# YouTube-8M: A Large-Scale Video Classification Benchmark

Sami Abu-El-Haija
haija@google.com

Nisarg Kothari
ndk@google.com

Joonseok Lee
joonseok@google.com

Paul Natsev
natsev@google.com

George Toderici
gtoderici@google.com

Balakrishnan Varadarajan
balakrishnanv@google.com

Sudheendra Vijayanarasimhan
svnaras@google.com

Google Research



## ABSTRACT

Many recent advancements in Computer Vision are attributed to large datasets. Open-source software packages for Machine Learning and inexpensive commodity hardware have reduced the barrier of entry for exploring novel approaches at scale. It is possible to train models over millions of examples within a few days. Although large-scale datasets exist for image understanding, such as ImageNet, there are no comparable size video classification datasets.

In this paper, we introduce **YouTube-8M**, the largest *multi-label video classification dataset*, composed of $\sim$8 million videos—500$K$ hours of video—annotated with a vocabulary of 4800 visual entities. To get the videos and their (multiple) labels, we used a YouTube video annotation system, which labels videos with the main topics in them. While the labels are machine-generated, they have high-precision and are derived from a variety of human-based signals including metadata and query click signals, so they represent an excellent target for *content-based annotation* approaches. We filtered the video labels (Knowledge Graph entities) using both automated and manual curation strategies, including asking human raters if the labels are *visually recognizable*. Then, we decoded each video at one-frame-per-second, and used a Deep CNN pre-trained on ImageNet to extract the hidden representation immediately prior to the classification layer. Finally, we compressed the frame features and make both the features and video-level labels available for download. The dataset contains frame-level features for over 1.9 billion video frames and 8 million videos, making it the largest public multi-label video dataset.

We trained various (modest) classification models on the dataset, evaluated them using popular evaluation metrics, and report them as baselines. Despite the size of the dataset, some of our models train to convergence in less than a day on a single machine using the publicly-available TensorFlow framework. We plan to release code for training a basic TensorFlow model and for computing metrics.

We show that pre-training on large data generalizes to other datasets like Sports-1M and ActivityNet. We achieve state-of-the-art on ActivityNet, improving mAP from 53.8% to 77.6%. We hope that the unprecedented scale and diversity of YouTube-8M will lead to advances in video understanding and representation learning.

## 1. INTRODUCTION

Large-scale datasets such as ImageNet [6] have been key enablers of recent progress in image understanding [20, 14, 11]. By supporting the learning process of deep networks with millions of parameters, such datasets have played a crucial role for the rapid progress of image understanding to near-human level accuracy [30]. Furthermore, intermediate layer activations of such networks have proven to be powerful and interpretable for various tasks beyond classification [41, 9, 31]. In a similar vein, the amount and size of video benchmarks is growing with the availability of Sports-1M [19] for sports videos and ActivityNet [12] for human activities. However, unlike ImageNet, which contains a diverse and general set of objects/entities, existing video benchmarks are restricted to action and sports classes.

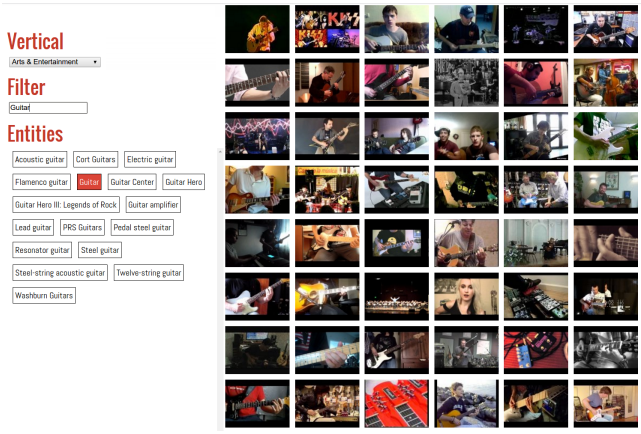



**Figure 1: YouTube-8M is a large-scale benchmark for general multi-label video classification. This screenshot of a dataset explorer depicts a subset of videos in the dataset annotated with the entity "Guitar". The dataset explorer allows browsing and searching of the full vocabulary of Knowledge Graph entities, grouped in 24 top-level verticals, along with corresponding videos.**

In this paper, we introduce YouTube-8M [1], a large-scale benchmark dataset for general *multi-label video classification*. We treat the task of video classification as that of producing labels that are relevant to a video given its frames. Therefore, unlike Sports-1M and ActivityNet, YouTube-8M is not restricted to action classes alone. For example, Figure 1 shows random video examples for the *Guitar* entity.

We first construct a *visual annotation vocabulary* from Knowledge Graph entities that appear as topic annotations for YouTube videos based on the YouTube video annotation system [2]. To ensure that our vocabulary consists of entities that are recognizable visually, we use various filtering criteria, including human raters. The entities in the dataset span activities (sports, games, hobbies), objects (autos, food, products), scenes (travel), and events. The

[1] http://research.google.com/youtube8m

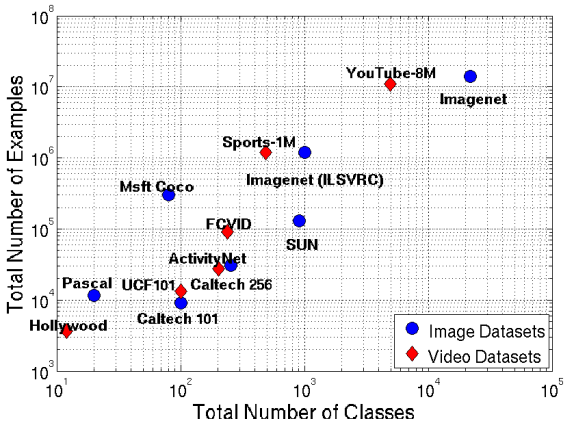



**Figure 2: The progression of datasets for image and video understanding tasks. Large datasets have played a key role for advances in both areas.**

entities were selected using a combination of their popularity on YouTube and manual ratings of their *visualness* according to human raters. They are an attempt to describe the central themes of videos using a few succinct labels.

We then collect a sample set of videos for each entity, and use a publicly available state-of-the-art Inception network [4] to extract features from them. Specifically, we decode videos at one frame-per-second and extract the last hidden representation before the classification layer for each frame. We compress the frame-level features and make them available on our website for download.

Overall, YouTube-8M contains more than 8 million videos—over 500,000 hours of video—from 4,800 classes. Figure 2 illustrates the scale of YouTube-8M, compared to existing image and video datasets. We hope that the unprecedented scale and diversity of this dataset will be a useful resource for developing advanced video understanding and representation learning techniques.

Towards this end, we provide extensive experiments comparing several state-of-the-art techniques for video representation learning, including Deep Networks [26], and LSTMs (Long Short-Term Memory Networks) [13] on this dataset. In addition, we show that transfering video feature representations learned on this dataset leads to significant improvements on other benchmarks such as Sports-1M and ActivityNet.

