Compositional Event Detection Using Weak Supervision

Mark Cramer
AI Product Management
Xerox at PARC
Palo Alto, CA
mdcramer@stanford.edu

Aasavari Kakne
MS in Comp & Math Eng
Stanford University
Stanford, CA
adkakne@stanford.edu

Sundararajan Renganathan
PhD in Computer Science
Stanford University
Stanford, CA
rsundar@stanford.edu

Abstract

Despite recent advances in computer vision, detecting domain-specific events in videos remains a challenging task due to the amount of labelled data required for building models to detect such events. Rekall [1] is a system which detects compositional events by using off-the-shelf object and face detectors to generate spatiotemporal knowledge of the entities present in videos. The task of detecting abstract events then reduces to writing queries over the generated spatiotemporal knowledge. We investigate whether the outputs of Rekall queries can be supplied as weak supervision to build adequate, fresh machine learning models. Our deep learning experiments involve transfer learning through fine-tuning pre-trained ResNet-50 models. Increasing the amount of weak supervision data supplied will eventually produce models that outperform those trained on lesser amounts of ground-truth labelled (i.e. human labelled) data. We also apply classical ML techniques like GDA and SVM to this problem. Unsurprisingly, we find that these techniques are not able to perform as well as deep learning models, although do perform relatively well. Furthermore, the models converged relatively quickly and increasing the amount of weak supervision data supplied to these algorithms had no material difference on model performance.

1 Introduction

Detecting domain-specific events in videos, such as commercials during TV programs or game-winning sequences during sports competitions, is of great interest to video processing applications. The task is challenging as pre-trained models for identifying events often do not exist and building new models generally requires prohibitive amounts of hand-labelled data and computation. While human-generated labels are typically high quality, they're also expensive to produce, and a low volume of labelled data may, in fact, adversely impact prediction accuracy.

Rekall [1] provides a programming interface and data model for detecting domain-specific events in videos. Rekall queries operate by combining the outputs of pre-trained models (e.g. object detection, facial recognition, etc.) in order to identify events. As such, instead of having to train new models from scratch, domain experts may issue Rekall queries over unlabeled videos in order to detect compositional events.

Making Rekall queries "accurate enough," however, typically involves numerous iterations by a skilled domain expert in addition to significant manual tuning. To address this problem, we aim to use the outputs of Rekall queries as a form of weak supervision to train models, reducing manual effort involved in the tuning of Rekall queries or the building of end-to-end models from scratch using laboriously hand-labeled data. Our hypothesis is that sufficient quantities of weak supervision data in the form of relatively imperfect labels (coming from the outputs of Rekall queries) can be used to build models which perform better than the ones trained on smaller amounts of human labelled (or ground truth) data.

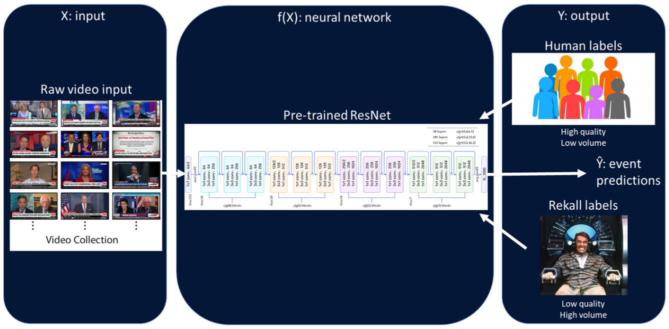

Figure 1: ResNet architecture with competing labels

Figure 2: Commercial

Figure 3: Programming

2 Learning Task

We focus on the task of detecting the presence of commercials in cable TV news programs. (A binary classification task with $y^{(i)} = 1$ indicating whether the frame is from a commercial.) We fine-tuned a ResNet-50 [2] model using PyTorch [3] by feeding in video frames as input. We also apply classical ML techniques for this task.

3 Datasets

The Rekall queries for this task have been written by domain experts. The Rekall queries are simply database queries over the spatiotemporal knowledge produced by the Rekall system. These queries output whether a given frame is a commercial or not. We call the outputs of Rekall queries (when run on frames) to be the Rekall labels for these frames. The ground truth labels, on the other hand, are obtained by humans labelers annotating the frames. Therefore, both the Rekall labels and the ground truth labels are binary values.

