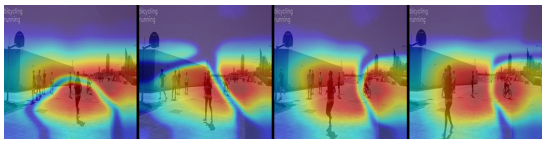
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3126682, IEEE Transactions on Pattern Analysis and Machine Intelligence

Multi-Moments in Time: Learning and Interpreting Models for Multi-Action Video Understanding

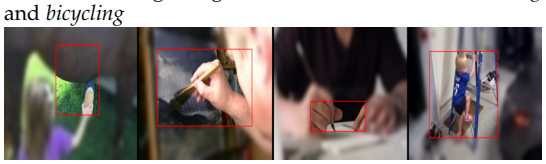
Mathew Monfort[13], Bowen Pan[1], Kandan Ramakrishnan[3], Alex Andonian[1], Barry A McNamara [1], Alex Lascelles [1], Quanfu Fan[23], Dan Gutfreund[23], Rogerio Feris[23], Aude Oliva[13]
[1] MIT CSAIL, [2] IBM Research, [3] MIT-IBM Watson AI Lab

Abstract—Videos capture events that typically contain multiple sequential, and simultaneous, actions even in the span of only a few seconds. However, most large-scale datasets built to train models for action recognition in video only provide a single label per video. Consequently, models can be incorrectly penalized for classifying actions that exist in the videos but are not explicitly labeled and do not learn the full spectrum of information present in each video in training. Towards this goal, we present the Multi-Moments in Time dataset (M-MiT) which includes over two million action labels for over one million three second videos. This multi-label dataset introduces novel challenges on how to train and analyze models for multi-action detection. Here, we present baseline results for multi-action recognition using loss functions adapted for long tail multi-label learning, provide improved methods for visualizing and interpreting models trained for multi-label action detection and show the strength of transferring models trained on M-MiT to smaller datasets.

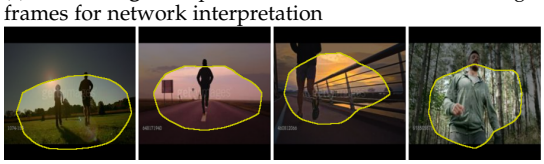
(a) **Multi-Moments in Time** 2 million action labels for 1 million 3 second videos



(b) **Multi-Regions** Localizing multiple visual regions involved in recognizing simultaneous actions, like *running* and *bicycling*



(c) **Action Regions** Spatial localization of actions in single frames for network interpretation



(d) **Action Concepts** Interpretable action features learned by a trained model (i.e. jogging)

1 Introduction

In this paper we present the Multi-Moments in Time dataset (M-MiT). This is a multi-label extension to the Moments in Time dataset (MiT) [27] which includes one million 3-second videos each with a human annotated action label.

Videos by their nature are dynamic. In contrast to images, events can evolve over time and a single action label for an event may not fully capture the set of actions being depicted. For example, a short video of a person *raising* their hand and *snapping* their fingers before *laughing* has multiple actions (e.g. *raising*, *snapping*, *laughing*) as part of the main activity. Single label action datasets for video so do not provide annotations on this full set of information and instead label a single action in each video. This leads to information loss in training, as only partial labels are provided per video and results in an incomplete/incorrect evaluation of trained models. For example, an action model may return *laughing* for the video previously described but the single label provided by the dataset for the video may be *snapping*. In this case evaluation will flag this as an incorrect prediction when it is actually correctly identifying a present action. This problem is exacerbated when we consider simultaneous actions that may take place in other parts of the video (e.g. a person *coughing* in the background). See Section 6.3.2 for empirical validation of these claims.

Several large-scale video datasets provide a large diversity and coverage in terms of the categories of activities and exemplars they capture [20], [17], [27]. However, these labeled datasets only provide a single annotated label for each video and this label may not cover the rich spectrum of events occurring in the video. For example a video of an audience *applauding* may also include a person on a stage *performing*, *playing music*, *singing*, *dancing*, etc. There are a number of existing multi-label datasets for action detection

in video [36], [18], [29], [11], however these datasets either do not share the scale in diversity of content and the number of annotated videos of the large single label datasets, like Kinetics [20] or Moments in Time [27], or suffer from low specificity in their label set, as in Youtube-8M [1] which contains a large set of videos with multiple weak labels sourced from YouTube topic tags. A likely cause is the increased cost of collecting multiple labels for for a large set of videos compared to single label annotations. However, these large-scale datasets have been shown to be important for training models that can capture robust representations that can be used to transfer to smaller datasets downstream [20], [27]. With this in mind we have decided to build a large multi-label dataset that captures the scale and diversity of large single label video datasets. While the visual, audio and semantic complexity of each video is challenging to fully annotate, we took a step toward this goal by extending the Moments in Time dataset, to contain multiple action labels describing one or more events occurring in each clip.

Building a multi-label dataset introduces new challenges in how to train and analyze models for multi-label action detection, including loss function to optimize model learning such as to take advantage of distinct labels for the same visual inputs. For model's representation analysis, recently introduced methods such as Class Activation Mapping (CAM) [41] and Network Dissection (NetDissect) [3] focused on single label interpretation (CAM) or did not provide methods for analyzing learned action concepts (NetDissect). We address these limitations by extending both of these approaches to multi-action models (see Figure 1).

In Section 5.1 we present a multi-label extension to Class Activation Mapping that identifies the important image regions for predicting multiple simultaneous actions in a given scene. In Section 5.2 we outline our approach for adding action concepts to the NetDissect framework for interpreting internal units of a deep network. We additionally examine different multi-label loss functions applied to our dataset in Section 6. This includes a modification to the recently introduced LSEP loss function [22] to support weighted learning for imbalanced datasets to handle the natural long tail distribution of actions in video. In the next section, we describe related work in the area followed by a description of the Multi-Moments in Time dataset and our annotation procedure. Overall the key contributions of this paper include:

- **Multi-Moments in Time (M-MiT)**: A large-scale multi-label action dataset for video understanding with over two million action labels [1].
- **wLSEP**: A novel multi-label loss function that supports learning from an imbalanced class distribution where some classes have more examples than others.
- **mCAM**: Multi-Label Class Activation Mapping for identifying multiple important visual features for model predictions.
- **Action Network Dissection**: We present a single frame dataset (Action Boxes) with bounding boxes on visible actions that we use for incorporating **action concepts** into Network Dissection [3] to identify key interpretable features learned by action models.

1. The data is available on our site, http://moments.csail.mit.edu.

2 Related work

2.1 Action Datasets for Video Understanding

Video understanding has recently seen fast progress partly due to the availability of large scale video datasets for action recognition such as Kinetics [20], Moments in Time [27] and ActivityNet [8]. These large datasets are used to pretrain large video models that can be fine-tuned on smaller action recognition datasets such as UCF101 [31], HMDB [21], THUMOS [19], and "something something" [17]. Fine-tuning in this way allows the models to learn robust representations from the large variety of videos in the large datasets and transfer their features to the smaller sets.