In the rest of the paper, we first review existing benchmarks for image and video classification in Section 2. We present the details of our dataset including the collection process and a brief analysis of the categories and videos in Section 3. In Section 4, we review several approaches for the task of multi-label video classification given fixed frame-level features, and evaluate the approaches on the dataset. In Section 5, we show that features and models learned on our large-scale dataset generalize very well on other benchmarks. We offer concluding remarks with Section 6.

## 2. RELATED WORK

Image benchmarks have played a significant role in advancing computer vision algorithms for image understanding. Starting from a number of well labeled small-scale datasets such as Caltech 101/256 [8, 10], MSRC [32], PASCAL [7], image understanding research has rapidly advanced to utilizing larger datasets such as ImageNet [6] and SUN [38] for the next generation of vision algorithms. ImageNet in particular has enabled the development of deep feature learning techniques with millions of parameters such as the AlexNet [20] and Inception [14] architectures due to the number of classes (21841), the diversity of the classes (27 top-level categories) and the millions of labeled images available.

A similar effort is in progress in the video understanding domain where the community has quickly progressed from small, well-labeled datasets such as KTH [22], Hollywood 2 [23], Weizmann [5], with a few thousand video clips, to medium-scale datasets such as UCF101 [33], Thumos'14 [16] and HMDB51 [21], with more than 50 action categories. Currently, the largest available video benchmarks are the Sports-1M [19], with 487 sports related activities and 1M videos, the YFCC-100M [34], with 800K videos and raw metadata (titles, descriptions, tags) for some of them, the FCVID [17] dataset of 91,223 videos manually annotated with 239 categories, and ActivityNet [12], with ∼200 human activity classes and a few thousand videos. However, almost all current video benchmarks are restricted to recognizing action and activity categories, and have less than 500 categories.

YouTube-8M fills the gap in video benchmarks as follows:

- A **large-scale video annotation and representation learning benchmark**, reflecting the main themes of a video.
- A significant jump in the number and diversity of annotation classes—**4800 Knowledge Graph entities** vs. less than 500 categories for all other datasets.
- A substantial increase in the number of labeled videos—over **8 million videos, more than 500,000 hours of video**.
- Availability of **pre-computed state-of-the-art features for 1.9 billion video frames**.

We hope the pre-computed features will remove computational barriers, level the playing field, and enable researchers to explore new technologies in the video domain at an unprecedented scale.

## 3. YOUTUBE-8M DATASET

YouTube-8M is a benchmark dataset for video understanding, where the main task is to determine the key topical themes of a video. We start with YouTube videos since they are a good (albeit noisy) source of knowledge for diverse categories including various sports, activities, animals, foods, products, tourist attractions, games, and many more. We use the YouTube video annotation system [2] to obtain topic annotations for a video, and to retrieve videos for a given topic. The annotations are provided in the form of Knowledge Graph entities [3] (formerly, Freebase topics [1]). They are associated with each video based on the video's metadata, context, and content signals [2].

We use Knowledge Graph entities to succinctly describe the *main themes* of a video. For example, a video of biking on dirt roads and cliffs would have a central topic/theme of *Mountain Biking*, not *Dirt*, *Road*, *Person*, *Sky*, and so on. Therefore, the aim of the dataset is not only to understand what is present in each frame of the video, but also to identify the few key topics that best describe what the video is about. Note that this is different than typical event or scene recognition tasks, where each item belongs to a single event or scene. [38, 28] It is also different than most object recognition tasks, where the goal is to label everything visible in an image. This would produce thousands of labels on each video but without answering what the video is really about. The goal of this benchmark is to understand what is in the video and to summarize that into a few key topics. In the following sub-sections, we describe our vocabulary and video selection scheme, followed by a brief summary of dataset statistics.


**Figure 3: A tag-cloud representation of the top 200 entities. Font size is proportional to the number of videos labeled with the entity.**

| Top-level Category | 1st Entity | 2nd Entity | 3rd Entity | 4th Entity | 5th Entity | 6th Entity | 7th Entity |
|---|---|---|---|---|---|---|---|
| **Arts & Entertainment** | Concert | Animation | Music video | Dance | Guitar | Disc jockey | Trailer |
| **Autos & Vehicles** | Vehicle | Car | Motorcycle | Bicycle | Aircraft | Truck | Boat |
| **Beauty & Fitness** | Fashion | Hair | Cosmetics | Weight training | Hairstyle | Nail | Mascara |
| **Books & Literature** | Book | Harry Potter | The Bible | Writing | Magazine | Alice | E-book |
| **Business & Industrial** | Train | Model aircraft | Fish | Water | Tractor pulling | Advertising | Landing |
| **Computers & Electronics** | Personal computer | Video game console | iPhone | PlayStation 3 | Tablet computer | Xbox 360 | Microsoft Windows |
| **Finance** | Money | Bank | Foreign Exchange | Euro | United States Dollar | Credit card | Cash |
| **Food & Drink** | Food | Cooking | Recipe | Cake | Chocolate | Egg | Eating |
| **Games** | Video game | Minecraft | Action-adventure game | Strategy video game | Sports game | Call of Duty | Grand Theft Auto V |
| **Health** | Medicine | Raw food | Ear | Glasses | Injury | Dietary supplement | Dental braces |
| **Hobbies & Leisure** | Fishing | Outdoor recreation | Radio-controlled model | Wedding | Christmas | Hunting | Diving |
| **Home & Garden** | Gardening | Home improvement | House | Kitchen | Garden | Door | Swimming pool |
| **Internet & Telecom** | Mobile phone | Smartphone | Telephone | Website | Sony Xperia | Google Nexus | World Wide Web |
| **Jobs & Education** | School | University | High school | Teacher | Kindergarten | Campus | Classroom |
| **Law & Government** | Tank | Firefighter | President of the U.S.A. | Soldier | President | Police officer | Fighter aircraft |
| **News** | Weather | Snow | Rain | News broadcasting | Newspaper | Mattel | Hail |
| **People & Society** | Prayer | Family | Play-Doh | Human | Dragon | Angel | Tarot |
| **Pets & Animals** | Animal | Dog | Horse | Cat | Bird | Aquarium | Puppy |
| **Real Estate** | House | Apartment | Condominium | Dormitory | Mansion | Skyscraper | Loft |
| **Reference** | Vampire | Bus | River | City | Mermaid | Village | Samurai |
| **Science** | Nature | Robot | Eye | Ice | Biology | Skin | Light |
| **Shopping** | Toy | LEGO | Sledding | Doll | Shoe | My Little Pony | Nike; Inc. |
| **Sports** | Motorsport | Football | Winter sport | Cycling | Basketball | Gymnastics | Wrestling |
| **Travel** | Amusement park | Hotel | Airport | Beach | Roller coaster | Lake | Resort |
| **Full vocabulary** | **Vehicle** | **Concert** | **Animation** | **Music video** | **Video game** | **Motorsport** | **Football** |

**Table 1: Most frequent entities for each of the top-level categories.**

## 3.1 Vocabulary Construction

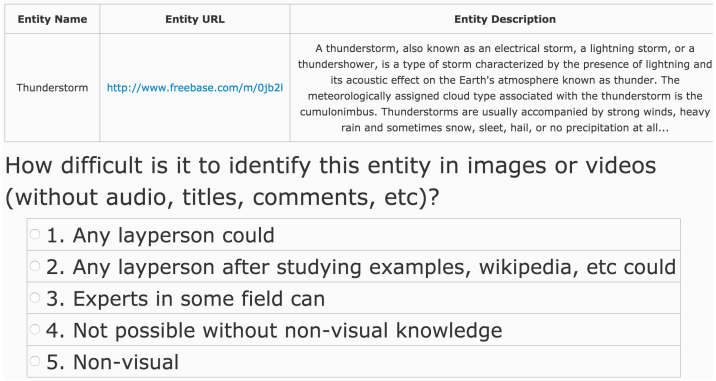
We followed two main tenets when designing the vocabulary for the dataset; namely 1) every label in the dataset should be distinguishable using visual information alone, and 2) each label should have sufficient number of videos for training models and for computing reliable metrics on the test set. For the former, we used a combination of manually curated topics and human ratings to prune the vocabulary into a visual set. For the latter, we considered only entities having at least 200 videos in the dataset.