We employed the following datasets in order to perform deep learning experiments:

- 64,000 video frames ($360 \times 640 \times 3$) with true (i.e. human) labels as well as Rekall labels, used for training,
- 280,000 video frames with Rekall labels, also used for training, and
- 7500 frames with ground truth labels, used for testing.

We determined it was not necessary to hold out validation or cross-validation data for these experiments. 28.62% of the true-labeled frames were from commercials.

For the same task, we also extracted features from the frames to apply classical ML methods. We identified following features: mean and standard deviation of pixel intensity for (a) each frame as a whole, (b) the R, G and B channels separately for each frame, (c) each of nine equally-sized rectangular regions of the frame and (d) the R, G and B channels separately for each of the nine regions. We thus produced vectors $x^{(i)} \in \mathbb{R}^2$ using features from (a), $x^{(i)} \in \mathbb{R}^8$ when then adding features from (b), $x^{(i)} \in \mathbb{R}^{26}$ when also adding features from (c) and $x^{(i)} \in \mathbb{R}^{60}$ when finally adding features from (d).

4 Methods and Results

4.1 Deep Learning

Our deep learning experiments involved taking a ResNet-50 model pre-trained on ImageNet and fine-tuning all the layers of this model. We trained on progressively larger amounts of weak supervision data (with Rekall labels). The trade-off between the high quality but low volume humans labels and the high volume but lower quality Rekall labels is captured in Fig. 1.

4.1.1 Baselines

We have two baselines for our experiments. The first is obtained by running the Rekall queries over the test set and measuring the F1 score by comparing with the ground truth labels. This is the Rekall queries baseline (F1 of 0.9346) in Fig. 4. The second is obtained by fine-tuning the ResNet-50 model with the 64,000 video frames with ground truth labels. This baseline gives us a measure of how good the model can perform given all the ground truth labels that we possess. This is the true labels baseline (F1 of 0.9353) also in Fig. 4. Note that the true labels baseline is computed only once for the 64,000 video frames. This baseline doesn't involve training on progressively larger amounts of data because training on all the ground truth data that we have gives an upper limit on the performance of this baseline.

4.1.2 Experiment setup

We fine-tuned all layers of the ResNet-50 [2] by optimizing binary cross entropy loss using mini-batch gradient descent (batch size 16) on a Google cloud NVIDIA 4992-core Tesla K80 and a local NVIDIA 1024-core M2200. We performed experiments by fine-tuning the ResNet-50 on progressively larger sets of training data with Rekall labels as output, using sets of 6400, 12800, 25600, 36400, 51200, 64000, 128000, 192000, 256000 and 320000 examples. For each of the above experiments, we ran our training for 25 epochs. In order to perform standardized comparisons, we measured against both baselines, as described above. It is important to note that even though our training sets are different, the test F1 scores are always computed on the ground truth test set composed of 7500 frames because we ultimately want to see if training models with weak supervision can lead to better models on the ground truth labels. This means that weak supervision can help cut the amount of hand-labelled data needed but it cannot completely eliminate the need for hand-labelled data.

4.1.3 Results of Deep Learning experiments

While running deep learning experiments, owing to the non-convexity of loss function we should ideally run the same experiment multiple times, with different random initializations, with the aim of finding the global optimum. Due to limited time and compute resources, however, we only ran each experiment once, which may have resulted in achieving local optimum. This can be seen in Fig. 4 where the F1 score dips at training on 300% data (i.e. 192k data points) with imperfect Rekall labels. The results can be seen in a more comprehensive format in Table 1.

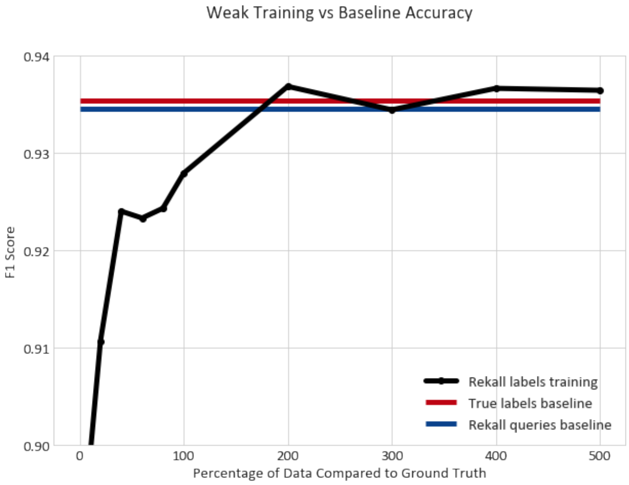

Figure 4: ResNet learning curve

4.2 Classical machine learning

In addition to the cost of having humans generate labels, there is the significant time and expense associated with training or fine-tuning deep neural networks. Each experiment took 2 or 3 days to run and we relatively quickly exhausted our compute budget. As such, we explored what sorts of results could be achieve by extracting various features from the frames and then running classical machine learning algorithms that did not require significant computational overheard.