Each of these datasets contains a single action label for each video clip provided. This has been shown to be a limitation as actions tend to be intricately connected [36] and a single label on a video often misses background events and richer descriptions of events. For example, a video of a person swimming in a pool may be labeled with the action "swimming". This is correct, but consider that they are participating in a swim meet and are racing other swimmers. Adding the labels "competing" and "racing" may help capture this information. Additionally, there may be a person "running" along side the pool in the background. Our proposed dataset aims to address this problem and provide multiple labels for each video. There have been a number of multi-label video datasets introduced for action detection. However, they either tend to be much smaller in scale than the large single-label datasets, such as MultiThumos [36], AVA [18] and Charades [29], or constrained to specific domains, such as EPIC-KITCHENS [11]. Our proposed Multi-Moments in Time dataset is large-scale, diverse and includes manually annotated action labels for each video.

2.2 Models for Video Understanding

To take advantage of the different datasets described in the previous section, many different models have been proposed. A popular architecture is the two-stream CNNs [30] which separately processes optical flow and RGB frames. 3D CNNs [32] use a 3-dimensional kernel to learn temporal information directly from the frame sequence of a video and I3D models [9] combine 3D CNNs with optical flow to form a two-stream 3D network "inflated" from 2D filters pre-trained on ImageNet [12]. More recently a temporal shift module (TSM) has been used to integrate temporal information into 2D models by shifting frame representations across the temporal dimension [23]. We compare results using both I3D and TSM baseline models on our proposed dataset in Section 6.

2.3 Multi-Label Optimization

Multi-label optimization is a common problem in object dectection where multiple objects may be present in a single image. Given the variety of appearances of objects within a scene, it is challenging for global CNN features to correctly predict multiple labels. Approaches instead use object proposals [16], [35] or learn spatial co-occurrences of labels using LSTMs [33], [37]. Convolutional neural networks (CNN) have also been applied on raw images of multiple objects to learn image-level deep visual representations for multi-label classification [33].

To improve performance on multi-label detection tasks, different loss functions have been proposed. A common approach is binary cross entropy which optimizes each class label individually. However, when treating each class individually, it is also difficult to learn the correlations between different classes [13], [38], [22]. Additionally, this approach may incorrectly penalize some examples which do not have full label coverage as it assumes the absence of a label is a negative label. Another approach is pair-wise ranking [34] which encourages the model to generally assign higher ranks to positive labels. This method has the added benefit of reducing the strength of a models' mistake as incorrect predictions tend to still include highly ranked positive labels. WARP [16], [34] expands on this by including a monotonically increasing weighting function that increases the error for positive labels that are poorly ranked, thus prioritizing them in learning. BP-MLL [38] is another approach that provides a smooth calculation of the ranking error but suffers from exceedingly large values when the positive classes are poorly ranked and the vocabulary size is large. LSEP [22] is a variation of the BP-MLL loss function which addresses it's numerical stability issues but can become dominated by poorly ranked positive classes and does not allow for weighting the loss accross classes in imbalanced datasets. We propose a modification of the LSEP loss function to reduce the effect of these poorly ranked positive classes while integrating a class weighting term that aims to balance learning in datasets with imbalanced label distributions which are common in multi-label problems.

2.4 Model Interpretation

Recent work in network interpretation performs a weighted inflation of the feature maps in the final convolutional layer of a CNN to visualize the important visual regions the network is using to make a prediction [40]. These class activation maps (CAMs) can be thought of as a method for visualizing the learned attention model of the network and have been shown to localize actions occurring in videos [27]. Additionally, network interpretation has been extended to not just visualize important visual regions for a prediction but to also identify the different concepts a network has learned [4], [39]. This is important for understanding the representation of the network and diagnosing class biases that the network is learning. However, the previous work in network dissection (NetDissect) does not include action concepts in its interpretation instead relying on objects, scenes, object parts, textures and materials. In this paper we extend the dataset used (Broden dataset) for this interpretation to include actions as well so that we can better interpret action recognition models. We also extend class activation mapping to visualize specific regions important to different actions in images with multiple detected actions.

3 Building a Multi-Action Dataset

In this section we describe our approach to building and annotating our Multi-Moments in Time dataset (M-MiT) and present summary statistics on our dataset characteristics.

3.1 Annotation

We began by annotating our added action labels using the same process as the Moments in Time dataset. The annotation phase used Amazon Mechanical Turk for crowd sourcing where each worker is presented with a video-verb pair and asked to respond Yes or No if the action is either seen or heard in the video. We additionally embed ground truth video-action pairs into every set that have been manually verified by our team to either be a positive pair (the action is in the video) or a negative pair (the action is not in the video). We randomly replace 10% of the questions in every annotation set with these ground truth pairs and use them to evaluate workers. If a worker gets more than one of these questions wrong then we do not let them submit the set and if a worker repeatedly submits sets with an incorrect ground truth response then we flag the worker and prevent them from completing future jobs. Each action-video pair is annotated at least three times for the training set and at least 5 times for the validation set. This ensures high-quality annotations are collected efficiently. We refer the reader to the Moments in Time paper for more details [27].

3.1.1 Generating Action Candidates

A difficulty of the annotation task is to choose candidate actions that are likely to return positive responses (i.e. actions that occur in the videos). We generated candidate actions for each video using few different techniques.

Approach 1: First, we began by selecting candidates using WordNet [26] relationships where we iteratively picked actions for annotation that were closely linked in the WordNet semantic graph to the existing action labels for each video. We constrained ourselves to the original Moments in Time vocabulary of 339 actions in order to simplify this process and stopped our candidate selection when the harvest rate (positive worker response rate) dropped below 20%. For example, if we have a video with the action *running*, the WordNet graph structure shows a short semantic path (distance between word nodes in the graph) to the action *exercising* and begin annotating all videos that have the action *running* with the action *exercising*. We iteratively increase the node distance used to find related actions in the graph until the harvest drops below the 20%. It is important to note that the hierarchy in action relationships are less well defined than that of objects. For example, not every video of *eating* is defined by the action *dining*. Similarly there are videos in the dataset of animals *running* that are *chasing* each other which do not relate to *exercising* in the same way a video of a person *jogging* does. It is important to note that these approaches are to generate action candidates that we verify through multiple rounds of human annotation. In this way we are able to verify whether a video of a person *running* is in fact *jogging* as opposed to *sprinting* and/or *chasing*.

Approach 2: Similar to the WordNet approach, we use Word2Vec [25] similarity scores to select action candidates based on the existing action labels for a video. Word2Vec allows us to generate candidates for actions that describe commonly co-occurring events such as *stirring* and *boiling* or *running* and *jumping*. Similarly, this approach helps to generate opposite, but commonly co-ocurring action pairs such as *opening* and *closing* or *throwing* and *catching*. A limitation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3126682, IEEE Transactions on Pattern Analysis and Machine Intelligence

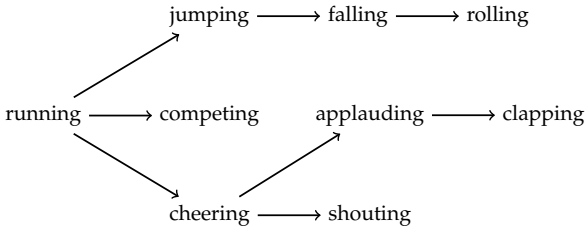


Fig. 2: An example of the path of generating new candidate verbs from previously annotated classes.

of this approach is that it is based on NLP embeddings and may not be fully reflective of action relationships in video.