The Knowledge Graph contains millions of topics. Each topic has one or more *types*, that are curated with high precision. For example, there is an exhaustive list of animals with type *animal* and an exhaustive list of foods with type *food*. To start with our initial vocabulary, we manually selected a whitelist of 25 entity types that we considered visual (e.g. *sport*, *tourist_attraction*, *inventions*), and also blacklisted types that we thought are non-visual (e.g. *music artists*, *music compositions*, *album*, *software*). We then obtained all entities that have at least one whitelisted type and no blacklisted types, which resulted in an initial vocabulary of $\sim 50,000$ entities.

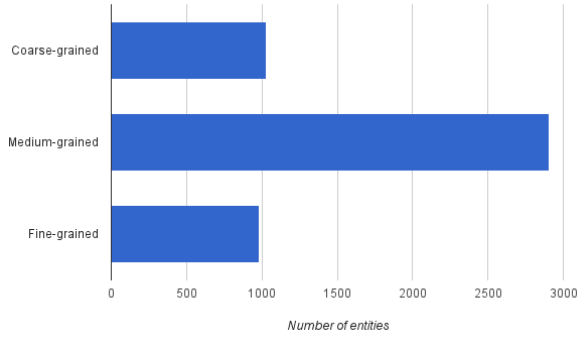
Following this, we used human raters in order to manually prune this set into a smaller set of entities that are considered visual with high confidence, and are also recognizable without very deep domain expertise. Raters were provided with instructions and examples. Each entity was rated by 3 raters and the ratings were averaged. Figure 4a shows the main rating question. The process resulted in a total of $\sim 10,000$ entities that are considered visually recognizable and are not too fine-grained (i.e. can be recognized by non-domain experts after studying some examples). These entities were further pruned: we only kept entities that have more than 200 popular videos, as explained in the next section. The final set of entities in the dataset are fairly balanced in terms of the specificity of the topic they describe, and span both coarse-grained and fine-grained entities, as shown in Figure 4b.

## 3.2 Collecting Videos

Having established the initial target vocabulary, we followed these




**(a) Screenshot of the question displayed to human raters.**




**(b) Distribution of vocabulary topics in terms of specificity.**

**Figure 4: Rater guidelines to assess how specific and visually recognizable each entity is, on a discrete scale of (1 to 5), where 1 is most visual and easily recognizable by a layperson. Each entity was rated by 3 raters. We kept only entities with a maximum average score of 2.5, and categorized them by specificity, into coarse-grained, medium-grained, and fine-grained entities, using equally sized score range buckets.**

steps to obtain the videos:

- Collected all videos corresponding to the $10,000$ visual entities and have at least $1,000$ views, using the YouTube video annotation system [2]. We excluded too short ($< 120$ secs) or too long ($> 500$ secs) videos.
- Randomly sampled 10 million videos among them.
- Obtained all entities for the sampled 10 million videos using the YouTube video annotation system. This completes the annotations.
- Filtered out entities with less than 200 videos, and videos with no remaining entities. This reduced the size of our data to $8,264,650$ videos.
- Split our videos into 3 partitions, *Train* : *Validate* : *Test*, with ratios $70\% : 20\% : 10\%$. We publish features for all splits, but only publish labels for the *Train* and *Validate* partitions.

## 3.3 Features

The original size of the video dataset is hundreds of Terabytes, and covers over $500,000$ hours of video. This is impractical to process by most research teams (using a real-time video processing engine, it would take over 50 years to go through the data). Therefore, we pre-process the videos and extract frame-level features using a state-of-the-art deep model: the publicly available Inception network [4] trained on ImageNet [14]. Concretely, we decode each video at 1 frame-per-second up to the first 360 seconds (6 minutes), feed the decoded frames into the Inception network, and fetch the ReLu activation of the last hidden layer, before the classification layer (layer name `pool_3/_reshape`). The feature vector is 2048-dimensional per second of video. While this removes motion information from the videos, recent work shows diminishing returns from motion features as the size and diversity of the video data increases [26, 35]. The static frame-level features provide an excellent baseline, and constructing compact and efficient motion features is beyond the scope of this paper. Nonetheless, we hope to extend the dataset with audio and motion features in the future. We cap processing of each video up to the first 360 seconds for storage and computational reasons. For comparison, the average length of videos in UCF-101 is $10 - 15$ seconds, Sports-1M is 336 seconds and in this dataset, it is 230 seconds.

| Dataset | Train | Validate | Test | Total |
|---|---|---|---|---|
| YouTube-8M | 5,786,881 | 1,652,167 | 825,602 | 8,264,650 |

**Table 2: Dataset partition sizes.**

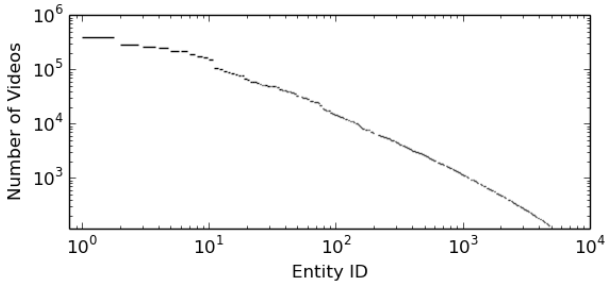


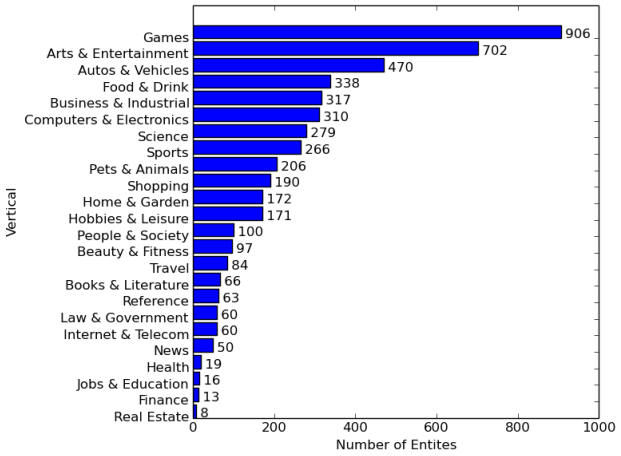

**Figure 5: Number of videos in log-scale versus entity rank in log scale. Entities were sorted by number of videos. We note that this somewhat follows the natural Zipf distribution.**

Afterwards, we apply PCA (+ whitening) to reduce feature dimensions to 1024, followed by quantization (1 byte per coefficient). These two compression techniques reduce the size of the data by a factor of 8. The mean vector and covariance matrix for PCA was computed on all frames from the *Train* partition. We quantize each 32-bit float into 256 distinct values (8 bits) using optimally computed (non-uniform) quantization bin boundaries. We confirmed that the size reduction does not significantly hurt the evaluation metrics. In fact, training all baselines on the full-size data (8 times larger than what we publish), increases all evaluation metrics by less than $1\%$.

