DEEP LEARNING FOR MULTI-LABEL VIDEO CLASSIFICATION ON THE *YouTube-8M* VIDEO DATASET

Ashwin Ravishankar
Department of Computer Science
George Mason University
Fairfax, VA 22030
aravisha@gmu.edu

May 8, 2019

ABSTRACT

Extremely large datasets are attributed for the recent advancements in the field of Computer Vision applications. Open-Source software packages for machine learning, deep learning and easy to use hardware have opened avenues for deep learning research on novel architectures. ImageNet [1] helped fast forward image understanding process, while the YouTube-8M video dataset is helping accelerate machine capability of video understanding. This curated version 3 of the dataset consists of over 6.1 million videos (350, 000 hours of running time) spanning over 3862 labelled classes [2]. Over 2.6 billion audio/visual features are available to use.

In an effort to help further the field of video understanding research, through this paper, several model architectures are explored that utilize combinations of video-level visual and audio features. In an exploratory search to identify the best performing model for video tag prediction task, promising results are obtained by **Mixture of Experts**, especially a hierarchical setting, Hierarchical Mixture of Experts (HMoE). The gating function used for the MoE is softmax over 200 logistic classifiers as expert hidden states per class. Empirically this model performs better than the baseline models to achieve a **GAP of 82.4%** and **Hit@1 of 0.858** which is a significant improvement over **Hit@1 of 0.645** for the best model Google trained in their initial analysis [3].

***K*eywords** Video Classification · YouTube-8M dataset · Mixture of Experts

1 Introduction

The digital era has brought with it an enormous explosion of data. Video traffic from sites such as YouTube has increased in the recent years are an indisputable part of our daily lives. It is a solid contributor to our daily dose of entertainment, knowledge, information etc. A post on Quora [4] claims that 48 hours of video got uploaded to YouTube, per minute, in 2011, and, another blog [5] estimates that number is now 300+ hours per minute. Given such enormous numbers, how sure are we about the quality of those videos? Given the statistics, an efficient methodology needs to be employed to classify videos at scale whose application can be further expanded to content discovery, filtering, spam identification, flagging of copyright infringement, hidden information extraction. The applications are endless. Empowering a machine to understand videos with precision would be valuable and is still an open field of research. A step towards achieving that capability would be to classify videos and predict multiple tags for each video.

In this project, experiments with application of multi layer vanilla neural network, convolutional neural networks, recurrent neural networks are done. Input to the model is the pre-computed video level features extracted, and at output, labels for the test video are predicted.

Video classification is inherently a difficult task for three main reasons. One, dataset for video classification is limited to a particular activity and discriminates visual and audio features. Second, the trade-off between computational cost

Figure 1: Video - viewed as contiguous image frames.

and accuracy of the model. Finally, labeling of videos can be very subjective – perceptible to noise, erratic change in scene, prone to losing context of activity.

In rest of the paper, *Section 2* describes what the objective of the project is and details out the problem I aimed at solving. The existing research that went into video classification is elicited in *section 3*. In *section 4*, I detail out what approaches and deep models are experimented with. The detailed description of the dataset and the evaluation metrics used to test my models are outlined in *section 5*. Right before concluding my work in *section6*, I comprehensively analyze the results obtained from my experiments in *section 5.3*.

2 Problem Statement

Videos can be observed to be contiguous frames of images *i.e.,* from Figure 1, each consecutive frame stacked together captures the temporal aspect of the scene, resulting in video. As a human expert, when I view the video represented in figure 1, I immediately relate to the theme of the video - *Golf, Green Grass, Game*. The objective of the video classification task is to predict the class label that the video belongs to. Often times this can be mis-construed to be scene detection or object in the scene identification task [6]. But, the goal of this project is to predict the main theme of the video. With the availability of large curated dataset for image classification, deep learning models have progressed to achieve better than human level accuracy for image classification task [6]. For long, the unavailability of such datasets for video understanding was the handicap, that was addressed by YouTube-8M dataset. With experimental deep architectures, we aim to empower a machine to better understand videos. The process followed is to plug in the features observed from the video into a function to get label outputs.