Approach 3: The final approach we use for candidate selection is to run a trained model over the videos and select the top-1 predictions for each video that are not currently annotated. The model used for this task was trained on the single label dataset and is provided by the authors of the original work [27]. We restricted the actions to the top-1 predictions as taking a larger range of predictions quickly dropped the harvest rate which limited our ability to scale. This approach can capture events that may be seemingly unrelated but still co-occurring in a video such as a child *jumping* in a doorway while an adult is *cooking* in a kitchen. This does introduce some model bias into the candidate generation process but when combined with the other approaches we consider it acceptable to reach a stronger coverage of the events in our videos.

For each of the above approaches we regenerate new candidates as new labels are verified through annotation. Figure 2 outlines the candidate generation path of a video that was originally only labeled with the action *running*. In this case we first generate and annotate the action candidates *jumping*, *competing* and *cheering* which then eventually lead to subsequent annotated actions such as *clapping* and *rolling* such that we are able to provide a much more thorough description of the actions taking place in the video than simply *running*. Also note that *cheering*, *applauding*, *clapping* and *shouting* are auditory actions that may not be visible in the video itself but instead heard.

These methods of candidate generation allow us to efficiently annotate multiple actions in each video, but it does result in videos where we do not annotate every present action. We consider this an acceptable trade-off for the ability to scale to such a large set of labels and we ensure that in training models we do not directly penalize predictions for actions that are not labeled (see Section 4).

3.2 Dataset Statistics

There are 802,244 video-label pairs in the training set, and 33,900 in the validation set of the Moments in Time Dataset. We increased this dataset to 2.01 million labels for 1.02 million videos by adding new videos, generating and annotating action candidates as described in the previous section and adding new action classes to the dataset.

Our *Multi-Moments in Time* dataset includes 292 annotated action classes where new actions have been added to the Moments in Time vocabulary (e.g. *skateboarding*), ambiguous/noisy actions were removed (e.g. *working*) and similar actions have been merged into a single class (e.g. *rising* and *ascending* -> *ascending/rising*). We additionally added novel actions not found in the original Moments in Time (e.g. *unplugging*) and removed actions that we deemed either too vague or noisy based on a random sample of 500 videos per class (e.g. *working*). Figure 4 shows the changes made to the class vocabulary from MiT.

This new vocabulary should cover an increased breadth

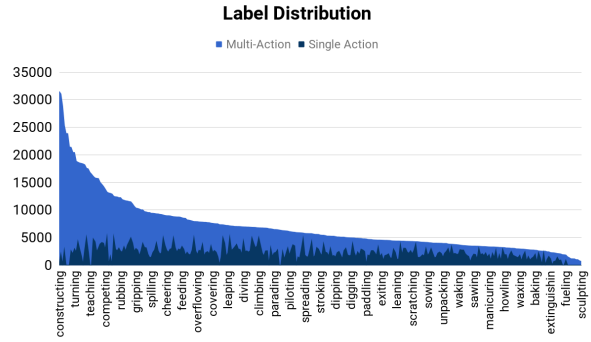


Fig. 3: The label distribution of our proposed Multi-Moments in Time dataset compared to the Moments in Time dataset

Combined

stroking/petting	teaching/instructing
smiling/grinning	rotating/spinning
stacking/piling	shoveling/digging
breaking/destroying	eating/feeding
filming/photographing	tearing/ripping
cleaning/washing	smelling/sniffing
constructing/assembling	hitting/colliding
ascending/rising	laughing/giggling
combusting/burning	buying/selling/shopping
descending/lowering	snuggling/cuddling/hugging

Added

paddling	waxing	inserting	ironing
weightlifting	scooping	unplugging	ice+skating
parading	erasing	rubbing	approaching
squeezing	laying	massaging	snowboarding
sculpting	bandaging	kayaking	skateboarding
punting	shivering	sleeping	smashing
texting	dialing	flossing	gasping

Removed

coaching	serving	exiting	imitating
boarding	sketching	blocking	drenching
flicking	paying	cramming	bouncing
stopping	putting	tickling	guarding
working	talking	joining	building
asking	entering	raising	leaning
starting	placing	watering	playing

Fig. 4: Lists of vocabulary differences in M-MiT compared to MiT. We combined similar classes, removed ambiguous/noisy classes and added some new classes found during annotation.

of events while improving the boundary between different classes. Using this new action set we were able to increase the training set to include over 2 million labels where 553,535 videos are annotated with more than one label and 257,491 videos are annotated with three or more labels. In addition, we have created new validation and test sets each consisting of 10K videos with over 30K labels each. Figure 3 shows a comparison of the distribution of our proposed M-MiT dataset to single label MiT dataset. The classes in the training set have an average of 6,432 example videos and a median of 5,478 videos while the classes in the validation set have an average of 96 example videos and a median of 31 videos.

4 Multi-Label Loss Functions

In this section we present a set of multi-label loss functions that we use to train models on our multi-action dataset. For each loss we normalize by the number of labels in each data example to handle the variability in the number of labels per video and incorporate a class weighting term w_i that helps to balance learning when the training set has an imbalanced number of examples per label. This imbalance is common for action datasets as action labels tend to follow a long tail distribution in practice. Additionally, in prior work sampling was used to address the quadratic complexity of the different loss functions [22]. However we have found that parallelizing the loss computation through matrix operations eliminates the need for sampling. We compare the results of each approach in Section 6.

4.1 BCE

A common approach to multi-label optimization is to optimize for binary cross entropy and treat each label as an independent classifier. Given the output of a model, x_i for class i and an indicator of whether this class is positive, y_i,

$$\mathcal{L}_{BCE} = -w_i[y_i \log x_i + (1 - y_i) \log(1 - x_i)], \quad (1)$$

where w_i is a weight balancing term that scales the strength of the loss applied to class i to improve optimization for underrepresented classes in imbalanced datasets.

However, this has been shown to not consider correlations between different classes [13], [38], [22] and makes the assumption that cases where a class does not have a positive label are negative examples. While we have verified all positive labels in the proposed dataset, we do not assume that an unlabeled class is guaranteed to not be present.

4.2 WARP

Pair-wise ranking presents another approach to multi-label optimization stemming from the motivation that while it is important to correctly classify a positive label, it is also important to reduce the strength of a mistake by encouraging the model to assign higher ranks to positive labels [34]. The WARP loss function [16], [34] proposes a weighted pair-wise ranking function that prioritizes poorly ranked positive classes via an additional monotonically increasing weighting function $\mathcal{W}(x_i)$,

$$\mathcal{L}_{WARP} = \frac{1}{|\mathcal{Y}|} \sum_{i \in \mathcal{Y}} w_i \mathcal{W}(\mathcal{R}(x_i)) \sum_{j \notin \mathcal{Y}} \max(0, 1 + x_j - x_i), \quad (2)$$

where $\mathcal{Y}$ represents the set of positively labeled classes.