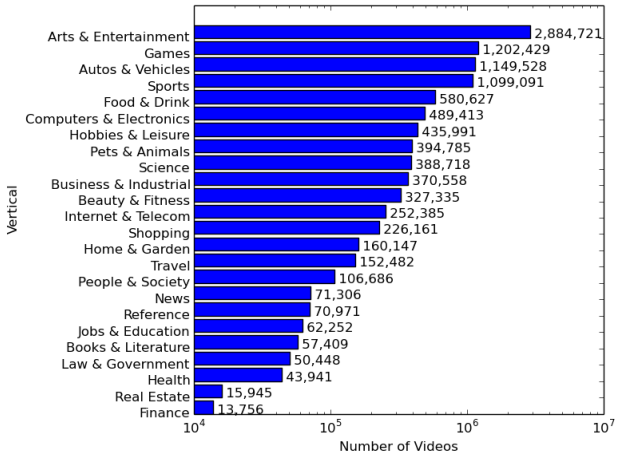
Note that while this dataset comes with standard frame-level features, it leaves a lot of room for investigating video representation learning approaches on top of the fixed frame-level features (see Section 4 for approaches we explored).

## 3.4 Dataset Statistics

The YouTube-8M dataset contains $4,800$ classes and a total of




**(a) Number of entities in each top-level category.**




**(b) Number of *train* videos in log-scale per top-level category.**

**Figure 6: Top-level category statistics of the YouTube-8M dataset.**

$8,264,650$ videos. A video may be annotated with more than one class and the average number of classes per video is 1.8. Table 2 shows the number of videos for which we are releasing features, across the three datasets.

We processed only the first six minutes of each video, at 1 frame-per-second. The average length of a video in the dataset is 229.6 seconds, which amounts to $\sim$1.9 billion frames (and corresponding features) across the dataset.

We grouped the $4,800$ entities into 24 top-level categories to measure statistics and illustrate diversity. Although we do not use these categories during training, we are releasing the entity-to-category mapping for completeness. Table 1 shows the top entities per category. Note that while some categories themselves may not seem visual, most of the entities within them are visual. For instance, Jobs & Education includes universities, classrooms, lectures, etc., and Law & Government includes police, emergency vehicles, military-related entities, which are well represented and visual.

Figure 5 shows a log-log scale distribution of entities and videos. Figures 6a and 6b show the size of categories, respectively, in terms of the number of entities and the number of videos.

## 3.5 Human Rated Test Set

The annotations from the YouTube video annotation system can be noisy and incomplete, as they are automatically generated from metadata, anchor text, comments, and user engagement signals [2]. To quantify the noise, we uniformly sampled over 8000 videos from the *Test* partition, and used 3 human raters per video to exhaustively rate their labels. We measured the precision and recall of the ground truth labels to be $78.8\%$ and $14.5\%$, respectively, with respect to the human raters. Note that typical inter-rater agreement on similar annotation tasks with human raters is also around 80% so the precision of these ground truth labels is perhaps comparable to (non-expert) human-provided labels. The recall, however, is low, which makes this an excellent test bed for approaches that deal with missing data. We report the accuracy of our models primarily on the (noisy) *Validate* partition but also show some results on the much smaller human-rated set, showing that some of the metrics are surprisingly similar on the two datasets.

While the baselines in section 4 show very promising results, we believe that they can be significantly improved (when evaluated on the human-based ground truth), if one explicitly models incorrect [29] ($78.8\%$ precision) or missing [40, 25] ($14.5\%$ recall) training labels. We believe this is an exciting area of research that this dataset will enable at scale.

# 4. BASELINE APPROACHES

## 4.1 Models from Frame Features

One of the challenges with this dataset is that we only have video-level ground-truth labels. We do not have any additional information that specifies how the labels are localized within the video, nor their relative prominence in the video, yet we want to infer their importance for the full video. In this section, we consider models trained to predict the main themes of the video using the input frame-level features. Frame-level models have shown competitive performance for video-level tasks in previous work [19, 26]. A video $v$ is given by a sequence of frame-level features $\mathbf{x}^v_{1:F_v}$, where $\mathbf{x}^v_j$ is the feature of the $j^{th}$ frame from video $v$.

### *4.1.1 Frame-Level Models and Average Pooling*

Since we do not have frame-level ground-truth, we assign the video-level ground-truth to every frame within that video. More sophisticated formulations based on multiple-instance learning are left for future work. From each video, we sample 20 random frames and associate all frames to the video-level ground-truth. This results in about 120 million frames. For each entity $e$, we get $120M$ instances of $(\mathbf{x}_i, y^e_i)$ pairs, where $\mathbf{x}_i \in \mathbb{R}^{1024}$ is the inception feature and $y^e_i \in 0, 1$ is the ground-truth associated with entity $e$ for the $i^{th}$ example. We train 4800 independent one-vs-all classifiers for each entity $e$. We use the online training framework after parallelizing the work for each entity across multiple workers. During inference, we score every frame in the test video using the models for all classes. Since all our evaluations are based on video-level ground truths, we need to aggregate the frame-level scores (for each entity) to a single video-level score. The frame-level probabilities are aggregated to the video-level using a simple average. We choose average instead of max pooling since we want to reduce the effect of outlier detections and capture the prominence of each entity in the entire video. In other words, let $p(e|\mathbf{x})$ be the probability of existence of $e$ given the features $\mathbf{x}$. We compute the probability

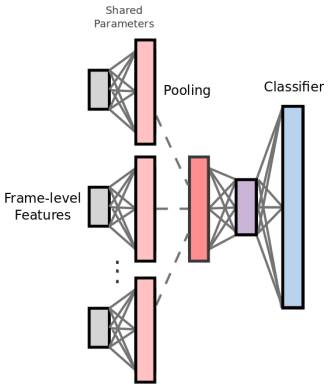



**Figure 7: The network architecture of the DBoF approach. Input frame features are first fed into a up-projection layer with shared parameters for all frames. This is followed by a pooling layer that converts the frame-level sparse codes into a video-level representation. A few hidden layers and a classification layer provide the final video-level predictions.**

$p_v(e|\mathbf{x}^v_{1:F_v})$ of the entity $e$ associated with the video $v$ as

$$p_v(e|\mathbf{x}^v_{1:F_v}) = \frac{1}{F_v}\sum_{j=1}^{F_v} p(e|\mathbf{x}^v_j). \quad (1)$$

#### 4.1.2 *Deep Bag of Frame (DBoF) Pooling*

Inspired by the success of various classic bag of words representations for video classification [23, 36], we next consider a Deep Bag-of-Frames (DBoF) approach. Figure 7 shows the overall architecture of our DBoF network for video classification. The $N$-dimensional input frame level features from $k$ randomly selected frames of a video are first fed into a fully connected layer of $M$ units with RELU activations. Typically, with $M > N$, the input features are projected onto a higher dimensional space. Crucially, the parameters of the fully connected layer are shared across the $k$ input frames. Along with the RELU activation, this leads to a sparse coding of the input features in the $M$-dimensional space.