3 Literature Review

Benchmarks for image datasets [1] have played a significant role in advancing vision algorithms for image understanding. Research in the area of multi-label classification for rich video data has been limited in the past by lack of an accurately labeled, large scale database for such videos. In the past it was make do with small well labeled dataset like human activity recognition (KTH) [7], Hollywood 2 [8] etc. Prior to the availability of the YouTube-8M dataset, the largest collection of labeled videos was the Sport-1M video tagged with 487 sport related activities [9]. YouTube-8M is the key player who changed the game of video classification research with 6.1 million videos to work with.

In the domain of video understanding, two editions of Kaggle challenge [10] was sponsored by Google to support open community research. The advancement of video classification involving working with frame-level features, supersedes studies, that work with video-level features.

Empirically strong results are reported from using approaches like Fisher Vectors [11] and NetVLAD [12] as demonstrated in [13, 14]. [13] also utilizes an approach involving soft bag-of-words to deal with sequential feature vectors, a novel Context Gating is introduced.

Motivated by the promising results of applying CNN to images, several investigations on effective using CNN on multi label video classifications include [15]'s best performing model to be a 3-layer spatial-temporal CNN-Fully Connected model. [9] focused primarily on improvements to the upstream CNN architecture to enhance predictive capability.

Other architectures experimented with are ensemble of models, using LSTM or GRU to capture temporal aspect of the videos, Mixture of Experts. Some not so extensive study was done on the impacts of audio-visual feature combination. In most cases, using both features in conjunction proved effective over model training only using visual features.

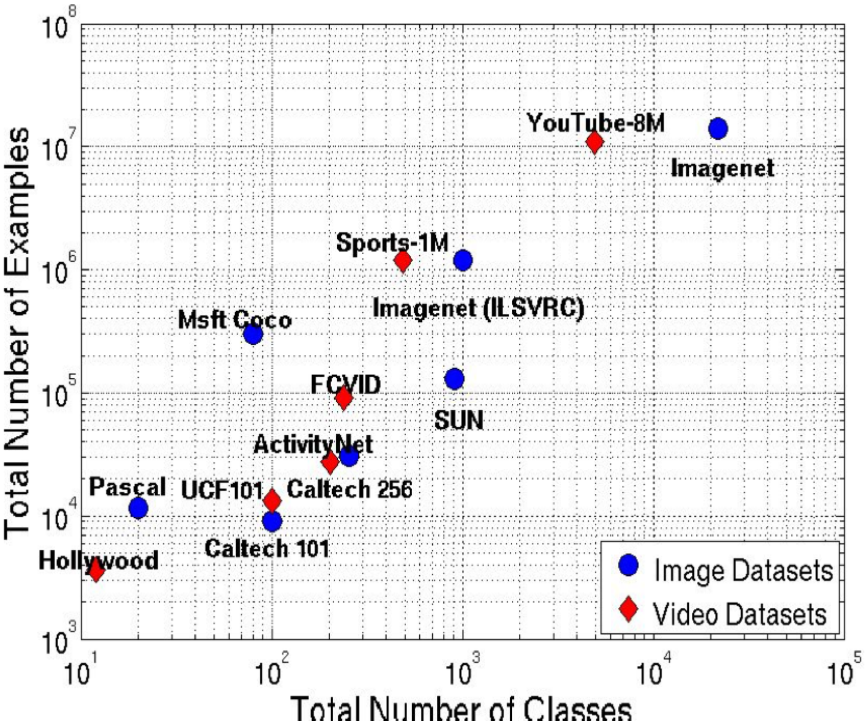

Figure 2: The progression of datasets for image and video understanding tasks [3]

4 Methods and Techniques

I chose to train the models only using the video-level visual and audio features generated by the Inception-V3 network due to the computational complexity. The total audio-visual video-level features is 31GB in size. Concretely, cross entropy is loss is minimized, of the predicted label probabilities from the true label probabilities, where the predicted label probability is 1 if the video is tagged with the true label and 0 if not.

The YouTube-8M TensorFlow starter code [16] was used as a placeholder for training and testing the models experimented in this paper. Inference on the test split can be done to secure a spot the Kaggle leaderboard, based on the precision of predicted tags. Each model was trained for 5 epochs with initial learning rate of 0.01 and a decay of 0.95 every 4 million examples and a batch size of 1024. The Adam optimizer was chosen for training as it adaptively anneals the learning rate in each dimension, thus reducing the dependence on initial learning rate selection and helps converge faster. Also the residual model is trained with dropout regularizer[17]. The models were trained on Google cloud AI platform. Few models were trained on 1 GPU and some on 4 GPU.