The rank of a predicted class, $\mathcal{R}(x)$, is used to increase the error penalization for lower ranked classes. In practice we set $\mathcal{W}(r) = \sum_{k=1}^{r} \frac{1}{k}$.

4.3 LSEP

The recently proposed multi-label ranking function LSEP [22] takes the log of the BP-MLL function [38], with the addition of a single bias term, to increase the numerical stability,

$$\mathcal{L}_{LSEP} = \log\big(1 + \sum_{i \in \mathcal{Y}} \sum_{j \notin \mathcal{Y}} e^{x_j - x_i}\big). \quad (3)$$

This prioritizes positive classes that are poorly ranked without the need of an additional weighting function as in WARP. However, poorly ranked classes can dominate the loss and the function does not allow for a weighting term to be included to improve optimization in imbalanced datasets.

wLSEP

We propose modifying the LSEP loss to add a weight balancing term to aid optimization in our imbalanced dataset. This is not straight forward as the gradient computation for each class is non-separable from other classes. We address this by modifying the LSEP loss to apply individually to each class with a positive label allowing us to simply add a weight term, w_i, to the function,

$$\mathcal{L}_{wLSEP} = \frac{1}{|\mathcal{Y}|} \sum_{i \in \mathcal{Y}} w_i \log\big(1 + \sum_{j \notin \mathcal{Y}} e^{x_j - x_i}\big). \quad (4)$$

This looses the prioritization of low ranked positive classes in the original loss function by summing the softmax of the ranking error $(x_j - x_i)$ for each positive label i, however it avoids the problem of taking a global softmax over all positive labels which prevents the loss from being dominated by poorly ranked positive labels.

5 Analyzing Multi-Label Models

5.1 Multi-Label Class Activation Mapping

Here we extend the class activation mapping (CAM) [40] technique to multi-label tasks. The simplest approach is to inflate the CAM for each predicted action, taking the maximum value of each map so that we can visualize the discriminative features used for our multi-label prediction. However, in practice, the combined CAM filters cover a large area of the input image making it difficult to gain a useful understanding of the models' decisions. CAM is very useful for identifying the key visual features that contribute to a model's prediction. In our multi-label setting we want to identify the unique features that are contributing to each prediction. This is clear for very different actions such as *jumping* and *swimming* but the distinct features used to detect *dining* and *eating* may be more ambiguous. We care about separating these regions to better understand our model and how it is making it's decisions. This can help us identify any biases learned by the model and may help us understand small differences between seemingly similar actions.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3126682, IEEE Transactions on Pattern Analysis and Machine Intelligence

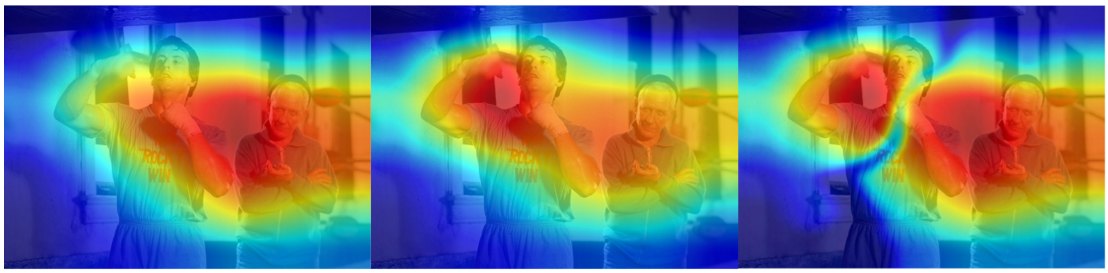


Fig. 5: **Region Separation:** Example showing single class CAM images and the separation of relevant features in a multi-class CAM image. The red regions specify important areas of the image used by the model to infer the detected action.

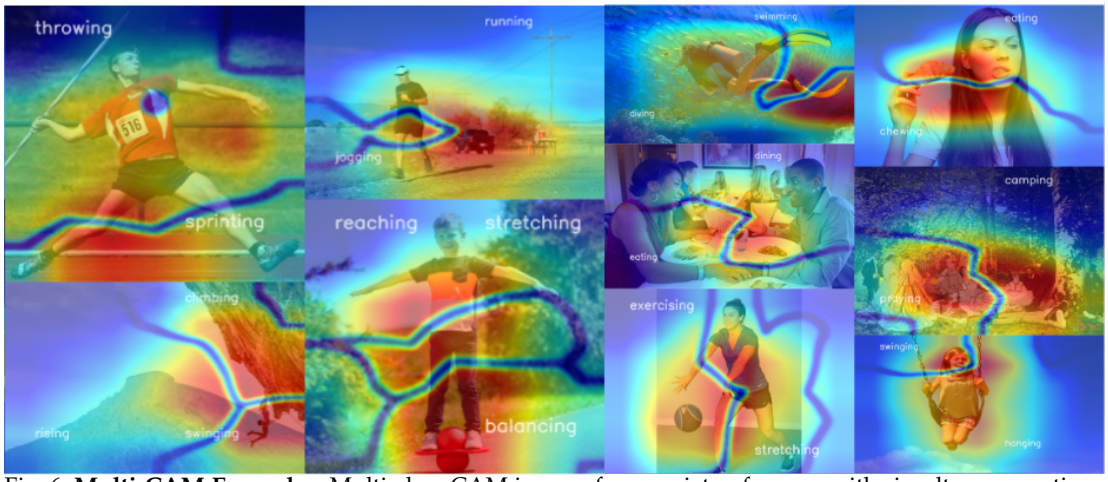


Fig. 6: **Multi-CAM Examples:** Multi-class CAM images for a variety of scenes with simultaneous actions. Action labels are placed near the important image regions used by the model for identifying each specific action. This area is signified by the red overlay with blue edges separating image regions distinct to each detected action showing that our model is able to localize multiple actions present in each scene despite not being trained for localization.

To address this issue, before we take the maximum value of each actions CAM, we first compare the different CAM filters for each of our predicted classes and when two filters with a cosine distance greater than 1^{-4} have similar values at the same pixel locations we set the values of the pixels for both filters to zero. This eliminates the overlapping area and, after performing maximum pooling over the altered CAM filters, creates a boundary between the distinct features associated with each predicted action. Only altering filters with a cosine distance greater than some threshold, in this case 1^{-4}, ensures that detections that use the same regions do not have their CAMs erased. In practice we found that this value strikes a strong balance in generating distinct visualizations of regions used by the model for multiple detections while merging highly similar regions used by the model such as *emptying* and *pouring* in the image of wine being poured into a glass in Figure 6. If a strict comparison of features used between two detected actions in an image is desired then this threshold should be set to zero. To improve the visualizations we also apply Gaussian smoothing with a 5x5 kernel to reduce the sharpness of some of the boundary edges. We show the results of this method for analyzing multi-label predictions in Section 7.1.

