RESOUND: Towards Action Recognition Without Representation Bias

Yingwei Li, Yi Li(✉), and Nuno Vasconcelos

UC San Diego, San Diego, USA
{yil325,yil898,nvasconcelos}@ucsd.edu

Abstract. While large datasets have proven to be a key enabler for progress in computer vision, they can have biases that lead to erroneous conclusions. The notion of the representation bias of a dataset is proposed to combat this problem. It captures the fact that representations other than the ground-truth representation can achieve good performance on any given dataset. When this is the case, the dataset is said not to be well calibrated. Dataset calibration is shown to be a necessary condition for the standard state-of-the-art evaluation practice to converge to the ground-truth representation. A procedure, RESOUND, is proposed to quantify and minimize representation bias. Its application to the problem of action recognition shows that current datasets are biased towards static representations (objects, scenes and people). Two versions of RESOUND are studied. An Explicit RESOUND procedure is proposed to assemble new datasets by sampling existing datasets. An implicit RESOUND procedure is used to guide the creation of a new dataset, Diving48, of over 18,000 video clips of competitive diving actions, spanning 48 fine-grained dive classes. Experimental evaluation confirms the effectiveness of RESOUND to reduce the static biases of current datasets.

1 Introduction

In recent years, convolutional neural networks (CNNs) have achieved great success in image understanding problems, such as object recognition or semantic segmentation. A key enabling factor was the introduction of large scale image datasets such as ImageNet, MS COCO, and others. These have two main properties. First, they contain enough samples to constrain the millions of parameters of modern CNNs. Second, they cover a large enough variety of visual concepts to enable the learning of visual representations that generalize across many tasks. While similar efforts have been pursued for video, progress has been slower. One difficulty is that video classes can be discriminated over different time spans. This results on a hierarchy of representations for temporal discrimination.

Static representations, which span single video frames, lie at the bottom of this hierarchy. They suffice for video classification when static cues, such as objects, are discriminant for different video classes. For example, the classes in the "playing musical instrument" branch of ActivityNet [3] differ in the instrument being played. The next hierarchy level is that of short-term motion representations, typically based on optical flow, spanning a pair of frames. They

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11210, pp. 520–535, 2018.
https://doi.org/10.1007/978-3-030-01231-1_32

suffice when classes have identical static cues, but different short-term motion patterns. Finally, the top level of the hierarchy includes representations of video dynamics. These address video classes with identical static elements and short-term motion, but different in the temporal arrangement of these elements. They are needed for discrimination between classes such as "triple jump" and "long jump," in an Olympic sportscast, with identical backgrounds and short-term motions (running and jumping), only differing in the composition of the latter.

Clearly, more sophisticated temporal reasoning requires representations at higher levels of the hierarchy. What is less clear is how to evaluate the relative importance of the different levels for action recognition. Current video CNNs tend to use very simple temporal representations. For example, the prevalent two-stream CNN model [17] augments a static CNN with a stream that processes optical flow. There have been attempts to deploy networks with more sophisticated temporal representations, e.g. RNN [5,24] and 3D CNN [7,20], yet existing benchmarks have not produced strong evidence in favor of these models. It is unclear, at this point, if this is a limitation of the models or of the benchmarks.

One suspicious observation is that, on many of the existing datasets, static representations achieve reasonably good performance. This is because the datasets exhibit at least three types of static biases. The first is object bias. For example, "playing piano" is the only class depicting pianos, in both ActivitNet and UCF101. A piano detector is enough to pick out this class. The second is scene bias. For example, while the "basketball dunk" and "soccer juggling" classes have distinct temporal patterns, they can be discriminated by classifying the background into basketball court or soccer field. Finally, there is frequently a person bias. While classes like "brushing hair" contain mostly face close-ups, "military marching" videos usually contain long shots of groups in military uniforms.

It should be noted that there is nothing intrinsically wrong about biases. If a person detector is useful to recognize certain actions, action recognition systems should use person detectors. The problem is that, if care is not exercised during dataset assembly, these biases could undermine the evaluation of action recognition systems. For example, an action recognition dataset could be solvable by cobbling together enough object detectors. This would elicit the inference that "action recognition is simply object recognition." Such an inference would likely be met with skepticism by most vision researchers. The problem is compounded by the fact that biases do not even need to be obvious, since modern deep networks can easily identify and "overfit to" any biases due to a skewed data collection. Finally, to make matters worse, biases are cumulative, i.e. static biases combine with motion biases and dynamics biases to enable artificial discrimination. Hence, investigating the importance of representations at a certain level of hierarchy requires eliminating the biases of all levels below it.

These problems are frequently faced by social scientists, who spend substantial time introducing "controls" in their data: A study of whether exercise prevents heart attacks has to "control" factors such as age, wealth, or family

history, so that subjects are chosen to avoid biases towards any of these factors. Similarly, vision researchers can only draw conclusions from their datasets if they are not biased towards certain representations.