The below sections 4.1 to 4.7 outline the different models that were trained for this project.

4.1 Logistic Regression - Baseline Model

The input to the model is the video-level feature x_i of the video *i* and probability of entity *j* as $\sigma(w_j^T x_i)$, the model was trained to minimize the log loss of training data to learn the weight parameters using gradient descent algorithm.

$$\lambda||w_j||_2^2 + \sum_{i=1}^{N} L(y_i, \sigma(w_j^T x_i)) \tag{1}$$

where the sigmoid function $\sigma(x)$ is given as:

$$\sigma(x) = 1/(1 + exp(-x)) \tag{2}$$

and the cross entropy loss function to minimize is given by

$$\sum_i y_i log(\hat{y_i}) + (1 - y_i) log(1 - \hat{y_i}) \tag{3}$$

In simple observation, it is the linear projection of the video-level features into label space, and converting the log values into class probabilities using a sigmoid function. The cross entropy loss function is minimized. Results from feeding two types of inputs - visual features, both visual and audio features combined, are observed.

4.2 Feed Forward Neural Network Model

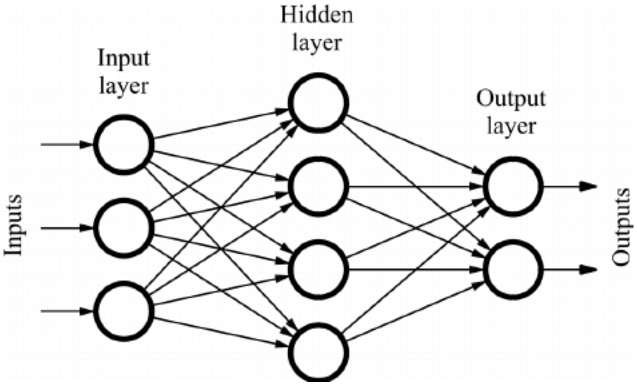

Figure 3: Feed Forward Neural Network

This is a video-level visual input features only, fully connected model as displayed in *Figure* 3. The model is simple matrix multiplication of weights with input parameters. The hidden layer has 512 neurons which causes dimensionality reduction of the input features by half. ReLU activation function is used for the hidden layer. Rectified Linear Unit (ReLU) activation is computed as $f(x) = max(0, x_i)$. Hidden layer output is computed as $h_i = f(W^T x_i)$ where x_i is the input. The output layer is a softmax classifier which assigns a probability score to each of the classes in the dataset. Using backpropogation approach, the parameters are learned.

4.3 CNN Model

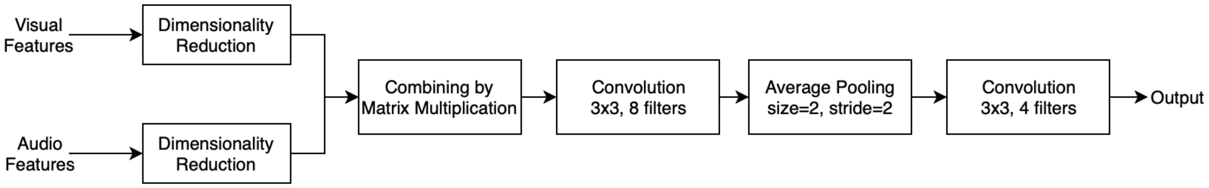
Visual Features
Dimensionality Reduction
Audio Features
Dimensionality Reduction
Combining by Matrix Multiplication
Convolution 3x3, 8 filters
Average Pooling size=2, stride=2
Convolution 3x3, 4 filters
Output

Figure 4: Schematic of my VGGNet extended CNN architecture

The visual and audio features are separately bottle necked into 32 dimensions each and combined using matrix multiplication [1]. In the dimensionality reduction step, ReLU activation is used in the hidden layer. Unlike the traditional spatio-temporal CNN models that captures both spatial and temporal features, I designed this model to work only with spatial features. A convolution step is performed on the combined feature space 32 x 32 x 1, using eight 3 x 3 filters. Average Pooling is done on the feature maps with a stride and size of two. The pooled features are again convolved with four 3 x 3 kernels. Right before the output layer, the convolved features are flattened and passed over softmax activation function to predict class probability [2].

