Video Understanding

Sanmathi Kamath, Pranjali Kokare, Alekhya Munagala

Abstract— **With an increase in generation and consumption of videos, efficient retrieval of videos has become a challenging task. By providing multiple labels/categories to each video, we hope to develop a better video understanding, as a step towards solving this problem.**

I. Problem Statement

For huge amounts of unorganized video data, classifying and assigning labels/categories (popularly known as tags) to videos and improving video understanding can lead to better video search and discovery, improved video recommendations and pave way for efficient video retrieval.

The YouTube-8M challenge [1] provides a great source of data and inspires us to explore the area of Video Understanding. Given a video, the challenge is to provide multiple labels/categories that could be used to describe the video. Using this dataset, we aim to create a compact model for large-scale video understanding to tag each video with multiple labels.

II. Literature Review

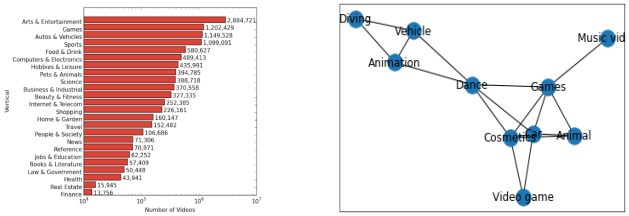
Miech et.al [2] investigates several learnable pooling techniques such as Bag-of visual-words, VLAD and Fisher Vector for aggregating into a 1024 dimensional compact representation to pass it to a soft Mixture-of-Experts (MoE) classifier. They also introduce Context Gating, which captures the non-linear inter dependencies between features as well as among output labels. In [3], Wang et.al proposes a deep learning architecture 'Chaining', which utilizes not only the input features but also predictions of labels from previous model. The final state of LSTM and the max-pooled feature map are used as feature representation in Chaining models and go through a MoE model. They also proposed attention weighted stacking, which uses an attention network to generate weights for pooling LSTM output. [4] focuses on temporal modelling approaches to aggregate the frame-level features that yield robust and discriminative video representation for further multi-label recognition. They propose, (1) two-stream sequence model: which trains two bidirectional LSTM and GRU models for audio and video features separately and then runs them through attention layers and concatenates them, (2) fast-forward sequence model: where the fast-forward connection takes the outputs of previous fast-forward and recurrent layer as input, and uses a fully-connected layer to embed them, (3) temporal residual neural networks: where temporal convolution neural networks are utilized to transform the original frame-level features into a more discriminative feature sequence, which can be further fed to LSTM. [5] proposes averaging video level features and aggregating frame level features into compact video-level representation to be fed to Mixture of Experts (MoE) model. To better leverage temporal information, they take difference between adjacent frame pairs as a feature transformation. They also take into account the label imbalance in the dataset and propose using label filters for models. The predictions from multiple model checkpoints are fused, followed by fusing the outputs of many such models.

III. Approach

A. Dataset Pre-Processing and Analysis

YouTube-8M is the largest multi-label video classification dataset, composed of 8 million videos—500K hours of video—annotated with a vocabulary of 4800 visual entities. Each video is between 100 to 500 seconds long.

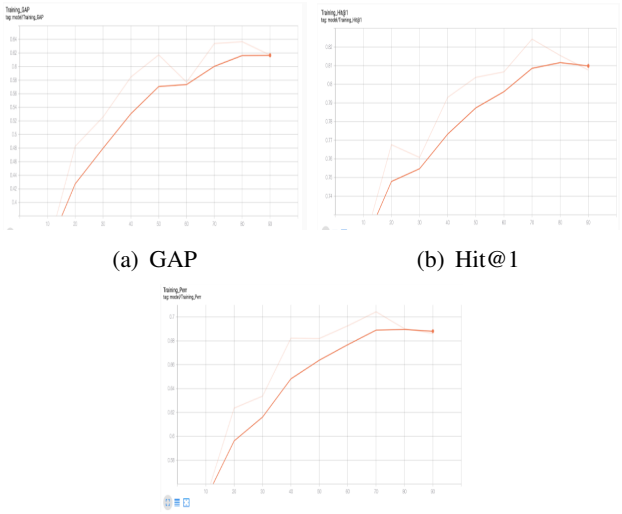
As full-size video datasets are impractical to work with, the YouTube-8M challenge dataset provides us with extracted features of the videos for frame-level and video level analysis. The frame-level features, extracted using an Inception network trained on ImageNet are 1024 dimensions per second. The video level features are mean of frame-level features across all frames [1].



(a) Distribution of top 24 most labelled visual entities in the YouTube-8M Dataset

(b) Inter-dependencies of 10 visual entities

Fig. 1. Plots for analysing dataset



(a) GAP

(b) Hit@1

(c) PERR

Fig. 2. Plots of performance metrics during training

When designing any machine learning algorithm, it is important to understand the structure and distribution of the data. We analysed the distribution and inter-dependencies of visual entities in the dataset. Figure 1(a) shows the number of videos of top 24 most labelled visual entities on a log scale. The map of the inter-dependencies on a subset of labels in the dataset are shown in Figure1(b).