In this work, we investigate the question of assembling datasets without such biases. Despite extensive recent efforts in dataset collection, this question has received surprisingly little attention. One reason is that, until recently, vision researchers were concerned about more fundamental forms of bias, such as dataset bias [19], which captures how algorithms trained on one dataset generalize to other datasets of the same task. Dataset bias can be analyzed with the classical statistical tools of bias and variance. It occurs because (1) learning algorithms are statistical estimators, and (2) estimates from too little data have high variance and generalize poorly. With the introduction of large datasets, such as ImageNet [4], dataset bias has been drastically reduced in the past few years. However, simply collecting larger datasets will not eliminate representation bias.

While dataset bias is a property of the algorithm (ameliorated by large datasets), representation bias is a property of the dataset. As in social science research, it can only be avoided by controlling biases during dataset collection. We formalize this concept with the notion of a *well calibrated dataset*, which only favors the ground-truth representation for the vision task at hand, i.e. has no significant biases for other representations. We then show that the standard vision practice of identifying the "state of the art" representation only converges to the ground-truth representation if datasets are well calibrated. This motivates a new measure of the representation bias of a dataset, which guides a new *RepreSentatiOn UNbiased Dataset* (RESOUND) collection framework.

RESOUND is a generic procedure, applicable to the assembly of datasets for many tasks. Its distinguishing features are that it (1) explicitly defines a set of representation classes, (2) quantifies the biases of a dataset with respect to them, and (3) enables the formulation of explicit optimization methods for assembling unbiased datasets. In this work, this is in two ways. First, by using RESOUND to guide the assembly of a new video dataset, Diving48, aimed for studies on the importance of different levels of the representation hierarchy for action recognition. This is a dataset of competitive diving, with few noticeable biases for static representations. RESOUND is used to quantify these biases, showing that they are much smaller than in previous action recognition datasets. Second, by formulating an optimization problem to sample new datasets, with minimal representation bias, from the existing ones.

Overall, the paper makes four main contributions. First, it formalizes the notion of representation bias and provides some theoretical justification for how to measure it. Second, it introduces a new dataset collection procedure, RESOUND, that (1) forces vision researchers to establish controls for vision tasks (the representation families against which bias is computed), and (2) objectively quantifies representation biases. Third, it demonstrates the effectiveness of RESOUND, by introducing a new action recognition dataset, Diving48, that is shown to drastically reduce several biases of previous datasets. Fourth, the RESOUND procedure is also used to sample existing datasets to reduce bias.

2 Related Work

Action recognition has many possible sources of bias. Early datasets (Weizmann [2], KTH [14]) were collected in controlled environments, minimizing static biases. Nevertheless, most classes were distinguishable at the short-term motion level. These datasets were also too small for training deep CNNs. Modern datasets, such as UCF101 [18], HMDB51 [10], ActivityNet [3] and Kinetics [8] are much larger in size and numbers of classes. However, they have strong static biases that enable static representations to perform surprisingly well. For example, the RGB stream of Temporal Segment Network [22] with 3 frames of input achieves 85.1% accuracy on UCF101.

The idea that biases of datasets can lead to erroneous conclusions on the merit of different representations is not new. It has motivated efforts in fine grained classification, where classes are defined within a narrow domain, e.g. birds [21], dogs [9], or shoes [23]. This eliminates many of the biases present in more general problems. Large scale generic object recognition datasets, such as ImageNet, account for this through a mix of breadth and depth, i.e. by including large numbers of classes but making subsets of them fine-grained. For action recognition, the effect of biases on the evaluation of different representations is more subtle. A general rule is that representations in the higher levels of the temporal discrimination hierarchy are needed for finer grained video recognition. However, it does not suffice to consider fine grained recognition problems. As illustrated by Weizmann and KTH, short-term motion biases could suffice for class discrimination, even when static biases are eliminated.

A popular fine-grained action recognition dataset is the MPII-Cooking Activities Dataset [13]. It has some controls for static and motion bias, by capturing all videos in the same kitchen, using a static camera, and focusing on the hands of food preparers. However, because it focuses on short-term activities, such as "putting on" vs "removing" a lid, or various forms of cutting food, it has strong short-term motion biases. Hence, it cannot be used to investigate the importance of representations at higher levels of the temporal discrimination hierarchy. Furthermore, because different actions classes (e.g. "cutting" vs. "opening/closing") are by definition associated with different objects it has a non-trivial amount of object bias. This is unlike the now proposed Diving48 dataset, where all classes have identical objects (divers) and similar forms of short-term motion.

Recently, [15] analyzed action recognition by considering multiple datasets and algorithms and pointed out future directions for algorithm design. In this work, we are more focused on the process of dataset assembly. This is a new idea, we are not aware of any dataset with explicit controls for representation bias. While it is expected that dataset authors would consider the issue and try to control for some biases, it is not known what these are, and the biases have not been quantified. In fact, we are not aware of any previous attempt to develop an objective and replicable procedure to quantify and minimize dataset bias, such as RESOUND, or a dataset that with objectively quantified biases, such as Diving48.

3 Representation Bias

In this section, we introduce the notion of representation bias and discuss how it can be avoided.

3.1 Dataset Bias

While many datasets have been assembled for computer vision, there has been limited progress in establishing an objective and quantitative characterization of them. Over the years, vision researchers have grown a healthy skepticism of "good dataset performance". It has long been known that an algorithm that performs well in a given dataset, does not necessarily perform well on others. This is denoted dataset bias [19]. In recent years, significant effort has been devoted to combating such bias, with significant success.