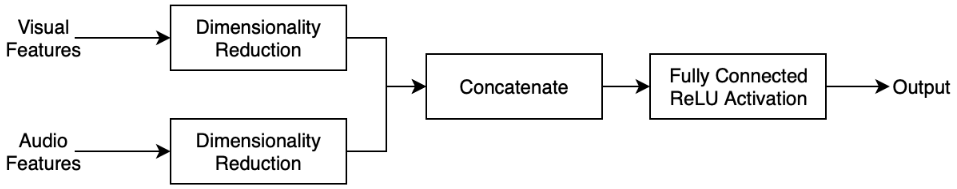

Figure 5: Schematic of Branched Neural Network model consuming audio and visual features separately.

4.4 Branched Neural Network Model

Inspired by the Feed Forward Neural Network model's results, this experiment was designed to pre process the visual and audio features separately. In this branched neural network model (*Figure* 5), the visual and audio features are separately reduced to half the original dimension. ReLU activation is used for the dimentionality reduction layer. The 512 dimension visual features and 64 dimension audio features are concatenated and fed into a fully connected dense hidden layer with ReLU activation. Finally a softmax activation is applied at the output layer to obtain entity probabilities. Minimizing cross entropy loss was the objective of this model training.

4.5 Residual Model

The residual network is a deep multi-layer video level feed forward design made up of residual connections. Residual or skip connections is the phenomenon where a certain set of feature vectors which can be the input, or output of any hidden layer, skip an arbitrary number of steps in the network and again induced into the progressive network. The residual network design is inspired by ResNet [6], the winner of ImageNet 2015 challenge, which is widely regarded as the state-of-the-art convolutional neural network, in image classification tasks, comprising of 152 layers. Formally[6], denoting the desired underlying mapping as $H(x)$, the stack of non linear layers fit another mapping $F(x) = H(x) - x$. The original mapping is recast into $F(x) + x$.

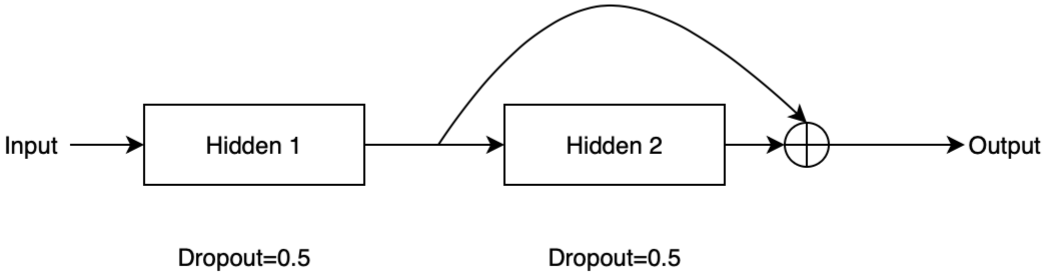

Figure 6: Residual Network Architecture.

In this experiment, the residual architecture is modelled as shown in Figure 6, the input is fed into a first hidden layer consisting of 10K neurons with ReLU activation. The output is then propagated into a fully connected hidden layer 2 of 10K neurons and ReLU activation. The residual step is - output from hidden layer 1 and hidden layer 2 are concatenated and passed into a output layer with softmax activation. A dropout of 0.5 is enforced on both the hidden layers during training time.

4.6 Mixture of Experts Model

This architecture is binary classifier comprising a number of N experts, a set of independent hidden states $E_1, E_2, ..., E_n$ and a trainable gating network G which determines the optimal mix of the different experts over different inputs. A model architecture is shown in Figure 7.

[1] Please refer to *Section 5.1* for dataset details

[2] This approach was an effort for further process the VGGNet Inception-v3 model's output using CNN architecture.