For each frame we computed the mean and standard deviation of the pixel intensities for the frames as a whole as well as for each channel (R, G and B), producing a feature vector $x^{(i)} \in \mathbb{R}^8$. We then expanded this by dividing the frames into nine equal regions and then re-computing the features for each region. An examination of the data, as shown in Figs. 5 and 6, with the positive examples

Deep Learning **size of training data**	**Accuracy**	**Precision**	**F1 score**
6400 (10%)	0.9333	0.8636	0.8989
12800 (20%)	0.9425	0.8950	0.9107
25600 (40%)	0.9510	0.9080	0.9240
38400 (60%)	0.9506	0.9082	0.9233
51200 (80%)	0.9521	0.9245	0.9243
64000 (100%)	0.9542	0.9250	0.9279
64000 (100%) (g-truth)	0.9598	0.9535	0.9353
128000 (200%)	0.9600	0.9360	0.9368
192000 (300%)	0.9585	0.9342	0.9344
256000 (400%)	0.9604	0.9485	0.9366
320000 (500%)	0.9604	0.9509	0.9364

Table 1: Deep Learning Accuracy Results

(e.g. commercials, Fig. 2) in red and the negative examples (e.g. programming, Fig. 3) in blue, demonstrated a divergence in distributions. As such, we conducted binary classification with Gaussian Discriminant Analysis and Support Vector Machines.

4.2.1 Results of classical ML experiments

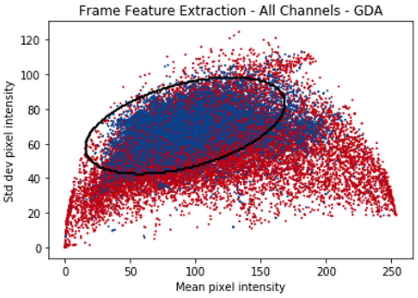

Figure 5: Gaussian Discriminant Analysis

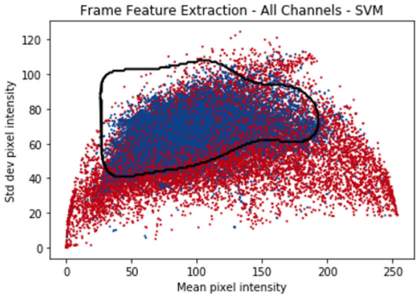

Figure 6: Support Vector Machine (RBF kernel)

While Figs. 5 and 6 depict the input data and decision boundaries in two dimensions, for ease of viewing, in $\mathbb{R}^8$ the single-Σ GDA produced an accuracy of 83% with an F1 score of 68% while the SVM with the RBF kernel produced an accuracy of 84% and an F1 score of 70% (see Table 2). (It is interesting to note how the GDA performed slightly better with a single co-variance matrix as opposed to two, independent co-variance matrices.) With $x^{(i)} \in \mathbb{R}^{60}$, however, the results begin to significantly approach the baselines, with an accuracy of 91% and F1 score of 86%.

It is worth noting that these classical ML algorithms converged to stable accuracy and F1 scores with much less data. The results presented in Table 2 were achieved with only a few thousand examples, as opposed to tens of thousands or even hundreds of thousands.

While still far from the accuracy of the deep learning model, this analysis could be run in minutes (although the linear kernel took hours in $\mathbb{R}^2$ and didn't converge after 6 hours in $\mathbb{R}^8$ as so was terminated; additionally the data is not linearly separable so we cannot expect the linear SVM to perform well) and clearly demonstrated the ability to quickly perform some crude classification.