These advances have been guided by well known principles in statistics. This is because a CNN learned with cross-entropy loss is a maximum likelihood (ML) estimator $\hat{\theta}$ of ground-truth parameters θ. Consider in this discussion a simpler problem of estimating the head probability p in a coin toss. Given a dataset $\mathcal{D} = \{x_1, \ldots, x_n\}$ of samples from n independent Bernoulli random variables X_i of probability p, the ML estimator is well known to be the sample mean

$$\hat{p}_{\mathrm{ML}} = \frac{1}{n} \sum_i x_i. \tag{1}$$

Over the years, statisticians have developed many measures of goodness of such algorithms. The most commonly used are bias and variance

$$\mathrm{Bias}(\hat{p}_{\mathrm{ML}}) = \mathbb{E}[\hat{p}_{\mathrm{ML}}] - p \tag{2}$$

$$\mathrm{Var}(\hat{p}_{\mathrm{ML}}) = \mathbb{E}[(\hat{p}_{\mathrm{ML}} - \mathbb{E}[\hat{p}_{\mathrm{ML}}])^2]. \tag{3}$$

The algorithm of (1) is known to be unbiased and have variance that decreases as the dataset size n grows, according to $\mathrm{Var}(\hat{p}_{\mathrm{ML}}) = \frac{1}{n} p(1-p)$. Similar but more complex formulas can be derived for many ML algorithms, including CNN learning. These results justify the common practice of evaluation on multiple datasets. If the algorithm is an unbiased estimate of the optimal algorithm it will, on average, produce optimal results. If is also has low variance, it produces close to optimal results when applied to any dataset. Hence, when evaluated over a few datasets, the algorithm is likely to beat other algorithms and become the state of the art.

Note that the common definition of "dataset bias" [19], i.e. that an algorithm performs well on dataset A but not on dataset B, simply means that the algorithm has large variance. Since variance decreases with dataset size n, it has always been known that, to avoid it, datasets should be "large enough". The extensive data collection efforts of the recent past have produced some more objective rules of thumb, e.g. "1,000 examples per class," that appear to suffice to control the variance of current CNN models.

3.2 Representation Bias

Unfortunately, dataset bias is not the only bias that affects vision. A second, and more subtle, type of bias is *representation bias*. To understand this, we return to the coin toss example. For most coins in the world, the probability of heads is $p = 0.5$. However, it is possible that a dataset researcher would only have access to biased coins, say with $p = 0.3$. By using the algorithm of (1) to estimate p, with a large enough n, the researcher would eventually conclude that $p = 0.3$. Furthermore, using (2)–(3), he would conclude that there is no dataset bias and announce to the world that $p = 0.3$. Note that there is nothing wrong with this practice, except the final conclusion that there is something universal about $p = 0.3$. On the contrary, because the scientist used a biased dataset, he obtained a biased response.

The important observation is that standard dataset collection practices, such as "make n larger," will not solve the problem. These practices address dataset bias, which is a property of the representation. On the other hand, representation bias is a property of the dataset. While evaluating the representation ϕ on multiple (or larger) datasets $\mathcal{D}_i$ is an effective way to detect dataset bias, representation bias can only be detected by comparing the performance of multiple representations ϕ_i on the dataset $\mathcal{D}$. More importantly, the two are unrelated, in the sense that a representation ϕ may be unbiased towards a dataset $\mathcal{D}$, even when $\mathcal{D}$ has a strong bias for ϕ. It follows that standard evaluation practices, which mostly measure dataset bias, fail to guarantee that their conclusions are not tainted by representation bias.

This problem is difficult to avoid in computer vision, where biases can be very subtle. For example, a single object in the background could give away the class of a video. It is certainly possible to assemble datasets of video classes that can be discriminated by the presence or absence of certain objects. This does not mean that object recognition is sufficient for video classification. Only that the datasets are biased towards object-based representations. To avoid this problem, the datasets must be well calibrated.

3.3 Calibrated Datasets

Representation is a mathematical characterization of some property of the visual world. For example, optical flow is a representation of motion. A representation ϕ can be used to design many algorithms γ_ϕ to accomplish any task of interest, e.g. different algorithms that use optical flow to classify video. A representation family $\mathcal{R}$ is a set of representations that share some property. For example, the family of static representations includes all representations for visual properties of single images, i.e. representations that do not account for motion.

Let $\mathcal{M}(\mathcal{D}, \gamma)$ be a measure of performance, e.g. classification accuracy, of algorithm γ on dataset $\mathcal{D}$. The performance of the representation ϕ is defined as

$$\mathcal{M}(\mathcal{D}, \phi) = \max_{\gamma_\phi} \mathcal{M}(\mathcal{D}, \gamma_\phi) \tag{4}$$

where the max is taken over all algorithms based on the representation. Representation bias reflects the fact that a dataset $\mathcal{D}$ has a preference for some representation ϕ, i.e. $\mathcal{M}(\mathcal{D}, \phi)$ is high.