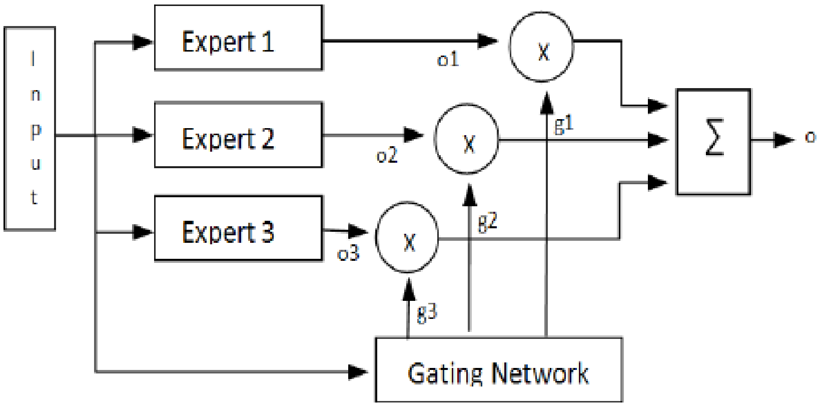

Figure 7: Mixture of Experts Network Architecture.

Representing gating network as $G(x)$, output of the i-th expert as $E_i(x)$, output of the model is

$$o = \sum_{i=1}^{n} G_i(x) E_i(x) \tag{4}$$

Sparsity value of $G(x)$ determines the contribution of an expert to predict the class. A softmax activation is used for the gating function while sigmoid is chosen as the activation of the binary classifier expert E_i. Softmax is given by:

$$p(x_j|x) = s(x_j) = \frac{e_j^s}{\sum_{i=1}^{n} e_j^s} \tag{5}$$

In this paper, I use only visual features at video-level to train the model. Surprisingly good results are observed. The model was trained with 8 experts.

4.7 Hierarchical Mixture of Experts Model

This model is hierarchical setting of the Mixture of Experts (MoE) model. The hierarchical nature of the model takes effect with layering *i.e.,* the output of the first MoE model is consumed as the input to the second MoE model.

It was computationally exhaustive to train this model as there were two MoE worth of trainable parameters to learn. The first level MoE was trained with 200 experts and second level MoE consisted of 8 experts.

The best results from my experiments are observed from this approach.

5 Discussion And Results

5.1 Dataset - YouTube-8M

Thanks to Google and YouTube, we now have a gigantic video dataset [2] that can now be used for video understanding research. The dataset was first released in 2016 with about 8 million YouTube videos. The current 3^{rd} edition of the dataset now contains $6.1 million$ videos that have about 350,000 hours of running time. The dataset was prepared with carefully chosen videos. The first 300 seconds of the videos are sampled at 1 frame per second, fed into Google's VGGNet Inception-v3 model [3], and the features are extracted before the final classification layer, as frame-level visual features.

These frame level features are averaged over the entire video to compute the video-level visual feature. From a VGGNet inspired model, audio features of the media are also computed and made available. The frame-level features is about 1.53 TB in size, while the video level features are over 31 GB. Due to the computational complexity of video classification task, I adapted the video-level visual and audio features to work with, on this project.

A total of 2.6 billion features are pre-computed and made available in this dataset. The computed features are then PCAd along with whitening process to constrain them to - 1024 dimentional visual features and 128 dimensional audio features. A total of 3862 classes are identified, and 3 labels are tagged to a video on average. The dataset is stored as tensorflow.Example protocol buffers, which have been saved as *.tfrecord* files. The dataset split is 70% for training data, 20% for validation data, and 10% for test data. The label tagging of videos are made available only for the training and validation dataset.

Hence, for the scope of the project, the validation dataset is used as the test dataset with the below 2 assumptions made:

- The training and validation data are evenly sampled across all 3862 classes.
- There is no mis-labeling in training data.

5.2 Evaluation Metrics

GAP To test the effectiveness of the classification task, the Kaggle video classification challenge's evaluation metric is adapted for the project - Global Average Precision (GAP):

$$GAP = \sum_{i=1}^{N} p(i) \Delta r(i) \tag{6}$$

This metric is the area under the precision recall curve. $p(i)$ denotes the precision and $\Delta r(i)$ denotes the recall - of prediction i as shown in Equation 7. N is the number of predictions represented as label confidence pairs. N is set to 20 for evaluation.

Hit@*k* The fraction of the test examples that contains at least one of the ground truth labels in the top k predictions [3].