5.2 Identifying Learned Action Features

To better analyze our learned action models we extend the set of concepts used by NetDissect [4], [39] to include actions. NetDissect uses the Broden dataset, an image segmentation dataset consisting of pixel-level annotations of objects, scenes, parts, materials and colors. For each image input into a model the labeled segmentations are compared to the different activation regions in internal feature maps of the model. For example, when the activation regions for a feature map share a strong correlation, quantified using IoU, with the segmented regions of a specific object class in a set of images then the feature is interpreted to be correlated to that object. We refer to the NetDissect paper for more information. Here, we extend the Broden dataset to include images with segmented action regions so that we can correlate features with action classes.

5.2.1 Annotation

As with annotating the action labels in the videos, we collect bounding box annotations via Amazon Mechanical Turk (AMT). We begin by selecting a single frame from the center of 500 randomly selected videos from each of the action classes in Moments in Time [27], Kinetics [20] and the Something-Something [17] datasets. We then present a binary annotation task to the workers on AMT asking if an action from the source video's label set is visible in the frame shown. This binary interface is very similar to that used for collecting the action labels for the Moments in Time dataset [27] with the main difference being the use of images rather than video. We run this task for at least 2 rounds of

Model	Loss	Top-1	Top-5	Micro mAP	Macro mAP
I3D	BCE	52.9	77.8	55.4	37.8
	WARP	51.9	79.2	56.5	32.7
	LSEP	52.7	78.8	56.1	31.7
	wLSEP	**58.5**	**81.4**	**61.7**	**39.4**
TSM	BCE	56.9	80.1	58.7	**37.8**
	WARP	55.8	81.0	60.0	33.4
	LSEP	55.2	81.4	59.8	27.6
	wLSEP	**58.1**	**81.6**	**62.4**	35.4
Audio	BCE	6.8	19.3	6.9	3.1
	WARP	6.3	18.6	6.6	2.8
	LSEP	5.3	16.1	6.0	2.6
	wLSEP	**6.9**	**20.9**	**6.9**	**3.2**
Fusion	wLSEP	**59.3**	**82.8**	**61.8**	**41.1**

TABLE 1: **Validation Results:** Performance of the baseline models with different loss functions on the multi-label validation set. We show Top1, Top5 and both micro and macro mAP.

annotation to verify that the action is visible in each frame. We then take the set of verified action-frame pairs and pass them to a separate annotation interface on AMT that asks the workers to select the regions most important in the image for identifying the action. Multiple regions can be selected for an image, as in the jogging example in Figure 8, and the workers are allowed to skip an image if there are no useful regions for detecting the action (i.e. the action is not visible in the image).

We run this region selection task through multiple rounds and only consider overlapping regions from the different rounds as most important for detecting the actions. After this stage the regions selected are cropped from the original images and passed through the binary annotation task previously described for a final verification that the actions are present and recognizable in the selected regions. After our complete annotation process our total set of verified images with segmented action regions consists of 67,468 images from 549 different action classes. Figure 8 displays some examples of the selected regions collected through this process.

5.2.2 Action Interpretation

To integrate our new action region dataset into the Network Dissection framework we first consider each selected region to be a mask on the segmented area of the image relating to the action. This is similar to part, material and object masks used for other segmentation datasets [42], [10], [28], [6]. With the data formatted in this manner we extend the Broden dataset to include our action segmentations and extract the set of learned action concepts detected via NetDissect. This process allows us to identify not just object, scene, texture and color concepts learned by our models, but action concepts as well. In Section 7.2 we show some of the key results from interpreting action networks in this way.

6 Baseline Results

We trained two architectures on our dataset, an inflated 3D ResNet-50 (I3D) [9] for the visual modality and a SoundNet network [2] for learning audio features and compare across different multi-label loss functions.

Training Dataset	Evaluation Dataset	Top-1	Top-5	Micro mAP	Macro mAP
MiT	MiT-S	23.0	53.0	36.7	18.9
MiT	M-MiT-S	40.0	68.5	42.1	24.5
M-MiT	M-MiT-S	45.0	79.1	54.5	32.7

TABLE 2: **MiT vs M-MiT Model Validation Results:** Performance of models trained on MiT and M-MiT on a subset of the M-MiT validation set containing only classes shared between MiT and M-MiT. We evaluate with both a single label version (MiT-S) and a multi-label version (M-MiT-S). We show Top1, Top5 and both micro and macro mAP. We see that multiple labels improves our evaluation of the MiT model while the M-MiT model achieves the best results.

Dataset	Loss	Top-1	Top-5	Micro mAP	Macro mAP
MS-COCO	BCE	90.3	98.6	82.6	63.6
	WARP	92.0	98.8	83.7	60.9
	LSEP	94.0	98.8	85.4	61.0
	wLSEP	**94.3**	**98.9**	**86.3**	**64.5**
VOC	BCE	91.7	97.9	91.6	83.7
	WARP	93.2	99.4	92.9	85.1
	LSEP	92.8	99.2	92.6	85.2
	wLSEP	**93.8**	**99.5**	**93.4**	**87.2**
AVA	BCE	58.3	92.1	65.5	9.3
	WARP	47.6	85.1	54.4	11.2
	LSEP	71.0	95.7	71.2	10.9
	wLSEP	**71.2**	**96.0**	**72.3**	**12.7**
MultiThumos	BCE	63.4	91.1	65.5	58.9
	WARP	75.8	91.8	77.5	63.7
	LSEP	75.1	91.7	77.2	65.5
	wLSEP	**75.9**	**92.0**	**77.6**	**68.0**

TABLE 3: **Loss function comparison:** We validate our proposed wLSEP loss function on four different multi-label datasets. A ResNet-50 model is trained for MS-COCO and VOC while a ResNet-50 I3D model is trained for AVA and MultiThumos

6.1 Models

Inflated 3D Convolutional Networks (I3D)

I3D networks offer improved weight initialization by simply *inflating* the convolutional and pooling kernels of pretrained 2D networks [9]. This is done by initializing the inflated 3D kernel with pretrained weights from 2D models by repeating the parameters from the 2D kernel over the temporal dimension. This greatly improves learning efficiency and performance since 3D models contain a large number of parameters and are difficult to train from scratch. For our experiments we use an inflated 3D ResNet-50 pretrained on ImageNet with 16 frames as input.

Temporal Shift Model (TSM)

We additionally compare results using a ResNet-50 temporal shift module pretrained on ImageNet. This model integrates temporal information into the 2D architecture by shifting frame representations across the temporal dimension [23]. In our experiments we used 8 frames as input into the model.

SoundNet Network (Audio)

In MiT, action classes are labeled for both visual and auditory information. Therefore we feel it would be incomplete to evaluate visual models and not include a model trained on audio. We finetune a SoundNet network [2] which was pretrained on unlabeled videos from Flickr.

Spatio-temporal-Auditory Fusion

We fuse the predictions from the audio and visual modalities by concatenating the spatio-temporal features of the I3D network with the auditory features from SoundNet and train a single linear layer to rank the detected action classes using the loss functions described in the following section.