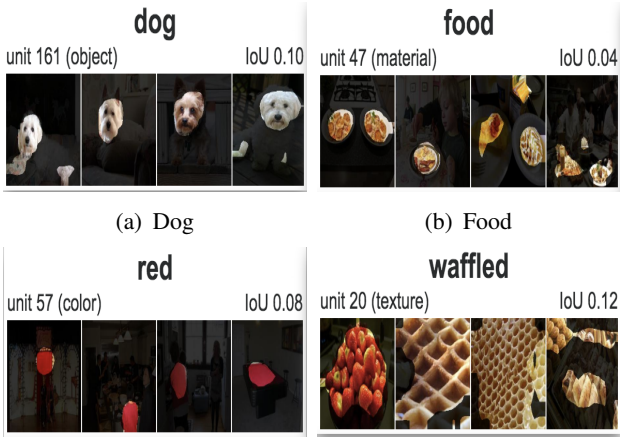
B. Progress

We started by working on the frame-level classifier. Each video is sampled to produce frame-level features. This leverages the fact that adjacent frames are highly correlated.

To get started with the data and training process, we trained a Fully Connected model with sigmoid non-linearity which works like a Logistic classifier. We trained a one-vs-all classifier for each of the labels on a subset of frame level data. For inference, the scores of each frame are aggregated over the video to predict top k labels for the video. To aggregate the scores, we implemented average pooling. Average pooling is preferred over max pooling to reduce the effect of outlier detection and capture prominence of each entity in the entire video.

We monitored our training process using the GAP, Hit@1 and PERR [V]. Figure 2 shows the performance plots obtained during training.

To get started with understanding the framework for analyzing deep learnt Convolutional Neural Networks, we experimented with the Network Dissection tool [6]. This tool compares activations generated by each unit of convolution layers with the masks for concepts of Broden dataset. IoU scores are used as a metric for estimating confidence in detecting different concepts [6]. Figure 3 show the results of dissection of AlexNet which was trained on ImageNet data.



(a) Dog

(b) Food

(c) Red

(d) Waffled

Fig. 3. Segments with high activations for different concepts

C. Next Steps

We are working on applying transfer learning with pre-trained models to build a frame-level neural network classifier.

Next, we wish to employ literature review and implement a Deep Neural Network Architecture for a video-level classifier to capture the temporal information. This will extract video-level features to predict the k labels with highest confidence scores for each video.

The next step would be to analyze our implemented models using the network dissection tools to understand how weights are being learnt and what features are being extracted by these models. Eventually, we would like to understand if the features extracted by the deep architectures can be encoded and utilized for solving our goal via lightweight architectures. At the same time, we would like to evaluate the trade-offs of using a lightweight architecture model using complex features vs using a deep neural network architecture for the same task.

IV. Validation of Implementation

Validating our implementation is heavily related to evaluating the performance of our models. Therefore, for this section, we will focus on answering how we will evaluate our code. The implemented model will generate a vector of tags for each given video. The correctness of these tags will be evaluated based on the evaluation metrics described in the following section.

V. Performance Evaluation

We have described our evaluation metrics below. As there are more than one correct labels for a given video, basic accuracy calculation is insufficient to measure the performance of our model.

- Hit@k:
  This is the fraction of test samples that contain at least one of the ground truth labels in the top k predictions.
- Global Average Precision (GAP):
  The evaluation takes the predicted labels that have the highest k confidence scores for each video, then treats each prediction and the confidence score as an individual data point in a long list of global predictions, to compute the Average Precision across all of the predictions and all the videos.[1]

$$GAP = \sum_{i=1}^{N} p(i)r(i) \tag{1}$$

  where N is the number of predictions, $p(i)$ is precision and $r(i)$ is recall.
- Precision at Equal Recall Rate(PERR):
  The PERR is the average precision of the top k scoring values across all videos,where k is the number of labels in the ground truth for that video.

VI. Resources

The resources that we will use are :

- Software: PyTorch, TensorFlow
- Hardware/Cloud: Google Cloud, Google Colab, GPU Cluster
- Dataset: YouTube-8M Dataset [1]

VII. Goals

The goals we aim for this project are as follows:

- ☐ 75%
  - ☑ Data Analysis and Pre-Processing
  - ☑ Review and Analyze existing model architectures
  - ☐ Implementing Frame-level Model
- ☐ 100%
  - ☐ Researching and Implementing Video-level model
  - ☐ Analysis and Evaluation of the implemented methods
- ☐ 125 %
  - ☐ Analyzing the weights learnt by deep neural network to explore the idea of using traditional machine learning models
  - ☐ Evaluating the trade-offs between model architecture and performance

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, Sudheendra Vijayanarasimhan, "YouTube-8M: A Large-Scale Video Classification Benchmark", arXiv:1609.08675

[2] Antoine Miech, Ivan Laptev, Josef Sivic, "Learnable pooling with context gating for video classification", arXiv:1706.06905

[3] He-Da Wang, Teng Zhang, "The Monkeytyping Solution to the YouTube-8M Video Understanding Challenge", arXiv:1706.05150

[4] Fu Li, Chuang Gan, Xiao Liu, Yunlong Bian, Xiang Long, Yandong Li, Zhichao Li, Jie Zhou, Shilei Wen, "Temporal Modeling Approaches for Large-scale Youtube-8M Video Understanding", arXiv:1707.04555

[5] Shaoxiang Chen, Xi Wang, Yongyi Tang, Xinpeng Chen, Zuxuan Wu, Yu-Gang Jiang, "Aggregating Frame-level Features for Large-Scale Video Classification", arXiv:1707.00803

[6] David Bau , Bolei Zhou , Aditya Khosla, Aude Oliva, and Antonio Torralba, "Network Dissection: Quantifying Interpretability of Deep Visual Representations", arXiv:1704.05796