5.3 Experimental Results

The results of the project are tabulated in *Table 1*.

The two terribly performing models are - Residual Networks and CNN based model. In a surprising turn of results, I observed that the residual network produced best accuracy in [18]. Also in [15], it is observed that their best performing model is CNN based.

Due to the time and monetary constraints, I prepared to work with video level features. However, frame lever feature exploitation would be instrumental in achieving at par results to the toppers of Kaggle challenge leader board.

Table 1: Results of model evaluation on the validation dataset. Time is in *hh:mm:ss*

Model	GAP	Hit@1	Training Time	# GPUs
Baseline - Logistic Classifier (without audio)	0.776	0.828	00:49:25	1
Baseline - Logistic Classifier (with audio)	0.808	0.854	00:47:02	1
Feed Forward Neural Network	0.782	0.828	00:46:06	1
CNN Network	0.699	0.809	00:50:07	1
Branched Neural Network	0.723	0.826	00:48:09	1
Residual Network	0.697	0.803	03:21:00	4
Mixture of Experts	0.817	0.852	01:48:00	4
Hierarchical Mixture of Experts	**0.824**	0.858	02:25:00	4

6 Conclusion

In summary, restricting the scope of the project to video-level models, I've improved, significantly, upon the initial results published by Google on their analysis of video understanding comparing the Hit@1 metric. However, a concrete test of model effectiveness is GAP which results Google did not publish as a part of their original findings. Despite ignoring temporal specs of the dataset and working only with video level features, models like MoE, HMoE, outperform frame level models that use LSTM. The best performing model of the project was - Hierarchical Mixture of Expert. It was able to achieve GAP of 82.4% and Hit@1 of 85.8%.

These numbers seem promising for video classification task. Considering that 83% hits of a YouTube keyword search results in relevant response, odds of finding a satisfying video is fairly high. There exists a possibility that a particular video of interest may fall into the mis-classified 17% videos, and may never show up in the results. Hence, its imperative to further the research in video understanding.

6.1 Scope of Future Enhancements

Going beyond the computational complexity costs, we can train more sophisticated architectures in a hope to see better results. From the research literature in video classification, it is empirically observed that frame level models perform best as it captures every aspect of available dataset. We should also look at effects of employing proved mechanisms like Attention model, methods of frame aggregation, use better than VGGNet architectures - ResNet, ResNeXT, Dense ResNet to compute the features of raw videos.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[2] Google. Youtube-8m Video Dataset. `https://research.google.com/youtube8m/index.html`, 2016. [Online; accessed 06-May-2019].

[3] Sami Abu-El-Haija et al. Youtube-8m: A large-scale video classification benchmark. In *arXiv:1609.08675*, 2016.

[4] Quora User. Youtube statistics - quora. `4.https://www.quora.com/How-many-videos-are-uploaded-to-Facebook-per-day-week-and-or-month`, 2011. [Online; accessed 06-May-2019].

[5] Blog. Youtube statistics - blog. `http://videonitch.com/2017/12/13/36-mind-blowing-youtube-facts-figures-statistics-2017-re-post/`, 2018. [Online; accessed 06-May-2019].

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[7] Barbara Caputo Ivan Laptev. Recognition of human actions. `http://www.nada.kth.se/cvap/actions/`. [Online; accessed 06-May-2019].

[8] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.

[9] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[10] Kaggle and Google. Kaggle challenge for video classificatio using youtube-8m dataset. `https://www.kaggle.com/c/youtube8m/data`. [Online; accessed 06-May-2019].

[11] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. pages 222–245, 2013.

[12] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[13] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. In *arXiv:1706.06905*, 2017.

[14] Kwangsoo Shin, Junhyeong Jeon, Seungbin Lee, Boyoung Lim, Minsoo Jeong, and Jongho Nang. Approach for video classification with multi-label on youtube-8m dataset. 2018.

[15] Alexandre GAuthier and Haiyu Lu. Youtube-8m video clasification.

[16] Google. Starter code for working with youtube-8m dataset. `https://github.com/google/youtube-8m`. [Online; accessed 06-May-2019].

[17] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, pages 1929–1958, 2014.

[18] Ryan Wong Hyun Sik Kim. Google cloud and youtube-8m video understanding challenge.