6.2 Performance Metrics

Accuracy

We report both the top-1 and top-5 classification accuracy for each of our models in order to be consistent with the results reported from the original Moments in Time paper [27]. Top-1 accuracy indicates the percentage of testing videos where the top predicted class is a positive label for the video. Similarly, top-5 accuracy indicates the percentage of the testing videos where any of the top predicted 5 classes for a video is a positive label.

Mean Average Precision (mAP)

We use mAP as our main evaluation metric as it captures errors in the ranking of relevant actions for a video. For each positive label, mAP computes the proportion of relevant labels ranked before it and averages over all of the labels.

In order to properly evaluate our models we report both the micro and the macro mAP. The micro mAP is the mean average precision over all videos and the macro mAP is the average of the mAP for each class. In the case of imbalanced datasets the micro mAP depicts the full performance of the model on the dataset while the macro mAP displays the models class-wise consistency. These numbers can differ greatly as a high mAP in a highly represented class and a low mAP in all other classes can lead to a high micro mAP and a low macro mAP.

6.3 Loss Function Comparison

6.3.1 Results on M-MiT

Table 1 displays results of the models trained using different loss functions on the proposed Multi-Moments in Time dataset. We can see that the proposed wLSEP loss function significantly outperforms the other approaches at optimization. The combination of stable pair-wise ranking with class balancing is very effective in training our multi-label network. Interestingly, for the TSM model BCE produced a slightly better macro mAP even though wLSEP performed better in terms of micro mAP, top-1 and top-5 accuracy. BCE is well suited for weighted learning, as shown in Equation 1, which is likely aiding in the balanced class prediction performance here. We want to point out that wLSEP does beat it in all of the other metrics and achieves a superior macro mAP score with the I3D model (39.4 compared to 37.8 for BCE in both I3D and TSM). This is consistent with the results from other datasets as shown in the following section.

For our audio network the performance separation was much smaller but still results in our proposed wLSEP loss function achieving the best results. For audio we only train and evaluate our models on videos containing audio streams. Fusing the audio and I3D networks via an SVM results in slightly higher top-1 and top-5 accuracies than using I3D alone but does not significantly improve the performance. We choose I3D here due to it having, on average, better performance across the metrics. This small improvement in the fusion results is likely due to the small number of videos in the validation set that contain audio streams as well as the dominance of visual-based labels in the dataset.

6.3.2 Comparing MiT and M-MiT Models

To validate the claims that adding multiple labels improves model evaluation we compared the results of a model trained on MiT using an evaluation set of videos with their original single label, from MiT, to results using the same set of videos and model but with additional annotated labels, from M-MiT, added to each video (Table 2). For a fair comparison we only consider labels that are shared between MiT and M-MiT resulting in a set of 8,761 videos that we use for both single and multi-label evaluation in the table. These results show that adding additional labels to the videos improves our ability to properly evaluate the model. When only considering a single label for each video, the evaluation points to the model performing much worse than when we add in the additional labels from M-MiT. This is due to not having the full set of labels in the MiT evaluation set. We additionally present results comparing the our best M-MiT on the same set of videos and labels showing that adding additional labels in training significantly improves model results. Both models in the table are I3D Resnet50 networks.

6.3.3 Results on Other Datasets

To further validate our wLSEP loss function, Table 3 shows a comparison between wLSEP, LSEP, WARP and BCE on four different multi-label datasets. This includes two image datasets for object detection (Pascal VOC [15] and MS-COCO [24]) as well as two video datasets for action detection (AVA [18] and MultiThumos [36]). For the image datasets we trained a ResNet-50 and we used a ResNet-50 I3D model with 16 frames for the video datasets. We can see in the table that the proposed wLSEP loss function consistently produces the strongest results in each dataset.

6.4 Transfer experiments

To evaluate the strength of the features learned from M-MiT we conducted a set of transfer experiments comparing ResNet-50 I3D models pretrained on Kinetics [20], Moments in Time (MiT) and Multi-Moments in Time (M-MiT). We use the ResNet-50 I3D model trained with the wLSEP loss function as this model gave us the best single stream performance on our dataset (Table 1). We compare our results when transferring to two single label datasets UCF-101 [31] and HMDB [21] as well as three multi-label datasets AVA [18], MultiThumos [36] and Charades [29].

Table 4 shows the results of the transfer task where the top-1, top-5 and mAP scores are calculated by evaluating on the validation set of the dataset used to fine-tune the model. Here we refer to the micro mAP as mAP. We can see from the results that pretraining on M-MiT consistently results in better performance on the multi-label datasets. This makes sense as MiT and Kinetics are single-label datasets and were not trained to handle multiple co-occurring actions. For the single label datasets, MiT achieves very close performance to Kinetics on UCF-101 with M-MiT following in third. On

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3126682, IEEE Transactions on Pattern Analysis and Machine Intelligence

Pretrained	Fine-Tuned UCF-101			HMDB			AVA			MultiTHUMOS			Charades		
	Top-1	Top-5	mAP	Top-1	Top-5	mAP	Top-1	Top-5	mAP	Top-1	Top-5	mAP	Top-1	Top-5	mAP
Kinetics	0.905	0.987	0.942	0.726	0.911	0.809	0.781	0.973	0.772	0.837	0.964	0.821	0.388	0.731	0.303
MiT	**0.908**	0.986	**0.943**	**0.756**	**0.937**	**0.836**	0.802	0.976	0.791	0.849	0.968	0.838	0.383	0.711	0.305
M-MiT	0.892	0.980	0.932	0.740	0.921	0.819	**0.807**	**0.977**	**0.795**	**0.873**	**0.972**	**0.869**	**0.414**	**0.740**	**0.306**

TABLE 4: Dataset transfer performance using ResNet-50 I3D models pretrained on Kinetics, Moments in Time (MiT) and the proposed Multi-Moments in Time dataset (M-MiT).

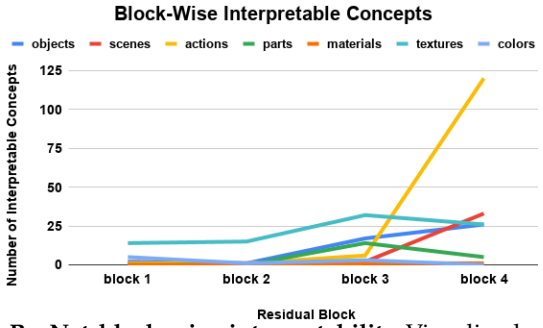


Fig. 7: **ResNet block-wise interpretability** Visualize how different semantic concepts - objects, scenes and actions emerge across residual blocks of the ResNet-50 network.

HMDB, the M-MiT model performs a little worse than MiT and a little better than Kinetics. These results are fairly consistent with prior comparisons between Kinetics and MiT [27] and show that M-MiT pretrained models excel when transferring to multi-label settings.

7 Model Analysis

7.1 Multi-Label Class Activation Mapping

In Figure 5 we see an example of applying our multi-label CAM filter to an image with two actions. Using standard CAM filters it is difficult to disambiguate unique regions used for the different actions detected in the image (*coaching* and *punching*). As with many actions, the image regions used by the model for detection are strongly correlated. We use the proposed multi-CAM approach described in Section 5.1 to highlight the differences between the CAM results for each predicted class. These differences help us visualize how the model has learned to discriminate different classes that may be present in the same image. This is especially useful for classes that may share multiple visual characteristics. In Figure 6 we include a diverse set of examples showing results of this method for different action combinations.