The obtained sparse codes are fed into a pooling layer that aggregates the codes of the $k$ frames into a single fixed-length video representation. We use max pooling to perform the aggregation. We use a batch normalization layer before pooling to improve stability and speed-up convergence. The obtained fixed length descriptor of the video can now be classified into the output classes using a Logistic or Softmax layer with additional fully connected layers in between. The $M$-dimensions of the projection layer could be thought of as $M$ discriminative clusters which can be trained in a single network end to end using backpropagation.

The entire network is trained using Stocastic Gradient Descent (SGD) with logistic loss for a logistic layer and cross-entropy loss for a softmax layer. The backpropagated gradients from the top layer train the weight vectors of the projection layer in a discriminative fashion in order to provide a powerful representation of the input bag of features. A similar network was proposed in [26] where the convolutional layer outputs are pooled across all the frames of a video to obtain a fixed length descriptor. However, the network in [26] does not use an intermediate projection layer which we found to be a crucial difference when learning from input frame features. Note that the up-projection layer into sparse codes is similar to what Fisher Vectors [27] and VLAD [15] approaches do but the projection (i.e., clustering) is done discriminatively here. We also experimented with Fisher Vectors and VLAD but were not able to obtain competitive results using comparable codebook sizes.

**Hyperparameters:** We considered values of $\{2048, 4096, 8192\}$ for the number of units in the projection layer of the network and found that larger values lead to better results. We used $8192$ for all datasets. We used a single hidden layer with $1024$ units between the pooling layer and the final classification layer in all experiments. The network was trained using SGD with AdaGrad, a learning rate of $0.1$, and a weight decay penalty of $0.0005$.

#### 4.1.3 *Long Short-Term Memory (LSTM)*

We take a similar approach to [26] to utilize LSTMs for video-level prediction. However, unlike that work, we do not have access to the raw video frames. This means that we can only train the LSTM and Softmax layers.

We experimented with the number of stacked LSTM layers and the number of hidden units. We empirically found that 2 layers with 1024 units provided the highest performance on the validation set. Similarly to [26], we also employ linearly increasing per-frame weights going from $1/N$ to 1 for the last frame.

During the training time, the LSTM was unrolled for 60 iterations. Therefore, the gradient horizon for LSTM was 60 seconds. We experimented with a larger number of unroll iterations, but that slowed down the training process considerably. In the end, the best model was the one trained for the largest number of steps (rather than the most real time).

In order to transfer the learned model to ActivityNet, we used a fully-connected model which uses as inputs the concatenation of the LSTM layers' outputs as computed at the last frame of the videos in each of these two benchmarks. Unlike traditional transfer learning methods, we do not fine-tune the LSTM layers. This approach is more robust to overfitting than traditional methods, which is crucial for obtaining competitive performance on ActivityNet due to its size. We did perform full fine-tuning experiments on Sports-1M, which is large enough to fine-tune the entire LSTM model after pre-training.

### 4.2 Video level representations

Instead of training classifiers directly on frame-level features, we also explore extracting a task-independent fixed-length video-level feature vector from the frame-level features $\mathbf{x}^v_{1:F_v}$ for each video $v$. There are several benefits of extracting fixed-length video features:

1. **Standard classifiers can apply:** Since the dimensionality of the representations are fixed across videos, we may train standard classifiers like logistic, SVM, mixture of experts.
2. **Compactness:** We get a compact representation for the entire video, thereby reducing the training data size by a few orders of magnitude.
3. **More suitable for domain adaptation:** Since the video-level representations are unsupervised (extracted independently of the labels), these representations are far less specialized to the labels associated with the current dataset, and can generalize better to new tasks or video domains.

Formally, a video-level feature $\varphi(\mathbf{x}^v_{1:F_v})$ is a fixed-length representation (at the video-level). We explore a simple aggregation technique for getting these video-level representations. We also experimented with Fisher Vectors (FV) [27] and VLAD [15] approaches for task-independent video-level representations but were not able to achieve competitive results for FV or VLAD representations of similar dimensionality. We leave it as future work to come up with compact FV or VLAD type representations that outperform the much simpler approach described below.

#### 4.2.1 First, second order and ordinal statistics

From the frame-level features $\mathbf{x}^v_{1:F_v}$ where $\mathbf{x}^v_j \in \mathbb{R}^{1024}$, we extract the mean $\mu^v \in \mathbb{R}^{1024}$ and the standard-deviation $\sigma^v \in \mathbb{R}^{1024}$. Additionally, we also extract the top 5 ordinal statistics for each dimension. Formally, $\text{Top}_K(\mathbf{x}^v(j)_{1:F_v})$ returns a $K$ dimensional vector where the $p^{th}$ dimension contains the $p^{th}$ highest value of the feature-vector's $j^{th}$ dimension over the entire video. We denote $\text{Top}_K(\mathbf{x}^v_{1:F_v})$ to be a $KD$ dimensional vector obtained by concatenating the ordinal statistics for each dimension. Thus, the resulting feature-vector $\varphi(\mathbf{x}^v_{1:F_v})$ for the video becomes:

$$\varphi(\mathbf{x}^v_{1:F_v}) = \begin{bmatrix} \mu(\mathbf{x}^v_{1:F_v}) \\ \sigma(\mathbf{x}^v_{1:F_v}) \\ \text{Top}_K(\mathbf{x}^v_{1:F_v}) \end{bmatrix}. \quad (2)$$

#### 4.2.2 Feature normalization

Standardization of features has been proven to help with online learning algorithms[14, 37] as it makes the updates using Stochastic Gradient Descent (SGD) based algorithms (like Adagrad) more robust to learning rates, and speeds up convergence.

Before training our one-vs-all classifiers on the video-level representation, we apply global normalization to the feature vectors $\varphi(\mathbf{x}^v_{1:F_v})$ (defined in equation 2). Similar to how we processed the frame features, we substract the mean $\varphi(.)$ then use PCA to decorrelate and whiten the features. The normalized video features are now approximately multivariate gaussian with zero mean and identity covariance. This makes the gradient steps across the various dimensions independent, and learning algorithm gets an unbiased view of each dimension (since the same learning rate is applied to each dimension). Finally, the resulting features are $L_2$ normalized. We found that these normalization techniques make our models train faster.

### 4.3 Models from Video Features

Given the video-level representations, we train independent binary classifiers for each label using all the data. Exploiting the structure information between the various labels is left for future work. A key challenge is training these classifiers at the scale of this dataset. Even with a compact video-level representation for the $6M$ training videos, it is unfeasible to train batch optimization classifiers, like SVM. Instead, we use online learning algorithms, and use Adagrad to perform model updates on the weight vectors given a small mini-batch of examples (each example is associated with a binary ground-truth value).