The fact that a dataset has a preference for ϕ is not necessarily good or bad. In fact, all datasets are expected to be biased for the *ground truth representation,* (GTR) ϕ_g, the representation that is truly needed to solve the vision problem. A dataset $\mathcal{D}$ is said to be *well calibrated* if this representation has the best performance

$$\phi_g = \arg\max_{\phi} \mathcal{M}(\mathcal{D}, \phi) \tag{5}$$

and the maximum is *unique*, i.e.

$$\mathcal{M}(\mathcal{D}, \phi) < \mathcal{M}(\mathcal{D}, \phi_g) \quad \forall \phi \neq \phi_g. \tag{6}$$

In general, the GTR is unknown. A commonly used proxy in vision is the *state-of-the-art* (SoA) representation

$$\phi_{soa} = \arg\max_{\phi \in \mathcal{S}} \mathcal{M}(\mathcal{D}, \phi) \tag{7}$$

where $\mathcal{S}$ is a finite set of representations proposed in the literature. If the dataset $\mathcal{D}$ is well calibrated, ϕ_{soa} will converge to ϕ_g as $\mathcal{S}$ expands, i.e. as more representations are tested. This is not guaranteed when $\mathcal{D}$ is not well calibrated. Unfortunately, it is usually impossible to know if this is the case. An alternative is to measure of bias.

3.4 Measuring Representation Bias

While the best possible performance on a dataset, e.g. the Bayes error of a classification task, is usually impossible to determine, the contrary holds for the worst performance. For classification, this corresponds to the random assignment of examples to classes, or "chance level performance". This is denoted as

$$\mathcal{M}_{rnd} = \min_{\phi} \mathcal{M}(\mathcal{D}, \phi). \tag{8}$$

The bias of a dataset $\mathcal{D}$ for a representation ϕ is defined as

$$\mathcal{B}(\mathcal{D}, \phi) = \log \frac{\mathcal{M}(\mathcal{D}, \phi)}{\mathcal{M}_{rnd}}. \tag{9}$$

When bias is zero, the representation has chance level performance and the dataset is said to be unbiased for the representation.

A dataset for which (5) holds but (6) does not, since there is a family of representations $\mathcal{R}$ such that $\mathcal{M}(\mathcal{D}, \phi) = \mathcal{M}(\mathcal{D}, \phi_g)\ \forall \phi \in \mathcal{R}$, can be made well calibrated by addition of data $\mathcal{D}'$ that reduces the bias towards the representations in $\mathcal{R}$, i.e. $\mathcal{B}(\mathcal{D} \cup \mathcal{D}', \phi) < \mathcal{B}(\mathcal{D}, \phi)\ \forall \phi \in \mathcal{R}$, while guaranteeing that (5) still holds. Similarly, a dataset can be designed to be minimally biased towards a representation family $\mathcal{R}$. This consists of selecting the dataset

$$\mathcal{D}^* = \arg \min_{\mathcal{D} \in \mathcal{T}(\phi_g)} \max_{\phi \in \mathcal{R}} \mathcal{B}(\mathcal{D}, \phi) \tag{10}$$

Algorithm 1: Representation biases.

Input : Dataset $\mathcal{D}$; representation families $\{\mathcal{R}_1, \ldots, \mathcal{R}_K\}$.
Output: Representation biases $\{b_1, \ldots, b_K\}$.

for $k = 1, \ldots, K$ **do**
  R_k = number of representations in $\mathcal{R}_k$;
  for $r = 1, \ldots, R_k$ **do**
    $M_{k,r}$ = number of algorithms based on representation $\phi_{k,r}$;
    for $m = 1, \ldots, M_{k,r}$ **do**
      $\gamma^m_{\phi_{k,r}}$: m^{th} algorithm based on $\phi_{k,r}$; Measure $\mathcal{M}(\mathcal{D}, \gamma^m_{\phi_{k,r}})$
    end
    Measure $\mathcal{M}(\mathcal{D}, \phi_{k,r})$ with (4);
    Measure bias $\mathcal{B}(\mathcal{D}, \phi_{k,r})$ with (9);
  end
  Compute $b_k = \max_r \mathcal{B}(\mathcal{D}, \phi_{k,r})$;
end

where $\mathcal{T}(\phi_g)$ is the set of datasets for which (5) holds.

Note that the constraint $\mathcal{D} \in \mathcal{T}(\phi_g)$ is somewhat redundant, since it has to hold for any valid dataset collection effort. It simply means that the dataset is an object recognition dataset or an action recognition dataset. Researchers assembling such datasets already need to make sure that they assign the highest score to the GTR for object recognition or action recognition, respectively. The main novelty of (10) is the notion that the datasets should also be minimally biased towards the family of representations $\mathcal{R}$.

3.5 Measuring Bias at the Class Level

Definition (9) can be extended to measure class-level bias. Consider a dataset of C classes. Rather than using a single classification problem to measure $\mathcal{M}(\mathcal{D}, \phi)$, C one-vs-all binary classifiers are defined. The bias for class c is then defined as

$$\mathcal{B}_c(\mathcal{D}, \phi) = \log \frac{\mathcal{M}_c(\mathcal{D}, \phi)}{\mathcal{M}_{rnd}}, \tag{11}$$

where $\mathcal{M}_c$ is the performance on the classification problem that opposes c to all other classes. To alleviate the effects of sample imbalance, performance is measured with average precision instead of classification accuracy.