7.2 Learned Action Interpretation

Using the approach described in Section 5.2 we are able to identify a total of 211 concepts consisting of 1 material, 5 part, 26 texture, 26 object, 33 scene and 120 action concepts learned in 2023 different features out of 2048 (Figure 9) units in the final convolutional layer (block4) of a ResNet-50 network trained on the Multi-Moments in Time dataset. Figure 10 highlights some of the learned concepts. For example, the network associates *crawling* with babies as many of our videos of crawling typically depict babies *crawling*. These are the types of data and class biases that are useful to identify via network interpretation that may have gone unnoticed without the ability to identify action concepts.

Category	Concepts	Interpretable Features
Broden	185	1351
Action Regions	140	1995
Broden+Action Regions	211	2023

TABLE 5: Comparison of the number of concepts and interpretable features identified by NetDissect given the Broden dataset, the Action Region dataset and the combined dataset on block 4 of a ResNet-50 trained for action recognition.

Table 5 highlights the fact that including actions in the Broden dataset helps to interpret a much larger portion of the features in block 4 of a ResNet-50 trained for action recognition. With the original Broden set (no actions) NetDissect identified 185 concepts in 1351/2048 features. Adding actions to this set allowed us to identify 2023/2048 features that can be interpreted for 211 different concepts including 120 actions. This jump in the number of interpretable features makes sense for the final block of a model trained for action recognition and suggests that excluding action concepts misses a large portion of useful information when interpreting these models.

The results from the combined set highlight that some of the features previously interpreted by the original Broden set as object or texture concepts are more closely aligned with actions. A few examples of this behavior, and how adding actions to the Broden dataset improves our interpretation, can be seen in Figure 11. For example, unit, or feature, 2025 was previously interpreted to be most closely associated with the texture "bubbly", but after adding the new action regions to the Broden set we found that the feature is actually more strongly correlated with the action "surfing".

Examining these interpretable features and how they contribute to a networks output allows us to build a better understanding of how our models will function when presented with different data. This understanding can be an important tool for model design. For example, improved understanding of model interpretation has recently been shown to be useful in improving adversarial robustness [7], obtaining state-of-the-art autoencoder-based generative models [14] and manipulating the output of a GAN [5].

7.2.1 *Block-wise Interpretability*

To understand how individual units evolve over residual blocks we evaluate the interpretability of features from different blocks of a ResNet-50 network trained for action recognition on the Moments in Time dataset [27] in terms of objects, scenes, actions and textures. In Figure 7 we observe that action features mainly emerge in the last convolutional block (block 4) of the model. It is interesting to note that object and scene features are learned even if the model is not explicitly trained to recognize objects or scenes suggesting that object and scene recognition aids action classification.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3126682, IEEE Transactions on Pattern Analysis and Machine Intelligence

bicycling floating grooming writing jogging

Fig. 8: **Localized action regions:** Bounding boxes annotated around 549 different action categories in 67,468 image frames each selected from unique videos in the Moments in Time, Kinetics, and Something-Something datasets.

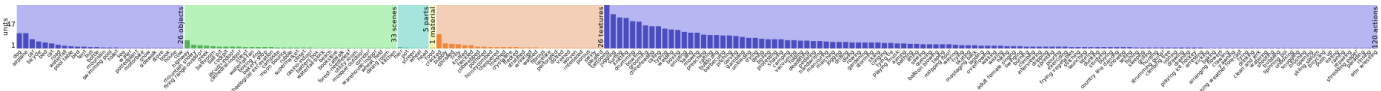


Fig. 9: **Graph of learned action concepts ordered by the number of features associated with each concept.**

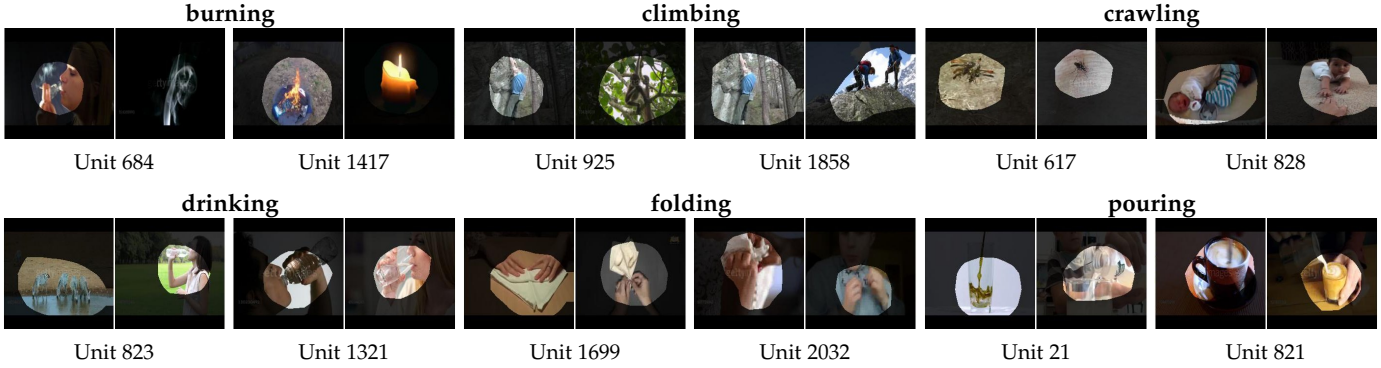


Fig. 10: **Visualization of learned action concepts:** Different features learn different representations of the same action. For example, units 684 and 1417 can both be interpreted as learning the concept of *burning* (top left). However, unit 684 learns to correlate *smoke* with the action while unit 1417 correlates it with a *flame*.

Fig. 11: **Improved feature interpretation:** Examples of the 953 units from the final residual block of a ResNet-50 that changed their main interpretation when actions were added to the Broden dataset.

7.2.2 *Interpretable feature relationships*

Examining these interpretable features and how they contribute to a networks output allows us to build a better understanding of how our models will function when presented with different data. For example, we can consider a feature in the final residual block of the network that has been interpreted as a *highway* scene feature. If we activate only this unit by setting its values to 1 and all other feature (including bias) values to 0 we can identify which actions are correlated with the fact that a video may take place on a *highway*. In this case the actions that achieve the highest output are *hitchhiking*, *towing*, *swerving*, *riding*, and *driving*. These interpretable feature-class relationships make sense as all of these actions are likely to occur near a *highway*.

8 Conclusion

Progress in the field of video understanding will come from many fronts, including training our models with richer and more complete information, so they can start achieving recognition performances in par with humans. Augmenting a large-scale dataset by doubling the number of activity labels, we present baseline results on the Multi-Moments dataset as well as improved methods for visualizing and interpreting models trained for multi-label action detection.

Acknowledgements: This work was supported by the MIT-IBM Watson AI Lab, Google faculty award and SystemsThatLearn@CSAIL award (to A.O), as well as the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number D17PC00341. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

Special thanks to David Bau for assisting in extending the Broden Dataset for Network Dissection and Allen Lee for aiding in the design of our demo.