#### 4.3.1 Logistic Regression

Given $D$ dimensional video-level features, the parameters $\mathbf{\Theta}$ of the logistic regression classifier are the entity specific weights $\mathbf{w}_e$. During scoring, given $\mathbf{x} \in \mathbb{R}^{D+1}$ to be the video-level feature of the test example, the probability of the entity $e$ is given as $p(e|\mathbf{x}) = \sigma(\mathbf{w}_e^T\mathbf{x})$. The weights $\mathbf{w}_e$ are obtained by minimizing the total log-loss on the training data given as:

$$\lambda\|\mathbf{w}_e\|_2^2 + \sum_{i=1}^{N} \mathcal{L}(y_{i,e}, \sigma(\mathbf{w}_e^T\mathbf{x}_i)), \quad (3)$$

where $\sigma(.)$ is the standard logistic, $\sigma(z) = 1/(1+\exp(-z))$.

#### 4.3.2 Hinge Loss

Since training batch SVMs on such a large dataset is impossible, we use the online SVM approach. As in the conventional SVM framework, we use $\pm 1$ to represent negative and positive labels respectively. Given binary ground-truth labels $y$ (0 or 1), and predicted labels $\hat{y}$ (positive or negative scalars), the hinge loss is:

$$\mathcal{L}(y, \hat{y}) = \max(0, b - (2y-1)\hat{y}), \quad (4)$$

where $b$ is the hinge-loss parameter which can be fine-tuned further or set to 1.0. Due to the presence of the max function, there is a discontinuity in the first derivative. This results in the subgradient being used in the updates, slowing convergence significantly.

#### 4.3.3 Mixture of Experts (MoE)

Mixture of experts (MoE) was first proposed by Jacobs and Jordan [18]. The binary classifier for an entity $e$ is composed of a set of hidden states, or experts, $\mathcal{H}_e$. A softmax is typically used to model the probability of choosing each expert. Given an expert, we can use a sigmoid to model the existence of the entity. Thus, the final probability for entity $e$'s existence is $p(e|\mathbf{x}) = \sum_{h\in\mathcal{H}_e} p(h|\mathbf{x})\sigma(\mathbf{u}_h^T\mathbf{x})$, where $p(h|\mathbf{x})$ is a softmax over $|\mathcal{H}_e|+1$ states. In other words, $p(h|\mathbf{x}) = \frac{\exp(\mathbf{w}_h^T\mathbf{x})}{1+\sum_{h'\in\mathcal{H}_e}\exp(\mathbf{w}_{h'}^T\mathbf{x})}$. The last, $(|\mathcal{H}_e|+1)^{th}$, state is a dummy state that always results in the non-existence of the entity. Denote $p_{y|\mathbf{x}} = p(y=1|\mathbf{x})$, $p_{h|\mathbf{x}} = p(h|\mathbf{x})$ and $p_h = p(y=1|\mathbf{x},h)$. Given a set of training examples $(\mathbf{x}_i, g_i)_{i=1\ldots N}$ for a binary classifier, where $\mathbf{x}_i$ is the feature vector and $g_i \in [0,1]$ is the ground-truth, let $\mathcal{L}(p_i, g_i)$ be the log-loss between the predicted probability and the ground-truth:

$$\mathcal{L}(p, g) = -g\log p - (1-g)\log(1-p). \quad (5)$$

We could directly write the derivative of $\mathcal{L}\left[p_{y|\mathbf{x}}, g\right]$ with respect to the softmax weight $\mathbf{w}_h$ and the logistic weight $\mathbf{u}_h$ as

$$\frac{\partial\mathcal{L}\left[p_{y|\mathbf{x}}, g\right]}{\partial\mathbf{w}_h} = \mathbf{x}\frac{p_{h|\mathbf{x}}\left(p_{y|h,\mathbf{x}} - p_{y|\mathbf{x}}\right)\left(p_{y|\mathbf{x}} - g\right)}{p_{y|\mathbf{x}}(1-p_{y|\mathbf{x}})}, \quad (6)$$

$$\frac{\partial\mathcal{L}\left[p_{y|\mathbf{x}}, g\right]}{\partial\mathbf{u}_h} = \mathbf{x}\frac{p_{h|\mathbf{x}}p_{y|h,\mathbf{x}}(1-p_{y|h,\mathbf{x}})\left(p_{y|\mathbf{x}} - g\right)}{p_{y|\mathbf{x}}(1-p_{y|\mathbf{x}})}. \quad (7)$$

We use Adagrad with a learning rate of 1.0 and batch size of 32 to learn the weights. Since we are training independent classifiers for each label, the work is distributed across multiple machines.

For MoE models, we experimented with varying number of mixtures (1, 2, 4), and found that performance increases by 0.5%-1% on all metrics as we go from 1 to 2, and then to 4 mixtures, but the number of model parameters correspondingly increases by 2 or 4 times. We chose 2 mixtures as a good compromise and report numbers with the 2-mixture MoE model for all datasets.

## 5. EXPERIMENTS

In this section, we first provide benchmark baseline results for the above multi-label classification approaches on the YouTube-8M dataset. We then evaluate the usefulness of video representations learned on this dataset for other tasks, such as Sports-1M sports classification and AcitvityNet activity classification.

### 5.1 Evaluation Metrics

**Mean Average Precision (mAP):** For each entity, we first round the annotation scores in buckets of $10^{-4}$ and sort all the *non-zero* annotations according to the model score. At a given threshold $\tau$, the precision $P(\tau)$ and recall $R(\tau)$ are given by

$$P(\tau) = \frac{\sum_{t\in T}\mathbb{I}(\mathbf{y}_t \geq \tau)g_t}{\sum_{t\in T}\mathbb{I}(\mathbf{y}_t \geq \tau)}, \quad (8)$$

$$R(\tau) = \frac{\sum_{t\in T}\mathbb{I}(\mathbf{y}_t \geq \tau)g_t}{\sum_{t\in T} g_t}, \quad (9)$$

| Input Features | Modeling Approach | mAP | Hit@1 | PERR |
|---|---|---|---|---|
| Frame-level, $\{\mathbf{x}^v_{1:F_v}\}$ | Logistic + Average (4.1.1) | 11.0 | 50.8 | 42.2 |
| Frame-level, $\{\mathbf{x}^v_{1:F_v}\}$ | Deep Bag of Frames (4.1.2) | 26.9 | 62.7 | 55.1 |
| Frame-level, $\{\mathbf{x}^v_{1:F_v}\}$ | LSTM (4.1.3) | 26.6 | **64.5** | **57.3** |
| Video-level, $\mu$ | Hinge loss (4.3) | 17.0 | 56.3 | 47.9 |
| Video-level, $\mu$ | Logistic Regression (4.3) | 28.1 | 60.5 | 53.0 |
| Video-level, $\mu$ | Mixture-of-2-Experts (4.3) | 29.6 | 62.3 | 54.9 |
| Video-level, $[\mu; \sigma; \text{Top}_5]$ | Mixture-of-2-Experts (4.3) | **30.0** | 63.3 | 55.8 |