4 RESOUND Dataset Collection

In general, it is impossible to guarantee that a dataset is minimally biased towards all representation families that do not contain ϕ_g. In fact, it is usually impossible to even list all such families. What is possible is to *define* a set of representation families $\mathcal{R}_i$ towards which the dataset aims to be unbiased, *measure* the bias of the dataset for at least one representation in each $\mathcal{R}_i$, and

show that the biases are smaller than previous datasets in the literature. This is denoted *REpreSentatiOn UNbiased Dataset* (RESOUND) collection. The steps taken to measure the biases of the dataset are summarized in Algorithm 1.

Two strategies are possible to implement RESOUND in practice. The first is explicit optimization, where dataset $\mathcal{D}^*$ is produced by an algorithm. This could, for example, start from an existing dataset $\mathcal{D}$ and add or eliminate examples so as to optimize (10). The second is an implicit optimization, which identifies classes likely to be unbiased with respect to the representation family $\mathcal{R}$. For example, if $\mathcal{R}$ is the family of object representations, this requires defining classes without distinguishable objects in either foreground or background. We next illustrate this by applying RESOUND to the problem of action recognition.

4.1 Explicit RESOUND

One possible strategy to assemble a K-class dataset $\mathcal{D}^*$ of minimal bias is to select K classes from an existing dataset $\mathcal{D}$. Let $\mathcal{D}$ have $C > K$ classes, i.e. a set of class labels $\mathcal{D}_y = \{d_1, \ldots, d_C\}$, where d_i denoted the i^{th} class of $\mathcal{D}$. The goal is to find the label set of $\mathcal{D}^*$ i.e. a set $\mathcal{D}_y^* = \{c_1, \ldots, c_K\}$, such that: (1) c_i are classes from $\mathcal{D}$, i.e. $c_i \in \mathcal{D}_y$; (2) c_i are mutually exclusive, $c_i \neq c_j, \forall i \neq j$; (3) $\mathcal{D}^*$ has minimal bias.

Using the class-level bias measurement of (11) then leads to the following optimization problem.

$$\mathcal{D}_y^* = \underset{c_1,\ldots,c_K \in \mathcal{D}_y}{\arg\min} \quad \sum_{k=1}^{K} \mathcal{B}_{c_k}(\mathcal{D}^*, \phi) \tag{12}$$

$$\text{subject to} \quad 1 \leq c_i \leq C; c_i \neq c_j, \quad \forall i \neq j \tag{13}$$

Since this is a combinatorial problem, a global optimum can only be achieved by exhaustive search. Furthermore, because the bias $\mathcal{B}_{c_k}(\mathcal{D}^*, \phi)$ of class c_k depends on other classes in $\mathcal{D}^*$, the biases have to be computed for each class configuration. For small values of K, the time complexity of this search is acceptable. The problem of how to scale up this process is left for future research.

4.2 Implicit RESOUND: The Diving48 Dataset

In this section, we describe the application of RESOUND in creating an action recognition dataset, Diving48. The goal of this data collection effort was to enable further study of the question "what is the right level of representation for action recognition?" The current evidence is that optical flow representations, such as the two-stream network of [17], are sufficient. However, current datasets exhibit biases that could lead to this conclusion, even if it is incorrect. By producing a dataset with no (or small) such biases, we expect to use it to investigate the importance of short-time motion vs long-term dynamics representations. Since we were not interested in the role of static cues, the dataset should be unbiased towards static representations. However, it would be too difficult to consider all

static cues. To keep the problem manageable, it was decided to emphasize the most prevalent static biases of existing datasets: *objects*, *scenes,* and *people*. For this, we considered the domain of competitive diving.

Diving is an interesting domain for the study of action recognition, for various reasons. First, there is a finite set of action (dive) classes, which are unambiguously defined and standardized by FINA [1]. Second, the dives differ in subtle sub-components, known as *elements,* that the divers perform and are graded on. This generates a very rich set of fine-grained action classes. Since some of the dives defined in [1] are rarely performed by athletes (due to their difficulty), a subset of 48 dives were selected as classes of the Diving48 Dataset. Third, and perhaps most important, diving scenes give rise to much fewer biases than other scenes commonly used for action recognition. This is because there are many different divers per competition, there are no background objects that give away the dive class, the scenes tend to be quite similar (a board, a pool, and spectators in the background) in all dives, and the divers have more or less the same static visual attributes. In this way, the diving domain addressed all biases that we had set out to eliminate. This was verified by comparing the biases of Diving48 to those of previous datasets.