REFERENCES

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *CoRR*, abs/1609.08675, 2016.

[2] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*. 2016.

[3] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017.

[4] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[5] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[6] S. Bell, P. Upchurch, N. Snavely, and K. Bala. OpenSurfaces: A richly annotated catalog of surface appearance. *ACM Trans. on Graphics (SIGGRAPH)*, 32(4), 2013.

[7] A. Boopathy, S. Liu, G. Zhang, C. Liu, P.-Y. Chen, S. Chang, and L. Daniel. Proper network interpretability helps adversarial robustness in classification. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1014–1023. PMLR, 13–18 Jul 2020.

[8] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[9] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *Proc. ICCV*, 2017.

[10] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[11] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[13] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 681–687, Cambridge, MA, USA, 2001. MIT Press.

[14] P. Esser, R. Rombach, and B. Ommer. A disentangling invertible interpretation network for explaining latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[16] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe. Deep convolutional ranking for multilabel image annotation. *arXiv preprint arXiv:1312.4894*, 2013.

[17] R. Goyal, S. Kahou, V. Michalski, J. Materzyńska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. *arXiv preprint arXiv:1706.04261*, 2017.

[18] C. Gu, C. Sun, S. Vijayanarasimhan, C. Pantofaru, D. A. Ross, G. Toderici, Y. Li, S. Ricco, R. Sukthankar, C. Schmid, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. *arXiv preprint arXiv:1705.08421*, 2017.

[19] Y. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2014.

[20] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[21] H. Kuehne, H. Jhuang, R. Stiefelhagen, and T. Serre. Hmdb51: A large video database for human motion recognition. In W. E. Nagel, D. H. Kröner, and M. M. Resch, editors, *High Performance Computing in Science and Engineering '12*, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[22] Y. Li, Y. Song, and J. Luo. Improving pairwise ranking for multi-label image classification. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1837–1845, 2017.

[23] J. Lin, C. Gan, and S. Han. Tsm: Temporal shift module for efficient video understanding. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[26] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[27] M. Monfort, A. Andonian, B. Zhou, K. Ramakrishnan, S. A. Bargal, T. Yan, L. Brown, Q. Fan, D. Gutfruend, C. Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–8, 2019.

[28] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[29] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. *CoRR*, abs/1604.01753, 2016.

[30] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[31] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[32] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *CVPR*, 2015.

[33] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2285–2294. IEEE, 2016.

[34] J. Weston, S. Bengio, and N. Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2764–2770. AAAI Press, 2011.

[35] H. Yang, J. Tianyi Zhou, Y. Zhang, B.-B. Gao, J. Wu, and J. Cai. Exploit bounding box annotations for multi-label object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–288, 2016.

[36] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *arXiv preprint arXiv:1507.05738*, 2015.

[37] J. Zhang, Q. Wu, C. Shen, J. Zhang, and J. Lu. Multi-label image classification with regional latent semantic dependencies. *IEEE Transactions on Multimedia*, 2018.

[38] M.-L. Zhang and Z.-H. Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Trans. on Knowl. and Data Eng.*, 18(10):1338–1351, Oct. 2006.

[39] B. Zhou, D. Bau, A. Oliva, and A. Torralba. Interpreting deep visual representations via network dissection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018.

[40] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016.

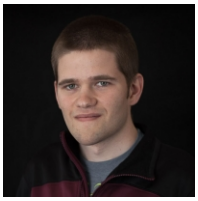
[41] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[42] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ade20k dataset. *arXiv preprint arXiv:1608.05442*, 2016.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3126682, IEEE Transactions on Pattern Analysis and Machine Intelligence

Mathew Monfort is a Research Scientist at the MIT Computer Science and Artificial Intelligence Laboratory. He received a Ph.D. in computer science from the University of Illinois at Chicago in 2016, a M.S. in Computer Science from Florida State University in 2011 and a B.A. in Mathematics from Franklin and Marshall College in 2009. His research focuses on machine learning, inverse planning, deep learning and areas related to learning from human behavior.

Kandan Ramakrishnan is a Postdoctoral associate at the MIT Computer Science and Artificial Intelligence Lab. He received a Ph.D in computer science from Universiteit van Amsterdam in 2017. Before that he received a M.S. degree in Electrical Engineering from University of Minnesota, Twin Cities and B.E. in Electrical Engineering from S.R.M. University, Chennai. His research interests are computer and human vision, machine learning.



Barry A McNamara is currently a Software Engineer at YouTube. He received his M.Eng. and B.S. in Computer Science at the Massachusetts Institute of Technology in 2019. His research interests are in human-computer interaction, including user interfaces and programming languages.

Quanfu Fan is a Research Staff Member at the MIT-IBM Watson AI Lab in Cambridge MA. He is a member of the core R&D team of the IBM Intelligent Video Analysis (IVA) offering. He received his Ph.D. degree in Computer Science from the University of Arizona. His research interests are computer vision and machine learning, with a focus on video understanding.

Rogerio Schmidt Feris is a principal scientist and manager of the computer vision and multimedia department at IBM T.J. Watson Research Center. He joined IBM in 2006 after receiving a Ph.D. from the University of California, Santa Barbara. His work has been integrated into multiple IBM products including Watson Visual Recognition, Watson Media, and Intelligent Video Analytics.

Bowen Pan is a Ph.D. student at the MIT Computer Science and Artificial Intelligence Laboratory. He received a B.Eng. degree in Electrical Engineering from Shanghai Jiao Tong University. His research interests include computer vision and machine learning.

Alex Andonian is a Ph.D. student at the MIT Computer Science and Artificial Intelligence Laboratory. He received a B.S. in Neuroscience, Physics and Mathematics from Bates College in 2017. His research interests include computer vision, machine learning and computational neuroscience.

Alex Lascelles gained an undergraduate masters degree in Physics and Astronomy at the University of Southampton in the UK.He spent his final year of research abroad at the Harvard-Smithsonian Center for Astrophysics under the supervision of Dr Cecilia Garraffo and Dr Jeremy Drake and obtained a masters under Prof Joydeep Bhattacharya from Goldsmiths, University of London. He currently works in Dr Aude Oliva's lab at MIT as a research assistant.

Dan Gutfruend is a research staff member at the IBM Research lab in Cambridge MA and a principal investigator at the MIT-IBM Watson AI Lab. in 2005 he received a Ph.D. in computer science from the Hebrew University in Jerusalem Israel, where he also earned his B.Sc. and M.Sc. degrees. Prior to joining IBM he was a postdoctoral fellow and a lecturer at Harvard University and MIT. Currently his research focuses on machine learning and computer vision.

Aude Oliva is a Senior Research Scientist at the MIT Computer Science and Artificial Intelligence Laboratory. After a baccalaureate in Physics and Mathematics, she received M.Sc and Ph.D degrees in Cognitive Sciences from the Institut National Polytechnique of Grenoble, France. She received the 2006 National Science Foundation Career award, the 2014 Guggenheim award and the 2016 Vannevar Bush fellowship. Her research spans cognitive science, neuroscience and computer vision.