**Table 3: Results of the various benchmark baselines on the YouTube-8M dataset. We find that binary classifiers on simple video-level representations perform substantially better than frame-level approaches. Deep learning methods such as DBoF and LSTMs do not provide a substantial boost over traditional dense feature aggregation methods because the underlying frame-level features are already very strong.**

| Approach | Hit@1 | PERR | Hit@5 |
|---|---|---|---|
| Deep Bag of Frames (DBoF) (4.1.2) | 68.6 | 29.0 | 83.5 |
| LSTM (4.1.3) | 69.1 | **30.5** | **84.7** |
| Mixture-of-2-Experts ($[\mu; \sigma; \text{Top}_5]$) (4.3) | **70.1** | 29.1 | **84.8** |

**Table 4: Results of the three best approaches on the human rated test set of the YouTube-8M dataset. A comparison with the results on the validation set (Table 3) shows that the relative strengths of the different approaches are largely preserved on both sets.**

where $\mathbb{I}(.)$ is the indicator function. The average precision, approximating the area under the precision-recall curve, can then be computed as

$$\text{AP} = \sum_{j=1}^{10000} P(\tau_j)[R(\tau_j) - R(\tau_{j+1})], \tag{10}$$

where where $\tau_j = \frac{j}{10000}$. The mean average precision is computed as the *unweighted* mean of all the per-class average precisions.

**Hit@k:** This is the fraction of test samples that contain at least one of the ground truth labels in the top $k$ predictions. If $\text{rank}_{v,e}$ is the rank of entity $e$ on video $v$ (with the best scoring entity having rank 1), and $G_v$ is the set of ground-truth entities for $v$, then Hit@$k$ can be written as:

$$\frac{1}{|V|} \sum_{v \in V} \vee_{e \in G_v} \mathbb{I}(\text{rank}_{v,e} \leq k), \tag{11}$$

where $\vee$ is logical OR.

**Precision at equal recall rate (PERR):** We measure the video-level annotation precision when we retrieve the same number of entities per video as there are in the ground-truth. With the same notation as for Hit@$k$, PERR can be written as:

$$\frac{1}{|V : |G_v| > 0|} \sum_{v \in V : |G_v| > 0} \left[ \frac{1}{|G_v|} \sum_{e \in G_v} \mathbb{I}(\text{rank}_{v,e} \leq |G_v|) \right].$$

## 5.2 Results on YouTube-8M

Table 3 shows results for all approaches on the YouTube-8M dataset. Frame-level models (row 1), trained on the strong Inception features and logistic regression, followed by simple averaging of predictions across all frames, perform poorly on this dataset. This shows that the video-level prediction task cannot be reduced to simple frame-level classification.

Aggregating the frame-level *features* at the video-level using simple mean pooling of frame-level features, followed by a hinge loss or logistic regression model, provides a non-trivial improvement in video level accuracies over naive averaging of the frame-level predictions. Further improvements are observed by using mixture-of-experts models and by adding other statistics, like the standard deviation and ordinal features, computed over the frame-level features. Note that the standard deviation and ordinal statistics are more meaningful in the original RELU activation space so we reconstruct the RELU features from the PCA-ed and quantized features by inverting the quantization and the PCA using the provided PCA matrix, computing the collection statistics over the reconstructed frame-level RELU features, and then re-applying PCA, whitening, and L2 normalization as described in Section 4.2.2. This simple task-independent feature pooling and normalization strategy yields some of the most competitive results on this dataset.

Finally, we also evaluate two deep network architectures that have produced state-of-art results on previous benchmarks [26]. The DBoF architecture ignores sequence information and treats the input video as a bag of frames whereas LSTMs use state information to preserve the video sequence. The DBoF approach with a logistic classification layer produces 2% (absolute) gains in Hit@1 and PERR metrics over using simple mean feature pooling and a single-layer logistic model, which shows the benefits of discriminatively training a projection layer to obtain a task-specific video-level representation. The mAP results for DBoF are slightly worse than mean pooling + logistic model, which we attribute to slower training and convergence of DBoF on rare classes (mAP is strongly affected by results on rare classes and the joint class training of DBoF is a disadvantage for those classes).

The LSTM network generally performs best, except for mAP, where the 1-vs-all binary MoE classifiers perform better, likely for the same reasons of slower convergence on rare classes. LSTM does improve on Hit@1 and PERR metrics, as expected given its ability to learn long-term correlations in the time domain. Also, in [26], the authors used data augmentation by sampling multiple snippets of fixed length from a video and averaged the results, which could produce even better accuracies than our current results.

We also considered Fisher vectors and VLAD given their recent success in aggregating CNN features at the video-level in [39]. However, for the same dimensionality as the video-level representations of the LSTM, DBoF and mean features, they did not produce competitive results.

### 5.2.1 *Human Rated Test Set*

We also report results on the human rated test set of over 8000 videos (see Section 3.5) in Table 4 for the top three approaches. We report PERR, Hit@1, and Hit@5, since the mAP is not reliable given the size of the test set. The Hit@1 numbers are uniformly higher for all approaches when compared to the incomplete validation set in Table 3 whereas the PERR numbers are uniformly lower. This is largely attributable to the missing labels in the validation set (recall of the Validation set labels is around 15% compared to exhaustive human ratings). However, the relative ordering of the various approaches is fairly consistent between the two sets, showing that the validation set results are still reliable enough to compare different approaches.

## 5.3 Results on Sports-1M

Next, we investigate generalization of the video-level features learned using the YouTube-8M dataset and perform transfer learning experiments on the Sports-1M dataset. The Sports-1M dataset [19] consists of 487 sports activities with 1.2 million YouTube videos and is one of the largest benchmarks available for sports/activity recognition. We use the first 360 seconds of a video sampled at 1 frame per second for all experiments.

To evaluate transfer learning on this dataset, in one experiment we simply use the aggregated video-level descriptors, based on the PCA matrix learned on the YouTube-8M dataset, and train MoE or

| Approach | mAP | Hit@1 | Hit@5 |
|---|---|---|---|
| Logistic Regression ($\mu$) (4.3) | 58.0 | 60.1 | 79.6 |
| Mixture-of-2-Experts ($\mu$) (4.3) | 59.1 | 61.5 | 80.4 |
| Mixture-of-2-Experts ($[\mu; \sigma; \text{Top}_5]$) (4.2.1) | 61.3 | 63.2 | 82.6 |
| LSTM (4.1.3) | 66.7 | 64.9 | 85.6 |
| *+Pretrained on YT-8M* (4.1.3) | 67.6 | 65.7 | 86.2 |
| Hierarchical 3D Convolutions [19] | - | 61.0 | 80.0 |
| Stacked 3D Convolutions [35] | - | 61.0 | 85.0 |
| LSTM with Optical Flow and Pixels [26] | - | **73.0** | **91.0** |