Because there are many diving videos on the web, it was relatively easy to find and download a sufficient number of videos of diving platform and springboard, shot in major diving competitions. However, these event videos are usually not segmented. They are usually long videos, including hundreds of diving instances, performed by different divers, and replayed from different camera views and at different playback speeds. To ease the labeling process, the videos were automatically segmented into clips approximately one-minute-long, which were then annotated on Amazon Mechanical Turk with two major tasks. The first was to transcribe the information board that appears in each clip before the start of the dive. This contains meta information such as the diving type and difficulty score, which is used to produce ground truth for the dataset. The second was to precisely segment each diving instance, by determining the start and end video frames of the dive, and labeling the playback view and speed. Each segmentation task was assigned to 3 Turkers and a majority vote based on IOU of temporal intervals was used to reduce labeling noise. This produced 18,404 segmented dive video clips, which were used to create Diving48. A random set of 16,067 clips was selected as train set and the remaining 2,337 as test set. To avoid biases for certain competitions, the train/test split guaranteed that not all clips from the same competition were assigned into the same split.

Figure 1 shows a prefix tree that summarizes the 48 dive classes in the dataset. Each class is defined by the path from the root node to a leaf node. For example, dive 32 is defined by the sequence "Backwards take-off → 1.5 Somersault → Half Twist, with a Free body position". Note that discrimination between many of the classes requires a fine-grained representation of dynamics. For example, dive 16 and dive 18 only differ in the number of somersaults; while dive 33 and dive 34 differ only in the flight position.

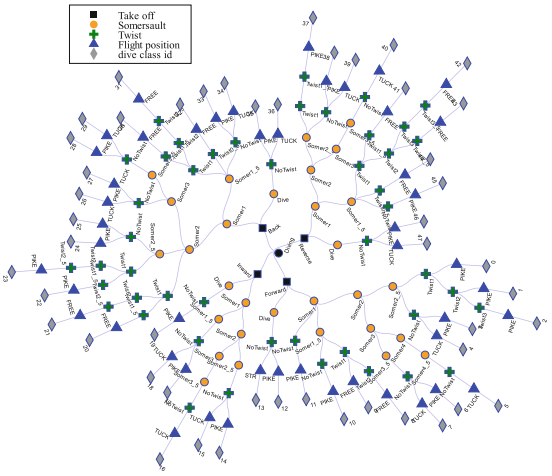

Fig. 1. Definitions of dive class in Diving48 as a prefix tree.

5 Experiments

Three sets of experiments were performed. The first was a RESOUND experiment, aimed to measure biases on existing and the proposed Diving48 dataset. The second was meant to confirm that RESOUND sampling of existing datasets can effectively produce datasets with minimal biases. The third aimed to investigate the original question of the importance of dynamic modeling for action recognition.

5.1 Datasets

The biases of Diving48 were compared to those of seven popular datasets, whose statistics are shown in Table 1. KTH [14], Hollywood2 [11] are small datasets introduced in the early history of video action recognition. They were collected in a more controlled fashion, with e.g. fixed background. HMDB51 [10] and UCF101 [18] are modern datasets with larger scale and less constrained videos. ActivityNet [3], Kinetics [8] and Charades [16] are three recent datasets, collected by crowd-sourcing. All experiments were used on the official train/test splits for each dataset. Dataset level bias is measured with (9), using accuracy as performance metric. For class level bias, average precision is used in (11).

5.2 RESOUND Experiments

A set of RESOUND experiments were performed to compare the representation biases of Diving48 and existing datasets. Three static biases were considered in Algorithm 1, using three representation families $\mathcal{R} = \{\mathcal{R}_{object}, \mathcal{R}_{scene}, \mathcal{R}_{people}\}$. For each family, we considered a single representation—CNN features, and a

Table 1. Statistics and biases of various video action recognition datasets.

Dataset	#samples	#classes	Avg. #frames	$\mathcal{B}(\mathcal{D}, \phi_{object})$	$\mathcal{B}(\mathcal{D}, \phi_{scene})$	$\mathcal{B}(\mathcal{D}, \phi_{people})$	$\mathcal{M}_{rnd}$
KTH	599	6	482.7	**1.47**	1.39	1.47	0.17
Hollywood2	823	10	345.2	1.69	1.61	1.64	0.10
HMDB51	6766	51	96.6	3.16	2.92	2.98	0.020
UCF101	13320	101	187.3	4.33	4.09	4.23	0.010
ActivityNet	28108	200	1365.5	3.69	3.37	3.49	0.0050
Kinetics	429256	400	279.1	4.51	3.96	4.31	0.0025
Charades	99618	157	310.0	2.12	2.01	2.04	0.0063
Diving48	18404	48	159.6	**1.48**	**1.26**	**1.44**	0.021

single algorithm—ResNet50 [6]. The networks varied on how they were trained: ϕ_{object} was trained on the 1,000 object classes of ImageNet [4], ϕ_{scene} on the 365 scene classes of the Places365 scene classification dataset [25], and ϕ_{people} on the 204 classes of people attributes of the COCO-attributes dataset [12].

These networks were used, without fine-tuning, to measure the representation bias of each dataset. A 2,048 dimensional feature vector was extracted at the penultimate layer, per video frame. A linear classifier was then trained with cross-entropy loss to perform action recognition, using the feature vectors extracted from each action class. It was then applied to 25 frames drawn uniformly from each test clip, and the prediction scores were averaged to obtain a clip-level score. Finally, the clips were assigned to the class of largest score. The resulting classification rates were used to compute the bias $\mathcal{B}(\mathcal{D}, \phi)$, according to (9).

