fusion
        - Has a much more sophisticated scheme in which 4 partially overlapping contiguous segments are progressively combined in the Convolutional Layers



The best overall results were found by averaging results across all models, (Single + Early + Late + Slow).

Reference

https://towardsdatascience.com/introduction-to-video-classification-6c6acbc57356