**(a) Sports-1M: Our learned features are competitive on this dataset beating all but the approach of [26], which learned directly from the video pixels. Both [26] and [35] included motion features.**

| Approach | mAP | Hit@1 | Hit@5 | |
|---|---|---|---|---|
| Mixture-of-2-Experts ($\mu$) (4.3) | 69.1 | 68.7 | 85.4 | |
| *+Pretrained PCA on YT-8M* | 74.1 | 72.5 | 89.3 | |
| Mixture-of-2-Experts ($[\mu; \sigma; \text{Top}_5]$) (4.2.1) | NO | 74.2 | 72.3 | 89.6 |
| *+Pretrained PCA on YT-8M* | 77.6 | 74.9 | 91.6 | |
| LSTM (4.1.3) | 57.9 | 63.4 | 81.0 | |
| *+Pretrained on YT-8M* (4.1.3) | 75.6 | 74.2 | 92.4 | |
| Ma, Bargal et al.[24] | 53.8 | - | - | |
| Heilbron et al.[12] | 43.0 | - | - | |

**(b) ActivityNet: Since the dataset is small, we see a substantial boost in performance by pre-training on YouTube-8M or using the transfer learnt PCA versus the one learnt from scratch on ActivityNet.**

**Table 5: Results of transferring video representations learned on the YouTube-8M dataset to the (a) Sports-1M and (b) ActivityNet.**

logistic models on top using target domain training data.

For the LSTM networks, we have two scenarios: 1) we use the PCA transformed features and learn a LSTM model from scratch using these features; or 2) we use the LSTM layers pre-trained on the YouTube-8M task, and fine-tune them on the Sports-1M dataset (along with a new softmax classifier).

Table 5a shows the evaluation metrics for the various video-level representations on the Sports-1M dataset. Our learned features are competitive on this dataset, with the best approach beating all but the approach of [26], which learned directly from the pixels of the videos in the Sports-1M dataset, including optical flow, and made use of data augmentation strategies and multiple inferences over several video segments. We also show that even on such a large dataset (1M videos), pre-training on YouTube-8M still helps, and improves the LSTM performance by $\sim$1% on all metrics (vs. no pre-training).

## 5.4 Results on ActivityNet

Our final set of experiments demonstrate the generality of our learned features for the ActivityNet untrimmed video classification task. Similar to Sports-1M experiments, we compare directly training on the ActivityNet dataset against pre-training on YouTube-8M for aggregation based and LSTM approaches. As seen in Table 5b, all of the transferred features are much better in terms of all metrics than training on ActivityNet alone. Notably, without the use of motion information, our best feature is better by up to 80% than the HOG, HOF, MBH, FC-6, FC-7 features used in [12]. This result shows that features learned on YouTube-8M generalize very well to other datasets/tasks. We believe this is because of the diversity and scale of the videos present in YouTube-8M.

# 6. CONCLUSIONS

In this paper, we introduce YouTube-8M, a large-scale video benchmark for video classification and representation learning. With YouTube-8M, our goal is to advance the field of video understanding, similarly to what large-scale image datasets have done for image understanding. Specifically, we address the two main challenges with large-scale video understanding—(1) collecting a large ***labeled*** video dataset, with reasonable quality labels, and (2) **removing computational barriers** by pre-processing the dataset and providing state-of-the-art frame-level features to build from. We process over 50 years worth of video, and provide features for nearly 2 billion frames from more than 8 million videos, which enables training a reasonable model at this scale within 1 day, using an open source framework on a single machine! We expect this dataset to level the playing field for academia researchers, bridge the gap with large-scale labeled video datasets, and significantly accelerate research on video understanding. We hope this dataset will prove to be a test bed for developing novel video representation learning algorithms, and especially approaches that deal effectively with noisy or incomplete labels.

As a side effect, we also provide one of the largest and most diverse public visual annotation vocabularies (consisting of 4800 visual Knowledge Graph entities), constructed from popularity signals on YouTube as well as manual curation, and organized into 24 top-level categories.

We provide extensive experiments comparing several strong baselines for video representation learning, including Deep Networks and LSTMs, on this dataset. We demonstrate the efficacy of using a fairly unexplored class of models (mixture-of-experts) and show that they can outperform popular classifiers like logistic regression and SVMs. This is particularly true for our large dataset where many classes can be multi-modal. We explore various video-level representations using simple statistics extracted from the frame-level features and model the probability of an entity given the aggregated vector as an MoE. We show that this yields competitive performance compared to more complex approaches (that directly use frame-level information) such as LSTM and DBoF. This also demonstrates that if the underlying frame-level features are strong, the need for more sophisticated video-level modeling techniques is reduced.

Finally, we illustrate the usefulness of the dataset by performing transfer learning experiments on existing video benchmarks—Sports-1M and ActivityNet. Our experiments show that features learned on this dataset generalize well on these benchmarks, including setting a new state-of-the-art on ActivityNet.

# 7. REFERENCES


[1] Freebase: A community-curated database of well-known people, places, and things. https://www.freebase.com.

[2] Google I/O 2013 - semantic video annotations in the Youtube Topics API: Theory and applications. https://www.youtube.com/watch?v=wf_77z1H-vQ.

[3] Knowledge Graph Search API. https://developers.google.com/knowledge-graph/.

[4] Tensorflow: Image recognition. https://www.tensorflow.org/tutorials/image_recognition.

[5] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2005.

[6] J. Deng, W. Dong, R. Socher, L. jia Li, K. Li, and L. Fei-fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[7] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge, 2009.

[8] L. Fei-fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28, 2006.

[9] R. Girshick. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[10] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[12] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015.

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computing*, 9(8), Nov. 1997.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

[15] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(9), Sept. 2012.

[16] Y. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14, 2014.

[17] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *arXiv preprint arXiv:1502.07209*, 2015.

[18] M. I. Jordan. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6, 1994.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, Columbus, Ohio, USA, 2014.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[21] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[22] I. Laptev and T. Lindeberg. Space-time interest points. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2003.

[23] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[24] S. Ma, S. A. Bargal, J. Zhang, L. Sigal, and S. Sclaroff. Do less and achieve more: Training cnns for action recognition utilizing action images from the web. *CoRR*, abs/1512.07155, 2015.

[25] V. Mnih and G. Hinton. Learning to label aerial images from noisy data. In *Proceedings of the 29th Annual International Conference on Machine Learning (ICML)*, June 2012.

[26] J. Y.-H. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4694–4702, 2015.

[27] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[28] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[29] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *ArXiv e-prints*, Dec. 2014.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[31] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations (ICLR)*.

[32] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[33] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[34] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. Li. The new data and new challenges in multimedia research. *CoRR*, abs/1503.01817, 2015.

[35] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3D: generic features for video analysis. *CoRR*, abs/1412.0767, 2014.

[36] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *Proc. BMVC*, 2009.

[37] S. Wiesler, A. Richard, R. Schlüter, and H. Ney. Mean-normalized stochastic gradient for large-scale deep learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, pages 180–184. IEEE, 2014.

[38] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, A. Oliva, and J. Xiao. Sun database: Exploring a large collection of scene categories, 2013.

[39] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[40] H.-F. Yu, P. Jain, P. Kar, and I. Dhillon. Large-scale multi-label learning with missing labels. In *Proceedings of The 31st International Conference on Machine Learning (ICML)*, pages 593–601, 2014.

[41] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.