The biases of all datasets are listed in Table 1. Note that bias is a logarithmic measure and small variations of bias can mean non-trivial differences in recognition accuracy. A few observations can be made from the table. First, all existing datasets have much larger biases than Diving48. This suggests that the latter is more suitable for studying the importance of dynamics in action recognition. Second, all datasets have stronger bias for objects, then people, and then scenes. Interestingly, the three biases are *similar* for each dataset. This suggests that there is interdependency between the biases of any dataset. Third, all biases vary significantly *across* datasets. Clearly, the amount of bias does not appear to be mitigated by dataset size: the largest dataset (Kinetics) is also the most biased. This shows that, while a good strategy to mitigate dataset bias, simply increasing dataset size is not a solution for the problem of representation bias. On the other hand, a small dataset does not guarantee low representation bias. For example, UCF101 is relatively small but has the second largest average bias, and is the dataset most strongly biased to scene representations. Fourth, bias appears to be positively correlated with the number of classes. This makes intuitive sense. Note, however, that this effect is dependent on how the dataset is assembled. For example, HMDB51 has a number of classes similar to Diving48, but much larger object bias. In fact, Diving48 has bias equivalent to that of the 6 class KTH dataset. Nevertheless, the positive correlation with number of classes suggests that representation bias *will become a more important problem*

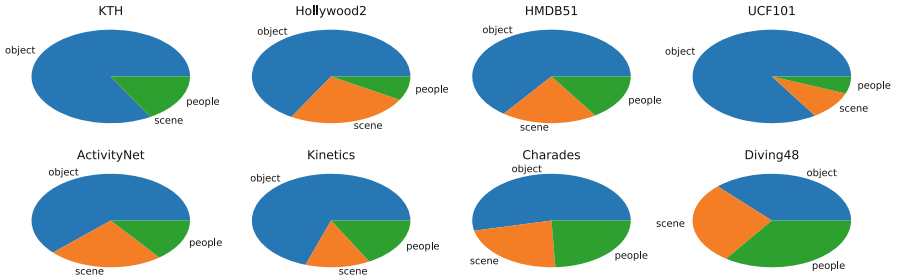

Fig. 2. Distribution of the dominant class bias $\phi^c_{r^*}$.

as datasets grow. Certainly, some of the most recent datasets, e.g. ActivityNet and Kinetics, have some of the largest amounts of representation bias.

5.3 Class-Level Dominant Bias

We next evaluate biases at the class level, using (11). For each class c, a **dominant bias** $\phi^c_{r^*}$ is identified with

$$r^* = \arg\max_r \ \mathcal{B}_c(\mathcal{D}, \phi_r). \tag{14}$$

Figure 2 summarizes the distribution of this dominant bias on each dataset of Table 1. It is clear that most classes of all datasets are dominantly biased for object representation. However, different datasets have different bias properties. For example, KTH classes are more biased to people representation than scene representation, while this is reverse for Hollywood2. Diving48 has the most uniform distribution. These plots can be used to derive guidelines on how to mitigate the biases of the different datasets. For example, object bias can be decreased by augmenting all the classes where it is dominant with videos where the objects do not appear, have a larger diversity of appearance and/or motion, etc.

5.4 Explicit RESOUND

We have also investigated the possibility of creating unbiased datasets from existing biased datasets, using the explicit RESOUND procedure of (13). Due to the large computational complexity, we have so far only used $K = 3$. This is geared more to test the feasibility of the approach than a practical solution, which will require the development of special purpose optimization algorithms. To test the effectiveness of explicit RESOUND sampling, the biases of the resulting datasets were compared to those obtained by random sampling. Table 2 shows that in all cases, the datasets produced by explicit RESOUND have significantly smaller biases than those produced by random sampling. And the optimization results make intuitive sense, e.g. for ActivityNet, the selected classes are {"Hanging wallpaper", "Installing carpet", "Painting"}, which are all household actions.

Table 2. Explicit RESOUND ($\mathcal{D}_y^*$) biases after sampling. Results of random sampling ($\mathcal{D}_{rand}$) were evaluated on 10 runs and are reported as mean ± std.

Dataset	$\phi = \phi_{object}$		$\phi = \phi_{scene}$		$\phi = \phi_{people}$	
	$\mathcal{B}(\mathcal{D}_y^*, \phi)$	$\mathcal{B}(\mathcal{D}_{rand}, \phi)$	$\mathcal{B}(\mathcal{D}_y^*, \phi)$	$\mathcal{B}(\mathcal{D}_{rand}, \phi)$	$\mathcal{B}(\mathcal{D}_y^*, \phi)$	$\mathcal{B}(\mathcal{D}_{rand}, \phi)$
KTH	0.39	0.99 ± 0.09	0.29	0.80 ± 0.17	0.44	0.86 ± 0.20
Hollywood2	0.44	0.86 ± 0.07	0.28	0.66 ± 0.13	0.33	0.68 ± 0.08
HMDB51	0.00	0.82 ± 0.54	0.00	0.99 ± 0.05	0.00	0.90 ± 0.13
UCF101	0.55	1.08 ± 0.02	0.65	1.02 ± 0.09	0.46	1.08 ± 0.02
ActivityNet	0.41	0.89 ± 0.10	0.14	0.79 ± 0.09	0.00	0.84 ± 0.11
Kinetics	0.41	1.00 ± 0.11	0.30	1.01 ± 0.08	0.33	0.94 ± 0.11
Charades	0.00	0.62 ± 0.20	0.00	0.67 ± 0.18	0.00	0.73 ± 0.14

Table 3. Recognition accuracy on Diving48.