5 Conclusion and Future Work

As it can be seen from Fig. 4, for commercial detection, the ResNet-50, as shown in Fig. [2], when trained on sufficiently large sets of imperfectly labelled data (of sizes 128k, 192k, 256k, 320k),

Algorithm	$x^{(i)} \in$	**Accuracy**	**F1 Score**	**Confusion Matrix**	**Norm. Conf. Matrix**
GDA - 1-Σ	$\mathbb{R}^2$	0.7791	0.5192	$\begin{bmatrix}894, 179 \\ 1477, 4946\end{bmatrix}$	$\begin{bmatrix}0.119, 0.024 \\ 0.197, 0.660\end{bmatrix}$
	$\mathbb{R}^8$	0.8325	0.6841	$\begin{bmatrix}1359, 243 \\ 1012, 4882\end{bmatrix}$	$\begin{bmatrix}0.181, 0.032 \\ 0.135, 0.651\end{bmatrix}$
GDA - 2-Σs	$\mathbb{R}^2$	0.7882	0.5731	$\begin{bmatrix}1066, 283 \\ 1305, 4842\end{bmatrix}$	$\begin{bmatrix}0.142, 0.038 \\ 0.174, 0.646\end{bmatrix}$
	$\mathbb{R}^8$	0.8201	0.6719	$\begin{bmatrix}1380, 357 \\ 991, 4768\end{bmatrix}$	$\begin{bmatrix}0.184, 0.048 \\ 0.132, 0.636\end{bmatrix}$
SVM - σ	$\mathbb{R}^2$	0.6234	0.3685	$\begin{bmatrix}824, 1276 \\ 1547, 3849\end{bmatrix}$	$\begin{bmatrix}0.110, 0.170 \\ 0.206, 0.513\end{bmatrix}$
	$\mathbb{R}^8$	0.6000	0.3211	$\begin{bmatrix}709, 1336 \\ 1662, 3789\end{bmatrix}$	$\begin{bmatrix}0.094, 0.178 \\ 0.221, 0.505\end{bmatrix}$
SVM - RBF	$\mathbb{R}^2$	0.7896	0.5449	$\begin{bmatrix}944, 150 \\ 1427, 4975\end{bmatrix}$	$\begin{bmatrix}0.126, 0.020 \\ 0.190, 0.664\end{bmatrix}$
	$\mathbb{R}^8$	0.8436	0.7048	$\begin{bmatrix}1399, 200 \\ 972, 4925\end{bmatrix}$	$\begin{bmatrix}0.187, 0.026 \\ 0.130, 0.657\end{bmatrix}$
	$\mathbb{R}^{26}$	0.8714	0.7813	$\begin{bmatrix}1722, 315 \\ 649, 4810\end{bmatrix}$	$\begin{bmatrix}0.230, 0.042 \\ 0.087, 0.641\end{bmatrix}$
	$\mathbb{R}^{60}$	0.9105	0.8557	$\begin{bmatrix}1990, 290 \\ 381, 4835\end{bmatrix}$	$\begin{bmatrix}0.265, 0.039 \\ 0.051, 0.645\end{bmatrix}$

Table 2: Classical Machine Learning Accuracy Results

outperformed the model trained on smaller perfectly labelled data (64k) as well as the Rekall queries baseline. It will be interesting to see if this discovery can be applied to detecting other events, like interviews with politicians, action sequences in movies and shot scales. Moving forward, we can investigate richer forms of weak supervision, such as using the Rekall queries to output the probabilities of classes where we can then make the models more informed, as compared to just providing them with binary labels.

6 Contributions

Our code may be found on GitHub [4]. Mark refactored the code base to facilitate running experiments, trained with Rekall labels on a local GPU and wrote and ran the classical ML experiments. Aasavari extracted feature from ORB (we deduced that number of ORB keypoints could not be used a reliable feature for ML models), learnt SIFT and SURF but they could not be used because of patent issues. Sundararajan ran experiments on the K-80 GPU with Rekall labels and authored code for pulling as well as storing data in relevant structures from the directory. We all collaborated in writing this final report.

7 Acknowledgements

We are deeply grateful to Dan Fu (Computer Science PhD candidate, Stanford University) for providing us access to data sets for Deep Learning experiments, sharing his code which we revamped to perform Deep Learning experiments and guiding us throughout the course of this project.

References

[1] Daniel Y. Fu, Will Crichton, James Hong, Xinwei Yao, Haotian Zhang, Anh Truong, Avanika Narayan, Maneesh Agrawala, Christopher Ré, and Kayvon Fatahalian. Rekall: Specifying video events using compositions of spatiotemporal labels, 2019.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[3] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch, 2017.

[4] Mark Cramer, Aasavari Kakne, and Sundararajan Renganathan. cs229-rekall. `https://github.com/mdcramer/cs229-Rekall`, 2019.