TSN (RGB)	TSN (Flow)	TSN (RGB + Flow)	C3D (L = 8)	C3D (L = 16)	C3D(L = 32)	C3D(L = 64)
16.77	19.64	20.28	11.51	16.43	21.01	**27.60**

5.5 Classification with Dynamics

We finish by using Diving48 to investigate the importance of dynamics for action recognition. The goal was not to introduce new algorithms but to rely on off-the-shelf models of dynamics. Existing models for this evaluation include TSN [22] and the C3D [7]. For C3D, varying number of frames L is an objective measure of the extent of dynamics modeling. We set $L = 8, 16, 32$ and 64. The action recognition performance on Diving48 is shown in Table 3. First, the best performing C3D model with the largest extent of dynamics modeling achieves the best result, verifying that Diving48 is more than flow modeling. Second, the C3D results improve monotonically with L, showing that a moderate level of dynamics modeling is required to achieve good performance on this dataset. Nevertheless, the best overall performance (27.60%) is still fairly low. This shows that research is needed on more sophisticated representations of dynamics.

6 Conclusion

In this paper, we have introduced the concepts of well calibrated datasets and representation bias, and the RESOUND algorithm to objectively quantify the representation biases of a dataset. An instantiation of RESOUND in its explicit optimization form was used to sample existing datasets, so as to assemble new datasets with smaller biases. Another instantiation of RESOUND was used to compare the static representation bias of a new action recognition dataset, Diving48, to those in the literature. This showed that existing datasets have too much bias for static representations to meaningfully evaluate the role of dynamics in action recognition. Diving48, which was shown to have much smaller biases,

is a better candidate for such studies. Preliminary classification results, with static representations and 3D CNNs, indicate that modeling of dynamics can indeed be important for action recognition. We hope that this work, and the proposed dataset, will inspire interest in action recognition tasks without static bias, as well as research in models of video dynamics. We also hope that procedures like RESOUND will become more prevalent in vision, enabling (1) more scientific approaches to dataset collection, and (2) control over factors that can undermine the conclusions derived from vision experiments.

References

1. Fédération internationale de natation. http://www.fina.org/
2. Bregonzio, M., Gong, S., Xiang, T.: Recognising action as clouds of space-time interest points. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 1948–1955. IEEE (2009)
3. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: ActivityNet: a large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–970 (2015)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 248–255. IEEE (2009)
5. Donahue, J., et al.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2625–2634 (2015)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
7. Ji, S., Xu, W., Yang, M., Yu, K.: 3D convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 221–231 (2013)
8. Kay, W., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
9. Khosla, A., Jayadevaprakash, N., Yao, B., Li, F.F.: Novel dataset for fine-grained image categorization: stanford dogs. In: Proceedings of CVPR Workshop on Fine-Grained Visual Categorization (FGVC), vol. 2, p. 1 (2011)
10. Kuehne, H., Jhuang, H., Stiefelhagen, R., Serre, T.: HMDB51: a large video database for human motion recognition. In: Nagel, W., Kroner, D., Resch, M. (eds.) High Performance Computing in Science and Engineering '12, pp. 571–582. Springer, Berlin (2013). https://doi.org/10.1007/978-3-642-33374-3_41
11. Marszałek, M., Laptev, I., Schmid, C.: Actions in context. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
12. Patterson, G., Hays, J.: COCO attributes: attributes for people, animals, and objects. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 85–100. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_6
13. Rohrbach, M., Amin, S., Andriluka, M., Schiele, B.: A database for fine grained activity detection of cooking activities. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1194–1201. IEEE (2012)

14. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: 2004 Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol. 3, pp. 32–36. IEEE (2004)
15. Sigurdsson, G.A., Russakovsky, O., Gupta, A.: What actions are needed for understanding human actions in videos? In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2156–2165. IEEE (2017)
16. Sigurdsson, G.A., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A.: Hollywood in homes: crowdsourcing data collection for activity understanding. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 510–526. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_31
17. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems, pp. 568–576 (2014)
18. Soomro, K., Zamir, A.R., Shah, M.: UCF101: a dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
19. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1521–1528. IEEE (2011)
20. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4489–4497 (2015)
21. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD birds-200-2011 dataset. Technical report CNS-TR-2011-001, California Institute of Technology (2011)
22. Wang, L., et al.: Temporal segment networks: towards good practices for deep action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_2
23. Yu, A., Grauman, K.: Fine-grained visual comparisons with local learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 192–199 (2014)
24. Ng, J.Y.-H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: deep networks for video classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4694–4702 (2015)
25. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: a 10 million image database for scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. **40**, 1452–1464 (2017)